

Keeping User Centred Design (UCD) Alive and Well in Your Organisation: Taking an Agile Approach

Colette Raison and Snezna Schmidt

Adept KM Pty Ltd, 118 Gibraltar Street, Bungendore, NSW 2621, Australia
adeptkm@apex.net.au

Abstract. Using the analogy of user centred design (UCD) as a garden, we explore how to establish, grow and cultivate it to maturity in an organisation. We consider the importance of: having a clear and agreed intent and scope at the start; understanding the environment and culture; planning for success; focusing on the expected outcomes at each iteration; dealing with barriers and risks as they occur; implementing quickly in a scalable manner (according to the Agile methodology); conducting regular 'health checks'; reporting progress; and celebrating achievements along the way.

Keywords: Agile methods, Scrum, Usability, User Centred Design.

1 Introduction

Just as with other organisations around the globe, Agile software development methods are becoming increasingly popular in Australian public sector organisations. Their appeal lies in the fast turn-around from business specifications to developed software and a focus on development rather than documentation. However, given an increasing focus on citizen-centric service delivery, many public sector organisations have discovered that a product developed to suit only business needs may not be usable, useful, or satisfactory from an end user (the voter) perspective. Within the last fifteen years, project managers in Australian public sector agencies have begun to experiment with Agile approaches which included user centred design (UCD) techniques in order to satisfy the political voter pressures and (the user) as well as their agency's business and management stakeholders.

While the intention to include UCD as part of an Agile process to improve quality and performance in systems is to be applauded, it is not an easy task and the outcome is not always achievable. Promoting Agile UCD presumes that there is an existing, and available set of UCD resources from which to draw upon and that the organisational processes, culture and maturity will allow for it to occur. The level of UCD experience and the prominence that the organisation accords to UCD will influence the success of implementing Agile UCD systems design and development.

UCD techniques work best when the practitioners are skilled, supported by management and have the capacity to partner with their Agile cohorts and other teams in the organisation. The 'UCD Garden' in an organisation must be tended, nourished and maintained regularly, like all gardens, in order for it to thrive, become well established and effective.

1.1 Providing context

Prior to any decision to embark on Agile UCD, organisations need to be clear about what it is they are integrating. To provide some context for this paper, the basic definition for UCD as taken from Webopedia¹ is that UCD is a design philosophy where the end-user's needs, wants and limitations are a focus at all stages within the design process and development lifecycle.

UCD is taken to encompass User eXperience (UX), and well as Usability and Interaction Design and to include Usability Evaluations. It is embodied in the ISO-standard 13407 Human-centred design processes for interactive systems. UCD is an interactive, iterative process with key principles [14] such as:

- The active involvement of users and clear understanding of user and task requirements;
- An appropriate allocation of function between user and system;
- Iteration of design solutions; and
- Multi-disciplinary teams.

Disciplined Agile software development is said to be an iterative and incremental approach to software development conducted by self-organising teams within an effective governance framework with stakeholder input and with just enough documentation to meet their needs. [2] Essentially, it includes any iterative design and development process that aligns with the values in the 'Agile Manifesto' [3] where value is assigned more to:

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

The popularity of the Agile Scrum method is well documented [8] and is the focus of this paper when considering Agile UCD. Scrum methods include an initial meeting to identify a Product Backlog (the entirety of the work to be done), a series of Sprint cycles (2- 3 weeks) to develop software components and daily Scrum 'stand-up' meetings facilitated by a Scrum Master .

This paper explores several projects in a number of Australian public sector organisations that integrated UCD with Agile Scrum processes. It provides practical insights about what worked and what did not and it makes some recommendations about the need to establish, support, grow and maintain UCD capability so that it works effectively in the Agile environment.

¹ Webopedia at http://www.webopedia.com/TERM/U/user_centered_design.html accessed on 28 February 2013

Our research questions were:

1. How is Agile UCD work (or UCD generally) conducted in projects?;
2. Why are some Agile UCD projects less successful (or perceived as less successful) when it comes to the UCD components?; and
3. What (if any) are the common elements shared by projects?;

The next section describes the related literature studies. Section 3 details the research method. Section 4 describes the findings and Section 5 concludes the paper with suggestions for further work.

2 Literature Review

There seems to be an increasing consensus that Agile UCD is a more effective method than either Agile or UCD alone. [8] [16] Early discussions with proponents from both fields (Kent Beck (Agile) and Alan Cooper (Interaction Design)) in 2002 [11] highlighted the process differences in the two approaches, but also seemed to move towards an embryonic consensus on the techniques that could be used as part of an integrated approach.

A systematic review of the existing literature on the integration of Agile software development with UCD approaches conducted in 2010 [14] highlighted how far the integration has progressed from those early days.

The consistent findings from this review and prior research [16] are that:

- The development and the UCD work must be done in two discrete streams with UCD work to always be at least one Sprint ahead;
- Teams must maintain the ‘Big Picture’ overview of the project throughout the design and development iterations; and
- It is possible to conduct some research and do quick conceptual design work (e.g. Little Design Up Front - LDUF) at Sprint 0 while the developers are also planning and arranging their infrastructure and operating environments.

The interrelationship between the UCD and Agile streams is illustrated by Figure 1.

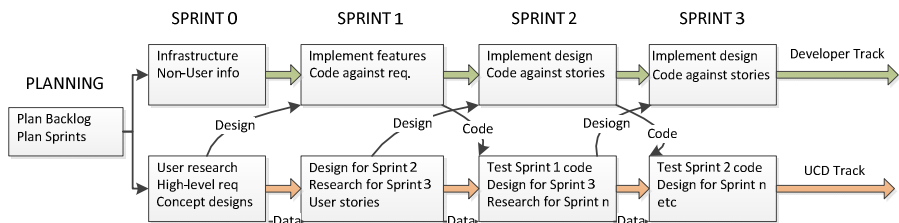


Fig. 1. Agile UCD Process

From a UCD (analysis) perspective, Sprint 0 involves research, a brief requirements-gathering exercise, and clear goals and a clear vision (Intent Statement). The development team prepare the architectural infrastructure and arrange the lightweight agile communication tool (JIRA, TFS, or similar). Developers start coding when the UCD team provides their brief documentation, conceptual design and user stories.

When Agile UCD is done well, it provides demonstrable gains for the organisation [7] [8], but success is not always guaranteed. In some organisations UCD is allocated a less prominent role, is not championed, is under resourced and developing ongoing capability is not seen as an essential organisational priority. This has flow on affects for the individual Agile UCD projects. In other words, the UCD Garden is neglected.

The current research illuminates what is often unseen – the incorrect assumptions about the UCD element of Agile UCD. The exploratory based research using case studies highlights instances of less than optimal projects and provides possible options for rectifying UCD-related issues in future projects. To put it simply, well-tended UCD Gardens provide excellent harvests.

3 Method

This research was very clearly exploratory in nature. We were aware that there were problems with some Agile UCD projects but the extent of the problems and the impact on the organisations (and people) were unknown until subsequent analysis was conducted following the closure of the projects.

We elected to use the case study method [13] because we wanted to find out how Agile UCD was conducted in the projects and why some projects were less successful in their UCD integration efforts. We also wanted to find out if there were any perceived commonalities related to UCD issues shared by those the more successful (and less successful) projects.

During the execution of the projects we were able to speak to some of the team members and we observed interactions between project stakeholders. We also viewed and discussed the tools used to communicate and undertake the work and the project documentation (e.g. UI Specs, one-page Intent Statements and Storyboards).

The roles associated with the projects teams varied, but they usually included:

- Business Stakeholders (Executives and Committee Representatives);
- Project Owner;
- Project Manager (also Scrum Master);
- Team Member (Developer/s and Architects);
- Team Member (UCD e.g. UI Designer, Accessibility, User Researcher);
- Team Member (Business Analyst – sometimes part of the UCD team);
- Subject Matter Experts (Business Representatives – called in as needed); and
- Testers (sometimes remote – who received products to text)

The six focal case studies and the projects are shown in Table 1. Each of the projects related to the design and development of systems (or platforms) within Australian Public Service (APS) agencies. The projects were undertaken from 2010 to 2013 and most had durations of less than 6 months.

Table 1. Six Agile UCD Projects in Australian Public Service agencies

Agency	Project	Type (Agile UCD)	Success
A	Develop a data capture system	Agile UCD	Yes
B	Implement records and sites	Agile UCD	Yes
C	Implement a reporting system	Quasi Agile, min UCD	No
D	Provide a ‘single view of client’	UCD, min Agile	Partial
E	Develop an Intranet solution	Agile UCD	No
F	Develop electronic lodgement	Quasi Agile UCD	Partial

4 Findings

4.1 Agency A – Agile UCD - Successful

Agency A is a mid-size and geographically dispersed public sector organisation with a high public profile and international reputation. The project was run largely by the development team who were well versed in Agile Scrum, but unused to UCD. Once the benefits of UCD had been explained and presented to Executives, UCD was endorsed as a core part of the overall analysis process. The team comprised the roles mentioned earlier in this paper and also used the full Scrum method and tools.

Initially, the project commenced with the Development and UCD Analysis being undertaken within the same Sprint because the developers were used to working this way. This meant that there was no time for UCD and analysis personnel to do any background research, to provide the overall intent and vision and to do any analysis on even high level user requirements.

Following a review, it was decided that the first Sprint was a ‘practice run’ to see how best to incorporate UCD into the mix. The project then adopted the process outlined in previous research where the Development and UCD (and analysis) streams were undertaken in parallel, but with the UCD (analysis) team always one Sprint ahead. The focus was on quick turnaround, minimal documentation and user stories.

What Worked?

- UCD eventually being conducted a Sprint ahead;
- UCD user stories included in the electronic Scrum system;
- UCD *brief* user research and artefacts (Intent Statement) during Sprint 0; and
- UCD and BAs working together and being co-located with developers.

What Didn’t Work?

- Initially doing UCD in the same Sprint as development; and
- Reliance on early Scrum tools (JIRA & Grasshopper later used for visibility).

4.2 Agency B Agile, Limited UCD – Unsuccessful

Agency B is a newly created small public sector organisation. It did not have the usual formal structures, governance, policy or procedures in place that would be

found in a more mature organisation. Resource limitations meant hiring external consultants and contractors who brought their own models and processes.

The project had experienced several iterations over the life of the agency – all failing to deliver on expectations. Service providers and team members were constantly changed at each review and remediation session. Although purporting to use an Agile Scrum approach, in reality this was only a loose ad-hoc arrangement. UCD had no visible champion and all UCD involvement was disregarded.

What Worked?

- The initial request for UCD services was appreciated; and
- UCD artefacts, design principles, strategies and approach (although not considered useful in this project) were used in all subsequent projects..

What Didn't Work?

- UCD was not co-located with developers and often prevented from access;
- UCD seen as a 'fluffy' add-on or (worse) a project 'blocker';
- UCD usability evaluations on prototypes seen as irrelevant as the prototypes were being shown to business stakeholders for 'feedback'; and
- Usability (and Accessibility) were not as important as getting a product out.

4.3 Agency C - Agile UCD – Successful

Agency B is a small public sector organisation based in one capital city. It was undergoing change to its organisational model and there was a focus on implementing enterprise-wide systems and processes.

The project had experienced several iterations over the life of the agency – all failing to deliver on expectations. There was a churn of project team members until the final project revamp – when a PM with UCD and Agile experience was hired. Once this happened, an effective Agile UCD processes was implemented using Scrum.

What Worked?

- UCD conducted one Sprint ahead;
- Developers using Sprint 0 to plan and setting up infrastructure (non-user related);
- UCD user stories included in the electronic Scrum system;
- UCD *brief* user research and artefacts during Sprint 0; and
- UCD and BAs working together and being co-located with developers.

What Didn't Work?

Earlier project instances with no UCD and no 'real' Agile capability

4.4 Agency D – UCD and Limited Agile – Partially Successful

This high profile public organisation dealt with emotive issues and was receiving negative publicity. Criticisms related to the organisation's ability to respond quickly

and to provide a consistent and appropriate client service. UCD was introduced as a panacea to address the client service issues through the inclusion of actual end users, however the introduction of UCD techniques was opportunistic and often personality driven. The iterative nature of UCD led to a recognition that there was a need for more agile processes but the implementation of an actual Agile environment was poor. The focus was on incorporating end users into existing processes rather than on changing the existing processes to make them agile and end user centric.

What Worked?

- Initial ‘Sprint’ was non-technical and involved data collection and analysis;
- Actual end users were involved;
- Development was iterative and changes from business and users were included;
- Stakeholders were given a voice at planning and update meetings;
- Design artefacts were concise and quick to produce and update; and
- Successes were well publicised.

What Didn’t Work?

- Project team members had limited experience with Agile methods;
- Communication between teams was ineffective;
- Stakeholder management was not effective; and
- Changes between each iteration were not prioritised or managed well.

4.5 Agency E – UCD and Quasi Agile - Unsuccessful

This was a high profile intranet redevelopment project in a large, distributed organisation which delivers essential services to clients who are often very critical of service delivery. Agile development was popular for business and executive stakeholders who were keen to see quick and tangible outcomes. UCD was supported by a limited number of sponsors and its implementation did not consider the existing organisational maturity, processes and governance structures. Initial UCD projects used expensive, external staff, promoted UCD as a ‘status symbol’ (e.g. ‘current best practice’) which made it inaccessible to internal project staff and caused resentment. Poor implementation did not create a sustainable outcome.

What Worked?

- Actual end users were involved; and
- UCD was integrated into each ‘Sprint’.

What Didn’t Work?

- UCD was not well integrated in the organisational governance model;
- Focus on documentation was too high;
- Organisational maturity was not considered;
- Internal staff were not included in UCD implementation;
- Communication between teams was ineffective; and
- No process to measure the value of UCD within the Agile process.

4.6 Agency F – Quasi Agile and UCD – Partially Successful

This organisation is compliance focused and encourages applications from its customers – previously mostly in hard copy. Development teams were using an Agile approach but this was inconsistently applied and was not well documented.

The adoption of UCD was considered at a time when external stakeholder support was important. It was hoped that UCD would address the perceived issues with Agile processes. UCD was introduced by a well-respected sponsor who had seen its value demonstrated in other organisations, but support for UCD was inconsistent and it was unfairly ‘blamed’ when existing issues with Agile did not improve.

What Worked?

- Actual end users were involved;
- Development was iterative;
- Business and executive stakeholders were given a voice at meetings; and
- Design artifacts were concise and quick to produce and update.

What Didn't Work?

- Introducing change on top of existing unstable processes;
- In-effective or not timely communication (e.g. dispersed team); and
- UCD practitioners not familiar with Agile approach and vice versa.

4.7 Summary of Findings

A significant problem was that UCD was seen as an optional ‘add-on’ or even a blocker to the real development work. The findings for the less successful projects replicated the findings from projects in other organisations [17] where the lack of strategic support for UCD posed problems.

Co-location within an Agile project team and a collaborative team culture was an important finding and is an important Agile UCD principle [5]. When not co-located, UCD was seen as irrelevant or purely a checkbox ticking exercise.

Replicating results from other research [16] Agile UCD conducted in Sprints requires that UCD and analysis is done at least one Sprint ahead.

Perhaps most importantly, analysis of the findings from the projects indicated that success with one particular Agile UCD project does not create or support a sustainable long-term approach for the organisation. Significant cultural change, ongoing sponsorship and an innovative, incremental and holistic approach is needed.

5 Conclusion

Research shows that if Agile methods and UCD are successfully integrated within a project team, Agile UCD is more likely to deliver benefits to the business and to the end user as well. [5]. The research conducted over the past decade shows that as Agile and UCD practitioners we have moved beyond asking why we should integrate UCD and Agile. The benefits in this combined approach clearly demonstrate value to organisations.

We have also moved beyond asking how to integrate them. Agile UCD offers a range of options for how this might occur, from selecting artefacts for small jobs right through to parallel streams for development and for UCD in short Sprints. Research conducted by colleagues has also helped to answer who is involved in Agile UCD and who our customers and users are. Roles are explicitly specified in various Agile approaches and we can identify the differences between customers, stakeholders and users and can recognise the circumstance and need to involve all of them.

We now need to ask what must be in place to achieve success – not only for the project, but for the wider organisation. We need to ask what can be done to ensure UCD is operating well in the projects and within the agency itself. Put simply, we now need to look at the best ways of establishing, growing, and cultivating UCD in our various agencies so that any future Agile UCD project can be assured of success.

This research was exploratory. It provided practical insight into Agile UCD projects and highlighted some outcome project commonalities; however it cannot enable any correlations to be made amongst project elements (variables). Empirical research is now warranted to consider in more detail those common elements contributing to the success or otherwise of an Agile UCD project.

References

1. Agile Defence Adoption Proponents Team, The Business Case for Agile Methods, Association for Enterprise Information (2011), <http://www.afei.org/WorkingGroups/ADAPT/Documents/Business%20Case%20for%20Agile.pdf> (accessed on February 28, 2013)
2. Ambler, S.: (undated) Disciplined Agile Software Development: Definition, <http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm> (accessed on February 28, 2013)
3. Beck, K., et al.: Manifesto for Agile Software Development (2001), agilemanifesto.org (retrieved on February 28, 2013)
4. Blomkvist, S.: Towards a model for Bridging Agile Development and User – Centred Design. In: Seffah, A., Gulliksen, J., Desmarais, M. (eds.) Human-Centred Software Engineering – Integrating Usability in The Development Process, Springer, Dordrecht (2005)
5. Chamberlain, S., Sharp, H., Maiden, N.: Towards a Framework for Integrating Agile Development and User-Centred Design. In: Abrahamsson, P., Marchesi, M., Succi, G. (eds.) XP 2006. LNCS, vol. 4044, pp. 143–153. Springer, Heidelberg (2006)
6. Eklund, J., Levingston, C.: Usability in Agile development, UX Research, pp. 1–7 (2008)
7. Fox, D., Sillito, L., Maurer, F.: Agile Methods and User Centred Design: How These Two Methodologies Are Being Successfully Integrated in Industry. In: Agile 2008 Conference, pp. 63–72. IEEE Computer Society (2008)
8. Hussain, Z., Holzinger, A., Slany, W.: Current State of Agile User-Centred Design: A Survey. Institute for Software Technology, Austria (2009)
9. ISO, 13407: Human-centred design processes for interactive systems, International Standards Organisation, Geneva (1999)
10. McNeil, M.: User Centred Design in Agile Application Development (2006), http://www.thoughtworks.com/pdfs/agile_and_UCD_MM.pdf (accessed on February 28, 2013)

11. Nelson, E.: Extreme programming vs Interaction Design (2002), [http://web.archive.org/web/20070313205440/](http://web.archive.org/web/20070313205440/http://www.fawcette.com/interviews/beck_cooper/), http://www.fawcette.com/interviews/beck_cooper/ (accessed)
12. Patton, J.: Hitting the Target: Adding Interaction Design to Agile Software Development. In: OOPSLA 2002 Practitioners Reports, pp. 1–7. ACM Press (2002)
13. Rowley, J.: Using Case Studies in research. *Management Research News*, pp. 16–27 (2002)
14. da Silva, T., Martin, A., Maurer, F., Silveria, M.: User-Centred Design and Agile Methods: A Systematic Review, pp. 77–86. IEEE Computer Society (2011)
15. Sutherland, J.: The Scrum Papers: Nuts, Bolts, and Origins of an Agile Framework. Scrum Inc., Cambridge (2012), <http://jeffsutherland.com/ScrumPapers.pdf> (accessed February 28, 2013)
16. Sy, D.: Adapting Usability investigations for Agile User-Centred Design. *Journal of Usability Studies* 2(3), 112–132 (2007)
17. Vukelja, L., Opwis, K., Muller, L.: A Case Study of User-Centred Design in Four Swiss RUP Projects. *Advances in Human-Computer Interaction 2010*, 1–10 (2010)
18. Williams, H., Feguson, A.: The User Perspective: Before and after Agile. IEEE, The Computer Society (2007), http://agileproductdesign.com/useful_papers/ucd_perspective.pdf (accessed on February 28, 2013)