Chapter 5
Integration of Human-Centred Design and Agile Software Development Practices: Experience Report from a SME

Carmelo Ardito, Maria Teresa Baldassarre, Danilo Caivano, and Rosa Lanzilotti

Abstract The integration of Human-Centred Design (HCD) and Agile software development approaches is gaining momentum in both the Human-Computer Interaction and Software Engineering communities. The common principles shared by the two approaches, i.e., iterative design, user involvement, continuous testing and prototyping, should facilitate their integration which, however, is not without problems. In this chapter we report a study conducted in a Small-Medium sized Enterprise (SME) that adopts a Scrum-based methodology. After identifying the integration points between HCD and Agile development activities, a tailored HCD-Scrum methodology has been applied to the development of a web application aimed at retrieving and comparing data related to public institutions.

Keywords Tailored HCD-Agile methodology • Case study

5.1 Introduction

Many research papers have suggested methodologies and techniques to be used in software development processes, in order to maximise the usability and user experience (UX) of the final product. Unfortunately, they are very seldom used in industrial settings, as our recent studies have also pointed out [1]. However, Ardito et al. [2] provide evidence that Human-Centered Design (HCD) techniques can successfully be integrated in the software engineering practices of a company adopting a waterfall model, provided that a depth analysis of company practices is performed by HCD researchers along with the practitioners, working together during the entire software development process from inside the company.

C. Ardito • M.T. Baldassarre • D. Caivano • R. Lanzilotti (✉)
Dipartimento di Informatica, Università degli Studi di Bari Aldo Moro,
Via Orabona 4, 70125 Bari, Italy
e-mail: carmelo.ardito@uniba.it; mariateresa.baldassarre@uniba.it;
danilo.caivano@uniba.it; rosa.lanzilotti@uniba.it

G. Cockton et al. (eds.), *Integrating User-Centred Design in Agile Development*,
Human–Computer Interaction Series, DOI 10.1007/978-3-319-32165-3_5

HCD was proposed in the 1980s and since then, Human-Computer Interaction (HCI) researchers have worked in order to define methods to design and evaluate usable interactive systems. Nevertheless, even recent literature provides many examples revealing that this research has had little impact on software development practices for different reasons, i.e. lack of knowledge about what usability is, practitioners' feeling that too many resources (e.g. time and costs) are necessary, the difficulty in recruiting trained usability/HCD experts [3–6].

In recent years, our research has focused on deeply analysing the reasons that prevent a fruitful integration of HCD principles in the practice of software development [1, 2]. Specifically, in this chapter we focus on Agile development and report a study conducted in a Small-Medium sized Enterprise (SME) that adopts a Scrum-based methodology. The first objective was to identify, in collaboration with experienced project managers of the company, the integration points between Scrum and HCD activities. Thus, a tailored HCD-Scrum methodology was defined. It was applied to a company project concerning the development of a web application for retrieving data related to public institutions and comparing them based on several economic indexes (e.g., number of employees, incomes, outcomes, taxes issued, etc.).

The main novelty of the defined methodology is related to the need of an initial step, longer than the following development sprints, for creating a software system skeleton (i.e., the basic architecture for enabling the core functionalities and the wireframe of the main user interfaces) according to the requirements identified during the initial meetings with the customer. Starting from the next sprints, such an evolutional prototype is discussed and evaluated with the customer, as is usually done in Scrum methodologies [7, 8].

Another significant result showed that both face-to-face and remote informal Verification & Validation sessions performed during a sprint proved to be very valuable for improving usability and UX. The customer is also involved and actively participates in the co-design of the prototype. This methodology has been generalised and further experimented in other projects and is currently used by the software company.

The rest of the chapter is organised as follows. Section 5.2 discusses related work on the topic of HCD in Agile software development. In Sect. 5.3 the Scrum methodology tailored to and adopted by an Italian SME is presented. Section 5.4 describes a case study that shows how the Agile technique has been adopted and how user experience has been involved during the development process. A final discussion concludes the chapter.

5.2 Related Work

Interest in the integration of HCD and Agile development approaches is growing, as demonstrated by the number of papers published in the last decade [9]. Although HCD and Agile are two different software development approaches, they share

common principles, i.e., iterative design, user involvement, continuous testing and prototyping [10], that are analysed from different viewpoints by the two communities. The integration of these two approaches will result in complementing each other, in order to allow us to gain the advantages of both worlds and at the same time to minimise the deficiencies of both approaches: HCD can improve Agile development by providing a systematic way to analyse end-user needs, whereas Agile can improve HCD by providing more frequent iterations, which leads to more frequent usability evaluations [11].

However, there are studies indicating that the combination HCD-Agile presents two important problems. The first one is related to the communication between developers and designers [12]. HCD practitioners concentrate on issues such as ease of use, ease of learning, user performance, user satisfaction, aesthetics; Agile practitioners, on the other hand, mainly focus on implementing functional requirements into a running system [13]. The other important problem regards the distinction of the role of the two different actors, i.e. the customer and the user, participating in HCD-Agile development approaches [14]. Differently from the HCD, in the Agile development approach customer and user play the same role, not distinguishing between customers, who have required the system but could not use it, and users, who will actually use the system.

Several studies examine various aspects of the integration of HCD and Agile approaches and suggest ways for this integration. In many cases, researchers report about their experience and provide recommendations suggesting how HCD can effectively be integrated in Agile approaches. An interesting systematic review carried out by Silva da Silva et al. identified six main aspects concerning the integration of such two development approaches [15]: (1) Little Design Up Front (LDUF), (2) Prototyping, (3) User stories, (4) Inspection evaluation, (5) User testing, (6) One sprint ahead (see Table 5.1).

Concerning the first aspect, i.e. LDUF, many researchers agree that user research activities should be performed before the project kickoff meeting is held [16, 17] or in a Sprint 0 through a contextual inquiry and/or user interviews [12, 18, 19]. One of the HCD techniques suggested is Extreme Personas (as called by the authors in [20]), an extension of XP's user stories.

Table 5.1 The six main aspects concerning the integration of HCD and Agile approaches

Aspect	Strengths
Little Design Up Front	User research activities performed before the kickoff meeting or in Sprint 0
Prototyping	Prototypes throughout the whole project
User stories	User requirements are created in coordination with all the stakeholders
Inspection evaluation	Paper prototypes are evaluated for refining the user interface
User testing	Interactive prototypes are evaluated involving end users
One sprint ahead	HCD specialists work one sprint ahead or in Sprint 0

The importance of prototyping is recognised for reaching an effective integration of the two approaches. Researchers suggest the creation of prototypes throughout the whole project [2, 16, 18] also because they are a good communication tool between developers and HCD specialists [2]. Prototypes can be generated from personas and user stories previously defined and evaluated through inspection evaluations and usability testing [11].

The third aspect concerns the definition of user stories. User stories illustrate the user-required application features that are created by the customer in coordination with all the stakeholders. As previously mentioned, they are instrumental for creating system prototypes [11, 18]. Researchers recommend defining user stories that address the usability issues and acceptance testing criteria [21].

Evaluation takes into account both user testing and inspection. The common recommendation is to slim down the evaluation procedure. Such evaluations should be performed at different stages of the system development lifecycle. Specifically, some researchers suggest carrying out evaluation on paper prototypes, with the goal of refining the user interface for the next iteration [11, 17, 18, 22]. Others recommend performing user testing only on interactive prototypes [23], while others suggest integrating user testing into the acceptance tests to validate the user interface, e.g. [24].

Inspection is a cost-effective technique recommended to be performed on paper prototypes that should be carried out until the prototype is stable to serve as a basis for the user interface implementation [25]. Heuristic Evaluation is the most used inspection technique.

Finally, concerning One Sprint Ahead, it is suggested that HCD specialists work one sprint ahead of the development team and recommend that this practice should start in Sprint 0 or even two or three iterations ahead of the rest of the team [12, 26, 27].

In this paper we report our experience in integrating HCD activities in an Agile software development approach. In our work, we have taken into account the recommendations reported in literature.

5.3 Integrating HCD Activities in the Scrum Process of the SME

Software development trends are moving more and more towards Agile methods pushing, at the same time, for a constant integration of customer feedback and HCD within the Agile approach practices themselves. After describing in Sect. 5.3.1 a typical Scrum-Like process, in Sect. 5.3.2 we illustrate how the Action Research method was applied in order to identify the integration points between HCD and Agile development activities and, finally in Sect. 5.3.3 how the Scrum-Like process has been tailored to the needs of an Italian SME called SER&Practices (from here on SER&P) and how customer feedback and HCD have been integrated into their

Agile development approach. The concept underlying the process is its adherence to Scrum-based practices, which have been adapted to conform to the needs of the SME. Key points are: constant involvement of the customer; rapid development of code through iterative stepwise refinements; high frequency of releases due to the continuous iterations.

5.3.1 A Typical Scrum-Like Process

A typical Scrum-Like process is depicted in Fig. 5.1. It is made of three major steps – *Inception*, *Development*, *Deliver* – even if they are often not clearly identified and are referred to in several and diverse manners. There are clear key roles involved – *Product Owner*, *Scrum Master*, *Team*, *Customer* – there are also a set of time-box events – *Sprint Planning* meeting, *Daily Standup* meeting, *Sprint Review*, *Sprint*

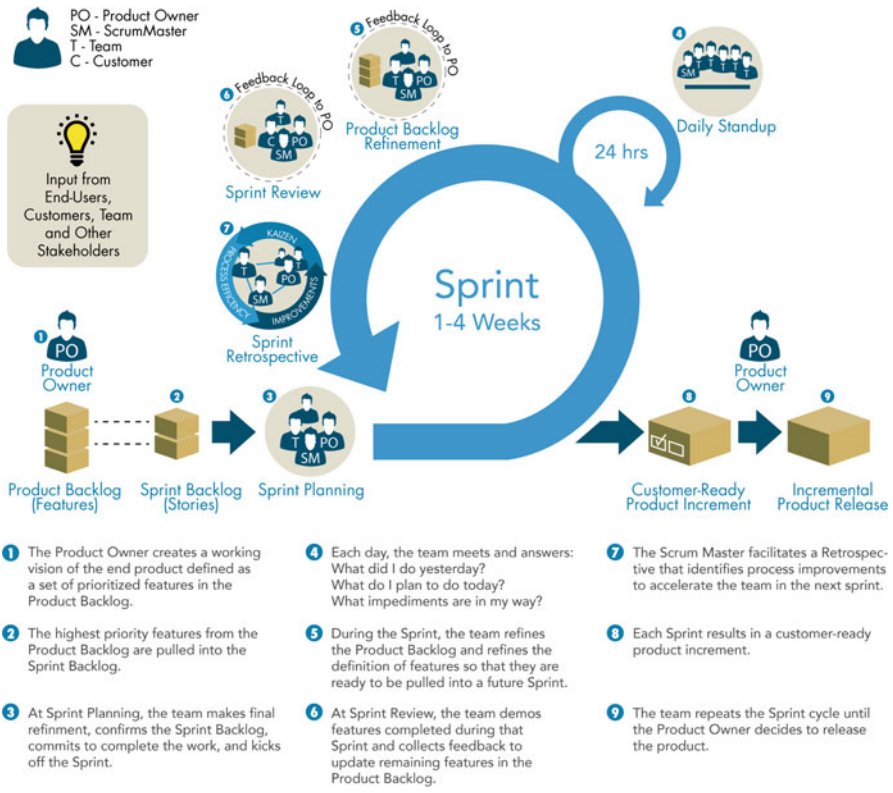

Fig. 5.1 A Scrum process representation rearranged from http://www.scruminc.com

Retrospective – and a couple of fundamental artifacts – *Product Backlog* and *Sprint Backlog*.

A key aspect of Scrum is the use of self-organised, cross-functional and empowered teams who organise their work into short development cycles, called Sprints.

A cycle begins with the input provided by stakeholders (End User, Customer, Team etc.). The Product Owner then develops a Prioritised Product Backlog, which contains a prioritised list of business and project requirements written in the form of user stories. Each Sprint begins with a Sprint Planning Meeting during which high priority user stories are considered for inclusion in the Sprint. A Sprint generally spans between 1 and 4 weeks and involves the Scrum Team working to create potentially shippable Customer-Ready Product Increments. During the Sprint, short, highly focused Daily Standup Meetings are conducted where team members discuss daily progresses. Toward the end of the Sprint, a Sprint Review Meeting is held during which the Product Owner and relevant stakeholders are provided a demonstration of the Product Increments. The Product Owner accepts the Product Increments only if they meet the predefined Acceptance Criteria. The Sprint cycle ends with a Retrospect Sprint Meeting where the team discusses ways to improve processes and performances as they move forward into the subsequent Sprint.

5.3.2 The Action Research Method Applied into a SME Process

Action research [28] is a social science methodology designed to help communities and organisations improve the way they address issues and solve problems and at the same time develop scientific knowledge about the problem and its solution. An action research process requires the active participation of individuals, i.e. researchers and practitioners, working together in a team to change the situation in an organisation, whilst conducting research. This collaboration results in a better understanding of the factors affecting the existing situation and the most suitable remedies. In past work with software companies aiming at integrating HCD in a company's software development practice [28], an action research approach was able to persuade practitioners and managers to incorporate UX activities into their software development life cycle.

In the current experience with SER&P, we performed action research in collaboration with experienced project managers of the company, in order to identify the integration points between HCD and Scrum activities. Thus, an HCI researcher was introduced to the software development team, in order to observe the team performing a Scrum-based approach, understand existing practices and identify aspects that were problematic from the involved practitioners' point of view. Data about current practices included notes of the practices reported in the researcher's diary,

observations related to artifacts, and comments pertaining informal discussions with the other team members. Project managers and the HCI researcher met every 2 weeks. During these meetings, the researcher discussed with the project managers about significant episodes observed in the company and suggested possible HCI activities that could be integrated in the process.

At the end of the Action Research study, a methodology for integrating HCD and Scrum activities was defined, as shown in the next sub-section.

5.3.3 Tailored HCD-Scrum Methodology

In the current experience with SER&P, a methodology was defined which introduces in a Scrum-like process some fundamental improvements from both an HCI and an SE point of view. Its main characteristics are the following:

1. **Customer Committee**. One of the key roles in the process is the Product Owner. The Product Owner represents all the stakeholders and is the voice of the Customer. He or she is accountable for ensuring that the team delivers value to the business. The Product Owner writes customer-centric items (typically user stories), ranks and prioritises them, and adds them to the product backlog. "Typically" a Scrum team should have one product owner. This role is equivalent to the customer representative role in some other Agile approaches. SER&P has introduced the concept of Customer Committee that includes a Product Owner selected within the personnel of the Scrum Team and at least two people from the customer side, desirably an end user and a business domain expert. The Customer Committee is actively involved in the Product and Sprint Backlogs definition and Sprint Review.
2. **Inception**. In Scrum-based processes, at the beginning, the development of a software product or service usually starts with stakeholder inputs. In SER&P the inception phase consists of interviews to stakeholders, as well as a context and field study carried out jointly with the customer. The study provides the essential context background and knowledge necessary to design the application being developed and enable the Customer Committee and Team (the latter, typically, 4 persons or less) to evaluate and prioritise the issues/features to develop. It consists of market study, cost, portfolio and competitor analyses. In this way SER&P merges the needs and modus operandi usually adopted by HCI communities – that typically and almost exclusively refer to end users, lead users and customers during requirement definition – and software engineers – who refer prevalently to field studies, competitor legacy systems in use, market studies etc., with limited contact with end users or customers. This fundamental step allows the Customer Committee to pin down priorities, backlogs, user stories and features that customers consider most relevant. In each case, during the process itself, given its iterative characteristic of gradually adding features to the

final product, customers are continuously involved in designing and approving developed features and versions before they are released.

3. **Sprint n.0**. In a Scrum-like process, during this step a preliminary architecture of the application is conceived and main modules and sub-systems are identified. Nevertheless, the software development and the software source code usually take place in the Development phase during the Sprint execution. The SER&P process explicitly identifies a *Sprint n.0* in which a high level prototype is built. The basic architecture and infrastructure of the system are developed with the aim of having the foundations where the bricks produced in the next Sprints can be easily added. The goal is to obtain a highly modularised software system skeleton with basic software services such as data, communication, reporting, computation services etc., that enable effective software modules and functions development during the next Sprints. Another important objective is to reduce as much as possible the time needed for having a first version of working software. Only after having obtained this result, the Development phase can be executed effectively. Otherwise, the system development proceeds in a risky way due to the impossibility to have a constructive and effective feedback from customers.
4. **Scrum Islands**. SER&P typically uses teams of 4 persons or less. One of them is the Product Owner included in the Customer Committee that also covers the role of team member (involved in the software design and development) when not engaged in any committee activity. Another team member is usually a graphic designer with a basic knowledge of HCI techniques. The others are software engineers. When a project begins, the selected team members are transferred in a "Scrum Island" (see Fig. 5.2). A Scrum Island is a 4-seat working desk. This solution allows maximising the information flow, avoiding management gap and improving communication and collaboration between team members. For this reason in SER&P, the Daily Standup Meeting is not formally adopted. If a member is included in different islands or involved in more than one project, he/she physically moves between islands. Obviously, the company tries to minimise the sharing of persons across ongoing projects.
5. **1 Week Time Boxed Sprint.** SER&P adopts sprints with a fixed duration of 1 week. Every week the sprint outputs are reviewed and the next sprint backlog is defined by including the new tasks to be executed together with the still open and not closed ones. So the sprint duration is not the result of an ex-ante estimation but it is predefined. Estimations concerning the whole project are done by the Customer Committee at the end of every sprint, during the Product Backlog Refinement. This is possible thanks to the active involvement of the customer in the process. Thus, estimations about project termination and release deliver are carried out at the start of the project and updated weekly. A first draft of issues and their priority is made during the so-called *Inception* phase in order to have a general idea of the roadmap to follow and a first estimate of the effort required.
6. **(IN)Sprint Review.** The SER&P process includes a continuous Verification & Validation (V&V) activity. This activity temporally spans along the entire sprint. When a first/draft working software is available (after Sprint n.0 or few sprints

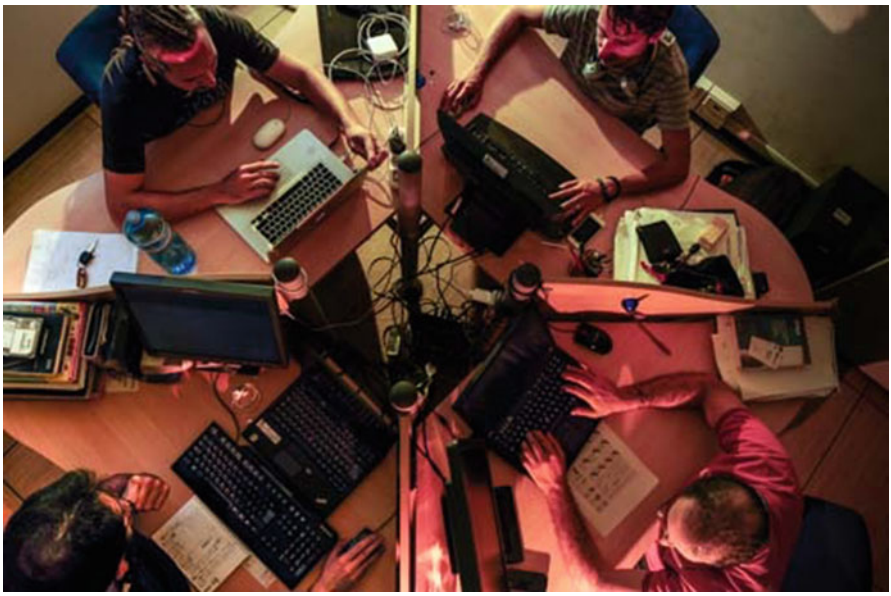

Fig. 5.2 A Scrum Island

later), the end users of the *Customer Committee* are asked to use the system and make a sort of user test or functional system test. This allows stressing and assessing the functionalities developed during a sprint and verifying their proper integration with what already exists. The results of the V&V activities are then discussed and analysed within the Sprint Review at the end of the Sprint. Here the Customer Committee is involved in the analysis of the results, from both functional and technical point of view, passing throughout the operating and business issues.

7. **Project Retrospective.** A Scrum process typically includes the *Sprint Retrospective* that is an official event in the Scrum methodology where all the parties (Product Owner, Scrum Master, Team) involved in the product development try to improve the process by sharing its strength and weaknesses. SER&P's process does not include it. Scrum means self-organised, cross-functional and empowered teams who organise their work autonomously. Thanks to the adoption of the Customer Committee, (IN)Sprint Review and Scrum Island an excellent information flow, a good communication level, a good coordination and prompt feedbacks from both team members and customer are assured. Thus, the Team can share, analyse and improve the process continuously without the need of a formal Sprint Retrospective event. Instead, what SER&P does is a *Project Retrospective*. Here all the available project data are analysed, strengths and weaknesses are highlighted and improvement opportunities and initiatives are defined and executed. In doing so SER&P refers to well known best practices and approaches in the Software Engineering community inspired to the Learning

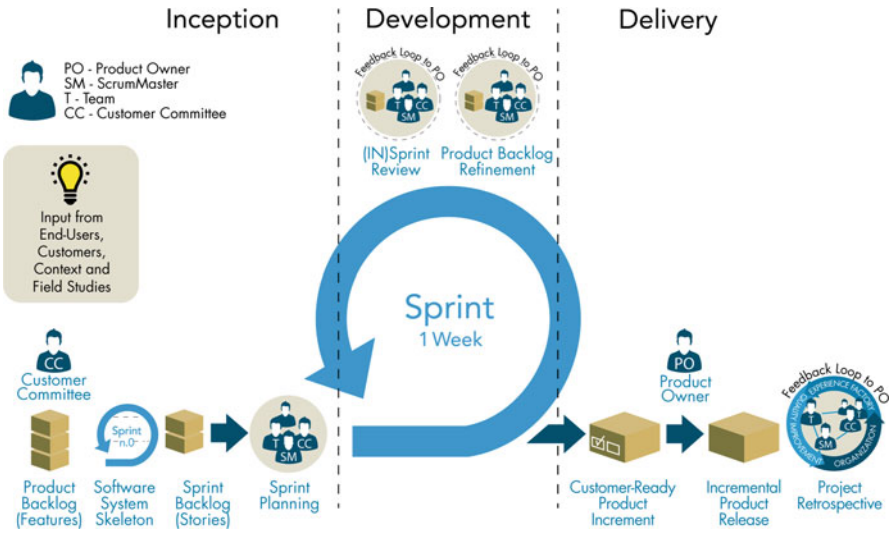

Fig. 5.3 Scrum process tailored by SER&P

Organisation and Experience Factory Models [29, 30]. This allows having a feedback loop within the entire SER&P organisation that enables and stimulates the company's growth. Another key point of this event is the active involvement of the customers in the retrospective analysis by means of a thematic focus group. During the focus group the customers are informally interviewed (to establish a friendly and relaxed atmosphere), in order to explore improvement opportunities and better address customer satisfaction and product quality improvements. The feedback collected is then used by SER&P to globally improve its strategies, management organisation and process.

An overview of the SER&P tailored HCD-Scrum process is showed in Fig. 5.3.

SER&P uses an *Application Lifecycle Management* (ALM) tool to manage its Scrum-based process called Redmine (www.redmine.org). In accordance to the tool characteristics, requirements are classified into the following possible categories: TO DO (new feature to develop), BUG (a corrective maintenance intervention to fix a bug related to a functionality that has already been developed), FEATURE (an evolutionary maintenance, i.e. a change in terms of modification or evolution of a functionality to add to the system). All issues are then prioritised into one of the following: low, normal, high, urgent, and immediate. The ALM therefore becomes the Product Owner's central channel towards the development team. The Product Owner uses the tool to formalise backlogs, classify issues and assign them to development team members, who in turn use their accounts to track the issues to develop during each sprint. It allows following progress of the project through an interactive Scrum board interface and tracking information such as tasks by category, statuses, workload of developers, issue tracker and so on.

Another tool used by SER&P during development to address software product quality is Kiuwan (www.kiuwan.com). It allows measuring, assessing and tracking software product quality and also improves it by means of a focused improvement plan. The Product Owner and Team use it during each Sprint Review where a quality report is produced for each sprint.

Data automatically collected through Redmine and Kiuwan are also used in the Project Retrospective for deciding on possible improvements.

5.4 Case Study

This section presents a case study that explores how the Scrum methodology, tailored to SER&P's production processes, has been used to carry out an industrial project called "PublicAccounts". The basic research question of this case study was: "How has SER&P integrated stakeholder input and feedback into their Agile software practices related to the PublicAccounts project?"

The project consists of creating a web portal able to integrate and elaborate several different data sources, in order to obtain precious information pertaining to the performance of public administrations and public bodies. The portal draws data in the form of open data from official sources on the web, such as for example www.soldipubblici.it, and from other databases provided by the Central Bank of Italy as well as the Ministry of Economy and Finance. It elaborates and classifies the data according to specific criteria and produces reports showing economical values of public institutions such as expenses, revenue, how public money is spent and invested, and so on. The application currently integrates data from 8000 Italian cities and will be extended to include all provinces, regions, and healthcare institutions. It provides a general overview of each single administration (Fig. 5.4) and also allows comparing public administrations, producing reports, elaborating statistics and rank virtuous cities compared to non-virtuous ones. The perspective users of the web portal are citizens, journalists, and public administration employees.

From a technological point of view, two critical issues for the project are the difficulty of contemporarily integrating and elaborating several heterogeneous data sources in terms of contents and structure on one hand, and on the other maximising the portal's usability assuring it is easy, immediate and intuitive to use.

At the time of writing the project has been running for 10 months and is expected to last a total of 18 months employing five staff each with at least 5 years of experience. Overall up to now a total of 1000 person days (including management effort) have been spent on the project.

SER&P adopted the Scrum-based process, illustrated in the previous section, to develop the portal. In accordance to the process, the work was carried out in incremental iterations. The work performed in the current Sprint and the next activities were planned.

The Customer Committee was defined. It included one person from SER&P and two from the customer side: an end user, i.e. a technician of the customer company

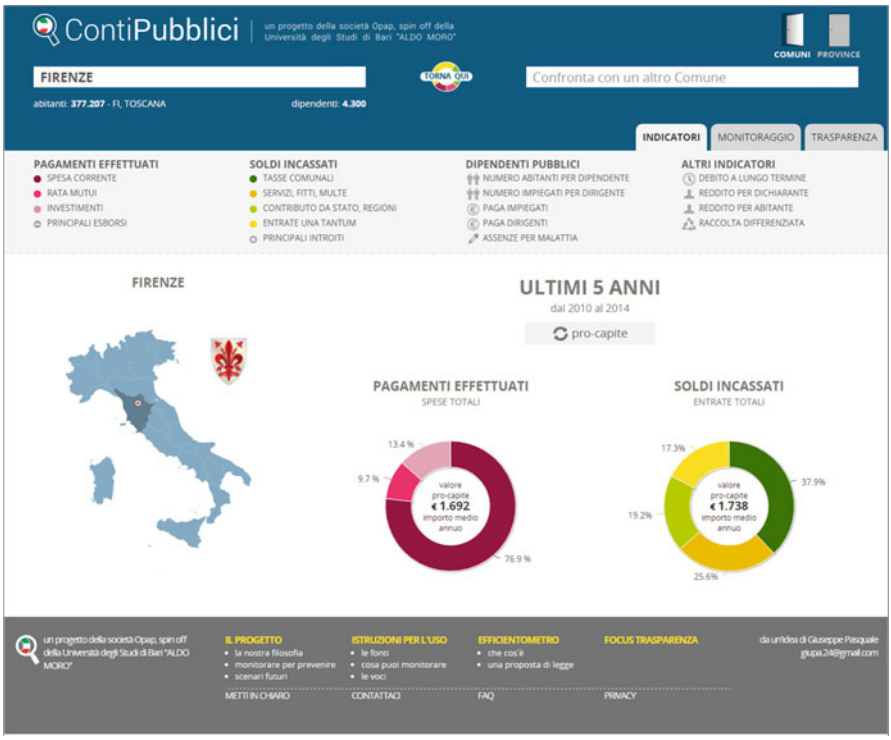

Fig. 5.4 Overview of revenues and expenses for the city of Firenze in the last 5 years

who uses the portal for his every day job, and a top manager with experience of the business role and business domain the portal refers to. Three development team members, two of which full time and one part time, were dedicated to the project. Daily Standup meetings were not conducted, given the characteristic of the Scrum Islands in SER&P, but instead weekly Sprint Reviews and Product Backlogs were scheduled.

Weekly meetings with the Customer Committee were scheduled. In particular, in the Sprint n.0 they were systematic and feedback was continuous, in order to define a general picture of the project, customer needs and the software system skeleton. Sprint n.0 was indeed the longest (40 days). The longer duration allowed to define a more detailed picture of requirements, architecture and front-end design, and to produce a first working prototype. In this sprint, more than in others, a greater portion of time and effort was arranged for planning and gathering customer data to produce upfront design and a starting-point prototype. Meetings were arranged in both SER&P sites and in the customer offices and took place mainly face to face. Communication channels such as video conferencing, email and teleconference were also used, but more during the later project sprints and especially during the (IN)Sprint Review in order to jointly analyse between the end user included in the

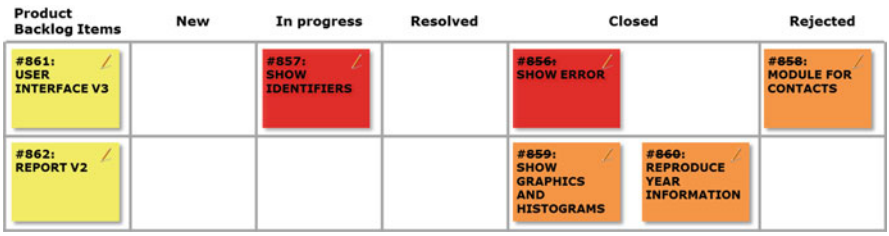

Fig. 5.5 Project sprint board

Customer Committee and Product Owner from SER&P, the defects, bugs or non conformities discovered by the end user during the continuous V&V activities.

Overall 17 sprints were carried out, the first lasting 40 working days (in about 3 months) and 1 week each for the other ones. The ALM tool, Redmine, was used to conduct and manage the project in term of sprints, tasks assigned to the development teams, prioritisation of user stories, features and tasks, customer feedback and suggestions. Figure 5.5 illustrates an example of a snapshot of the Scrum board used for showing tasks related to specific product backlog items and moving them from one column to another depending on their status. A backlog item can either be *new*, if it is a new feature assigned to the sprint; *in progress* if it is being developed during the current spring, *resolved* if it is a bug that has been fixed; *closed* once the task has been completed during the sprint or *rejected* when a task (either todo, bug or feature) suggested by the Customer Committee is not accepted by the Product Owner to be taken into consideration for the product being developed. The colors of the post-its are also meaningful in indicating the type of task category. In this case red means TODO (new feature to develop) and orange indicates a BUG (corrective maintenance).

So for example, a backlog item related to the user interface (#861) has some tasks, belonging to both TODO and BUGS categories, that are InProgress (#857: show identifiers), Closed (~~#856~~: show error) and Rejected (~~#858~~: module for contacts). The interface also specifies the name of the developer the task has been assigned to. For sake of space, only a small portion of the board has been reported.

Figure 5.6 provides a snapshot of monitoring and controlling activities. Moreover, thanks to the features of the process, the Product Owner is able to have an overview of the entire project. In particular tasks are grouped by status, category and management. This helps summarising the information of the project sprint board related to all of the Sprints, identifying for example which tasks are new, in progress, resolved, closed or rejected; which categories they belong to, i.e. how many are bugs, features or todo and what their delivery status is.

As the project went on, at the end of each (IN)Sprint Review features were verified by the Customer Committee and Product Owner. In particular, product quality was verified in terms of usability features and also from a perspective of internal product quality through the Kiuwan platform. Figure 5.7 shows an example of the indicators generated for the report produced following to a Sprint.

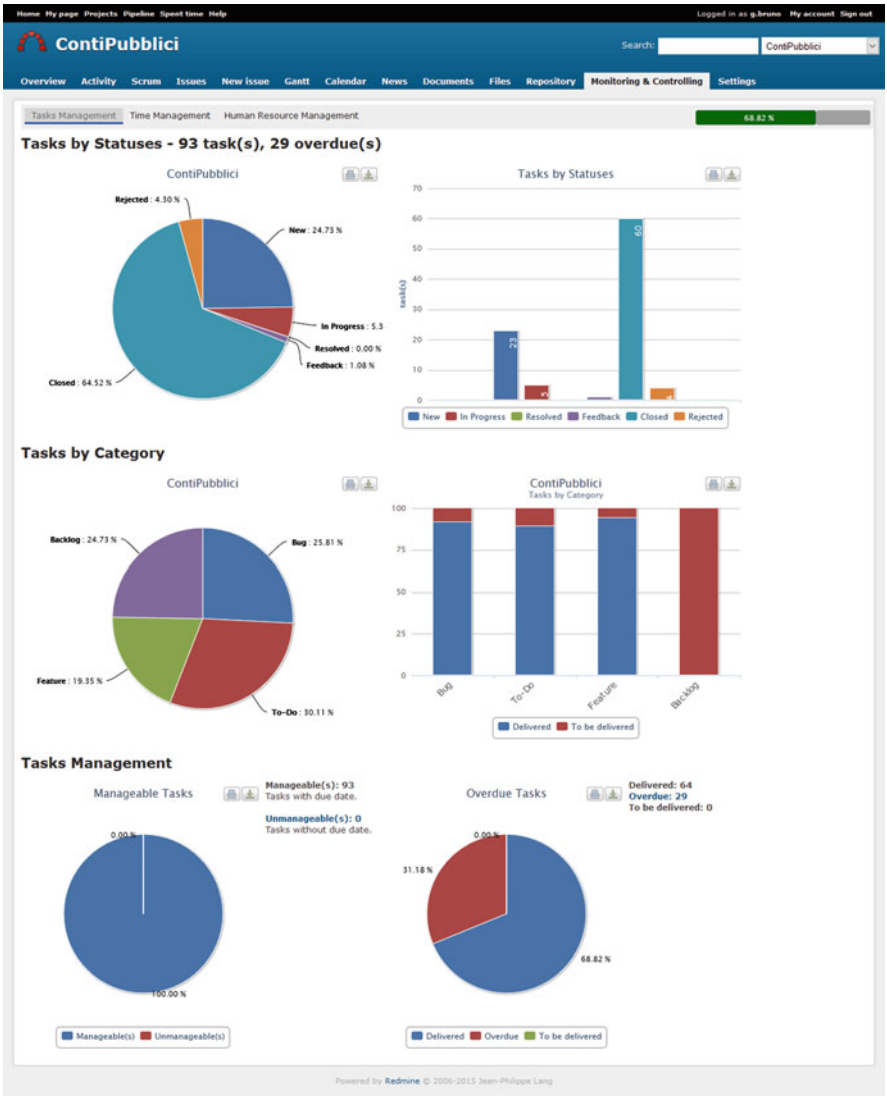

Fig. 5.6 Use of Redmine for managing the Scrum-based process

More specifically, the report summarises the general characteristics of the software being analysed providing structural information such as lines of code, number of files, level of complexity and amount of duplicated source code. In terms of internal quality, Kiuwan analyses source code with respect to quality characteristics such as: maintainability, reliability, portability, efficiency and security. It compares the current values of these quality characteristics to the baseline target threshold

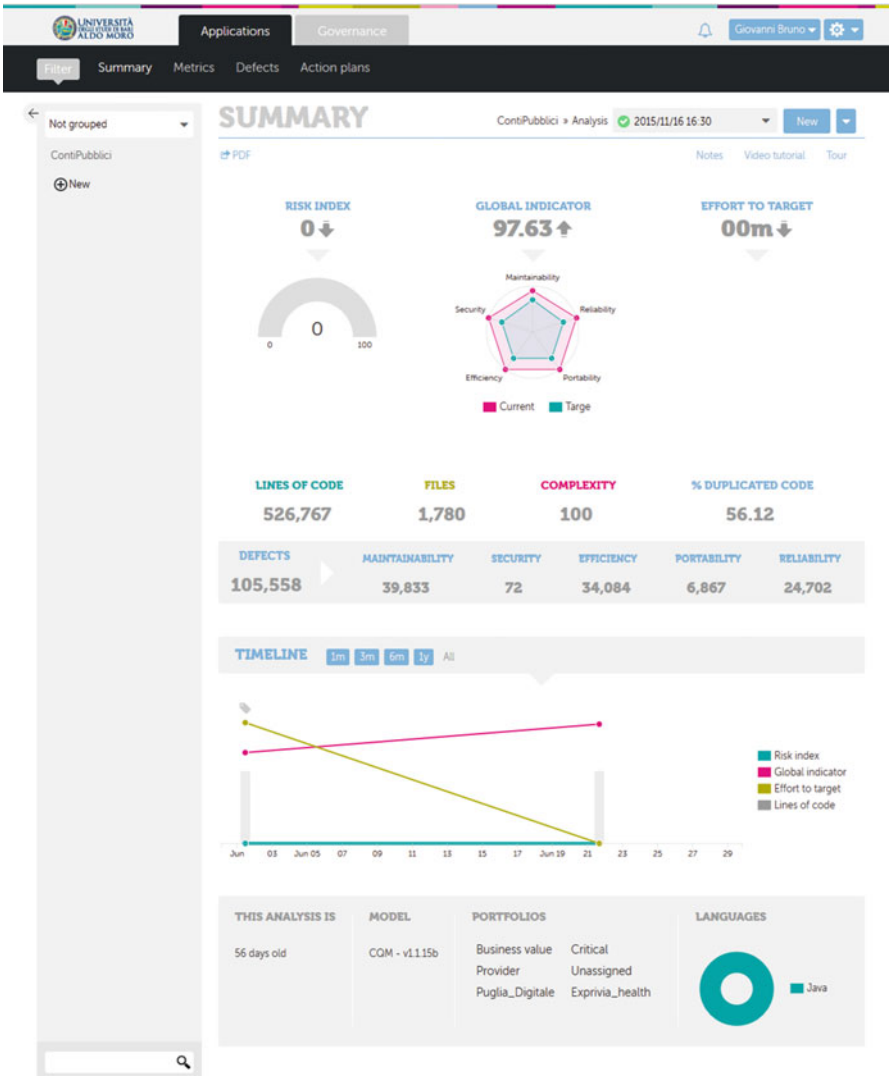

Fig. 5.7 Use of Kiuwan for product quality control

values, which can be set on a scale from 0 to 100 for the analysis. In this case, the target values were all set to 70 and the graph shows that the thresholds are fulfilled except for maintainability and portability. Improving these characteristics to target values will require an effort of 102 person hours, to be planned in the following sprints. Indeed, the report is used as discussion point for the (IN)Sprint Review. It is also used in the Project Retrospective phase of the Delivery step, before a version of the product is released to the customer. This activity enforces

interaction between Customer Committee, Product Owner, and Team in the light of a continuous improvement cycle towards the development and release of the final product.

5.5 Discussion and Conclusions

The successful development of the web application in the case study has shown the possibility of effectively integrating HCD techniques in the SME's Scrum practices. This has certainly required some tailoring of the Scrum practices. Changes have involved including staff who are totally dedicated to the project, the possibility of considering a flexible sprint length based on the workload and priority of features to be developed, use of an iterative-incremental progression of the Scrum development practices and most of all, customer interviews/field studies concentrated especially early on in the project.

The Project Owner with this iterative-incremental Scrum-based approach tries to combine her/his ideas of the final product with customers' input, and prioritises and sorts out feedback to make issues feasible for development. Customer feedback is crucial for the entire process. It has to be systematically gathered and appropriately filtered during each step of the process starting from end user and customer input collected during the Inception step; verified and validated during the (IN)Sprint review of the Development step; and feedback has to be provided by the Customer Committee in the Project Retrospective of the Delivery step before releasing the final product. On the other hand, it often happens that in the early stages of the process communication with the customer is difficult due to the different background of people involved. However, this difficulty is quickly and naturally overcome after the short period needed for participants to become accustomed to each other's habits and practices. We have also found that use of an Application Lifecycle Management (ALM) tool is crucial as it provides an important infrastructure for the entire team.

Strategically speaking, positive lessons learnt from integrating HCD into Scrum-based practices that have made SER&P successful in project execution include:

- carrying out an initial sprint, longer than the other development sprints, for creating a software system skeleton according to the requirements identified during the initial meetings with the customer;
- considering customer feedback early in the project sprints (especially in the first one) for upfront design and as a guideline for developing a prototype used as a communication means in the rest of the process in addition to face to face communication;
- working iteratively and incrementally during each sprint;
- planning for testing during the sprint and iterating in the next sprint based on the results, allowing customers to actively interact in the acceptance test of individual issues as well as of the entire release;

– using a project management tool, i.e. an ALM tool, which includes Scrum-based practices, to support the entire process and simplify organisation for Product Owners and development team members.

The experience described in this chapter has shown that customer feedback is valuable for HCD and in turn Agile practices seem well suited to be mapped to customer involvement. It is important that feedback and customer involvement are set at multiple points throughout the development process. To this end, the Scrum-based process adopted by SER&P indeed includes several points in all steps (Inception, Development, Delivery) that take into account such aspects. We have found that it is important that interviews/meetings/field studies are carried out in the first part of the project, during the first sprint, in order to have a clear picture of what is going to be developed. This is because the first sprint is most likely to be longer than the subsequent ones, which focus more on development, evaluation, implementation and integration aspects.

Customer feedback should be filtered, i.e., not everything a customer wants can be carried out or is actually a good idea. In this sense, SER&P practices involve decision points where suggestions can either lead to a new issue/change request or be rejected. As implementation issues can be affected by customer feedback it is important to closely evaluate the requests and discuss them, in order to reach trade-offs during project sprints.

All the status and evolution points should be tracked as the project proceeds. For this reason, decision making and project status monitoring on behalf of the Project Owner with respect to the development team members is simplified by using an ALM tool to support organisation of sprints, daily meetings, issue assignments and so on.

The Scrum process described in this chapter has been generalised and further experimented on in other projects and is currently used by SER&P. We are confident that the community is slowly moving in the right direction as user needs are becoming systematically ingrained into the production processes, especially Agile ones.

Acknowledgments This work is partially supported by the Italian Ministry of University and Research (MIUR) under grants PON02_00563_3470993 “VINCENTE”, PON04a2_B “EDOC@WORK3.0”, and PON03PE_00136_1 “DSE” and by the Italian Ministry of Economic Development (MISE) under grant PON Industria 2015 MI01_00294 “LOGIN”. We are also immensely grateful to Prof. Maria Francesca Costabile and Prof. Giuseppe Visaggio for their valuable and constant support.

References

1. Ardito C, Buono P, Caivano D, Costabile MF, Lanzilotti R, Bruun A, Stage J (2011) Usability evaluation: a survey of software development organisations. In: International conference on software engineering and knowledge engineering (SEKE’11). Knowledge Systems Institute, Skokie, pp 282–287

2. Ardito C, Buono P, Caivano D, Costabile MF, Lanzilotti R (2014) Investigating and promoting UX practice in industry: an experimental study. Int J Hum Comput Stud 72(6):542–551
3. Rosenbaum S, Rohn JA, Humburg J (2000) A toolkit for strategic usability: results from workshops, panels, and surveys. In: SIGCHI conference on human factors in computing systems (CHI'00). ACM, New York, pp 337–344
4. Boivie I, Aaborg C, Persson J, Lofberg M (2003) Why usability gets lost or usability in In-house software development. Interact Comput 15(4):623–639
5. Cajander A, Gulliksen J, Boivie I (2006) Management perspectives on usability in a public authority: a case study. In: Nordic conference on human-computer interaction (NordiCHI'06). ACM, New York, pp 38–47
6. Seffah A, Donyaee M, Kline RB, Padda HK (2006) Usability measurement and metrics: a consolidated model. Softw Qual Control 14(2):159–178
7. Schwaber K, Beedle M (2001) Agile software development with Scrum. Prentice Hall PTR, Upper Saddle River
8. Dybå T, Dingsøyr T (2008) Empirical studies of agile software development: a systematic review. Inf Softw Technol 50(9–10):833–859
9. Jurca G, Hellmann TD, Maurer F (2014) Integrating Agile and user-centered design: a systematic mapping and review of evaluation and validation studies of Agile-UX. In: Agile conference (AGILE'14). IEEE Computer Society, pp 24–32
10. Blomkvist S (2005) Towards a model for bridging agile development and user-centered design. In: Seffah A, Gulliksen J, Desmarais M (eds) Human-centered software engineering – integrating usability in the software development lifecycle, vol 8. Springer, Dordrecht, pp 219–244
11. Hussain Z, Milchrahm H, Shahzad S, Slany W, Tscheligi M, Wolkerstorfer P (2009) Integration of extreme programming and user-centered design: lessons learned. In: Abrahamsson P, Marchesi M, Maurer F (eds) Agile processes in software engineering and extreme programming, vol 31. Springer, Berlin/Heidelberg, pp 174–179
12. Chamberlain S, Sharp H, Maiden N (2006) Towards a framework for integrating agile development and user-centred design. In: Abrahamsson P, Marchesi M, Succi G (eds) Extreme programming and agile processes in software engineering – XP'06, vol LNCS 4044. Springer, Berlin/Heidelberg, pp 143–153
13. Memmel T, Gundelsweiler F, Reiterer H (2007) Agile human-centered software engineering. In: British HCI group annual conference on people and computers: HCI . . . but not as we know it, vol 1 (BCS-HCI'07). British Computer Society, Swinton, pp 167–175
14. Highsmith J (2002) Agile software development ecosystems. Addison-Wesley Longman Publishing Co., Inc., Boston
15. Silva da Silva T, Martin A, Maurer F, Silveira M (2011) User-centered design and agile methods: a systematic review. In: Agile conference (AGILE'11). IEEE Computer Society, pp 77–86
16. Detweiler M (2007) Managing UCD within agile projects. Interactions 14(3):40–42
17. Williams H, Ferguson A (2007) The UCD perspective: before and after agile. In: Agile conference (AGILE'07). pp 285–290
18. Fox D, Sillito J, Maurer F (2008) Agile methods and user-centered design: how these two methodologies are being successfully integrated in industry. In: Agile conference (AGILE'08). IEEE Computer Society, pp 63–72
19. Kollmann J, Sharp H, Blandford A (2009) The importance of identity and vision to user experience designers on agile projects. In: Agile conference (AGILE'09). IEEE Computer Society, pp 11–18
20. Wolkerstorfer P, Tscheligi M, Sefelin R, Milchrahm H, Hussain Z, Lechner M, Shahzad S (2008) Probing an agile usability process. In: Extended abstracts of SIGCHI conference on human factors human factors in computing systems (CHI EA'08). ACM, New York, pp 2151–2158

21. Beyer H, Holtzblatt K, Baker L (2004) An agile customer-centered method: rapid contextual design. In: Zannier C, Erdogmus H, Lindstrom L (eds) Extreme programming and agile methods – XP/Agile Universe 2004, vol LNCS 3134. Springer, Berlin/Heidelberg, pp 50–59
22. Larusdottir MK (2012) User centred evaluation in experimental and practical settings. Media Technology and Interaction Design, KTH Royal Institute of Technology, Stockholm
23. Federoff M, Villamor C, Miller L, Patton J, Rosenstein A, Baxter K, Kelkar K (2008) Extreme usability: adapting research approaches for agile development. In: Extended abstracts of SIGCHI conference on human factors human factors in computing systems (CHI EA'08). ACM, New York, pp 2269–2272
24. Benigni G, Gervasi O, Passeri F, Kim, TH (2010) USABAGILE_Web: a web agile usability approach for web site design. In: Taniar D, Gervasi O, Murgante B, Pardede E, Apduhan B (eds) computational science and its applications – ICCSA 2010, vol LNCS 6017. Springer, Berlin/Heidelberg, pp 422–431
25. Hellmann TD, Hosseini-Khayat A, Maurer F (2010) Supporting test-driven development of graphical user interfaces using agile interaction design. In: International conference on software testing, verification, and validation workshops (ICSTW'10). IEEE Computer Society, pp 444–447
26. Sy D, Miller L (2008) Optimizing agile user-centred design. In: Extended abstracts of SIGCHI conference on human factors human factors in computing systems (CHI EA'08). ACM, New York, pp 3897–3900
27. Illmensee T, Muff A (2009) 5 Users every friday: a case study in applied research. In: Agile conference (AGILE'09). IEEE Computer Society, pp 404–409
28. Checkland P, Holwell S (1998) Action research: its nature and validity. Syst Pract Action Res 11(1):9–21
29. Basili VR, Caldiera G, Rombach HD (2002) Experience factory. In: Encyclopedia of software engineering. Wiley, New York
30. Schneider K, Von Hunnius J-P, Basili V (2002) Experience in implementing a learning software organization. IEEE Softw 19(3):46–49