Scrum@PA: Tailoring an Agile Methodology to the Digital Transformation in the Public Sector

Paolo Ciancarini [1,*,†] , Raffaele Giancarlo [2,†] and Gennaro Grimaudo [3,†]

1 Department of Computer Science and Engineering, University of Bologna, 40126 Bologna, Italy
2 Department of Mathematics and Computer Science, University of Palermo, 90123 Palermo, Italy; raffaele.giancarlo@unipa.it
3 Department of Engineering, University of Palermo, 90128 Palermo, Italy; gennaro.grimaudo@unipa.it
* Correspondence: paolo.ciancarini@unibo.it
† All authors contributed equally to the design and validation of the methods. RG and GG wrote a draft of the paper, that was integrated and revised by all authors.

Abstract: Digital transformation in the public sector provides digital services to the citizens aiming at increasing their quality of life, as well as the transparency and accountability of a public administration. Since adaptation to the citizens changing needs is central for its success, Agile methodologies seem best suited for the software development of digital services in that area. However, as well documented by an attempt to use Scrum for an important Public Administration in Italy, substantial modifications to standard Agile were needed, giving rise to a new proposal called improved Agile (in short, iAgile). Another notable example is the Scrum@IMI method developed by the City of Barcelona for the deployment of its digital services. However, given the importance of digital transformation in the public sector and the scarcity of efforts (documented in the scholarly literature) to effectively bring Agile within it, a strategically important contribution that Computer Science can offer is a general paradigm describing how to tailor Agile methodologies and, in particular, Scrum, for such a specific context. Our proposal, called Scrum@PA, addresses this strategic need. Based on it, 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. We show the validity of our proposal by describing how the quite successful Scrum@IMI approach can be derived from Scrum@PA. Although iAgile can also be derived from our paradigm, we have chosen Scrum@IMI as a pilot example since it is publicly available on GitHub.

Keywords: digital transformation; public sector; Agile; Scrum

1. Introduction

Although no universally accepted definition is available [1], the mission of Digital Transformation (DT, for short) in the Public Administration (PA, for short) can be summarized as the process of offering novel digital services that enhance the quality of life of citizens, adding the transparency and accountability of the PA.

Such a transformation is by no means simple, since it involves regulatory and social issues [2] as well as challenges for many areas of Computer Science [3]. Synthetically, one of its key features is the involvement of the citizens in the DT [4,5] and a very relevant measure of its success is the level of responsiveness it has in addressing the changing needs of citizens [6,7]. Quite surprisingly, although DT is a widely studied interdisciplinary subject, the public sector is somewhat neglected in the scholarly literature, as pointed out in [8]. Moreover, as evident from the presentation in [3], there is a great deal of ambiguity regarding which and how the techniques proper of Computer Science could be used for the successful realization of any DT in a PA context. In summary, the potential contributions that Computer Science can give are intuitively clear, yet they deserve to be better formalized

and understood. From now on, unless otherwise specified, DT refers specifically to the public sector.

One of the key technical aspects concerns which software engineering methodologies are best suited for the management and development of software projects in the DT. As justified in the next section, Agile methodologies seem to be the most appropriate, although it is not entirely clear how to apply them. This paper concentrates on such an aspect, assuming that the reader is familiar with notions and concepts of software development, particularly in regard to Agile methodologies. However, in order to make the reading of this manuscript appropriate for a wide audience, we have added a glossary of basic Scrum [9] terms in Appendix A. Moreover, for the convenience of the reader, we also provide additional references to introductory material regarding the subject [9–12].

1.1. Agile for Software Development in the DT: Motivation and Difficulties

We first provide motivation for the adoption of Agile methodologies in the DT and then outline some difficulties that have emerged in using standard, i.e., textbook or basic, Agile approaches for the DT.

1.1.1. Motivation

Due to its dinamicity and the requirement of flexibility, a DT strategy requires: (a) the rapid adoption of novel technologies, e.g., [13]; (b) the implementation of an inclusive coaching strategy that promptly makes available new digital competences to citizens, administrators, and policy makers, e.g., [14]; (c) the adoption of adaptive and flexible organizational models for the design, implementation, and deployment of digital services, e.g., [15]. Therefore, a conventional plan-driven approach to the design and implementation of digital services is not perceived as appropriate in this context [16]. Instead, a transition to Agile methodologies finds more and more support, e.g., [16–24], although such a transition is not so simple and it is proceeding very slowly [25].

1.1.2. Difficulties

In order to encourage the application of Agile methodologies in the DT, guidelines and recommendations are given by several PAs, e.g., the United Kingdom [26,27] and USA federal government [28,29]. For completeness, we mention also a recent review that illustrate challenges and best practices for the adoption of Agile project management in the public sector [30]. However, those recommendations and best practices are too generic and, in fact, there are very few concrete examples on how to adopt Agile methodologies in the public sector [31], since their standard versions need to be tailored to the context in which they are applied [32], in this case, the DT. This may be challenging, as well pointed out in a study reporting the development of an extension of standard Agile, denoted improved Agile (iAgile, for short) [33] for a specific PA. The context in which such a project management strategy has been first developed and then successfully applied is for an Italian Public Administration. 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 as inadequate in some of its key figures and one team was not judged as sufficient for realizing the project. The contribution of iAgile is to extend the responsibilities of some existing Scrum professional figures, introduce new ones, together with a set of teams whose coordination and cooperation go beyond Scrum of Scrum [34], and to redefine the workflow of the software development process. The interested reader can find details in [33], 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. Quite remarkably, such a city set up its own Agile policy [35] in order to address its specific, but rather broad, needs.

Such a policy is supported by the Scrum@IMI methodology, an extension of Scrum [9], 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 [36]. Moreover, it is fully documented in a public repository [37]. As in the case of iAgile, Scrum@IMI has requested the expansion of the responsibilities of standard Scrum roles and the introduction of new ones, as well as the use of more than one team.

1.2. Our Contribution: From Examples to the General Paradigm Scrum@PA

Given all of the above, it is quite natural to concentrate on Scrum and, given the importance and pervasiveness of the DT, to consider the problem of developing a technically sound, in terms of software engineering, paradigm describing how to adapt Scrum for the DT, accounting for the lessons learned from the two examples mentioned earlier. Such a paradigm must account for the major aspects characterizing the DT and be flexible enough to be transformed in a Scrum methodology for a specific PA; that is, it should be a general "blueprint" that an organization can use to obtain its own Scrum. Our contribution toward a solution to the posed problem is the proposal of Scrum@PA.

In particular, the experiences reported in the development of iAgile and Scrum@IMI indicate that, with respect to the standard Agile methodologies, new professional figures are needed. In this respect, Section 2 provides a taxonomy of those figures, generalizing the two mentioned examples. The next two sections provide a description of the Scrum@PA paradigm that we propose here, following a bottom-up approach; that is, in Section 3, the building blocks of our proposal are presented: the professional figures and the teams that can be considered for Scrum@PA, placed within the taxonomy outlined in Section 2. Section 4 is dedicated to the Scrum@PA process, including also the roles of key professional figures, teams and their interactions.

As for the validation of our approach, it is useful to recall that there are no accepted methodologies for the DT [38,39]. However, Scrum@PA being 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 [40]. For more complex cases, although iAgile can be derived from Scrum@PA, in order not to clutter this contribution with details, we opt to concentrate on Scrum@IMI, given also the fact that it is public. Indeed, in Section 5, we provide a description of Scrum@IMI in terms of our paradigm. The considerations above and such a derivation indicate that: (a) Scrum@PA specialization to specific contexts can be successfully carried out; (b) PAs that are willing to adopt the Scrum methodology have a "blueprint" of it, which they can adapt to their needs.

Finally, Section 6 gives some practical recommendations and outlines some directions of further investigation, whereas Section 7 offers our conclusions.

2. Agile for the DT: Professional Figures

Although there is plenty of literature that gives advice on how to transition to Agile, e.g., [41,42], and some documents that even deal with the PA contexts [28], the need for new professional figures to be added to the ones already part of standard Agile approaches, with particular attention to the DT, has been overlooked or only sketched, to the best of our knowledge. In view of the peculiarities of the DT, we provide a brief synoptic list of those figures in Section 2.1 and then they are placed in a taxonomy of job families in Section 2.2.

2.1. The Need for New Professional Figures

1. In the context of the PA, data are both an asset to exploit and a social "infrastructure" useful for policy makers and citizens to take informed decisions and improve the well-being of society [43,44]. However, for their profitable use, major problems arise in data governance and management [45], specifically in regard to quality assurance [46,47], security and privacy [48–50]. Therefore, professional figures with expertise in the

mentioned areas are needed. Some of those figures must be also proficient with rules and regulations such as the EU General Data Protection Regulation [51].

2. Since the services provided by the PAs should be valuable for the citizens, there has been a paradigm shift in the evaluation of how valuable a service is. As in the past, efficiency has a privileged position in such an evaluation, but it is by no means sufficient since a people-driven delivery model is more and more important [19,52,53]. Therefore, within the Agile methodologies, professional figures are needed, whose role is to obtain regular and continuous feedback from citizens and other stakeholders during all the phases of the development of a digital service.

3. As stated in the Introduction, citizen participation is central to the DT and in setting priorities in regard to service realization. Therefore, new professional figures are needed to educate citizens in the participatory policy process.

2.2. A Taxonomy

Here, we introduce two categories of professional figures and, for each of them, we define the corresponding job families. The first contains job families related to data management and information and communication technologies, accounting for point 1. in the previous sub-section. The second contains job families related to the creation of a software product and the various stages related to its life cycle. This accounts for points 2. and 3. in the previous sub-section.

- **Data management and ICT category**

  D.1 **Data Job Family**. Given the attention that needs to be placed on data (see the previous section), 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. For the DT, their semantics change since they must have the ability to improve the architectural design of public services while accounting for the needs of citizens. By way of example, a **Chief Technology Officer** is needed, possibly in a leadership position.

  D.3 **Quality, Assurance and Testing (QAT) Job Family**. When the classical Agile methodologies are extended to the DT, the semantics of figures in this family change, since they must focus on managing the complexity of the testing process in the PA. 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. By way of example, a **Chief Quality Assurance Testing Officer** is needed, possibly in a leadership position.

- **Product Creation and Support Category**

  P.1 **Product Design and Delivery Job Family**. As in the previous family, also here the semantics of the classic professional figures change. Indeed, 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, these figures must have the ability to consider many of the aspects related to the management and continuous and evolutionary maintenance of the digital service portfolio offered by the 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.

3. From Generic to Specific Professional Figures: The Scrum Case

Following the motivation given in the Introduction regarding our choice of Scrum, we now detail job figures for it coming from the families described earlier. We identify two types of them: 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.

3.1. Leaders

Those figures are schematically provided in Figures 1 and 2, suitably placed in the job family taxonomy that we have described in Section 2. The roots of those two trees are offspring of the taxonomy root node, which 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.
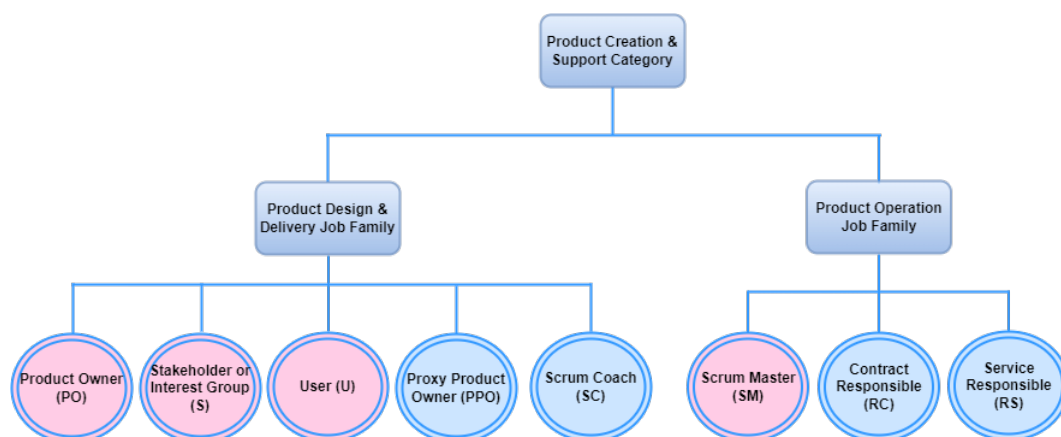
Figure 1. Leading professional figures for a PA within the taxonomy: product creation and support category. Starting at the root, the first two levels of the tree are a verbatim rendering of the taxonomy described in Section 2, while the leaf nodes represent the set of leader professional figures involved in the entire life cycle of the project that the PA intends to realize following the Scrum methodology. In the leaf 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 and 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 been given in point (D.1) in Section 2.2. They must be coherently placed within the generic description of this figure. They are responsible for overseeing and managing all data-related activities within an organization to ensure that data are managed efficiently, securely and in compliance with regulations. They coordinate the collection and integration of data from various sources, ensuring that they are accurate, complete and compliant with privacy regulations and relevant laws, while ensuring the consistency and efficiency of the processes of the organization that they represent. To this end, they define and implement

appropriate data storage and protection policies, through policies such as encryption, restricted access and constant monitoring. They implement regular backup strategies to ensure continuous data availability and plan recovery procedures in case of emergencies. They monitor and improve data quality through cleansing, normalization and standardization processes, e.g., open data standards (see, e.g., [54]). They collaborate with data analysts to extract meaningful information and support data-driven decision making. In terms of training and communication, they provide support and training to staff on the importance of data management and communicate 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 [45].
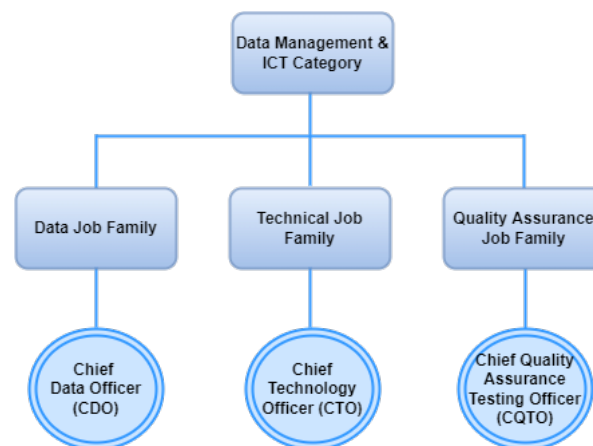
Figure 2. Leading professional figures for the PA within the taxonomy: data management and ICT category. The figure legend is as in Figure 1.

– **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 2.2. They must be coherently placed within the following generic description of this figure. They are responsible for playing a key role in shaping the technological direction of an organization, and have various responsibilities related to technology management and implementation, ensuring that technologies are aligned with the objectives of the organization and contribute to its overall success. They are involved in defining the technology strategy of the organization, helping to establish long-term goals in line with corporate objectives. They monitor technology trends and evaluate new opportunities and innovative solutions to improve business efficiency and competitiveness. In terms of security, they are responsible for implementing appropriate measures to protect corporate data from external and internal threats. In terms of technology resource management, they ensure the efficient allocation of technology resources, including budget, personnel and infrastructure, to ensure the achievement of technology objectives. They collaborate with other stakeholders and teams, such as marketing, sales and finance, to ensure alignment of the technology strategy with the overall needs of the organization. They monitor the performance of technology systems, identify possible areas for improvement and implement solutions to optimize the processes of the organization. In addition, they interact 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 2.2. They must be coherently placed within the following generic description of this figure. They are responsible for a role that may vary depending on the organization and the specific sector but, in general, it refers to a manager responsible for managing and supervising the entire quality assurance and testing process within an organization. They define the necessary policies and procedures to ensure the quality of the product/service offered by the company. They supervise personnel involved in quality assurance and testing, ensuring that they are well trained and follow best practices. To this end, they collaborate with business leaders to develop a quality assurance and testing strategy in line with overall business objectives. They identify and implement testing tools, technologies and methodologies, such as quality metric 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. They collaborate with other departments, such as software development, to ensure effective communication and a mutual understanding of needs and objectives. Finally, they provide 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 and Support Category**

  - – **Product Design and 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, to optimize 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 is certainly required at the beginning of the project and during the entire product backlog life cycle. 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 [35]. As for iAgile, the PO is now a board of people, including stakeholders and domain experts [33,55]. It is worth mentioning that an earlier instance of such a model has been used in [56].
    - ∗ **Stakeholder or Interest Group** (S, for short). We note that the role of the stakeholders turns out to be the same as in the standard Scrum methodology, i.e., managers and customers.
    - ∗ **User, i.e, Citizen** (U, for short). 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, 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 [36]. 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 2.2. They must be coherently placed within the following generic description of this figure. In general, in accordance with financial regulations and within the terms of ongoing contracts, they manage and plan the necessary resources within the Scrum project. Under delegation from the contractor, they are 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 [57,58]. Upon the occurrence of appropriate specific circumstances, they are 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 can be delegates 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 as that of the standard Scrum methodology.
* **Contract Responsible** (RC, for short). The specifics of this figure in regard to the DT have been given in point (P.2) in Section 2.2. They must be coherently placed within the following generic description of this figure. The main responsibility of the RC is to keep the contract and contractor relationships under control. They are 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. Their 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, they represent 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 2.2. They must be coherently placed within the following generic description of this figure. They are 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, they are "responsible" for ensuring its adequate performance. They collaborate in defining the criteria and requirements of the service during the build phases. Our proposal of adapting Scrum for the PA consists of a group of teams, which support the planning and execution of the transition from product increment to service, providing the necessary information for the full implementation, such as service level agreements (SLAs), optimal timing for maintenance execution, user volume estimates, resource sizing for the production environment, and requirements for the basic information model and the general data protection regulation [51]. They ensure that the project also meets non-functional criteria specified 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. They collaborate 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 their functions, they are completely free to attend as many events and meetings as they see fit. An example of this professional figure can be found in the Scrum@IMI framework. In that framework, their role, together with the PO, has 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 properties, which had been included in the product backlog.
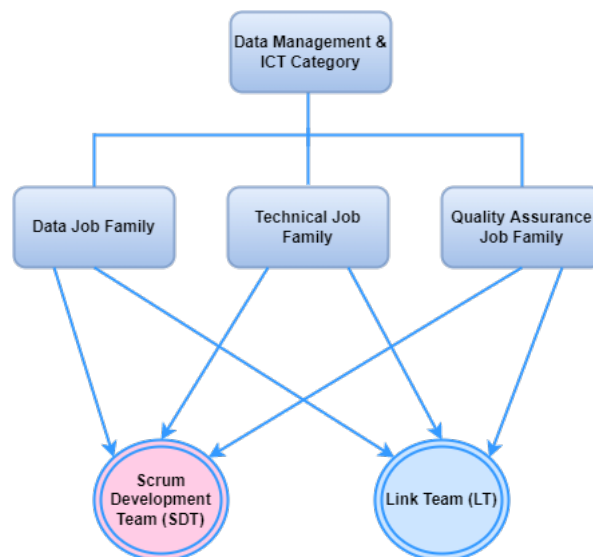
3.2. Teams and Their Members

Those teams, and their members, are schematically provided in Figures 3 and 4, where members composing each team are suitably placed in the job family taxonomy that we have described in Section 2. The roots of those two trees are offspring of the taxonomy root node, which 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.

* **Data Management and 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 the PA. We note that the functions of the SDT members are similar to those of the standard 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 and 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 3.1, can be part of this team.
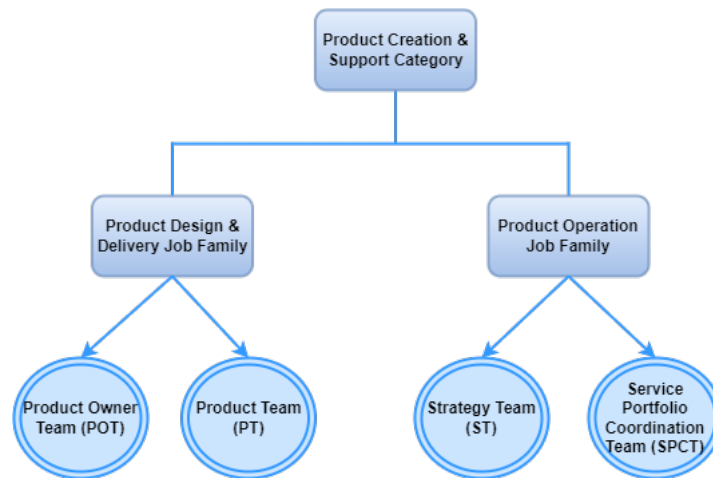
Figure 3. Taxonomy of teams and their members' professional figures, within the Scrum methodology, for their adoption in the context of the PA: product creation and support category. The figure legend is as in Figure 1.

Figure 4. Taxonomy of teams and their members' professional figures, within the Scrum methodology, for their adoption in the context of the PA: data management and ICT category. The figure legend is as in Figure 1.

- **Product Creation and Support Category**
  - **Product Design and 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 their 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

maximizing 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 [59]. 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 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). This team has to solve the problems arising from the management and integration of services within the services portfolio offered by the 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, 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. A General Paradigm for Scrum in the DT: Scrum@PA

Once its building blocks are specified, we now describe Scrum@PA, starting with an overview of it.

4.1. Scrum@PA Overview

The development of a software product, via the Scrum methodology, is schematically represented in Figure 5. It consists of "an idea" (left), a series of Scrum sprints (center) and the delivery of the product (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 citizens, civil servants and stakeholders that address 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. 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 and product compliance and verification. Schematically, the process is represented in Figure 6, including interactions between each pair of components. The incoming edge,

labeled transform in Figure 5, has been omitted to enhance the readability of Figure 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.
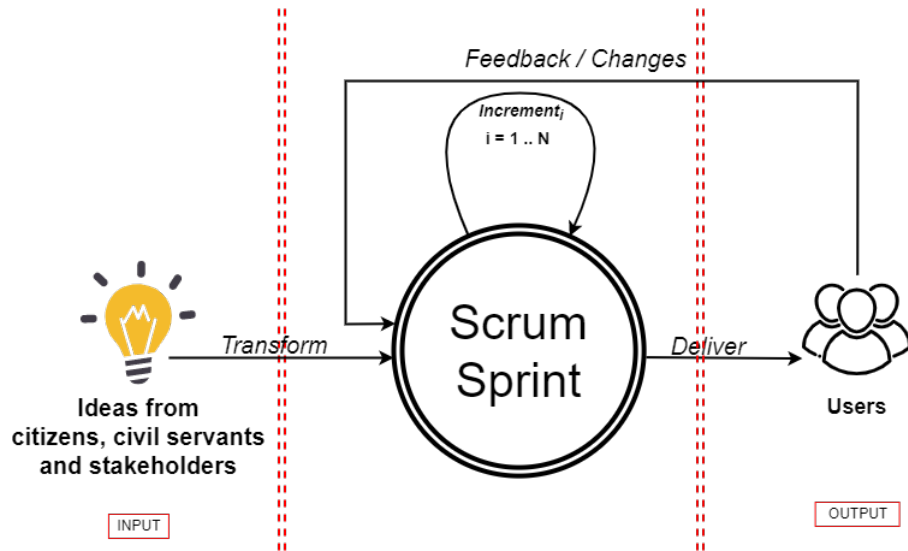
Figure 5. A generic scheme of the software development process via the Scrum methodology. The components of this figure are illustrated in the main text.
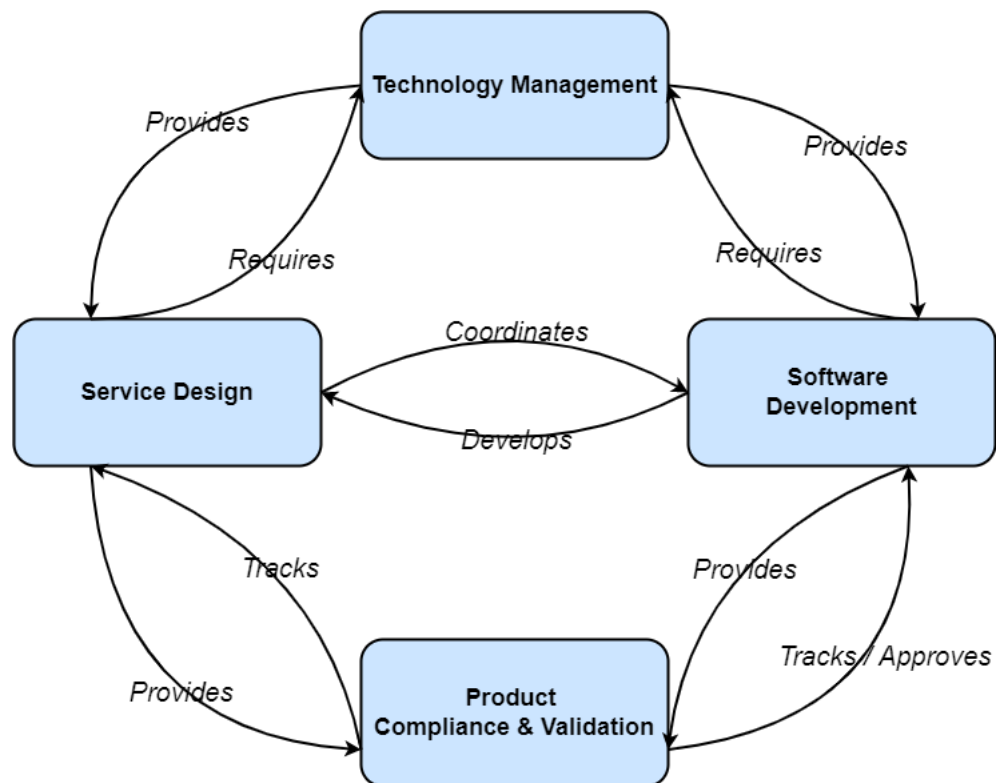
Figure 6. A paradigmatic Scrum model for the PA: Scrum sprint. The activities carried out within each component, together with the leader figures and teams involved, are described in Section 4.2, while the interactions, here encoded by edges, are described in Section 4.3.

Scrum@PA is presented as follows. Section 4.2 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 4.3 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 4.4 provides a synopsis of the main responsibilities of leaders and teams with respect to some fundamental Scrum artifacts (see the glossary in Appendix A), in terms of a RACI matrix. It is useful to recall that, in general, a RACI matrix [60] is a concise way of indicating who is responsible, and to what degree, in 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 the Scrum sprint varies from two weeks to a month, depending on the complexity of the project at hand (see [61] 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 [35] and that a more detailed description, such as the one given for Scrum@IMI in [36], would considerably expand the description of Scrum@PA, making the key points of our contribution difficult to enucleate and then specialize in specific contexts.

### *4.2. Scrum@PA Details: Activities, Leaders and Teams*

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-organization of the development team; initially identify, within this component, 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. Completion of project deliverables, iteratively, incrementally and within each sprint, giving priority according to the relevance of each deliverable. Creation of documents that accompany 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 3.2 that the PT may be composed of one or more SDTs, based on the complexity of the development process, at this stage of abstraction, we use one PT and two leader professional figures: the SM and SC. The interactions within these teams and leader professional figures are as follows, with Figure 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. They also inform 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.

SOFTWARE DEVELOPMENT
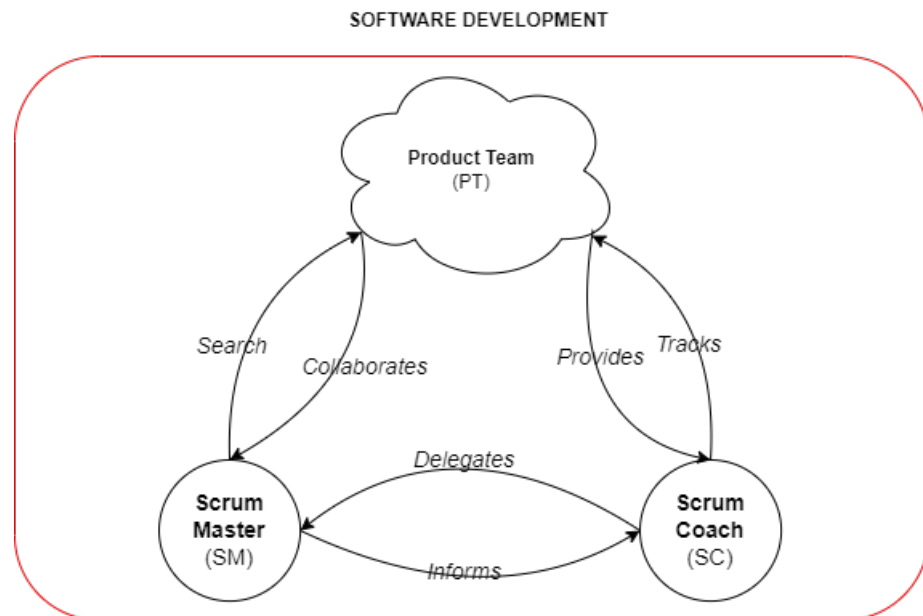
Figure 7. Scrum@PA software development component. A node symbolized by a cloud represents a team, while a node symbolized by a circle represents a leader professional figure. This component consists of a single team, i.e., PT, and two leaders, i.e., SM and SC. The description of the interactions (represented by edges) between the team PT and the leaders is in the main text.
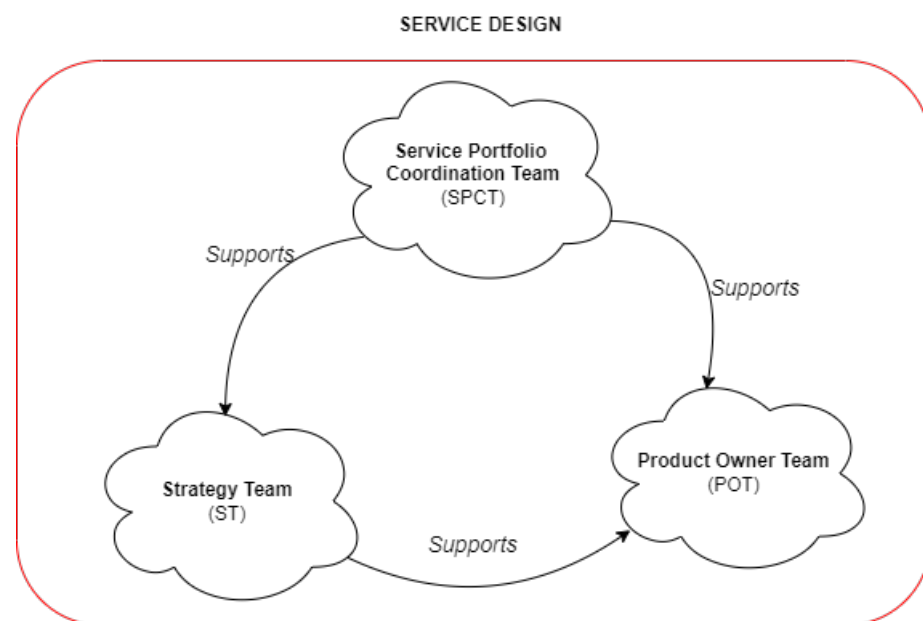
- **Service Design**

  - **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 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 needs of the users, resulting in changes in requirements, must be accounted for 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, stakeholders are kept 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 8 giving a schematic representation of this component.

* SPCT. This team supports the ST in defining non-functional requirements that are related to the integration of the services offered by the PA. Those requirements may give rise to user stories that are included in the product backlog. SPCT also supports the stakeholders afferent to the POT, which represent 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, which are then encoded in epics and user stories. These 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 8. Scrum@PA service design component. Following the same notation as in Figure 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 the 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 3.2. Figure 9 gives a schematic representation of this component.
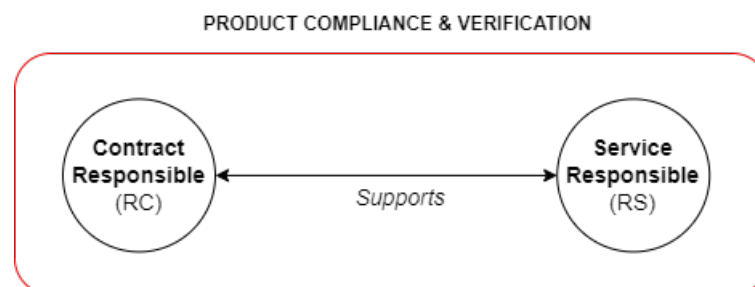
Figure 9. Scrum@PA technology management component. Following the same notation as in Figure 7, this component consists of a single team, i.e., LT.

- **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 toward 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 10 giving a schematic representation of this component.
    * RC. With the support of the RS, the RC ensures compliance with all contractual aspects and the 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 10. Scrum@PA product compliance and validation component. Following the same notation as in Figure 7, this component consists of two leaders, i.e., RC and RS. The description of the interactions among leaders (edges) follows the main text.

4.3. Scrum@PA Details: Interactions Among Teams

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 and 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. It interacts with the technology management component in terms of resource 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 and 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.

4.4. 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. 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 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 1. Scrum@PA responsibility matrix. This table defines the responsibilities of the leading professional figures and teams within Scrum@PA, providing a comprehensive overview of who is responsible (R), accountable (A), consulted (C) and informed (I) for each artifact included within this Agile methodology. Empty cells are assumed as informed (see above).

Teams and Leaders	Scrum Artifacts					
	Contract	Product Backlog	Sprint Backlog	DOD	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

5. The Scrum@IMI Framework of the Barcelona Municipality 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 3 and 4, and [35,36]).

Technically, the Scrum sprint in Figure 5, for the Scrum@IMI methodology, consists of four phases: on-boarding, sprint 0, sprint i (iterated) and service transition, schematically depicted in Figure 11. Moreover, the incoming edge, labeled transform in Figure 5 has been omitted to enhance the readability of Figure 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.
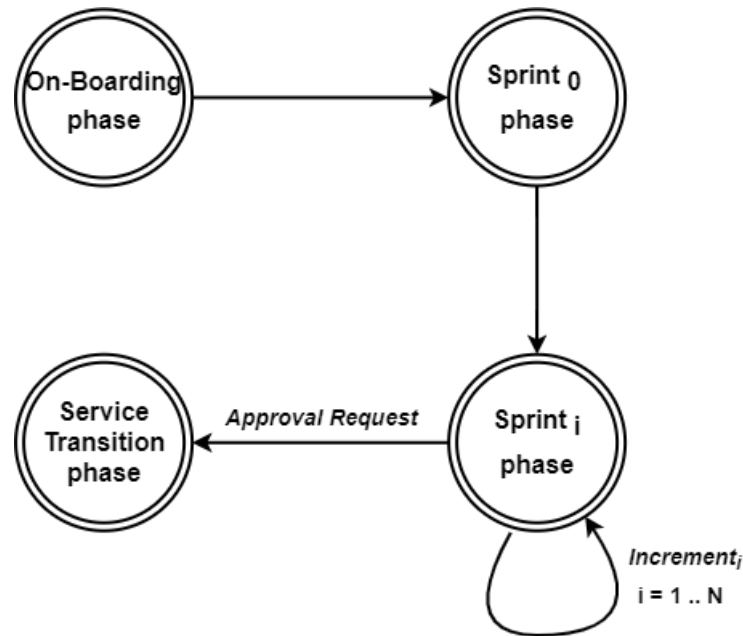
Figure 11. Scrum@IMI methodology: Scrum sprint. Starting from the on-boarding phase, the arrows provide a temporal succession of the four phases. Each phase is detailed in the main text, together with the edges that enter such a Scrum sprint.

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 [62], regulations [25], privacy, service and data ownership [63,64].

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 a "Meta Team", which is indicated by a red cloud in the relevant figures (see Figure 12 for an example).

5.1. On Boarding

The activities and teams involved in the on-boarding phase are described synoptically in Figure 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 12. Service design: activity (1.a), contributing to the Scrum@IMI activities described in dots (1.a), (1.b) and (2.b) in Figure 12. Software development: activity (1), contributing to the Scrum@IMI activities described in dots (1.a), (1.b) and (2.b) in Figure 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 12. Product compliance and verification: activity (1) and (2), corresponding to dots (3.a) and (1.b) in Figure 12.
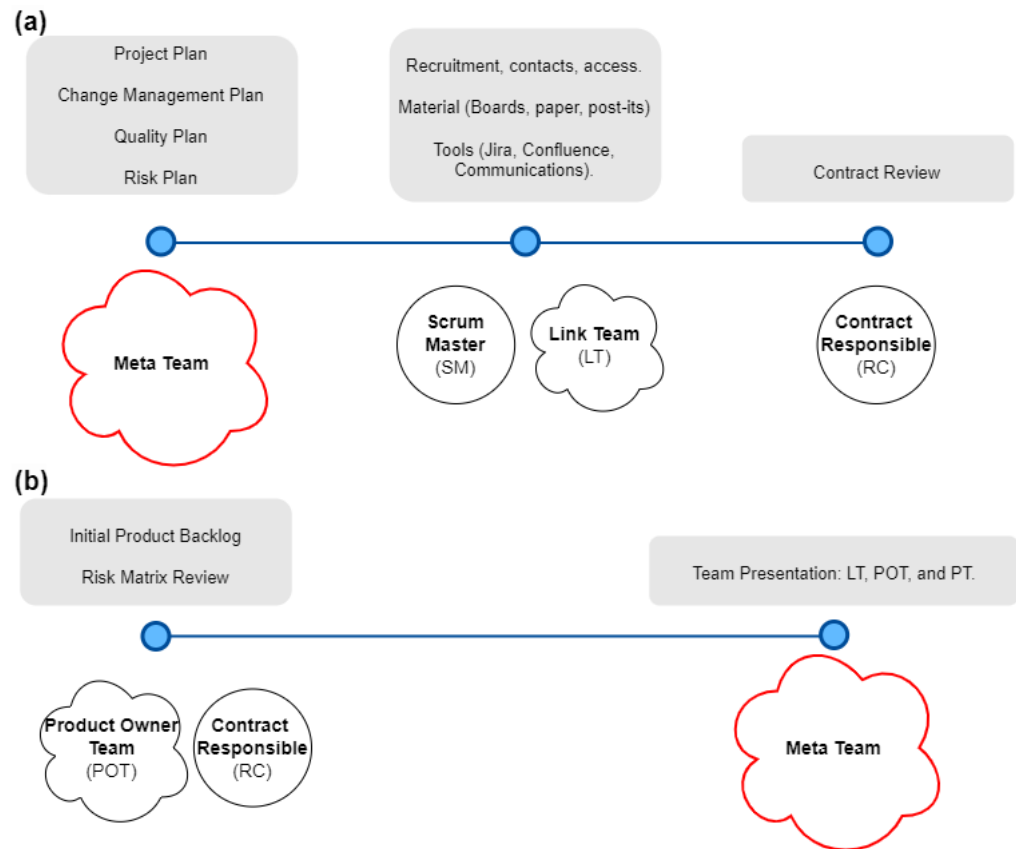
Figure 12. The on-boarding phase of the Scrum@IMI methodology. The teams and the leader professional figures are indicated as in Figure 7. For each line, dots are considered as numbered from left to right, but such an order does not imply time of execution priority. For each dot, the activities are above it, while who is involved in carrying them out is below it. Subfigure (**a**) shows the process managing the project up to the contract review; subfigure (**b**) shows the initialization of the Product Backlog.

5.2. Sprint Zero

The activities and teams involved in the sprint 0 phase are described synoptically in Figure 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 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 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 13. Technology management: activity (1), corresponding in part to the Scrum@IMI activities described in dots (1.c) and (3.c) in Figure 13. Product compliance and verification: activity (3), corresponding to dots (1.b) and (1.c) in Figure 13.
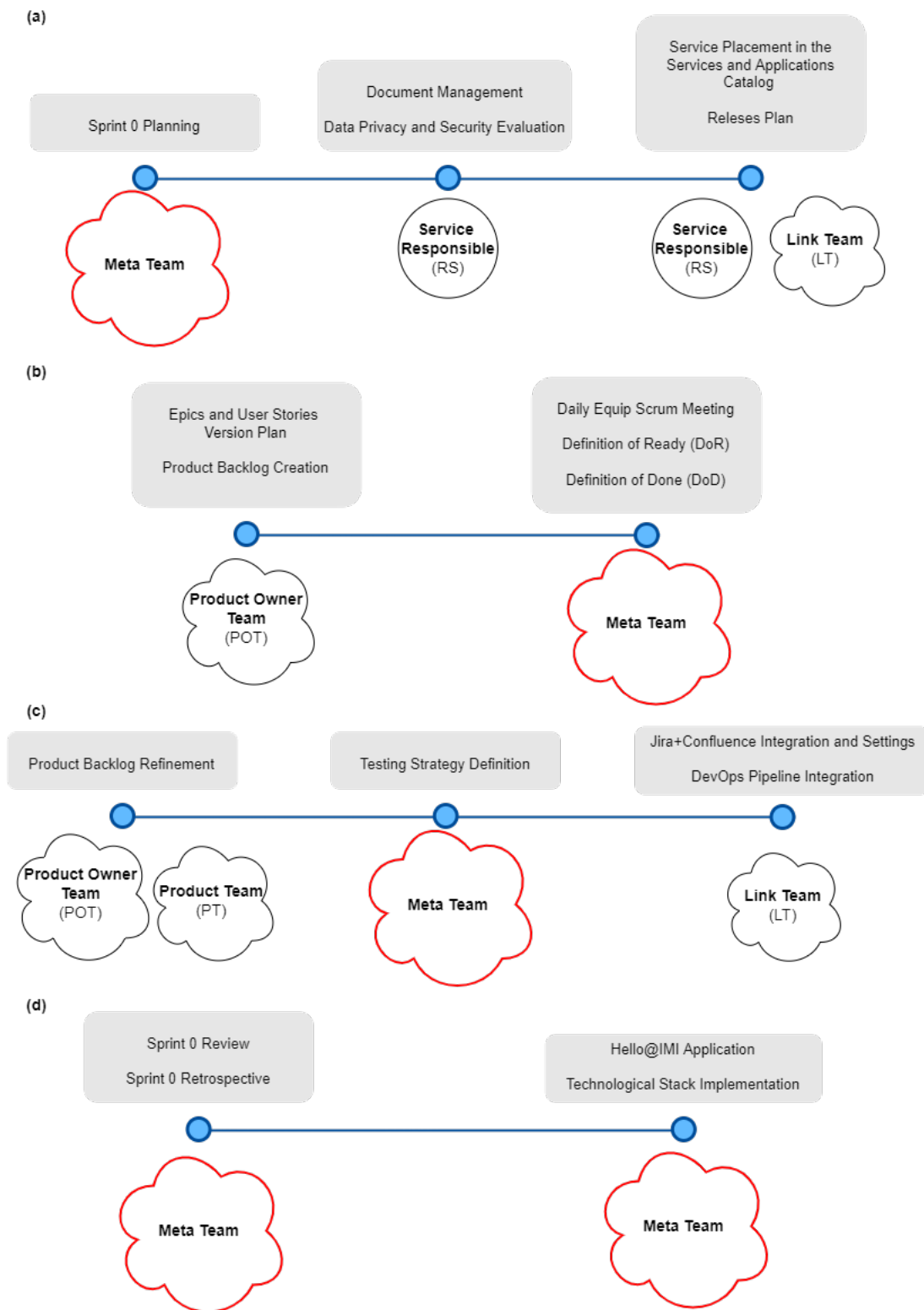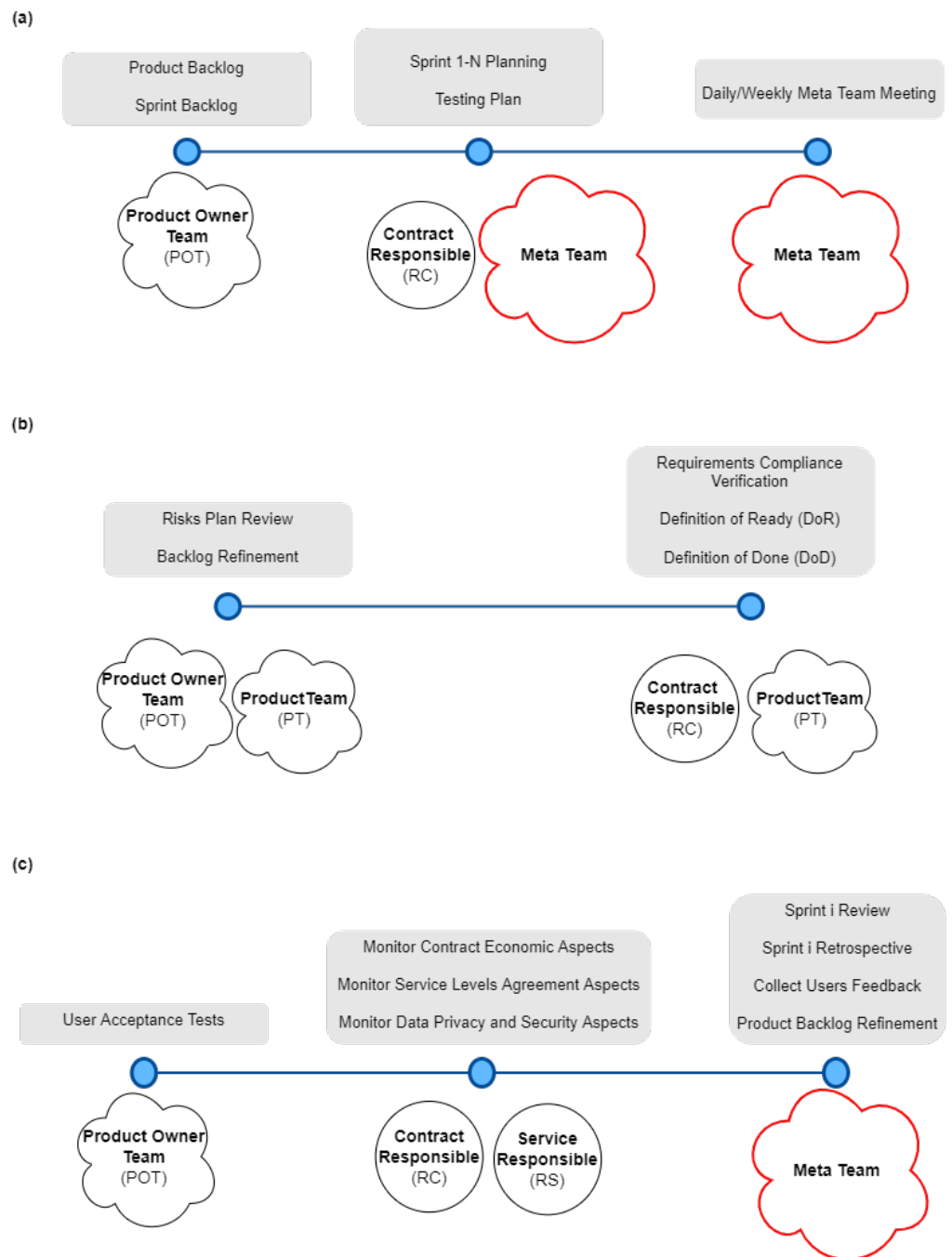
Figure 13. The sprint 0 Phase of the Scrum@IMI framework. (**a**) Planning; (**b**) backlog creation; (**c**) backlog refinement; (**d**) conclusion of sprint 0. The figure legend is as in Figure 12.

5.3. Sprint i

The activities and teams involved in the sprint i phase are described synoptically in Figure 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 a "Release Sprint" and its product is provided to the service transition phase (see the next sub-section).

Figure 14. The sprint i Phase of the Scrum@IMI framework. (**a**) creation of the sprint backlog; (**b**) backlog refinement; (**c**) sprint conclusion. The figure legend is as in Figure 12.

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 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 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 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 14. Product compliance and verification: activities (2) and (3), contributing to the Scrum@IMI activities described in dots (1.b), (2.b) and (3.b) in Figure 14.

5.4. Service Transition

The activities and teams involved in the service transition phase are described synoptically in Figure 15. This phase consists of the transition to the 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 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 15. Software development: activities (1) and (2), contributing to the Scrum@IMI activities described in dot (1.a) in Figure 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 15. Product compliance and 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 15.

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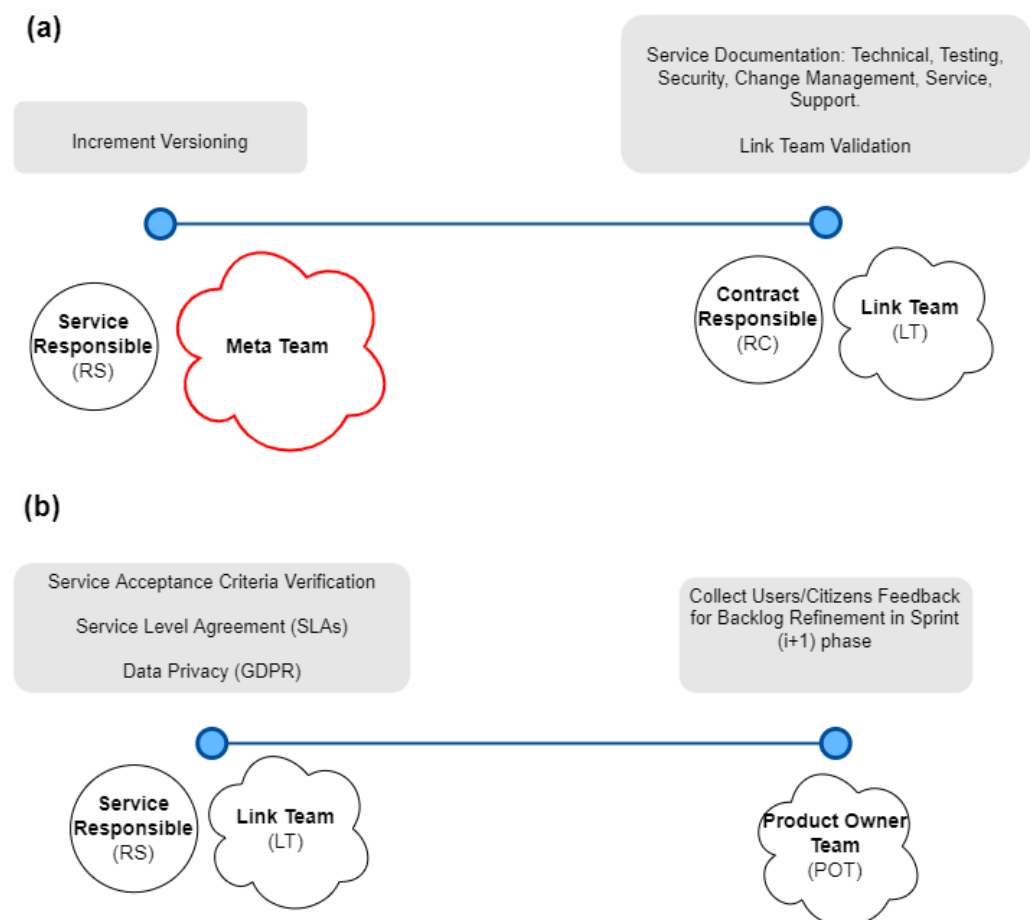

Figure 15. The service transition phase of the Scrum@IMI framework. (**a**) versioning each increment; (**b**) service validation and citizens' feedback. The figure legend is as in Figure 12.

6. Discussion

We now provide a comparison of Scrum@PA with existing Agile methodologies and then highlight some recommendations for its application.

6.1. Comparison with Existing Methods

The few Scrum methodologies that have been tailored to the PA are special cases of Scrum@PA. Therefore, no comparison is needed. However, one aspect deserves further elaboration. Our paradigm is scalable. Indeed, depending on the service ecosystem that the PA intends to adopt, it can be customized in terms of multiple teams, making the set of professional figures involved in the management and development of the relevant public service more or less articulate. We found only a mention that SAFe could be used, but with no technical details (see, i.e., [65]). For the sake of completeness, we review the SAFe framework in its essential configuration and compare it with our proposed 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. It 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 and product owner. The professional figures and their roles characterizing 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 1).
- **Product Management**. It owns and prioritizes the program backlog. In our proposal, it would correspond to the PO professional figure (see Figure 1).
- **Business Owners**. They are critical stakeholders in the ART. In our proposal, they would correspond to the POT members' professional figures (see Figure 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).
- **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), while, for evaluation aspects, these figures should correspond mainly to the RS professional figure (see Figure 1).

In light of this, we can observe some overlap between what is in our proposal and what is in the essential SAFe framework. Nevertheless, our proposal offers a greater level of detail in the professional figures and teams involved (see Figures 1–4) and in their main interactions (see Figures 6–10).

6.2. Practical Recommendations

When advising a public administration on adopting the Scrum@PA framework for a digital transformation project, there are some key suggestions, which are also based on the iAgile and Scrum@IMI experiences.

- Ensure that all team members and stakeholders receive adequate training on the Scrum@PA framework, emphasizing the roles, ceremonies and artifacts involved. Provide specialized training for the product owners and other related roles to deepen their understanding of the problem being addressed.

- Establish clear governance structures that support the Scrum@PA approach. Clearly articulate the objectives and goals of the digital transformation project. This will help in aligning the efforts of all teams and stakeholders toward a common vision.
- Create cross-functional teams and host them in the administration premises. Form cross-functional teams that encompass diverse skills and expertise relevant to the project. If the project targets the citizens, define an informal focus group of citizen stakeholders. The developers should meet in the premises of administration, if possible, to manage more effectively the requirements and the interaction with stakeholders.
- Organize teams into groups that plan, commit and execute together, using release train engineers. This promotes better coordination and synchronization across teams. Regular meetings of the service portfolio coordination team should facilitate communication between teams and address cross-team dependencies.

The adoption of Scrum@PA is a significant change, and it is important to approach it with patience and a commitment to continuous learning and improvement.

7. Conclusions

We have developed a framework, referred to as Scrum@PA, for implementing the Agile Scrum methodology in the context of DT. This approach involves new professional roles particularly suited for the PA that are not available in the traditional Scrum methodology. In particular, we have provided a taxonomy of them. Then, we have described the Scrum@PA framework and its components, including leader professional figures, teams and their interactions. Finally, 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.

Overall, the key finding of this research is that it is possible to provide a general blueprint of how to specialize Scrum to project management and software development for the PA. Such a key finding opens the avenue to further directions of research, which we now outline.

The first direction concerns training. As well pointed out in [33], a knowledge of software engineering fundamentals is essential for all the professional figures involved in an Agile project for the PA, with varying degrees of depth; that is, the people developing software or in a technical management role must be very proficient in software engineering, while other figures must have some knowledge about it. On the other hand, even technical figures must be aware of legal and regulatory aspects of the PA characterizing a given project that they are involved in.

The second direction concerns validation and evaluation methods used to assess an Agile approach for the PA. As pointed out in a recent review on the subject [39], there is currently no set of commonly accepted criteria. One difficulty in formalizing these criteria is certainly the heterogeneity of how Agile techniques have been applied so far to the PA. Regarding this, the generality and homogeneity offered by Scrum@PA may help in the identification of a homogeneous and coherent set of criteria for validation and evaluation.

A third direction in this area consists in developing a maturity model for teams following the Scrum@PA framework; we could follow, for instance, the approach suggested in [66].

A fourth concern is if SCRUM@PA can smoothly scale to several teams working on several products or services. In order to shed light on this topic, the approaches described in [67] may be very useful.

A fifth direction consists of providing the Scrum@PA teams and, more in general, Agile teams working for the public sector, with a set of collaborative integrated development environment tools for creating digital services (see, e.g., [68,69]). This part is under development.

Appendix A. A Basic Glossary of Scrum Terms

The following glossary, reported in Table A1, provides some basic Scrum terms, in alphabetic order. It is based on the glossary provided in [70].

Table A1. Basic Scrum terms, in alphabetic order.

Term	Description
Artifact	An Agile Scrum artifact refers to the information that the Scrum team and stakeholders use to define the product being developed, the actions required to produce it and the actions performed during the project. The primary Scrum artifacts include the contract, product backlog, sprint backlog, definition of done (DOD, for short) and increments.
Compliance	Based on product increment, to meet the expectations and needs of stakeholders, it represents an overall assessment of product conformity based on the project-specific requirements included in the contract, including those related to regulations, guidelines or sector standards.
Contract	IT contract means any material contract for the provision of information and communication technology services (including hardware, software and databases) and maintenance services to parts of the contracting station.
Definition of Done	It is the formal description of the completed increment that meets the quality measures required for the product, providing transparency by sharing a common understanding of the work completed. Failure to meet the DoD means that the product backlog item cannot be released or presented at the sprint review.
Developer	Any member of a Scrum team committed to creating a usable increment in a sprint, regardless of specialization.
Increment	A Scrum artifact that captures all the valuable work produced by developers during a sprint. The sum of all increments forms a product.
Product Backlog	It is an ordered list of all the work that is required to be carried out to create, maintain and sustain a product. It is managed by the product owner.
Product Backlog Refinement	It is the activity where the product owner and developers collaborate to add more detail to the product backlog items.
Product Owner	It is responsible for maximizing product value by managing and expressing business and functional expectations to developers.
Scrum	It is a lightweight framework that helps teams to solve complex problems through adaptive solutions and generated value.
Scrum Master	It is a role, within a Scrum team, responsible for guiding, coaching, teaching and assisting a Scrum team and its environments toward a proper understanding and use of Scrum.
Scrum Team	A self-managed team comprising a Product Owner, some developers, and a Scrum Master.

Table A1. *Cont.*

Term	Description
Sprint	A sprint is a time-boxed Scrum event whose duration depends on the project. It contains and facilitates other Scrum events and activities, and is followed consecutively without any intermediate gaps.
Sprint backlog	It is a Scrum artifact managed by the developers that provides an overview of the development work required to achieve a sprint's goal. It includes a forecast of functionality and the work needed to deliver that functionality.
Sprint goal	A sprint's purpose is to solve a business problem. Features can be modified during the sprint to achieve the goal.
Sprint planning	It is a time-boxed meeting event that lasts for a maximum of 8 h, where the Scrum team comes together to decide which items from the product backlog should be prioritized for the upcoming sprint. During this event, the team inspects the work thoroughly and designs a plan for the sprint backlog.
Sprint retrospective	A 3-hour or less Scrum event for ending a sprint, inspecting it and planning for future improvements.
Sprint review	It is a Scrum event held for a time-boxed period of 4 h or less to finalize the development work of a sprint. It involves examining the product increment, assessing the impact of the work carried out on progress toward the product goal and updating the product backlog to maximize the next period's value.
Stakeholder	It is a person who is external to the Scrum team with specific product interests and knowledge needed for incremental discovery. It is represented by the product owner and actively engaged with the Scrum team during the sprint review.
User	It is an end-user, regardless of being a consumer of the product or an administrator.

References

1. Mahraz, M.I.; Benabbou, L.; Berrado, A. A Systematic Literature Review of Digital Transformation. In Proceedings of the International Conference on Industrial Engineering and Operations Management, Bangkok, Thailand, 5–7 March 2019; IEOM Society International: Southfield, MI, USA, 2019; pp. 917–931.
2. Mountasser, T.; Abdellatif, M. Digital Transformation in Public Administration: A Systematic Literature Review. *Int. J. Prof. Bus. Rev.* **2023**, *8*, e02372. [CrossRef]
3. Ciancarini, P.; Giancarlo, R.; Grimaudo, G. Digital Transformation in the Public Administrations: A Guided Tour For Computer Scientists. *IEEEAccess* **2024**. [CrossRef]
4. Filograna, A.; Smiraglia, P.; Gilsanz, C.; Krco, S.; Medela, A.; Su, T. Cloudification of Public Services in Smart Cities: The Clips Project. In Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 153–158.
5. McBride, K.; Aavik, G.; Toots, M.; Kalvet, T.; Krimmer, R. How Does Open Government Data Driven Co-Creation Occur? Six Factors and a 'Perfect Storm'; Insights From Chicago's Food Inspection Forecasting Model. *Gov. Inf. Q.* **2019**, *36*, 88–97. [CrossRef]
6. Vestues, K. Using Digital Platforms to Promote Value Co-Creation: A Case Study of a Public Sector Organization. Ph.D. Thesis, NTNU, Trondheim, Norway, 2021.
7. Ylinen, M. Incorporating Agile Practices in Public Sector IT Management: A Nudge Toward Adaptive Governance. *Inf. Polity* **2021**, *26*, 251–271. [CrossRef]
8. Reis, J.; Amorim, M.; Melão, N.; Matos, P. Digital Transformation: A Literature Review and Guidelines for Future Research. In *Trends and Advances in Information Systems and Technologies*; Rocha, Á., Adeli, H., Reis, L.P., Costanzo, S., Eds.; Springer: Cham, Switzerland, 2018; pp. 411–421.
9. Maximini, D. *The Scrum Culture: Introducing Agile Methods in Organizations*; Springer: Berlin/Heidelberg, Germany, 2018.
10. Shore, J.; Warden, S. *The Art of Agile Development*; Theory in practice; O'Reilly Media, Incorporated: Sebastopol, CA, USA, 2008.
11. Kniberg, H. *Scrum and XP from the Trenches*; Lulu.com: Morrisville, NC, USA, 2015.
12. Anderson, D.J.; Carmichael, A. *Essential Kanban Condensed*; Blue Hole Press: Chicago, IL, USA, 2016.
13. Baheer, B.A.; Lamas, D.; Sousa, S. A Systematic Literature Review on Existing Digital Government Architectures: State-of-the-Art, Challenges, and Prospects. *Adm. Sci.* **2020**, *10*, 25. [CrossRef]

14. Liu, C.; Zowghi, D. Citizen Involvement in Digital Transformation: A Systematic Review and a Framework. *Online Inf. Rev.* **2022**, *47*, 644–660. [CrossRef]

15. Mohagheghi, P.; Lassenius, C. Organizational Implications of Agile Adoption: A Case Study From the Public Sector. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, 23–28 August 2021; pp. 1444–1454.

16. Dwivedula, R.; Bolloju, N. Transitioning from Plan-Driven Methods to Agile Methods-Preparation for a Systematic Literature Review. In Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 10–12 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 944–950.

17. Aleinikova, O.; Kravchenko, S.; Hurochkina, V.; Zvonar, V.; Brechko, O.; Buryk, Z. Project Management Technologies in Public Administration. *J. Manag. Inf. Decis. Sci.* **2020**, *23*, 564–576.

18. Bogdanova, M.; Parashkevova, E.; Stoyanova, M. Agile Project Management in Public Sector—Methodological Aspects. *J. Eur. Econ.* **2020**, *19*, 283–298. [CrossRef]

19. Bounabat, B. From E-Government to Digital Government: Stakes and Evolution Models Du E-Gouvernement au Gouvernement Digital: Enjeux et Modèles d'Évolution. *Electron. J. Inf. Technol.* **2017**, *10*, 1–20.

20. Chaudhury, K.; Barua, A.; Deka, R.; Gogoi, T.; Gupta, A.C.; Pyarelal, S. Reforming and Strengthening Digital Service Delivery: Case of Government of Assam, Proc. National Conference for e-Governance, Mumbai, India, 2020. Available online: https://www.researchgate.net/publication/342344029_Reforming_and_Strengthening_Digital_Service_Delivery_Case_of_Government_of_Assam_Sub-Theme_Improving_Service_Delivery (accessed on 30 January 2014).

21. Jonathan, G.M. Digital Transformation in the Public Sector: Identifying Critical Success Factors. In Proceedings of the European, Mediterranean, and Middle Eastern Conference on Information Systems, Dubai, United Arab Emirates, 9–10 December 2019; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 223–235.

22. Mantovani Fontana, R.; Marczak, S. Characteristics and Challenges of Agile Software Development Adoption in Brazilian Government. *J. Technol. Manag. Innov.* **2020**, *15*, 3–10. [CrossRef]

23. Mubarkoot, M. Assessment of Factors Influencing Adoption of DevOps Practices in Public Sector and Their Impact on Organizational Culture. In Proceedings of the International Conference on Science and Technology (ICST), Ternate, Indonesia, 27–28 October 2021; Volume 2, pp. 475–483.

24. Nachit, H.; Jaafari, M.; El Fikri, I.; Belhcen, L. Digital Transformation in the Moroccan Public Sector: Drivers and Barriers. *SSRN* **2021**. [CrossRef]

25. Barcelona Ciutat Digital. Agile Methodologies at Barcelona City Council. Available online: https://www.barcelona.cat/digitalstandards/en/agile-methodologies/0.1/_attachments/barcelona_agile_methodologies_0.1.en.pdf (accessed on 20 January 2023).

26. UK Government. United Kingdom Government—Service Manual, 2023. Available online: https://www.gov.uk/service-manual (accessed on 2 May 2023).

27. Agile Delivery Community. Agile and Government Services: An Introduction. Available online: https://www.gov.uk/service-manual/agile-delivery/agile-government-services-introduction (accessed on 10 January 2024).

28. Powner, D. *Software Development: Effective Practices and Federal Challenges in Applying Agile Methods*; United States Government Accountability Office: Washington, DC, USA, 2012.

29. U.S. Government Accountability Office. Agile Assessment Guide: Best Practices for Agile Adoption and Implementation. Available online: https://www.gao.gov/products/gao-20-590g (accessed on 10 January 2024).

30. Abdullah, P.P.; Raharjo, T.; Hardian, B.; Simanungkalit, T. Challenges and Best Practices Solution of Agile Project Management in Public Sector: A Systematic Literature Review. *JOIV Int. J. Inform. Vis.* **2023**, *7*, 606–614. [CrossRef]

31. Edison, H.; Wang, X.; Conboy, K. Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review. *IEEE Trans. Softw. Eng.* **2021**, *48*, 2709–2731. [CrossRef]

32. Jovanović, M.; Mesquida, A.L.; Mas, A.; Colomo-Palacios, R. Agile Transition and Adoption Frameworks, Issues and Factors: A Systematic Mapping. *IEEE Access* **2020**, *8*, 15711–15735. [CrossRef]

33. Benedicenti, L.; Messina, A.; Sillitti, A. iAgile: Mission Critical Military Software Development. In Proceedings of the 2017 International Conference on High Performance Computing & Simulation (HPCS), Genoa, Italy, 17–21 July 2017; pp. 545–552.

34. Mundra, A.; Misra, S.; Dhawale, C.A. Practical Scrum-Scrum Team: Way to Produce Successful and Quality Software. In Proceedings of the 2013 13th International Conference on Computational Science and Its Applications, Ho Chi Minh City, Vietnam, 24–27 June 2013; pp. 119–123.

35. Barcelona City Council. Barcelona City Council: Agile Policy. Available online: https://ajuntamentdebarcelona.github.io/ethical-digital-standards-site/agile-methodologies/0.1/policy.html (accessed on 7 January 2024).

36. Barcelona Municipal Institute of Information Technology (IMI). Scrum@IMI Framework. Available online: https://ajuntament.barcelona.cat/imi/sites/default/files/marc_de_treball_scrumimi_per_proveidors.pdf (accessed on 11 May 2023).

37. Ajuntament de Barcelona. Ajuntament de Barcelona GitHub Repository. Available online: https://github.com/AjuntamentdeBarcelona/agile-methodologies-bcn-en (accessed on 20 November 2023).

38. Fernandes, M.C.; Alencar, A.J.; Schmitz, E.A.; da Silva, M.F.; Stefaneas, P.S. Evaluation of Agile Software Projects in the Public Sector: A Literature Review. *J. Softw.* **2016**, *11*, 312–325. [CrossRef]

39. Baxter, D.; Dacre, N.; Dong, H.; Ceylan, S. Institutional Challenges in Agile Adoption: Evidence from a Public Sector IT Project. *Gov. Inf. Q.* **2023**, *40*, 101858. [CrossRef]

40. Vacari, I.; Prikladnicki, R. Adopting Agile Methods in the Public Sector: A Systematic Literature Review. In Proceedings of the International Conferences on Software Engineering and Knowledge Engineering (SEKE), Pittsburgh, PA, USA, 6–8 July 2015; KSI Research Inc.: Pittsburgh, PA, USA, 2015; pp. 709–714.

41. Rigby, D.K.; Sutherland, J.; Takeuchi, H. Embracing Agile: How to Master the Process That's Transforming Management: Experimentation. *Harv. Bus. Rev.* **2016**. Available online: https://hbr.org/2016/05/embracing-agile (accessed on 10 January 2024).

42. Shokoya, A. *Waterfall to Agile: A Practical Guide to Agile Transition*; TamaRe House: London, UK, 2012.

43. Harrison, T.; Canestraro, D.; Pardo, T.; Avila-Maravilla, M.; Soto, N.; Sutherland, M.; Burke, G.B.; Gasco, M. Applying an Enterprise Data Model in Government. In Proceedings of the 20th Annual International Conference on Digital Government Research, Dubai, United Arab Emirates, 18–20 June 2019; ACM: New York, NY, USA, 2019; pp. 265–271.

44. Hastings, J.S.; Howison, M.; Lawless, T.; Ucles, J.; White, P. Unlocking Data to Improve Public Policy. *Commun. ACM* **2019**, *62*, 48–53. [CrossRef]

45. Nie, Y.; Talburt, J.; Li, X.; Xiao, Z. Chief Data Officer (CDO) Role and Responsibility Analysis. *J. Comput. Sci. Coll.* **2018**, *33*, 4–12.

46. Goericke, S. *The Future of Software Quality Assurance*; Springer Nature: Berlin/Heidelberg, Germany, 2020.

47. Basu, A. *Software Quality Assurance, Testing and Metrics*; PHI Learning Pvt. Ltd.: Delhi, India, 2015.

48. Venkataramanan, N.; Shriram, A. *Data Privacy: Principles and Practice*; CRC Press: Boca Raton, FL, USA, 2016.

49. Stallings, W. *Information Privacy Engineering and Privacy by Design: Understanding Privacy Threats, Technology, and Regulations Based on Standards and Best Practices*; Addison-Wesley Professional: Boston, MA, USA, 2019.

50. Whitman, M.E.; Mattord, H.J. *Principles of Information Security*; Cengage Learning: Singapore, 2021.

51. European Commission. General Data Protection Regulation (GDPR), 2022. Available online: https://eur-lex.europa.eu/eli/reg/2016/679 (accessed on 25 June 2022).

52. Battisti, D. The Digital Transformation of Italy's Public Sector: Government Cannot Be Left Behind! *JeDEM—eJournal eDemocracy Open Gov.* **2020**, *12*, 25–39. [CrossRef]

53. Cordella, A.; Paletti, A. Government as a Platform, Orchestration, and Public Value Creation: The Italian Case. *Gov. Inf. Q.* **2019**, *36*, 101409. [CrossRef]

54. Charalabidis, Y.; Zuiderwijk, A.; Alexopoulos, C.; Janssen, M.; Lampoltshammer, T.; Ferro, E. *The World of Open Data: Concepts, Methods, Tools and Experiences*; Springer: Berlin/Heidelberg, Germany, 2019.

55. Benedicenti, L.; Ciancarini, P.; Cotugno, F.; Messina, A.; Sillitti, A.; Succi, G. Improved Agile: A Customized Scrum Process For Project Management In Defense And Security. In *Software Project Management for Distributed Computing: Life-Cycle Methods for Developing Scalable and Reliable Tools*; Springer: Cham, Switzerland, 2017; pp. 289–314.

56. Upender, B. Staying agile in government software projects. In Proceedings of the Agile Development Conference (ADC'05), Denver, CO, USA, 24–29 July 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 153–159.

57. Beck, K. Embracing Change With Extreme Programming. *Computer* **1999**, *32*, 70–77. [CrossRef]

58. Beck, K. *Extreme Programming Explained: Embrace Change*; Addison-Wesley Professional: Boston, MA, USA, 2000.

59. Sillitti, A.; Succi, G. Requirements Engineering For Agile Methods. In *Engineering and Managing Software Requirements*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 309–326.

60. Institute, P.M. *A Guide to the Project Management Body of Knowledge: PMBOK Guide*; Project Management Institute: Newtown Square, PA, USA, 2013.

61. Mergel, I.; Gong, Y.; Bertot, J. Agile government: Systematic literature review and future research. *Gov. Inf. Q.* **2018**, *35*, 291–298. [CrossRef]

62. de Treball, À.; Bria, F.; Rodriguez, P.; Bain, M.; Batlle, J.; Bastida Vila, A.; Barandiaran Fernández, X.E.; Boada Pla, M.; Marpons Ucero, G.; Roca Vilalta, X.; et al. Barcelona City Council ICT Public Procurement Guide, 2018. Available online: https://ajuntament.barcelona.cat/contractaciopublica/sites/default/files/barcelona_city_council_ict_public_procurement_guide.pdf (accessed on 10 January 2024).

63. Comissionat/da de Tecnologia i Innovació Digital; Comissió d'Economia i Hisenda; Àrea de Treball, Economia i Planificació Estratègica. *A Government Measure for Open Digitisation: Free Software and Agile Development of Public Administration Services*; Barcelona Ciutat Digital: Barcelona, Spain, 2017.

64. Monge, F.; Initiative, B.H.C.L.; Barns, S.; Kattel, R.; Bria, F. *A New Data Deal: The Case of Barcelona*; Working Paper Series (No. WP 2022/02); Technical report; UCL Institute for Innovation and Public Purpose, UCL: London, UK, 2022.

65. Fagarasan, C.; Cristea, C.; Cristea, M.; Popa, O.; Mihele, C.; Pisla, A. The Delivery of Large-Scale Software Products through the Adoption of the SAFe Framework. In Proceedings of the International Conference on Development and Application Systems (DAS), Suceava, Romania, 26–28 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 137–143.

66. Looks, H.; Fangmann, J.; Thomaschewski, J.; Escalona, M.J.; Schön, E.M. Towards improving agility in Public Administration. *Softw. Qual. J.* **2024**, *32*, 283–311. [CrossRef]

67. Block, S. *Large-Scale Agile Frameworks: Agile Frameworks, Agile Infrastructure and Pragmatic Solutions for Digital Transformation*; Springer Nature: Berlin/Heidelberg, Germany, 2023.

68. Ciancarini, P.; Missiroli, M.; Poggi, F.; Russo, D. An Open Source Environment for an Agile Development Model. In Proceedings of the Open Source Systems: 16th IFIP WG 2.13 International Conference, OSS 2020, Innopolis, Russia, 12–14 May 2020; Proceedings 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 148–162.

69. Marzolo, P.; Guazzaloca, M.; Ciancarini, P. "Extreme Development" as a Means for Learning Agile. In Proceedings of the 1st International Conference on the Frontiers of Software Engineering, Innopolis, Russia, 17–18 June 2021; Volume 1523 of Communications in Computer and Information Science; Springer: Berlin/Heidelberg, Germany, 2021; pp. 158–175.

70. Ken Schwaber. Scrum.org: The Home of Scrum. Available online: https://www.scrum.org/ (accessed on 4 January 2024).