Barcelona Ciutat Digital

Agile Methodologies at Barcelona City Council

The Open Digitisation Programme from Barcelona City Council's Office for Technology and Digital Innovation

This Guide was prepared by a team led by Francesca Bria, the Commissioner for Technology and Digital Innovation. Its members included Francesca Bria, Paco Rodríguez, Malcolm Bain, Joan Batlle, Ana Bastide Vila, Xabier Barandiaran Fernández, Marius Boada Pla, Guillem Marpons, Xavier Roca Vilalta, Xavier Bes Segovia, Josep Carles Collazos, Joan Domènech Bas, Oscar Sanz García, Carlos Echevarría Mesegur, LLuis Girona Frisach, Anna Majó Crespo, Tomás Gea Calza, Javier Ruiz, Gemma Galdon, John Michaelides, Esther Bretschneider.

September 2017

Ajuntament de Barcelona

CONTENTS

1 Introduction and Context

This is document supports the Barcelona City Council Technology Code of Practice, and provides a practical explanation of the Agile transformation concept. It expands on the principles guiding the new leadership at the Municipal Institute of Information Technology (IMI) in terms of technology and innovation: interoperability, agility, re-use, ethics, and open knowledge and technology. These principles and practices stem from the City Council Agile Digital Transformation Plan of September 2017.

It seeks to guide and contribute to implementation by the IMI – in conjunction with the city's leaders and authorities – of the strategic aim of Barcelona City Council to undertake digital service projects using Agile methodologies. Together with the principle of technological sovereignty and responsible data management, these agile methodologies make up the main aspects of the medium-term outlook on agile digital transformation in the city of Barcelona.

The goal of this document is therefore to provide managers in municipal bodies and the IMI with a basic understanding of agile methods and demonstrate how the agile life-cycle differs from traditional approaches; to explain the IMI agile method and how it will be implemented by integrating it into the framework for new projects and for maintenance of digital services that already exist; and to provide guidelines for managing development projects and maintaining services based on agile methodologies. Generally-speaking, it aims to help encourage and manage this cultural change at the IMI and the units of Barcelona city from a perspective of agile digital transformation.

2 General Principles of Agile Development

2.1. Concepts and Definitions

The development of agile software is a philosophy and agile is an umbrella term for a series of methods and approaches that share certain common characteristics. Several agile methods exist, of which the main ones are *scrum and extreme programming (XP).*

Generally-speaking, they are a set of methods used in the field of software development and maintenance based on iterative short-term processes (typically lasting from one to four weeks) that lead to the initial delivery of a partial but operational product and various consecutive versions with increasingly improved performance. Through constant iterations, these methodologies seek to provide value from the very beginning of a project, as well as undertake ongoing assessment of the product. Their goal is to introduce improvements and ongoing product evolution until a final result of excellent quality is achieved that fully responds to all user requirements.

Such iterative strategies allow risks to be minimised because each iteration is viewed as a miniature project and includes all the necessary stages: planning, requirement analysis, design, coding, user testing and documentation. Hence, any implementation problems, adaptation to requirements or risks in a project come to light earlier and the corrective measures are less costly and more immediate than in a traditional development project (in which they tend to come to light during the final stage, following months of evolution).

Furthermore, agile methodologies are focused on user satisfaction because they require active user participation in the project during both conceptualisation and development (via validation of the partial deliveries). This ensures that the final product responds to the needs of the user and meets user expectations.

All this is based on the so-called "*Manifesto for the Agile Development of Software*", of 2001:

Manifiesto *ÁGIL*

We are uncovering better ways of developing software by doing it and helping others do it.

Through this work, we have come to value:

Individuals and interactions above processes and tools. Working software over comprehensive documentation. Customer collaboration over contract negotiation.
Responding to change over following a plan.

That is, while there is value in the items on the right, we value the items on the left more.

http://agilemanifesto.org/iso/ca/manifesto.html

The Agile Manifesto proposes 12 principles that should be followed by agile methodologies:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2. We accept that requirements change, even late in development. Agile processes harness change for the customer's competitive advantage.

3. We deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.

4. Business people and developers must work together daily throughout the project.

5. Projects are built around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7. Working software is the primary measure of progress.

8. Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.

10. Simplicity, or the art of maximising the amount of work not done, is essential.

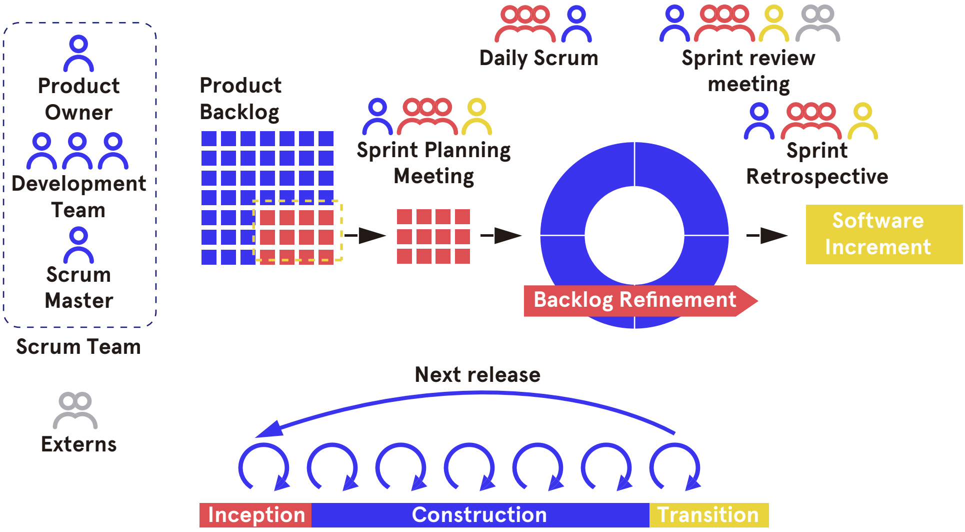
11. The best architectures, requirements and designs emerge from self-organising teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

http://agilemanifesto.org/iso/ca/priciples.html

The key to the Agile model is that an overall project is divided into a series of short-term development cycles (broadly referred to as "iterations" or – in scrum terminology – "sprints") each of which is approximately two to four weeks in length. For each iteration, the development team essentially carries out the same activities as it would if using the cascade model (i.e. planning, design, coding, testing and rollout) but according to agile working practices.

The image below shows an example of the life-cycle of a project based on the scrum agile model, together with the players involved in each stage. The various concepts introduced in the image will be explained in detail in subsequent sections of this document.



The usual activities within the scrum agile method are as follows:

1. Discovery
2. *Backlog*
3. Iterations/sprints

The **discovery** activity defines the minimum viable product (MVP) that must be produced. This is based on workshops and interviews that produce a **"product vision",** which becomes the basic definition of the system to be developed.

The product backlog is a prioritised list of all the estimated individual elements for completing the overall project; in other words, the distinguishable (programmable) elements that make up the product vision.

For each sprint (a development cycle of 2-4 weeks), the development team prepares a sprint backlog. This document states which development items in the product backlog will be implemented during the current sprint (starting with the items of highest priority). In other words, it is a "to-do" list for the current sprint.

The development team will design, program and test each one of the development items listed in the sprint backlog during the 2-4 weeks of the sprint. It should be noted that the team typically meets once a day to ensure that all the activities in the sprint move forward according to plan.

The result of each sprint should be a potentially shippable product increment, in which an increase in value or functionality can be seen when compared with the previous sprint.

Together, the three activities defined above form the **Agile cycle**, which is repeated until the result desired by the customer is obtained (or the budget/time allocated to the project expires).

2.2. Benefits

From among the benefits brought by working with an agile methodology, the following should be highlighted:

- **Focus on value and customer satisfaction** via the rapid and continuous delivery of useful software. This provides the customer with an ongoing vision of the product and its evolution until the final product is obtained.

- **Focus on the users.** Greater emphasis is placed on people and interactions than on the process and tools.Customers, developers and testers interact constantly to provide visions, assessments and improvements for the product.

- **Earlier deliveries.** The created software is delivered frequently (in weeks rather than months) via the various sprints, which means there is an evolutionary increment in each new delivery over the previous one.

- **Foreseeable costs and planning.** The cost is foreseeable and limited to the amount of work that the team can undertake because each review of the backlog calculates the development costs for the work packages included.

- **Commitment from the stakeholders.** An agile method requires close and daily cooperation with the users, both to correctly meet the project requirements and to verify the increase in product value from each sprint.

- **Transparency.** Progress on the project is the result of the entire process itself, meaning that the various players involved can see the progress at any given time.Furthermore, communication between those players is more direct ("face-to-face") and constant throughout the project.

- **Increased quality.** An agile method produces a constant focus on technical excellence and quality design. The use of specific tools and techniques, such as continuous integration or TDD (test-driven development), provides ongoing quality improvement in the product.

- **Rapid reaction to change.** An iterative cycle allows for regular adaptation to changing circumstances, such as the identification of potential changes or improvements to the product before finishing it. This process also allows late changes to the requirements to be accepted.

> **"Development productivity was increased, the transparency of development activities was enhanced and the relative proportion of administrative work was reduced (by up to 25%).The increased efficacy allows the public authority to develop more digital services on a limited budget".**
>
> *Finnish agency responsible for driving licences,*
> *quoted in Nuottila et al. (2013), about an Agile project in the Transport Department.*

2.3. Implications and Challenges

Without going into detail on the impacts of agile methodologies on organisations and their differences from traditional methodologies, such as the cascade method, the implementation of the former implies a major change in project management, the way the customer's staff (in this case, the public authorities) get involved and the way in which services and project results are contracted and delivered by the supplier.

As regards the organisation of projects, it should be noted that the results delivered when using this methodology are not necessarily the final results or even those that were defined at the start of the project. In contrast, the software resulting from the development process consists of versions stemming from an iterative process based on the needs of users and their response to the initial versions of the product.

For the customer (user), Agile places emphasis on collaboration. During the delivery life-cycle or iteration, both parties (customer and supplier) work together during each stage to achieve the objectives. Collaborative effort will be greater during the design stage, when the parties work together to determine the functional and non-functional requirements (such as the user experience, UX) of the iteration or deliverable result. Agile contracting therefore requires much more intense dedication (people and time) from the customer when compared with the traditional cascade contracting method.

In terms of the contracting process (beyond the impacts mentioned above), the customer does not contract a definitive and defined solution (from the start) but rather contracts the development of individual aspects of a solution that evolve over time and are incrementally incorporated to meet identified needs (user stories) and the general objectives of the customer. This usually means that the manner in which technology services are contracted when using an agile methodology is usually based on time and materials (contracting based on the time and resources to be used), which may vary according to the characteristics of the project.

Furthermore, the delivery and acceptance criteria for results must be more flexible due to the changes in planning stemming from changes to the functional and technical specifications that may result from the feedback provided by users on each iteration or development cycle.

Hence, we must work on the challenges posed by this development methodology if we wish to reap its benefits. The ongoing development of skills within the customer and supplier teams, their involvement in the project and the management of this personnel are key aspects of this methodology. Defining the scope and size of the projects and development teams are other criteria to be considered when applying this methodology. In large organisations with complex IT systems, where traditional methodologies are the mainstay, one of the most significant challenges is how to integrate and combine new agile practices into the environments and processes that already exist.

3

Agile and Public Administration

3.1. Agile Practice in Public Administration

The uptake of agile methods has been very slow in the public sector considering that the Agile Manifesto was drafted in 2001. Nonetheless, several cases involving the implementation of agile methodologies have been documented. These include Finland (a software development project for managing driving licences) and, closer to home, Andalusia (with two projects by the Ministry of Culture and Sport – eBOJA and TOPOS). The first was a development project for a system for using the official regional gazette and the second was an infrastructure project for the management of the regional ministry's systems. In the United States, the healthcare.gov website was redesigned using this methodology and the Defence Department has recognised the agile methodology and its instructions for the procurement of systems (Instruction 5000.2 of 2015) that support it. The UK Driver and Vehicle Standards Agency has successfully implemented an MOT management system with agile methodologies and short-term contracts.

In Spain, Catalonia has also conducted a pilot scheme applying scrum to a real project at the IMI to identify the implications from use of this "Agile Framework" and, above all, to reveal the current flaws in terms of both knowledge and infrastructure.

The conclusions from these projects indicate that implementation of this methodology has been successful but challenging (in general, and also regarding specific issues) for the public entities responsible for the projects. This approach allowed the teams to draw up an initial project plan that would guide the project and also correct the course taken based on the results. It also offers relevant and useful information on project status at all times, both for the team and for those responsible for management and supervision. It can be seen as a learning process in which the teams of users and developers improve their skills over time. In addition, this methodology converts estimation and planning efforts into an ongoing and participatory process that involves the whole team, rather than being a single stage at the start of the project, which is the case in other traditional methodologies such as the cascade method.

More generally-speaking, various authorities are in the process of implementing these methodologies, such as those in the United Kingdomo (*https://www.gov.uk/service-manual/agile-delivery*), New York (*http://playbook.cityofnewyork.us/*) and San Francisco (*http://digitalservices.sfgov.org*, for example, at the Human Services Agency and the Department of Community Housing and Development). These initiatives base the delivery of innovative digital services that are cheaper and more user-focused on agile methodologies from the outset.

3.2. Important Considerations

As stated previously in the section entitled "General Principles of Agile Development", agile methods have various consequences and pose specific implementation challenges in the development of digital services for public administration.

As regards the organisation of projects, all the agents involved in the project should agree on which method has to be used, the definition and allocation of duties or roles, especially within the "customer" department or unit (usually the project owner), the entity managing the contract (at Barcelona City Council, this will usually be the IMI) and the supplier, and how these roles should be undertaken. This choice of methodology and definition of roles is more or less difficult depending on the organisational complexity of the public authority in question and the integration of any new services with existing systems or platforms and other ongoing projects. Public authorities have many systems and services already in place, meaning that co-existence with traditional projects can be complicated.

Furthermore, it will be important to have access to stable project teams from both the customer/user and the developer. Technical solvency in terms of knowledge about agile methods and knowledge management is, again, an essential aspect for guaranteeing the success of projects in the short, medium and long term. Furthermore, when applying the scrum methodology, agile teams must coordinate with the teams working on existing or cross- departmental projects within the public authorities.

It will also be essential to guarantee the time spent by "customer/user" teams (personnel from the units and entities of the public authorities) during the evolution and development of the project: it is not simply a question of specifying a series of functionalities and awaiting delivery. The agile methodology requires more intensive communication or interaction to be encouraged between user and developer in order to reach the objectives for each iteration. With Agile, user departments are actively involved throughout the process and developers have a greater presence in the daily lives of those users in order to build a product vision and user stories, leading to feedback on each iteration.

One of the most relevant aspects for a public authority is the process and obligations with regard to the procurement of services, which must meet public procurement regulations that are not necessarily well-suited to this type of methodology. Given the characteristics of the above-mentioned agile methodologies, increased flexibility must be sought when contracting agile technology services (e.g. by dividing the project into lots or activities, clearly specifying the methodology to be used and the tasks allocated to each party, and being more flexible in terms of the final specifications and the delivery and acceptance of each stage/activity or iteration) and when overseeing contract performance once awarded.

In conclusion, it is important to recognise the need to work on a change of culture within public authorities as well as on agile transformation, because that change usually comes more slowly and with greater difficulty than in the private sector.

4 Barcelona City Council Agile Policy

This section presents the general principles and guidelines of the Barcelona City Council Agile Policy, which have been drawn up according to the Barcelona City Council digital service standards within the framework provided by the Digital Transformation Plan for Barcelona City Council. The most important targets with regard to agile methodologies are the following:

• PLACE GREATER EMPHASIS ON THE END USER:

- Give decision-making power to the end user.
- Be more flexible and deliver greater value.

• SHORTEN DEVELOPMENT TIMES:

- Simultaneously improve technical quality.
- Reliably deliver more often in order to bring forward the return on investment.

• TRANSFORM THE ORGANISATIONAL CULTURE:

- Place emphasis on collaboration and transparency.
- Focus the approach more towards value by maintaining control of developments.

• IMPLEMENT CONTINUOUS IMPROVEMENT:

- Systemise continuous improvement based on short review and adaptation cycles.
- Provide the organisation with active agents for systematic improvement and teams.

The implementation of these principles and guidelines will be carried out through the deployment of agile digital transformation projects by the City Council, mainly managed by IMI. These projects will allow resources to be dedicated, infrastructures created and skills acquisition to carry out these transformations. In this way, the management of the change caused by the application of these practices within the City Council will be carried out in an iterative way through concrete projects. Some projects and their extensions (for example, Decidim.Barcelona) already largely meet the guidelines indicated here, others will achieve them through a more progressive implementation process.

4.1. General Principles and Guidelines of the Barcelona City Council Agile Policy

As stated in the document entitled "Guidelines for the Provision of Agile Digital Services by Barcelona City Council", the general principles in agile development within departments of the City Council are as follows:

A) INTERDISCIPLINARY COLLABORATION

Multidisciplinary teams are created with people from various fields or working environments in which every member is fully involved and which include the roles and specialities needed to develop applications, services and processes. Only one or two people are directly responsible, but team success is the equal responsibility of each team member.

B) SELF-ORGANISED TEAMS

Teams that organise themselves autonomously and flexibly can create new approaches and independently adapt to new challenges in their environment. As a result, the roles and responsibilities within these teams will have a certain level of flexibility and fluidity based on the experience, skills and knowledge of the people making up the team in question, as well as on the needs of the specific project.

C) COMMUNICATION

It is necessary to put in place a communication plan that takes into account the need for transparency and responsibilities of the agents involved. The key service users and municipal processes must be identified, as well as the players involved in the technical issues. All these will be the recipients of the communication actions that must be undertaken as frequently as necessary based on the targets and needs of each project. Face-to-face conversations will take priority over other channels of communication.

D) TRANSPARENCY

The project status, priorities, risks, problems and potential "impediments" should be openly and transparently shared with the key players in order to tackle the challenges swiftly. This can be done via direct communication but also by using tools adapted to the agile methodology, such as Kanban boards.

E) ONGOING IMPROVEMENT

Collaborative working practices will be established in order to foster a culture of learning and continuous improvement, both in the service provided and in the team members.

F) RAPID FEEDBACK

The team will endeavour to build the shortest possible iterations. Team efforts will be undertaken in such a way as to facilitate rapid feedback and incorporate that feedback into the project backlog.

G) ACCEPTANCE OF CHANGE

It is understood that plans and projects will evolve and change during the time a service is provided. Analysis and planning practices will be established to reflect that fact.

H) ITERATIVE THINKING

The activities of analysis, design, planning and development will adopt an iterative approach in short cycles so as to enable feedback and priorities to be changed according to needs.

I) RAPID DEPLOYMENT, FREQUENT DEPLOYMENT

Emphasis should be placed on providing value to the end users as quickly as possible, reducing the time needed to produce applications and receiving feedback from users as quickly as possible. Priority will be placed on the identified minimum viable product capable of meeting the priority needs of the users, and subsequent deliveries will be undertaken according to that approach.

J) INTEREST IN PROVIDING VALUE TO USERS

Priority must be given to the needs of users. User needs stand above any other need, including those of the key players. Whenever a conflict of needs arises between groups, the existence of clear priorities must be guaranteed. Diversity within teams will be encouraged in order to foster inclusive designs.

K) QUALITY

Quality not only refers to technical quality but also to quality of service. Quality is present in each step of the process, and every member of the team has a responsibility to ensure that the service is of the highest quality.

L) TRACKING PROGRESS

Delivery of executable software is the main measurement of progress. Other measurements should also be applied that reflect the value provided to users and the business in order to ensure that real needs are met and real value is provided.

These principles and guidelines will be applied to ICT projects and services via the practices and measures set out in the following section.

5 Implementation of the Agile Policy to ICT Projects and Services

In order to apply the guidelines described in the previous section, the IMI has developed an agile methodology based on scrum and adapted this to suit its role as ICT Manager for Barcelona City Council.

5.1 Basic and Structural Criteria

The IMI will undertake projects in line with the organisational structure of Barcelona City Council.

Although it is viable to combine agile development and projects with traditional methodologies (the "bimodal" approach), it is more difficult to manage development units with different life- cycles and with other external roles. For this reason, it is recommended that management of projects for units of the City Council (departments, sectors and districts) be undertaken in an agile or traditional manner but not in a manner that combines both models.

Furthermore, in a municipal unit where all developments are undertaken in an agile manner, it can be difficult to absorb peaks in demand without oversizing agile team capacity or overly extending response times. In this case, it may be advisable to contract traditional projects in parallel.

5.2 Scrum@IMI Methodology

The projects defined as agile will follow the agile development methodology for IMI applications, called Scrum@IMI. This is based on the scrum working framework and takes engineering practices from other models into account, such as DevOps.

This methodology will be supported by the use of an ALM platform (Application Lifecycle Management) with tools that include the following features, among others:

- Development planning (releases, sprints, work packages, defects, etc.).
- Documentation, code and binary repositories.
- Requirement and test management.
- Unit and function test automation.
- Ongoing integration and rollout.
- Code quality control.

Their use by a successful bidder will be compulsory, without leading to an increased licencing cost.

All the documentation generated internally during the course of development will be managed using the tools defined at the start of the project, preferably in wiki format.

The main characteristics foreseen for this methodology are described according to their life- cycle in the following sections.

The usual events (activities and milestones) for the Scrum@IMI methodology include the following steps:

Reunión/acontecimiento	Objetivos
Sprint	Development cycle (lasting from two to four weeks) in which a part of the system is delivered (including all sub-products, such as documentation and tests). The customer may choose to publish the increment delivered by the supplier or not.
Sprint planning	The product owner and the development team decide which part of the product will be produced during the sprint.
Daily Meeting	The development team monitor the sprint and the pertinent corrections. In the event of an overrun, the product owner and the scrum master are notified.
Sprint review	The scrum team and invited external players examine the result from the sprint and review the plan for the following sprints.
Sprint retrospective	The scrum team examines its own performance and plans specific improvements.

5.3 Roles and Responsibilities

Implementation of scrum roles at the IMI fully respects the standard objective and responsibilities for the roles while adapting them to the IMI context, which provides complex ICT management services for Barcelona City Council.

The main roles are as follows:

- **Product owner (City Council)**
- **Proxy product owner (IMI)**
- **Scrum master (IMI or supplier)**
- **Development team (supplier)**

PRODUCT OWNER (CITY COUNCIL)

The product owner (PO) is the role seeking to maximise the value delivered to users. The best examples of this role are those in which the PO is close to where the "business" decisions are taken, which is why this role is given to a certain key individual in the departments and management structure of the City Council. This position will provide that individual with transparent and frequent information on the status of all developments for the purpose of oversight and informed decision-making.

The City Council PO typically has a high workload and many partners with which to work. This means they may spend little time on usual PO activities, such as managing and detailing the product backlog or discussing issues with the development team. Hence, the POs will often have an assistant at the IMI, called proxy product owner. Under no circumstances should this detract transparency from the activities undertaken by the team or decision-making capabilities.

PROXY PRODUCT OWNER (IMI)

The proxy product owner (PPO) is not a standard scrum role but is common in situations where it is difficult to find a PO with both vision and decision-making capabilities from a business perspective or for reasons of proximity to the development team and availability for structuring and prioritising the backlog in a detailed manner. In the context of the City Council and the IMI, the IMI will often take on this role.

DEVELOPMENT TEAM (SUPPLIER)

Made up by a group of three members (preferably with multidisciplinary profiles), the development team carries out all the necessary activities related to project development. Its main objective is to achieve ongoing improvement. To do so, the team interacts with all the necessary IMI roles, of which the PO and PPO will be the main interlocutors regarding the work needing to be done, and the scrum master will act as methodological and organisational guide within the IMI.

SCRUM MASTER (IMI/SUPPLIER)

The scrum master (SM) seeks to teach scrum and use the working framework to improve value and team development effectiveness. Within an IMI context, this is achieved by acting as methodology guide; monitoring and controlling team performance or, if necessary, providing individual training to certain team members. In addition, the SM will help the team integrate into the IMI and City Council environment, especially with any cross-cutting departments.

5.4 Activities

Below is a description of the objectives and content of the main activities in the IMI agile methodology. If more detailed information is required on any of these activities, please consult the documentation provided via the IMI's Agile Space.

1. ACTIVITY: DELIVERY PLANNING

The task of activity planning is undertaken before building the product but can be re-planned during the course of this activity when deemed necessary. Its goal is to update and control the planning of sprints and deliveries in a way that is agreed by the whole scrum team, with leadership from the product owner and the proxy product owner. The result of this activity is the development plan, which is based on the proposal from the successful bidder in its offer and will need to meet the requirements detailed in the specifications.

2. ACTIVITY: BACKLOG REFINEMENT

Backlog refinement is carried out during project development and is led by the roles of Product Owner and Proxy Product Owner. Its purpose is to conduct a functional and technical analysis of the work packages in the product backlog.

A high level of interaction is expected among the users and other roles at the IMI during this activity. Building static models or dynamic prototypes that include the most important features of the system is recommended, so the user can validate them.

3. ACTIVITAD: SPRINT

The purpose of this activity is to build the system increment, represented by the sprint backlog, corresponding to the business and technical priorities agreed upon by the Product Owner and Proxy Product Owner with the development team. At the end of the sprint, the PO and PPO are responsible for formally validating and accepting the work packages making up the increment. The scrum manager will always assist in this task.

Failure by the successful bidder to deliver the increments produced at the end of the sprint will lead to the application of penalties by the IMI, as detailed in the 'Penalties' clause of the Administrative Specifications.

4. ACTIVITY: TRANSITION

The purpose of this activity is to deliver and implement the work packages determined by the Product Owner in line with the needs of users and other players at the IMI, such as the operations and user assistance service groups. These deliveries may be regular or on demand during the sprint (e.g. to resolve an urgent incident) or may be made at the end of the sprint.

Acceptance of the transition activity will require validation and approval by the Proxy Product Owner, the IMI. In the event of significant version changes, formal acceptance of the project will take place at a meeting of the project steering committee. Failing to achieve the objectives of this activity will halt the project.

5. ACTIVITY: OPERATION

The purpose of this activity (which will be of interest to the successful bidder if the scope of the contracted services includes user support) is to provide Level 2 and 3 support (ITIL terms) to the User Support Service group (USS) and the Operations Management.

The Proxy Product Owner will act as "service manager" when it must prioritise the incidents identified and raised by the USS and Operations groups, as well as monitor these situations. The User Support department describes the protocols and service level agreements (SLA) that determine the provision of this service.

5.5 Working Teams and their Management

The scrum model can be implemented with teams contracted in one of two ways: teams for the development of new projects; and application maintenance teams (AM). Below are the differences between them in terms of agile methodologies.

Multidisciplinary teams should include the skills for undertaking the following activities:

- Designing and building a digital service.
- Operating and maintaining a digital service.

The skills needed will change during the life-cycle of the service and support may be received from additional roles.

The entire team, but especially the designers, user researchers and developers, should work together to design, build, test and deliver the product.

1. TEAMS FOR NEW PROJECTS

Teams for new projects should be contracted once the scope of a service has been defined, before starting development. Definition of the contract requires a prior conceptualisation of the desired system, which will be jointly developed by the PO at the City Council and the PPO at the IMI. This conceptualisation should include the creation of the following elements:

- An initial backlog defining the scope of the development in terms or work items or packages.
- A delivery plan identifying the deliveries that will be made and which sprints form part of each one.

2. APPLICATION MAINTENANCE TEAMS (AM)

Application maintenance teams (AM) will be contracted a priori and will receive various types of request (corrective, evolutive and perfective), which will be added to the backlog. Urgent requests will be resolved as soon as possible, in line with the SLA targets. All other requests will be planned so they can be dealt with in sprints, probably shorter in duration (e.g. one week) because the packages will tend to be smaller and more independent.

3. COEXISTENCE OF PROJECT TEAMS AND AM TEAMS

The coexistence of Scrum teams for developing new products and Scrum teams maintaining those products will have various effects depending on the level of freedom in the contracts with the suppliers and the most suitable option according to the Proxy Product Owner and the Scrum Master. The options are as follows (from most to least suitable):

- The product team includes members from the AM team on a permanent basis in order to work together on the same applications.

- The AM team has an allocated time frame for working with the product team and preparing transferral of the application once delivered to the IMI.

- The AM team does not have an allocated time frame for work during development of the application but does have the frequent increments and the "facts" for preparing the transferral, and may also occasionally take part in the review of sprints.

5.6 Project execution

Agile projects managed by the IMI must be executed according to the following principles, as reflected in the corresponding contract with suppliers:

1) The agile recommendations and methodological guidelines from the IMI as PPO will be followed, as presented in the contract specifications and in the Agile Space of the IMI, as well as any other indications that may be applicable and notified by the Scrum Master, who nonetheless will respect team self-management as a general principle.

2) At the end of each sprint, an agreed deliverable or product increment (software increment) must be available, which will meet the minimum quality requirements defined in the IMI Agile Space. If it is not possible to deliver all the content that is planned, quality will take priority over quantity.

3) Transparency will be required with the Scrum Master (and potentially the PPO) with regard to any possible internal team problems, as well as any external impediments, in order to seek together the best solutions to any structural or individual development problems.

4) Suppliers will use cooperative tools, as well as the baseline methodological rules that favour team interoperability, such as the language, structure and level of detail in the software documentation and other supporting documents.

5) The team responsible for Levels 2 and 3 of incident management will need to have minimum knowledge of the systems and work with the IMI as closely as possible. In the future and depending on the ability to automate rollout of the platform used, it is foreseen that teams will be able to roll out their systems autonomously, in the pre-production and production environments.

Annex 1 – Further Information. References

Barcelona City Council Documentation

Barcelona City Council: "Guidelines for the provision of agile digital services", 2017.

References:

Agile Manifesto: *http://agilemanifesto.org/iso/en/manifesto.html*
Agile Principles: *http://agilemanifesto.org/iso/en/principles.html*

Scrum

• **"The Scrum Guide", Schwaber and Sutherland**

This 17-page document is the official definition of scrum. It is highly recommend reading for an understanding of the philosophy behind this working framework, as well as its component parts and rules.

• **Scrum Pocket Guide, Verheyen**

This short book of approximately 100 pages is a commented extension of "The Scrum Guide". It is recommended reading if you would like to have good theoretical knowledge of the working framework.

• ***Large-Scale Scrum*, Larman & Bodde**

This book describes the large-scale scrum model (LeSS) for applying scrum to numerous teams. It therefore includes advanced content. Nonetheless, it also reflexively reviews the basic concepts of scrum in detail, and also introduces many other more advanced concepts.

Defining products and backlogs

• ***User Stories Applied*, Cohn**

This book is a practical guide to understanding what user stories are and how to make good use of them.

• ***User Story Mapping*, Patton**

This book goes in to further detail about the usefulness of story mapping to better understand user problems and learn how to design rapid development plans that harness frequent deliveries for optimising the value offered to users.

Agile planning

• ***Agile Estimating and Planning*, Cohn**

This book is a practical guide to the process of estimating effort in product backlogs and how to use those estimates to plan versions and sprints and oversee plans.

- ***Scrum & XP from the trenches*, Kniberg**

This book is free to download from the InfoQ website. It is a complete and pragmatic guide on how to develop using Programming scrum and extreme programming.

- ***Agile Project Management with Scrum*, Schwaber**

This book contains practical experiences from the co-creator of scrum in its application at various companies, and explains what went well and what went badly.

Product Owner

- ***AGILE Product Management with Scrum***

This book offers a good introduction to the role of product owner, its responsibilities and how to undertake the standard activities within the life-cycle of a product.

- ***Scrum Product Ownership, Galen***

This book explains how to perform the role of product owner, both at small and large organisations, and especially includes those aspects of organisation involving all the players.

Scrum Master

- ***Scrum Mastery*, Watts**

This book is a quick and easy guide to correctly approaching the role of scrum master. It is a highly recommendable introduction.

- ***Coaching Agile Teams*, Adkins**

This is an advanced reference book for any professional coach or scrum master. It is highly detailed and contains advanced concepts.

- ***Succeeding with AGILE*, Cohn**

This book explains models for spreading agile methods within organisations. There is no unified methodology for introducing agile methods at a company; but this advice may be useful for application to the specific context of each organisation.

Other documentation consulted

Government of the United Kingdom. "*Agile Delivery*", *https://www.gov.uk/service-manual/agile-delivery*

Nuottila J., Aaltonen K., Kujala J. "*Challenges of adopting agile methods in a public Organization*", International Journal of Information Systems and Project Management, vol. 4, No. 3 (2016), pp. 65-85, DOI: 10.12821/ijispm040304.

Torrecilla-Salinas C. J., Sedeño J., Escalona M. J., Mejías M. "*Agile in Public Administration: Oxymoron or reality? An experience report*", CAiSE Industry Track, vol. 1017 of CEUR Workshop Proceedings, pp.1-8, CEUR-WS.org (2013), online at *http://ceur-ws.org/Vol-1017/Paper1CAiSE_IT2013.pdf*

Software Engineering Institute of the CMU. "*Contracting for Agile Software Development in the Department of Defense: An Introduction*" (August 2015), online at *resources.sei.cmu.edu/asset_files/TechnicalNote/2015_004_001_442515.pdf*

US Government Accountability Office. "*Software Development: Effective Practices and Federal Challenges in Applying Agile Methods*", GAO-12-681, online at *http://www.gao.gov/assets/600/593091.pdf*

Balter B. "*Towards a More Agile Government*", 41 Pub. Cont. L. J. 149, The Public Contract Law Journal (autumn 2011), online at *https://ben.balter.com/2011/11/29/towards-a-more-agile-government/*