Università degli Studi di Palermo

dipartimento di ingegneria unipa

UNIVERSITY OF PALERMO

DEPARTMENT OF ENGINEERING
PH.D. IN INFORMATION AND COMMUNICATION TECHNOLOGIES
XXXVI CYCLE

PH.D. THESIS

Digital Transformation in the Public Sector: A Model, An Agile Paradigm And Collaborative Software

Gennaro Grimaudo

TUTOR	CO-TUTOR
Prof. Raffaele Giancarlo	**Prof. Paolo Ciancarini**

June 2024

I dedicate this Dissertation:

- To my Family members, for their support, and to whom I have taken away the most precious resource I dispose of, which is the time;

- To my dreams, to whom I always look with unceasing determination;

- To myself, in general, for the sacrifice and courage invested in this professional path of growth.

Abstract

Digital Transformation is intended as the process of integrating digital technologies and solutions into all aspects of the activities of an organization, whether public or private. Here we concentrate on public organizations. Although there is a sizable amount of scholarly and regulatory papers, there are great margins of ambiguity about what it is, which technical contributions Computer Science can give for its success, and which software tools should be used to facilitate that process. In particular, although Agile technologies are perceived as the most suitable for software project management in that area, convincing general approaches to their application in the context of public administrations are missing. The first three parts of this Dissertation address those issues, in a coherent way. The last part offers conclusions and future avenues of research. A highlight of the first three parts follows.

- **Part I**. Based on an extensive Literature review, we provide the first model general enough to capture the main ingredients of Digital Transformation in the Public Sector (DT, for short), from a Computer Science point of view. The validity of the model is established through the discussion of DT initiatives implemented by two international cities, i.e., Barcelona and Chicago, in terms of technological, inclusive, and organizational responsiveness aspects.

- **Part II**. From part I, it emerges that there are very few examples of Agile project management in the context of public organizations for DT. Based on this fact, the first general paradigm for the adoption of the Agile Scrum methodology in the Public Administration context, referred to as Scrum@PA, is provided. The validity of our Agile methodological proposal is demonstrated by showing that Scrum@IMI, a quite successful Agile methodology developed by the Information Technology Institute for the city of Barcelona, can be a special case of Scrum@PA.

- **Part III**. From parts I and II, there emerges the need for better collaboration among the parties involved in the implementation of a DT project for Public Administration. Here, we provide a collaborative integrated development environment, referred to as Compositional Agile System, in which the main

Scrum@PA professional figures, can collaborate to design and implement digital services that are more efficient, transparent and tailored to the needs of citizens. The validity of this environment is demonstrated by conducting an ex-post analysis of an already implemented Italian digital public service, using the environment as a quality controller of the developed software.

Preface

Digital Transformation refers to the process by which organizations adopt and integrate advanced digital technologies to significantly improve their operations, processes and services. In the context of the Public Sector, Digital Transformation refers to the strategic adoption of digital technologies to improve public service delivery, make administrative processes more efficient, and promote greater transparency and citizen participation. This evolution involves, firstly, the transformation of traditional processes to streamline administrative activities, reduce response times, and minimize human error, and secondly, the implementation of solutions based on innovative digital technologies, such as cloud solutions to reduce infrastructure costs and improve the scalability and flexibility of services, and Big Data and Analytics technologies to use advanced analytics to extract meaningful information from data, enabling better evidence-based planning and decision-making. In our initial experience, we analyzed some 5,880 papers, divided among three digital libraries, i.e., ACM, IEEE, and Google Scholar, with the expectation that we would end up with good basic definitions of what is Digital Transformation, how it should be used in the context of public organizations, and what software tools should be used to successfully achieve the desired transformation goals. Unfortunately, and to the best our knowledge, what we have found are the following.

- The absence of a rigorous model that takes into account the State of the Art of Digital Transformation in the international context, and that enucleates the main areas of interest for Computer Science.

- Although Agile methodologies for project management are referred to as ideal in such a context, there is a lack of concrete indications on how they can be used in Public Administration, apart from sporadic examples.

- Given the use of Agile in Public Administration, there is a lack of software tools for teams of professional figures, involved in the development of IT projects, that facilitate collaboration and communication among team members and simultaneously provide some functionality to monitor and improve the quality of implemented services.

Based on the above, the contribution of this Dissertation is to provide consistent, if initial, answers to the problems posed.

In particular, in the first part of this Dissertation, Chapter 1 introduces the problem of Digital Transformation in the Public Administration sector, offering a first model for the public stakeholders to address their digital transition strategy by identifying the main ingredients of Digital Transformation, in terms of four knowledge domains, i.e., Data, Technologies, People and Process, extending them through some of the technological, inclusive and organizational aspects of responsiveness. In addition, the validity of our model is established via a discussion of Digital Transformation initiatives implemented by two specific case studies, i.e., Barcelona and Chicago, in terms of the responsiveness aspects mentioned earlier. Chapter 2 reports the State of the Art about the main insights on the knowledge domains identified in Chapter 1.

In the second part of this Dissertation, Chapter 3, starts from a comparison with traditional project management methodologies, to describe the Agile methodology most widely used in the context of Public Administration, i.e., Scrum. This methodology describes the process, involved professional figures, their main responsibilities, and experiences of use, together with a description of some useful Agile tools to support such processes. Then, in Chapter 4, we address the problem of the transition from traditional to Agile methodologies about their adoption in Public Administrations. To this end, we provide a coherent set of recommendations, first generic and then specialized, specific to Digital Transformation. Next, we introduce a taxonomy of categories of professional figures and show how these can be placed within an Agile methodology, specifically a specialization of the standard Scrum methodology, i.e., Scrum@PA, which is our proposed Agile methodology. For this latter, we show how Agile IT international experiences can be a special case of Scrum@PA. To show how our proposal can be scaled, we compare the SAFe framework, in its Essential configuration, with our Scrum@PA framework. We show how there is a good overlap between them, and how our proposal is more detailed about the professional figures involved and their main interactions.

In the third part of this Dissertation, Chapter 5, starting with a review of a selected set of basic Agile collaborative tools, free and open source, and commercial ones, we conduct an assessment of sustainability and interest in free and open-source solutions, identifying a subset of these tools that we consider as the building blocks of an integrated development environment in which teams and public stakeholders can collaborate to develop digital services. Based on these blocks, in Chapter 6 we present the Compositional Agile System (CAS) platform, showing how it can facilitate collaboration and foster transparency, efficiency, and quality of services developed from it. To this end, first, we present high level tasks, activities and tools supporting their realization. Then, we show how the high level tasks and activities, mentioned earlier, can be mapped within the Scrum methodology, considering the main professional figures involved, and detailing them through writing user stories. Next, we provide detailed information regarding the CAS platform, as well as extend the thinking done on Scrum to the main figures of Scrum@PA. Finally, to demonstrate the validity of the project, we conducted an ex-post analysis on an already developed public administration project, showing some of the steps within the development cycle, through the use of some of the tools provided by the CAS platform.

Finally, in the fourth part of this Dissertation, in Chapter 7, we present the conclusions common to the first three parts of this Dissertation, in which possible future developments of what was presented are provided.

Acknowledgments

Here we are. It is time to thank those who have allowed me, in different ways, to write some kind reflections on this page. From an initial balance of these three years, I do not disown that there have been difficult times, disappointments and renunciations, but also joys and satisfactions, both professionally and socially. This project would not have been possible without the support of many people.

First of all, I would like to thank **Prof. Raffaele Giancarlo**, full professor of the University of Palermo, tutor and mentor, who guided me, through this challenging and dense course of study, procuring advice, insights and a multitude of corrections to the various revisions of this Dissertation, often helping me to make sense of the confusion, and sometimes discouragement, over the various difficulties I encountered.

Not in chronological order, and certainly not in terms of importance, I would also like to thank the following people.

- My wife **Valentina** and my son **Giovan Battista Domenico** because during this period I could not have done what I did without their support;

- **Prof. Ilenia Tinnirello**, full professor of the University of Palermo, and coordinator of the Doctoral Area in Information and Communication Technologies (XXXVI cycle), for her cordiality and support in various administrative situations related to the doctoral period, ranging from the study period at the ALMA MATER STUDIORUM - University of Bologna to the extension of the term of the said study period;

- **Prof. Paolo Ciancarini**, co-tutor, full professor of the Department of Computer Science - Science and Engineering of the ALMA MATER STUDIORUM - University of Bologna, for first allowing me a collaboration to acquire specialized skills in Software Engineering and advanced techniques of software systems development, in remote mode, and then providing me with valuable suggestions that have improved the quality of this Dissertation;

- **Professors Simona Ester Rombo and Marinella Sciortino**, full professors of the University of Palermo, for acquiring specialized skills, respectively, the former in the subjects of Software Engineering and Big Data Management, and the latter in Information Theory and Data Compression;

- **Prof. Davide Taibi**, Senior Researcher of the Institute for Educational Technologies at the National Research Council of Italy (ITD CNR), to whom I am grateful for intriguing me about Open Data Ecosystems;
- **Dr. Tian Cheng Xia**, friend and student of the Master's Degree in Artificial Intelligence at the ALMA MATER STUDIORUM - University of Bologna, for his valuable collaboration in the set-up of the Compositional Agile System (CAS) platform to which we have made some modifications and several improvements;
- several fellow Ph.D. students who, in various capacities, supported me through various difficulties I encountered, such as **Dr. Mariella Bonomo (ph.D.)**, **Dr. Armando La Placa**, and **Dr. Antonino Pagano**, for making, at various times, this learning period a little more enjoyable.

Thank you all for the sake you have shown me.

Always grateful,

Gennaro.

Contents

List of Figures

List of Tables

Part I

A Model for Computer Scientists of Digital Transformation in the Public Sector

Motivation

We all are citizens of a digital era, which offers new possibilities, new rights, and new duties [1]. Digital citizens use ICT technologies to communicate, access information, and participate in social, economic, and political activities. The impact that the advent of this new era has had is not limited to citizens, but also involves public administrations (PAs, for short), being directed to be more and more digital in how they work, relate, and interact with citizens. As a matter of fact, a Digital Transformation (DT, for short) in the PAs is taking place. To date, there is no formally accepted definition of such a transformation: as pointed out in [2], there can actually be many of them. In the context of this work, we use the definition below that attempts to summarise most of them.

DT is the process of integrating digital technologies and solutions into all aspects of the activities of an organization, whether public or private. In turn, such an integration, in particular in the Public Sector, implies deep changes in organization and management, in order to account for regulations as well as for a citizen-centric methodology. For those reasons, it can be a complex, never-ending, and often discouraging process.

The pursuit of DT represents a cultural, organizational, social, legal and technological shift that leads organizations to initiate a change in the way they operate and relate to their users, trying to be more and more responsive to their needs. It must have an associated strategy, focusing on creating the capabilities within the organization to take full advantage of the opportunities of new technologies and their impact in the most useful and innovative manner, keeping in mind that the private and Public Sectors differ. Indeed, the technologies used in the private sector cannot be applied immediately in the public one without an analysis of possible differences of impact [3]. This is due to the fact that DT in the private sector concerns how it impacts employees and how digital technologies and processes improve the productivity and quality of the products offered to customers. In the Public Sector, DT has a different scope because it impacts not only the citizens, but also the governing and administrative processes, and the nature itself of the social contract [4].

From now on, unless otherwise specified, in this Dissertation DT refers specifically to the Public Sector.

We point out that, within the Public Sector, we include public administrations such as municipalities, national governments, and other governmental bodies. We exclude other specific Public Sector institutions such as the military and homeland defence organizations, educational organizations like universities, or health organizations like hospitals. Our primary goal is to explore how these public administrations can leverage digital technologies to improve their services, operations, and interactions with their citizens, becoming more responsive to their changing needs.

As DT projects are implemented, citizens become eager to access digital services to support their activities and lives. This requires public institutions to ensure that their digital solutions are user-friendly, secure, and accessible to all citizens [5]. Several institutions and administrations in the Public Sector are exploring the opportunities offered by Digital Transformation technologies to enhance their organizational flexibility necessary to adapt to changing contexts and meet new government and citizens demands [6, 7]. More in general, DT has become an increasingly pressing issue in recent years, with the growing need to modernize government services and improve their efficiency and transparency with respect to the needs of citizens.

It is not surprising that there are several areas in this context where Computer Science can have a major impact, while receiving further stimuli for its growth. One of them is the development and implementation of digital infrastructures to support specifically the Public Sector, with a range of topics that goes from privacy and security aspects of the data and of their processing to smart cities [8]. This includes creating systems for data collection, storage, and analysis [9], as well as building networks and platforms for communication and collaboration among different public institutions [10]. Another one is the study regarding how to make government data more accessible, transparent, and useful for the citizens, their administrations, and other stakeholders. This includes developing standards for data sharing and interoperability, as well as creating tools and applications that enable citizens to engage with public Open Data in useful ways [11].

Therefore, DT is certainly of interest for Computer Scientists but, to the best of our knowledge of the State of the Art, the complexity of the DT and the interplay among its key aspects does not seem to be well presented to a Computer Science audience. Filling this gap in the Literature is one of the aims of this part of this Dissertation. Indeed, based on the State of the Art, we identify four key pillars that sustain a successful DT. Specifically, (open) data, ICT technologies, digital skills of citizens and public administrators, and agile processes. Those pillars are described, together with their interactions. Indeed, although ICT technologies are essential in driving any Digital Transformation, and well known to Computer Scientists, they are not enough on their own. Therefore, our aim is to present the benefits of technology and avoid the potential negative bias that may arise from a rendering of DT restricted to technology only. This is a novelty for a presentation of this area.

Another novelty is that we focus on data as the first and foremost pillar of any Digital Transformation. The case of the Public Sector involves the integration of data technology in all aspects of governance to improve efficiency, transparency, and citizen engagement. Open data are crucial for digital transformation and are an important part of public sector data management activities, as they enable the public sector to make its data publicly available to be reused and repurposed by others. Indeed, in the context of DT, the role of Open Data, characterised by accessibility, usability and transparency, represents a distinct but interconnected milestone with the myriad of other data types under PA governance. Of course, there is also an extensive literature on the use and benefits of publishing data (see e.g. [9, 12–14]). These complement and challenge traditional data management practices within the PA, as they prioritise principles such as privacy, security and data integrity, while ensuring open access. As public administrations are faced with the increasing complexity of data governance, including sensitive citizen information, proprietary data and operational datasets, Open Data emerges as a driver of innovation, efficiency and accountability. They can include information on government spending, public transportation, healthcare, education, environmental issues, and much more. Their availability has the aim to allow citizens to better understand how the government operates, and how it spends public money. This increased transparency can improve public trust, accountability, and collaboration between the government and its citizens. It also provides valuable insights to policymakers, researchers, and other stakeholders. The corresponding implementation of Open Data policies is crucial for unlocking the potential of data in the PA.

The use of ICT technologies, including smart cities and data governance of public clouds, can help to improve the delivery of public services [8]. While smart cities leverage technology to improve quality of life, reduce costs and improve the sustainability of urban areas, the data governance of public clouds can provide a cost-effective and scalable infrastructure for public administrations, allowing them to deliver and monitor their services more efficiently and effectively.

We remark that citizens must have the necessary competencies to access and use digital services [15]. This encompasses digital literacy, data literacy and online security awareness. Governments need to invest in programs that promote digital skills development, particularly for underprivileged and marginalized groups. Digital transformation is an ongoing process of integrating technology into various aspects of Society, and digital competencies of the citizens play a crucial role in driving this integration. Citizens must be able to navigate the risks associated with online activity and understand the importance of protecting their personal information.

Beyond basic digital skills, citizens should also have a strong understanding of the role of technology in the Public Sector, and the benefits it can bring to government services. This requires a strong understanding of public policy, as well as an awareness of the needs and interests of different stakeholders. An active digital citizenship also requires a willingness to collaborate and work with others. This includes engaging in online communities, collaborating on digital projects, and building partnerships with government and other stakeholders to co-create digital contents and services.

Finally, we show why we believe that any DT is a process that should exploit an agile approach, when introducing or adopting new services for citizens. Such an approach allows for rapid development and testing of new services, with continuous feedback and improvement. Since DT is an iterative process that requires continuous adaptation to the needs of the citizens and improvements in the policies of the administrators, Agile processes are the ideal “operational tool” for DT.

Chapter 1

The Main Ingredients of the Digital Transformation in the Public Sector

Abstract

Given the definition of DT provided in the Introduction to this part of this Dissertation, we present here its main ingredients in terms of four domains that emerge from an analytic reading of the papers we have considered for this research. We also discuss the interactions among them. The end result is a graph model of DT, proposed here for the first time, and that can be used as a "summary map" to describe the DT process. Moreover, the concepts and notions summarized by the model are exemplified via two paradigmatic examples: the Cities of Barcelona and Chicago. Such a choice is motivated by the fact that, although complex cities, their DT scale is well suited for the crisp identification and evaluation of the specific actions regarding their transition to digital.

1.1 Introduction

This chapter presents the main ingredients of Digital Transformation in the Public Sector, in terms of knowledge domains, and proposes a model for it. This model is extracted from a systematic literature review. The criteria by which we considered the papers are described in Section 1.2. In Section 1.3, we present our graph model. First, we describe the four knowledge domains (*nodes*), namely Data, Technology, People and Process that emerge as central from the Literature. Then, we describe the main interactions (*edges*) among these domains. Since the concept of "Responsiveness" within DT turns out to be of fundamental importance, we briefly recall the essence of this concept. By Responsiveness, we mean the ability of a PA to adapt quickly to the changing needs of its citizens. For this concept, we have identified three main aspects: technological, inclusive and organizational. At this point, after introducing this concept, in Section 1.3 we extend our graph model by linking the responsiveness aspects to the knowledge domains of our model. Then, for each of the aspects of Responsiveness described, we present their main facets. In Section 1.3, based on the responsiveness facets seen earlier, we discuss two paradigmatic examples, namely the cities of Barcelona and Chicago, mapping the initiatives taken by the respective Governments to the facets identified and discussed next.

1.2 Literature Selection

Our effort to provide a systematic homogeneous presentation of DT is based on the current State of the Art. Therefore, a first essential step is to resort to well known approaches to collect relevant papers as one would do for a Literature Review. However, while the former describes relevant papers covering the State of the Art, we use the selected papers to extract the main ingredients of DT and to propose models for it, with the addition of illustrative examples.

Our effort to provide a systematic homogeneous presentation of DT is based on the current State of the Art. Therefore, it is advisable to resort to established methods to collect relevant papers for a Systematic Literature Review (SLR). However, there are differences. Indeed, while an SLR usually describes relevant papers covering the State of the Art in order to answer to some research questions, we use the selected papers to extract the main ingredients of DT and to propose models for it, with the addition of illustrative examples. Once stated such a fundamental difference, the paper selection process proceeds in stages, each of which has an outcome in terms of papers that are analysed and selected for inclusion in this study. A synopsis of such a process, with the corresponding outcome, is provided in Table 1.1, and details now follow.

The first stage consists of a literature search in the ACM and IEEE digital libraries, in addition to Google Scholar. Given that, as stated in the Introduction, the focus is on Agile methodologies, the query term is very focused:

“**agile**” AND “**digital transformation**” AND “**public services**”

The period of time is January 2017-February 2022.

The second stage consists of an inclusion/exclusion process, with duplicate elimination. The selection criteria are as follows.

- The papers should be centered on DT in the Public Sector, with focus on software development topics and related technical aspects.

- The papers must be written in English.
- The papers must be published and available in scientific journals and conferences.

The result of this stage is 151 papers.

The third stage consists of assessing the eligibility of the 151 papers for consideration, based on a full text evaluation. They all passed such a quality check.

Following standard practices in Literature review, we have considered papers that cite the ones that are output of the third stage. This is denoted as forward snowballing. Based on relevance and full-text reading, a total of 24 more papers were selected. This is the fourth stage.

Finally, to make this part of the Dissertation accessible to a wide audience, we have included a total of 62 papers. They provide background regarding technical issues, e.g. Open Data [16], but given that this part of this Dissertation deals with Public Organisations, including Municipalities and Governments, we felt as appropriate to include also key papers, not accounted for in the scholarly literature that deal with fundamental standards and regulatory issues in DT, e.g., [17].

Table 1.1: **A Synopsis of the Literature Selection Process**. The terminology is as in the main text. The first column indicates the stage of the selection process. For each stage, the number of selected papers is indicated. Only rows 3-5 contribute to the total.

PHASE	**OUTCOME**
Search	5,881
Inclusion, Exclusion, Duplicate Elimination	151
Eligibility based on Full-Text	151
Forward Snowballing	24
Background	62
TOTAL	237

1.3 A Model, with Examples

A Graph Model for DT

With reference to Table 1.1, rows 3-4, there are 175 “core papers” that characterize this research, the other being background papers included to make its content accessible to a wide audience. We concentrate now on an examination of the former 175 papers, in order to obtain a synopsis of the computer science areas they belong to. Although, as expected, they address various aspects related to how PAs plan and implement their DT strategies, 167 out of 175 predominantly focus on one of four knowledge domains: **Data** (28 papers), **Technology** (55 papers), **People** (31 papers), and **Process** (53 papers). For completeness, the remaining 8 do not have a specific and well-determined focus, being mostly comparative studies involving various DT initiatives around the world, e.g., [18]. The mentioned domains are briefly discussed below providing first the key points characterizing them.

- **Data**.
  - **Key Takeaways.** Open Data are the main source of information in the DT process of a PA, being crucial for administrators and citizens. Their collection and management should include an interoperability framework to handle their heterogeneity in order to derive actionable insights from data to enhance public service delivery and policymaking.
  - **A First Level of Detail.** The availability of public data has changed significantly over the past decade, resulting in a greater awareness of how it is collected, represented, owned, and managed. As a consequence, the data lifecycle has changed with respect to the past [12, 19–22], posing new technical problems even to mature areas such as databases [23] and requiring new ways to design software for their management [24]. As far as this part of this Dissertation is concerned, it is important to point out that data are no longer seen as an asset to exploit for a competitive advantage, but as a social "infrastructure" that must be made available to policymakers and citizens to ensure and improve the well-being of Society, e.g., [25,26]. With this in mind, more and more PAs are making available their data to improve transparency and accountability [20, 27–31]. However, due to the heterogeneity and lack of interoperability of the data sources, major problems arise. One is how to exploit at its best the information contained in those data. Another is the realization of the sound technical principle of "only once", i.e., data collected by one administration should be available to other administrations. Scale factors make those problems even more difficult, since "data" can refer to a continent [32], nation or be sector-specific [33, 34]. In order to address the problems alluded to earlier and of which we have provided two examples, an entire data ecosystem is shaping up, ranging from infrastructures to data analysis tools and applications. Following [9], the term ecosystem is used here, instead of environment, because like real ecosystems, data ecosystems are designed in such a way to have an "evolutionary" part aimed at improving data quality levels over time. It is a node of the proposed graph model and, in what follows, we use the terms data ecosystem

and data interchangeably. Moreover, being data the source of information that powers the DT, its corresponding node is the central one in the model, as shown in Figure 1.1. Details regarding the components of such a node are presented in Section 2.2.

- **Technology**.
  - **Key Takeaways**. Smart Cities seem to be a very promising technology in this context, particularly concerning local PAs, such as Municipalities. Data Governance, i.e., a set of practices, policies, procedures and technologies designed to ensure the quality, integrity, security and accessibility of data within an organisation. In addition to ensuring regulatory compliance and risk management, Data Governance aims to optimise the use and value of corporate data. In the context of DT, the implementation of such practices requires strong technological support to ensure data security and privacy, which are crucial to instil trust in citizens and comply with regulations. For this reason, in this dissertation, we discuss next, in more detail, about these two aspects.
  - **A First Level of Detail**. The term technology refers to hardware and software systems supporting PAs in the DT process [27, 35–37]. Digital platforms that support all stages of government are in place or planned [10], since their realization is perceived as a way to increase the pace of the DT [38]. In particular, several PAs are moving to the Cloud [19, 39–41]. smart cities [22] are becoming a pillar in the DT, while blockchain technologies are also being considered, but they appear somewhat marginal at this stage [42]. Artificial intelligence is expected to plays a role, e.g, [43–48], although its impact and pervasiveness on privacy, transparency and accountability in the realm of DT is still under study [10]. Those technologies are intended to be a tool supporting decision-making in public administrations which, in turn, have the goal to provide a higher and higher quality of life for their citizens. In order to achieve this goal, in particular for limited geographic areas such as cities, ruling bodies and decision-makers are accepting more and more stimulating challenges related to the creation of complex digital models of cities, that would allow them to respond to the needs of citizens faster than in the past (e.g. [22]). Those new models must ensure privacy and security, making necessary for the PAs to possess regulations about data governance, e.g., the European Data Protection Act (GDPR) [17,49], cybersecurity technologies, e.g., the National Cyber-Security Agencies [50–52] and a flexible and modular strategy to data access and sharing, e.g., the European DECODE project [53]. Technology for the DT is the node

shown in Figure 1.1. Details regarding the components of such a node are presented in Section 2.3.

- **People**.
  - **Key Takeways.** Citizens are central for the DT. They are involved in the Co-Creation process of services and, since they are the end users of those services, they need adequate digital skills.
  - **A First Level of Detail.** As outlined in the Introduction to the first part of this Dissertation, the services provided by PAs must be considered valuable by the citizens. Such a fact has an important consequence regarding efficiency, which has had a privileged position in the design and deployment of services: the aim for it, although valuable, is by no means sufficient in generating services perceived of value to the citizens [36]. In fact, for the design of services, a people-driven delivery model is more and more the one of choice [27, 35]. Such a new model places citizen participation at the center of most service design and implementation initiatives, whose success must be evaluated by their users, namely the citizens and various kinds of decision-makers, according to their perception of the value created [54–58]. Interestingly, although the meaning of services valuable to the citizens is clear, the meaning of the apparently related term of "business value" in the digital PA is not so clear, although intuitively it relates to the provision of better service to the citizen, efficient operation of the services and the enforcement of the law [59]. It is to be pointed out that digital skills are essential tools unlocking the door to innovation, empowering individuals to effectively navigate today's interconnected and technology-driven world. These skills allow people to actively engage in the digital Society, adjust to changes, and fully capitalize on the possibilities presented by technology [15, 58, 60–63]. People for the DT is the node shown in Figure 1.1. Details regarding the components of such a node are presented in Section 2.4.
- **Process**.
  - **Key Takeaways.** Since the DT must improve the responsiveness of an organization, Agile methodologies seem to be the most adequate for software project management. Moreover, the digital maturity of a public organization also needs to be assessed, via Index and Maturity Models.
  - **A First Level of Detail.** As part of a DT strategy whose aim, as already stated, is to obtain services that are more citizen-centered, it is to be expected that new ways of process engineering are developed

and deployed [64]. Although the transition from established process engineering to new ones is not so simple [62, 65], Agile methodologies approaches seem to be the "best" candidates to support the mentioned transition [27, 40, 57, 64, 66–69]. Another crucial aspect regarding process management is the need for new ways to measure success, i.e., in terms of a meaning of "value" that is certainly application-specific, but with a rather broad spectrum. Rather than being specific on those measurements, the trend is to measure the degree of maturity achieved by the processes in the DT strategy implementation. In this regard, in the Literature used to derive our model, we find several papers proposing different Frameworks and Maturity Indexes, e.g., [70], with which stakeholders could measure the progress achieved in the digital transition of their Organizations. However, among the many available, the GovTech Maturity Index (GTMI, for short) proposed by the World Bank [71] seems to be the most reliable one. Process for the DT is the node shown in Figure 1.1. Details regarding the components of such a node are presented in Section 2.5.

Interactions Among Knowledge Domains

The papers we reviewed show that the four knowledge domains presented above have several mutual interactions and dependencies, summarized in terms of edges in the graph model in Figure 1.1. Each edge between two nodes (knowledge domains) encodes an interaction between its end-points, while the direction of the edge encodes the dependence, i.e., an edge (a, b) indicates that a depends on b with the label indicating the nature of such a dependency. Details are provided next.

- **Interactions with Data**. At the heart of our graph model there is the data ecosystem, subject to incoming and outgoing data flows. In one direction, Data is the source of information for other nodes, and in the other direction, it grows from the information it receives from other nodes. In the graph model, we encode this bi-directionality in terms of a *provide/receive* paradigm. People provide data and receive information and know-how. Processes provide business intelligence, statistics, etc, and receive data. Technologies provide tools for a better governance over the data, and receive data. It should be emphasized that this paradigm encodes well the flow of information in a data ecosystem [12].

- **Interactions with Technology**. Technologies are at the service of the citizens. Indeed, given that a needs-based holism means the reunification of government services around citizens rather than business processes [72], these

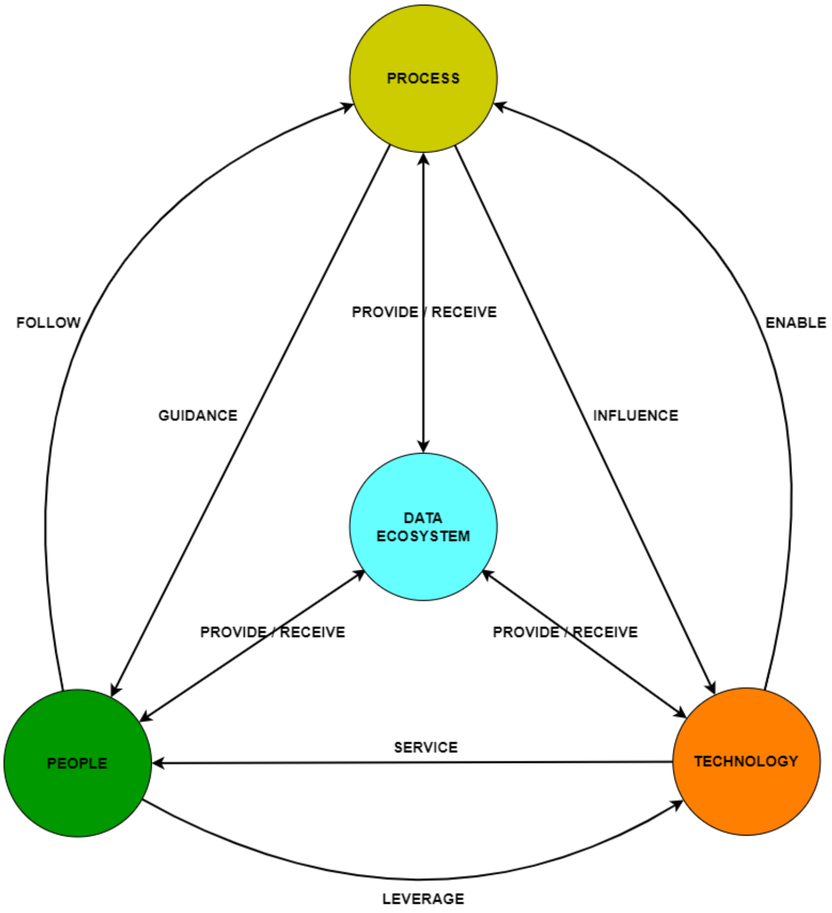


Figure 1.1: **The DT Graph Model**. Nodes represent the knowledge domains. Each edge represents interactions and dependencies between its end nodes, while the label on each edge indicates the type of relationship between its nodes, following the main text.

technologies enable it, with the end results that increase the capacity of PAs to respond to the emerging needs of citizens [60, 73]. Technologies also enable processes and facilitate their re-engineering phase [74].

- **Interactions with People**. People, who follow the processes and initiatives provided by PAs, leverage different emergent technologies [74] to enable the improvement of public processes (process re-engineering).
- **Interactions with Process**. Processes guide people, through the provision of quality digital services, to respond more effectively and promptly to the evolving needs of citizens [75]. At the same time, technological choices are often influenced by the processes, and how these are designed or re-engineered [74].

Accounting for Responsiveness

As mentioned in the Introduction to the first part of this Dissertation, one of the main goals of DT is to increase the responsiveness of an organization to the changing needs of citizens.

It is evident that in response to changes, such a goal can be reached by being able to: (a) quickly use novel technologies; (b) implement an inclusive strategy that promptly makes available new skills to citizens, administrators and policy-makers; (c) adopt flexible organizational models for the design, implementation and deployment of services. Those main aspects of responsiveness in DT can be summarized as follows: technological responsiveness, which naturally connects to the knowledge domain of **Technology**; inclusive responsiveness, which naturally connects to the knowledge domain of **People**; and organizational responsiveness, which naturally connects to the knowledge domain of **Process**. Therefore, it is felt appropriate to extend the graph model proposed here accordingly, as in Figure 1.2. We now discuss the terms we have just introduced.

- **Technological Responsiveness**. It concerns the flexibility and versatility of the solutions adopted for the collection, representation, and management of the data, together with the appropriate infrastructures to host and manage them [60, 76–79]. Those solutions must account for good levels of quality and privacy. The meaning of quality is given via a set of properties to which data should respond. Specifically: accuracy, completeness, consistency, timeliness, validity, and uniqueness [80]. For instance, we mention the ISO/IEC 25012 standard, which defines fifteen data quality characteristics and is the Italian reference standard for data ecosystem quality and interoperability. As for privacy, in addition to the meaning given to it in the domain of IT security, the solutions granting it must be compliant with current legislation, e.g., the European General Data Protection Regulation (GDPR) [17]. A particularly important and novel aspect of data processing is to account for the requirement that users must be given the option to decide who can process their data and for which purposes. As for computer architectures, to date, there are many of them supporting technological responsiveness in the DT [81] and even new ones have been proposed, although it is not clear how widespread their adoption is [82]. Moreover, the possibility of migrating from monolithic systems offering services to microservices technologies is also considered [83]: although this suggestion is somewhat isolated, the results are encouraging. From our Literature Review, and in regard to the achievement of responsiveness in the DT, it is evident that Smart Cities are the most promising technologies, while Data Governance is one of its most delicate aspects. Therefore, among the many facets characterizing technological responsiveness, we concentrate on those two, which are briefly discussed next.

- **Smart Cities**. According to the ISO/IEC [84], (but see also [85]) a Smart City is "an innovative city that uses ICT and other means to improve quality of life, the efficiency of urban operation and services, and competitiveness, while ensuring that it meets the needs of present and future generations with respect to economic, social and environmental aspects". Moreover, based on a Literature Review, including both academic papers and practical tools, a proposal regarding the key components that make a city smart has been made in [86] and validated in [87], specifically for Brazil. The components structure that comes out is hierarchical, with the top level consisting of (a) government; (b) society; (c) physical environment and (d) technology and data. A second level follows, e.g., point (d) is further detailed into (d.1) ICTs and other technologies, (d.2) data and information. A third level concludes the hierarchy, e.g., point (d.2) is further broken into (d.2.1) data management, (d.2.2) information processing, (d.2.3) information sharing and integration.

  Technology is essential for the sustainable development of a smart city (see above and [88]), in particular IoT (see for instance [89–92]). However, technology alone is not enough [93]. Indeed, starting from the fact that a difficulty for the realization of a smart city is the fragmented understanding of the interaction between Information Technologies and novel City Governance Models [94–96], changes involving Governance and Management seem to be required. For instance, project and risk management need to be changed: the realization of the infrastructural innovations required to transform a city into a smart one need to be planned carefully in order to avoid delays and over-spending [97]. Moreover, there is a need to rethink how software-intensive services are used, in order to implement a more flexible infrastructures [22, 29, 98].

  Section 2.3 is devoted to this topic, with a focus on Digital Twins [99], which is a new and promising approach to design and implement a smart city, based on the a virtual representation of its main physical city objects, including the inhabitants, that interacts with the real objects and evolves with them [99]. For completeness, we point out that Digital Twins are not a new concept, having been introduced by Greives in 2002 and have been the object of rigorous studies in an attempt to provide sound definitions and in order to identify their range of application domains [100, 101].

- **Data Governance**. For data governance, it is meant a set of processes, roles, policies, standards, and metrics useful for controlling data management [20, 29, 30, 98, 102, 103]. Via the effective and efficient management of the amount of structured and unstructured information coming from a multitude of PA processes and procedures, its goal is to transform those data into a strategic asset, serving the citizens while preserving their privacy. The issue of data governance is so important and strategic that a new professional figure is emerging: Chief Data Officer, with its role and responsibilities still being the object of study [104,105]. Certainly, such a figure should be able to manage issues regarding privacy, security, regulatory compliance, access control, and the resolution of problems caused by poor data quality across the data lifecycle [12,49,70,106]. Section 2.3 is devoted to Data Governance.

- **Inclusive Responsiveness**. It concerns how fast and broad are the cultural changes associated to the acquisition of multidisciplinary skills, ranging from digital to managerial, aimed at gaining greater awareness of the efforts of DT [60,79]. Although inclusive responsiveness can be further divided into many categories, here we concentrate on some important ones, i.e., skills development, co-creation, and leadership. We point out that skills development and co-creation are treated synergistically here, inspired by a case study regarding the city of Chicago [55], which justifies this approach.

  - **Skills and Co-Creation**. Skills development is a well-known concept that needs no further elaboration. Co-creation is a concept that strongly depends on team-building and on the digitization culture that, together with correct communication, enables the actors involved to work together to produce public services successfully [30,58,60,107]. It is a continuous improvement process, in which PAs must implement the necessary tools to successfully exploit feedback from the citizens in the evolution phase of a service [54,74,98]. This approach changes the way in which public services are evaluated, placing the users at the "center". Indeed, following earlier research regarding how to measure service quality offered by PAs [108], models and procedures for such a novel "user-centered" evaluation of public services are being investigated [109,110], together with models that identify possible areas, ranging from architectures to risk management, whose improvement would result in the deployment of better services [111]. Section 2.4 is devoted to this topic.
  - **Leadership**. It is perceived as a fundamental pillar driving DT in organizations, including the PAs (see [112] and references therein), in particular regarding the definition and implementation of mechanisms that strengthen the governance of digital and smart societies. Although, as pointed out in [113], strategy rather than technology is the key to success in DT, according to the study in [114], PAs that have reached a certain degree of maturity in the DT process are quite likely to have had the support of their managers and their involvement in the formulation of

DT strategy plans to create new public value. Therefore, IT managers and leaders still play a fundamental part regarding innovation, even with respect to DT, but they must also have a deep understanding of which organizational culture is most effective, depending on the type of innovation being implemented [115]. In addition, they must have knowledge and training in regard to a specialized set of skills on modern technologies and related cultural changes [61, 62]. Indeed, the current level of expertise, related to emerging technologies, is a barrier to the adoption of these technologies [114], while for the creation of services perceived of value critically depends on the level of competencies that managers and decision-makers have regarding technology [116]. Furthermore, managers should behave more like product owners of the new services aiming at meeting the needs of citizens [65, 74, 98, 117–120]. Yet another key to speed up DT is a coordinated policy involving National State, Local States and Municipalities [121].

Interestingly, a technological framework based on Digital Twins has been proposed to help IT governance [122]. The framework denoted Digital Twin for Governed IT Management (DG4GITM, for short), links the management of three interconnected systems: IT governance processes, IT management processes, and IT organizational assets by leveraging the technology of Knowledge Graphs and the resulting computational infrastructure. In particular, a given city virtual entity is created through an enterprise ontology "GITM Domain Ontology" that is connected to the organization via data flows to populate it with real data from the resources of the organization. This point is not the object of further discussion, since we have accounted for all the papers that cover this subject and that we have included in the Literature review.

- **Organizational Responsiveness**. It concerns the ability to adopt rapid organizational changes and to undertake new ways of operating within the PA [61, 120, 123, 124]. DT is a continuously evolving process that needs to be monitored in order to evaluate its progress and to identify directions for improvement [124]. Two key features to consider are Change Management and Frameworks and Maturity Models:

  - **Change Management**. It refers to the ability to accept innovation while producing quality services [61, 79]. In the PA context, one of the essential parts of this point is the promulgation of laws, regulations, and guidelines, which promote the use of the services offered, enabling the creation of new public value [57, 125]. There is also a corresponding technical part regarding project management. In what follows, the change management in terms of laws and regulations is best accounted for in the areas that are affected by those regulations and laws, e.g., Data. Consequently, the part of this Dissertation specifically devoted to Change Management refers to project management engineering. Section 2.5 is devoted to this topic.

  - **Frameworks and Maturity Models**. These models focus on the major technological, inclusive, and organizational elements of which a PA is composed, in order to be able to measure their performance and establish the progress achieved in the DT strategy undertaken [71, 126]. Section 2.5 is devoted to this topic.

Two Paradigmatic Examples

Each Public Organization may have its own DT agenda and plan, which may vary according to factors such as geographic location, size, cultural, economic and infrastructural contexts, e.g., [18, 127–129]. Comparative studies also exist, for instance: China, Canada and Estonia; [130]; US and UK [131]; Australia, Denmark and the Republic of Korea [132]. Estonia is particularly appreciated in terms of DT [71, 133], to the point of being covered in the general press, e.g., the New Yorker [134], although some criticism is present [135].

Given the above State of the Art, as anticipated and motivated at the beginning of this section, we now introduce two real examples by focusing on the responsiveness aspects: Barcelona [20, 98] and Chicago [55].

- **Technological Responsiveness**

  - **Smart Cities**. Barcelona, thanks to a budget allocation of 1.288 million EUR, has launched three key DT initiatives. The first is the reorganization of data localization, through the establishment of a Municipal Data Office, headed by a Chief Data Officer. The second is the mapping of the entire Barcelona Data System, integrating each of the existing datasets into a single *data lake* [72], developed for this purpose, according to the Open Standards defined by the World Wide Web Consortium (W3C) [136] and referred to as the City Operating System (CityOS). It is based on API and the data within it are now organized and interconnected thanks to the design of a standardized ontology for the city of Barcelona. An additional data-sharing platform, referred to as *Data Exchange*, is connected with the CityOS data lake to ensure a continuous two-way flow of data between the city and the World. The third is the renewal of the Open Data portal through the CKAN tool [137], to

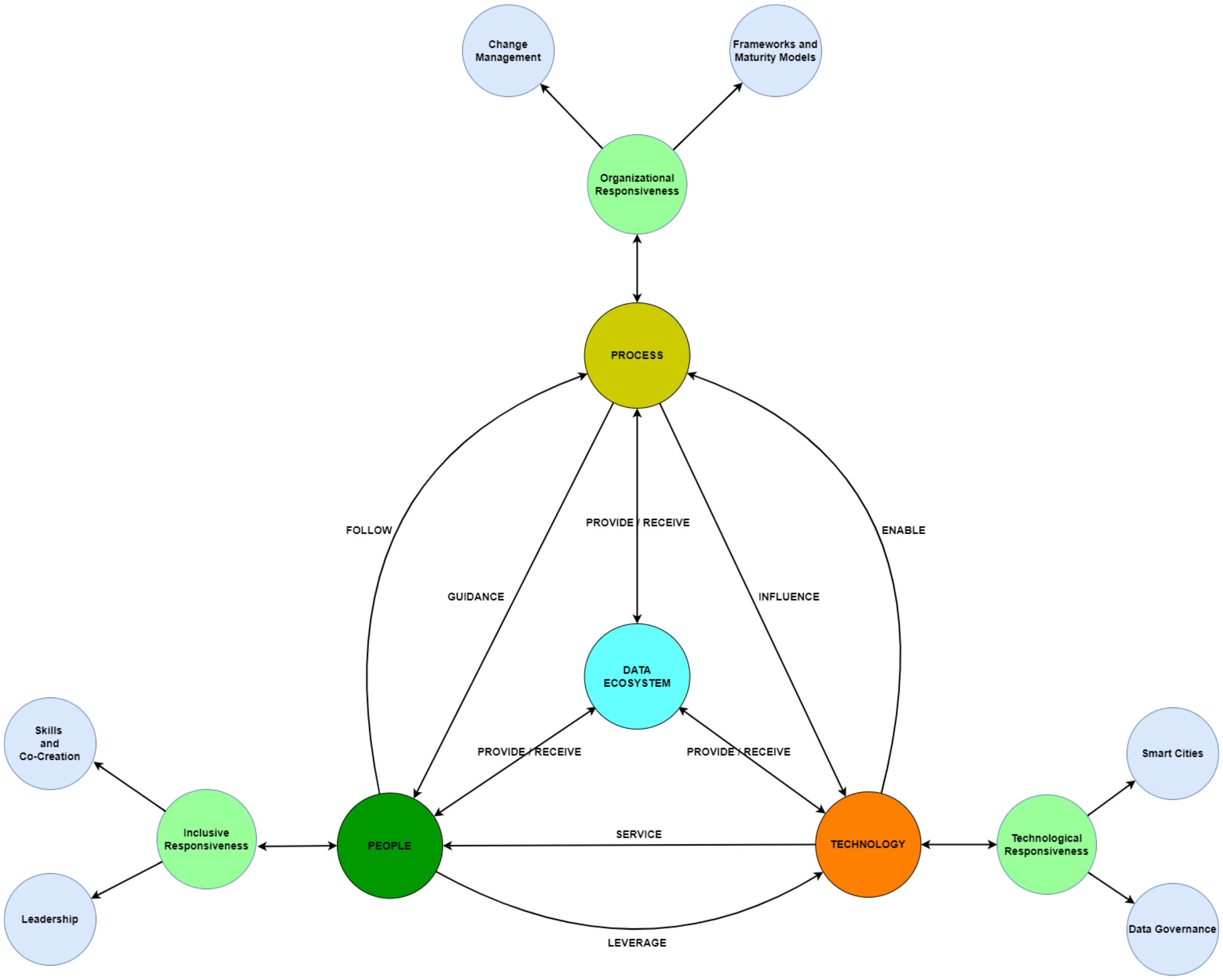


Figure 1.2: **DT Graph Model Augmented with Responsiveness Terms**. The DT Graph Model in Figure 1.1 is augmented in correspondence with the Knowledge Domains. Namely, **Technology**, **People** and **Process**. **Technology** is augmented with the Technological Responsiveness aspects, such as Smart Cities and Data Governance. **People** is augmented with Inclusive Responsiveness aspects, such as Skills, Co-Creation, and Leadership. **Process** is augmented with the Organizational Responsiveness aspects, such as Change Management and Frameworks and Maturity Models.

ensure that public, private and personal data can be transformed into a new data-driven social infrastructure. It is worth pointing out that the city-wide data governance model of Barcelona is an extension of the open government agenda promoted by several cities around the world [138,139], whereby cities support Open Data platforms for civic engagement and improved digital services to address a range of broader challenges, such as the implementation of Smart Cities.

In order to make clear what follows, it is useful to recall the EU DECODE (Decentralised Citizen Owned Data Ecosystem) project [53]. Its goal is to develop a combination of decentralized software technologies, such as Blockchain and Cryptography, to give citizens more control over access and use of their data. The DECODE technology allows data to be encoded and shared anonymously. In addition to what mentioned so far, Barcelona has leveraged DECODE through the *citizen Science Data Governance* pilot project, which uses environmental sensors, placed inside and outside participants homes, to detect noise and pollution levels.

A non-trivial part of this project is the level of detail with which these data are visualized. Data from the IoT networks of sensors are collected through the open source platform Sentilo [140] in such a detailed and specific manner that individual homes can be identified. This raised concerns about the privacy of this data, as homeowners feared that its use could result in the profiling of pollution-prone buildings and homes, which would hurt house prices or insurance premiums. With the mentioned DECODE pilot project, the focus was on developing rules that would allow users to code and share their data with different target groups and with different specificities, generating more trust in the use of the collected data.

Chicago, through the creation of a good quality Open Government Data (OGD, for short) portal, continuously improved since 2012, provides data visualization tools on over 550 datasets, a number that continues to grow, and is relevant to the city. Currently, in the available Literature on OGD, it is generally pointed out that the OGD in the portals that host them are often not accessible, clean, or easy to use [141]. Remarkably, none of these shortcomings seems to have been reported for the city of Chicago. Indeed, from the responses acquired through interviews, several interviewees were appreciative of the availability and quality of the OGD that are available in the Chicago OGD portal [142]. This goal was achieved through a careful processing pipeline in which data were extracted from data owners, e.g. PAs, cleaned and transformed through data cleaning techniques, and uploaded periodically to the OGD portal. Approximately 99% of the data in the OGD portal follows this processing pipeline. Maintaining the data quality levels present on the OGD portal requires great citizen participation, and an active engagement of the PAs that are owners and providers of this data [11]. With this initiative, Chicago is becoming a reference model of increasing sensitivity to data, which is useful for the creation of digital services, following a paradigm that is more and more open and collaborative, and less and less driven by top-down approaches [11]. Another important example of Smart Cities in Chicago is in the reduction of the exposure of citizens to foodborne diseases [143]. The city of Chicago, in collaboration with its Department of Quantitative Research and Analysis of Allstate, has developed a predictive machine learning model that takes into account various data sources, such as waste, crime, and sanitation data, to support the numerically small staff of the Chicago Department of Public Health (CDPH) in prioritizing food inspections to be carried out [144]. The model works by ranking restaurants by the probability that they have a critical food safety violation. The head of the CDPH, through a simple Shiny web application [145], is able to assign food inspectors first to the highest-risk restaurants. By using this model, potential foodborne illnesses could be prevented or their severity limited, as the violations were identified and treated earlier with respect to what would have been possible with previous selection methods.

- **Data Governance**. In addition to the aspects of data governance regarding Smart Cities, Barcelona has adopted a series of new standards, technologies, and practices, which have inevitably enabled new ways of managing data by different stakeholders [20,98], with the result of increasing transparency, simplicity and objectivity, thereby providing a route to technological and data sovereignty. This has been achieved through the appropriate use of procurement clauses, e.g., contracts. It is useful to provide a definition of the contract concept. For instance, under current European public procurement law (Directive 2014/24/EU), a contract, or public procurement, is defined as a legal transaction for pecuniary interest by which one or more contracting authorities and one or more economic operators have as its object the execution of works, the supply of products or the provision of services. Contract law should be seen as a system of rules regulating the exchange of individual claims to things (i.e., *interests*). In any contract, the interests of PAs and contractors must be aligned in terms of citizen needs, quality and safety, cost and time [146].

  The interested reader can find the Barcelona ICT Procurement Guide in [147]. In particular, and in regard to data, a minimum set of requirements are mentioned in regard to availability, accessibility, privacy-compliance, and share-ability as Open Data among the various city Departments. In particular, they ensure that decisions around who produces, owns and exploits the data generated in the city remain in public hands. Those procurement guidelines are useful case studies for other cities [98]. Although Barcelona is a successful case in this area, it is to be mentioned that innovative and effective public procurements involving digital systems in the PA may be challenging [148].

It is not sufficiently clear how some aspects of data governance have been handled in Chicago, but in the blog of the Open Data Portal Development Team of the city [149], the process of data collection and accountability is documented specifically for the different types of data collected. The city of Chicago prioritizes personal privacy in the development of datasets for publication. For example, for the Taxi and Transportation Network Provider Trips (TNP or "ride-share") datasets, an anonymization and aggregation technique has been designed and implemented to reduce the risk of passenger re-identification, while enabling favourable public use of the data (see [150] for further details).

- **Inclusive Responsiveness**

  - **Skills and Co-Creation**. In October 2016, the Barcelona City Council, with an allocation of 75 million EUR to be spent annually on DT, planned to provide public services through an approach based on free software, Open Data sovereignty, and the adoption of Agile development methods, as discussed in [20].

    The main challenges addressed in their DT plans give rise to several initiatives as follows. First, the launch of an educational programme (*Steam Barcelona*), focusing on building competencies within city organizations, with the aim of strengthening the digital skills of the citizens. Second, the combined utilization of iterative and Agile development methods, for reducing the burden on citizens to use services (*City Empowerment*). Third, the design and deployment of new guidelines on the design and accessibility of public services.

    With reference to [55], regarding the city of Chicago, the relationship between OGD and co-creation is addressed, in relation to factors that play a role in the co-creation component of OGD-driven public services. The result is the identification of a set of key factors for OGD-driven co-creation. Specifically: motivated stakeholders, innovative leaders, proper communication, existing OGD portal, external funding, and Agile development. The interested reader is referred to [55] for further details regarding those factors, since we limit ourselves to discuss Agile development within the **Organizational Responsiveness** below.

    There are also some lessons to be learned from this study. In fact, the authors also reported the main barriers to the publication and reuse of OGD, such as the widespread lack of understanding of OGD and their benefits. One of the main challenges to the co-creation of public services is the need to redefine the roles of public and private actors in the public service creation process. Some other barriers are connected to the figure of the citizen, such as the internal motivation of participants, personal characteristics, awareness of participation opportunities and participatory skills, perceived ability to participate in co-creation initiatives, trust in co-creation initiatives, the relative importance of the service to be co-created and mutual trust between Government and citizens.

  - **Leadership**. There are many facets to this topic. Barcelona exemplifies one of them. Specifically, the establishment of a managerial figure such as the Chief Technology and Digital Innovation Officer, to support the administration of the city, thanks to which a series of politically and managerially strong reforms could be initiated [98]. Chicago exemplifies another one. Specifically: technologies, e.g., data analysis techniques that allow better leadership because they support decision-making processes, aiding managers in exploring and solving some of the most difficult problems facing the city [55].

- **Organizational Responsiveness**

  - **Change Management**.

    The city of Barcelona, in 2017, within its DT transformation plans, has provided guidelines for project management that recommend the use of Agile methodologies [20,98]. As a matter of fact, Barcelona has developed its own Agile methodology as a variation of the Scrum Framework, referred to as SCRUM@IMI since the Institut Municipal d'Informàtica has had a major role in the adaptation of Scrum to the Barcelona ICT needs. The interested reader can find a detailed account of this initiative at [151]. This initiative is certainly important. Therefore, it turns out to be useful to anticipate that one of the major contributions of this Dissertation, which is present in the second part, consists of the generalization of what the Barcelona City Council accomplished.

    As for Chicago, in terms of Agile development [55], according to the opinion of several interviewed stakeholders involved in the development of many projects, although the implementation of services did not explicitly follow Agile development methodologies, many of the characteristics of such approaches were however present in the development of services. The interviewees have emphasized some of these characteristics, considering them crucial to the success of the project, in the design, implementation, and service delivery phases. Namely, speed of development;

release of a minimum viable product (MVP); validated learning; incremental development; constant testing; and the ability to respond quickly to feedback and evaluations.

- **Frameworks and Maturity Models**. The Barcelona City Council has continuously collected feedback and, in terms of metrics, measured various performance indicators on the services provided in order to monitor signs of progress on the expected results of the adopted DT strategy [20]. It is not clear how progress on the expected outcomes of the initiatives implemented in the city of Chicago is measured, as we found no authoritative documents on this topic.

1.4 Conclusions

In this chapter, we presented a graph model for Public Administration based on a systematic literature review. Specifically, we identified four knowledge domains and then described the main interactions among these domains. Then, we extended our graph model, corresponding to the knowledge domains Technology, People, and Process, through the concept of responsiveness, identifying for it three fundamental aspects, respectively: technological, inclusive, and organizational. Finally, we validated our model by exemplifying two illustrative examples: the cities of Barcelona and Chicago.

Chapter 2

Details And A Look To The Future

Abstract

For each of the knowledge domains discussed in the previous chapter, we present, in more detail, some concrete examples. Then, we give insights of interest on how these domains can be useful as part of Digital Transformation strategies in the context of a given PA. Then, we offer some suggestions, for each of the knowledge domains identified in our graph model, allowing stakeholders easier planning in the DT strategies that pertain to them.

2.1 Introduction

This chapter provides concrete examples of how to address Digital Transformation challenges, in the context of the Public Sector, in terms of Data, Technology, People and Process.

Regarding Data, in Section 2.2 we first provide a glossary of useful terms to delineate the data context in which we find ourselves, then, in Section 2.2 it is emphasized how the collection and management of Open Data are of key importance, and finally, in Section 2.2 we provide two concrete examples of the use of Open Data in the public context.

Regarding Technology, in Section 2.3, we consider two main technical aspects: smart cities and aspects related to data privacy and security (Data Governance).

Regarding People, in Section 2.4, we focus on aspects related to the acquisition of new digital skills and the collaborative and participatory creation of the digital services that the PA intends to adopt for the benefit of its citizens (*co-creation*).

Regarding Process, in Section 2.5 we focus on two key aspects: Change Management, and Frameworks and Maturity Models. In terms of Change Management, in

Section 2.5 we discuss Agile project management methodologies, which are useful for migrating processes to more iterative and incremental approaches than traditional (waterfall) methodologies. In terms of Frameworks and Maturity Models, in Section 2.5, we discuss the main digital maturity models, large and small scale, and how they can contribute to determining the progress achieved in adopting a DT strategy.

Finally, in Section 2.6 we provide some suggestions (future directions) on the State of the Art concerning the knowledge domains identified in our graph model, in order to enable the PA to be able to approach their DT strategy with greater awareness.

2.2 Data

We discuss here, in detail, the data ecosystem.

A Glossary of the Data Ecosystem

For the convenience of the reader, we describe the following well-known general terms: Open Data, Linked Data, and Linked Open Data.

- Open Data are accessible, exploitable, modifiable, shareable by anyone for any purpose, including commercial purposes, and released under an open license [16].

- Linked Data are structured in such a way as to be interconnected with other data sources to become more useful, promoting discoverability and interoperability. They are built on standard Web technologies such as HTTP, RDF, and URI, but instead of using them only to serve web pages to human readers, they are as well used to share information in a machine-readable way [152]. This type of data has evolved to encode and model knowledge coming from different sources. Thanks to the use of the RDF standard (*Resource Description Framework*) [153], knowledge graphs [154] can be built where facts are described using so called ontologies; namely, explicit, formal and shared representations of domains of interest represented by entities and relationships between entities.

- Linked Open Data are the intersection of the previous two categories.

When data are public, we distinguish Open Government Data (OGD), Linked Government Data, and Linked Open Government Data (LOGD), respectively. Figure 2.1 represents the relationships among Government Data, Open Data, and Linked Data.

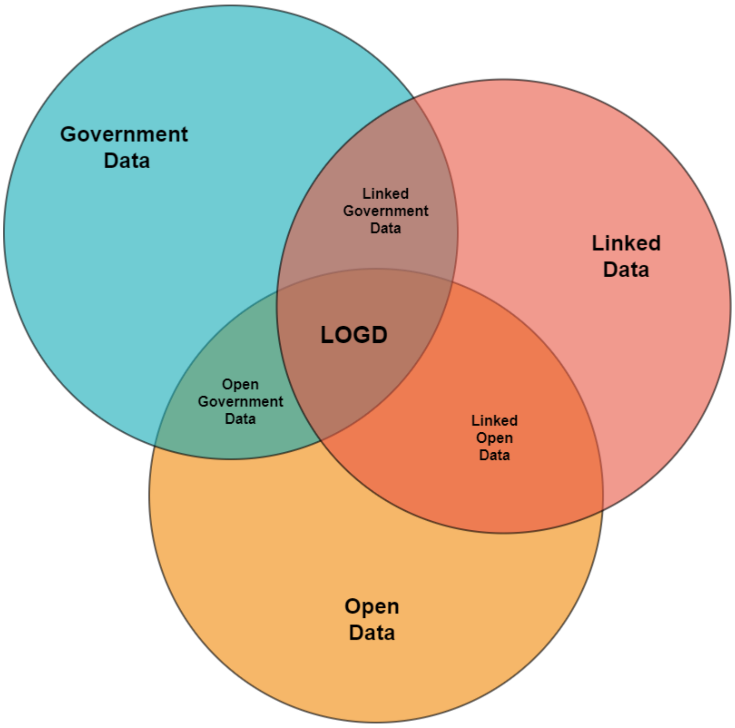


Figure 2.1: **Data Ecosystem**. Relationships between Open, Government and Linked Data.

Collection and Management of Open Data: PAs Are Special

An increasing number of data that we have described are represented using technologies related to the Semantic Web [155], via Open Data standards, which are defined by the W3C and supported by most technology providers, especially those offering data management tools. However, the collection and management of Open Data by a given PA seem to require innovation in the processes adopted to carry out those tasks [14]. In the study just mentioned, such innovation is characterized in terms of Agile methodologies: the ability of an organization to capture emerging needs and promptly associate them to the current data processes, in order to obtain innovative data-driven products and services. Based on the empirical study of four PAs, a process model for the achievement of the mentioned agility is proposed (see [14] for details). This is coherent with the concept of responsiveness of our graph model, presented and discussed in Section 1.3, with particular attention to technological and organizational aspects.

How to improve the usability of the collected data also requires innovation. Indeed, as discussed in [156], although there is a proliferation of Open Data platforms, their usability for the non-specialist is perceived as poor, mainly due to the fact that they have been designed by software specialists. The mentioned study also provides the evaluation of the usability of an Open Data platform by non-specialists, for the City of Dublin, pointing out the need for innovative designs of user interfaces. Open Data integration is also a serious obstacle to their fruitful use, with some proposals on how to overcome them, accounting for implementation strategies and organizational models [157].

From a Data Ecosystem Abstraction to its Concrete Realization: Some Examples

Figure 2.1 provides a general description of a data ecosystem, which can then be realised in several ways, that should be compliant with the Open Standards defined by the W3C [136]. Two incarnations of a data ecosystem have already been presented and discussed in Section 1.3: the Barcelona CityOS data lake and the Chicago Open Data portal. We now provide three additional examples, concentrating on their technical aspects and pointing out their usefulness.

- **Open Data Catalogs for the PA**. They are software applications that build inventories of data resources of a given PA, in order to help data professionals and stakeholders to find relevant data for analysis-related uses [158]. They are based on metadata, which provide additional data/information about data resources. The intent is to help catalogue users to understand the structure, nature and context of the data available in computer systems and decide whether they are suitable for their needs. One of the earliest relevant examples of an Open Data Catalog proposal is the OGD Catalog of the Czech Republic: it serves as a single access point to the OGD datasets, supporting the discoverability and reusability of the available OGD. Another, more recent example, is provided in a case study of the Italian PA [36], conducted during the period April-December 2017, in which services (or digital platforms) such as *pagoPA* (payments system to PAs and public service providers in Italy), *SPID* (Italian Public Digital Identity System), and *ANPR* (Italian Register of Resident Population), although not directly based on Open Data, interact with platforms and services that use Open Data to improve the accessibility and efficiency of services offered to citizens. These new services are based on private and public Open Data Catalogs, and in addition to the use of Open Standards, and Open Software, are designed as modular structures, which facilitate their evolution and reduce the complexity of coordination between the actors involved in the co-creation processes of the various projects, as also as reported in [63, 114].

  For a successful integration of services, a good interoperability framework of information sources must be provided. In general, it organises the exchange of data and interoperability between different services, data centers and PAs. It consists of a set of specific design rules, documents and toolkits for software developers (e.g. Software Development Toolkits).

For the specific case study, the main part of the interoperability framework is the Data Analytics Framework (DAF), which collects and processes data from PAs and external actors to make them publicly available and accessible through a Web user interface, and defines protocols and regulations that facilitate the integration and orchestration of services. The DAF empowers each PA to orchestrate the creation of public value by establishing the actors that can have access to the data and the terms under which they can access them. Uploaded data are supervised by the Data Protection Authority [159], which safeguards the privacy of citizens and evaluates how other public agencies use their data. Therefore, each level of government and the different public agencies are responsible for regulating how data is accessed, according to their administrative and political responsibilities.

Specifically, a PA, as well as a private company, can make data available to the public through the DAF, and can also indicate who can access that data and the ecosystem on which that data should operate. As expectations and needs change, data access settings can be modified to adapt to emerging needs and requirements. When public agencies upload their data to the DAF, they fill out a privacy form to ensure that the data is privacy compliant, so as to avoid any negative effects on citizen privacy. Since 2020, the DAF is no longer an Italian PA project because it has been replaced by a different platform referred as Piattaforma Digitale Nazionale Dati (PDND). The original DAF was very different from the current PDND. In summary, while the Data Analytics Framework focuses on data analysis and the extraction of useful information, the PDND provides the technological and operational infrastructure for data management and sharing at the national level. The interested reader can find more details regarding the PDND in [160].

- **Cloud-based Open Data Federation: the CLIPS experience** [19]. CLIPS is a cloud-based approach for migrating public services to the Cloud, based on the use of microservices (MSs). It involves four European Cities: Bremerhaven (Germany), Lecce (Italy), Novisad (Serbia) and Santander (Spain). It is based on the Open Data because, in addition to being a useful resource for developing new value-added services, they seem to be valuable for exploring potential transnational business opportunities. The CLIPS platform includes an *Open Data Federation* node to allow access to the Open Data sets from different federated Municipalities, as if they were a single data source for front-end applications. The main innovation of CLIPS is to provide a usable methodology, that enables Government employees and other external stakeholders to collaborate on new projects and service delivery from a set of basic building blocks, available in the Cloud. This offers the ability to respond more quickly, reduce service delivery costs and be more responsive to end-user needs. It defines an approach for building an ecosystem in which PAs, small and medium-size enterprises, and citizens can co-create new and innovative public utility services. The CLIPS platform is designed as a three-tier cloud platform, including: an Infrastructure-as-a-service (IaaS) which includes all the required modules to provide basic cloud resources like computation, storage and networking; an application serving and development functionality of traditional Platform-as-a-service (PaaS); and a convenient marketplace for the developed cloud-based services and MSs, typical of a Software-as-a-service (SaaS). It consists of several modules, such as authorization, authentication, and monitoring of data access, as well as providing an API to connect the MSs present with each other. Data security aspects are also addressed. In fact, the CLIPS Security strategy, in addition to common security best practices (e.g., ISO/IEC 27002, ISO/IEC 27017, ISO/IEC 27018), is to adopt some innovative techniques and approaches from the open source community as well as from other European Projects, such as "Secure idenTity acrOss boRders linKed" (STORK) [161], enabling citizens to use their national credentials in PA applications provided by foreign States and to securely transfer their sensitive data between the States.

- **Data and Smart Cities**. Smart Cities are perceived as data engines [85,162], e.g., IoT infrastructures, social networks, wearable devices, etc. generate valuable data that can be used to improve or offer new services to the citizens. However, due to its volume and heterogeneity, the collection of data produced by Smart Cities (including the creation of related metadata) requires a non-trivial effort in the verification of its correctness and quality [162]. Metadata can describe different information sources and can be collected and catalogued within appropriate Open Data Catalogs, such as the open source solution already mentioned CKAN. Moreover, the assembled metadata can be stored in a vocabulary designed to facilitate interoperability between Open Data Catalogs available on the Web, e.g., possibly with the use of DCAT-AP [163]. Fundamental turns out to be the implementation of a set of guidelines and documents such as API documentation, and planning documents, systematically discussed and agreed upon with public officials responsible for providing datasets to improve discoverability, understandability, and further processing of data. As shown in [164], the implementation of these solutions involves a careful design phase of the technology infrastructure (cloud/edge) related to Smart Cities, with an emphasis on the data acquisition plan. The infrastructure must be the pillar of processing and storage of data and also include data analytic tools and methods finalized for the implementation of robust machine intelligence solutions available to the city government for the benefit of its citizens. To this end, three distinct taxonomies of data analytic tools serving Smart Cities are proposed in [85] and referred to as the DMS Taxonomy, i.e., data, methods and services.

2.3 Technology

We discuss here, in detail, the two main technical aspects we have considered concerning **Technology**: Smart Cities and data governance. As anticipated in Section 1.3, for Smart Cities, we concentrate on Digital Twins. It is useful to remind the growing importance of AI and blockchain techniques for the DT. In particular, in regard to Smart Cities, we point out that, when discussing Barcelona and Chicago, we have provided two examples of such an important role within those Smart Cities (see Section 1.3). Indeed, Barcelona uses the blockchain technology of the EU DECODE project to handle data access granting privacy, while Chicago uses machine learning techniques to perform sanitary controls and disease prevention.

Smart Cities: Digital Twins

Being specific to Smart Cities and following [99], the major characteristics of Digital Twins are: Accurate City Mapping, e.g., roads and public illumination; interaction between the virtual and real "objects", e.g., people and their "avatars"; software definition, e.g., platforms that simulate the real city in a virtual space; intelligent feedback, e.g., evaluation of the effects of city plans and initiatives before realization. Interestingly, it has been argued that their realization may enable an acceleration of NetZero emissions in Government critical infrastructures [165]. A further refinement of the technical characteristics of Digital Twins is proposed in [166], although its major contribution seems to be the account of Digital Twins initiatives in China, USA, and France.

Although there are many national and city initiatives regarding Digital Twins, e.g., [166–169], we have found only a limited number of academic papers covering the subject. One is in regard to cross-border Smart Cities, i.e., Helsinki and Tallinn. Recalling that an urban operating system is a network of sensors that can acquire data regarding the city which, in turn, can be transformed into "knowledge" [170] and pointing out that the X-Road data infrastructure [171] is one of the pillars of Estonian DT, a cross-border urban operating system involving both cities is proposed in [172]. The intent is to have an integration of the DT that involves only each of the mentioned cities. For completeness as well as relevancy for this part of this Dissertation, we mention that the notion of an urban operating system is investigated in depth in [173], with various examples of it. The study points out the modest impact that it may have on city planning and its contradictions.

Data Governance

In the context of our review, we focus on two particular aspects of data governance, i.e., privacy and cyber-security.

In terms of privacy, in the international scenario, there are several National Data Protection Authorities. An exhaustive list of these Authorities can be found through an interactive map on the website of the French National Commission for Information Technology and Civil Liberties (CNIL) [174]. The main function of the individual Authorities is to protect the privacy of citizens and assess how PAs (or other organizations) use their data, for example, by keeping under control the data they publish on their respective institutional web portals. Barcelona is a good example in terms of control of the data, regarding availability and detail of access, as discussed in Section 1.3. As well argued in [175], the amount of data that is collected within Smart Cities initiatives, once made public, even in an anonymous form, can be subject to cross-reference attacks that could capture private information. In order to address this problem, the mentioned paper proposes solutions and use-cases. Interestingly, that study is a pilot project funded by the U.S. Department of Homeland Security that has the intent to demonstrate how data privacy technologies can be of help.

In terms of cyber-security, it is well known that there is a proliferation of Cyber-Security Agencies, e.g., the European Union Agency for Cyber-Security (ENISA) [176]. This is not surprising, given the increase in the number and quality of the attacks of which we have news in the past few years [177]. However, since PAs are also the object of those attacks, it is surprising that there is only a limited number of papers that have emerged from the literature regarding DT that address cyber-security issues, as we outline next.

With regards to data, in [27], the Organisation for Economic Co-operation and Development (OECD) [178] recommends maintaining a strong balance between the need to provide timely official data and the need to provide reliable data, as well as to manage the risks associated with the increased availability of data in open formats and those related to digital security and privacy. A related issue is the design and management of government data centers architectures, in particular regarding security. Indeed, those centers, due to the heterogeneous nature of services they offer and software they host, are vulnerable from the point of view of security. A proposal on how to achieve ISO/IEC 27000 security standard, a model of government data centers architecture has been proposed in [179]. More in general, as discussed in [180], there are several initiatives in many countries having the goal to provide methodologies for security assessment. This latter consists of evaluating an information system from the attacker's point of view, with the aim to provide a systematic review of weaknesses in information systems, with a corresponding assignment of probabilities of attack via each weakness, offering also a scale of severity levels of damages. Recommendations for corrections are also offered.

Smart Cities and their associated technologies, being relatively novel, are also object of study in terms of security. A specific analysis regarding IoT devices and related technological infrastructures, is given in [22] (but see also [181] and references therein). Indeed, due to their interconnected nature, IoT technologies make data security a more complex challenge with respect to the past. Therefore, ensuring the security of IoT products and services has become a top priority. To this end, an entire framework, referred to as SAO, regarding the automation of IoT security has been proposed in [181]. It has the merit of being grounded on a recent review of the State of the Art, clearly describing challenges and proposing solutions. SAO integrates the key elements for security automation and orchestration for IoT systems, including threat modelling, security and privacy by design, trust management, security configuration, threat monitoring, patching, compliance check, and secure data sharing. Another specific analysis is provided in [182], regarding Digital Twins. Indeed, the confluence of a broad set of technologies, ranging from cyber-physical systems to artificial intelligence, and the implicit interaction with the real objects modelled by the Digital Twin, poses new security threats. The mentioned paper offers a classification of them, together with security recommendations on how to address them, via a paradigm that classifies the threats based on the functionality levels composing a Digital Twin.

2.4 People

We discuss here, in detail, the technical aspect we have found concerning **People** that deserves further attention, based on the Literature search: Skills and Co-Creation.

Skills and Co-Creation

A successful DT process requires users not only to acquire new skills but also to know how to interact effectively with them [63]. Those skills required to handle DT do not only relate to a particular discipline but require a multidisciplinary approach, where the importance of knowing the specific competency levels of the individuals that are part of an organisation and the know-how of the entire organisation itself is recognised as a fundamental requirement. The lack of a coherent educational approach to the acquisition of appropriate skills also hurts e-government users, which could generate problems in the usability of the PAs digital services [35]. In [40], the authors, as a possible solution to this shortcoming, propose an educational framework composed of five basic components designed, developed, and tested to achieve the educational goals necessary for a successful DT strategy. The different components of the framework were intended to: (a) define a useful competency model to describe the required competencies; (b) define an educational approach that can be provided by the professional or academic context; (c) define a maturity model to monitor progress in the process of acquisition of the required competencies; and (d) define a competency certification system to facilitate organizations and citizens to understand and communicate their competencies, ensuring transparency and quality.

As for co-creation, it is useful to recall from the previous sections that the ultimate goal of a digital PA is the co-design and deployment of services that are perceived as being of “value” (see **People** in Section 1.3). Accordingly, how to achieve that goal and with which methodologies and supporting technologies is an emerging area of research [37], that we outline next.

In its simplest and easiest to realize form, a co-creation methodology is limited to the participation of a strictly selected set of users, particularly in the initial phase of the creation process, and to the related measurement of their perceived satisfaction degree, through constant feedback collection [60]. However, the intent is to have co-creation methodologies that can handle millions of users, i.e., citizens. It is natural, then, that the IT platforms supporting the PA must support those “in the Large” co-creation methodologies. Recalling from [36] that Government as a Platform (GaaP) is a new way of building digital public services using a collaborative development model by a community of partners, providers and citizens to share and enhance digital public processes and capabilities, or to extend them for the benefit of Society, its realizations seem to be designed to achieve efficiency. However, according to the mentioned study, the efficiency granted by GaaP does not necessarily imply the creation of value for the citizens, a point also made in [10, 59]. Indeed, as discussed in the mentioned paper, the key to the creation of public value seems to be the modularity of the platform configuration and the ability to consistently coordinate different ecosystems that support public agencies. To this end, a few examples are provided, borrowed from the private sector and involving IT giants such as Apple, Google and Amazon. Here we limit ourselves to mention the Apple iOS Support Service [183], which enables multiple ecosystems, different in nature, to interact and coexist. According to the analysis reported in [36], the adoption of analogous models would allow the co-creation (PAs and citizens) of value services with a “large scale” involvement of active actors. The importance of adequate digital platforms for the co-creation of value involving a large number of actors is also identified as a key success factor in [60]. A paradigm shift from crowd-sourcing and social media monitoring to IoT has also been proposed, with a pilot project that has been set-up in a Municipality in Sweden [184].

In addition to what we have mentioned so far, the notion of participatory design, e.g., the involvement of citizens in urban planning, is being analyzed in view of DT. A historic account of how that notion has changed over the decades and how it fits a modern view of DT is provided in [185]. An important related topic is the co-creation of integrated public services. That is, ideally, a one-stop platform for the citizens that integrates the available services to them. The State of the Art, mostly regarding EU, is well presented in [186].

For completeness, we also mention that, in terms of PAs and co-creation of value, the Italian public administration as a platform is studied in [36]; the Norwegian Labour and Welfare Department is studied in [60], while a platform supporting co-creation at different levels of governance in Portugal is presented in [187]. A specific platform for co-creation in the area of Urban Planning and in support of previous initiatives, i.e., the International Laboratory of Architecture and Urban Design, has been proposed in [188]. Finally, a model based on Digital Twins that allows co-creation, as well as evaluation of the final result regarding public services has been proposed in [189], with a planned test of the model in Sofia.

2.5 Process

We discuss here, in detail, the two main technical aspects we have found concerning **Process**: Change Management and Frameworks and Maturity Models.

Change Management

As well put in [151], although the Agile Manifesto dates back to 2001 and despite the remarkable success that the corresponding methodologies have had in the private sector, their adoption in the PA is rather slow. Yet, in the DT, Agile methodologies (see [190]) seem to be the ones that should replace more classic ones, such as Waterfall [191]. In order to exemplify this point, the experience reported in [192, 193] suffices. In the mentioned studies, the authors point out that the implementation of the e-governance project Digital India Land Records Modernization Program (DILRMP) has highlighted major challenges and complexities, typical of traditional project management. They discuss how an Agile management approach can play a key role in transforming such implementation from slow and ineffective to be more responsive, flexible and effective.

Documented difficulties in the adoption of Agile methodologies have emerged [62, 65, 119, 120, 194, 195]. The cause is common: PAs and the private sector operate in very different contexts that significantly influence how they deal with change and adopt new methodologies. The private sector often has greater flexibility and agility in responding to market changes and adopting new practices and technologies. They are mainly motivated by profit and market competition, which drives them to be innovative and adaptable to remain competitive. Consequently, they tend to be more disposed to experiment with new methodologies and adopt more agile approaches in systems design and development. On the other hand, PAs often have to navigate through a labyrinth of regulations, bureaucratic procedures and budget constraints. This can make it difficult for them to quickly adopt new methodologies or technologies, as they have to comply with strict regulations and ensure transparency and accountability in decision-making processes. In addition, PAs also have to consider the crucial role of public service continuity, which may limit their ability to take risks or experiment with new approaches. This difference in context often leads to a disparity in resilience to change between PAs and the private sector. Fortunately, studies [196] seem to have identified "agility enables", i.e., possible actions that can facilitate the transition to Agile models. However, as pointed out in [197], the transition to Agile development models requires the writing of appropriate guidelines to be used to ensure that the development process is Agile. These depend on the particular requirements of the organization involved in the transition process.

Although the highlighted difficulties persist, there are many PA project management initiatives that use the Agile methodologies, e.g., in the software development Census of the Swedish Government Agencies, the majority of Government Agencies consider their approach to be more agile than planned [198, 199]. In addition to Barcelona and Chicago, mentioned in Section 1.3, the Agile methodologies are applied in several PAs [20, 151], ranging from National (e.g. UK), Large Cities (e.g. New York), and Regional Governments (e.g. Andalusia). Apart from the above noteworthy examples, a systematic and technical presentation of the adoption of Agile methodologies for project management in the PA is reported in [64]. The paper makes also a list of the Agile process automation technologies that are in use, i.e., Scrum, Kanban, and SAFe (see again [190]). A comparison is also performed with classic Waterfall methodologies and it is stated that the Agile ones allow for more transparent projects, effective team building, adaptability to change, lack of hierarchy, lack of bureaucracy, continuous education. Some disadvantages are also reported, such as: the risk of endless product changes; the high dependence on the qualification and experience level of the development team; the difficulty of determining total project costs in a timely manner. Although unclear in its impact, an effort is also made for the identification of the specific characteristics that the Agile methodology should have for its use in the public sector [66]. The mentioned paper reinforces the difficulties already mentioned and that must be overcome for such a change of project management. Moreover, it stresses that project management should be reconfigured to provide team autonomy, to some extend. Once again, the barrier being routine practices difficult to abandon and obsolete regulations.

A more specific evaluation of Agile methodologies in the PA, regarding DevOps [200], is provided in [69, 201], where it is considered how to bring best practices from the production world into PA, making the flow of information more fluid. As a result, the adoption of DevOps promotes organizational responsiveness, which is useful for improving productivity and performance. At the same time, DevOps breaks down organizational barriers by promoting information exchange through the use of shared metrics and feedback mechanisms between development teams, as reported in [202].

By bringing DevOps into the public sector, a more collaborative working methodology and an open flow of knowledge among PA employees are expected [203,204]. There are initiatives in this regard, as for instance the ones of the Brazilian Federal Government, referred to as Brazilian Public Software (SPB). The objective is to promote sharing and collaboration enabled by Free/Libre/Open Source Software (FLOSS) solutions for PA [203]. SPB is an interconnected platform based on different FLOSS tools that provides different solutions for collaborative software development, with the purpose of enabling Brazilian PAs to share information, experiences, and best practices about the use of these tools (see [205] for more details about the architecture and operational manuals).

Furthermore, since transparency and openness are among the core principles of DevOps practices, their use is expected to simplify bureaucracy and decrease corruption in public service delivery. A punctual analysis regarding the benefits of using DevOps Process Model in the PA is presented in [206]. It involves seven Saudi Arabia PAs, evaluated with the use of the Bucena DevOps Maturity Model [207]. That study concludes that the use of DevOps is promising although DevOps cultural aspects, process, and technologies need to be strengthened. An additional study proposing DevOps for the generic support of Digital Transformation is presented in [208], being in agreement with the papers mentioned so far.

We mention that there are also experiences indicating that classic Waterfall and Agile methodologies can synergically co-exist. Indeed, we learn from the case study in [68], involving the development of projects through Agile methodologies of some Brazilian governmental organizations, that although the adoption of such methodologies fosters an improvement in the quality of the public services created, these projects achieve greater success when conducted in combination with other traditional software development approaches. The authors of [209] have conducted an analysis on the need for a balance among Agile and plan-driven software development approaches in the e-services sector. The result of this study is that the use of a combination of both approaches is necessary for project success. In addition, risk analysis can help determine the appropriate level of planning. Finally, Agile software development in the public sector must be scalable, i.e. able to work for relatively small projects, coming from small realities such as cities, to large national and international projects.

It is clear that there are difficulties, as well explained in [210] for the case of New Zealand. It is worth pointing out that a recent review [211] regarding the use of Agile methodologies on a large scale provides an up-to-date State of the Art in terms of "business domains" where "large scale" Agile methodologies have been applied. Somewhat, unfortunately, only 5% of the reported case study deal with the Public Sector. Even more unfortunately, although it is claimed that pointers to the case studies are provided in Appendix C of the online version, such an Appendix is not there. Therefore, we are not able to account for those experiences in this Dissertation. It is also of interest to mention that a recent systematic Literature review specific to SAFe specifically includes the Public Sector, with enough technical details to be of use in this Dissertation.

Although the initiatives mentioned so far are part of the scholarly Literature, at the technical level, it is very difficult to find examples of how to be able to consistently and cohesively drop Agile methodologies within the PA. That is, the "how to" is somewhat scarce and somewhat sketchy. The second part of this Dissertation is devoted to fill this gap for the deployment of the Agile methodologies in the PA.

Frameworks and Maturity Models

Over the past decade, various frameworks and models have been developed to measure and monitor the degree of digital maturity achieved by Digital Transformation Strategies. To date, however, it is not possible to choose one among them for which any organizational reality can be perfectly modelled, whether private or public. Each of these captures a particular set of indicators and uses different tools to collect information to be used to quantify the indicators. One of the tools is certainly interviews, with the possible addition of document analysis [36,61,67,68,74,78,113,125,212–219]. In the mentioned case studies, semi-structured interviews are mainly conducted with IT professionals from public and private organizations, actively involved in DT processes, over different periods to measure the degree of digital maturity gained. Several barriers and success factors emerged from the interviews, which are useful for a comprehensive understanding of DT. The results show that this survey instrument is quite valid, as effectively reported in [61, 78, 218] (see respective Appendix Sections).

In addition to the model-specific to interviews, many general maturity models have been developed over time. For most of them, based on variables specifying the model, the "end-result" is the value of an index that assesses the level of achieved maturity. Some follow macro-economic factors on a national or international level, as in [71, 220–228]. Others, however, refer to micro-economic factors related to individual organizations, as in [126, 207, 229–231].

In regard to the first group, we discuss only the GovTech Maturity Index (GTMI) developed by the World Bank [71], as part of their GovTech initiative (Government and Technology) [232], since it appears to be the most exhaustive maturity model currently available. It is worth pointing out that GovTech is an approach to the modernization of the public sector, through innovative technological solutions, that promotes a simple, efficient, and transparent Administration with the citizens at the center of the reforms. There are about 80 GovTech initiatives worldwide, with good practices observable in 43 countries out of the 198 observed. In this context, GTMI is a comprehensive measure of the DT in a given country. It is based on 48 key indicators and it is defined to collect data from 198 countries. GTMI measures key aspects of four focus areas of the GovTech initiative: supporting Core Government Systems (CGSI, 15 indicators), improving Service Delivery (PSDI, 6 indicators), Engaging citizens (CEI, 12 indicators), and promoting the Enabling factors of the GovTech initiative, such as building digital skills in the public sector and an environment conducive to innovation in the public sector (GTEI, 15 indicators). Each of the indicators is associated with a certain score and a certain weight, the latter based on the opinions of some domain experts on the relative importance of the selected indicator. Using these scores and weights, the CGSI, PSDI, CEI, and GTEI scores are calculated. The final GTMI score, on a 0–1 scale, is calculated as the arithmetic mean of the four scores just mentioned. See [71] for more explanatory details on the indicators. All 198 countries were grouped into four categories: from A (leaders in GovTech) to D (minimal attention in GovTech) according to their GTMI score. Based on analyses comparing the GTMI with other relevant indices, the GTMI indicators were found to be consistent and robust, even concerning the analysis of lesser-known dimensions related to particular characteristics of a given Government. Results and good practices presented in [71] demonstrate how the GovTech focus areas identified by the World Bank are highly relevant to the DT agenda in most countries.

As for the second group of models, which relates to the micro-economic factors of individual organizations. For conciseness, we will only briefly discuss the Digital Maturity Balance Model [126]. It is oriented towards PAs and is based on two axes: digital maturity and importance ratio. The focus is on measuring the balance between the two. Each maturity dimension is assessed by taking into account the importance ratio of this dimension in the Organization. The main categories of maturity dimensions involved are data, IT governance, strategy, organisation, and process. The construction of the model essentially consists of three steps. First, a method must be defined to assess digital maturity. Secondly, a method must be defined to measure the importance of each dimension of digital maturity pertaining to each of the categories involved. Third, a self-assessment tool must be provided that combines the methods just mentioned, e.g., in the form of an online questionnaire, in which the questions allow the assessment of the digital maturity criteria and the digital relationship attributes. Results show that the use of the model and of the self-assessment tool is useful and relevant, but needs further refinement to fully correspond to the reality of a given PA.

Interestingly, micro-economic maturity indexes may be of use in measuring other aspects of DT, far from the ones they have been designed for. By way of example, the CMMI index [229] has been adapted in [233] in order to measure the success of the adoption of the Agile DevOps methodology in the PA project management.

From the presentation above, it is clear that many indexes and maturity models exist for evaluating the DT progress of an organization. However, they can be limiting because they tend to favour a sequential, linear approach to digital growth, leaving out the intricacy and flexibility needed in an ever-changing organizational environment, as in the case of the PAs. In addition, they may not fully account for the unique challenges of each organization, as they are often standardized and not always adaptable to the specific needs of individual organization contexts.

2.6 A Look To The Future

- **Data**

  From what has been discussed in Section 2.2, it is evident that data innovations come from using Open Semantic Web standards in the context of PA to represent their information assets. The introduction and use of Open Data is certainly a big step forward since the advanced functionalities they make available have transformed the OGD landscape [234]. Unfortunately, little attention has been dedicated to the LOGD, in particular, to all those activities related to the production and maintenance of quality levels, which facilitate interoperability with other data sources [12], according to the Open Government principles [13, 235]. In particular, a domain that needs attention for the DT is the one regarding the use of *RDF Knowledge Graphs*, since their use would facilitate the discovery of new data sources and improve their interoperability among different PAs. Another aspect that needs to be developed is to set-up mechanisms that strengthen the trust among citizens and PAs regarding the use of the collected data [10]. A related topic is the one of security, in particular regarding the creation of a more balanced system of protection among the needs of PAs and the resulting risks. One aspect that seems to have attracted little attention in the DT is the efficient storage of the vast amount of data that is and would be available. To this end, there is extensive experimental research addressing compression of RDF data in [236–240], but those techniques are hardly used in the DT.

- **Technology**

  As outlined in Section 1.3, Cloud Computing is a main component of any DT. Moreover, as discussed in Section 2.3, Smart Cities and Digital Twins seem to be the future. Somewhat unfortunately, the complexities related to their full scale realization are far from being addressed and resolved. The difficulty of that the scale of what has to be managed poses is best exemplified by a study regarding energy consumption optimization of “only” sixteen buildings in Rome, via Digital Twins [241]. More in general, a recent review clearly outlines the five major challenges that need to be addressed [242]. Not surprisingly, they range from data collection, storage and analysis to computing power. Although some research directions are also mentioned, they lack specificity and a clear assessment of how the scale of what has to be managed via Digital Twins affects costs: a city, even a major one, may not be able to economically sustain its full fledged Digital Twin.

  Concerning privacy and security, the adoption of recognized standards, such as ISO/IEC 27001 is strongly recommended, as indicated in [19, 179]. To this end, it is suggested that a more collaborative approach be taken to support security in developing effective and appropriate solutions to security challenges, including increased efforts on technologies IoT [22], to prevent attacks or minimize their effects. At the State of the Art, there are no evident documented outcomes in the Literature on how these recommendations have been understood and pursued by PAs. The actions, however, appear to be in place, as shown in the timelines of the NRRP Plans, i.e., [243].

- **People**

  One of the major problems that emerge in terms of digital skills is the necessity of proper educational efforts, such as courses and tutorials, in particular in developing countries [40]. In summary, the development of a digital education ecosystem is one of the major needs for an effective DT. By way of example, actions in this direction are planned in Europe [244] and recommendations are given in the U.S. [245]. In terms of co-creation, its widespread adoption within PAs requires relevant structural changes, including a sourcing strategy, a governance structure, and a more flexible digital infrastructure, as reported in [60].

- **Process** It is clear from Sections 1.3 and 2.5 that the way in which projects are designed, managed and implemented must change in order to achieve an effective DT. Agile technologies are one technical way of realizing such a change. However, the DT is a dynamic process that may generate the need for “new and higher transformations” that may impact the mission of an organization. For instance, the IT department of a large Finnish municipality, transformed its mission from problem-solving to proactive service delivery, partly through a collaborative approach with business units, as reported in [74]. Therefore, Agile technologies may well be the “tools”, but a clear plan of what is DT is essential. Such a plan and vision may change depending on the scale (local, regional, national), although some coherence among the various levels of the scale must be ensured. To the best of our knowledge, a DT approach that accounts for the granularity and hierarchy of the components involved is not present. It is to be said that Agile technologies reinforce the need for capacity-development of stakeholders [79], i.e., the acquisitions of digital skills. Moreover, although more collaborative project management approaches are felt as necessary with the goal of interoperability, the lack of agreed processes, the difficulties of interpreting administrative and legislative procedures, and the difficulty of defining authorities and responsibilities are just some of the reasons why interoperability between PAs is not achieved, as outlined in [21]. Again, solutions to this problem are related to the scale at which we look at DT: interoperability may be simple to achieve in a restricted and uniform community and much more difficult in larger and more heterogeneous ones. As for indexes and maturity models, as outlined earlier in the relevant section, they have limits. In particular in regard to their scalability, e.g., a model that works well on a national scale may be too coarse to be applied to a local organization. Although it is quite complex to have a universal maturity model, an effort has to be made in order to devise models flexible enough to scale well with the complexity of the organizations where they are supposed to be used. Possibly, hierarchical maturity models could be a promising avenue of research. It is also evident that there is a lack of organic indications on how to govern the transition of project management in PA towards Agile, and how Agile approaches must be specialised to be suitable for the PA. A homogeneous proposal pointing to these future directions is a contribution that this Dissertation offers in the second and third parts. In addition, it is essential to have software engineering tools that support collaboration between professional figures and teams within Agile approaches. The third part of this Dissertation is devoted to this topic.

2.7 Conclusions

In this chapter, we discussed the main challenges of digital transformation in the public sector, in terms of the identified knowledge domains (pillars) and provided concrete examples of what is available in the Literature for each of them. Finally, we have also offered suggestions for each pillar, hoping to facilitate the PA in the planning aspects of its DT strategy.

Part II

From Generic to Specific Agile Methodologies for the Digital Transformation in the Public Sector

Motivation

The aim of this part of the Dissertation is to provide sound approaches to the adoption of Agile methodologies for the development and management of PA digital services, a central part of DT. It consists of two chapters, whose contents are briefly presented next.

Chapter 3 presents the generic textbook material related to the comparison of traditional (Waterfall) versus Agile methodologies, necessary to make this part of the Dissertation technically self-contained. Then, among the many Agile methodologies available, we present in detail Scrum, which, in light of the Literature collected for this Dissertation, seems to be the most widely used in the few examples that offer technical details on how to actually apply this methodology in the PA context.

In order to implement Agile methodologies successfully, it is essential to have software engineering tools that support them, particularly regarding the collaboration aspects. In Section 3.4, among the available tools to support Agile practitioners, we present the "Kanban Boards", an efficient task management tool that provides a visual representation of work elements to provide visibility, efficiency, collaboration, and adaptability, all of which contribute to more successful project outcomes and stakeholder satisfaction. Then, we briefly discuss a specific commercial software tool, widely used by the Agile community, that integrates the Kanban Boards internally, e.g., Jira.

According to what was anticipated at the end of Section 2.5, Chapter 4 contains our proposal on how to transition from current to Agile methodologies in the PA and how those latter can be specialized for and then successfully deployed in the Public Sector. This proposal is based on a careful reading, with analysis and synthesis, of the Literature presented in the first part of this Dissertation. Based on that, we propose original technical solutions, which are in line with, and to some extent generalizes, the recommendations provided by the US and UK Governments [246–248], over the years, in addition to more specific experiences such as the Bulgarian Public Sector [66], the Italian Army Agile Initiative (IAA) [249–253], and the Agile project management of the Barcelona City Council [254]. First, we present a set of recommendations for the transition to Agile methodologies, motivating them in terms of material presented earlier in this Dissertation. Second, in view of what we presented in Section 1.3, it is clear that new professional figures are needed, with expertise both in DT and Agile methodologies. Therefore, we propose them, in terms of a taxonomy, again with motivation coming from the first part of this Dissertation. Those new professional figures, in turn, also requires the introduction of new teams, that must interact with each other.

To test the validity of our proposal, in terms of development, we consider a particular use case related to the application of a variant of Scrum adopted by the Municipality of Barcelona, called Scrum@IMI, for the development of public services offered by PA. How the Barcelona use case was derived from our paradigm is described in detail in Section 4.6 of this Dissertation.

As mentioned in 2.5, of the first part of this Dissertation, the scalability of Agile project management methodologies is essential for their use in the PA. To establish a relationship between large-scale Agile methodologies and the one proposed here, in Section 4.7 we compare our methodology with Essential SAFe since it is the only one for which there are some technical details for its use in the PA.

Chapter 3

A Quick Look To Agile Methodologies

Abstract

This chapter first presents the basic principles of Agile methodologies. Next, we compare them with those related to traditional Waterfall methodologies. Then, among the various Agile methodologies, we focus on the Scrum methodology, which seems to have more technical details on how it can be used in the context of digital service development for the PA. Finally, among the available tools to support Agile practitioners, we present Kanban Boards, and briefly discuss a specific commercial solution, widely used by Agile practitioners, that integrates them within it, e.g., Jira.

3.1 Introduction

Agile, based on specific values and principles that are described in the Agile Manifesto published in 2001 [255], is a set of methods whose aim is to facilitate the migration of production processes from a planning-based (also known as “Waterfall”) logic toward a more iterative and incremental approach. Emerging as a response to the limitations of traditional Waterfall approaches to software development, and more in general to project management, which are often rigid and heavily focused on upstream planning, Agile methodologies refer to a set of principles and practices that prioritize collaboration, flexibility, and responsiveness to change [256]. These practices are aimed at improving software development by emphasizing a more iterative and adaptable approach, which allows for more efficient workflows and better outcomes. It has had quite a bit of success and it is now extensively used in many areas of project management, including Software Engineering. The reason for their success is due to the fact that organisations seek to become more responsive to changing customer needs and market conditions. By now, there is ample coverage of the Agile methodologies, even in textbook material, e.g., [249, 254, 257–262]. However, it is important to point out three incarnations of Agile, i.e., Scrum [258], Kanban [263], and Extreme Programming (XP) [264]. The content of this chapter, which is necessarily succinct with respect to the mentioned body of knowledge regarding Agile, concentrates on the following. Section 3.2 presents a comparison of Agile and Waterfall project management methodologies. Then, in Section 3.3, we explore the Scrum approach, which as we have already mentioned, seems to be the most suitable methodology for its implementation in the context relevant to this Dissertation.

Finally, in Section 3.4, among the tools made available to Agile teams, we present one in particular, namely, **Kanban Boards**. Then, we highlight, only for instance, the commercial software engineering solution that integrates them and is widely used by Agile practitioners.

3.2 Waterfall Versus Agile

As stated in the previous section, one of the main goals of DT is to increase the responsiveness of a public organization to the changing needs of citizens, e.g., [123]. It is evident that in response to changes, such a goal can be reached by being able to: (a) quickly use novel technologies, e.g., [81]; (b) implement an inclusive strategy that promptly makes available new skills to citizens, administrators and policy-makers, e.g., [15]; (c) adopt flexible organizational models for the design, implementation and deployment of services, e.g., [120]. Given such a characteristic feature of the DT, a traditional Waterfall software development and project management is no longer appropriate [194], i.e., the production of a full software product at the end of a sequence of phases does not comply well with the requirement of responsiveness. For the convenience of the reader, the two approaches are briefly described in 3.1 and now we highlight their main differences, following [246]. The interested reader may find additional comparisons in [265].

- **Timing and Scope of Software Development and Delivery**. In an Agile project, the development of working software is done in iterations that have a predetermined and equal duration, with each iteration providing a segment of functionality. For each iteration, the development team identifies the requirements, designs and develops software to meet those requirements, and tests the resulting software to determine whether it satisfies the stated requirements. On the other hand, plan-driven development follows a sequential and non-constant process that involves fixed-duration stages to produce a complete system. Only in the final phase, the software is tested and verified for compliance with the identified requirements.

- **Timing and Scope of Project Planning**. In Agile project management, the initial planning for cost, scope, and time is conducted at a high level. However, these plans can be updated with more specific plans iteration after iteration, and these plans can be revised based on lessons learned from iterations already completed. In contrast, traditional Waterfall project management requires detailed documentation of this analysis at the beginning of the project for the entire scope of effort.

- **Project Status Evaluation**. Agile approaches evaluate project status by showcasing workable product software to customers and stakeholders during a single iteration. This is in contrast to traditional project management where progress is evaluated through a review of data and documents at predetermined milestones and checkpoints. These reviews may occur at the end of the requirements definition or at scheduled intervals such as monthly check-ins.

- **Collaboration**. Collaboration is a key aspect of Agile project management, which distinguishes it from traditional approaches. In Agile, customers, e.g., PAs, work closely with technical professionals to coordinate tasks such as design and testing. The teams in Agile projects are typically self-directed, meaning that tasks and deadlines are managed within the team, with coordination as needed from project sponsors and stakeholders. In contrast, traditional project management often separates customers and technical workers, with project activities being prescribed and supervised by a Project Manager, who reports to organizations such as the project management office of a given PA.

Since responsiveness in DT must account for a very dynamic environment, it terms of regulations, technology and citizen needs, it is evident from the differences outlined above that Agile is better suited for software development and project management sustaining the DT. For completeness, we mention that, on occasions, the use of both methodologies can be helpful [68].

3.3 Scrum: A Relevant Agile Methodology for this Dissertation

Scrum describes an adaptable, fast, self-organizing product development process that originated in Japan [266]. The term Scrum is derived from a rugby tactic that indicates "getting an out-of-play ball back into the game" through collaboration [266]. The Scrum technique was created to manage the system development process. It is an empirical method of system development that applies the notions of

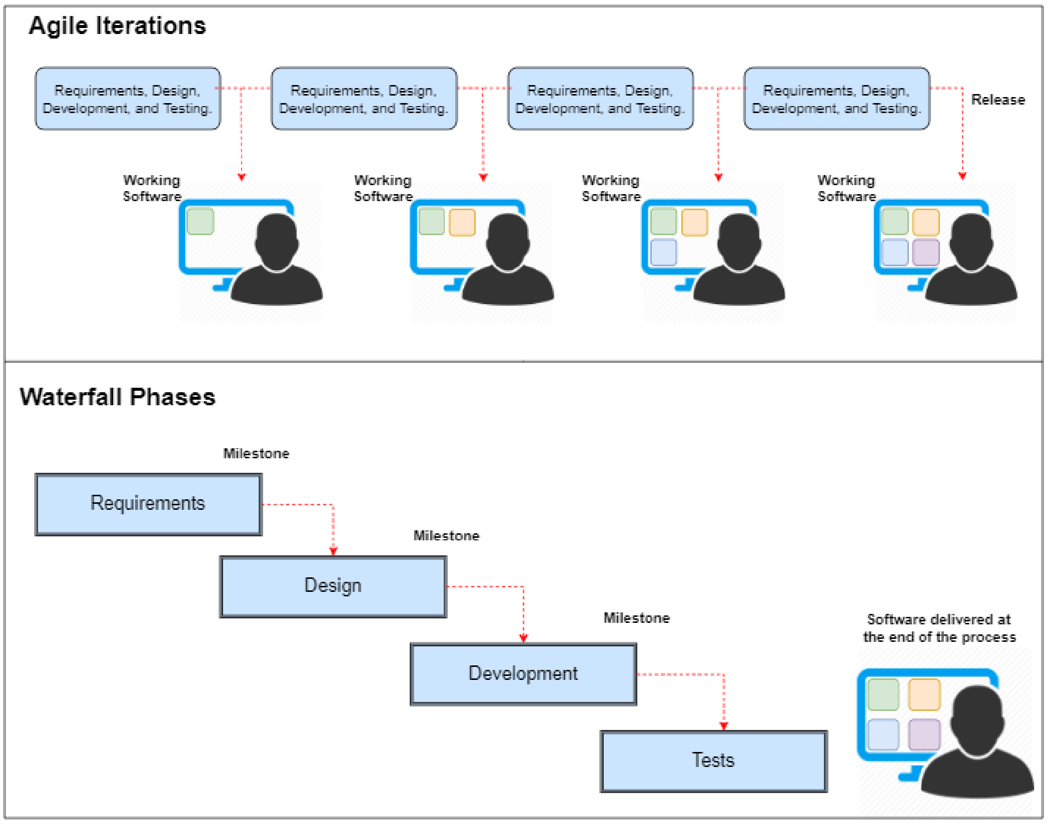

Agile Iterations
Requirements, Design, Development, and Testing.
Requirements, Design, Development, and Testing.
Requirements, Design, Development, and Testing.
Requirements, Design, Development, and Testing.
Release
Working Software
Working Software
Working Software
Working Software
Waterfall Phases
Milestone
Requirements
Milestone
Design
Milestone
Development
Tests
Software delivered at the end of the process

Figure 3.1: **Waterfall versus Agile Software Development Approaches**. The process is sequential when using Waterfall methodologies. It starts with the requirements collection, planning, and talent and technology acquisition process. The product is then designed and built. Finally, it is tested and made available to the public. Only at this point do you get feedback and understand whether the product works for citizens. Since there is no going back to the previous stages, one has only one chance to get each part of the project right. Agile methodologies offer a unique approach. These are an iterative and incremental approach to software development. All previous tasks are done simultaneously: requirements collection, planning, design, construction, and testing.

industrial process control theory, resulting in an approach that reintroduces the concepts of flexibility, adaptability, and productivity [266]. It does not provide any specific software development strategies for software implementation. It focuses on how team members should interact in order to develop a flexible system in a continuously changing environment. The core assumption of Scrum is that system development involves a number of environmental and technological factors (e.g., requirements, time frame, resources, and technology) that are subject to change during the process. This makes the development process unpredictable and complex, necessitating system development process flexibility in order to adapt to changes. As a result of the development process, a system is created that is useful when delivered [266].

We now describe the fundamental aspects of Scrum, following closely the work of [256], that is, describing the process, roles and responsibilities of the main professional figures involved in the project, practices, and usage examples of the methodology.

- **Process**. The process is defined as the description of the stages in the product lifecycle through which the software is produced.
- **Roles And Responsibilities**. They refer to the assignment of specific roles through which software is produced in a development team.
- **Practices**. Practices refer to specific activities and deliverables defined by a method for use in the process.
- **Adoptions And Experiences**. They refer primarily to existing experience reports on the use of the method in practice and to the method developers on how the method should be introduced in an organisation.

Process

Scrum process includes three phases: **Pre-Game**, **Development** and **Post-Game**, as depicted in Figure 3.2. The Pre-Game phase includes two sub-phases: **Planning** and **Architecture/High Level Design**.

The definition of the system under development is part of the Planning process. A Product Backlog list (see Section 3.3) is created with all presently known requirements. Those latter could arrive from the client, the sales and marketing division, customer service, or the software developers. The requirements are prioritized, and the time required to carry out them is estimated. The Product Backlog is always being updated with new and more specific items, as well as more precise estimates and new priority requirements. Planning includes establishing the project team, tools and other resources, risk assessment and control issues, training needs, and verification management approval. The revised Product Backlog is assessed by the Scrum Team at each iteration in order to acquire their commitment for the next iteration.

The Development phase (also known as the *Game phase*) is the agile part of the Scrum process. The various environmental and technical variables identified in Scrum, which may change during the process (e.g., time frame, quality, requirements, resources, implementation technologies and tools, and even development methods), are observed and controlled through various Scrum practices during the Sprints (see Section 3.3). Rather than considering these issues merely at the start of a software

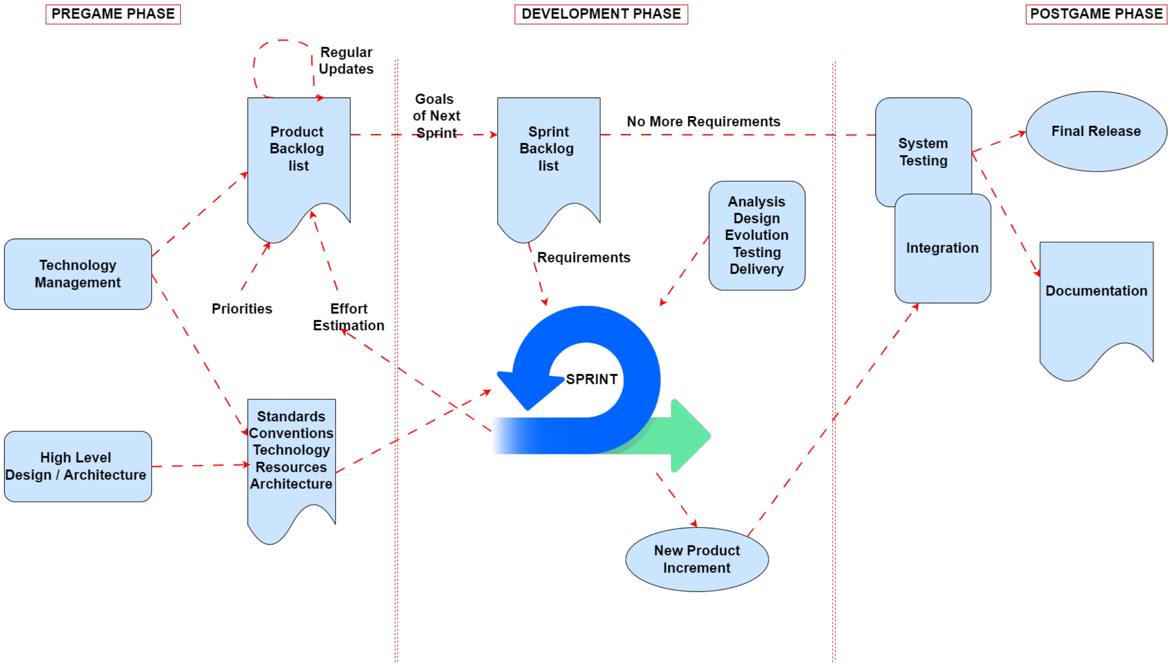


Figure 3.2: **Scrum Agile Methodology**. Process Life Cycle.

development project, Scrum seeks to regulate them continuously in order to be able to respond to changes in a flexible manner.

Sprints are used to create the system during the Development phase (see Figure 3.2 and Section 3.3). Sprints are iterative cycles in which functionality is constructed or improved to generate new increments. We emphasize how an increment represents all the valuable work produced by developers during a Sprint. The sum of all increments forms a product. Each Sprint contains the conventional software development phases, typical of the Waterfall approaches, specifically: Requirements, Analysis, Design, Evolution, and Delivery. During the Sprint development, the architecture and design of the system evolve. One Sprint will last from one week to one month. In one system development process, i.e., there may be three to eight Sprints before the system is ready for release. Additionally, the increment may be produced by more than one team. The release concludes in the Post-Game phase. This phase begins once it has been established that the environmental factors, such as the requirements, have been finished. In this circumstance, no new items or issues can be identified, nor can any new ones be formed. The system is now ready for release, and preparations for this are being completed during the Post-Game phase, which includes tasks such as integration, system testing, and documentation.

Roles and Responsibilities

Six identifiable roles in Scrum have different tasks and purposes during the process and its practices: **Scrum Master**, **Product Owner**, **Scrum Team**, **Customer**, **User** and **Management**. Here, according to the definitions of [266], we describe these roles in further details:

- **Product Owner**. S/he is formally responsible, on behalf of the contracting PA, over the project, contract and compliance aspects, priority management, regulating, and making the Product Backlog list available. The Scrum Master, the Client, and Management all vote to choose her/him. S/he makes final choices on product Backlog tasks (see Section 3.3), contributes to estimating development work for Backlog items, and converts issues in the Backlog into features to be implemented.

- **Scrum Master**. S/he is an expert in the use of Agile methodology and plays a central role in coaching the development team and supporting them in managing the Product Backlog. During the project development cycle, the Scrum Master communicates with the development team, Product Owner (customer) and Management. In the context of the PA, it can also be an administrative officer with experience and knowledge in Agile methodologies. The Scrum Master helps the development team solve problems in the use and adaptation of the Agile methodology to the specific context in which the team works. S/he also drives the continuous improvement expected of any agile team. Another task of the Scrum Master is to protect the development team from outside interference and to guide the team in removing obstacles that may negatively impact productivity. S/he is still a facilitator, but he must take control of the team, taking into account the different cultural backgrounds of the members. The Scrum Master and Product Owner must collaborate closely and continuously to clarify requirements and obtain feedback. S/he must also take responsibility for overall progress and initiate corrective actions when necessary. In addition, S/he must build and sustain effective communications with key personnel and organizational elements involved in development as required.

- **Scrum Team**. It is the development team, which can decide on the necessary activities and organize itself to achieve the goals of the Sprint. It is typically composed of three to nine members, all experienced software developers. It must be cross-functional, i.e., formed by a group of people with different functional expertise working toward a common task. Scrum team members can be internal or external to the Organization working on the project. Each member offers an alternative perspective to the problem and a potential solution to the task. The Scrum team is involved in activities such as estimating effort, defining the Sprint Backlog (see Section 3.3), evaluating the Product Backlog list, and recommending obstacles to be removed from the project. All development team members are collectively responsible for the implementation of the entire project, step by step, increment by increment, and there are no specialists who focus on limited areas such as design, testing, or code development. All team members contribute to the entire production. Although, as mentioned above, team members may come from different companies, the team is self-organized in determining without external constraints how to manage the Sprint Backlog (the selected part of the Product Backlog that has been identified to be executed in a Sprint (also known as *iteration*)) according to the priorities defined by the Product Owner.

- **Customer**. S/he takes part in tasks related to product Backlog items (see Section 3.3 for the system that is being developed or improved.

- **User**. Scrum users are responsible for frequently monitoring the progress of released Scrum artifacts to detect any unwanted deviations, through the use of the software product. It is useful to recall the concept of a Scrum artifact, which is the information that the Scrum team and stakeholders use to define the product to be developed, the actions needed to produce it, and the actions performed during the project. The main Scrum artifacts are the Contract, Product Backlog, Sprint Backlog, Definition of Done, Increment and Compliance. Scrum teams can have internal users (e.g., civil servants) as well as external users (e.g., citizens) or a mix of internal and external users. All of this determines the place of users in Agile methodologies. The Sprint Review is the most sensible place for user participation. It focuses on conversations and input from stakeholders.

- **Management**. The Management (stakeholder) is defined primarily by the interactions S/he has with other members of the Scrum Team. S/he is responsible for final decision-making, as well as project documents, standards, and conventions. S/he is involved in the development of goals and requirements. For example, it participates in Product Owner selection, progress monitoring, and Backlog minimization together with the Scrum Master. It actively collaborates in the process of regular inspection of Scrum artefacts to detect unwanted variations and provide opportunities to correct them (inspect/adapt paradigm). It is not a monolithic figure. The Product Backlog can express the interest of different stakeholders, and it is on the satisfaction of these interests that each Sprint Goal is defined. Finally, stakeholders have a role in reviewing, during the review and together with the Scrum Team, what has been completed (Definition of Done).

Practices

Scrum does not require or recommend any specific software development methodology or processes. Instead, particular management strategies and tools are required in the various Scrum phases to reduce the chaos generated by unpredictability and complexity [261]. Following is a description of Scrum practices based on [266].

- **Product Backlog**. Based on existing knowledge, the Product Backlog describes everything that is required in the final product through writing epics and user stories. Recall that: a user story is a general, informal explanation of a software feature that is written from the perspective of the end-user to articulate how it provides value to the customer. On the other hand, an epic is a large user story that would take more than a few weeks to develop and test). If possible, split a large story or epic into smaller user stories that can be completed within an iteration. Thus, the Product Backlog specifies the tasks of the project. It is a prioritized and regularly updated list of business and technical requirements for the system that is being implemented or improved. Backlog items might contain features, functionalities, bug patches, faults, desired additions, and technological upgrades, among other things. The list also includes issues that must be resolved before other Backlog items may be completed. Customers, project teams, marketing and sales, management, and customer support are all examples of players that might contribute to the creation of Product Backlog items. This practice comprises responsibilities of creating the Product Backlog list and continuously controlling it throughout the process by adding, deleting, specifying, updating, and prioritizing Product Backlog items. The Product Owner is responsible for maintaining the Product Backlog up to date.

- **Effort Estimation**. When additional information is provided on a specific Product Backlog item, effort estimation becomes an iterative process in which the Backlog item estimates are focused on a more precise level. The Product Owner, in collaboration with the Scrum Team, is responsible for handling estimating work.

- **Sprint**. Sprint is the process of reacting to changing external variables such as requirements, time, resources, expertise, technology, and so on. The Scrum Team organizes itself to create a new executable product increment over an amount of thirty calendar days in a Sprint. Sprint Planning Meetings, Sprint Backlog, and Daily Scrum Meetings are the working tools of the team (see below). The Sprint, its practice and inputs are depicted in 3.3.

- **Sprint Planning Meeting**. The Scrum Master organizes a Sprint Planning Meeting in two phases. In the first phase of the meeting, Customers, Users,

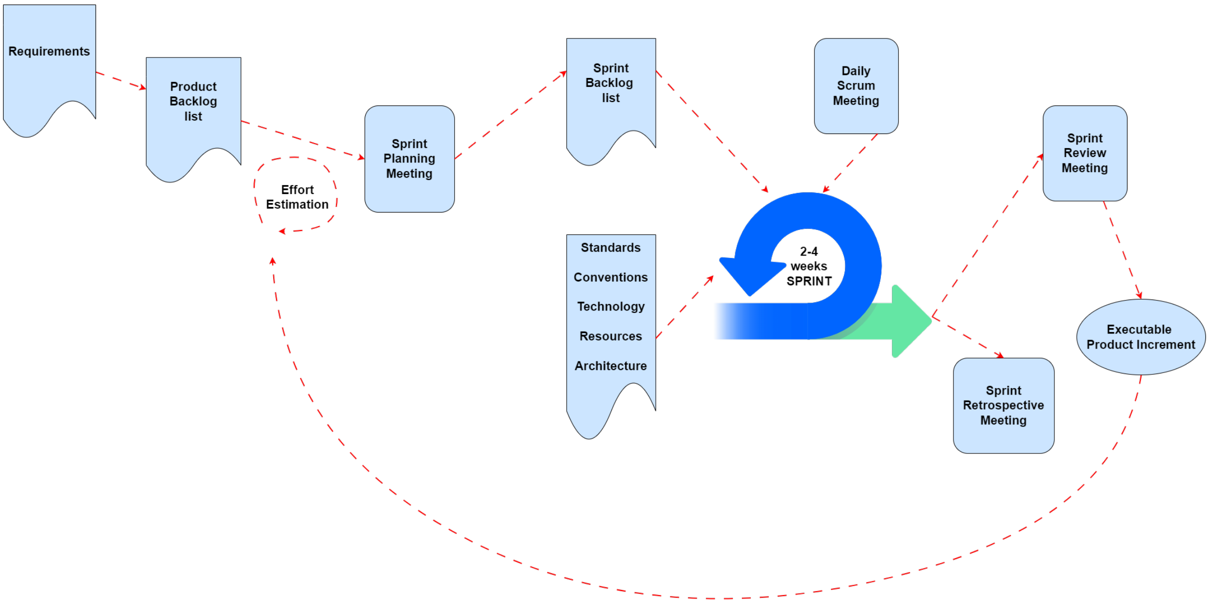


Figure 3.3: **Scrum Agile Methodology**. Sprint Workflow.

Management, Product Owner, and the Scrum Team decide on the goals and features of the next Sprint (see Sprint Backlog below). The Scrum Master and the Scrum Team attend the second phase of the meeting, which focuses on how the product increment is implemented during the Sprint.

- **Sprint Backlog**. Each Sprint begins with the Sprint Backlog. It is a list of Product Backlog items that have been chosen to be implemented in the next Sprint. In the Sprint Planning Meeting, the Scrum Team, along with the Scrum Master and the Product Owner, choose the items based on the prioritized items (see Section 3.3) and Sprint goals. The Sprint Backlog, unlike the Product Backlog, remains stable until the Sprint (e.g. four weeks) is completed. When all of the Sprint Backlog items have been finished, a new version of the system is deployed.

- **Daily Scrum Meeting**. Daily Scrum Meetings are held to maintain track of the work of the Scrum Team in real time, as well as to plan what has been done since the last meeting and what has to be done before the next one. This short (approximately 15 minutes) discussion is also used to address and control challenges and other changeable concerns. Any shortcomings or barriers in the systems development process or engineering methods are investigated, discovered, and eliminated to enhance the process. Scrum meetings are directed by the Scrum Master. The meeting can be attended by anyone other than the Scrum team, such as Management.

- **Sprint Review Meeting**. The Scrum Team and the Scrum Master provide the Sprint results (i.e. working product increment) to Management, Customers, Users, and the Product Owner in an informal meeting on the last day of the Sprint. The participants evaluate the product increment and decide on the next tasks. The review meeting may result in the addition of additional Backlog items or possibly a change in the direction of the system being developed.

- **Sprint Retrospective Meeting**. The goal of the Sprint Retrospective is to plan strategies to improve quality and efficiency. The Scrum Team evaluates how the last Sprint proceeded in terms of individuals, interactions, processes, tools, and their Definition of Done. The elements that are selected vary depending on the work environment. The assumptions that have led them off the road are identified, as are their origins. The Scrum Team discusses how successfully the Sprint went, what problems emerged, and how those problems were (or were not) resolved. The Scrum Team identifies the most useful changes to improve their own efficiency. Improvements with a higher impact are addressed as soon as possible. They could also be added to the Sprint Backlog for the next Sprint. The Sprint Retrospective concludes the Sprint. It is limited to a maximum of three hours for a monthly Sprint. For the shorter Sprints, the event is solely shorter. The Scrum Master encourages the rest of the Scrum Team to improve their processes and practices in order to make them more efficient for the next Sprint. During each Sprint Retrospective, the Scrum Team plans strategies to improve product quality by improving work processes or adapting the "Definition of Done" when appropriate and not in conflict with product or organizational standards. By the end of the Sprint Retrospective, the Scrum Team should have identified the improvements that will be implemented in the next Sprint. Implementing these enhancements in the next Sprint represents an adaptation to the Scrum Team internal inspection. Although improvements can be implemented at any time, the Sprint Retrospective provides a formal opportunity to focus on evaluation and adaptation.

Adoptions and Experiences

Although Scrum has the potential to significantly alter the job descriptions and conventions of the Scrum project team, as already mentioned, Scrum is not dependent on any specific engineering techniques, therefore it may be used to manage any engineering processes that an organization adopts. The project manager, i.e. the Scrum Master, for instance, no longer supervises the team; instead, the team organizes itself and decides what to get done. The role of the Scrum Master becomes more and more that of a coach and less that of a manager. S/he must be able to remove process barriers, make decisions in daily Scrums and evaluate them with the Management.

According to [266], the authors have identified two types of situations in which Scrum can be adopted: an existing project and a new project. These are detailed further below.

- **Existing Project**. Introducing Scrum into an existing project is a common scenario when the development environment and technology are already in place, but the project team challenges itself with changing requirements and the complexity of the technology. Scrum implementation would begin with daily Sprints facilitated by a Scrum Master. The first Sprint should aim to demonstrate any user functionality on the selected technology, as stated in [266]. This helps the team trust itself and encourages the customer to invest in the team. During daily Scrums in the first Sprint, the team identifies and discusses any project obstacles to enable the team to move forward. The client, Scrum Team, and Scrum Master meet for a Sprint Review at the end of the first Sprint to decide on the next steps. If the project continues, a Sprint Planning Meeting is held to determine the objectives and requirements for the next Sprint.
- **New Project**. The authors suggest spending several days with the team and the Client to establish an initial Product Backlog, which may include business tasks and technological requirements. The primary goal of the first Sprint is to demonstrate a key piece of user functionality on the selected technology. For this, a basic functional structure of the system should be planned and built, to which additional features can be added later. The Sprint Backlog should contain all the necessary work required to achieve the Sprint goal. As the primary challenge at this time is the adoption of Scrum, the Sprint Backlog includes tasks such as establishing the Team and Scrum roles, developing management procedures, and executing the demo. Meanwhile, the Product Owner interacts with the Customer to create a more comprehensive Product Backlog while the Scrum Team is working on the Sprint Backlog. This helps to schedule the next Sprint following the first Sprint Review.

As an example of profitable adoption of the Scrum framework, three different use cases from software development projects are given in [260]. Moreover, the paper [259] discusses the results of a survey that examines the challenges and factors affecting the adoption and success of agile methods, such as Scrum, including lack of Management commitment, unclear objectives, and inexperience of the development team regarding the competencies to manage projects in an agile manner. At the same time, indications are given on the size of the Scrum Team, i.e., the most common team size for Scrum projects is between seven and nine practitioners. If more than nine people are available, multiple teams should be formed.

Recently, efforts to integrate eXtreme Programming Methodology [264,267], also known as XP, and Scrum can be found in [268–271]. Scrum provides a project management framework that is reinforced by XP practices to provide an integrated package for software development teams, allowing XP to be more scalable on larger projects. For more details regarding the integration among these Agile methodologies, see, e.g., [268–271].

3.4 Agile Tools Supporting Scrum-Like Processes

As mentioned in Section 2.6, Agile methodologies require effective collaboration between professionals and teams. To enable this collaboration, software engineering tools are essential. In this section, we will briefly discuss the primary tools that support Agile practitioners in managing Scrum processes.

Among the tools made available to Agile teams, Kanban Boards are widely used. Here, we present some specifics about them. The idea of Kanban originated in Japan among automobile manufacturers. Unlike traditional project management systems, Kanban is anchored in Lean values and thinking [272]. David Anderson, who experienced Lean concepts while working at Microsoft and Corbis, developed Kanban as a method for improving the software development process. His approach involves identifying recurring problems, creating tools to solve them, and analyzing the current way of working as a series of adjustable, repeatable steps. Kanban starts with the existing team process, making it a flexible approach to process improvement rather than a single solution. Teamwork is all about recognizing the habits and patterns of building software and documenting each step taken. There are already project management systems like XP and Scrum in place to manage these processes. Kanban improves upon these systems by analyzing and understanding the current project management system. The ultimate goal of Kanban is to make incremental and evolutionary improvements to the system. Many Scrum teams rely heavily on stories as their input and produce code as their output. The Kanban Board is a visual representation of the workflow of the team. This board usually consists of columns drawn on a whiteboard, with sticky notes pasted in each column. Each sticky note on the board represents a user story. The column titles on the Kanban Board vary from team to team based on how they do their work, for example, **“To Do”**, **“In Progress”**, **“Test”**, and **“Done”**. This scenario is represented in Figure 3.4.

The Kanban Board can set a limit on the amount of work in a single column. When a team adopts Kanban, the first step is to visualize the workflow by creating a Kanban Board. The Kanban team holds a daily meeting called “walking the board”

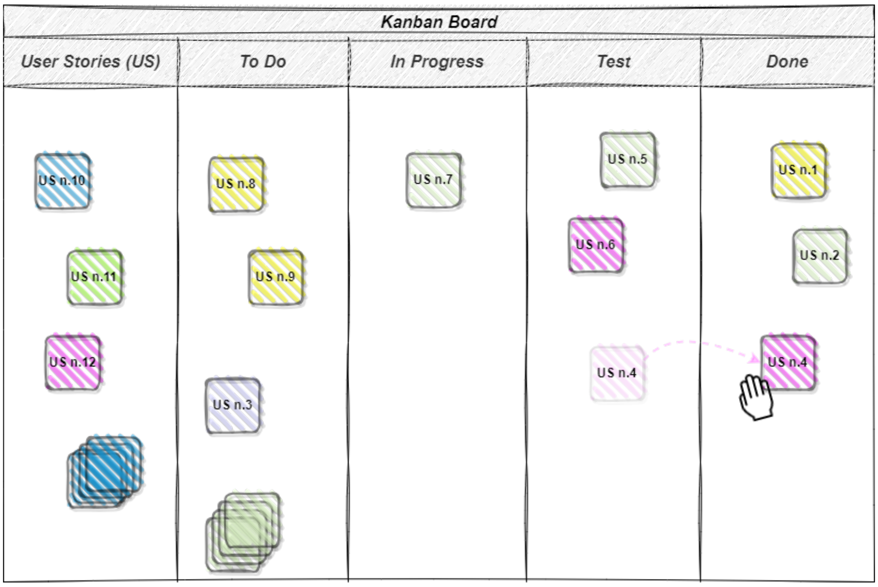


Figure 3.4: **Typical Kanban Board Scenario**. A sticky note is placed in the first column of the board for each of the user stories (u.s.) to be developed. Then, depending on the selection of user stories from the Product Backlog to be developed, the relevant sticky notes are moved to the "**To Do**" column. When the development team starts with the production of the code corresponding to the u.s. the sticky note is moved to the "**In Progress**" column. When the development of the u.s. is completed, the testing phase is carried out, so the sticky note should be moved to the "**Test**" column. Having passed the tests, the u.s. moves to the "**Done**" state (see the transition of u.s. n.4 from the **Test** column to the **Done** column).

to discuss the status of each item on the board. The board should already reflect the current state of the system. Every item that has completed a step should have its sticky note moved into the next column. The team ensures that the board is up-to-date during the meeting.

A commercial project and task management tool, widely used by the Agile community, developed by Atlassian [273], that integrates the Kanban Boards to help teams plan, track, and manage tasks effectively, whether in software development or other areas, is Jira [274]. This tool allows users to create and assign tasks, or "issues" as they are called in Jira, to team members. These issues can represent tasks, bugs, feature requests, or anything else that requires management within a project. Jira offers highly customizable Kanban Boards. Users can assign issues to Kanban Board columns to indicate their current status. This software solution offers advanced automation capabilities that allow users to create rules for the automatic movement of issues between columns based on certain conditions. This simplifies workflow management and reduces manual work. Jira offers a number of reports and dashboards that enable teams to monitor project performance. These reports provide key metrics such as average issue completion time, team member workload, and more. Jira can be integrated with other applications used in the software development cycle, such as Slack [275](for communication), Bitbucket [276] (for source code management) and many others. These integrations make it possible to create a complete and integrated development environment.

Part three of this Dissertation presents and discusses our additional contribution to this Dissertation regarding the creation of an integrated and collaborative development environment.

3.5 Conclusions

In this chapter, we discussed the main differences between traditional and Agile project management methodologies. Then, among the different Agile methodologies, we focused on one in particular, i.e., Scrum, since it seems to turn out to be the one for which we have found more technical evidence in the Literature regarding its adoption in the PA context, highlighting its main features in terms of process, roles and responsibilities of the professional figures and teams involved, practices and use cases. Finally, as already anticipated at the end of Section 2.6, of the first part of this Dissertation, given the importance of software engineering tools, supporting Scrum-like processes, we also offered some suggestions about the main tools used by Agile professional figures. The latter topic is our further contribution to this Dissertation and is to be covered in part three of this Dissertation.

Chapter 4

Agile Methodologies For the Public Administration: A Novel Proposal

Abstract

This chapter proposes a framework for applying Agile methodologies in the context of a given PA. We start with a set of recommendations specific to DT. The set of recommendations consists of two groups: one general and the other specialized. Then, we provide a taxonomy of professional figures, useful for dealing with the creation of a digital service offered by the PA over its entire lifecycle. We place these professional figures first in categories and then in Job Families. Then, we specialize the Agile Scrum methodology with a set of Leaders and Team Members Figures, mapping them into the categories and Job Families mentioned earlier, and describing the main interactions between them through a graph model. Based on this model, we show its validity by describing how the Scrum@IMI approach of the Barcelona Municipality can be derived from it. Finally, in view of what has been stated in Section 4.7, we relate our proposed methodology referred to as Scrum@PA, to the Agile framework most widely used by the Agile community for large-scale project management, i.e., SAFe, in its "Essential" configuration.

4.1 Introduction

This chapter proposes, for the first time, a general and technically specific paradigm for the adoption of Scrum in the PA context. It is abstracted from the few Agile experiences that provide technical details on how to use Agile in the PA, namely: the Municipality of Barcelona [254], the Bulgarian Public Sector [66], and the Agile development experience of the Italian Army [249–253]. This paradigm is the end result of a series of steps that we have distilled from the state of the art and organized each of them in order to obtain a homogeneous view for the adoption of Agile methodologies in the PA. Those steps are highlighted next.

We start by describing the transition from traditional (Waterfall) to Agile methodologies in the context of DT (Section 4.3). For this purpose, we first present a list of DT-specific recommendations, which are consistent with what has been in place by the US and UK Governments [246, 248, 277, 278]. These recommendations are divided into two groups: one generic and the other more specialized. Then, we define a taxonomy of new professional figures that we first divide into categories, and within these, into Job Families. This taxonomy is useful to address the creation of digital services offered by the PA, and is consistent with what is done by the UK Government [247].

Ones we have defined the above professional figures, we place them in the context of the Agile Scrum methodology to be able to use this methodology in the PA context (Section 4.4). To this end, we have differentiated those figures between Leaders and Teams. The choice of Scrum is due to the fact that the very few technical approaches to Agile methodologies in the PA use that particular version of Agile.

Then, in Section 4.5 we propose the mentioned paradigmatic Scrum model for its adoption in the context of the public service development offered by PA, focusing particular attention on the service development phase. We first extracted the components of the development phase, identifying for each of them the internal interactions among the figures involved, and then described the external relationships among the identified components. Based on our paradigmatic Scrum model, a Public Administration has a technically sound avenue to follow to adopt Scrum, rather than a generic set of guidelines as in the current State of the Art.

As for the validation of our approach, it is useful to recall that there are no accepted methodologies for the DT [279, 280]. However, being Scrum@PA a paradigm, its validity can be assessed by showing that existing methodologies for Agile in the DT are special cases of Scrum@PA. This is immediate when standard Scrum is applied, as in the cases described in [281]. For more complex cases, although iAgile can be derived from Scrum@PA, in order not to clutter this Dissertation with details, we opt to concentrate on Scrum@IMI, given also the fact that it is public. Indeed, in Section 4.6, we describe how the quite successful Scrum@IMI can be derived from Scrum@PA.

Finally, scalability is a crucial subject that demands special attention. As mentioned in Section 2.5, of the first part of this Dissertation, although scalability is desirable for Agile in the PA, the only proposal that has some technical details is “Essential SAFe” [211]. Therefore, it is important to compare our proposal with the Essential SAFe framework. Section 4.7 discusses these aspects.

4.2 Limits of Standard Agile

As mentioned in the introduction, there are recommendations and guidelines on the application of the Agile methodologies in the DT, however, there are very few concrete examples [211] on how to really adopt them in the DT, since standard methodologies seem not to be enough. This point is well highlighted in a study reporting the development of an extension of standard Agile, denoted improved Agile (iAgile, for short) [250] for a specific PA. The context in which such a project management strategy has been first developed, and then successfully applied, is for the Italian Ministry of Defense. The considered software service had to meet very stringent security criteria, i.e., the service was strongly regulated and to be used within a hierarchical organization. In order to ensure the proper level of agility and software quality, the standard Scrum approach revealed itself inadequate in some of its key figures and one team was not judged sufficient to realize the project. The contribution of iAgile is to extend the responsibilities of some existing Scrum professional figure, introduce new ones, together with a set of teams whose coordination and cooperation goes beyond Scrum of Scrum [282], and to redefine the workflow of the software development process. The interested reader can find details in [250], together with an evaluation of the methodology.

Another compelling example is the one provided by the City of Barcelona. Here the area is also highly regulated, but emphasis is on the satisfaction of citizen needs, with quick reaction to their changes and interaction with them, privacy regarding user data, transparency of services and accountability of the Municipality. It is useful to recall that, from part I of this Dissertation, such a City set-up its own Agile policy [283] in order to address its specific, but rather broad, needs. Such a policy is supported by the Scrum@IMI methodology, an extension of Scrum [258], developed by the Institut Municipal d'Informática, City of Barcelona. It is a successful methodology since it is operational, i.e., regularly used for the development of their IT projects [254]. Moreover, it is fully documented in a public repository [284]. As in the case of iAgile, Scrum@IMI has requested the expansion of the responsibilities of standard Scrum roles, the introduction of new ones, as well as the use of more than one team.

4.3 A Transition to Agile Methodologies in the Public Sector

Based on the recommendations provided by the UK and US Governments, mentioned in the Introduction, and the two examples briefly discussed in the previous section, we provide a set of recommendations and new professional figures, that should facilitate the adoption of the Agile methodologies in the DT.

A Coherent Set of Recommendations

The aim of this section is to provide a set of coherent recommendations that can guide a PA towards the use of Agile methodologies in their organizational and project management initiatives. Those recommendations account also for the specifics of the DT. They can be divided into two groups. The first is rather standard, concerning the application of generic Agile techniques, e.g., along the lines that would be followed for the application of Agile methodologies discussed in Chapter 3 to a given specific context. Here the context is the PA and, at this level of detail, attention is given to the development of public services. On the other hand, the other group is much more detailed and specific, since it accounts for aspects, such as Data Governance and citizen involvement, that characterize the full spectrum of what is DT (see the first part of this Dissertation).

1. **General Recommendations**

1.1 The starting point is an Agile guide and an Agile adoption strategy. To this end, the entire content of Chapter 3, with the bibliographic references therein, can be useful.

1.2 The use of Agile terms and examples is essential to improve migration to Agile concepts. To this end, the entire content of Chapter 3, with the bibliographic references therein, can be useful.

1.3 The adoption of Agile methodologies, at both the organizational and project management levels, must be continuously improved, via the identification of the encountered obstacles and their resolution. To this end, the content of Section 3.3 of Chapter 3 can be useful.

1.4 The size of an Agile development team must be carefully determined. Teams with three to nine members should be preferred. Indeed, small teams may have difficulty completing product increments, causing a decrease in productivity, while large teams may encounter teamwork and communication problems due to increased complexity. This is suggested by the content of Section 3.3.

1.5 It is necessary to define an accurate list of the professional figures and related technical skills required to achieve Agile service development objectives. This is suggested by the content of Sections 1.3 and 2.4.

1.6 It is necessary to know the real needs of citizens by obtaining regular and continuous feedback from them and other stakeholders during all the phases of service development. This is suggested by the content of Sections I and 1.3.

1.7 It is necessary to establish an understanding of the interactions of citizens with the PAs to determine priorities for service transformation in an Agile manner and to educate citizens in the participatory policy process. This is suggested by the content of Sections 1.3, 2.2, 2.4 and 2.5.

1.8 Progress should be tracked using tools and metrics, on a daily basis, ensuring visibility to citizens and other stakeholders. The inclusion of security and progress monitoring requirements in the unfinished work pipeline is recommended, such as the Product Backlog (presented and discussed in Section 3.3), in order to achieve more trust regarding the value of the product increment at the end of each Sprint (presented and discussed in Section 3.3). This is suggested by the content of Sections 1.3, 1.3 and 2.5.

2. **Specialized Recommendations: Data and Technology**

The following recommendations are divided into two very specific parts for the DT: Data and Technology.

- **Data**

2.1.1 The creation of a Data Standards Authority, possibly presided by the Chief Data Officer (see Section 1.3), is recommended to improve awareness of Data Governance and its importance in service development, along with the creation of an appropriate framework for Data Quality Assessment. This is based on what has been presented and discussed in Sections 1.3 and 2.2.

2.1.2 The provision of a list of recommendations for Open Data Standards is necessary. Publication of Open Data, source code in a public repository, and utilization of open source software are crucial for enhancing transparency, flexibility, and accountability. This is based on what has been presented and discussed in Sections 1.3 and 2.2.

2.1.3 The creation of an Open Data ecosystem focused on the interoperability of various information sources of the involved PAs, aiming at establishing an exchange mechanism between citizens and the Government, according to the "*once-only*" principle and facilitating integration with future technologies by minimizing required efforts. This is based on what has been presented and discussed in Sections 1.3 and 2.2.

- **Technology**

2.2.1 The Creation of an intergovernmental reference architecture is necessary. Indeed, the use of public Cloud solutions should be considered as a first choice (*Cloud-First*). Obviously, Cloud Technologies and Smart Cities are natural candidates. This is based on what has been presented and discussed in Sections 1.3 and 2.3.

New Professional Figures: A Taxonomy

Here, we introduce two categories of professional Figures and, for each of them, we define the corresponding Job Families. The first category contains those Job Families related to Data Management and Information and Communication Technologies. The second category contains those Job Families related to the creation of a software product and the various stages related to its lifecycle.

- **Data Management & ICT Category**

D.1 **Data Job Family**. Given the attention that needs to be placed on Data (see the previous section, our graph model in Section 1.3 and what is discussed in Section 2.2), it is quite obvious that a whole range of specialized data figures, not provided by the classical Agile methodologies, are needed to take into account many of the aspects related to the Data Management and Governance, in particular regarding compliance with PA regulations, privacy and security. By way of example, a **Chief Data Officer** is needed, possibly in a leadership position.

D.2 **Technical Job Family**. In the classical Agile methodologies, it is clear that there must be technical figures, as evident from what has been presented in the previous Chapter. Here, in light of Section 1.3, and the previous section, semantics of these figures changes since they must have the ability to improve the architectural design of public services, while accounting for the needs of citizens. Although those figures can be part of Agile methodologies, here their semantic definition is centered on the DT and the PA. By way of example, a **Chief Technology Officer** is needed, possibly in a leadership position.

D.3 **Quality Assurance Testing (QAT) Job Family**. When the classical Agile methodologies are extended to the DT, the semantics of figures in this family changes, since they must focus on managing the complexity of the Testing process in the PA, as evident from what has been presented in the previous Chapter. This complexity emerges from the need to ensure the privacy and security of the data and services that are offered by the PAs, on the one hand, and compliance to regulations on the other hand. Here, in light of the previous Chapter, these figures must have the ability to consider many of the aspects related to: (a) the management of quality test plan, functional and not, appropriate to ensure that each requirement has been fully achieved; (b) the understanding and identification of techniques, datasets, and the most effective tools to be used; (c) clear and effective communication of defects or trends to the development team members, outlining how defects and possible causes are identified. By way of example, a **Chief Quality Assurance Testing Officer** is needed, possibly in a leadership position.

- **Product Creation & Support Category**

P.1 **Product Design and Delivery Job Family**. As in the previous Family, also here the semantics of the classic professional figures changes. Indeed, in light of Sections 1.3 and 2.4, and the previous chapter, they must have the ability to consider many of the aspects related to citizen-centered design to improve the content, comprehensibility and usability of the services delivered. Moreover, attention must be given to the continuous improvement of team skills as a central point of expertise. By way of example, a **Scrum Coach** is needed, possibly in a leadership position

P.2 **Product Operations Job Family**. Although contemplated in the classical Agile methodologies, for the DT, in light of Sections 1.3 and 2.4, and the previous chapter, these figures must have the ability to consider many of the aspects related to the management, continuous and evolutionary maintenance of the digital service portfolio offered by PAs. In particular, they must focus on issues of service change management, configuration elements, organizational change, vendor change, and related documentation, while simultaneously promoting the development of expertise in the use of change management tools and processes. Moreover, they must provide an accurate monitoring of services, in real time, to identify potential problems or areas for improvement that can then be examined. By way of example, a **Contract Responsible** and a **Service Responsible** are needed, possibly in a leadership position.

4.4 From Generic to Specific Professional Figures: The Scrum Case

We now consider the professional figures that have been defined in the previous section and show how they can be placed within an Agile methodology. As specified in Chapter II, we have chosen Scrum as the Agile methodology of reference. In particular, we identify two types of figures: Leaders and Team Members. The first are key figures whose main task and responsibility is to coordinate a group of professionals, i.e., a team, to the achievement of a specific goal. The second are figures with specific responsibilities and skills, that are part of a team. We provide a description of those teams as well.

Leaders

Those figures are schematically provided in Figures 4.1-4.2, suitably placed in the Job Family taxonomy, that we have described in Section 4.3, mainly mapped to the Product Creation & Support Category. The roots of those two trees are offsprings of the taxonomy root node, that has been omitted in order to ensure readability of the figures. We anticipate that some of those figures are standard for the Scrum methodology, but here they are placed within the taxonomy, granting homogeneity. Then, in addition to the classic Scrum figures, there are other professional figures that specialize Scrum for its adoption in the PA context.

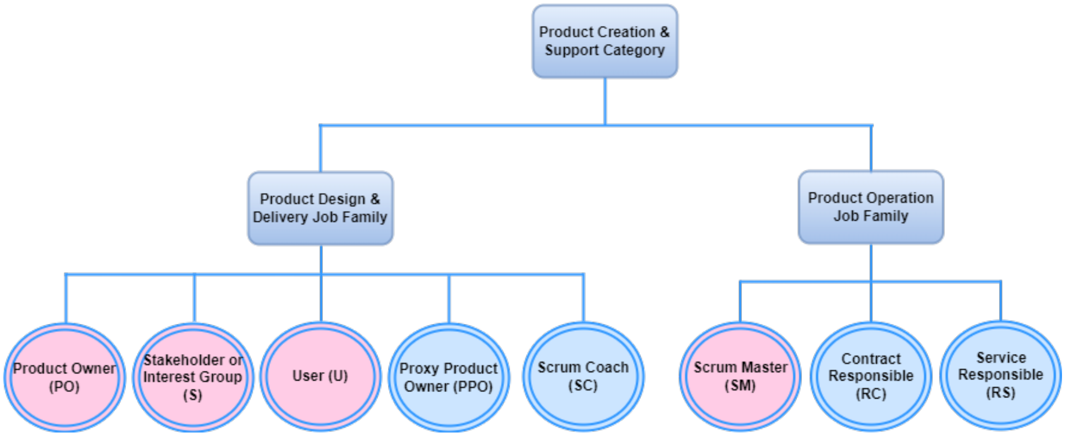

Product Creation & Support Category
Product Design & Delivery Job Family
Product Operation Job Family
Product Owner (PO)
Stakeholder or Interest Group (S)
User (U)
Proxy Product Owner (PPO)
Scrum Coach (SC)
Scrum Master (SM)
Contract Responsible (RC)
Service Responsible (RS)

Figure 4.1: **Scrum leader professional figures for the PA within the taxonomy of new professional figures: Product Creation & Support Category.** Starting at the root, the first two levels of the tree are a verbatim rendering of the taxonomy described in Section 4.3, while the leaf nodes represents the set of leader professional figures involved in the entire life cycle of the project that the PA intends to realize following the Scrum methodology. In this latter level, two types of professional figures are represented, i.e., those belonging to the standard Scrum methodology (see pink leaves) and those specialized in its adoption in the PA context (see light blue leaves).

- **Data Management & ICT Category**
  - **Data Job Family**
    - **Chief Data Officer** (CDO, for short). The specifics of this figure, also known as the *Data Manager*, in regard to the DT, have

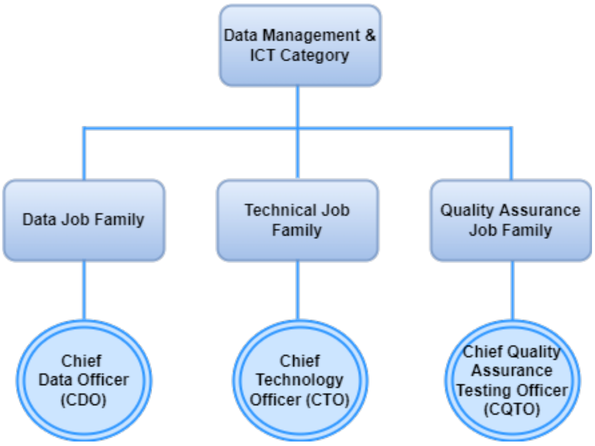

Data Management & ICT Category
Data Job Family
Technical Job Family
Quality Assurance Job Family
Chief Data Officer (CDO)
Chief Technology Officer (CTO)
Chief Quality Assurance Testing Officer (CQTO)

Figure 4.2: **Scrum leader professional figures for the PA within the taxonomy of new professional figures: Data Management & ICT Category**. The Figure legend is as in Figure 4.1.

been given in point (D.1) in Section 4.3. They must be coherently placed within the following generic description of this figure. S/he is responsible for overseeing and managing all data-related activities within an organisation to ensure that data is managed efficiently, securely and in compliance with regulations. S/he coordinates the collection and integration of data from various sources, ensuring that it is accurate, complete and compliant with privacy regulations and relevant laws, while ensuring the consistency and efficiency of the processes of the organisation s/he represents. To this end, s/he defines and implements appropriate data storage and protection policies, through policies such as encryption, restricted access and constant monitoring. S/he implements regular backup strategies to ensure continuous data availability and plans recovery procedures in case of emergencies. S/he monitors and improves data quality through cleansing, normalisation and standardisation processes, e.g., Open Data Standards (see, e.g., [285]). S/he collaborates with Data Analysts to extract meaningful information and support data-driven decision-making. In terms of training and communication, s/he provides support and training to staff on the importance of data management and communicates relevant procedures and policies. This is a new professional figure for its use in Agile IT project management. The interested reader can find additional details about its role and responsibilities in [104].

– **Technical Job Family**

  - **Chief Technology Officer** (CTO, for short). The specifics of this figure, also known as the *Technology Manager*, in regard to the DT have been given in point (D.2) in Section 4.3. In general, s/he is responsible for playing a key role in shaping the technological direction of an organisation, and has various responsibilities related to technology management and implementation, ensuring that technologies are aligned with the objectives of the organisation and contribute to its overall success. S/he is involved in defining the technology strategy of the organisation, helping to establish long-term goals in line with corporate objectives. S/he monitors technology trends and evaluates new opportunities and innovative solutions to improve business efficiency and competitiveness. In terms of security, s/he is responsible for implementing appropriate measures to protect corporate data from external and internal threats. In terms of technology resource management, s/he ensures the efficient allocation of technology resources, including budget, personnel and infrastructure, to ensure the achievement of technology objectives. S/he collaborates with other stakeholders and teams, such as marketing, sales and finance, to ensure alignment of the technology strategy with the overall needs of the organisation. S/he monitors the performance of technology systems, identifies possible areas for improvement and implements solutions to optimise the processes of the organization. In addition, s/he interacts with external suppliers for the selection and integration of external technology solutions. This is a new professional figure, for which we found no evidence in the Literature about its use in Agile IT project management.

- **Quality, Assurance and Testing Job Family**

  - **Chief Quality Assurance Testing Officer**
    (CQTO, for short). The specifics of this figure, also known as *Quality Assurance Manager*, in regard to the DT have been given in point (D.3) in Section 4.3. In general, s/he is responsible for a role that may vary depending on the organization and the specific sector, but in general, s/he refers to a manager responsible for managing and supervising the entire quality assurance and testing process within an organization. S/he defines the necessary policies and procedures to ensure the quality of the product/service offered by the company. S/he supervises personnel involved in quality assurance and testing, ensuring that they are well-trained and follow best practices. To this end, s/he collaborates with business leaders to develop a quality assurance and testing strategy in line with overall business object-

ives. S/he identifies and implements testing tools, technologies and methodologies, such as quality metrics monitoring systems, effective in improving testing efficiency and coverage, working to prevent problems or resolve them promptly, to mitigate risks associated with product/service quality. S/he collaborates with other departments, such as software development, to ensure effective communication and mutual understanding of needs and objectives. Finally, s/he provides regular reports to organizational leadership on the effectiveness of the quality assurance process and testing, with suggestions for continuous improvement. This is a new professional figure, for which we found no evidence in the Literature about its use in Agile IT project management.

- **Product Creation & Support Category**
  - **Product Design & Delivery Job Family**
    - **Product Owner** (PO, for short). It is to be noted that this professional figure can play a role also in the Product Operations Job Families. The PO helps to build the product from scratch, guiding and coordinating the work of the standard Scrum Development Teams (presented and discussed in Section 3.3), to optimise the value of the product by ensuring that a tangible increment is produced at each Sprint cycle. The contribution of the PO can be significant and certainly required at the beginning of the project and during the entire Product Backlog lifecycle. Although the PO is one of the classic Scrum figures, its role and consequent responsibilities change significantly, given the substantial regulatory scaffolding to which the PA must refer. The mentioned changes with respect to standard Scrum are well presented in the realm of iAgile and Scrum@IMI, where the PO is no longer a single professional figure. Indeed, for Scrum@IMI, the PO is the entire City Council of Barcelona, possibly via a delegate [283]. As for iAgile, the PO is now a board of people, including stakeholders and domain experts [249, 250] . It is worth of mention that an earlier instance of such a model has been used in [286].
    - **Stakeholder or Interest Group** (S, for short). We note that the role of the Stakeholders turns out to be the same as compared to what we have referred to in the traditional Scrum methodology, i.e., managers and customers. For this reason, the reader is referred to Section 3.3 for a description of its main functions.
    - **User, i.e, Citizen** (U, for short). This figure, despite its centrality in the proposal and development of the user-centered services

offered by PA, is not mapped among the categories of Job Families discussed in Section 4.3, since here the focus is on professional figures directly involved in the software development of public digital services provided by PA.
This professional figure plays a major role, with particular regard to the proposal and development phases of the user-centered services offered by PA. An example of increased user centrality is found in the Scrum@IMI framework, with a focus on testing activities regarding the degree of user acceptance by the PO and PPOs (see below) that are associated with the collection of feedback and change requests from them.

∗ **Proxy Product Owner** (PPO, for short). It is to be noted that this professional figure can play a role also in the Product Operations Job Families. The PO often has a strenuous agenda and numerous stakeholders to interact with. As a result, its availability to carry out all necessary activities is limited. Consequently, an additional profile known as a PPO primarily provides support to the PO to ensure that it can focus on defining and specifying business requirements during the Product Backlog refinement sessions, the review, the retrospective, and the validation of the product delivered by the development team at the end of each Sprint. Furthermore, the PPO is highly valuable in tasks like requirements analysis and the subsequent writing and convalidation of user stories (see Section 3.3), as well as staying connected with the Development Team to follow up on and assist with any technical and business issues that may emerge. PPOs can, in collaboration with the SM, defined below, actively participate in developing and promoting team-building activities. An example of this professional figure is found in the Scrum@IMI framework [254] . In fact, in such a context, the PPO, e.g., collaborates with the PO to ensure the creation of the agreed testing strategy with the entire Scrum team.

∗ **Scrum Coach** (SC, for short). The specifics of this figure, also known as *Agile Coach*, in regard to the DT have been given in point (P.1) in Section 4.3. In general, in accordance with financial regulations and within the terms of ongoing contracts, s/he manages and plans the necessary resources within the Scrum project. Under delegation from the contractor, s/he is responsible for creating and updating the Competence Matrix. For the convenience of the reader, we recall from the Literature that a Competence Matrix is a table, where some needed technical competences are represented by rows and developers by columns. A “marked” entry states that a developer has the corresponding competence, with some level of expertise. Based on such a matrix, the SC can activate the Pair Programming process, which consists of creating pairs of developers that can best work together based on their competence similarities. For further details, the reader is referred to [264, 267]. Upon the occurrence of appropriate specific circumstances, s/he is also responsible for the coordination of the pool of human resources needed for the project. In addition, the SC has the authority to assign the required professional figures to the various tasks and negotiate their timelines. It is useful to note that, on behalf of the SC, SMs may be directly responsible for the recruitment process of the professional figures involved in the project development phase. An example of this professional figure is found in the iAgile methodology. Indeed, in such a context, the SC, must take into account a unique mix of consultants and armed forces personnel necessary to address the complexity of user requirements and the operational environment in which the armed forces are asked to operate today.

- **Product Operation Job Family**
  - **Scrum Master** (SM, for short). We note that the role of the SM turns out to be the same compared to that of the standard Scrum methodology. For this reason, the reader is referred to Section 3.3 for a description of its main functions.
  - **Contract Responsible** (RC, for short). The specifics of this figure in regard to the DT have been given in point (P.2) in Section 4.3. In general, the main responsibility of the RC is to keep the contract and contractor relationships under control. S/he is largely responsible for compliance with the terms of the contract, including all billing requirements, administration, and management of changes in requirements, functional and non-functional. Her/His main responsibilities include: Monitoring deliveries and formalizing the substitution of some deliverables with others, if needed; Economic analysis of the Sprint Backlog (value Vs. cost Vs. effort); Execution of contract economic monitoring meetings. An example of this professional figure can be found in the Scrum@IMI framework. In such a context, it is afferent to IMI, and the main objective of its role is to carry out the monitoring of the contract and contractual relationship model with the supplier. To this end, s/he represents the highest responsibility for the contract and the fulfilment of the conditions described therein.

- **Service Responsible** (RS, for short). The specifics of this figure in regard to the DT have been given in point (P.2) in Section 4.3. S/he is essential during the transition from product to service and during the implementation of new versions because after the first increment of the product is launched into production, s/he is "responsible" for ensuring its optimal performance. S/he collaborates in defining the criteria and requirements of the service during the build phases. Anticipating that the envisioned proposal of Scrum for the PA consists of an equipe of teams, s/he supports the Scrum equipe in the planning and execution of the transition from product increment to service, providing the necessary information for the subsequent implementation, such as service level agreements (SLAs), optimal timing for maintenance execution, user volume estimates, resource sizing for the production environment, requirements for the Basic Information Model and the General Data Protection Regulation [17]. S/he ensures that the project also meets non-functional criteria before implementation. The RS is responsible for approving implementations with the PO to ensure that the quality of the delivered product is adequate and that the service functions as efficiently as possible. S/he collaborates with the SM, whose job is to resolve any impediments encountered. From the beginning of the project, the RC and RS work closely together to monitor the various product increments developed by the development team, which we discuss in the next section. Finally, in the performance of her/his functions, s/he is completely free to attend as many events and meetings as s/he sees fit. An example of this professional figure can be found in the Scrum@IMI framework. In that framework, her/his role, together with the PO, is primarily to ensure the optimal functioning of the implemented service, at each version, while concurrently ensuring that the project meets the requirements of all IMI guidelines and non-functional requirements, which have been included in the Product Backlog.

Teams and Their Members

Those Teams, and their members, are schematically provided in Figures 4.3-4.4, where members composing each team are suitably placed in the Job Family taxonomy that we have described in Section 4.3. The roots of those two trees are offsprings of the taxonomy root node, that has been omitted in order to ensure readability of the figures. We anticipate that some of those teams are standard for the Scrum methodology, but here they are placed within the taxonomy, granting homogeneity. In our professional figures taxonomy, its members are mapped to the

Data Management & ICT Category. Then, in addition to such a classic team, we propose additional professional teams for the specialization of Scrum for its adoption in the PA context.

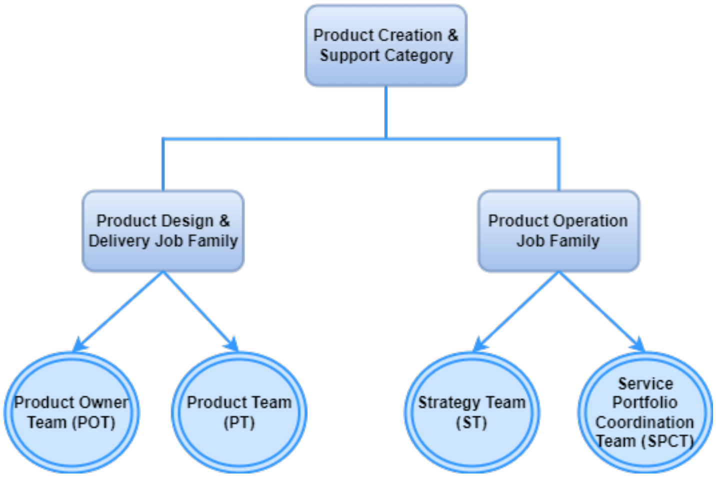

Product Creation & Support Category
Product Design & Delivery Job Family
Product Operation Job Family
Product Owner Team (POT)
Product Team (PT)
Strategy Team (ST)
Service Portfolio Coordination Team (SPCT)

Figure 4.3: **Taxonomy of Teams and their Members Professional Figures, within the Scrum methodology, for their adoption in the context of the PA: Product Creation & Support Category**. The Figure legend is as in Figure 4.1.

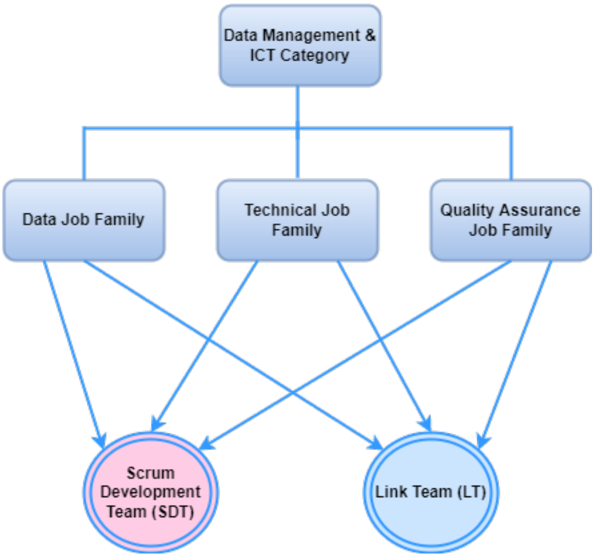

Data Management & ICT Category
Data Job Family
Technical Job Family
Quality Assurance Job Family
Scrum Development Team (SDT)
Link Team (LT)

Figure 4.4: **Taxonomy of Teams and their Members Professional Figures, within the Scrum methodology, for their adoption in the context of the PA: Data Management & ICT Category**. The Figure legend is as in Figure 4.1.

- **Data Management & ICT Category**
  - **Scrum Development Team** (SDT, for short). The task of this team, and therefore of its members, is to provide the software development of the services offered by PA. We note that the functions of the SDT members are similar to those of the classic Scrum methodology, except for a greater ability to address many aspects of Data Management and Governance, with the related QAT techniques.
  - **Link Team** (LT, for short). It is composed of domain experts from all technical areas involved in project development, e.g., Data, Architecture, Security, Telecommunications, Operations and Systems, Quality Assurance, Testing, and Service Management. The task of this team, and therefore of its members, is to support the SDT, to resolve non-functional requirements of the project, which are included in the Product Backlog. The CDO, CTO and CQTO Leader professional figures, defined in Section 4.4, can be part of this team.

- **Product Creation & Support Category**
  - **Product Design & Delivery Job Family**
    - **Product Owner Team** (POT, for short). The task of this team, and therefore of its members, is to provide the vision of the final product to the entire set of Scrum teams allocated to the project. It is a high level abstraction of the PO, i.e., the professional figure representing the developer team for the implementation of the digital service, providing an overview of the product to be developed. Stakeholders representing all levels of governance are assigned to the POT, e.g., the PO, the PPOs, the general manager of the contracting authority or her/his delegates. Placing the stakeholders within a team, making them well aware of the methodology and enabling them to interface directly with the developers, is a way of addressing the goal of maximising customer satisfaction. The availability of stakeholders to fully support the SDT and the SM has a significant impact on the development itself, both as a vehicle for requirements and as a ground for building a sense of ownership of the end product that, in turn, is believed to facilitate operational and user acceptance [287]. For these reasons, the SDT members must possess a broad knowledge of the system being developed and a clear vision of the expected results.
    - **Product Team** (PT, for short). The task of this team, and therefore of its members, is to ensure that the project produces the product

described by the relevant user stories (see Section 3.3), according to the required quality standards and within the specified time and cost constraints, and that everyone involved knows what is expected and helps to keep costs, time and risks under control. A PT is defined as a temporary organization created to develop software related to a single product. Several SDTs may be incorporated into a single PT. If the team developing a single product is unique, the PT can be consolidated through the SDT, without the need to have additional professional figures, often from the POT, to coordinate the different development teams.

- **Product Operation Job Family**
    - **Service Portfolio Coordination Team** (SPCT, for short). The task of this team, and therefore of its members, is to solve problems arising from the management and integration of services within the services portfolio offered by PA by keeping all members of the organization informed of the issues encountered, identifying high level deficiencies in the Agile process and initiating appropriate actions to resolve them.
    - **Strategy Team** (ST, for short). The task of this team, and therefore of its members, is to encode project parts into epics and user stories (see Section 3.3), and oversee the processes that are employed to exercise high level control over the entire development process. The ST supports the POT (a member is part of the POT as well) in managing the Product Backlog and the PT/SPCT to deliver product iterations to users while pursuing overall system consistency.

4.5 A General Paradigm for Scrum in the DT: Scrum@PA

In this section, as mentioned in the introduction of this chapter and with the use of what has been presented in Section 4.4, we propose a paradigmatic Scrum Model for the PA. We denote this paradigm as Scrum@PA.

From what was discussed in the previous chapter, and for the purposes of this one, it is convenient to schematize the Scrum methodology as follows. With reference to Figure 4.5, it consists of “an idea” (left), a series of Scrum Sprints representing the product development phase (center), and the delivery of the product to the end users (right), including increments after each Sprint.

In our paradigm, the end parts are conceptually the same, although they can be further specified as follows. The “idea” is the result of contributions coming from

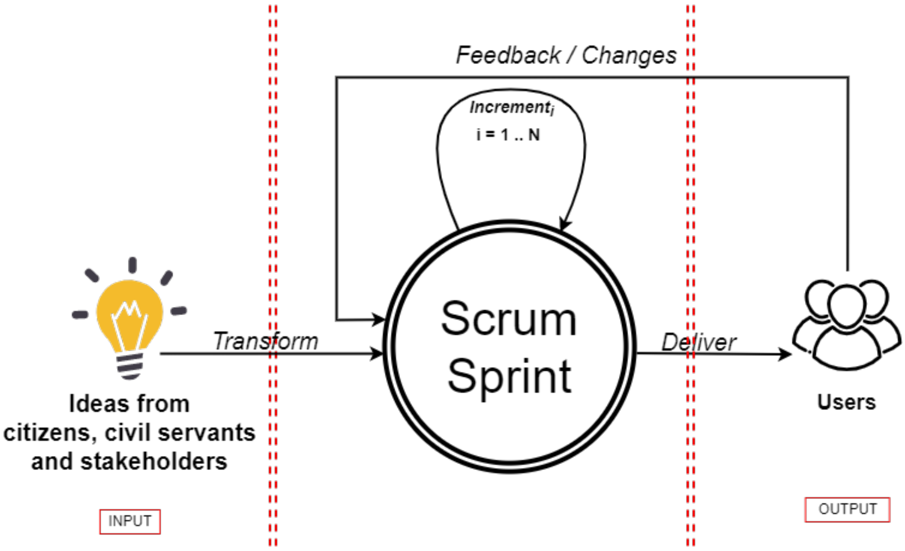


Figure 4.5: **A Generic Scheme of the Software Development Process via the Scrum Methodology**. This figure is divided into three parts. The left part consists of an idea of the public service that is intended to be realised. The middle part consists of a series of Scrum Sprints representing the product development phase, while the right part represents the delivery of the product to the end users, including increments after each Sprint.

citizens, civil servants, and stakeholders, that addresses a specific set of needs and that can be translated into the offering of a public digital service. The "product" will evolve through a set of software releases implementing and supporting the intended public service. It is worth mentioning that those releases must be approved before being made available to the public, i.e., this is the transition to the service phase. As for the central part of that schematic representation, which we refer to as product development, in a standard Scrum methodology, it is composed of four components: Requirements, Design, Software Development, and Testing (see one Sprint in Figure 3.1). Our paradigm, i.e., Scrum@PA, follows the same approach, although the product development phase is more complex with respect to the standard one. Namely, we have four components, which may include standard components within it: Service Design, Technology Management, Software Development, Product Compliance and Verification. Schematically, the process is represented in Figure 4.6, including interactions between each pair of components. The incoming edge, labeled transform in Figure 4.5 has been omitted to enhance readability of Figure 4.6. However, it enters the Service Design component. For the same reasons, the incoming edge, labeled feedback/changes, has also been omitted. Such an edge also enters the Service Design component.

Scrum@PA is presented as follows. Section 4.5 describes the activities carried out in each of the components mentioned above and indicates which Leader professional figures and Teams are involved in them. Section 4.5 describes the interactions among the various components, i.e., how the Leader professional figure and Teams in one component interact with the other corresponding elements in another component. Finally, based on what is presented in the previous two sub-sections, Section 4.5 provides a synopsis of the main responsibilities of Leaders and Teams with respect to some fundamental Scrum Artifacts), in terms of a RACI matrix. It is useful to recall that, in general, a RACI matrix [288] is a concise way of indicating who is responsible, and to what degree, of carrying out a predetermined set of activities considered essential for the successful realization of a project. It is to be noted that, for the PA, the recommended duration of Scrum Sprint varies from two-week to a month, depending on the complexity of the project at hand (see [289] and references therein, in particular regarding Agile Project Management and Software Development.

In terms of details, our description is high level, given that Scrum@PA is a general paradigm. However, it is comparable to the level of detail given in the presentation of Scrum@IMI on GitHub [283] and that a more detailed description, as the one given for Scrum@IMI in [254] , would considerably expand the description of Scrum@PA, making the key points of our contribution difficult to enucleate and then specialize in specific contexts.

Components of the Software Development Phase

We now describe the activities characterizing each of the components mentioned earlier, together with the Leader professional figures and the Teams that carry them out. We anticipate that professional figures may be well involved in performing activities, while not being part of a team.

- **Software Development**
  - **Activities**
    1. Facilitate the team development process, removing any obstacles; Ensure that goals are understood and achieved; Promote collaboration, transparency and self-organisation of the development team. Initial identification, within this component, of teams and related professional figures, needed for the successful development of the project, with possible updates as the project realization progresses.
    2. Implementation of the technology stack on which the system to be developed would be based. Testing of development standards, data security and privacy, code quality, and architecture. Completition of project deliverables, iteratively, incrementally and within each

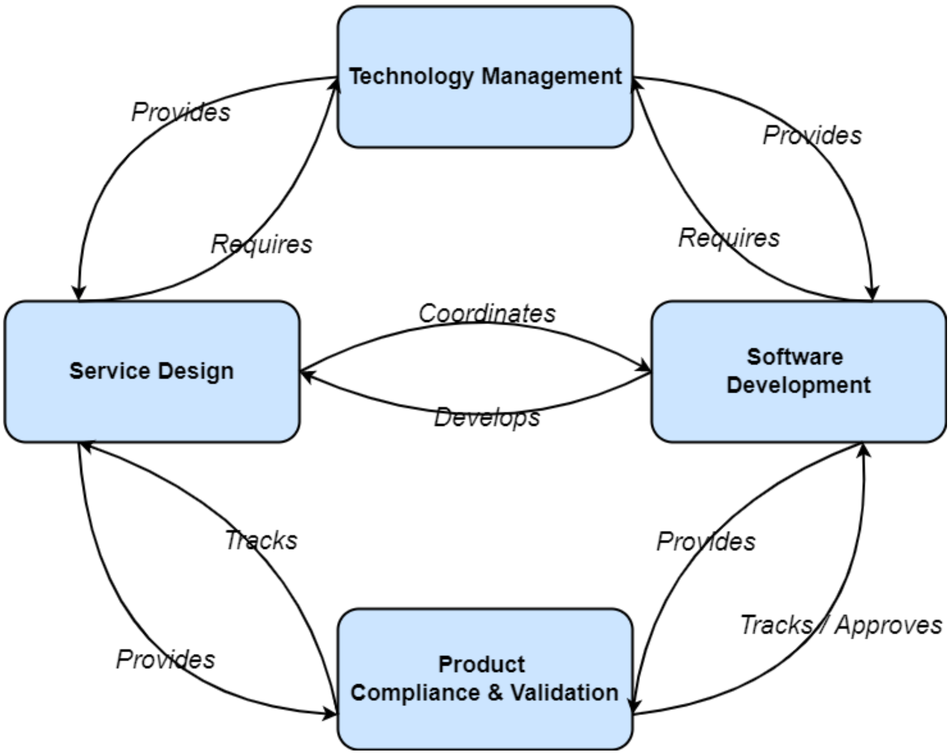


Figure 4.6: **A Paradigmatic Scrum Model for the PA: Scrum Sprint**. The activities carried out within each component, together with the Leader professional figures and Teams involved in them, are described in Section 4.5, while the interactions between each two components, here encoded by edges, are described in Section 4.5.

Sprint, giving priority according to the relevance of each deliverable. Creation of documents that accompanies the product, such as architectural map, functionality description, and security controls.

3. Creation and update of the Competence Matrix of developers. Activation of the Pair Programming Process and assignment of developers to specific project periods, as required.

– **Leaders and Teams: Roles And Intra-component Interactions Among Them**

Recalling from Section 4.4 that the PT may be composed of one or more SDT, based on the complexity of the development process, at this stage of abstraction, we use one PT and two Leader professional figures: SM, and SC. The interactions within these Teams and Leader professional figures are as follows, with Figure 4.7 giving a schematic representation of this component.

∗ PT. It collaborates with the SM, throughout the product development cycle, so that the Scrum methodology adoption is ensured. On the other hand, it provides the necessary information to the SC to accomplish its functions.

∗ SM. This Leader figure, in addition to the PT collaboration mentioned earlier, is also responsible, along with the PO, for promoting activities aiming at the identification of the main characteristics needed by the development team. S/he also informs the SC of the competencies present in the team.

∗ SC. Based on information coming from the team and the SM, the SC keeps track of the different technical competencies of the various software developers to perform and refine the Pair Programming Process. The SC can delegate the recruitment process to the SM, in part or fully.

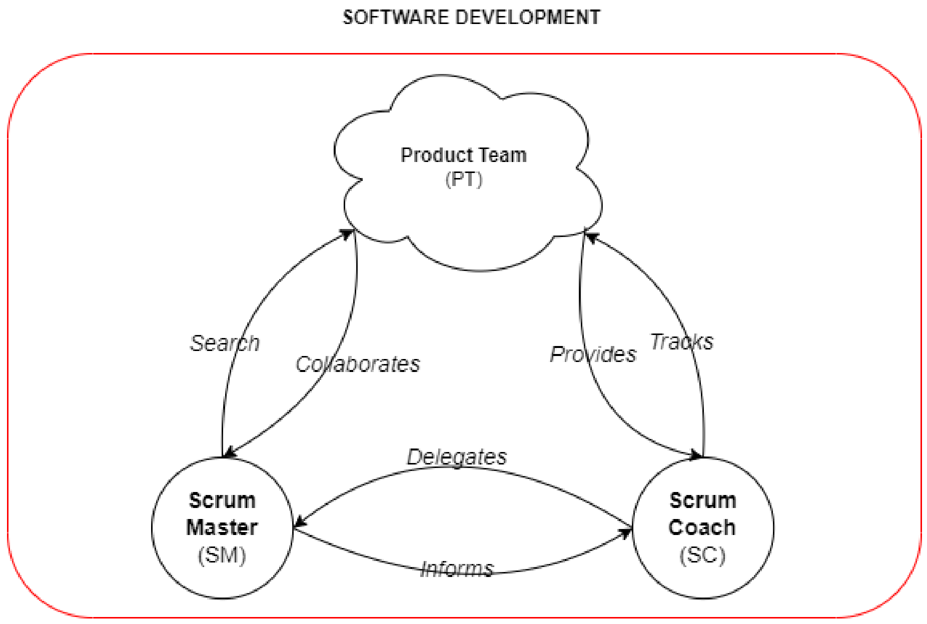


Figure 4.7: **Scrum@PA Software Development Component**. A node symbolised by a cloud represents a team, while a node symbolised by a circle represents a Leader professional figure. This component consists of a single Team, i.e., PT, and two Leaders, i.e., SM and SC. The description of the interactions among the team and the Leaders (edges) follows the Main Text.

- **Service Design**

- **Activities**
    1. Translation of the functional requirements of the Project into user stories and epics to be included in the Product Backlog. Such an activity can be divided into two, as follows.
        (a) Initial. Once that a service development project is approved, the production of a number of documents useful for project specification, such as technical plan, quality standards, and risk assessment, must be produced. In addition, an initial Product Backlog must be produced, ensuring that the requirements and proposals of the supplier are met.
        (b) Ongoing. During the progress of the project realization, its coherence with the functional requirements must be monitored. Moreover, since we are within an Agile methodology, the feedback and possible changes in need of the users, resulting in changes in requirements, must be accounted here.
    2. Management and integration of the service being developed within the ecosystem of services offered by the PA. Organization of meetings regarding the possible problems encountered in the Agile methodology and identification of an appropriate strategy for resolving them.
    3. Stakeholder coordination to ensure a global vision of the product to be implemented. In particular, keep stakeholders informed in regard to the software development process, in order to achieve a greater sense of ownership of the project and improve end-user experience and satisfaction.
- **Leaders and Teams: Roles And Intra-component Interactions Among Them**

    At this stage of abstraction, we use three teams: SPCT, ST, and POT. The interactions within these teams are as follows, with Figure 4.8 giving a schematic representation of this component.

    ∗ SPCT. This team supports the ST in the aspects associated with defining non-functional requirements that are related to the integration of the services offered by the PA. Those requirements may give raise user stories that are included in the Product Backlog. SPCT also supports the stakeholders afferent to the POT, which represents the different levels of governance, by informing them of any problems encountered, to ensure the functioning of the services offered and to facilitate their integration.

    ∗ ST. This team, based on the vision of the Product Owner (PO), Proxy Product Owners (PPOs), and stakeholders in general, supports the POT in decomposing the project into smaller parts, that are then encoded in epics and user stories. Those are placed in the Product Backlog. In particular, this team is responsible for the production of the initial Product Backlog.

∗ POT. This team, prioritizes the items within each version of the Product Backlog, starting with the initial one and proceeding with updated priorities at the end of each Sprint. Within this component, POT is supported by the ST and the SPCT in regard to the Product Backlog.

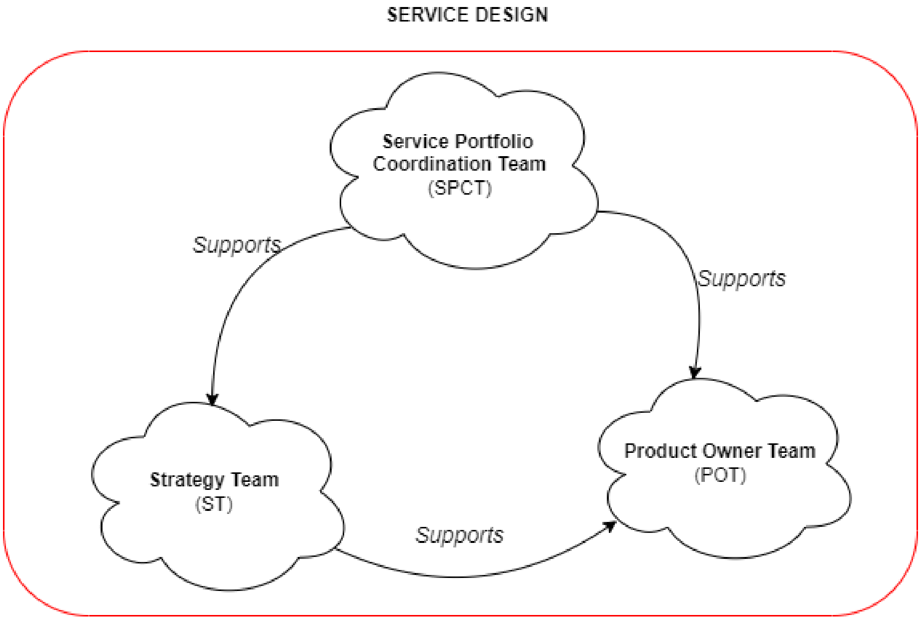


Figure 4.8: **Scrum@PA Service Design Component**. Following the same notation as in Figure 4.7, this component consists of three teams, i.e., SPCT, ST, and POT. The description of the interactions among teams (edges) follows the Main Text.

- **Technology Management**
  - **Activities**
    1. Provide support to Software Development component for the resolution of non-functional Project requirements, mainly in terms of Data, Architecture, Security, Networks, Systems, Applications, and Quality Assurance Testing.

- **Leaders and Teams: Roles And Intra-component Interactions Among Them**

  At this stage of abstraction, we use only one Team, i.e., LT, which has been described in Section 4.4. Figure 4.9 gives a schematic representation of this component.

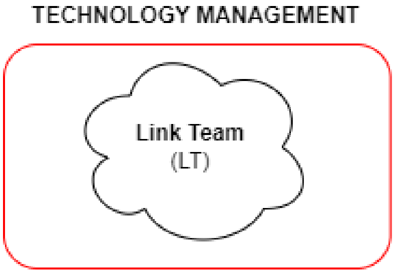


Figure 4.9: **Scrum@PA Technology Management Component**. Following the same notation as in Figure 4.7, this component consists of a single Team, i.e., LT.

- **Product Compliance and Verification**

  - **Activities**

    1. Drafting of contracts for the provision of artwork and services for PA related to the project.
    2. Checklist of Product Backlog items oriented towards controlling the compliance of the project with the contractual terms , mainly administrative, economic and managerial.
    3. Checklist of agreed service levels (SLAs), e.g., Basic Information Model, and the GDPR.

  - **Leaders and Teams: Roles And Intra-component Interactions Among Them**

    At this stage of abstraction, we use two Leader professional figures: RC, and RS. The interactions within these Leaders are as follows, with Figure 4.10 giving a schematic representation of this component.

    - RC. With the support of the RS, the RC ensures compliance with all contractual aspects and monitoring of all product increments.
    - RS. With the support of the RC, the RS ensures that each time a new product increment is released, it satisfies: service acceptance criteria (SLAs), data privacy requirements, and the needs of the users/citizens using the service.

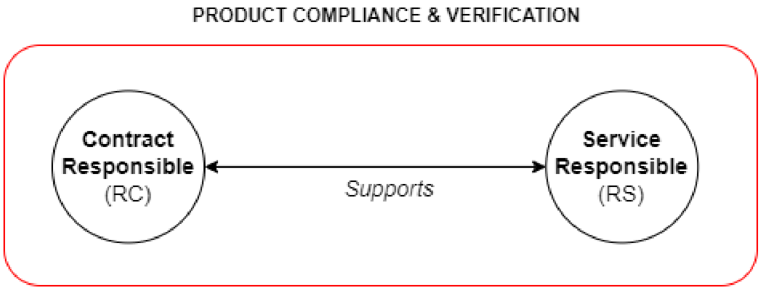


Figure 4.10: **Scrum@PA Product Compliance and Validation Component**. Following the same notation as in Figure 4.7, this component consists of two Leaders, i.e., RC, and RS. The description of the interactions among Leaders (edges) follows the Main Text.

Interactions Among Components of the Software Development Phase

We now describe the external interactions among the components of the Scrum model listed above.

- **Interactions with Software Development**. PT, afferent to this component, ensures the implementation of the software product in the various Sprints of the development cycle, providing the software increments with additional features to the RS and RC, afferent to the Product Compliance & Validation component. They are respectively responsible for their approval before making them available to citizens and for compliance with contractual terms and conditions.

- **Interactions with Service Design**. POT, afferent to this component, coordinates the management and prioritization of items in the Product Backlog so that the PT, afferent to the Software Development component, can ensure their implementation in the various Sprints concerning the software product development cycle. Interaction with the Technology Management component in terms of resources requests for the realization of the project.

- **Interactions with Technology Management**. The LT, afferent to this component, mainly supports PT, afferent to the Software Development component, for resolving non-functional requirements. In addition, it also supports various stakeholders, afferent to the Service Design component, for resolving functional requirements.

- **Interactions with Product Compliance & Verification**. The RC ensures that the software developed by the PT meets the functional requirements provided by the ST, in accordance with the priorities given by the POT. The latter two are part of the Service Design component. The RS checks the compliance of the various software increments distributed to the end users, e.g., citizens, with service acceptance (SLAs), privacy and data security criteria. That is, the RS approves the software releases.

Scrum@PA RACI Responsibilities Matrix

We now provide a synopsis of the Scrum Artifacts that we consider as a result of the activities described earlier, together with the level of responsibility of Leaders and Teams in achieving their realization. As stated in Section 3.3 , the artifacts are quite standard for Scrum: Contract, Product Backlog, Sprint Backlog, Definition of Done, Increment, and Compliance. They are listed in the top row of the RACI matrix depicted in Table 4.1. As for the Leader professional figures and Teams, they are the ones indicated in the previous two sub-sections, and reported as the first column of the mentioned table. The intersection of rows and columns is marked with the level of responsibility that each Leader professional figure and Team has with regard to an Artifact, according to a standard scale reported in the caption of the mentioned table.

Table 4.1: **Scrum@PA Responsibility Matrix**. This table defines the responsibilities of leader professional figures, and teams, within Scrum@PA, providing a comprehensive overview of who is responsible (R), accountable (A), consulted (C), and informed (I), in each of the artifacts present within this Agile methodology. Empty cells are assumed as informed (see above).

Teams and Leaders	**Scrum Artifacts**					
	Contract	**Product Backlog**	**Sprint Backlog**	**Definition Of Done**	**Increment**	**Compliance**
LT		I	C			
POT	A	R		R	R	C
PT		I	R	A	C	C
SPCT	C	C			C	
ST	I	A				
RS	A				A	R
RC	R					A

4.6 The Scrum@IMI Framework as a Special Case of Scrum@PA

We now show how to map the Scrum@IMI methodology within Scrum@PA, concentrating on the Scrum Sprint. It is worth pointing out, leaving the details to the interested reader, that the Leader professional figures and Teams involved in Scrum@IMI are part of our taxonomy (see Sections 4.4-4.5 , and [254, 283]).

Technically, the Scrum Sprint in Figure 4.5 , for the Scrum@IMI methodology consists of four phases: On-Boarding, Sprint 0, Sprint i (iterated), and Service Transition, schematically depicted in Figure 4.11. Moreover, the incoming edge, labeled transform in Figure 4.5 has been omitted to enhance readability of Figure 4.11. However, it enters the On-Boarding phase. For the same reasons, the incoming edge, labeled feedback/changes, has also been omitted. Such an edge enters the Sprint i phase. Although Scrum@IMI precedes the paradigm proposed here, it is quite relevant to outline how it fits this latter. In what follows, we provide the corresponding details by describing Scrum@IMI and by showing which parts of the Scrum@PA paradigm it matches. In particular, we describe the Scrum@IMI phases and indicate how they can be seen as instances of the Scrum@PA components. It is also worth pointing out that the specificity of Scrum@IMI with respect to Scrum@PA comes out in terms of what the City of Barcelona established regarding the governance of the design and realization of a specific project, for instance, public procurement [147], regulations [151], privacy, service and data ownership [20, 98].

The terminology regarding teams and figures of Scrum@IMI follows the one of Scrum@PA. Moreover, for ease of readability, when the SM, POT, PT and LT act together, they are referenced as “Meta Team”, which it is indicated by a red cloud in the relevant figures (see Figure 4.12 for an example).

On Boarding

The activities and teams involved in the On-Boarding phase are described synoptically in Figure 4.12, which has been divided into two panels to make it more readable. We point out that this phase produces no output for the service transition phase. In terms of Scrum@PA, this phase involves all four of its components. Namely, two Leader professional figures, i.e., SM and RC, and three Teams, i.e., POT, LT, and PT. The activities that are carried out can be phrased in terms of our paradigm as follows, with the dot numbering specified in the legend of Figure 4.12. Service Design: Activity (1.a), contributing to the Scrum@IMI activities described in dots (1.a), (1.b) and (2.b) in Figure 4.12. Software Development: Activity (1), contributing to the Scrum@IMI activities described in dots (1.a), (1.b), and (2.b) in Figure 4.12. Technology Management: Activity (1), corresponding in part to the Scrum@IMI activities described in dots (1.a), (1.b), and in dot (2.b) in Figure 4.12. Product Compliance & Verification: Activity (1) and (2), corresponding to dots (3.a) and (1.b) in Figure 4.12.

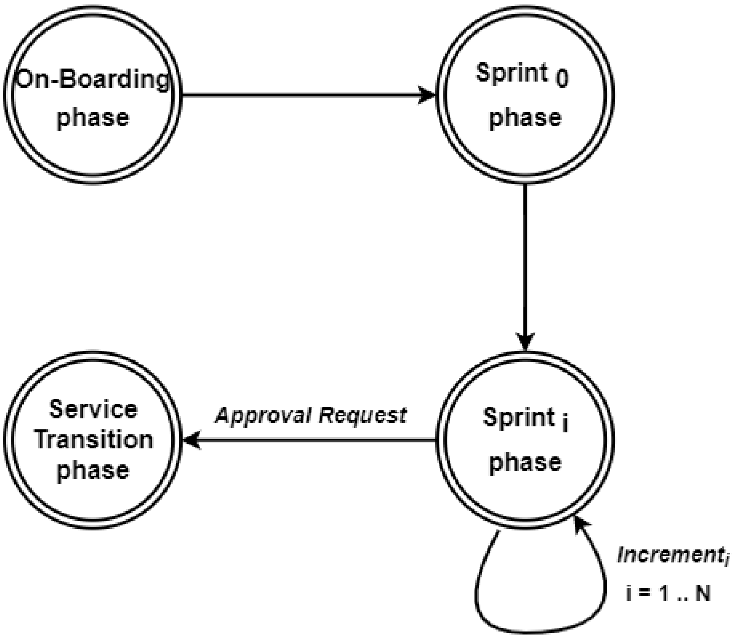


Figure 4.11: **Scrum@IMI Methodology : Scrum Sprint**. Starting from the On-Boarding phase, the arrows provide a temporal succession of the four phases. Each phase is detailed in the Main Text, together with the edges that enter such a Scrum Sprint.

Sprint Zero

The activities and teams involved in the Sprint 0 phase are described synoptically in Figure 4.13, which has been divided into four panels to make it more readable. This phase does not produce any artifacts for the Service Transition phase, but it does produce a small application, referred to as Hello@IMI, particularly useful to prepare the PT for the complete technology ecosystem on which the system to be developed would be based. This latter offers the opportunity to test, for instance, development standards, data and privacy security, code quality, and architecture.

In terms of Scrum@PA, this phase involves all four of its components. Namely, two Leader professional figures, i.e., SM and RS, and three Teams, i.e., POT, LT, and PT. The activities that are carried out can be phrased in terms of our paradigm as follows, with the dot numbering specified in the legend of Figure 4.12. Service Design: Activity (1.a), contributing to the Scrum@IMI activities described in dots (1.a), (2.a), (2.b), (3.a), (3.b), and (4.a) in Figure 4.13. Software Development: Activities (1) and (2), contributing to the Scrum@IMI activities described in dots (1.b), (2.a), (2.b), (3.a), (3.b), and (4.b) in Figure 4.13. Technology Management: Activity (1), corresponding in part to the Scrum@IMI activities described in dots (1.c), and (3.c) in Figure 4.13. Product Compliance & Verification: Activity (3),

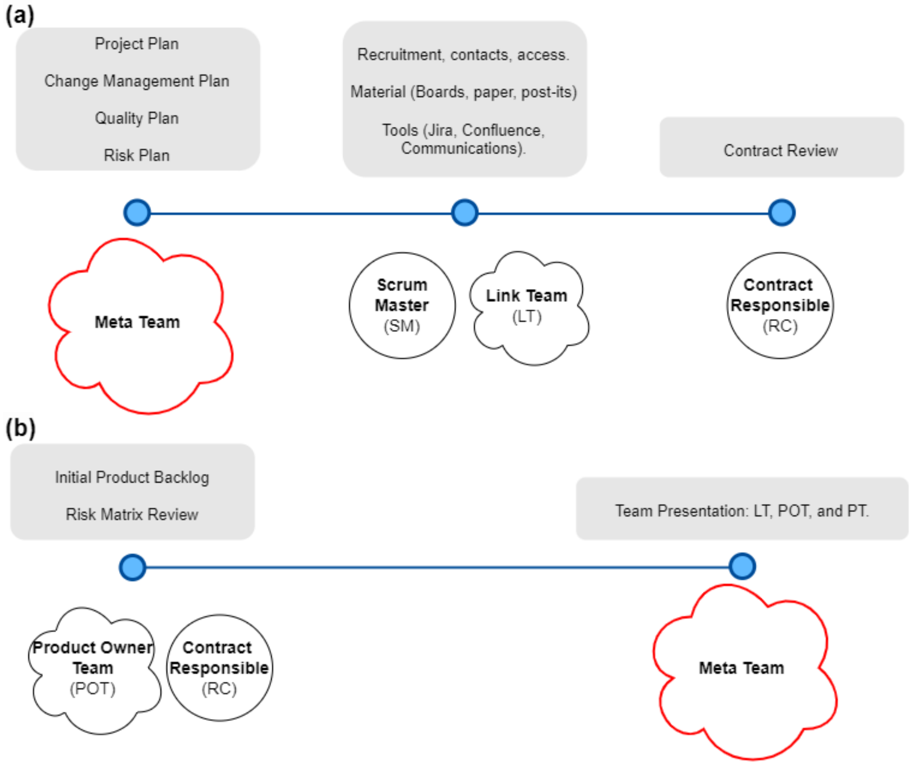

(a)
Project Plan
Change Management Plan
Quality Plan
Risk Plan
Recruitment, contacts, access.
Material (Boards, paper, post-its)
Tools (Jira, Confluence, Communications).
Contract Review
Meta Team
Scrum Master (SM)
Link Team (LT)
Contract Responsible (RC)
(b)
Initial Product Backlog
Risk Matrix Review
Team Presentation: LT, POT, and PT.
Product Owner Team (POT)
Contract Responsible (RC)
Meta Team

Figure 4.12: **The On-Boarding Phase of the Scrum@IMI Methodology**. The Teams and the Leader professional figures are indicated as in Figure 4.7. For each line, dots are considered as numbered from left to right, but such an order does not imply time of execution priority. For each dot, the activities are above it, while who is involved in carrying them out is below it.

corresponding to dots (1.b), and (1.c) in Figure 4.13.

Sprint i

The activities and teams involved in the Sprint i phase are described synoptically in Figure 4.14, which has been divided into three panels to make it more readable. This phase produces the product increments. It has to be pointed out that, for some Sprints, it is decided to test the corresponding product increment with collaboration from the end users. This type of Sprint is referred to as "Release Sprint" and its product is provided to the Service Transition phase (see next sub-section).

In terms of Scrum@PA, this phase involves all four of its components. Namely, three Leader professional figures, i.e., SM, RC, and RS, and three Teams, i.e., POT, LT, and PT. The activities that are carried out can be phrased in terms of our paradigm as follows, with the dot numbering specified in the legend of Figure 4.12.

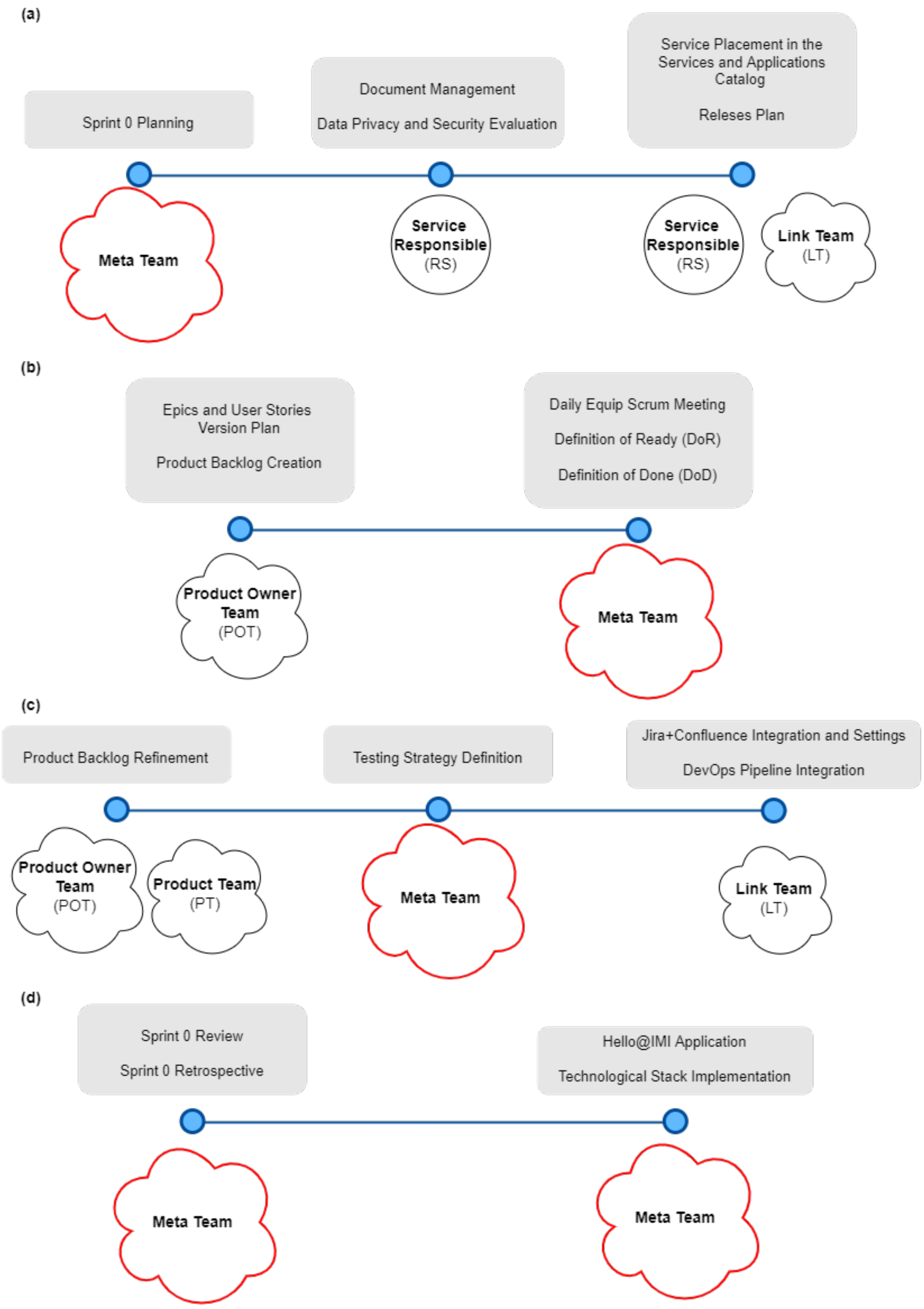


Figure 4.13: **The Sprint 0 Phase of the Scrum@IMI Framework**. The Figure legend is as in Figure 4.12.

Service Design: Activities (1.b), (2), and (3), contributing to the Scrum@IMI activities described in dots (1.a), (1.b), (1.c), (2.a), (3.a), and (3.c) in Figure 4.14. Software Development: Activities (1) and (2), contributing to the Scrum@IMI activities described in dots (1.b), (1.c), (2.a), (2.b), and (3.c) in Figure 4.14. Technology Management: Activity (1), contributing in part to the Scrum@IMI activities described in dots (1.b), (1.c), and (3.c) in Figure 4.14. Product Compliance & Verification: Activities (2) and (3), contributing to the Scrum@IMI activities described in dots (1.b), (2.b), and (3.b) in Figure 4.14.

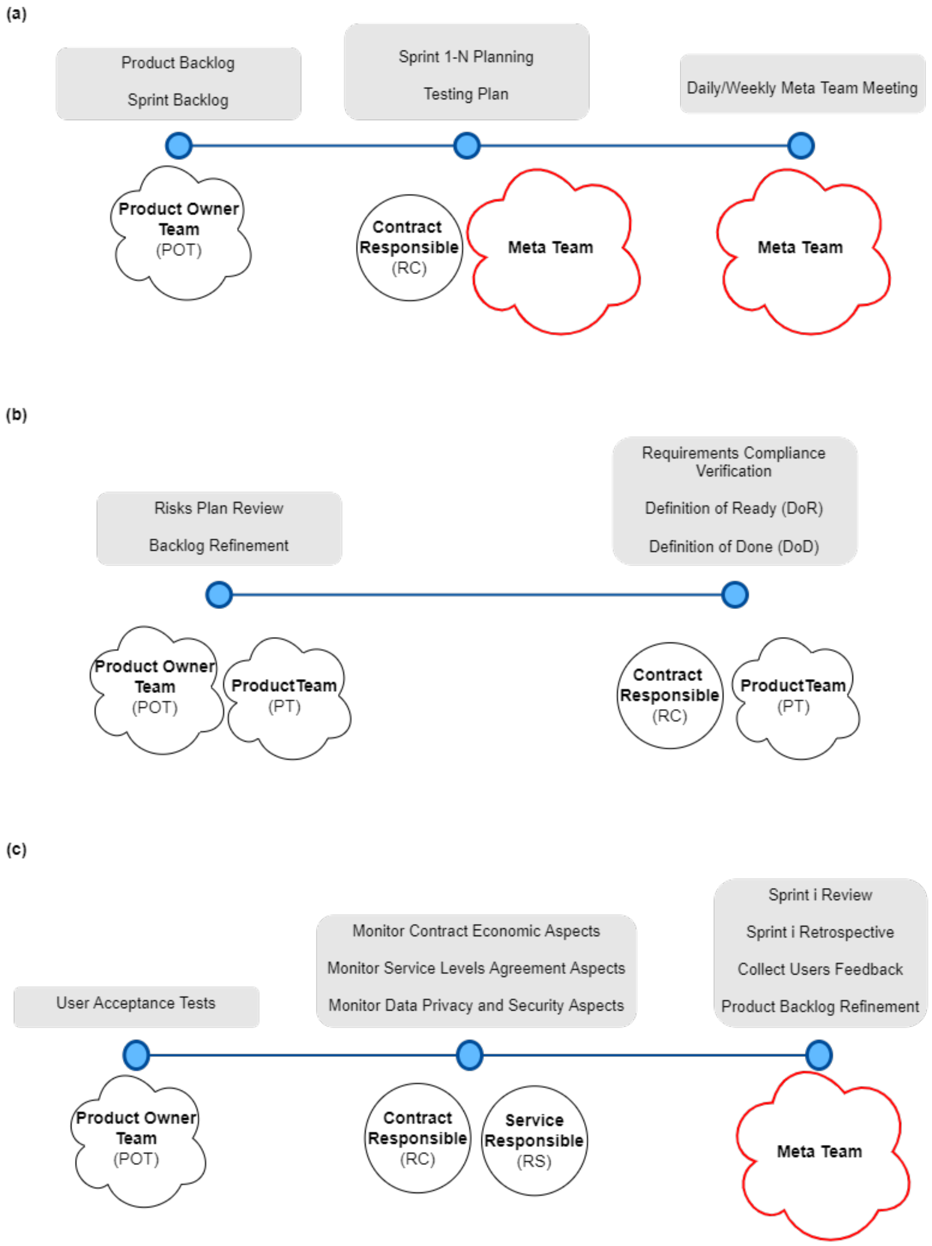


Figure 4.14: **The Sprint i Phase of the Scrum@IMI Framework**. The Figure legend is as in Figure 4.12.

Service Transition

The activities and teams involved in the Service Transition phase are described synoptically in Figure 4.15. This phase consists of the transition to service of a subset of implemented product increments and the consequent delivery to citizens.

In terms of Scrum@PA, this phase involves all four of its components. Namely, three Leader professional figures, i.e., SM, RC, and RS, and three Teams, i.e., POT, LT, and PT. The activities that are carried out can be phrased in terms of our paradigm as follows, with the dot numbering specified in the legend of Figure 4.12. Service Design: Activities (1.b), (2), and (3), contributing to the Scrum@IMI activities described in dots (1.a) and (2.b) in Figure 4.15. Software Development: Activities (1) and (2), contributing to the Scrum@IMI activities described in dot (1.a) in Figure 4.15. Technology Management: Activity (1), corresponding in part to the Scrum@IMI activities described in dots (1.a), (1.b), and in dot (2.a) in Figure 4.15. Product Compliance & Verification: Activities (2) and (3), contributing to the Scrum@IMI activities described in dots (1.a), (1.b) and in dot (2.a) in Figure 4.15.

Moreover, new feedback from the users is taken into account, which will be processed and incorporated into the Backlog to help adapt the product so that it more closely intercepts their needs.

4.7 Scalability: Our Proposal versus SAFe in the PA

As mentioned in the Introduction of this chapter, Agile methodologies for the PA must be scalable. Our proposal offers scalability. Indeed, depending on the service ecosystem that the PA intends to adopt, it can be customized in terms of Leaders and Team Member Figures involved, making the set of professional figures involved in the management and development of the relevant public service more or less articulate. There is lack of scalable and general paradigms for the adoption of Agile in the Digital Transformation. In fact, there is only a mention that SAFe could be used, but with no technical details (see, i.e., [210]). For the sake of completeness, we review the SAFe framework in its Essential configuration and compare it with our proposed Agile Scrum@PA methodology. In this regard, Essential SAFe is a framework that is sustainably built and configurable to be adapted to different "business" needs, in which Business and Technical team figures are organized into an Agile Release Train (ART), or core structure of SAFe, that is a long-running Agile team that develops, delivers, and often incrementally manages one or more solutions in a value stream. Usually refers to a group of five to twelve cross-functional Agile development teams. Each Agile team, which is part of the ART, has the classic roles of the classical Scrum methodology, i.e., Development Team, Scrum Master

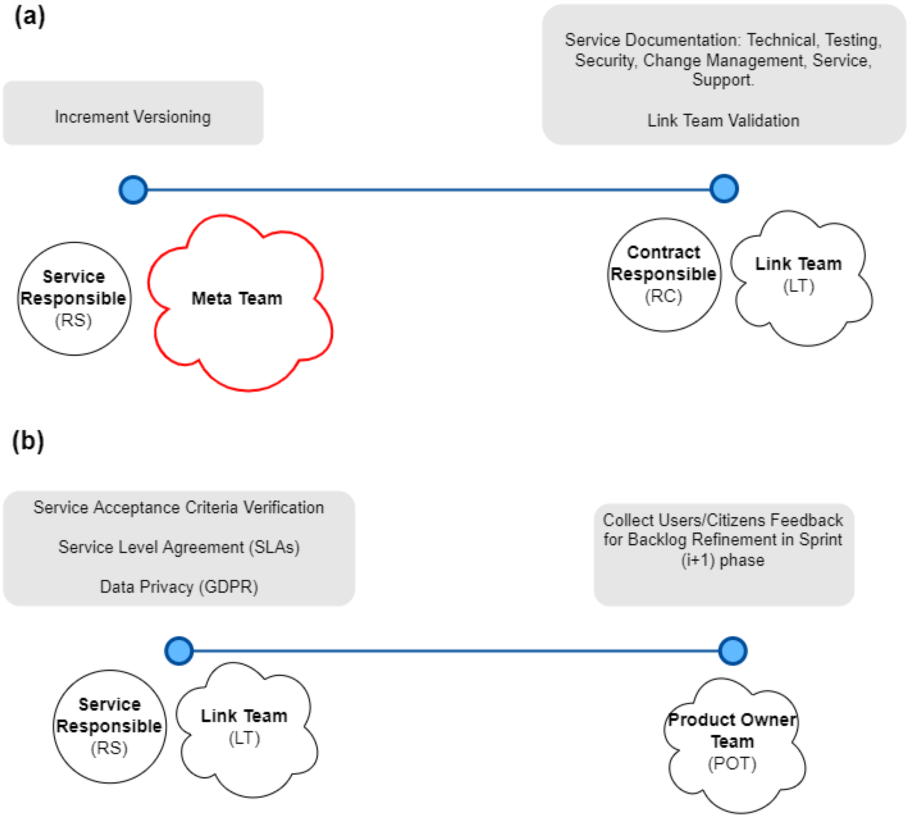

(a)
Increment Versioning
Service Documentation: Technical, Testing, Security, Change Management, Service, Support.
Link Team Validation
Service Responsible (RS)
Meta Team
Contract Responsible (RC)
Link Team (LT)
(b)
Service Acceptance Criteria Verification
Service Level Agreement (SLAs)
Data Privacy (GDPR)
Collect Users/Citizens Feedback for Backlog Refinement in Sprint (i+1) phase
Service Responsible (RS)
Link Team (LT)
Product Owner Team (POT)

Figure 4.15: **The Service Transition Phase of the Scrum@IMI Framework**. The Figure legend is as in Figure 4.12.

and Product Owner. The roles that characterize the ART are as follows:

- **Release Train Engineer**. It serves as the Scrum Master for the ART. In our proposal, it would correspond to the SM professional figure (see Figure 4.1).
- **Product Management**. It owns and prioritizes the Program Backlog. In our proposal, it would correspond to the PO professional figure (see Figure 4.1).
- **Business Owners**. They are critical stakeholders in the ART. In our proposal, they would correspond to the POT members professional figures (see Figure 4.3).
- **System Architect/Engineering**. It provides architectural guidance to ART teams. In our proposal, it would correspond to the professional figure of the architectural domain expert afferent to the LT (see Figure 4.4).
- **System Team**. It allows the ART to integrate and evaluate the tasks accomplished. In our proposal, we distinguish these team members into two

categories: one related to the integration aspects and the other related to the evaluation aspects of the completed work. For integration aspects, these figures should correspond to the domain experts pertaining to the LT, e.g., Security, Telecommunications, Operations and Systems, Service Management Office, and so on (see Figure 4.4), while for evaluation aspects, these figures should correspond mainly to the RS professional figure (see Figure 4.1).

In light of this, we can observe some overlap between what is in our proposal and what is in Essential SAFe framework. Nevertheless, our proposal offers a greater level of detail on the professional figures and teams involved (see Figures 4.1-4.2 and 4.3-4.4) and on their main interactions (see Figures 4.6-4.10).

4.8 Conclusions

In this chapter, we proposed a paradigm for the adoption of Scrum in the context of a PA, for the first time in the Literature, through the description of a set of steps organized in such a way as to achieve a structured and homogeneous adoption of this methodology in the public context. We first discussed the main ingredients useful for the transition from traditional to Agile methodologies in the DT context, through the provision of a set of recommendations that we have differentiated into two groups, one generic and the other more specific. Then, following what the UK Government accomplished, we defined a taxonomy of professional figures useful for addressing the challenges of creating digital services in the public context. These professional figures, first classified as Leaders and Team Members figures, were dropped within the Agile Scrum methodology for their adoption in the context of a PA. Moreover, focusing on the service development phase, we first extracted the main components, identifying for them the internal interactions among the figures and teams involved, and then we described the external interactions among the identified components. In addition, in order to show the validity of our proposal, we have showed that the quite successful Scrum@IMI, used for the DT in the City of Barcelona, can be successfully obtained as a special case of Scrum@PA.

Finally, we have discussed the scalability of Scrum@PA, and we compared it with the Essential SAFe framework, highlighting how the professional figures and teams within SAFe can mirror some of the Leaders and Team Members Figures from our proposal.

Part III

Agile Collaborative Software Development: The CAS Platform

Motivation

The aim of this part of the Dissertation is to present a novel and integrated development environment (IDE), consisting of a set of free and open source existing tools, primarily oriented to the support of Agile methodologies. This IDE would be utilized by the PAs to create and manage software projects, with the ultimate goal of providing better digital services to their citizens.

Indeed, as mentioned in Section 2.6 of the first part of this Dissertation, it is evident that professionals in many fields need to collaborate, even when using Agile methodologies for project management within a PA. Therefore, it is crucial to have a suitable set of software tools that can assist in facilitating such a collaboration. Not surprisingly, the tools available, both free and open source or commercial, are constantly evaluated to drive industrial and public institutions to make the best choices [290].

We now briefly recall the benefits of working collaboration tools within Agile methodologies.

- **Full Team Visibility**. It is critical for the team to have a clear understanding of the common goals in order to work together effectively to achieve them. Each team member must be aware of the outline of the project and each individual role in achieving the common goal. The use of brainstorming and team management tools can be helpful in achieving transparency, which in turn helps the team identify if a team member is struggling with a task and requires collaboration.

- **Tracking The Process**. Processes are always evolving, so teams require the ability to quickly adapt to changes, in particular in regard to the information related to the process. Those Agile collaboration tools can assist in managing evolving information, by simplifying workflows and promoting the desired agility. Moreover, these tools enable to change documents centrally, which ensures that everyone is working with the most up-to-date information.

- **Successful Implementation of The Project**. Team members bring their unique talents to the table, and agile collaboration tools help to facilitate the best possible use of these talents. Any tool that aids in collaborating with others and contributes to achieving a shared goal is valuable. The easier collaboration is, the more streamlined it is to achieve the desired outcomes.

Given the above, this part of the Dissertation consists of two chapters, the contents of which are briefly presented below.

Given the aim of this part of this Dissertation, already stated above, it is essential to come up with a selection of successful tools that are widely used by the community of Agile professional figures. Chapter 5 is dedicated to such a task. Indeed, we describe and evaluate a representative list of existing tools, including free and open source as well as commercial options, to assist Agile users in managing software projects. Given the amount of tools available, our selection is based on those that have been indicated as the most successful ones by the Agile users, as reported in [290, 291]. As a result of this chapter, we identify a set of successful tools, all free and open source, that can then be used as the building blocks mentioned above. Based on these building blocks, Chapter 6 presents an Agile IDE platform, referred to as Compositional Agile System (CAS), that has them as building blocks.

The CAS platform was proposed as a response to the need for an integrated environment created to support the Scrum project development methodologies and automate some of its tasks using open source software tools (see, e.g., [292]). It has also been successfully used by several groups of students in the Computer Science degree program at the University of Bologna. Starting from the available system, we have contributed the following for its growth.

- Rewriting docker containers for the building blocks setting up
- Refinement of the module regarding the developer's development activity tracking mode, i.e., CAS Logger Service.
- Development and integration of new features, i.e., productivity trackers for continuous monitoring of the productivity of the development team. These activities are typical of Agile development methodologies.
- Creating automation scripts (Ansible playbooks) for Setting Up the CAS platform.

We have tested the validity of the new CAS platform through the adoption of our proposed Agile methodology, i.e., Scrum@PA, using it with the associated tools and applying it to a specific case study concerning an Italian digital public service already widely and successfully used by more than 36 million Italian citizens.

Chapter 5

Collaborative Tools For Agile Methodologies Available Today

Abstract

The aim of this chapter is to come up with a selection of existing collaborative tools that are most widely used by the Agile community, which would then be used as building blocks for an integrated development environment, referred to as Compositional Agile System (CAS, for short), which we discuss in the next chapter. We proceed as follows. First, tools that are perceived to be valuable in the current State of the Art regarding the development of software products through the adoption of the Agile methodologies are presented, together with an evaluation of them. Then, we compare homologous tools, focusing on key aspects of the software development lifecycle, such as source code management, project management using Agile methodologies, static software quality analysis, and communication among the professionals involved in a project. Finally, after evaluating different options, we support our choice of free and open-source tools over commercial ones by comparing the benefits of an open-source tool versus a commercial one.

5.1 Introduction

Considering the importance of Agile collaborative tools in software project management, special care should be given to the choice of the most appropriate tools to adopt during the development process of a given software product related to a digital service offered by a specific PA.

The choice of development tools and platforms available to Agile developers is quite large, and indications regarding their effective use within such a community are scarce. Indeed, there are very few studies in this regard. One of them is [291] and, to the best of our knowledge, it is still quite useful, and, in view of a Literature search we have performed, it provides current information, with the exception of a few tools that have emerged after its publication (see, e.g., [293]).

This chapter is organized as follows. First, in Section 5.2, we present a selected set of basic existing collaborative tools most widely used by the Agile developers. Some of the latter tools are free and open-source, while others are commercial. Then, in Section 5.3, considering key aspects of the development process, e.g., source code management, project management using Agile methodologies, static software quality analysis, and communication among professionals involved in an agile project, we compare homologous tools, where each comparison places free and open-source tools up against commercial ones, among those identified in Section 5.2. Starting from the result of these comparisons, we focus our attention on free and open-source tools, motivating how they can better intercept the needs of many organizations. They are the foundation for the next chapter.

5.2 A Selected Set of Basic Tools

With reference to [291], and classifying development tools according to whether they are free and open-source or commercial, in the next two subsections, we briefly describe the ones that are of interest for the presentation of the material in this Dissertation.

Free and Open-Source

These are free and open-source tools, but it is worth pointing out that some of them have chargeable additional features and functionality.

- **GitLab, Incorporated**. It offers the following solution.
  - **GitLab** [294]. It provides a code repository and a collaborative software development environment for large DevOps projects. Based on the Git version control system [295], it allows to support standard practices that ensure software quality and rapid delivery, e.g., archive code online, track issues, and automate continuous integration/delivery (CI/CD) processes. Moreover, in its paid version, referred to as GitLab Enterprise Edition (GitLab EE), it provides additional end-to-end DevOps features for each stage of the software development lifecycle. It is worth pointing out that the online community of GitLab connects developers to share code and learn from millions of contributors. Additional details are available online in the GitLab Documentation portal [296].

- **Kaleidos, Incorporated**. It offers the following solution.

  - **Taiga** [297]. It is a project management tool for cross-functional teams working in an Agile manner, using either Scrum or Kanban frameworks. Taiga is a tool that facilitates, through its interface, collaborative project management. No training or complex configuration is required. It can be web-based or self-hosted, via on-premise instances. The self-hosted option is ideal for teams that wish to exploit an "internal development platform", storing all software documentation and project data on their private infrastructure or customize such a tool to fit their needs. Taiga enables efficient Sprint scheduling based on role-specific estimations and first-rate management of the tasks to be accomplished in the Sprint Backlog (see Section 3.3). Additional details are available online in [297].

- **Online Community of Contributors**. It offers the following solution.

  - **Jenkins** [298]. It is a project, maintained by an online community of contributors, and without a specific owner. It is a server tool that automates software development processes such as building, testing, and deploying. It supports various version control tools and it can execute projects based on, e.g., Apache Ant [299] and Apache Maven [300], shell scripts, and Windows batch commands. This tool is widely used in DevOps for continuous integration/delivery (CI/CD), allowing developers to integrate changes into the build and test them continuously. Jenkins is the most widely used CI/CD tool today and comes equipped with hundreds of plugins for building, deploying, and automating any project. In addition, Jenkins can distribute work across multiple machines, making it faster and more efficient to run builds, tests, and deployments across multiple platforms. Additional details are available online in [298].

- **Mattermost, Incorporated**. It offers the following solution.

  - **Mattermost** [301]. It is an integrated platform of productivity tools that provides workflow orchestration and eliminates the need for switching contexts among different tasks. It provides secure collaboration for technical and operational teams that work in environments with complex national-level security and trust requirements. It provides enterprise-grade collaboration tools through channel-based communication with unlimited and fully searchable message history, structured workflow execution, and project workflows. Channels are used for real time information sharing, flexibility, and *ad-hoc* activities, where communication workflows typically last for seconds to hours. Mattermost channels are ideal for various activities, such as bug triage, daily standups, posting announcements, and team conversations. It enables users to share various types of content such as files, images, emoji, GIFs, and links. Additionally, it supports rich markdown formatting, code syntax highlighting, and multilingual communication. In addition, integrated voice/video conferencing allows face-to-face connections. It also gives companies full control over their data. With self-hosted and private cloud deployment options, it provides access to the source code, allowing developers to be able to contribute directly to a shared, flexible and extensible platform built specifically for them. At this end, the platform is highly extensible and supports a rich ecosystem of third-party applications and integrations. Additional details are available online in [301].

- **JGraph, Limited Company**. It offers the following solution.

  - **Draw.io** [302]. It is a cross-platform, cloud and self-hosted configurable diagramming tool, with source code that the users can view and modify according to their needs. Its current name is *diagrams.net*. It ensures real time collaboration, making it easier for remote teams to work on the same project simultaneously. It contains many predefined shapes and templates internally and, at the same time, it is customizable. Indeed, it allows each organization to create new forms and libraries for the fulfilment of their goals. It can help teams to create different types of objects, such as visual charts, flowcharts, diagrams, mind maps, organization charts, E-R diagrams, Data Bases schemas, UML diagrams, using predefined templates on a drag-and-drop interface, while also being able to import and export diagrams in different formats, such as *.png*, *.jpg*, and *.pdf*. In its paid version, other import/export formats and additional functionalities are available. Additional details are available online in [302].

- **SonarSource, Société Anonyme**. It offers the following solution.

  - **SonarQube** [303]. It is a static code analysis tool that can be used for error identification and safety testing. It is available in free and paid versions. It allows users to measure and monitor quality continuously over time. SonarQube inspects and evaluates everything from minor stylistic choices to design errors, providing users with a rich searchable code history to analyze where code is cluttered. It helps users to determine whether it is a style issue, code error, code smell, code duplication, lack of test coverage, or overly complex code. The software analyzes source code from different aspects and analyzes code layer by layer, moving from the module level down to the individual class level. With SonarQube, code reliability and application security are ensured, and “technical debt”, i.e., the implicit cost of additional work due to implementing a simple, limited solution, rather than a more complex solution that would require more time to implement, is reduced by making the code base clean and maintainable. SonarQube supports over thirty different programming languages. However, popular programming languages such as C, C++ and Obj-C, are supported only in the paid version, together with Swift, ABAP, T-SQL and PL/SQL. Moreover, such a version also provides integration with CI/CD and feedback during code review with branch analysis and pull request decoration. Additional details are available online in [303].

Commercial

We now list some of the commercial software tools most widely used by the Agile community of professional figures for software project management and development. It is worth pointing out that there are free versions of these solutions that are limited in certain functionalities (e.g. they can only support a limited number of participants).

- **GitHub, Incorporated**. It offers the following solution.

    - **GitHub** [304]. It is a platform that provides a code repository and a collaborative software development environment, similar to GitLab, for source code management. It offers a variety of collaborative software development practices such as code review, pull requests, and multiple assignment features, that make the work of developers much easier. Code review helps to review code branches and combine changes with automated checks, while pull requests notify developers when new changes are made to a repository. There are also functions involving AI, such as GitHub Copilot [305], jointly developed by GitHub and OpenAI [306], which assists programmers in on-the-fly code snippets with suitable suggestions for their needs. It is worth pointing out that the online community of GitHub connects developers to share code and learn from millions of contributors. Additional details are available online in the GitHub Documentation portal [307].

- **Atlassian Corporation** [273]. It offers the following solutions.

- **Jira** [274]. It is a versatile and widely adopted project management tool, that excels in facilitating Agile methodologies such as Scrum and Kanban. This tool enables configuration on both cloud and on-premise infrastructures. It offers a comprehensive ecosystem for project planning, tracking, real time collaboration, and reporting. Beyond its foundational capabilities of managing backlogs, creating user stories, scheduling sprints, and tracking progress, it offers a comprehensive suite of features to empower teams to tailor the tool to their specific needs. It fosters transparency by providing a shared space where stakeholders can discuss issues, share updates, and make informed decisions. It provides meticulously tracked issues, defects, and tasks throughout the entire development lifecycle. Its robust issue-tracking capabilities enable teams to identify, prioritize, and resolve issues efficiently. It seamlessly integrates with numerous third-party tools and platforms, such as version control systems like Git to communication tools like Slack [275] (see below), making it a central hub for project-related activities. It places a strong emphasis on security and compliance, making it an ideal choice for organizations that require robust data protection measures and adherence to industry standards. Additional details are available online in [274].

- **Trello** [308]. It is a user-friendly and versatile tool for task and project management that employs visual boards and cards to help individuals and teams stay organized and productive. Its simplicity and flexibility make it suitable for a wide range of tasks and project management styles. With Trello, users can create custom boards, add tasks as cards, and move them through visual columns that represent workflow stages, providing a clear and intuitive way to track tasks, collaborate in real time, set deadlines, and manage projects effectively. The integration capabilities of Trello and the extensive ecosystem of add-ons further increase its usefulness, making it a popular choice for individuals, small teams, and large organizations looking for an easy-to-use, yet powerful tool for task and project management. Additional details are available online in [308].

- **Confluence** [309]. It is a robust and versatile collaborative platform designed to enhance teamwork, simplify documentation creation, as well as centralize knowledge management. It is a wiki-style tool that enables users to create and organize a wide range of objects, such as documents, spreadsheets, meeting notes, and project plans. It offers robust search capabilities, helping users to quickly locate specific information across a vast repository of objects. Users can set granular permissions to control who can view, edit, or comment on specific content. This ensures data security and privacy. Its user-friendly interface and extensive feature set make it a valuable tool for organizations and teams looking to improve communication, productivity, and information sharing. It seamlessly integrates with various Atlassian products, such as Jira and Trello, as well as numerous third-party apps and plugins. This integration streamlines workflows and enhances functionality. Finally, it is suitable for teams of all sizes, from small startups to large enterprises, thanks to its scalability and performance. Additional details are available online in [309].

- **Asana, Incorporated** [273]. It offers the following solution.

  - **Asana** [310]. This is a project and work management platform that is both flexible and powerful. It is designed to help individuals, teams, and organizations manage, track, and plan tasks and projects effectively. It comes with an extensive set of features that streamline workflow, improve collaboration, and boost productivity across various sectors and use cases. Users have the option to work with Kanban boards, which provide a visual representation of tasks in different workflow stages. This makes it perfect for Agile and Lean [272] project management methodologies. It is user-friendly and enables users to create, organize, and assign tasks and projects effortlessly. Tasks can have descriptions, due dates, subtasks, and attachments. It offers a visual timeline view that assists users in planning and scheduling tasks, milestones, and project timelines. This feature provides a clear overview of project progress and dependencies. Additionally, it integrates with a wide range of third-party applications and services, such as calendars, email clients, cloud storage, and communication tools. Reporting features allow teams to track project progress, monitor performance, and gain insights into their work. The platform prioritizes security and provides features such as role-based permissions, SSL encryption, and compliance with data protection regulations. It is scalable and suitable for teams of all sizes, from small startups to large enterprises. Additional details are available online in [310].

- **Slack Technologies, Limited Liability Company** [273]. It offers the following solution.

  - **Slack** [275]. It is an enterprise-level communication tool, designed to optimize team collaboration and communication within organizations, with advanced security, compliance, and administration features, making it suitable for large organizations with complex requirements. It serves as a central hub for teams to exchange messages, share files, and integrate with various tools and services, improving productivity and teamwork. It provides real time messaging and chat capabilities, allowing team members to have one-on-one conversations or participate in group discussions through channels that can be organized by topic, project, or team, making it easy to find and reference conversations. Within it, users can easily share files, documents, images, and other media. This tool also supports collaborative editing and integration with various file storage and sharing services. It allows users to connect with a wide range of third-party applications and services, enable teams to automate tasks, receive notifications, and access external tools without leaving the Slack environment. It supports video and voice calls, and it archives all conversations, ensuring that historical data is accessible for reference or compliance purposes. It places a strong emphasis on security and compliance, offering features such as data encryption, two-factor authentication (2FA), and compliance with various industry standards and regulations. Additional details are available online in [275].

- **Microsoft Corporation** [311]. It offers the following solutions.
  - **Microsoft Teams** [312]. It is a communication platform, part of the Microsoft 365 suite of products. The Teams service includes features such as instant messaging, audio and video calling, advanced online meetings, shared files and apps, mobile experiences, and comprehensive web conferencing capabilities. This platform enables team members to easily share information, and discuss project details seamlessly. Additional details are available online in [312].
  - **Microsoft Azure DevOps** [313]. It is a comprehensive set of cloud-native tools and services to support the entire software development lifecycle. It provides a unified platform for planning, developing, testing, delivering, and monitoring software projects, fostering collaboration and automation throughout the process, in particular for Agile methodologies such as Scrum and Kanban. Teams can prioritize tasks, set sprint goals, and manage work items efficiently. It includes a Git-based version control system that enables developers to collaborate on code, track changes, and manage code repositories securely. It automates the software delivery pipeline, allowing teams to build, test, and deploy applications continuously, across various platforms and environments. It provides a secure and scalable artifact repository for storing and managing binary files, dependencies, and packages. It includes collaboration features such as wikis, boards, and dashboards to foster communication and transparency among development teams, product owners, and stakeholders. Teams can gain insights into project progress, code quality, and release

performance through customizable dashboards, charts, and reports. It is designed with security in mind, providing features like role-based access control, security scanning, and compliance reporting to meet regulatory requirements. It can be scaled and customized to meet the unique needs of diverse development projects. Additional details are available online in [313].

- **Checkmarx, Limited Company** [314]. It offers the following solution.
  - **Checkmarx Static Application Security Testing (SAST)**. It is a powerful security tool designed to identify and eliminate security vulnerabilities, including common coding errors, insecure coding practices, and potential backdoors, in application source code during the development phase. It provides rapid and in-depth incremental or full static analysis of source code and offers the flexibility, accuracy, integrations and coverage needed to protect applications. Moreover, it provides detailed and actionable remediation guidance to developers, offering specific recommendations to fix identified security vulnerabilities, including, e.g., SQL injection, Cross-Site Scripting (XSS), and cryptographic issues, and strengthen the overall security posture of the application. It supports automation and continuous scanning, enabling developers to detect, prioritize, and eliminate security issues in the various stages of the software development lifecycle. This proactive approach helps prevent vulnerabilities from being introduced into the final product. In terms of reporting and analytics, it generates comprehensive reports and analytics, providing development teams and security professionals with a clear overview of the security status of the application, including the severity of vulnerabilities and their potential impact. Finally, in terms of compliance and regulatory support, assists organizations in meeting compliance requirements by identifying and resolving security issues that may violate regulations and industry standards, providing training and support to development teams in understanding and implementing best practices for designing secure code, enabling them to proactively identify and resolve security issues encountered. Additional details are available online in [314].

5.3 Assessment of Suitability and Interest in Open-Source Tools

Now we propose a comparison of some free and open-source tools versus commercial ones, highlighting the strengths and weaknesses of the tools identified. We focus on the main aspects of the software development lifecycle, such as source code management, project management using Agile methodologies, software quality analysis, and communication among the professionals involved in the project. Upon completing the comparisons, the identified tools are compulsory for the next chapter.

- **GitHub Vs. GitLab**, for source code management.
- **Jira Vs. Taiga**, for project management and Agile planning.
- **Checkmarx SAST Vs. SonarQube**, for static software quality analysis.
- **Slack and Microsoft Teams Vs. Mattermost**, for the communication aspects.

The next four subsections are devoted to the listed comparisons.

GitHub Vs. GitLab

Given the description of these tools in the previous section, GitLab is preferred over GitHub, particularly as far as public sector organizations are concerned, because it is an open-source tool that provides a free version (GitLab CE) that includes a very extensive feature set for development teams. This is particularly attractive, for startups, small teams, and organizations with limited budgets, being that it provides many features of the DevOps pipeline, e.g., Git, CI/CD, container logging, issue tracking and code review, without requiring the purchase of user licenses. In the PA context, this translates into cost rationalization. GitLab, unlike GitHub, allows the option of being deployed independently on the infrastructure of the organization or used in the cloud-hosted version of `GitLab.com`. In highly regulated organizational settings, e.g., PAs and mission-critical scenarios, where there is a need to maintain control over applications and the data hosted therein, this flexibility allows organizations to choose the option that best fits their requirements and security policies. GitLab places a strong emphasis on security. It provides various features such as integrated container scanning, security dashboards, and automated security testing to simplify the identification of vulnerabilities in code and infrastructure and to handle them appropriately. Furthermore, GitLab simplifies access controls, implementation of audit trails, and policy reinforcement, which are crucial for sectors with strict regulatory requirements such as the PAs. Both GitLab and GitHub have internal features that allow teams to create wiki pages to document software products developed by them, thereby reducing the need for external tools and services. However, GitLab is preferable over GitHub because it offers the possibility to create complete static pages and to integrate the Draw.io tool, which has already been available since release 15.10. All files in GitLab, including the embedded objects created with Draw.io, have a version, which allows for tracking of all changes made. Finally, the use of GitLab, as an open-source DevOps platform for source code management, instead of commercial tools such as GitHub EE or Bitbucket [276] has several advantages, depending on the needs of the project and the specific priorities of the development team. On the other hand, commercial tools may offer additional functionality, support, or integrations that may be useful for some organizations. Table 5.1 summarizes this comparison.

Table 5.1: **Comparison Between GitHub and GitLab**. The first column displays the key features by which we compare the two tools. A "+" symbol indicates the presence of the feature. Two "++" symbols indicates greater emphasis on the potentials of that feature.

KEY FEATURE	GitHub	GitLab
Open Source	NO	YES
Self-Hosted Configuration	NO	YES
Strong Emphasis on Security	+	++
Create Product Documentation Capabilities	+	++

Jira Vs. Taiga

For Agile project management and issue tracking, given the description of these tools in the previous section, Taiga is preferred to Jira, mainly because of its ease of use. It is known for its intuitive interface and ease of configuration. It is often considered quite intuitive for small to medium-sized teams or projects that do not require the complexity of configuring tools such as Jira. The open source version of Taiga allows users greater flexibility in customization and integration with other tools. In fact, although Jira offers a wide range of integrations due to its popularity, Taiga also offers a good set of integrations with the most widely used tools and services, making it suitable for many workflows. Compared to enterprise-grade tools such as Jira, Taiga is Agile-focused in that it is designed with a strong emphasis on Agile methodologies. If the team primarily follows Agile methodologies such as Scrum or Kanban, Taiga can provide a more customized experience than Jira. Taiga has an active open source community, which can be useful for finding tools to common problems and getting support from other users. On the other hand, Jira has a larger user base and comprehensive documentation, but premium support can be expensive. An additional feature that makes Taiga preferable to Jira is the Visual Boards. They are often lauded for their simplicity and effectiveness in managing tasks and workflows. If users prefer a visual approach to project management, Taiga may be more appealing. However, it is important to note that Jira also has its strengths, including robust customization options, numerous add-ons, and integration with other Atlassian commercial products such as Confluence and Bitbucket. Jira is widely used for large companies and teams with complex project management needs. On the other hand, Taiga also adequately meets the needs of fairly large projects. Finally, in terms of cost, Taiga offers a free plan with basic functionality that essentially does not include premium support, which is provided in the paid plan. This choice can be beneficial for small teams or startups with limited budgets, but like Jira, relying on technical support can be costly, especially in large-scale projects with several teams or for using advanced features. Table 5.2 summarizes this comparison.

Table 5.2: **Comparison Between Jira and Taiga**. The table legend is as in Table 5.1.

KEY FEATURE	**Jira**	**Taiga**
Open Source	NO	YES
Self-Hosted Configuration	NO	YES
Easy of Use	NO	YES
Environment Customization	+	++
Emphasis on Agile Methodologies	NO	YES
Premium Support	YES	NO

Checkmarx SAST Vs. SonarQube

For static software quality analysis, given the description of these tools in the previous section, SonarQube is preferred to Checkmarx SAST, mainly because of its ease of use and its flexibility in handling small and medium-sized projects. SonarQube and Checkmarx SAST are both tools in the software development industry that serve different purposes but overlap in some areas. They both focus on security but approach it from slightly different angles. SonarQube is more focused on overall code quality, including security issues. It provides a wide range of code quality metrics and covers a broad spectrum of code-related issues. It is primarily used for continuous inspection of code quality to perform automated reviews with static code analysis to detect bugs, code smells, and security vulnerabilities. On the other hand, Checkmarx SAST is focused on integrating security within the entire software development lifecycle. It is particularly useful in identifying complex security threats in enterprise-level applications. It is specifically designed for security and provides comprehensive security analysis to identify vulnerabilities such as SQL injection, cross-site scripting (XSS) and others that can lead to serious security breaches. In terms of integration, both SonarQube and Checkmarx SAST can be easily integrated with a number of build tools and IDEs, such as Eclipse [315], Visual Studio [316], Visual Studio Code Extension [317] and IntelliJ [318], thus becoming an essential component of the development workflow. SonarQube can also be integrated with other tools such as Jenkins and Microsoft Azure DevOps for automated code reviews. Checkmarx SAST also gives integrations with bug-tracking systems such as Jira and build management tools such as Apache Ant and Apache Maven. These integrations ensure that security testing is seamlessly integrated into the development process. SonarQube and Checkmarx SAST offer different levels of customization and scalability. SonarQube is suitable for small and medium-sized projects and can handle a moderate volume of code. It offers a number of plugins and extensions that allow users to customize rules and extend functionality. On the other hand, Checkmarx SAST offers extensive customization options and is designed to meet the needs of large companies with complex software architectures. It is capable of scaling with increasing application size and complexity. Finally, in terms of Reporting and Compliance capabilities, SonarQube provides detailed reports on code quality metrics, making it easier to track improvements over time. However, its security reporting might not be as detailed as Checkmarx SAST. In fact, Checkmarx SAST provides comprehensive reports for compliance purposes, often necessary for industries with stringent security regulations, such as finance and healthcare sectors. Table 5.3 summarizes this comparison.

Table 5.3: **Comparison Between Checkmarx SAST and SonarQube**. The table legend is as in Table 5.1.

KEY FEATURE	Checkmarx SAST	SonarQube
Self-Hosted Configuration	NO	YES
Easy of Use	NO	YES
Overall Code Quality	NO	YES
Identifying complex security threats in enterprise-level application	YES	NO
Comprehensive Reporting for compliance purposes	YES	NO

Slack and Microsoft Teams Vs. Mattermost

Choosing between combining the Slack tools with Microsoft Teams and Mattermost for the communication aspects between stakeholders and Agile team members depends on several factors, again, including the specific needs, budget, and preferences of the development team. For the communication and collaboration aspects among Agile team members, given the description of these tools in the previous section, Mattermost is preferred over the combination of tools, such as Slack and Microsoft Teams, mainly due to its ease of use. Indeed, although Slack and Microsoft Teams have refined, customizable, and very comprehensive interfaces that make them easy for most users to adopt, Mattermost, on the other hand, offers a clean and customizable interface, but it may require more effort, during configuration, to achieve an equally refined look and feel that suits the needs of the organization that intends to adopt it. In terms of workspace integration and customization, Slack has a large ecosystem of integrations and applications available through its App Directory, including widely adopted third-party tools. Microsoft Teams, on the other hand, stands out for its deep integration with the Microsoft 365 product suite, making it a natural choice for organizations already using Microsoft products. However, Mattermost has a dedicated app market and offers a smaller ecosystem of add-ons than competing commercial tools such as Slack or Microsoft Teams. Among the most interesting Mattermost extensions, we find the audio/video calling feature via the free and open source Jitsi [319] plugin, a cross-platform voice, video conferencing and instant messaging application for the Web platform. Mattermost offers an advantage over Slack and Microsoft Teams in that it can be configured on an infrastructure of the organization, offering internal control over data and encryption capabilities over it. In highly regulated organizational settings, e.g., PAs and mission-critical scenarios, where it is necessary to maintain control over applications and the data hosted therein, this option allows organizations to choose the option that best fits their security requirements and policies. Slack and Microsoft Teams, on the other hand, are cloud-based tools and are hosted on their respective platforms on the Web. Finally, in terms of costs, Slack and Microsoft Teams both offer free plans with limitations as well as paid plans with advanced features. Mattermost is open source and can be hosted for free, making it a more cost-effective option, provided that organizations have the technical expertise in-house to run this tool on their infrastructure. Table 5.4 summarizes this comparison.

Table 5.4: **Comparison Between Slack and Microsoft Teams and Mattermost**. The table legend is as in Table 5.1.

KEY FEATURE	Slack and Microsoft Teams	Mattermost
Self-Hosted Configuration	NO	YES
Easy of Use	NO	YES
Environment Customization	+	++
Video Conferencing and instant messaging	+	++

5.4 Conclusions

In this chapter, we have proposed a selection of collaborative tools, which are currently used by the community of Agile professional developers. First, we presented a set of existing basic collaborative tools, including free and open source options as well as commercial tools. Then, considering key aspects of the software development lifecycle, such as source code management, project management using Agile methodologies, static software quality analysis, and communication among the professionals involved in the project, we compared homologous tools, where each comparison placed free and open source tools against commercial ones, among those that were identified in Section 5.2. From the result of these comparisons, we focused on the free and open source tools, motivating how they can better intercept the security requirements and policies of many organizations, public and private. The identified free and open source tools are a prerequisite for the next chapter. n fact, these tools can be used as building blocks of an integrated development environment, called Compositional Agile System (CAS, for short), and which we discuss in the next chapter.

Chapter 6

The CAS Platform

Abstract

The aim of this chapter is to present an Agile open source development environment, designed in a modular manner, which allows additional components to be easily added within the environment to have complete control over its implementation and extensions. We proceed as follows. First, we analyze the requirements for the design of an Agile IDE, in terms of high level tasks and activities, while also pointing out the Agile tools for its implementation, following the recommendations that we have provided in the previous chapter. Then, considering some of the Scrum professional figures, already mentioned in the second part of this Dissertation, we illustrate how the high level tasks and activities, mentioned earlier, can be mapped within the Scrum methodology, via the writing of some user stories. Furthermore, considering the Agile tools, mentioned earlier, as building blocks, we present an IDE, referred to as the Compositional Agile System (CAS) Platform, providing first a high level view, then giving details: on the type of authentication offered to its end users; on its architecture using microservices; and how it can be deployed on an organization's infrastructure. We have made available the CAS Platform on the GitHub public repository [320]. In addition, we present a comparison between CAS and the State of the Art, regarding comparable, or in some ways complementary, commercial and open-source systems. Finally, we provide a specialized version of CAS for Scrum@PA. Then, in order to demonstrate the validity of our approach, we show how to use the CAS platform to conduct an ex-post analysis on *IO App*, i.e., an Italian digital service widely used by more than 36 million citizens, for which the GitHub repository is available online in [321].

6.1 Introduction

We intend to build an IDE, which can be set up on a private premises or in a hybrid cloud to provide an *internal developer platform*, and consists of the building blocks that we identified in the previous chapter 5.

Thanks to these building blocks, which in themselves are quite complicated, it is possible to integrate them to achieve more complex systems. As stated in the motivations of this part of the Dissertation, we build on a known platform, for which we have given you some details about its history. Our contributions are about making some improvements and refinements to it. For the interested reader, the repository of scripts for setting up this platform is available in [320].

First, we show how this IDE is designed on the professional figures of the standard Scrum, and then, we specialize it on the leader professional figures of our Scrum@PA methodology. In this latter case, we emphasize that the professional figures to be supported are a subset of those considered for Scrum@PA, on which we limit ourselves to simplify the presentation of the CAS collaborative platform. Therefore, we have chosen only a few key figures, whose user stories are given. For the format of those user stories, we conform to the standard practices in the Scrum methodology.

This chapter is organized as follows. First, in Section 6.2, according to the recommendations, provided in the previous chapter, we analyze the requirements for designing an Agile IDE in terms of high level tasks and activities, while also pointing out the Agile tools for its implementation. Then, we illustrate how the high level tasks and activities, mentioned earlier, can be mapped within the Scrum methodology. Section 6.3 is devoted to the presentation of an IDE referred to as the Compositional Agile System (CAS) platform. To this end, we first present a high level description of the platform, then in order: mode of access, setting up the platform, server-side details about its microservice architecture, and client-side details regarding the contributions synchronization of the work of the developers with the CAS Server. Section 6.4 is devoted to a comparison between the CAS platform and the State of the Art of commercial systems and similar open-source, or in some ways, complementary tools. Finally, in Section 6.5, first we specialize CAS for some of the Scrum@PA professional figures that we identified in the previous chapter, and then, in order to demonstrate the validity of the project, we validated the CAS platform through an ex-post analysis on a public digital service of the Italian PA, i.e., *IO App*.

6.2 Designing an Agile IDE and Choosing Software Tools to Support It: Requirements

A development environment that can be used by professional figures working as an Agile team, on a given project, must match the specific needs and preferences of the team, and grant flexibility to adapt and scale with the evolving needs of the project and team. Moreover, professional figures must be enabled to collaborate and manage projects efficiently, ensure code quality, automate processes, and deliver high quality software iteratively and incrementally. We formalize those intuitive requirements in terms of high level tasks and activities in subsection 6.2. For each of them, we also indicate software tools that can be used to carry them out, following the recommendations provided in the previous chapter. Then, the developed framework is specifically applied to standard Scrum in subsection 6.2.

High Level Tasks and Activities and Tools Supporting Their Realization

1. **Collaboration and Communication**

   (a) Grant real time communication and collaboration among team members, even with the possible use of messaging platforms or chat tools. The suggested tool is Mattermost and, in part, the communication and collaboration tools in GitLab.

   (b) Grant the ability to share code snippets, files and documents within the team. The suggested tools are included in GitLab and in Taiga.

2. **Version Control System (VCS)**

   (a) Ensure collaborative code management, including code reviews and tracking changes made by team members. A VCS is required. The suggested tools are the git-based repositories, such as the ones supported by GitLab.

   (b) Ensure concurrent development and code integration. This requires support for branching and merging strategies. The suggested tool is GitLab.

3. **Agile Project Management**

   (a) Enable effective backlog management, sprint planning, and task tracking. The suggested tool is Taiga.

   (b) Grant support for user stories, epics, and task boards for organizing and prioritizing development activities. The suggested tool is Taiga.

   (c) Enable visualization of team progress, burn-down charts, and sprint velocity to monitor project status. The suggested tool is Taiga.

4. **Automated Testing and Continuous Integration**

   (a) Enable the automation of the build, testing, and deployment processes. The suggested tools are included in Jenkins and in GitLab.

   (b) Adopt the automated configuration of test suites and run them as part of the development workflow. The suggested tools are included in Jenkins and in GitLab.

   (c) Ensure timely issue resolution with notifications and alerts for failed builds or tests. The suggested tools are included in Jenkins and in GitLab.

5. **Code Quality and Static Analysis**

   (a) Perform static code analysis and enforce coding standards. The suggested tool is included in SonarQube.

   (b) Enhance code quality and maintainability via automated code review and feedback. The suggested tool is included in SonarQube.

   (c) Grant the identification of code smells, vulnerabilities, and potential performance issues. The suggested tool is included in SonarQube.

6. **Documentation and Knowledge Sharing**

   (a) Grant the sharing of project knowledge, design decisions, documentation and guidelines. The suggested tools are included in Taiga and in GitLab.

   (b) Enable the collaborative creation and maintenance of the project documentation, API documentation, and user guides including collaboration features. The suggested tools are included in Taiga and in GitLab.

7. **Issue Tracking and Bug Management**

   (a) Ensure the ability to assign, prioritize, and track issues, bugs, and feature requests, to grant timely resolution. The suggested tool is an issue-tracking systems such as the one included in GitLab.

   (b) Ensure the workflow to fit the development process and the problem-solving flow of the team. The suggested tool is included in GitLab.

8. **Continuous Monitoring and Feedback**

(a) Ensure the collection of metrics on system performance, usage, and errors via monitoring tools. The suggested tool is included in GitLab.

(b) Ensure the provision of input and reporting of problems through feedback mechanisms for end users or stakeholders. The suggested tool is included in GitLab.

(c) Enable the visualization of the project health, identify bottlenecks, and drive continuous improvement via monitoring and analytics dashboards. The suggested tools are included in GitLab and, in part, in Taiga.

Standard Scrum

In this section, we illustrate how the high level tasks and activities, mentioned in the previous subsection, can be mapped within the Scrum methodology. To this end, we need to consider the Scrum professional figures relevant for this Dissertation and detail them in terms of user stories. It is worth mentioning that three of these figures are Leaders (see Section 4.4), while one is part of the Development Team (see Section 3.3). In addition, for the notation of those professional figures we use the same terminology adopted in the mentioned sections.

- **As a Product Owner**.

  1. I need to monitor productivity data during the project to predict when a release would be possible to ensure on-time delivery. This corresponds to activities 1.(b), 2.(a), 3.(c), 6.(a), 7.(a) and 8.(c), mentioned in Section 6.2. Consequently, the tools to be used are Taiga, and GitLab. Since the PO, in the context of this activity, needs to interact with other professional figures, i.e., the SM, the Developer, and the S, these tools would also be available to them.
  2. I must check the code delivered by the team against the Product Backlog (percentage of work adopted), including the Definition of Done. This corresponds to activities 3.(a)-3.(c), and 4.(a)-4.(c), mentioned in Section 6.2. Consequently, the tools to be used are Taiga, Jenkins, and GitLab. Since the PO, in the context of this activity, needs to interact with the Developer, these tools would also be available to her/him.
  3. I have to keep track of how the user stories in the Product Backlog have evolved, and turned into functional code, to plan the next release. This corresponds to activities 3.(a)-3.(c), and 8.(c), mentioned in Section 6.2. Consequently, the tools to be used are Taiga, and GitLab. Since the PO, in the context of this activity, needs to interact with the Developer, these tools would also be available to her/him.

- **As a Developer or Stakeholder (or Interest Group)**.

4. I need to communicate easily, quickly and securely with all other Developers or Stakeholders (or Interest Groups). This corresponds to activities 1.(a)-1.(b), and 3.(a), 6.(a)-6.(b), mentioned in Section 6.2. Consequently, the tools to be used are Mattermost, Taiga, and GitLab. Since the Developer or Stakeholder, in the context of this activity, needs to interact with the PO, these tools would also be available to her/him.

- **As a Developer**.

5. I need a support tool for asymmetric pair programming to get help from senior Developers or Domain Experts. This corresponds to activities 2.(b), and 6.(b), mentioned in Section 6.2. Consequently, the tool to be used is GitLab. Since the Developer in the context of this activity, needs to interact with the SM, this tool would also be available to her/him.
6. I need a support tool to check the quality of my code to meet the Definition of Done. This corresponds to activities 5.(a)-5.(c), mentioned in Section 6.2. Consequently, the tool to be used is SonarQube. Since the Developer in the context of this activity, needs to interact with other professional figures, i.e., the SM and the PO, this tool would also be available to them.
7. I need a support tool to monitor defects in open-source libraries and components. This corresponds to activities 2.(a)-2.(b), 6.(b) and 7.(a), mentioned in Section 6.2. Consequently, the tools to be used are GitLab and Taiga. Since the Developer in the context of this activity, needs to interact with other professional figures, i.e., the SM and the PO, these tools would also be available to them.

- **As a Scrum Master**.

8. I need a support tool to ensure that the adoption of Agile methodology is followed through the entire software product development cycle. This corresponds to activities 1.(a), 3.(a)-3.(c), and 7.(b), mentioned in Section 6.2. Consequently, the tools to be used are Taiga, GitLab and Mattermost. Since the SM in the context of this activity, needs to interact with other professional figures, i.e., the PO and the S, these tools would also be available to them.

For the convenience of the reader and for later reference, we summarize the above presentation in Table 6.1.

Table 6.1: **Mapping User Stories and main Scrum Agile Roles to Some Open-Source Tools**. From left to right, in the first column, we find the identifier of the individual user story. In the second column, we find the goal of the user story. In the third column, we find the main tools, free and open-source, for high level programming environments, that developers, and in general, all the actors involved in the project, can use for the pursuit of their activities. In the fourth column, we find the main professional figures involved in the use of the tools indicated in the previous column.

#	USER STORY	TOOLS	SCRUM ROLES
U.S. 1	Checking the Team Productivity	Taiga, and GitLab	PO, SM, S, Developer.
U.S. 2	Check the Code (Test-Driven Development Strategy)	Taiga, GitLab, and Jenkins.	PO, Developer.
U.S. 3	Track User Stories into Code and Releases	Taiga and GitLab.	PO, Developer.
U.S. 4	Supporting Communication Among all Stakeholder	Mattermost, GitLab, and Taiga.	Developer, PO, S
U.S. 5	Asymmetric Remote Pair Programming	GitLab.	Developer, SM.
U.S. 6	Check Code Quality and Technical Debt	SonarQube.	Developer, SM, PO.
U.S. 7	Issue Tracking and Library Monitoring	GitLab, and Taiga.	Developer, PO, SM.
U.S. 8	Following Agile Methodology Processes	Taiga, Mattermost, and GitLab.	SM, PO, S.

6.3 The Compositional Agile System Platform for Scrum: An Agile IDE

In this section, we present an IDE referred to as the Compositional Agile System (CAS) Platform. Its aim is to offer the software tools that we have identified in Table 6.1, in terms of a set of integrated microservices, with the possibility of easy extensions. CAS is a web application accessible to any user connecting to it. However, we also consider the case in which a client machine can connect to it to utilize its services. Although such a client is external to this platform, we include it in our presentation since we expect that an Agile developer works on her/his machine but the contributions of the software project are managed by CAS. To this end, we provide a prototype description of a client in terms of software that is expected to have.

For the presentation of CAS we proceed as follows. We present a high level description of it in subsection 6.3. In subsection 6.3, we provide some details regarding the CAS user authentication, i.e., Single Sign-On through GitLab credentials. In subsection 6.3, we provide details regarding the setting up of the CAS Platform. The details of the CAS microservices architecture are presented in subsection 6.3. Finally, details about the synchronization of the contributions of the developer users with the CAS Server are provided in subsection 6.3.

CAS: High Level View

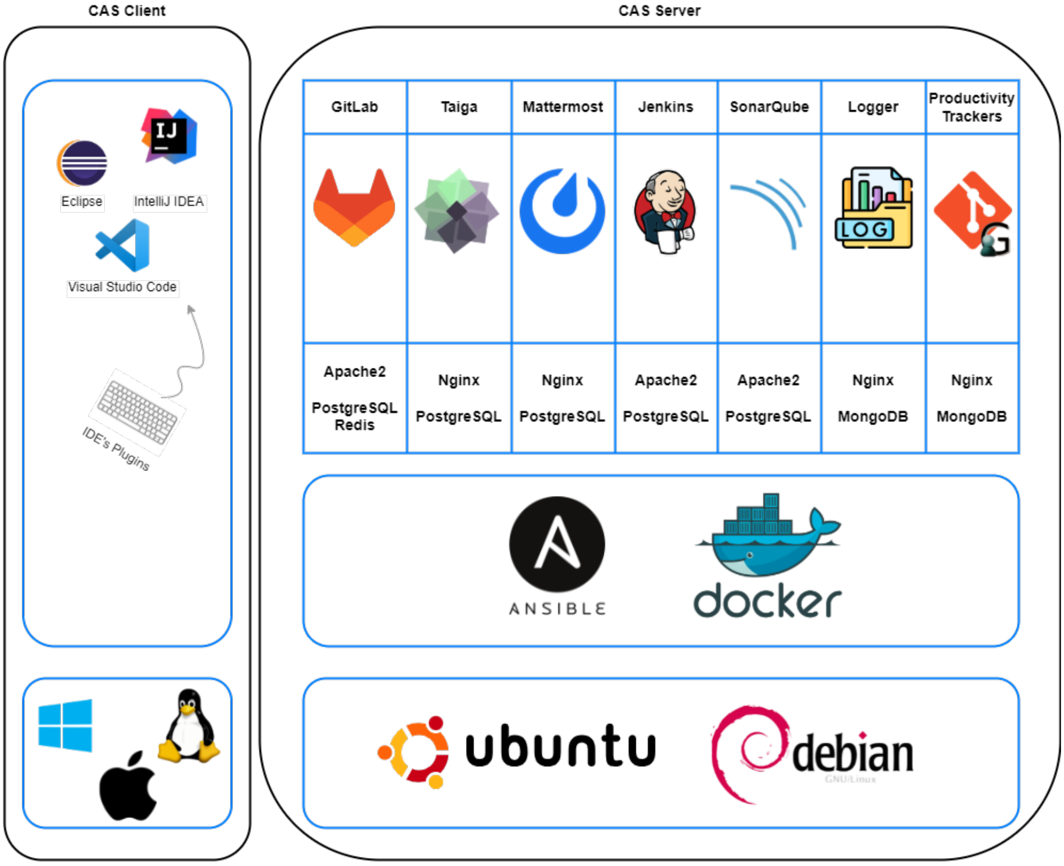
Figure 6.1 provides the representation of CAS, divided into server and client. Next, we describe them both.

- **CAS Server**. CAS offers Single Sign On authentication (SSO, for short) to access the microservices in an integrated way. The benefits of this type of authentication are discussed in Section 6.4. Therefore, prior to the first use of it, a user must register within the platform to obtain a GitLab user profile by choosing a username and password, and wait for approval from the CAS system administrator. With reference to Figure 6.1, at the bottom of this picture, it is indicated that the platform can be hosted on Debian/Ubuntu machines in which the well-known Docker [322] technology is configured, while at the top, we list the software systems hosted by CAS, as microservices, and the Database engines involved. It is to be noted that, in addition to the software mentioned in Table 6.1, the following microservices have been added: CAS Logger, Gitinspector [323] and Gource [324]. Those microservices are particularly useful to professional figures logging into CAS directly to supervise the productivity of the Development Team. They are described in detail in Section 6.3. How to set-up CAS in a real or virtual machine is described in Section 6.3. We anticipate that the Docker and Ansible [325] technologies are needed.

- **CAS Client**. The CAS platform assumes that multiple developers are simultaneously involved in software development of a given project, each with their client machine, contributing to the project through incremental synchronisations of their work on the source code resident in the CAS Server. Typically, as depicted in Figure 6.1, such an external machine must be equipped with a high level programming environment such as Eclipse, Visual Studio Code, and IntelliJ IDEA, and a browser capable of connecting to the microservices provided by the CAS platform. The synchronization between client and server, referred to as activities tracking, requires the installation of appropriate plugins, and are discussed in Section 6.3. Similar to what is required for server access, for a user to synchronize her/his contributions with the CAS server, s/he must log in using their own GitLab SSO credentials.

CAS User Authentication

Since the ultimate goal is service development, and GitLab is the main component of such a collaborative development platform, we chose to enable a GitLab SSO authentication. The adoption of this solution, for user authentication, generic or



GitLab	Taiga	Mattermost	Jenkins	SonarQube	Logger	Productivity Trackers
Apache2 PostgreSQL Redis	Nginx PostgreSQL	Nginx PostgreSQL	Apache2 PostgreSQL	Apache2 PostgreSQL	Nginx MongoDB	Nginx MongoDB

Figure 6.1: **Compositional Agile System (CAS) - Architectural Stack**. In this figure, we observe the logical separation between the client and server components of the CAS platform. Each component is described in the main text.

specialized (developer), reflects the growing recognition of the importance of simplified and secure access to services, in line with the principles of Identity and Access Management (IAM) in modern organizations. GitLab SSO implementation offers numerous advantages that go beyond simplifying the authentication process. Indeed, based on an identity federation model, it allows users to authenticate through external identity providers such as OAuth 2.0 [326], Lightweight Directory Access Protocol (LDAP) [327], and Security Assertion Markup Language (SAML) [328]. This flexibility in supporting different authentication standards allows organisations to easily integrate their existing systems and make the most of previous investments in IAM infrastructure. Recent studies, e.g., [329–331], have emphasised how the adoption of an SSO authentication system can significantly reduce the risks associated with multiple credential management. This is particularly relevant considering the increase in the frequency and complexity of cyber attacks, which often exploit vulnerabilities related to password management. The implementation of GitLab

SSO not only reduces these risks but can also facilitate regulatory compliance, e.g. with the General Data Protection Regulation (GDPR) in Europe. Furthermore, the ability for GitLab to automatically create user accounts based on the information provided by the SSO provider can improve operational efficiency and reduce costs associated with manual account management. This functionality has been demonstrated in several case studies conducted on medium and large-sized organisations that reported a significant increase in efficiency in accessing services and a reduction in human errors associated with manual account management, e.g., [332]. Ultimately, the adoption of GitLab SSO improves the user experience by simplifying access to services and represents an important step towards better security and access control, in line with current best practices in IAM.

Setting Up CAS: Docker and Ansible

It is useful to recall that Ansible is a simple, but powerful, open-source IT automation tool for Linux that automates provisioning, configuration management, software application and orchestration of IT administration tasks. It allows users to automate the process of creating, configuring, and managing all the components of the Docker container stack, even on different target hosts, needed to define the desired architecture. For the interested reader, further details are available in [325]. Each CAS service is contained within a Docker [322] container, where the automation scripts are managed via Ansible from a client machine, and all of them are hosted on a target Ubuntu/Debian machine. It is worth noting that Ansible also allows the execution of scripts for creating Docker containers from the target machine if configured on it.

CAS Server Details: Microservices Architecture

The top layer in Figure 6.1 can be conveniently divided into four parts: Databases, Building Blocks, e.g., the software components listed in Table 6.1, Activities Trackers, and Productivity Trackers. The content of each of these parts is provided in Figure 6.4, together with the *intra* and *extra* information flows. We have also added a part regarding the web server. The databases are somewhat standard. The interested reader can find more details regarding PostgreSQL, Redis, and MongoDB databases in [333–335]. As for the building blocks, they need no further presentation, since they have been discussed in the previous chapter. The basic building blocks need no further presentation. Therefore, we now describe the other parts and then outline the flow of information within the CAS server.

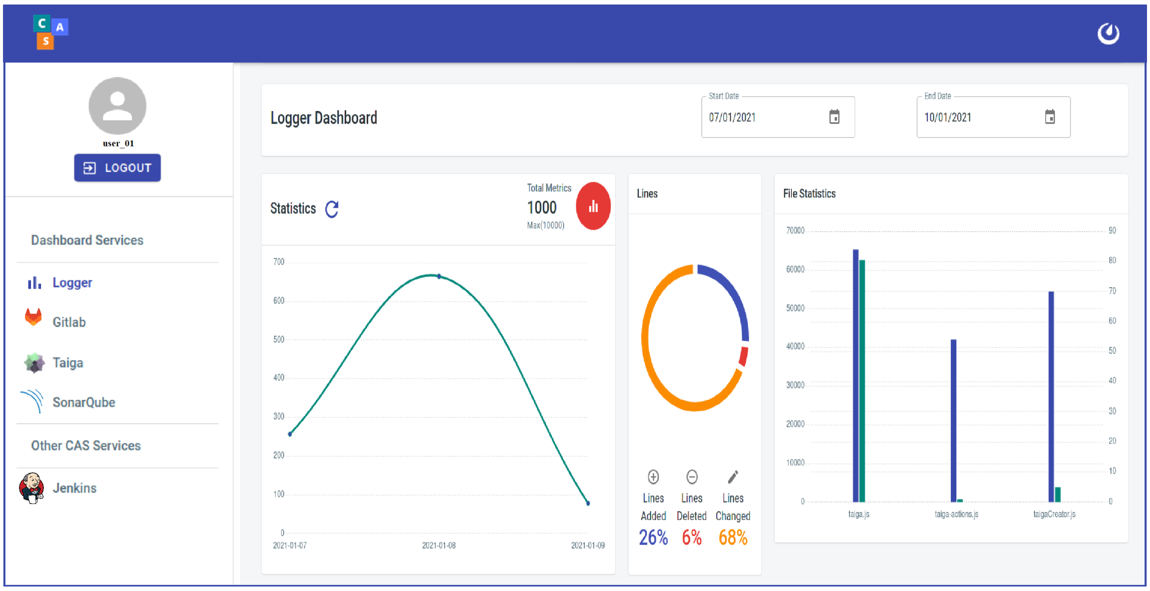
- **Web Server**. To protect network client-server communications within the CAS platform, all web traffic management, related to communication between different components within CAS, is handled through the Nginx [336] proxy manager. It is a lightweight open-source web server that can also be used as a reverse proxy, HTTP cache, and load balancer. It uses ports 80 (for HTTP), and 443 (for HTTPS) through the TLS presentation cryptographic protocol [337]. In addition, thanks to the use of the Linux tool Certbot [338], we obtain in a free way, and following the open-source philosophy, the "Let's Encrypt" [339] certificate, which enables HTTPS. Through the Linux tool *crontab* (see, e.g., Debian Manpages Documentation [340]) the certificate obtained is renewed each time 30 days after expiration (90-day duration). Finally, Nginx offers low memory consumption and high concurrency. One of its main qualities is that it does not create a new process for each web request, but uses an asynchronous, event-driven approach in which requests are handled by a single thread, proving to be a robust solution capable of offering stability to the CAS system.

- **Activities Trackers: The CAS Logger**. Server side, client synchronization is accomplished in this component. It is a self-tracking service that enables platform users to track their actions during code-writing activities in their preferred high level programming environment and store these information in a Mongo database.

  To use this service, users should first install the logger plugins into their environments. After this operation, users can start the logger plugin in their environments, and view programming activity statistics in detail, server-side, through the homologous component in the CAS Logger Dashboard, i.e., a web application that, for each developer user, reads from a Mongo database mentioned earlier the data collected on their client machine and displays it in aggregate form. This application is convenient because it allows users to view all user activity data in one place.

  With reference to Figure 6.2, this application allows users to view the information contained in the Mongo database mentioned earlier, intending to obtain a history of the activities performed by developers within the editor used to write lines of code. It is worth noting that the plugins installed on the developer clients do not interfere with the activities of the users, but are only concerned with keeping track of the operations performed, in terms of lines of code added, updated and deleted, whether they refer to comments, test or code refactoring.

  We provide below some technical details about the technologies used for the building of the CAS Logger Dashboard. These include the Javascript framework [341], such as React [342] and Redux [343], which were used to ensure modularity and flexibility. As depicted in the left part of Figure 6.2, within the dashboard there are a set of sections that partition the data from the individual microservices, which communicate through the APIs they provide. The data is then processed through the use of the ChartJS library [344], to return tables or charts (see right part of Figure 6.2).



Start Date 07/01/2021

End Date 10/01/2021

Figure 6.2: **CAS Logger Dashboard**. We can conveniently divide this figure into two sections: left and right. In the left section, the developer can logout from the dashboard or select the service whose aggregated data s/he wants to visualize (see the main text). In the right section, in the upper part, the user can choose the time interval of reference, while in the lower part, s/he visualizes in a graphical format her/his activities performed during its development working sessions (see the main text).

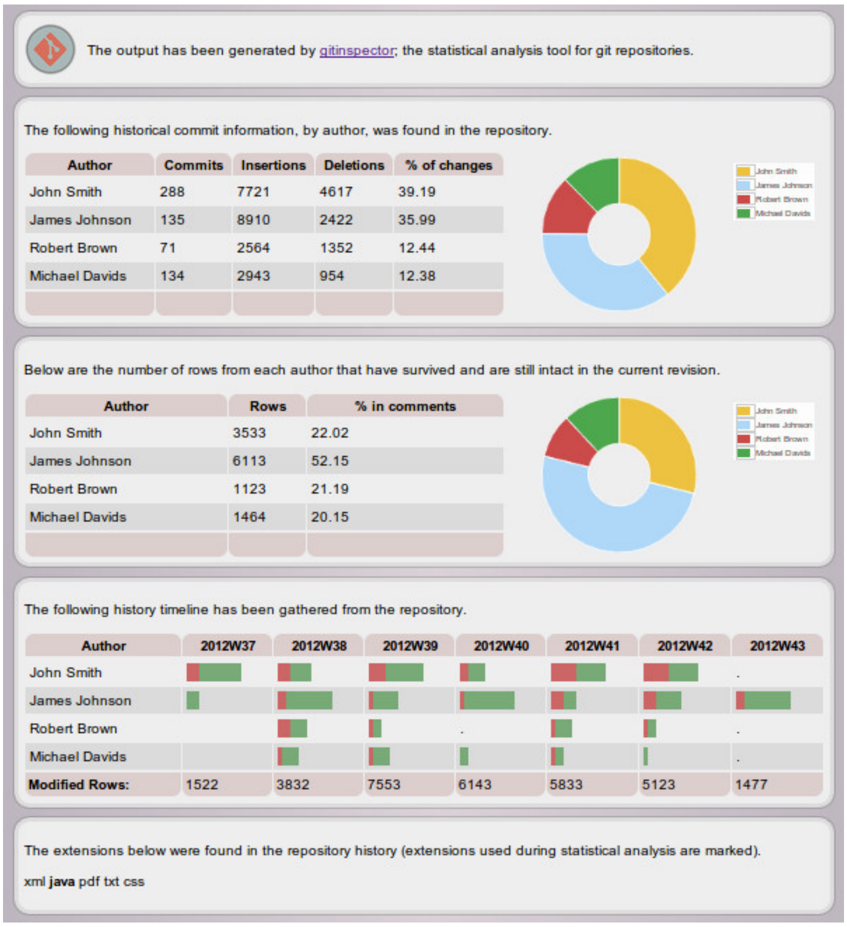
- **Productivity Trackers**. The productivity trackers that are integrated into CAS, to monitor the productivity of the development team, are described below.
  - **Gitinspector**. It is an open-source powerful statistical analysis tool specifically designed for Git repositories. Initially, it was developed to help retrieve repository statistics from student projects of the Object-oriented Programming Project course (TDA367/DIT211) at Chalmers University of Technology and Gothenburg University. Currently, Gitinspector is used by universities all over the world as an evaluation tool. It allows users to examine the source code of a project, providing a detailed overview of code metrics, e.g., lines of code, number of commits, and cyclomatic complexity. This analysis can provide a clear indication

of the quantity and quality of work accomplished by the team and makes it possible to assess the overall quality of the work performed and to identify possible areas for improvement in terms of code simplification and optimization. Users can effortlessly generate overall statistics based on the work of the developers and combine them with a timeline analysis that demonstrates the activities of the developers and their workload. In addition, this tool allows users to filter the results by choosing a specific number of extensions, and by default, it only includes source files in the statistical analysis. It can identify specific code patterns and provide metrics that can help identify potential problems or areas requiring optimization. This can help the team identify and resolve bugs promptly, reducing the risk of repetitive errors. It can provide detailed information on the activities of team members, including individual contributions, commit times and code changes. This can help team leaders assess the effectiveness and efficiency of each team member's work.

Additional details, including its options, are available in [345]. Figure 6.3 shows a small example of a report generated by this tool.

- **Gource**. It is a cross-platform open-source visualization tool for source control repositories. It can offer an innovative way to track the development process of a software project over time with its visual representation. Its visual representation can provide valuable insights into the activities of a development team, such as intuitive visualisation, changes identification, development patterns analysis and team involvement. The repository is displayed as a tree where the root of the repository is the center, directories are branches and files are leaves. Source code contributors, through their user icon, appear and disappear as they contribute to specific files and directories. This representation allows a certain degree of interaction. Indeed, it allows users to click on leaves to identify the corresponding code file in the repository. The visualisation of Gource works with OpenGL library [346] and requires a 3D-accelerated video card to perform. However, it is important to note that, considering the type of output generated by Gource, this tool may not offer detailed data to specifically assess team productivity. For a complete evaluation of productivity, it may be necessary to integrate Gource with other tools for analysing development activities, e.g., Gitinspector. Our current contribution, within the CAS platform, is to offer the combination of these two tools that are present under the Productivity Tracker section (see Figure 6.1), also referred as to Git Dashboard.

  Additional details, including its options, are available in [347].



The output has been generated by gitinspector; the statistical analysis tool for git repositories.

The following historical commit information, by author, was found in the repository.

Author	Commits	Insertions	Deletions	% of changes
John Smith	288	7721	4617	39.19
James Johnson	135	8910	2422	35.99
Robert Brown	71	2564	1352	12.44
Michael Davids	134	2943	954	12.38

Below are the number of rows from each author that have survived and are still intact in the current revision.

Author	Rows	% in comments
John Smith	3533	22.02
James Johnson	6113	52.15
Robert Brown	1123	21.19
Michael Davids	1464	20.15

The following history timeline has been gathered from the repository.

Author	2012W37	2012W38	2012W39	2012W40	2012W41	2012W42	2012W43
John Smith	🟥🟩	🟥🟩	🟥🟩	🟥🟩	🟥🟩	🟥🟩	.
James Johnson	🟩	🟥🟩	🟥🟩	🟥🟩	🟥🟩	🟥🟩	🟥🟩
Robert Brown		🟥🟩	🟥🟩	.	🟥🟩	🟥🟩	.
Michael Davids		🟥🟩	🟥🟩	🟩	🟥🟩	🟩	.
Modified Rows:	1522	3832	7553	6143	5833	5123	1477

The extensions below were found in the repository history (extensions used during statistical analysis are marked).

xml **java** pdf txt css

Figure 6.3: **Gitinspector Productivity Tracker**. This figure shows an abstract of a report generated by this tool.

- **Information Flows**. Given the open-source nature of the software tools in CAS, the various microservices, each with its docker container that includes local storage, communicate with each other by exchanging data through the use of application communication interfaces (APIs). These latter are independent of the programming language used, each developed and made available by the company or community maintaining the project, and major advantages include, e.g., shorter development cycles; greater scalability; independent services; ease of deployment; accessibility; and greater openness. For this reason, the reader is referred to the documentation sections of the relevant Web portals (see, e.g., GitLab Documentation [296]). The directionality of the information flow is indicated by the direction of the arrows in Figure 6.4.

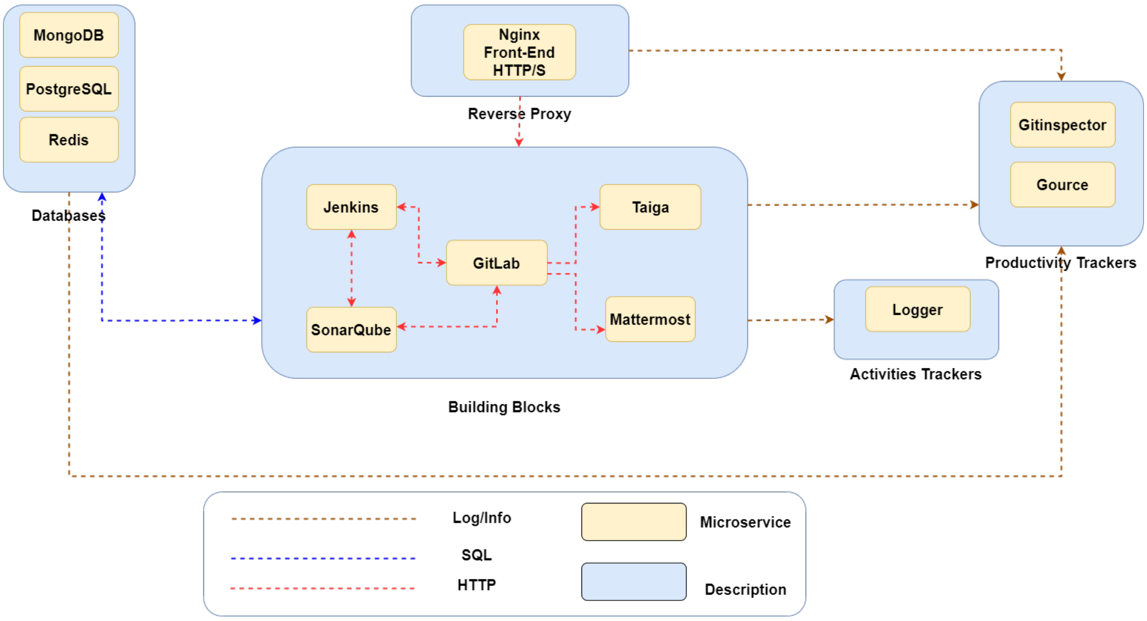


Figure 6.4: **The Graphical Perspective View of the CAS Architecture**. In this figure, we would follow the below notation. The yellow bricks represent the microservices and databases involved. The light blue bricks provide a brief description of the services within them. The directional arrows represent the information flows between the various services within the CAS platform, via dedicated APIs. We logically divide this figure into four parts. Specifically, Databases, Building Blocks (see Table 6.1), Activities Trackers, and Productivity Trackers.

CAS Clients: Synchronization with the Server

Once logged in, the developer to be able to perform commit operations on the source code repository, either through his development environment or by using the git [295] tool from the command line, needs an add-on that allows her/him to record, server-side, the number of lines of code s/he has inserted, modified, or deleted in its development sessions. This component is to be installed through the Plugins section on the development environment used by the developer and has been developed for major high level programming environments such as Eclipse, Visual Studio Code and IntelliJ IDEA. These are available in [348]. These plug-ins can be activated by the developer from their programming environments, at each work session, and once done, they connect to the homologous CAS Server service (CAS Logger) to store the above information in a Mongo database [335]. This activities tracker, together with updating the plugins, represents our contribution to the successful growth of the CAS platform.

6.4 A Comparison of CAS with the State of the Art

In this section, we compare the CAS platform with the State of the Art about comparable, or in some ways complementary, commercial and open-source systems. When comparing these systems, we would like to discuss the major advantages of adopting the CAS platform, regarding what is available in the basic plans of commercial tools and similar open-source tools focused on collaborative development. The CAS platform offers a key advantage for the PAs, enabling them to rationalize costs over the expensive software licenses required by commercial tools. In addition, CAS provides Single Sign-On authentication for users using GitLab credentials. Significantly, user credentials are stored in the storage volume of the GitLab service within the infrastructure perimeter of the organization hosting the platform. This feature provides greater security than commercial tools that usually run in the cloud, where users must trust the good faith of the provider. When we compare CAS with commercial tools such as Atlassian's products, in their free versions, e.g., Jira, Confluence, Trello, and BitBucket, we observe that these offer the possibility of working with teams of up to ten users. CAS, meanwhile, in terms of team size of professional figures involved in the project (users), places no limitation, which is shown to be a point in its favour.

GitLab Pages and Wiki, with the integration of Draw.io, offers a good alternative to Confluence's free version for creating high quality documentation of software products, without having to use external reporting tools that usually require payment. Unlike Confluence's free version, CAS does not limit the number of users who can participate in a project and allows for the management of user permission, a feature that is typically available only in paid versions. By allowing developers to create and update documentation in real time within the development platform, compliance with data regulations such as the European Data Protection Act (GDPR) can be assured, which is especially important for organizations that prioritize data security and privacy like the PAs. This feature within the CAS platform ensures greater consistency between the documentation and code, allowing for documentation to be updated in real time with incremental product developments. GitLab's versioning feature allows for collaboration between team members, enabling them to work simultaneously on both documentation and code, improving communication and information sharing. The objects created with Draw.io, within GitLab, are versioned, making it easy to track changes and revert to previous versions if needed.

When we compare CAS with similar solutions, the Brazilian government has provided its community with an integrated platform for collaborative development [203], in which it offers integration of some services through the open-source Colab component [349]. Colab acts as an intermediary among all services, providing the interface, authentication, and integration. Whenever a specific service is requested, Colab authenticates the user in the target tool, sends the request, and turns the HTML page into a visually appealing response. Colab has a built-in search engine that allows users to search the entire database for what they need on the platform. In addition, Colab provides a Web interface to GNU Mailman [350], for team communication, along with two other integrated tools, i.e., GitLab and Prezento [351]. GitLab provides a web interface for Git repositories and issue tracking, while Prezento is a front-end for static source code analysis. Mezuro [352] performs static analysis tools on source code stored in a repository and provides the data to Prezento. Noosfero [353] provides a social network and content management system (CMS) features. The Brazilian government insisted on using Colab, despite technical reasons to the contrary. The team had to completely rewrite it, resulting in an increased budget and project complexity. For more details, see [203]).

6.5 CAS for Scrum in the PA

In this section, we first specialize the requirements that we have seen in Section 6.2 to some professional figures afferent to our Scrum@PA proposal. Then, we show how the CAS platform can also work as quality control on existing working projects. To this end, taking a real project developed by the Italian PA as an example, we show some workflows on the CAS platform related to this specific case study. The next two subsections are devoted to this topic.

A Specialization of Scrum for the PA

About the user stories related to professional figures provided in the standard Scrum methodology, which are listed in Section 6.2, here we have included some relevant user stories for the leader professional figures that are included in our Scrum@PA proposal. Then, in analogy to Section 6.2, we report in Table 6.2 the mapping of the latter to some open source tools, present within the CAS platform, and some IDE plugins that are configurable in the client of the developers.

- **As a Scrum Coach**.

  9. I need some tool support to constantly keep track of the professional skills of the Development Team in order to be able to coordinate it and activate the related Pair Programming process among the developers. This corresponds to activities 1.(b), 6.(a)-6.(b), and 7.(b), mentioned in Section 6.2. Consequently, the tools to be used are GitLab, Taiga, and Draw.io. Since the SC, in the context of this activity, needs to interact

with other professional figures, i.e., the SM, and the Developer, these tools would also be available to them.

- **As a Contract Responsible**.

10. I need some tool support for monitoring and ensuring compliance with various contract requirements, functional and non-functional, including billing, administrative, and management of any changes to them. This corresponds to activities 1.(b), 3.(a)-3.(b), and 6.(a)-6.(b), mentioned in Section 6.2. Consequently, the tools to be used are GitLab, Taiga, Mattermost, and Draw.io. Since the RC, in the context of this activity, needs to interact with other professional figures, i.e., the RS, and the PO, these tools would also be available to them.

- **As a Service Responsible**.

11. I need a support tool to define and monitor service level agreements, and requirements, during the building phases of the software product. This corresponds to activities 1.(b), 3.(a)-3.(b), and 6.(a)-6.(b), mentioned in Section 6.2. Consequently, the tools to be used are GitLab, Taiga, Mattermost, and Draw.io. Since the RS, in the context of this activity, needs to interact with other professional figures, i.e., the RC, and the PO, these tools would also be available to them.

From these user stories and the tools that we indicated and chose in the previous chapter, the following correspondence emerges, which we show in Table 6.2. It is useful to highlight to the reader how in the first column of Table 6.2 the numbering of the user stories starts from the next integer following the one in Table 6.1.

Table 6.2: **Mapping User Stories and main Scrum@PA Agile Roles to Some Open-Source Tools**. The table legend is as in Table 6.1.

#	USER STORY	TOOLS	SCRUM@PA ROLES
U.S. 9	Create and Update the Competencies Matrix and other Project Documentation	GitLab, Taiga, and Draw.io.	SC, SM, Developer.
U.S. 10	Monitoring the Compliance with Various Contract Requirements	GitLab, Draw.io, Taiga, and Mattermost.	RC, RS, PO.
U.S. 11	Monitoring the Requirements and the Service Level Agreements	GitLab, Draw.io, Taiga, and Mattermost.	RS, RC.

CAS as Quality Control of a Public Administration Service: IO App

The CAS Platform can be used also for the quality verification of a software product. To this end, we show, through this platform, how this can be achieved in the PA domain by taking as a reference a specific service, namely IO App. It is the native mobile application for iOS and Android of the Digital Citizenship project [354], developed by the PagoPA S.p.A. and some volunteers who support the project, and has a dual purpose:

- to be an interface for citizens to manage their data and their digital citizen profile
- to act as a reference implementation of the integrations with the Digital Citizenship platform

It was developed from February 2017 to November 2023, and it is licensed under the European Union Public License 1.2 [355]. It is currently maintained and consists of more than 195K lines of code, and more than 5K commit operations, that have been performed on 75 different code branches.

Here, in the three following subsections, we show some aspects related to the quality control of a service, i.e., Static Code Quality Analysis, Team Productivity Metrics, and Source Code Testing. We anticipate that, having not been part of the project, the information we gather is partial, yet perceived as very useful to demonstrate the qualities of CAS.

Static Code Quality Analysis

First, we imported a new project into a dedicated GitLab repository that we referred to as “io-app”. Then, through the use of Continuous Integration tools found in CAS, i.e., GitLab and Jenkins, we created a process automation pipeline. In this scenario, we chose to use the GitLab Runners, which were activated during the platform setting up phase (see the CAS GitHub repository guideline [320]). Within SonarQube, we selected the io-app mentioned earlier, and then, we chose to run the setup of a new code analysis. Among the available options, the *build* option “Other (for JS, TS, Go, Python, PHP, ...)” was selected, and we followed the wizard suggested by the tool itself.

As a result of this operation, a yaml format file [356], referred to as *gitlab-ci.yml*, and a textual file, referred to as *sonar-project.properties* are generated in the root of the repository. We show below the *gitlab-ci.yml* file.

```
sonarqube-check:
  image:
    name: sonarsource/sonar-scanner-cli:latest
    entrypoint: [""]
  variables:
    SONAR_USER_HOME: "${CI_PROJECT_DIR}/.sonar"
    # Defines the location of the analysis task cache
    GIT_DEPTH: "0"  # Tells git to fetch all the branches
    of the project, required by the analysis task

  cache:
    key: "${CI_JOB_NAME}"
    paths:
      - .sonar/cache
  script:
    - sonar-scanner
  allow_failure: true
  only:
    - master
```

According to the intrinsic properties of the io-app project, we excluded from this analysis all *.java* files, recursively from all sub-folders of the repository, being that SonarQube needs the compiled *.class* files, not present within it. This customization was performed within the *sonar-project.properties* file, which we show below (see the third line of code).

```
sonar.projectKey=gitlab-instance-(#app_token)
sonar.qualitygate.wait=true
sonar.exclusions=**/*.java
```

To enable the generation of static software analysis reports on the IO project, that is quite large, we modified a setting within the Nginx configuration files, allowing the generation of reports with a maximum file size of 200 MB.

Now, we show how the use of SonarQube can report bugs, vulnerabilities, security hotspots, and provide an assessment of the Technical Debt.

- **Bugs**. A bug represents a functional software defect that leads to software malfunction or serious performance deficiencies. It, for example, produces an unexpected or incorrect result, typically due to an error in writing the source code of a program. SonarQube's analysis of the IO app source code reports

16 potential bugs, grouped into four categories, such as “Code unreachable”, “Convert the conditional to a boolean to avoid leaked value”, “Review this usage of “...” as it can only be empty here”, and “Throw this error or remove this useless statement”.

- **Vulnerabilities** represent flaws present in a code and are often used by malicious agents to gain unauthorized access to networks, steal valuable and sensitive data, and compromise the systems of an organization. SonarQube’s analysis reports only one potential vulnerability.

- **Security Hotspots**. They represent security-sensitive portions of code that require manual review to assess whether or not a vulnerability exists. Sonarqube’s analysis reports as many as 262 security hotspots ranked by priority: high (5 for authentication and 4 for Command Injection), medium (3 for Denial of Service (DoS) and 7 for Weak Cryptography) and low (24 for Encryption of Sensitive Data and 219 for Others).

- **Technical Debt**. In software development and other IT fields, it is the cost, in terms of time, of future rework when a limited solution is chosen over a better option that may take longer. The SonarQube analysis suggests a 20-day timeline to address a wide range of potential improvements by rewriting 1,350 unique code segments.

After the above information, SonarQube offers a section, referred to as “Activity Section”. The information contained therein, such as code progress on issues, code coverage, and percentage of duplicate blocks, would have been very useful during the different Sprints of the project development cycle. In fact, the Activity Section, associated with the output of the analysis reported above, allows the measurement of improvements resulting from code changes. Unfortunately, since we do not have the availability of intermediate versions of this master branch of the repository, we could not display this information. Such analysis could be very useful since, Sprint after Sprint, it allows the development team to provide increasingly robust and maintainable code increments over time, while simultaneously allowing a reduction in the costs associated with evolutionary maintenance of the implemented service.

After analyzing the source code, we draw some insights into the static analysis of SonarQube. As for bugs, the reliability rating of SonarQube is C, which means that there is at least one major bug. In particular, many of them concern the conversion of conditional expressions to Boolean variables to avoid errors. As for vulnerabilities, the reliability rating of SonarQube is D, which means that there is at least one critical vulnerability is present. In particular, it concerns enabling server certificate validation on SSL/TLS connections, which is essential for creating secure SSL/TLS sessions that are not vulnerable to man-in-the-middle attacks. As for security hotspots, the reliability rating of SonarQube is E, which means that there are less than 30% of security hotspots that are reviewed. As mentioned earlier, there are four high priority potential Command Injection security issues. For these latter, developers may secure Linux command execution within scripts to avoid exposing them to "Command Injection" security vulnerabilities. Moreover, there are ten medium-priority potential issues. For these latter, developers could secure a less vulnerable use of regular expressions at run-time, which due to backtracking can lead to denial of service (DoS), and secure a safer use of the pseudo-random number generator. As for technical debt, although 1,350 code smells were identified in the Technical Debt section of Sonarqube's analysis for a proposed refactoring of them, the maintainability rating is A, meaning the technical debt ratio is less than 5.0%. Again, although this rating is synonymous with code robustness, if the developers were following, Sprint after Sprint, a static quality check on the code, they would surely have achieved, in the last Sprint, even more robust code.

Team Productivity Metrics

The Gitinspector report shown in Figures 6.5-6.9 provides various information on the contributions of the development team members involved in the development of the project, in terms of commit operations, lines of code added, modified and deleted, during the entire development period divided into weekly time intervals, and the main files for which the developers are chiefly responsible.

In particular, Figure 6.5 contains, for each author, historical information about the number of commits performed on the repository, specifically highlighting the number of lines of code added, deleted, and modified, with some information about the stability and the percentage of comments of the code produced.

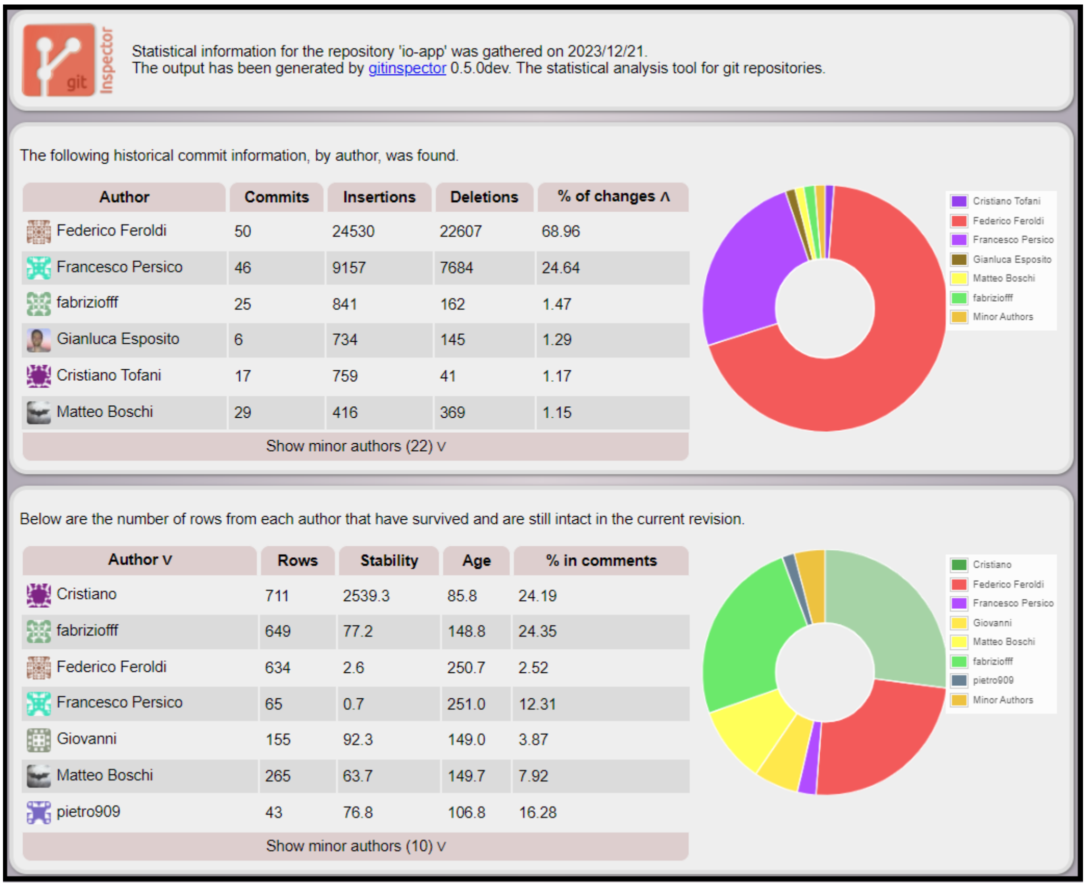
For each author involved in the development, Figures 6.6-6.7 provide an abstract, respectively at the beginning (February 2017) and at the end of the development cycle (November 2023), of their contributions during each weekly period. Each abstract includes a horizontal bar representing the percentages of lines added and deleted, as well as the number of updated lines.

Figure 6.8 contains information on suspiciously large files and those that have high cyclomatic complexity, listing them in order of severity.

Figure 6.9 contains information about the files for which each developer is primarily responsible, indicating for each one the number of lines of code added, and excluding from this count those lines of code that represent comments, if possible.

The analysis was performed on a subset of extensions, as shown in bold in Figure 6.10, i.e., those most impacting the source code, e.g., java, c, cc, cpp, h, hh, hpp, py, glsl, rb, js, and sql.

Finally, in Figure 6.11, we show a snapshot of the video, obtained from Gource, at the last contributions of the development team, showing, in tree form, the final



Statistical information for the repository 'io-app' was gathered on 2023/12/21.
The output has been generated by gitinspector 0.5.0dev. The statistical analysis tool for git repositories.

The following historical commit information, by author, was found.

Author	Commits	Insertions	Deletions	% of changes ∧
Federico Feroldi	50	24530	22607	68.96
Francesco Persico	46	9157	7684	24.64
fabriziofff	25	841	162	1.47
Gianluca Esposito	6	734	145	1.29
Cristiano Tofani	17	759	41	1.17
Matteo Boschi	29	416	369	1.15
Show minor authors (22) ∨				

Below are the number of rows from each author that have survived and are still intact in the current revision.

Author ∨	Rows	Stability	Age	% in comments
Cristiano	711	2539.3	85.8	24.19
fabriziofff	649	77.2	148.8	24.35
Federico Feroldi	634	2.6	250.7	2.52
Francesco Persico	65	0.7	251.0	12.31
Giovanni	155	92.3	149.0	3.87
Matteo Boschi	265	63.7	149.7	7.92
pietro909	43	76.8	106.8	16.28
Show minor authors (10) ∨				

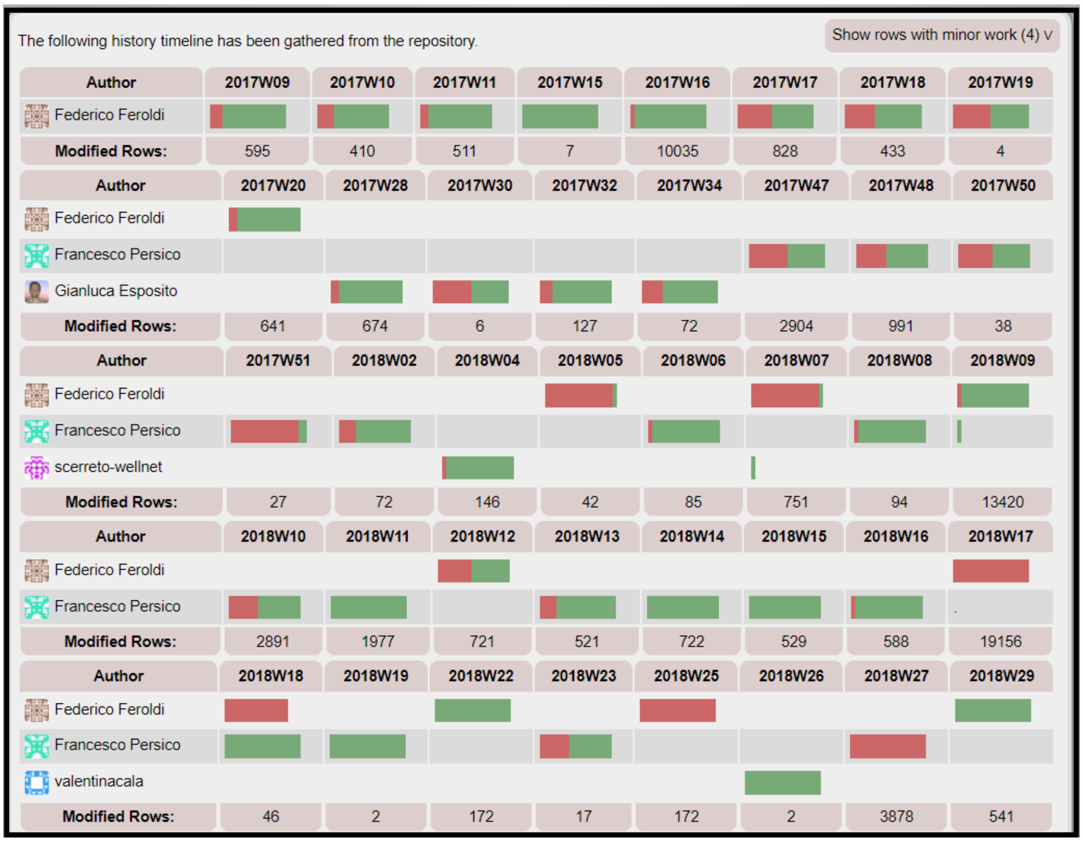
Figure 6.5: **Gitinspector Report Abstract on IO App Project - Historical Information on Commit Operations**. The Figure legend is as in the main text.

repository of the project.

In addition to the static analysis of software quality provided by SonarQube, the observation of certain metrics, such as those found in Gitinspector, makes this productivity analysis very interesting, as it allows for tracking the contributions of the various developers, step by step, throughout the entire development time frame.

From the analysis of the Gitinspector report, some interesting information emerges about the work of developers, such as establishing:

- which of them contributed most to the development of the project;
- the stability and degree of documentability of the code they produced;
- how much work, each of them has produced each week, over a given period;
- files for which each of them is primarily responsible, in terms of their contribution.



The following history timeline has been gathered from the repository.

Show rows with minor work (4) v

Author	2017W09	2017W10	2017W11	2017W15	2017W16	2017W17	2017W18	2017W19
Federico Feroldi	🟥🟩	🟥🟩	🟥🟩	🟩	🟩	🟥🟩	🟥🟩	🟥🟩
Modified Rows:	595	410	511	7	10035	828	433	4

Author	2017W20	2017W28	2017W30	2017W32	2017W34	2017W47	2017W48	2017W50
Federico Feroldi	🟥🟩							
Francesco Persico						🟥🟩	🟥🟩	🟥🟩
Gianluca Esposito		🟥🟩	🟥🟩	🟥🟩	🟥🟩			
Modified Rows:	641	674	6	127	72	2904	991	38

Author	2017W51	2018W02	2018W04	2018W05	2018W06	2018W07	2018W08	2018W09
Federico Feroldi				🟥🟩		🟥🟩		🟥🟩
Francesco Persico	🟥🟩	🟥🟩			🟥🟩		🟥🟩	🟩
scerreto-wellnet			🟥🟩			🟩		
Modified Rows:	27	72	146	42	85	751	94	13420

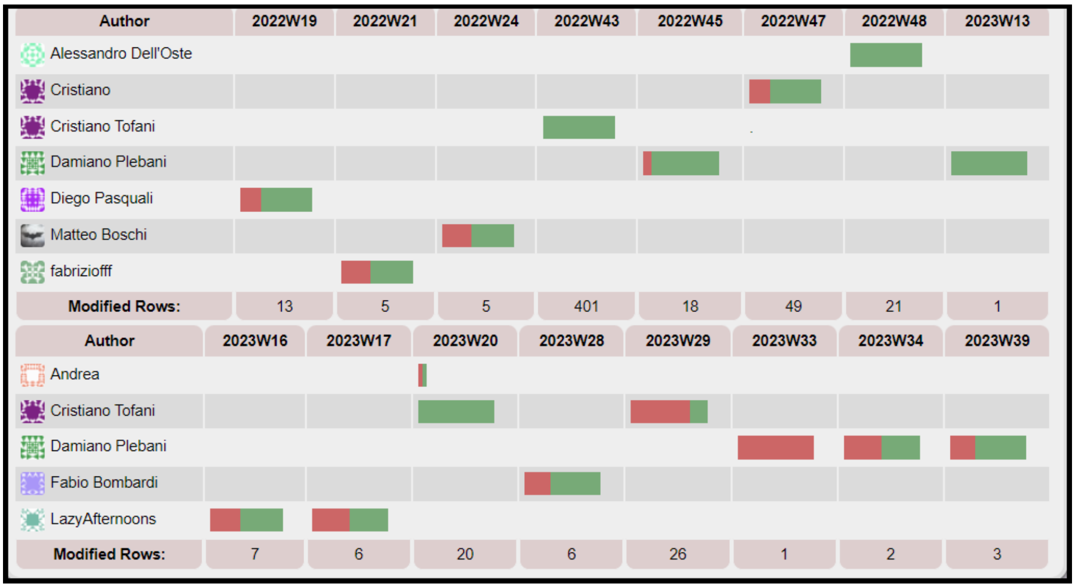
Author	2018W10	2018W11	2018W12	2018W13	2018W14	2018W15	2018W16	2018W17
Federico Feroldi			🟥🟩					🟥
Francesco Persico	🟥🟩	🟩		🟥🟩	🟩	🟩	🟥🟩	.
Modified Rows:	2891	1977	721	521	722	529	588	19156

Author	2018W18	2018W19	2018W22	2018W23	2018W25	2018W26	2018W27	2018W29
Federico Feroldi	🟥		🟩		🟥			🟩
Francesco Persico	🟩	🟩		🟥🟩			🟥	
valentinacala						🟩		
Modified Rows:	46	2	172	17	172	2	3878	541

Figure 6.6: **Gitinspector Report Abstract on IO App Project - Developer Contributions at the Beginning Period of the Development Cycle**. The Figure legend is as in the main text.

Source Code Testing

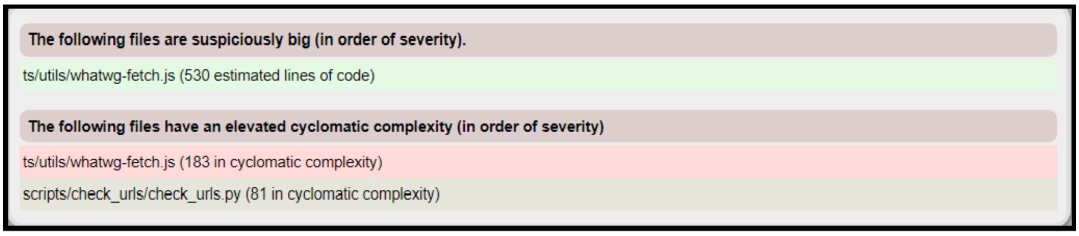
We also could have run the tests on different features in the source code. The authors of the IO App project, in the original [321] repository, while providing directions to replicate the operating environment, have officially stated that the tests fail in their repository as well. Analyzing the source code, we observed that the development team used the Java Jest library [357] to test several features of the project. However, Jest requires additional customizations based on the operating environment for which the IO application is targeted, i.e., Android and iOS. Considering the insufficient information available for the source code and the scope of this demonstration, we concluded that it was not appropriate to create any specific GitLab runner to conduct a testing strategy, as we would not gain any useful information to present to the readers.



Author	2022W19	2022W21	2022W24	2022W43	2022W45	2022W47	2022W48	2023W13
Alessandro Dell'Oste							🟩	
Cristiano						🟥🟩		
Cristiano Tofani				🟩				
Damiano Plebani					🟥🟩			🟩
Diego Pasquali	🟥🟩							
Matteo Boschi			🟥🟩					
fabriziofff		🟥🟩						
Modified Rows:	13	5	5	401	18	49	21	1

Author	2023W16	2023W17	2023W20	2023W28	2023W29	2023W33	2023W34	2023W39
Andrea			🟥🟩					
Cristiano Tofani			🟩		🟥🟩			
Damiano Plebani						🟥	🟥🟩	🟥🟩
Fabio Bombardi				🟥🟩				
LazyAfternoons	🟥🟩	🟥🟩						
Modified Rows:	7	6	20	6	26	1	2	3

Figure 6.7: **Gitinspector Report Abstract on IO App Project - Developer Contributions at the End Period of the Development Cycle**. The Figure legend is as in the main text.



The following files are suspiciously big (in order of severity).

ts/utils/whatwg-fetch.js (530 estimated lines of code)

The following files have an elevated cyclomatic complexity (in order of severity)

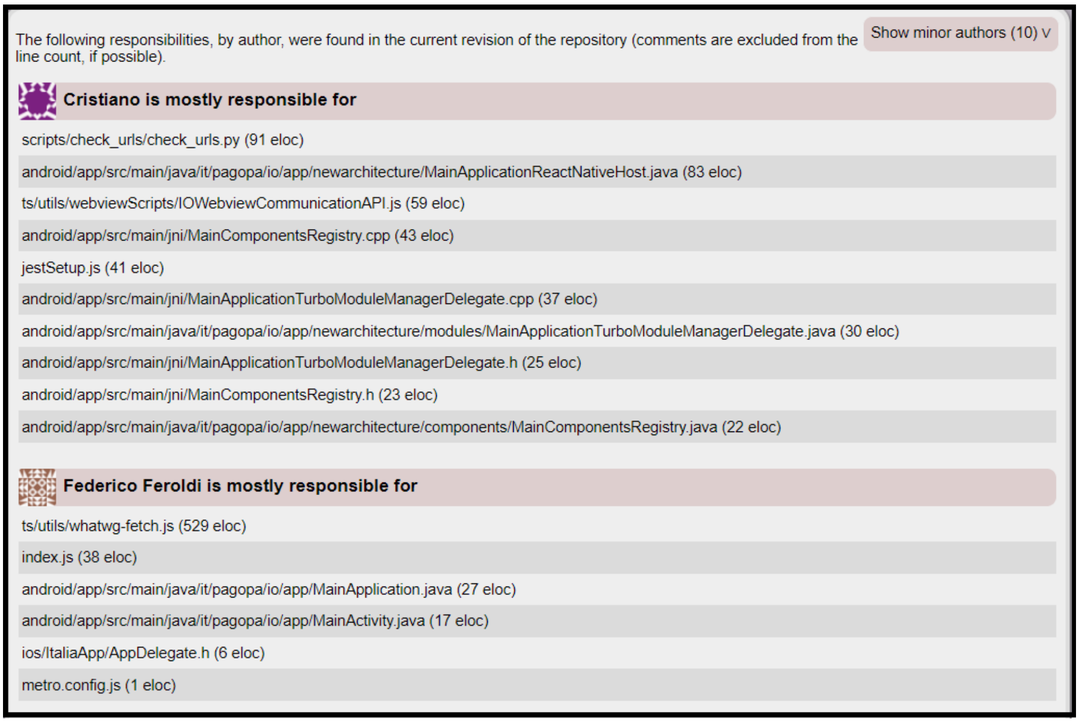
ts/utils/whatwg-fetch.js (183 in cyclomatic complexity)

scripts/check_urls/check_urls.py (81 in cyclomatic complexity)

Figure 6.8: **Gitinspector Report Abstract on IO App Project - Suspiciously Large and High Cyclomatic Complexity Files**. The Figure legend is as in the main text.

6.6 Conclusions

In this chapter, we presented the CAS Platform, as a modular system that can easily accommodate additional components. First, we discussed the requirements for designing such an IDE, and recommended Agile tools for its implementation, arguing how they can be successfully adopted, first by the professional figures within the standard Scrum methodology, and then by the specialized professional figures present within our proposed Scrum@PA Agile methodology. Then, we provided high level details of CAS, and more information on their aspects related to the authentication, configuration, architecture, and client-side synchronization of developer contributions. After this, we have compared the CAS platform with State of the



Show minor authors (10) ∨

The following responsibilities, by author, were found in the current revision of the repository (comments are excluded from the line count, if possible).

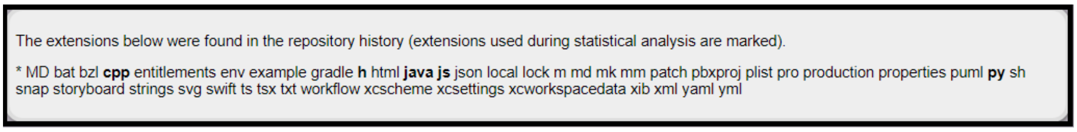
Cristiano is mostly responsible for

scripts/check_urls/check_urls.py (91 eloc)
android/app/src/main/java/it/pagopa/io/app/newarchitecture/MainApplicationReactNativeHost.java (83 eloc)
ts/utils/webviewScripts/IOWebviewCommunicationAPI.js (59 eloc)
android/app/src/main/jni/MainComponentsRegistry.cpp (43 eloc)
jestSetup.js (41 eloc)
android/app/src/main/jni/MainApplicationTurboModuleManagerDelegate.cpp (37 eloc)
android/app/src/main/java/it/pagopa/io/app/newarchitecture/modules/MainApplicationTurboModuleManagerDelegate.java (30 eloc)
android/app/src/main/jni/MainApplicationTurboModuleManagerDelegate.h (25 eloc)
android/app/src/main/jni/MainComponentsRegistry.h (23 eloc)
android/app/src/main/java/it/pagopa/io/app/newarchitecture/components/MainComponentsRegistry.java (22 eloc)

Federico Feroldi is mostly responsible for

ts/utils/whatwg-fetch.js (529 eloc)
index.js (38 eloc)
android/app/src/main/java/it/pagopa/io/app/MainApplication.java (27 eloc)
android/app/src/main/java/it/pagopa/io/app/MainActivity.java (17 eloc)
ios/ItaliaApp/AppDelegate.h (6 eloc)
metro.config.js (1 eloc)

Figure 6.9: **Gitinspector Report Abstract on IO App Project - Files for Which Each Developer is Largely Responsible**. The Figure legend is as in the main text.



The extensions below were found in the repository history (extensions used during statistical analysis are marked).

* MD bat bzl **cpp** entitlements env example gradle **h** html **java** **js** json local lock m md mk mm patch pbxproj plist pro production properties puml **py** sh snap storyboard strings svg swift ts tsx txt workflow xcscheme xcsettings xcworkspacedata xib xml yaml yml

Figure 6.10: **Gitinspector Report Abstract on IO App Project - Extensions Search Filter**. The Figure legend is as in the main text.

Art of commercial and similar open-source systems. This comparison highlighted several strengths of our platform compared to the proposed solutions, which encouraged further improvements to our platform. Finally, to demonstrate the validity of the project, we conducted an ex-post analysis on the mobile *IO App*, showing how the CAS Platform may be able to evaluate the static code quality analysis of an already developed software product, also allowing stakeholders to measure the goodness of the work accomplished by the development team.

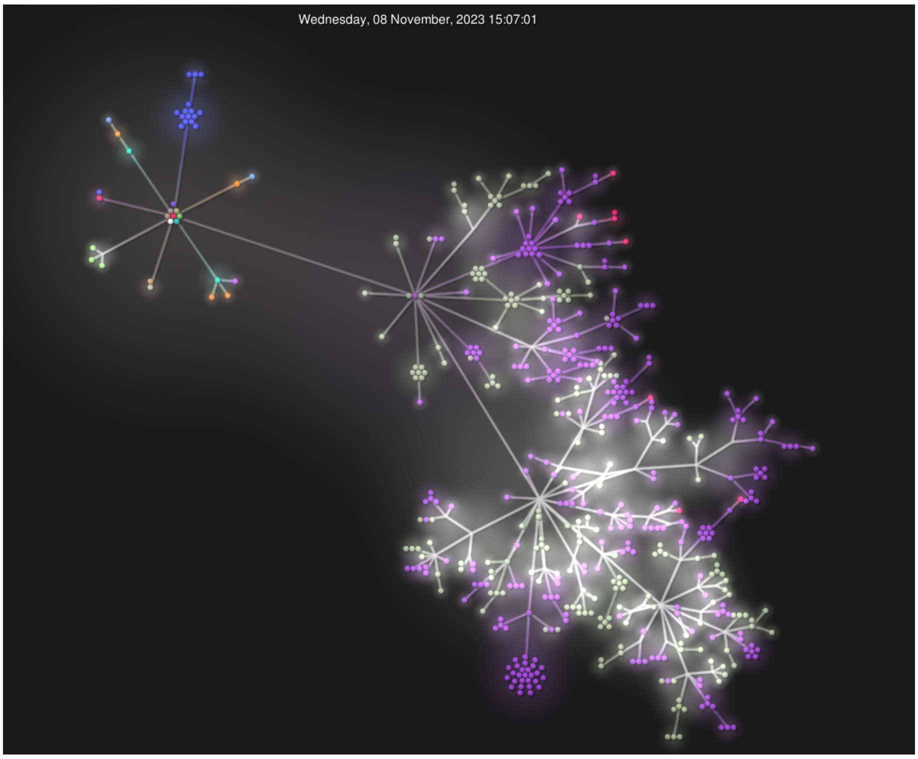


Figure 6.11: **Snapshot of the Gource Output**. This figure represents the final snapshot of the IO App project repository, represented in tree form by the Gource service. At the top of this figure, shows the date when the development team performed the last commit operation.

Part IV

Conclusions And Open Problems

Chapter 7

Conclusions and Open Problems

Abstract

This chapter summarizes the main findings of the research, covered by this Dissertation, and draws initial conclusions.

Here, in Sections 7.1-7.2, we first summarize the major contributions that we have presented in this Dissertation, and then we identify some possible research future directions.

7.1 Contributions of this Dissertation

There are three main contributions to this research. Below is a summary of them.

The first contribution of this research produced a significant result, which is a general and practical model for managing the complex Digital Transformation strategy planning process from a Computer Science perspective. This model can be used by public stakeholders to streamline their processes. Starting from the first contribution, the second of these is that it is possible to provide a general model for the specialization of Scrum in Agile project management and software development for PA, in the context of DT. This key finding opens the way for further research directions, which we have now outlined and indicated in Section 7.2. This approach provides new professional roles, particularly suited to PA, that are not available in traditional Scrum methodology. In particular, we provided a taxonomy of these. Then, we described the Scrum@PA framework and its components, including Leader professional figures, Teams and their main interactions. Moreover, to demonstrate the validity of our proposal, we showed that the Scrum@IMI framework, used for the DT of the city of Barcelona, can be obtained as a special case of our Scrum@PA methodology. Finally, the third and last of the contributions of this research, intimately related to the other mentioned contributions, regarding the provision of an Agile and collaborative toolset for Leader professional figures and Teams afferent to

Scrum@PA. To this end, we provide an integrated development environment that first facilitates communication and collaboration among those figures and teams, involved in IT project development, and then enables monitoring and evaluation of the software project quality implemented in it.

7.2 Open Problems

Three specific open problems were identified and, for each of them, we briefly described possible future directions.

- The first direction to consider is related to training, essential to successfully planning a Digital Transformation Strategy, particularly in the Public Sector context. As computer scientists, we have provided public stakeholders with a model to use as a roadmap for such planning for their respective organizations. Furthermore, one of the key technical issues concerns the most suitable Agile software engineering methodologies for managing and developing software projects. Indeed, as pointed out in [250], knowing software engineering fundamentals is crucial for all professionals working on an Agile project for PA, with varying levels of expertise. That is, individuals involved in software development or technical management must be highly knowledgeable about software engineering, while other professionals must have some understanding of it. Additionally, technical experts must also know the legal and regulatory aspects of PA that apply to a particular project they are working on.

- The second direction concerns the methods of validation and evaluation of an Agile approach for PA. As pointed out in a recent review on the subject [280], there is currently no set of commonly accepted criteria. One difficulty in formalizing such criteria is certainly the heterogeneity of how Agile techniques have been applied to PA to date. In this regard, the generality and homogeneity offered by Scrum@PA may help in identifying a homogeneous and consistent set of criteria for validation and evaluation.

- Finally, the third direction concerns providing Scrum@PA paradigm teams, and more generally, Agile teams working for the Public Sector, with a set of tools intended as building blocks of an integrated collaborative development environment for creating digital services (see, for example, [292, 358]). To this end, we have provided the CAS Platform for Leaders and professional Teams involved in Agile IT project management that facilitates collaboration and communication among them and simultaneously provides some functionality to monitor and improve the quality of implemented services. To validate our project, we conducted an ex-post analysis of an already developed Public Administration project, in which we showed how the CAS Platform can also be used as a quality controller of already implemented software projects. This result provides valuable insights into how to improve the CAS platform and offers insights about identifying potential software solutions that can be integrated into it to make it more attractive to Leader professional figures and Teams involved in public Agile IT projects.

Bibliography

[1] Engin Isin and Evelyn Ruppert. *Being Digital Citizens.* Rowman & Littlefield Publishers, 2 edition, 2020.

[2] Mohamed-Iliasse Mahraz, Loubna Benabbou, and Abdelaziz Berrado. A Systematic Literature Review of Digital Transformation. In *International Conference on Industrial Engineering and Operations Management. IEOM Society International*, pages 917–931, 2019.

[3] Gustaf Juell-Skielse, Ida Lindgren, and Maria Akesson. *Service Automation in the Public Sector - Concepts, Empirical Examples and Challenges.* Springer International Publishing, 2022.

[4] Gregory Vial. Understanding Digital Transformation: A Review and a Research Agenda. *The Journal of Strategic Information Systems*, 28(2):118–144, 2019.

[5] European Commission. European Interoperability Framework - Implementation Strategy (2017-2020). Luxembourg: Publications Office of the European Union, 2016.

[6] Frank Danielsen. Benefits and Challenges of Digitalization: An Expert Study on Norwegian Public Organizations. In *Proc. of the 22nd Annual International Conference on Digital Government Research (DG.O2021)*, pages 317–326. ACM, 2021.

[7] Heloise Acco Tives Leão and Edna Dias Canedo. Digitization of Public Services: A Systematic Literature Review. In *Proc. of the 17th Brazilian Symposium on Software Quality*, pages 91–100. ACM, 2018.

[8] Alexander Yukhno. Digital Transformation: Exploring Big Data Governance in Public Administration. *Public Organization Review*, pages 1–15, 2022.

[9] Boris Otto, Michael ten Hompel, and Stefan Wrobel. *Designing Data Spaces The Ecosystem Approach to Competitive Advantage.* Springer Nature, 2022.

[10] Viviana Vaira. Smart City Governance and the Challenge of Digital Platforms Within the Public Sector. In *2022 IEEE International Smart Cities Conference (ISC2)*, pages 1–7, 2022.

[11] Anastasija Nikiforova and Keegan McBride. Open Government Data Portal Usability: A User-Centred Usability Analysis of 41 Open Government Data Portals. *Telematics and Informatics*, 58:101539, 2021.

[12] Judie Attard, Fabrizio Orlandi, Simon Scerri, and Sören Auer. A Systematic Review of Open Government Data Initiatives. *Government Information Quarterly*, 32(4):399–418, 2015.

[13] Maarja Toots, Keegan Mcbride, Tarmo Kalvet, and Robert Krimmer. Open Data As Enabler of Public Service Co-Creation: Exploring the Drivers and Barriers. In *Proc. Conference for E-Democracy and Open Government (CeDEM)*, pages 102–112. IEEE, 2017.

[14] Fatemeh Ahmadi Zeleti and Adegboyega Ojo. Agile Mechanisms For Open Data Process Innovation in Public Sector Organizations: Towards Theory Building. In *Proc. of the 12th International Conference on Theory and Practice of Electronic Governance*, pages 164–174, 2019.

[15] Caihua Liu and Didar Zowghi. Citizen Involvement in Digital Transformation: A Systematic Review and a Framework. *Online Information Review*, 47(4):644–660, 2022.

[16] Rob Kitchin. *The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences.* Sage, 2014.

[17] European Commission. General Data Protection Regulation (GDPR). `https://eur-lex.europa.eu/eli/reg/2016/679`, 2022. [Online; accessed 25-June-2022].

[18] Aleksei Balashov, Anton Barabanov, Viktoriia Degtereva, and Maxim Ivanov. Prospects for Digital Transformation of Public Administration in Russia. In *Proc. of the 2nd International Scientific Conference on Innovations in Digital Economy: SPBPU IDE-2020*, pages 1–7. ACM, 2020.

[19] Antonio Filograna, Paolo Smiraglia, Celia Gilsanz, Srdjan Krco, Arturo Medela, and Tao Su. Cloudification of Public Services in Smart Cities: The Clips Project. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 153–158. IEEE, 2016.

[20] i Hisenda, Comissió d'Economia and de Treball, Àrea and i Economia, Gerència de Presidència and others. A Government Measure For Open Digitisation: Free Software And Agile Development Of Public Administration Services. *Barcelona Ciutat Digital*, 2017.

[21] Vicky Margariti, Dimosthenis Anagnostopoulos, Anastasia Papastilianou, Teta Stamati, and Sofia Angeli. Assessment of Organizational Interoperability in E-Government: A New Model and Tool for Assessing Organizational Interoperability Maturity of a Public Service in Practice. In *Proc. 13th International Conference on Theory and Practice of Electronic Governance*, pages 298–308, 2020.

[22] Djamel Saba, Youcef Sahli, and Abdelkader Hadidi. Toward Smart Cities Based on the Internet of Things. *Smart City Infrastructure: The Blockchain Perspective*, pages 33–75, 2022.

[23] Daniel Abadi, Anastasia Ailamaki, David G. Andersen, Peter Bailis, Magdalena Balazinska, Philip A. Bernstein, Peter A. Boncz, Surajit Chaudhuri, Alvin Cheung, AnHai Doan, Luna Dong, Michael J. Franklin, Juliana Freire, Alon Y. Halevy, Joseph M. Hellerstein, Stratos Idreos, Donald Kossmann, Tim Kraska, Sailesh Krishnamurthy, Volker Markl, Sergey Melnik, Tova Milo, C. Mohan, Thomas Neumann, Beng Chin Ooi, Fatma Ozcan, Jignesh M. Patel, Andrew Pavlo, Raluca A. Popa, Raghu Ramakrishnan, Christopher Ré, Michael Stonebraker, and Dan Suciu. The Seattle Report on Database Research. *Communications of the ACM*, 65(8):72–79, 2022.

[24] Ali Davoudian and Mengchi Liu. Big Data Systems: A Software Engineering Perspective. *ACM Computing Surveys (CSUR)*, 53(5):1–39, 2020.

[25] Teresa Harrison, Donna Canestraro, Theresa Pardo, Martha Avila-Maravilla, Nicolas Soto, Megan Sutherland, G Brian Burke, and Mila Gasco. Applying an Enterprise Data Model in Government. In *Proc. of the 20th Annual International Conference on Digital Government Research*, pages 265–271. ACM, 2019.

[26] Justine S Hastings, Mark Howison, Ted Lawless, John Ucles, and Preston White. Unlocking Data to Improve Public Policy. *Communications of the ACM*, 62(10):48–53, 2019.

[27] Bouchaïb Bounabat. From E-Government to Digital Government: Stakes and Evolution Models. *Electronic Journal of Information Technology*, 10(1):1–20, 2017.

[28] José Ortiz-Bejar and Jesús Ortiz-Bejar. Design and Implementation of Digital Platform for E-Government. In *2021 IEEE URUCON*, pages 547–551. IEEE, 2021.

[29] Li Rong. Smart Cities Lead the Digital Economy: Experience and Advice From China. `https://elib.bsu.by/bitstream/123456789/276760/1/106-117.pdf`, 2022.

[30] Maarja Toots, Keegan McBride, Tarmo Kalvet, Robert Krimmer, Efthimios Tambouris, Eleni Panopoulou, Evangelos Kalampokis, and Konstantinos Tarabanis. A Framework for Data-Driven Public Service Co-Production. In *International Conference on Electronic Government (EGOV)*, pages 264–275. Springer International Publishing, 2017.

[31] Johanna Ylipulli and Aale Luusua. Smart Cities With a Nordic Twist? Public Sector Digitalization in Finnish Data-Rich Cities. *Telematics and Informatics*, 55, 2020.

[32] Tarmo Kalvet, Maarja Toots, and Robert Krimmer. Contributing to a Digital Single Market for Europe: Barriers and Drivers of an EU-Wide Once-Only Principle. In *Proc. of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*, pages 1–8. ACM, 2018.

[33] Charru Malhotra, Vinod Kotwal, and Aniket Basu. Designing National Health Stack for Public Health: Role of ICT-Based Knowledge Management System. In *2019 ITU Kaleidoscope: ICT for Health: Networks, Standards and Innovation (ITU K)*, pages 1–8. IEEE, 2019.

[34] Victoria Kalogirou, Sander Van Dooren, Ilias Dimopoulos, Yannis Charalabidis, Jean-Paul De-Baets, and Georges Lobo. Linked Government Data Hub, an Ontology Agnostic Data Harvester and API. In *Proc. of the 13th International Conference on Theory and Practice of Electronic Governance*, pages 779–782. ACM, 2020.

[35] Daniela Battisti. The Digital Transformation of Italy's Public Sector: Government Cannot Be Left Behind! *JeDEM - eJournal of eDemocracy and Open Government*, 12(1):25–39, 2020.

[36] Antonio Cordella and Andrea Paletti. Government as a Platform, Orchestration, and Public Value Creation: The Italian Case. *Government Information Quarterly*, 36(4), 2019.

[37] Panos Panagiotopoulos, Bram Klievink, and Antonio Cordella. Public Value Creation in Digital Government. *Government Information Quarterly*, 36(4), 2019.

[38] Hyrmet Mydyti, Jaumin Ajdari, and Xhemal Zenuni. Cloud-Based Services Approach as Accelerator in Empowering Digital Transformation. In *Proc. of the 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 1390–1396. IEEE, 2020.

[39] Osama Abied, Othman Ibrahim, and Siti Nuur-Ila Mat Kamal. Adoption of Cloud Computing in E-Government: A Systematic Literature Review. *Pertanika Journal of Science & Technology*, 30(1), 2022.

[40] Kakoli Chaudhury, Anuradha Barua, Rahul Deka, Tapan Gogoi, Adesh Chand Gupta, and Suchitra Pyarelal. Reforming and Strengthening Digital Service Delivery: Case of Government of Assam, 2020.

[41] Luiz Pinheiro Junior, Maria Alexandra Cunha, Marijn Janssen, and Ricardo Matheus. Towards a Framework for Cloud Computing Use by Governments: Leaders, Followers and Laggers. In *Proc. of the 21st Annual International Conference on Digital Government Research*, pages 155–163. ACM, 2020.

[42] Johan Wingren and Zacharias Wesén. Blockchain in the Public Sector: Applications For Improving Services in Society, 2021.

[43] Yassine Jadi and Lin Jie. An Implementation Framework of Business Intelligence in E-Government Systems for Developing Countries: Case Study: Morocco E-Government System. In *Proc. of the International Conference on Information Society (i-Society)*, pages 138–142. IEEE, 2017.

[44] Oliver Neumann, Katharina Guirguis, and Reto Steiner. Exploring Artificial Intelligence Adoption in Public Organizations: A Comparative Case Study. *Public Management Review*, pages 1–28, 2022.

[45] José Ramón Saura and Felipe Debasa. *Handbook of Research on Artificial Intelligence in Government Practices and Processes.* IGI Global, 2022.

[46] Damian A. Tamburri, Willem-Jan Van Den Heuvel, and Martin Garriga. Dataops for Societal Intelligence: A Data Pipeline for Labor Market Skills Extraction and Matching. In *Proc. of the 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, pages 391–394. IEEE, 2020.

[47] Colin van Noordt and Gianluca Misuraca. Evaluating the Impact of Artificial Intelligence Technologies in Public Services: Towards an Assessment Framework. In *Proc. of the 13th International Conference on Theory and Practice of Electronic Governance*, pages 8–16. ACM, 2020.

[48] Gianluigi Viscusi, Aengus Collins, and Marie-Valentine Florin. Governments' Strategic Stance Toward Artificial Intelligence: An Interpretive Display on Europe. In *Proc. of the 13th International Conference on Theory and Practice of Electronic Governance*, pages 44–53. ACM, 2020.

[49] Pratim Datta, Laurie Walker, and Fabrizio Amarilli. Digital Transformation: Learning from Italy's Public Administration. *Journal of Information Technology Teaching Cases*, 10(2):54–71, 2020.

[50] Italian Government. National Cybersecurity Agency (ACN). `https://www.acn.gov.it/en`. [Online; accessed 05-September-2022].

[51] Spain Government. Spanish National Cybersecurity Institute (INCIBE). `https://www.incibe.es/en/`. [Online; accessed 05-September-2022].

[52] United States Government. Cybersecurity & Infrastructure Agency (CISA). `https://www.cisa.gov/`. [Online; accessed 05-September-2022].

[53] Various Authors. European DECODE Project. `https://decodeproject.eu/`. [Online; accessed 18-August-2022].

[54] Ebenezer Agbozo and Alexander N. Medvedev. Towards a Multi-Channel Service Delivery Model in the Data-Driven Public Sector. *Business Informatics*, 14(1):41–50, 2020.

[55] Keegan McBride, Gerli Aavik, Maarja Toots, Tarmo Kalvet, and Robert Krimmer. How Does Open Government Data Driven Co-Creation Occur? Six Factors and a 'Perfect Storm'; Insights From Chicago's Food Inspection Forecasting Model. *Government Information Quarterly*, 36(1):88–97, 2019.

[56] Ines Mergel, Rainer Kattel, Veiko Lember, and Keegan McBride. Citizen-Oriented Digital Transformation in the Public Sector. In *Proc. 19th Annual International Conference on Digital Government Research: Governance in the Data Age*, pages 1–3, 2018.

[57] Hicham Nachit, Mouna Jaafari, Islame El Fikri, and Lhacen Belhcen. Digital Transformation in the Moroccan Public Sector: Drivers and Barriers. *SSRN*, 2021.

[58] Ada Scupola and Ines Mergel. Value Co-Creation and Digital Service Transformation: The Case of Denmark. *SSRN Electronic Journal*, 2021.

[59] Francisco José Rêgo Lopes and Mariela Inés Cortés. Business Value Characterization in Software Projects for Electronic Government in the Brazilian

Federal Government. In *Proc. of the XXXII Brazilian Symposium on Software Engineering*, pages 82–91. ACM, 2018.

[60] Kathrine Vestues, Marius Mikalsen, and Eric Monteiro. Using Digital Platforms to Promote a Service-Oriented Logic in Public Sector Organizations: A Case Study. In *Proc. 54th Hawaii International Conference on System Sciences*, 2021.

[61] Luca Tangi, Marijn Janssen, Michele Benedetti, and Giuliano Noci. Digital Government Transformation: A Structural Equation Modelling Analysis of Driving and Impeding Factors. *International Journal of Information Management*, 60:102356, 2021.

[62] Warit Wipulanusat, Kriengsak Panuwatwanich, Rodney A. Stewart, and Jirapon Sunkpho. Drivers and Barriers to Innovation in the Australian Public Service: A Qualitative Thematic Analysis. *Engineering Management in Production and Services*, 11(1):7–22, 2019.

[63] Carsten Wolff, Azimbek Omar, and Yerlan Shildibekov. How Will We Build Competences For Managing the Digital Transformation? In *Proc. 10th IEEE Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, volume 2. IEEE, 2019.

[64] Olena Aleinikova, Sergey Kravchenko, Viktoriya Hurochkina, Viktor Zvonar, Oleksandr Brechko, and Zoriana Buryk. Project Management Technologies in Public Administration. *Journal of Management Information and Decision Sciences*, 23(5):564–576, 2020.

[65] Koos Themba Masombuka. *A Framework for a Successful Collaboration Culture in Software Development and Operations (DevOps) Environments*. PhD thesis, School of Computing, University of South Africa, 2020.

[66] Margarita Bogdanova, Evelina Parashkevova, and Mariela Stoyanova. Agile Project Management in Public Sector - Methodological Aspects. *Journal of European Economy*, 19(2):283–298, 2020.

[67] Gideon Mekonnen Jonathan. Digital Transformation in the Public Sector: Identifying Critical Success Factors. In *European, Mediterranean, and Middle Eastern Conference on Information Systems*, pages 223–235. Springer International Publishing, 2019.

[68] Rafaela Mantovani Fontana and Sabrina Marczak. Characteristics and Challenges of Agile Software Development Adoption in Brazilian Government. *Journal of Technology Management & Innovation*, 15(2):3–10, 2020.

[69] Mohammed Mubarkoot. Assessment of Factors Influencing Adoption of DevOps Practices in Public Sector and Their Impact on Organizational Culture. In *Proc. International Conference on Science and Technology (ICST)*, volume 2, pages 475–483, 2021.

[70] Karimah Abu Bakar, Roslina Ibrahim, and Yusminar Yunus. Digital Government Evolution and Maturity Models: A Review. *Open International Journal of Informatics (OIJI)*, 8(2):70–87, 2020.

[71] Cem Dener, Hubert Nii-Aponsah, Love E. Ghunney, and Kimberly D. Johns. *GovTech Maturity Index: The State of Public Sector Digital Transformation.* World Bank Publications, 2021.

[72] Helen Margetts and Patrick Dunleavy. The Second Wave of Digital-Era Governance: A Quasi-Paradigm for Government on the Web. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120382, 2013.

[73] Jameela Al-Jaroodi, Nader Mohamed, Nader Kesserwan, and Imad Jawhar. Healthcare 4.0 - Managing a Holistic Transformation. In *Proc. IEEE International Systems Conference (SysCon)*, pages 1–8. IEEE, 2022.

[74] Maija Ylinen. *Digital Transformation in a Finnish Municipality: Tensions as Drivers of Continuous Change.* PhD thesis, Tampere University, 2021.

[75] Andy Syahrizal, Assaf Arief, and Dana Indra Sensuse. Finding Preeminence: A Systematic Literature Review of E-Service Success Factors. In *Proc. 5th International Conference on Informatics and Computing (ICIC)*, pages 1–6. IEEE, 2020.

[76] Bader Alghamdi, Leigh Ellen Potter, and Steve Drew. Desinge and Implementation of Government Cloud Computing Requirements: TOGAF. In *Proc. 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*, pages 1–6. IEEE, 2017.

[77] R Ali Rakhman, Nina Anggraini, Nilo Legowo, and Emil R Kabururuan. A Design of Government Enterprise Architecture Framework Based on G-Cloud Services. *International Journal of Scientific & Technology Research*, 8(9):1692–1700, 2019.

[78] Luca Tangi, Marijn Janssen, Michele Benedetti, and Giuliano Noci. Barriers and Drivers of Digital Transformation in Public Organizations: Results From a Survey in the Netherlands. In *International Conference on Electronic Government*, pages 42–56. Springer International Publishing, 2020.

[79] Kathrine Vestues. *Using Digital Platforms to Promote Value Co-Creation: A Case Study of a Public Sector Organization.* PhD thesis, NTNU, 2021.

[80] Wei Dai, Isaac Wardlaw, Yu Cui, Kashif Mehdi, Yanyan Li, and Jun Long. Data Profiling Technology of Data Governance Regarding Big Data: Review and Rethinking. *Information Technology: New Generations*, pages 439–450, 2016.

[81] Baseer Ahmad Baheer, David Lamas, and Sónia Sousa. A Systematic Literature Review on Existing Digital Government Architectures: State-of-the-Art, Challenges, and Prospects. *Administrative Sciences*, 10(2):25, 2020.

[82] Reshma Agarwal, Vinay Thakur, and Radha Chauhan. Enterprise Architecture for E-Government. In *Proc. of the 10th International Conference on Theory and Practice of Electronic Governance*, pages 47–55, 2017.

[83] Welder Luz, Everton Agilar, Marcos César de Oliveira, Carlos Eduardo R de Melo, Gustavo Pinto, and Rodrigo Bonifácio. An Experience Report on the Adoption of Microservices in Three Brazilian Government Institutions. In *Proc. of the XXXII Brazilian Symposium on Software Engineering*, pages 32–41, 2018.

[84] ISO/IEC JTC1 Information Technology. Smart Cities - Preliminary Report 2014. ISO/IEC, 2015.

[85] Vaia Moustaka, Athena Vakali, and Leonidas G. Anthopoulos. A Systematic Review for Smart City Data Analytics. *ACM Computing Surveys (CSuR)*, 51(5):1–41, 2018.

[86] J Ramon Gil-Garcia, Theresa A Pardo, and Taewoo Nam. What Makes a City Smart? Identifying Core Components and Proposing an Integrative and Comprehensive Conceptualization. *Information Polity*, 20(1):61–87, 2015.

[87] Beatriz Barreto B. Lanza, J. Ramon Gil-Garcia, and Theresa A. Pardo. What Makes a City Smart? Reconsidering the Core Components in the Brazilian Context. In *Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance*, ICEGOV '20, page 638–645, New York, NY, USA, 2020. ACM.

[88] Sita Rani, Ram Krishn Mishra, Mohammed Usman, Aman Kataria, Pramod Kumar, Pankaj Bhambri, and Amit Kumar Mishra. Amalgamation of Advanced Technologies for Sustainable Development of Smart City Environment: A Review. *IEEE Access*, 9, 2021.

[89] Aleksandr Lepekhin, Alexandra Borremans, Igor Ilin, and Sami Jantunen. A Systematic Mapping Study on Internet of Things Challenges. In *Proc. of the IEEE/ACM 1st International Workshop on Software Engineering Research & Practices for the Internet of Things (SERP4IoT)*, pages 9–16. IEEE, 2019.

[90] Marcin Luckner, Maciej Grzenda, Robert Kunicki, and Jaroslaw Legierski. IoT Architecture for Urban Data-Centric Services and Applications. *ACM Transactions on Internet Technology*, 20(3), 2020.

[91] Asad Javed, Sylvain Kubler, Avleen Malhi, Antti Nurminen, Jeremy Robert, and Kary Främling. BIoTope: Building an IoT Open Innovation Ecosystem for Smart Cities. *IEEE Access*, 8:224318–224342, 2020.

[92] Yung-Chang Hsiao, Ming-Ho Wu, and Simon Cimin Li. Elevated Performance of the Smart City — A Case Study of the IoT by Innovation Mode. *IEEE Transactions on Engineering Management*, 68(5):1461–1475, 2021.

[93] Marta Calderón, Gustavo López, and Gabriela Marín. Smartness and Technical Readiness of Latin American Cities: A Critical Assessment. *IEEE Access*, 6:56839–56850, 2018.

[94] Gabriela Viale Pereira, Luis F Luna-Reyes, and J Ramon Gil-Garcia. Governance Innovations, Digital Transformation and the Generation of Public Value in Smart City Initiatives. In *Proc. of the 13th International Conference on Theory and Practice of Electronic Governance*, pages 602–608. ACM, 2020.

[95] Luiz Claudio Diogo Reis, Flavia Cristina Bernardini, Simone Bacellar Leal Ferreira, and Claudia Cappelli. Exploring the Challenges of ICT Governance in Brazilian Smart Cities. In *Proc. of the 14th International Conference on Theory and Practice of Electronic Governance*, pages 429–435. ACM, 2021.

[96] Luiz Claudio Diogo Reis, Flavia Cristina Bernardini, Simone Bacellar Leal Ferreira, and Claudia Cappelli. ICT Governance in Brazilian Smart Cities: An Integrative Approach in the Context of Digital Transformation. In *DG.O2021: The 22nd Annual International Conference on Digital Government Research*, DG.O'21, page 302–316, New York, NY, USA, 2021. ACM.

[97] Margherita Volpe, Iñigo González Rojas, Gabriele Gaffuri, Ramona Marfievici, Edoardo Genova, Ana Gheorghe, Jasmin Kniewallner, and Omar Veledar. Supporting Innovation in Smart Cities Through Cascade Funding: The Case of Water Management. In *2022 IEEE International Smart Cities Conference (ISC2)*, pages 1–7, 2022.

[98] Fernando Monge, Bloomberg-Harvard City Leadership Initiative, Sarah Barns, Rainer Kattel, and Francesca Bria. A New Data Deal: The Case of Barcelona. Technical report, UCL Institute for Innovation and Public Purpose, Working Paper Series (No. WP 2022/02), 2022.

[99] Li Deren, Yu Wenbo, and Shao Zhenfeng. Smart City Based on Digital Twins. *Computational Urban Science*, 1(4):11, 2021.

[100] Barbara Rita Barricelli, Elena Casiraghi, and Daniela Fogli. A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications. *IEEE access*, 7:167653–167671, 2019.

[101] Aidan Fuller, Zhong Fan, Charles Day, and Chris Barlow. Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Access*, 8:108952–108971, 2020.

[102] Thomas Abell, Arndt Husar, and Lim May-Ann. Cloud Computing as a Key Enabler for Tech Start-Ups across Asia and the Pacific. Asian Development Bank - Sustainable Development Working Paper Series, 2021.

[103] C. Vijai and Joyce D. Cloud-Based E-Governance in India. *Shanlax International Journal of Management*, 8(2):72–78, 2020.

[104] Yu Nie, John Talburt, Xinming Li, and Zhongdong Xiao. Chief Data Officer (CDO) Role and Responsibility Analysis. *Journal of Computing Sciences in Colleges*, 33(5):4–12, 2018.

[105] Ishak Firdauzi Ruslan, Muhammad Fahmi Alby, and Muharman Lubis. Applying Data Governance Using DAMA-DMBOK2 Framework: The Case for Human Capital Management Operations. In *Proc. of the 8th International Conference on Industrial and Business Engineering*, pages 336–342. ACM, 2022.

[106] Gideon Mekonnen Jonathan, Bemenet Kasahun Gebremeskel, and Sileshi Demesie Yalew. Privacy and Security in the Digitalisation Era. In *Proc. 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 0837–0844. IEEE, 2020.

[107] Jonathan Rösler, Tobias Söll, Louise Hancock, and Thomas Friedli. Value Co-Creation Between Public Service Organizations and the Private Sector: An Organizational Capabilities Perspective. *Administrative Sciences*, 11(2):1–20, 2021.

[108] María Isabel Arias and Antônio Carlos Gastaud Maçada. Digital Government for E-Government Service Quality: A Literature Review. In *Proc. of the 11th International Conference on Theory and Practice of Electronic Governance*, pages 7–17. ACM, 2018.

[109] Vanessa De Andrade Soares, Gabriela Yukari Iwama, Vitor Gomes Menezes, Marília Miranda Forte Gomes, Glauco Vitor Pedrosa, Wander CM Pereira da Silva, and Rejane Maria Da Costa Figueiredo. Evaluating Government Services Based on User Perspective. In *Proc. of the 20th Annual International Conference on Digital Government Research*, pages 425–432. ACM, 2019.

[110] Vítor G. de Menezes, Glauco V. Pedrosa, Marcos P.P. da Silva, and Rejane M. da C. Figueiredo. Evaluation of Public Services Considering the Expectations of Users — A Systematic Literature Review. *Information*, 13(4):162, 2022.

[111] Agus Hermanto, Rosziati Binti Ibrahim, and Geri Kusnanto. Improving Value-Based E-Government Towards the Achievement of Smart Government. In *Proc. of the 5th International Conference on Informatics and Computing (ICIC)*, pages 1–7. IEEE, 2020.

[112] More Ickson Manda. Leadership and Trust as Key Pillars in “Smart Governance” for Inclusive Growth in the 4th Industrial Revolution (4IR): Evidence from South Africa. In *Proc. of the 14th International Conference on Theory and Practice of Electronic Governance*, pages 308–315. ACM, 2021.

[113] Gerald C Kane, Doug Palmer, Anh Nguyen Phillips, David Kiron, Natasha Buckley, et al. Strategy, Not Technology, Drives Digital Transformation. *MIT Sloan Management Review and Deloitte University Press*, 14(1-25), 2015.

[114] Christine Legner, Torsten Eymann, Thomas Hess, Christian Matt, Tilo Böhmann, Paul Drews, Alexander Mädche, Nils Urbach, and Frederik Ahlemann. Digitalization: Opportunity and Challenge for the Business and Information Systems Engineering Community. *Business & Information Systems Engineering*, 59(4):301–308, 2017.

[115] Linda M. Pittenger, Nicholas Berente, and James Gaskin. Transformational IT Leaders and Digital Innovation: The Moderating Effect of Formal IT Governance. *ACM SIGMIS Database: The DATABASE for Advances in Information Systems*, 53(1):106–133, 2022.

[116] Edimara M. Luciano and Guilherme C. Wiedenhöft. The Role of Organizational Citizenship Behavior and Strategic Alignment in Increasing the Generation of Public Value Through Digital Transformation. In *Proc. of the 13th*

International Conference on Theory and Practice of Electronic Governance, ICEGOV '20, page 494–501, New York, NY, USA, 2020. ACM.

[117] Dipendra Ghimire, Stuart Charters, and Shirley Gibbs. Scaling Agile Software Development Approach in Government Organization in New Zealand. In *Proc. of the 3rd International Conference on Software Engineering and Information Management*, pages 100–104. ACM, 2020.

[118] Maximilian Kupi. *Late to the Party? Agile Methods in British and German Government Institutions.* PhD thesis, Hertie School, Berlin, 2021.

[119] Ines Mergel, Sukumar Ganapati, and Andrew B. Whitford. Agile: A New Way of Governing. *Public Administration Review*, 81(1):161–165, 2021.

[120] Parastoo Mohagheghi and Casper Lassenius. Organizational Implications of Agile Adoption: A Case Study From the Public Sector. In *Proc. of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1444–1454, 2021.

[121] Gleycianne Rodrigues Araujo, Thiago Jose Tavares Avila, and Beatriz Barreto Brasileiro Lanza. Impacts of an Articulation Group for the Development of the Digital Government in the Brazilian Subnational Government. In *Proc. of the 22nd Annual International Conference on Digital Government Research (DG.O2021)*, pages 339–350. ACM, 2021.

[122] Geert Poels, Henderik A Proper, and Dominik Bork. DT4GITM - A Vision for a Framework for Digital Twin Enabled IT Governance. In *Proc. of the 55th Hawaii International Conference on System Sciences (HICSS)*, 2022.

[123] Yiwei Gong, Jun Yang, and Xiaojie Shi. Towards a Comprehensive Understanding of Digital Transformation in Government: Analysis of Flexibility and Enterprise Architecture. *Government Information Quarterly*, 37(3), 2020.

[124] Krikor Maroukian and Stephen R Gulliver. Leading DevOps Practice and Principle Adoption. *arXiv preprint arXiv:2008.10515*, 2020.

[125] Maija Ylinen. Incorporating Agile Practices in Public Sector IT Management: A Nudge Toward Adaptive Governance. *Information Polity*, 26(3):251–271, 2021.

[126] Mateja Nerima and Jolita Ralyté. Towards a Digital Maturity Balance Model for Public Organizations. In *International Conference on Research Challenges in Information Science (RCIS)*, pages 295–310. Springer, 2021.

[127] Alexandros Bousdekis and Dimitris Kardaras. Digital Transformation of Local Government: A Case Study from Greece. In *Proc. IEEE 22nd Conference on Business Informatics (CBI)*, volume 2, pages 131–140. IEEE, 2020.

[128] Luis Felipe Miranda Ramos, Mariana Lameiras, Delfina Soares, and Luis Amaral. Who Is Behind the Scenes of the ICT Backstage? A Study of the ICT Resources in Local Governments. In *Proc. of the 14th International Conference on Theory and Practice of Electronic Governance*, pages 165–171, 2021.

[129] Amjad Umar. A Digital Transformation Lab for Developing Countries and Small to Medium Enterprises. In *Proc. IEEE European Technology and Engineering Management Summit (E-TEMS)*, pages 160–165. IEEE, 2022.

[130] Nuno Vasco Lopes and Soumaya Ben Dhaou. Public Service Delivery Framework: Case of Canada, China and Estonia. In *Proc. 11th International Conference on Theory and Practice of Electronic Governance*, pages 101–110, 2018.

[131] Hsiu-Ling Li. Keys to Promoting Governmental Digital Transformation. *International Journal of Automation and Smart Technology*, 8(4):145–148, 2018.

[132] Morten Meyerhoff Nielsen and Zoran Jordanoski. Digital Transformation, Governance and Coordination Models: A Comparative Study of Australia, Denmark and the Republic of Korea. In *Proc. of the 21st Annual International Conference on Digital Government Research*, pages 285–293. ACM, 2020.

[133] Meelis Kitsing. Alternative Futures for Digital Governance. In *Proc. of the 20th International Conference on Digital Government Research*, pages 48–59, 2019.

[134] The New Yorker Magazine. Estonia, the Digital Republic. `https://www.newyorker.com/magazine/2017/12/18/estonia-the-digital-republic`. [Online; accessed 10-January-2023].

[135] Meelis Kitsing. The Janus-faced Approach to Governance: A Mismatch Between Public Sector Reforms and Digital Government in Estonia. In *Proc. of the 11th International Conference on Theory and Practice of Electronic Governance*, pages 59–68. ACM, 2018.

[136] Jeffrey Jaffe Tim Berners-Lee. World Wide Web Consortium. `https://w3c.org/`, 2022. [Online; accessed 03-June-2022].

[137] Various Authors. Comprehensive Knowledge Archive Network (CKan): The Open-Source Data Management System for Powering Data Hubs and Data Portals. `https://ckan.org/`. [Online; accessed 18-August-2022].

[138] Sarah Barns. Mine your Data: Open Data, Digital Strategies and Entrepreneurial Governance by Code. *Urban Geography*, 37(4):554–571, 2016.

[139] Scott Hawken, Hoon Han, and Chris Pettit. *Open Cities — Open Data: Collaborative Cities in the Information Era.* Springer Nature, 2020.

[140] Various Authors. Sentilo: An Open-Source Software Sensor and Actuator Project. `https://sentilo.io/`. [Online; accessed 18-August-2022].

[141] Meg Young and An Yan. Civic Hackers' User Experiences and Expectations of Seattle's Open Municipal Data Program. In *Proc. of the 50th Hawaii International Conference on System Sciences (HICSS)*, 2017.

[142] City of Chicago. Chicago Open Data Portal. `http://data.cityofchicago.org/`. [Online; accessed 02-September-2022].

[143] City of Chicago. Reduction the Exposure of Citizens of Chicago to Foodborne Diseases. `https://www.chicago.gov/city/en/sites/gearupgetready/home/foodborne-illness.html`. [Online; accessed 12-September-2022].

[144] City of Chicago. Food Inspections Evaluation Machine Learning Model. `https://github.com/Chicago/food-inspections-evaluation`. [Online; accessed 12-September-2022].

[145] Shyny from RStudio. Shiny: R package That Makes It Easy to Build Interactive Web Aps Straight From R. `https://shiny.rstudio.com/`. [Online; accessed 17-September-2022].

[146] Benjamin E. Hermalin, Avery W. Katz, and Richard Craswell. Chapter 1 Contract Law. In A.M. Polinsky and S. Shavell, editors, *Handbook of Law and Economics*, volume 1, pages 3–138. Elsevier, 2007.

[147] Àrea de Treball, Francesca Bria, Paco Rodriguez, Malcolm Bain, Joan Batlle, Ana Bastida Vila, Xabier Eugenio Barandiaran Fernández, Màrius Boada Pla, Guillem Marpons Ucero, Xavier Roca Vilalta, et al. Barcelona City Council ICT Public Procurement Guide, 2018.

[148] Babak A Farshchian, Hilde Sætertrø, and Marianne Stålaker. Experiences From an Interpretative Case Study of Innovative Public Procurement of Digital Systems in the Norwegian Public Sector. In *Proc. of the International Conference on Evaluation and Assessment in Software Engineering (EASE)*, pages 373–374. ACM, 2020.

[149] City of Chicago. Chicago Open Data Portal Development Team. `https://dev.cityofchicago.org/`. [Online; accessed 02-September-2022].

[150] Chicago Open Data Portal Team. How Chicago Protects Privacy in TNP and TAXI Open Data. http://dev.cityofchicago.org/open%20data/data%20portal/2019/04/12/tnp-taxi-privacy.html. [Online; accessed 02-September-2022].

[151] Barcelona Ciutat Digital. Agile Methodologies at Barcelona City Council. https://www.barcelona.cat/digitalstandards/en/agile-methodologies/0.1/_attachments/barcelona_agile_methodologies_0.1.en.pdf. [Online; accessed 20-January-2023].

[152] Sören Auer, Volha Bryl, and Sebastian Tramp. *Linked Open Data - Creating Knowledge Out of Interlinked Data: Results of the LOD2 Project*, volume 8661. Springer, 2014.

[153] World Wide Web Consortium. Resource Description Framework (RDF). https://w3c.org/RDF, 2022. [Online; accessed 03-June-2022].

[154] Hiba Arnaout and Shady Elbassuoni. Effective Searching of RDF Knowledge Graphs. *Journal of Web Semantics*, 48:66–84, 2018.

[155] World Wide Web Consortium (W3C). Semantic Web. https://www.w3.org/standards/semanticweb/. [Online; accessed 03-October-2022].

[156] Edobor Osagie, Mohammad Waqar, Samuel Adebayo, Arkadiusz Stasiewicz, Lukasz Porwol, and Adegboyega Ojo. Usability Evaluation of an Open Data Platform. In *Proc. of the 18th Annual International Conference on Digital Government Research*, pages 495–504. ACM, 2017.

[157] Claire C. Austin. A Path to Big Data Readiness. In *Proc. International Conference on Big Data*, pages 4844–4853. IEEE, 2018.

[158] Jan Kučera, Dušan Chlapek, and Martin Nečaskỳ. Open Government Data Catalogs: Current Approaches and Quality Perspective. In *Proc. 2nd Int. Conf. on Electronic Government and the Information Systems Perspective*, volume 8061 of *LNISA*, pages 152–166, Prague, Czech Republic, 2013. Springer.

[159] The Italian Data Protection Authority. The Italian Data Protection Authority Website. https://www.garanteprivacy.it/web/garante-privacy-en/home_en. [Online; accessed 10-October-2022].

[160] PagoPA S.p.A. Piattaforma Digitale Nazionale Dati (PDND). https://www.pagopa.it/it/prodotti-e-servizi/piattaforma-digitale-nazionale-dati/. [Online; accessed 23-April-2024].

[161] European Commission. Secure idenTity acrOss boRders linKed (STORK) European Project. https://joinup.ec.europa.eu/collection/secure-identity-across-borders-linked-stork. [Online; accessed 10-October-2022].

[162] Philipp Lämmel, Benjamin Dittwald, Lina Bruns, Nikolay Tcholtchev, Yuri Glikman, Silke Cuno, Mathias Flügge, and Ina Schieferdecker. Metadata Harvesting and Quality Assurance Within Open Urban Platforms. *Journal of Data and Information Quality (JDIQ)*, 12(4):1–20, 2020.

[163] European Commission. DCAT Application Profile for Data Portals in Europe. https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/dcat-application-profile-data-portals-europe. [Online; accessed 21-February-2023].

[164] Hadi Habibzadeh, Cem Kaptan, Tolga Soyata, Burak Kantarci, and Azzedine Boukerche. Smart City System Design: A Comprehensive Study of the Application and Data Planes. *ACM Computing Surveys*, 52(2), may 2019.

[165] Kevin Reeves, Carsten Maple, and Gregory Epiphaniou. Digital Twins - Trusted Data Environments Enabling Acceleration of NetZero in Critical National Infrastructure. In *Competitive Advantage in the Digital Economy (CADE 2022)*. Institution of Engineering and Technology (IET), 2022.

[166] Jingtao Zhang, Cai Chen, YuXiong Zhang, Ying Cui, Peng Han, Nan Meng, and Yamin Xu. The Framework and Practices of Digital Twin City. In *Proc. of the 12th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 111–116. IEEE, 2022.

[167] Australia's National Science Agency. Data61: Australian "Digital Twin" Technology Set to Transform Manufacturing. https://www.csiro.au/en/News/News-releases/2019/Australian-digital-twin-technology-set-to-transform-manufacturing. [Online; accessed 15-December-2022].

[168] New South Wales Government (NSW). NSW Digital Twin. https://www.digital.nsw.gov.au/article/twinning-spatial-services-has-created-a-digital-twin-of-nsw. [Online; accessed 15-December-2022].

[169] Tomorrow City: The Heart of Urban Innovation. Virtual Singapore: Singapore Experiments With Its Digital Twin to Improve City Life. https://www.tomorrow.city/a/

singapore-experiments-with-its-digital-twin-to-improve-city-life. [Online; accessed 15-December-2022].

[170] Ralf-Martin Soe. Smart Twin Cities Via Urban Operating System. In *Proc. of the 10th International Conference on Theory and Practice of Electronic Governance*, pages 391–400. ACM, 2017.

[171] Rois Saputro, Ingrid Pappel, Heiko Vainsalu, Silvia Lips, and Dirk Draheim. Prerequisites for the Adoption of the X-Road Interoperability and Data Exchange Framework: A Comparative Study. In *Proc. of the 7th International Conference on eDemocracy & eGovernment (ICEDEG)*, pages 216–222. IEEE, 2020.

[172] Ralf-Martin Soe. FINEST Twins: Platform For Cross-Border Smart City Solutions. In *Proc. of the 18th Annual International Conference on Digital Government Research*, pages 352–357. ACM, 2017.

[173] Andrés Luque-Ayala and Simon Marvin. *Urban Operating Systems: Producing the Computational City.* MIT Press, 2020.

[174] Commission Nationale de l'Informatique et des Libertés (CNIL). CNIL: Map of the Data Protection Authorities Around the World. https://www.cnil.fr/en/data-protection-around-the-world. [Online; accessed 17-October-2022].

[175] Isaac Potoczny-Jones, Erin Kenneally, and John Ruffing. Encrypted Dataset Collaboration: Intelligent Privacy for Smart Cities. In *Proc. of the 2nd ACM/EIGSCC Symposium on Smart Cities and Communities*, pages 1–8. ACM, 2019.

[176] European Union Government. European Union Agency for Cybersecurity (ENISA) Website. https://www.enisa.europa.eu/. [Online; accessed 08-October-2022].

[177] CLUSIT. Italy Report on ICT Security. Technical report, Italian Association for IT Security, 2022.

[178] Organisation for Economic Co-Operation and Development (OECD). https://www.oecd.org/. [Online; accessed 30-September-2022].

[179] Wojciech Urbanczyk and Jan Werewka. Application of a Government Data Center (GDC) Reference Model for Security Management Analysis. In *Proc. of the International Conference on E-Business Engineering (ICEBE)*, pages 112–119. IEEE, 2021.

[180] Alana Almeida, João Pedro De Lima Cassiano, and Sérgio Ribeiro. TecSEG Project-Research, Development and Innovation in Security Assessment Methodologies to Brazil. In *Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pages 1–6. IEEE, 2022.

[181] Yifeng Zheng, Arindam Pal, Sharif Abuadbba, Shiva Raj Pokhrel, Surya Nepal, and Helge Janicke. Towards IoT Security Automation and Orchestration. In *Proc. of the 2nd IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 55–63. IEEE, 2020.

[182] Cristina Alcaraz and Javier Lopez. Digital Twin: A Comprehensive Survey of Security Threats. *IEEE Communications Surveys & Tutorials*, 24(3):1475–1503, 2022.

[183] Ben Eaton, Silvia Elaluf-Calderwood, Carsten Sørensen, and Youngjin Yoo. Distributed Tuning of Boundary Resources: The Case of Apple's iOS Service System. *MIS Quarterly*, 39(1):217–244, 2015.

[184] Ulf Hedestig, Daniel Skog, and Mikael Söderström. Co-Producing Public Value Through IoT and Social Media. In *Proc. of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*, pages 1–10. ACM, 2018.

[185] Sofie Pilemalm. Participatory Design in Emerging Civic Engagement Initiatives in the New Public Sector: Applying PD Concepts in Resource-Scarce Organizations. *ACM Transactions on Computer-Human Interaction*, 25(1), 2018.

[186] Efthimios Tambouris and Konstantinos Tarabanis. Towards Inclusive Integrated Public Service (IPS) Co-Creation and Provision. In *Proc. of the 22nd Annual International Conference on Digital Government Research (DG.O2021)*, pages 458–462. ACM, 2021.

[187] Artur Afonso Sousa, Pedro Agante, Steven Abrantes, and Luís Borges Gouveia. Liberopinion: A Web Platform for Public Participation. In *Proc. of the 11th International Conference on Theory and Practice of Electronic Governance*, ICEGOV '18, page 199–208, New York, NY, USA, 2018. ACM.

[188] Pilar Maria Guerrieri, Sara Comai, and Mariagrazia Fugini. Evolving Experiences of Participation: Towards an e-ILAUD Tool. In *Proc. of the IEEE International Smart Cities Conference (ISC2)*, pages 1–7. IEEE, 2021.

[189] Dessislava Petrova-Antonova and Sylvia Ilieva. Methodological Framework for Digital Transition and Performance Assessment of Smart Cities. In *Proc. of the 4th International Conference on Smart and Sustainable Technologies (SpliTech)*, pages 1–6. IEEE, 2019.

[190] J. Shore and S. Warden. *The Art of Agile Development*. Theory in practice. O'Reilly Media, Incorporated, 2008.

[191] Mounir Ajam. *Project Management Beyond Waterfall and Agile*. Auerbach Publications, 2018.

[192] Amrutaunshu Nerurkar and Indrajit Das. Agile Project Management in Large Scale Digital Transformation Projects in Government and Public Sector: A Case Study of DILRMP Project. In *Proc. of the 10th International Conference on Theory and Practice of Electronic Governance*, pages 580–581. ACM, 2017.

[193] Amrutaunshu Nerurkar and Indrajit Das. Analysis of DILRMP Project: Identifying the Applicability of Agile Project Management for Digital Transformation Projects in Government and Public Sector. In *Procs. the Special Collection on eGovernment Innovations in India*, pages 34–38. ACM, 2017.

[194] Rohit Dwivedula and Narasimha Bolloju. Transitioning from Plan-Driven Methods to Agile Methods-Preparation for a Systematic Literature Review. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pages 944–950. IEEE, 2020.

[195] Afonso Ribeiro and Luísa Domingues. Acceptance of an Agile Methodology in the Public Sector. *Procedia Computer Science*, 138:621–629, 2018.

[196] Fabio Reginaldo and Gleison Santos. Challenges in Agile Transformation Journey: A Qualitative Study. In *Proc. of the XXXIV Brazilian Symposium on Software Engineering*, pages 11–20. ACM, 2020.

[197] Miloš Jovanović, Antoni-Lluís Mesquida, Antonia Mas, and Ricardo Colomo-Palacios. Agile Transition and Adoption Frameworks, Issues and Factors: A Systematic Mapping. *IEEE Access*, 8, 2020.

[198] Markus Borg, Thomas Olsson, Ulrik Franke, and Saïd Assar. Digitalization of Swedish Government Agencies: A Perspective Through the Lens of a Software Development Census. In *Proc. 40th International Conference on Software Engineering: Software Engineering in Society*, pages 37–46, 2018.

[199] Ida Lindgren and Anne Fleur van Veenstra. Digital Government Transformation: A Case Illustrating Public E-Service Development as Part of Public

Sector Transformation. In *Proc. of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*, pages 1–6. ACM, 2018.

[200] Len Bass, Ingo Weber, and Liming Zhu. *DevOps: A Software Architect's Perspective*. Addison-Wesley Professional, 2015.

[201] Ravi Teja Yarlagadda. How Public Sectors Can Adopt the DevOps Practices to Enhance the System. *International Journal of Emerging Technologies and Innovative Research*, 5(9):796–801, 2018.

[202] Nicole Forsgren and Mik Kersten. DevOps Metrics. *Communications of the ACM*, 61(4):44–48, 2018.

[203] Paulo Meirelles, Melissa Wen, Antonio Terceiro, Rodrigo Siqueira, Lucas Kanashiro, and Hilmer Neri. Brazilian Public Software Portal: An Integrated Platform for Collaborative Development. In *Proc. of the 13th International Symposium on Open Collaboration*, pages 1–10. ACM, 2017.

[204] Vlad Stirbu and Tommi Mikkonen. Towards Agile Yet Regulatory-Compliant Development of Medical Software. In *Proc. International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pages 337–340. IEEE, 2018.

[205] Brazilian Federal Government. Brazilian Public Software: Operations Manual (prod). `https://softwarepublico.gov.br/doc/`. [Online; accessed 03-March-2023].

[206] Mohammad Zarour, Norah Alhammad, Mamdouh Alenezi, and Khalid Alsarayrah. Devops Process Model Adoption in Saudi Arabia: An Empirical Study. *Jordanian Journal of Computers and Information Technology (JJCIT)*, 6(03), 2020.

[207] Ineta Bucena and Marite Kirikova. Simplifying the DevOps Adoption Process. In *BIR Workshops*, pages 1–15, 2017.

[208] AL-Zahran Sarah and Bahjat Fakieh. How DevOps Practices Support Digital Transformation. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(3):2780–2788, 2020.

[209] Barry Boehm. Get Ready For Agile Methods, With Care. *Computer*, 35(1):64–69, 2002.

[210] Cristian Fagarasan, Ciprian Cristea, Maria Cristea, Ovidiu Popa, Cristina Mihele, and Adrian Pisla. The Delivery of Large-Scale Software Products Through the Adoption of the SAFe Framework. In *Proc. International Conference on Development and Application Systems (DAS)*, pages 137–143. IEEE, 2022.

[211] Henry Edison, Xiaofeng Wang, and Kieran Conboy. Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review. *IEEE Transactions on Software Engineering*, 2021.

[212] Nadine Anders and Birgit Schenk. Agility in Public Administration - Is Agility a Possibility and Where are its Limits? In *Central and Eastern European eDem and eGov Days*, pages 97–104, 2019.

[213] Jannik Fangmann, Hanna Looks, Jörg Thomaschewski, and Eva-Maria Schön. Agile Transformation in E-Government Projects. In *Proc. of the 15th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2020.

[214] Péter Fehér and Zoltán Szabó. Digitalization in the Public Sector - Findings of a Hungarian Survey. In *Proc. 12th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*, pages 1–6. IEEE, 2018.

[215] Muhammad Akmal Juniawan, Puspa Sandhyaduhita, Betty Purwandari, Satrio Baskoro Yudhoatmojo, and Made Ayu Aristyana Dewi. Smart Government Assessment Using Scottish Smart City Maturity Model: A Case Study of Depok City. In *Proc. International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 99–104. IEEE, 2017.

[216] Welder Pinheiro Luz, Gustavo Pinto, and Rodrigo Bonifácio. Building a Collaborative Culture: A Grounded Theory of Well Succeeded DevOps Adoption in Practice. In *Proc. 12th ACM/IEEE Int. Symposium on Empirical Software Engineering and Measurement*, pages 1–10. ACM, 2018.

[217] Welder Pinheiro Luz, Gustavo Pinto, and Rodrigo Bonifácio. Adopting DevOps in the Real World: A Theory, a Model, and a Case Study. *Journal of Systems and Software*, 157, 2019.

[218] Kuno Schedler, Ali Asker Guenduez, and Ruth Frischknecht. How Smart Can Government Be? Exploring Barriers to the Adoption of Smart Government. *Information Polity*, 24(1):3–20, 2019.

[219] Kristin Vogelsang, Kirsten Liere-Netheler, Sven Packmohr, and Uwe Hoppe. A Taxonomy of Barriers to Digital Transformation. In *Proc. 14th International Conference on Wirtschaftsinformatik*, pages 736–750, 2019.

[220] Cornell University, Institut Européen d'Administration des Affairs, and World Intellectual Property Organization. Global Innovation Index. https://www.globalinnovationindex.org/. [Online; accessed 30-September-2022].

[221] Zanna Aniscenko, Andrés Robalino-López, Tomás Escobar Rodríguez, and Bernabé Escobar Pérez. Regional E-Government Development: Evolution of EGDI in Andean Countries. In *Proc. 4th International Conference on eDemocracy & eGovernment (ICEDEG)*, pages 22–31. IEEE, 2017.

[222] Development Bank of Latin America. Development Bank of Latin America GovTech Index. https://www.caf.com/en/currently/news/2020/06/caf-publishes-the-first-iberoamerican-govtech-index/. [Online; accessed 30-September-2022].

[223] European Commission. EC eGovernment Benchmark. https://digital-strategy.ec.europa.eu/en/library/egovernment-benchmark-2022. [Online; accessed 30-September-2022].

[224] Organisation for Economic Co-Operation and Development (OECD). OECD Digital Government Index. https://www.oecd.org/gov/digital-government/. [Online; accessed 30-September-2022].

[225] The World Bank. World Bank Digital Adoption Index. https://www.worldbank.org/en/publication/wdr2016/Digital-Adoption-Index. [Online; accessed 30-September-2022].

[226] United Nations. United Nations E-Government Development Index. ”https://publicadministration.un.org/egovkb/en-us/About/Overview/-E-Government-Development-Index”. [Online; accessed 30-September-2022].

[227] Lukman Idowu Lamid, Isa Ali Ibrahim, Kashifu Inuwa Abdullahi, and Usman Gambo Abdullahi. A Framework for Digital Government Transformation Performance Assessment and Toolkit for Developing Countries. In *Proc. of the 14th International Conference on Theory and Practice of Electronic Governance*, pages 203–215. ACM, 2021.

[228] Gul Jussupova, Baurzhan Bokayev, and Dauletbay Zhussip. Digital Government Maturity as a Technologically New E-Government Maturity Model: Experience of Kazakhstan. In *Proc. of the 3rd International Conference on E-commerce, E-Business and E-Government*, pages 10–14. ACM, 2019.

[229] CMMI Institute. Capability Digital Maturity Model (CMMI V2.0). https://cmmiinstitute.com/cmmi. [Online; accessed 30-September-2022].

[230] Diogo Proença, Ricardo Vieira, and José Borbinha. Information Governance Maturity Model Final Development Iteration. In *International Conference on Theory and Practice of Digital Libraries*, pages 128–139. Springer, 2017.

[231] Gonzalo Valdés, Mauricio Solar, Hernán Astudillo, Marcelo Iribarren, Gastón Concha, and Marcello Visconti. Conception, Development and Implementation of an E-Government Maturity Model in Public Agencies. *Government Information Quarterly*, 28(2):176–187, 2011.

[232] The World Bank. (GovTech: The New Frontier in Digital Government Transformation). `https://www.worldbank.org/en/programs/govtech`. [Online; accessed 30-September-2022].

[233] Daniel Teixeira, Rúben Pereira, Telmo Henriques, Miguel Mira Da Silva, João Faustino, and Miguel Silva. A Maturity Model for DevOps. *International Journal of Agile Systems and Management*, 13(4):464–511, 2020.

[234] Yannis Charalabidis, Anneke Zuiderwijk, Charalampos Alexopoulos, Marijn Janssen, Thomas Lampoltshammer, and Enrico Ferro. The Multiple Life Cycles of Open Data Creation and Use. In *The World of Open Data*, pages 11–31. Springer, 2018.

[235] Marijn Janssen, Yannis Charalabidis, and Anneke Zuiderwijk. Benefits, Adoption Barriers and Myths of Open Data and Open Government. *Information Systems Management*, 29(4):258–268, 2012.

[236] Maciej Besta and Torsten Hoefler. Survey and Taxonomy of Lossless Graph Compression and Space-Efficient Graph Representations. *arXiv preprint arXiv:1806.01799*, 2018.

[237] Nieves R. Brisaboa, Ana Cerdeira-Pena, Guillermo de Bernardo, Antonio Fariña, and Gonzalo Navarro. Space/Time-Efficient RDF Stores Based on Circular Suffix Sorting. *The Journal of Supercomputing*, 79(5):5643–5683, 2022.

[238] Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. Graph Summarization Methods and Applications: A Survey. *ACM computing surveys (CSUR)*, 51(3):1–34, 2018.

[239] Sebastian Maneth and Fabian Peternek. A Survey on Methods and Systems for Graph Compression. *arXiv preprint arXiv:1504.00616*, 2015.

[240] Miguel A. Martínez-Prieto, Javier D. Fernández, Antonio Hernández-Illera, and Claudio Gutiérrez. *RDF Compression*, pages 1–11. Springer International Publishing, 2018.

[241] Sofia Agostinelli, Fabrizio Cumo, Giambattista Guidi, and Claudio Tomazzoli. Cyber-Physical Systems Improving Building Energy Management: Digital Twin and Artificial Intelligence. *Energies*, 14(8), 2021.

[242] Hao Wang, Xiaowei Chen, Fu Jia, and Xiaojuan Cheng. Digital Twin-Supported Smart City: Status, Challenges and Future Research Directions. *Expert Systems With Applications*, 217, 2023.

[243] Italian Ministry of Economy and Finance (MEF). The National Recovery and Resilience Plan (NRRP). `https://www.mef.gov.it/en/focus/The-National-Recovery-and-Resilience-Plan-NRRP/`. [Online; accessed 27-September-2022].

[244] European Commission. Digital Education Action Plan (2021-2027). `https://education.ec.europa.eu/focus-topics/digital-education/action-plan`. [Online; accessed 20-March-2023].

[245] US Department of Education - Office of Educational Technology. Advancing Digital Equity for All: Community-Based Recommendations for Developing Effective Digital Equity Plans to Close the Digital Divide and Enable Technology-Empowered Learning. `https://tech.ed.gov/files/2022/09/DEER-Resource-Guide_FINAL.pdf`. [Online; accessed 20-March-2023].

[246] David Powner. Software Development: Effective Practices and Federal Challenges in Applying Agile Methods. *United States Government Accountability Office*, 2012.

[247] United Kingdom Government. Profession Capabilities Framework. `https://www.gov.uk/government/collections/digital-data-and-technology-profession-capability-framework`, 2023. Accessed: 2023-05-03.

[248] UK Government. United Kingdom Government - Service Manual. `https://www.gov.uk/service-manual`, 2023. Accessed: 2023-05-02.

[249] Luigi Benedicenti, Paolo Ciancarini, Franco Cotugno, Angelo Messina, Alberto Sillitti, and Giancarlo Succi. Improved agile: A customized scrum process for project management in defense and security. *Software Project Management for Distributed Computing: Life-Cycle Methods for Developing Scalable and Reliable Tools*, pages 289–314, 2017.

[250] Luigi Benedicenti, Angelo Messina, and Alberto Sillitti. iagile: Mission critical military software development. In *2017 International Conference on High Performance Computing & Simulation (HPCS)*, pages 545–552, 2017.

[251] Franco Raffaele Cotugno. Managing Increasing User Needs Complexity Within The ITA Army Agile Framework. In *Proc. of 4th International Conference in Software Engineering for Defence Applications (SEDA 2015*, pages 1–11. Springer, 2016.

[252] Vincenzo Mauro and Angelo Messina. AMINSEP: Agile Methodology Implementation For A New Software Engineering Paradigm Definition. A Research Project Proposal. In *Proc. of the 4th International Conference in Software Engineering for Defence Applications (SEDA 2015)*, pages 27–33. Springer, 2016.

[253] Angelo Messina, Franco Fiore, Mauro Ruggiero, Paolo Ciancarini, and Daniel Russo. A New Agile Paradigm For Mission-Critical Software Development. *CrossTalk*, 29(6):25–30, 2016.

[254] Barcelona Municipal Institute of Information Technology (IMI). Scrum@IMI Framework. `https://ajuntament.barcelona.cat/imi/sites/default/files/marc_de_treball_scrumimi_per_proveidors.pdf`. Accessed: 2023-05-11.

[255] Martin Fowler, Jim Highsmith, et al. The Agile Manifesto. *Software Development*, 9(8):28–35, 2001.

[256] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen, and Juhani Warsta. Agile Software Development Methods: Review And Analysis. *arXiv preprint arXiv:1709.08439*, 2017.

[257] Mike Cohn. *Succeeding With Agile: Software Development Using Scrum*. Pearson Education, 2010.

[258] Dominik Maximini. *The Scrum Culture: Introducing Agile Methods in Organizations*. Springer, 2018.

[259] Mark C. Paulk. A Scrum Adoption Survey. *Software Quality Professional*, 15(2):27–34, 2013.

[260] Linda Rising and Norman S. Janoff. The Scrum Software Development Process For Small Teams. *IEEE Software*, 17(4):26–32, 2000.

[261] Ken Schwaber. Scrum Development Process. In *Business Object Design And Implementation: OOPSLA'95 Workshop Proceedings 16 October 1995, Austin, Texas*, pages 117–134. Springer, 1997.

[262] Jeff Sutherland. Inventing and Reinventing Scrum in Five Companies. *Cutter IT Journal*, 14(21):5–11, 2001.

[263] David J. Anderson and Andy Carmichael. *Essential Kanban Condensed.* Blue Hole Press, 2016.

[264] Kent Beck. *Extreme Programming Explained: Embrace Change.* Addison-Wesley Professional, 2000.

[265] M. Steven Palmquist, Mary Ann Lapham, Suzanne Miller, Timothy Chick, and Ipek Ozkaya. Parallel Worlds: Agile and Waterfall Differences and Similarities. Technical Report CMU-SEI-13-TN-21. *Software Engineering Institute, Carnegie Mellon University*, 1(1):1–3, 2013.

[266] Ken Schwaber and Mike Beedle. *Agile software development with scrum. Series in Agile Software Development*, volume 1. Prentice Hall Upper Saddle River, 2002.

[267] Kent Beck. Embracing Change With Extreme Programming. *Computer*, 32(10):70–77, 1999.

[268] Daud Alam and Uwe Gühl. Agility In Projects. In *Project Management For Practice: A Guide And Toolbox For Successful Projects*, pages 123–150. Springer, 2022.

[269] Zaigham Mushtaq and M. Rizwan Jameel Qureshi. Novel Hybrid Model: Integrating Scrum and XP. *International Journal of Information Technology and Computer Science (IJITCS)*, 4(6):39–44, 2012.

[270] Gleiston Guerrero-Ulloa, Carlos Rodríguez-Domínguez, and Miguel J Hornos. Agile Methodologies Applied To The Development Of Internet Of Things (IoT)-Based Systems: A Review. *Sensors*, 23(2):790, 2023.

[271] P.S. Shama and Angadi Shivamanth. A Review Of Agile Software Development Methodologies. *International Journal Of Advanced Studies In Computers, Science And Engineering*, 4(11):1–6, 2015.

[272] James P. Womack and Daniel T. Jones. Lean Thinking—Banish Waste And Create Wealth In Your Corporation. *Journal of the Operational Research Society*, 48(11):1148–1148, 1997.

[273] Atlassian Corporation. Atlassian Company Website. `https://www.atlassian.com/`. [Online; accessed 18-September-2023].

[274] Atlassian Corporation. Jira Project Management Software. `https://www.atlassian.com/software/jira`. [Online; accessed 18-September-2023].

[275] Slack Technologies, Limited Liability Company. Slack Company Website. https://slack.com/. [Online; accessed 18-September-2023].

[276] Atlassian Corporation. Bitbucket: Git Solution for Teams Using Jira. https://bitbucket.org/. [Online; accessed 18-September-2023].

[277] United States Government - Accountability Office (GAO). Agile Assessment Guide: Best Practices for Agile Adoption and Implementation. https://www.gao.gov/products/gao-20-590g. [Online; accessed 10-January-2024].

[278] Agile Delivery Community. Agile and Government Services: An Introduction. https://www.gov.uk/service-manual/agile-delivery/agile-government-services-introduction. [Online; accessed 10-January-2024].

[279] Marcelo Carvalho Fernandes, Antonio Juarez Alencar, Eber Assis Schmitz, Monica Ferreira da Silva, and Petros S Stefaneas. Evaluation of Agile Software Projects in the Public Sector: A Literature Review. *Journal of Software*, 11(3):312–325, 2016.

[280] David Baxter, Nicholas Dacre, Hao Dong, and Serkan Ceylan. Institutional Challenges in Agile Adoption: Evidence from a Public Sector IT Project. *Government Information Quarterly*, 2023.

[281] Isaque Vacari and Rafael Prikladnicki. Adopting Agile Methods in the Public Sector: A Systematic Literature Review. In *International Conferences on Software Engineering and Knowledge Engineering (SEKE)*, pages 709–714. KSI Research Inc. and Knowledge Systems Institute Graduate School, 2015.

[282] Ashish Mundra, Sanjay Misra, and Chitra A. Dhawale. Practical scrum-scrum team: Way to produce successful and quality software. In *2013 13th International Conference on Computational Science and Its Applications*, pages 119–123, 2013.

[283] Barcelona City Council. Barcelona City Council: Agile Policy. https://ajuntamentdebarcelona.github.io/ethical-digital-standards-site/agile-methodologies/0.1/policy.html. [Online; accessed 07-January-2024].

[284] Ajuntament de Barcelona. Ajuntament de Barcelona GitHub repository. https://github.com/AjuntamentdeBarcelona/agile-methodologies-bcn-en. [Online; accessed 20-November-2023].

[285] Yannis Charalabidis, Anneke Zuiderwijk, Charalampos Alexopoulos, Marijn Janssen, Thomas Lampoltshammer, and Enrico Ferro. *The World of Open Data: Concepts, Methods, Tools and Experiences*. Springer, 2019.

[286] Barg Upender. Staying agile in government software projects. In *Agile Development Conference (ADC'05)*, pages 153–159. IEEE, 2005.

[287] Alberto Sillitti and Giancarlo Succi. Requirements Engineering For Agile Methods. *Engineering and Managing Software Requirements*, pages 309–326, 2005.

[288] Project Management Institute. *A Guide to the Project Management Body of Knowledge: PMBOK Guide*. Project Management Institute, 2013. [Online; accessed 2024-01-18].

[289] Ines Mergel, Yiwei Gong, and John Bertot. Agile government: Systematic literature review and future research, 2018.

[290] Forrester. The Forrester Wave: Collaborative Work Management Tools, Q4 2022. `https://reprints2.forrester.com/#/assets/2/89/RES176413/report?userkey=50b804a7-c6ff-4ae0-8d7e-e098eab06333`. [Online; accessed 27-September-2023].

[291] Paolo Ciancarini, Marcello Missiroli, and Alberto Sillitti. Preferred Tools for Agile Development: A Sociocultural Perspective. In *Software Technology: Methods and Tools: 51st International Conference, TOOLS 2019, Innopolis, Russia, October 15–17, 2019, Proceedings 51*, pages 43–58. Springer, 2019.

[292] Paolo Ciancarini, Marcello Missiroli, Francesco Poggi, and Daniel Russo. An Open Source Environment for an Agile Development Model. In *Open Source Systems: 16th IFIP WG 2.13 International Conference, OSS 2020, Innopolis, Russia, May 12–14, 2020, Proceedings 16*, pages 148–162. Springer, 2020.

[293] Dipti Jadhav, Jyoti Kundale, Sumedha Bhagwat, and Jyoti Joshi. A Systematic Review of the Tools and Techniques in Distributed Agile Software Development. *Agile Software Development: Trends, Challenges and Applications*, pages 161–186, 2023.

[294] GitLab, Incorporated. GitLab: The DevSecOps Platform. `https://www.gitlab.com/`. [Online; accessed 26-September-2023].

[295] Git. Git: A Distributed Version Control Software. `https://git-scm.com/`. [Online; accessed 06-October-2023].

[296] Gitlab, Incorporated. Gitlab Documentation. `https://docs.gitlab.com/`. [Online; accessed 03-October-2023].

[297] Taiga. Taiga: The Free and Open-Source Project Management Tool. `https://taiga.io/`. [Online; accessed 28-September-2023].

[298] Jenkins. Jenkins: Build Great Things At Any Scale. `https://jenkins.io/`. [Online; accessed 28-September-2023].

[299] The Apache Software Foundation. Apache Ant: A Software for Automating the Build Process. `https://ant.apache.org/`. [Online; accessed 29-September-2023].

[300] The Apache Software Foundation. Apache Maven: A Software Project Management and Comprehension Tool. `https://maven.apache.org/`. [Online; accessed 28-September-2023].

[301] Mattermost. Mattermost: The Secure Collaboration Hub For Technical Teams. `https://mattermost.com/`. [Online; accessed 28-September-2023].

[302] Draw.io. Draw.io: Flowchart Maker and Online Diagram Tool. `https:/draw.io/`. [Online; accessed 28-September-2023].

[303] SonarSource, Société Anonyme. Sonarqube: Clean Code for Teams and Enterprises. `https://www.sonarsource.com/products/sonarqube/`. [Online; accessed 28-September-2023].

[304] GitHub, Incorporated. GitHub: The AI-Powered Developer Platform To Build, Scale, And Deliver Secure Software. `https://www.github.com/`. [Online; accessed 27-September-2023].

[305] GitHub, Incorporated. GitHub Copilot: Your AI Pair Programmer. `https://github.com/features/copilot`. [Online; accessed 27-September-2023].

[306] OpenAI. OpenAI: Creating Safe Artificial General Intelligence That Benefits All Of Humanity. `https://openai.com/`. [Online; accessed 27-September-2023].

[307] GitHub, Incorporated. GitHub Documentation. `https://docs.github.com/en`. [Online; accessed 05-October-2023].

[308] Atlassian Corporation. Trello: Manage Your Team's From Anywhere. `https://trello.com/`. [Online; accessed 25-September-2023].

[309] Atlassian Corporation. Confluence: Accomplish More Together. https://www.atlassian.com/software/confluence/. [Online; accessed 25-September-2023].

[310] Asana Incorporated. Asana: Meet The Smarter Way To Work. https://asana.com/. [Online; accessed 25-September-2023].

[311] Microsoft Corporation. Microsoft Website. https://www.microsoft.com/. [Online; accessed 26-September-2023].

[312] Microsoft Corporation. Microsoft Teams: The Microsoft 365 Teamwork Hub. https://www.microsoft.com/it-it/microsoft-365/project/project-management-software. [Online; accessed 22-September-2023].

[313] Microsoft Corporation. Microsoft Azure DevOps. https://azure.microsoft.com/it-it/products/devops. [Online; accessed 26-September-2023].

[314] Checkmarx. Checkmarx SAST: Scan With Ease At The Source Code Level. https://checkmarx.com/cxsast-source-code-scanning/. [Online; accessed 14-October-2023].

[315] Eclipse Foundation. Eclipse: An Integrated Development Environment Used in Computer Programming. https://www.eclipse.org/. [Online; accessed 15-October-2023].

[316] Microsoft Corporation. Visual Studio. https://visualstudio.microsoft.com/. [Online; accessed 27-September-2023].

[317] Microsoft Corporation. Visual Studio Code. https://code.visualstudio.com/. [Online; accessed 27-September-2023].

[318] JetBrains. IntelliJ: The Leading Java and Kotlin IDE. https://www.jetbrains.com/idea/. [Online; accessed 15-October-2023].

[319] Jitsi. Jitsi: A Secure Video Conferencing Solution. https://jitsi.org/. [Online; accessed 13-October-2023].

[320] Compositional Agile System (CAS) Platform. CAS GitHub Repository. https://github.com/cas-agile. [Online; accessed 19-December-2023].

[321] PagoPA S.p.A. IO: The Italian Public Services App. https://github.com/pagopa/io-app. [Online; accessed 20-December-2023].

[322] Docker. Docker: Accelerated Container Application Development. https://www.docker.com/. [Online; accessed 14-October-2023].

[323] Ejwa Software. Gitinspector. https://github.com/ejwa/gitinspector/. [Online; accessed 26-October-2023].

[324] Andrew Caudwell. Gource. https://gource.io/. [Online; accessed 26-October-2023].

[325] RedHat, Inc. Ansible: A Radically Simple IT Automation Platform. https://www.ansible.com/. [Online; accessed 14-October-2023].

[326] Internet Engineering Task Force (IETF) OAuth Working Group. OAuth 2.0: The Industry-Standard Protocol for Authorization. https://oauth.net/2/. [Online; accessed 04-November-2023].

[327] Internet Engineering Task Force (IETF). Lightweight Directory Access Protocol (LDAP): The Standard Protocol for Querying and Modifying Directory Services. https://ldap.com/. [Online; accessed 04-November-2023].

[328] Organization for the Advancement of Structured Information Standards (OASIS) - Security Services Technical Committee. Security Assertion Markup Language (SAML): The Standard for Exchanging Authentication and Authorization Identities Between Security Domains. https://saml.xml.org/. [Online; accessed 04-November-2023].

[329] Ujjwal Joshi, Sangwhan Cha, and Saeed Esmaili-Sardari. Towards Adoption of Authentication and Authorization in Identity Management and Single Sign-On. *Advances in Science, Technology and Engineering Systems (ASTES)*, 3(5):492–500, 2018.

[330] Furkan Alaca and Paul C. Van Oorschot. Comparative Analysis and Framework Evaluating Web Single Sign-On Systems. *ACM Computing Surveys (CSUR)*, 53(5):1–34, 2020.

[331] Mahyar Tourchi Moghaddam, Andreas Edal Pedersen, William Walter Lillebroe Bolding, and Torben Worm. A Performant and Secure Single Sign-On System Using Microservices. In *Proc. of the 38th ACM/SIGAPP Symposium on Applied Computing*, pages 1516–1519, 2023.

[332] Tri Waluyo et al. Comparative analysis of the performance of single sign-on authentication systems with openid and oauth protocols. *International Journal of Computer and Information Technology (2279-0764)*, 11(3), 2022.

[333] Joshua D. Drake and John C. Worsley. *Practical PostgreSQL.* " O'Reilly Media, Inc.", 2002.

[334] Josiah Carlson. *Redis in action.* Simon and Schuster, 2013.

[335] Kristina Chodorow and Michael Dirolf. *MongoDB: The Definitive Guide.* O'Reilly Media, Inc., 1st edition, 2010.

[336] Nginx. Nginx: Advanced Load Balancer, Web Server, & Reverse Proxy. https://www.nginx.com/. [Online; accessed 24-October-2023].

[337] Internet Engineering Task Force (IETF). The Transport Layer Security (TLS) Protocol (RFC 8446). https://www.nginx.com/. [Online; accessed 24-October-2023].

[338] Electronic Frontier Foundation (EFF). Certbot: A Free, Open-Source Software Tool for Automatically Using Let's Encrypt Certificates. https://certbot.eff.org/. [Online; accessed 24-October-2023].

[339] Internet Security Research Group (ISRG). Let's Encrypt: A Nonprofit Certificate Authority Providing TLS Certificates. https://letsencrypt.org/. [Online; accessed 24-October-2023].

[340] Software in the Public Interest, Incorporated. Debian Manpages Documentation. https://manpages.debian.org/. [Online; accessed 24-October-2023].

[341] JavaScript Framework. JavaScript Framework: Wikipedia Website. https://en.wikipedia.org/wiki/JavaScript. [Online; accessed 23-October-2023].

[342] React. React: The Library for Web and Native User Interfaces. https://react.dev/. [Online; accessed 23-October-2023].

[343] Redux. Redux: A Predictable State Container for JS Apps. https://redux.js.org/. [Online; accessed 23-October-2023].

[344] ChartJS. ChartJS: Open Source HTML5 Charts For Your Website. https://www.chartjs.org/. [Online; accessed 23-October-2023].

[345] Ejwa Software. Gitinspector Documentation. https://github.com/ejwa/gitinspector/wiki/Documentation. [Online; accessed 26-October-2023].

[346] Khronos Group, Incorporated and Silicon Graphics, Incorporated. OpenGL. https://www.opengl.org. [Online; accessed 26-October-2023].

[347] Andrew Caudwell. Gource Documentation. https://github.com/acaudwell/Gource/wiki. [Online; accessed 26-October-2023].

[348] Compositional Agile System (CAS) Platform. CAS Logger Plugins. https://github.com/cas-agile. [Online; accessed 15-December-2023].

[349] Colab: Collaborative Environment for Communities. Colab GitHub Repository. https://github.com/colab. [Online; accessed 19-December-2023].

[350] Mailman: The GNU Mailing List Manager. Mailman web site. https://list.org/. [Online; accessed 19-December-2023].

[351] Prezento: A jQuery Plugin To Showcase Your Web Designs. Prezento GitHub Repository. https://github.com/ivaldi/prezento. [Online; accessed 19-December-2023].

[352] Mezuro: A Free Software Platform For Code Metrics. Mezuro GitHub Repository. https://github.com/mezuro/. [Online; accessed 19-December-2023].

[353] Noosfero: A Web-based Social Platform. Noosfero GitHub Repository. https://github.com/noosfero/noosfero. [Online; accessed 19-December-2023].

[354] PagoPA, S.p.A. IO App: The Public Services Italian App. https://io.italia.it/. [Online; accessed 29-December-2023].

[355] European Commission. Updating the Open Source Software Licence EUPL to Further Facilitate the Sharing and Reuse of Software Developed by Public Administrations. https://joinup.ec.europa.eu/sites/default/files/custom-page/attachment/eupl_v1.2_en.pdf. [Online; accessed 27-December-2023].

[356] RedHat, Inc. Ansible Documentation: YAML Format. https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html. [Online; accessed 23-October-2023].

[357] Meta Platforms, Incorporated. Jest: A Delightful JavaScript Testing Framework. https://jestjs.io/. [Online; accessed 27-December-2023].

[358] Paolo Marzolo, Matteo Guazzaloca, and Paolo Ciancarini. "Extreme Development" as a Means for Learning Agile. In *Communications in Computer and Information Science*, pages 158–175. Springer, 2021.