Usability in Agile development

John Eklund and Ciaran Levingston
UX Research
info@uxresearch.com.au

ABSTRACT
Agile methods of development prescribe rapid prototyping and minimal documentation, and are particularly suitable for creating web interfaces for businesses, encouraging throw-away prototypes and allowing designers the freedom to be more creative. There are additional challenges in including usability techniques in Agile development, as it has multiple iterations and often substitutes the client as the user-advocate. This paper examines some of the ways Agile methods are being practiced in web development and discusses approaches to user centred methods that meet the challenges of rapid development to assure the usability of systems and ultimately customer satisfaction. Some of these methods are part of a more recent trend sometimes called Usability 2.0, which aims to make user centred methods more accessible, practical, affordable and of demonstrated value to the strategic directions of a business. As we shall see, none of these methods are really new, but useful adaptations of a well established approach that is proven to work well.

Keywords
HCI, User Centred Design, Agile, Usability, Software Development Methods.

1. INTRODUCTION

1.1 About this paper
This paper examines modern approaches to Agile development, its theory and common practices and examines ways that usability practitioners can effectively integrate their services into Agile development lifecycles. The research is a result of discussions with a number of software development organisations that provided a picture of some of the particular practices they use as part of their Agile approaches. The authors themselves are not developers, but usability practitioners and their aim is to give an overview of some Agile methods in practice. The paper briefly covers usability in the traditional software development lifecycle, and some of the considerations that will need to be addressed to allow usability to successfully integrate with Agile development approaches. Finally, we propose some tangible models for the ways that outsourced usability providers can engage with development teams that are following Agile approaches to make usability services more accessible and understandable.

1.2 A climate of change
We are in a period of significant change with the way technology is affecting our lives. Our children are growing up with access to new tools for business and recreation beyond anything we could have imagined a decade ago. Our society

now enjoys fully online banks, mobile internet and ecommerce on a scale of change that is unprecedented in our modern world. These changes are driven by the demands of users and the rapid pace of technological change. In the last ten years we have also seen dramatic changes in software development processes. Our once immature industry is changing and it is no longer acceptable nor advisable for development agencies to ignore sound development practices. Clients demand solid project management as well as an assurance that final deliverables will meet their expectations as well as those of their customers. The usability industry has taken a key role in software quality assurance. It is now generally accepted that some form of user testing must be built into development lifecycles. Usability and user centered design are understood, frequently used and it’s business benefits recognized by developers and business stakeholders.

In development circles, timeframes are contracting with clients demanding implemented solutions to be delivered in less elapsed time. These systems are often web-based as the suite of technologies available for rapid web development has increased and more self-service functionality is being made available to the end user. Websites are becoming far more the engine-rooms of businesses, no longer the static brochure-ware meant to attract customers, but attract and transact with them, making business easier, less costly and more globally delivered. All this has driven changes in software development, as more sophisticated software companies enter the marketplace, merging media and digital offerings, and seeking to distinguish themselves with solid methodologies and time-driven successes.

The key question for companies like UX Research, providers of outsource usability services, is “How can we adapt to meet this changing environment?”

2. THE EMERGENCE OF AGILE
2.1 The meaning of Agile development
Agile appears to be one of those terms that get used sometimes where it shouldn’t, and it is being applied in a great variety of settings and in many different ways. Possibly this is acceptable, after all it really means “flexible”, “adaptable”, “nimble”, and this can also be applied to its definition. How agile development is done in practice depends on many factors, including the particular skills of the development team, the particular project that it is being applied to, and the needs of the client. Some may see it as a way of shortcutting expensive processes early in development, such as specification, in order to save budget. For others it’s an exciting new way to get things done faster and getting deliverables to the client earlier. It requires quite an organisational shift within development teams that needs the cooperation of project managers, designers and importantly, the client. One clear outcome of our investigation in Agile is that
clients need to be onside and committed to the process for it to work.

At its base level we can say that Agile is a type of iterative development.

![Figure 1. Agile is based on iterative development](image)

Agile aims to move forward the design phase in a development lifecycle and replace or reduce the amount of time spent documenting a system before prototypes are developed. We will deal with this aspect of it, it’s impact on the design process, but it is also relevant to ask “why has it emerged?”

2.2 History

It is worth remembering that the core of Agile is not really new, and that there is a long established history of Incremental Development (ID or iterative development). The earliest documented instance of a project that used multiple iterations was in the 1930s, and it seems it was in the 1990s that the notion of iterative development superseded the Waterfall model (Larman and Basili, 2003). Agile became a properly established method when in 2001 an organisation called the Agile Alliance was established, with goal to share information and techniques on this method. Like most things, including iterative development or Usability 2.0, Agile in the modern world is not new, but it is different. It’s on the web; it’s developing business critical systems for Small to Medium Enterprises (SME); it’s done faster and with new tools.

Agile methods have emerged as a response to a changing environment for development. New technologies allow designs to be created and experimented with easily, later stages of development can be entered earlier, greater access to a range of prototyping tools and developers to be more creative. It acknowledges that web-based business critical systems must be developed rapidly but with sound processes, and it allows changing requirements to be discovered through designs (Eklund & Lowe, 2002).

2.3 Development models

Development models like Waterfall (Parida, 2006) depend on building a detailed specification on which design is based. These models are linear in that they have no feature to return to the designs and make changes, nor validation that requirements are complete. Crunch-time comes on delivery of the finished system, when it's typically the case that the client recognises that requirements are missing, introduces new requirements, or doesn’t approve of aspects of the interface design. It may be that this model is suited to developing large-scale systems but it also depends on stable requirements that are correct. There is clear statistical evidence indicating the importance of getting the requirements right and that the cost of modifying a system increases exponentially as the product gets closer to release.

Establishing clear and accurate requirements is a difficult and time consuming process. This means that in a more traditional model such as the V model, a lot of the work is done in the requirements analysis phase and in developing detailed specifications, and unless these are correct and stable, the system will fall short of expectation at some point.

Agile by contrast recognizes that requirements will change and this stage is not removed but simplified and blended with the production of high-level designs.

So Agile says “requirements will always change, we accept that and structure our project planning to allow for it.

![Figure 2. An iterative development lifecycle](image)

In an Agile world, the stages of gathering requirements and designing overlap. Agile prescribes a process of gathering some basic requirements, start designing, and as you do so the remaining requirements will emerge.

When you deconstruct Agile development, at its base level it is an iterative development methodology. The basic structure of Agile development is simple, cycle through the specification, design and evaluate stages several times refining requirements at each iteration. In the evaluation stage, whether it is an informal review of designs or formal usability testing, it is important that it be performed regularly and done with a certain degree of independence.

If we say that Analysis includes eliciting requirements, namely information gathering activities, then we could almost remove all but the high-level requirements from the process. In some ways that’s what Agile does: De-emphasise requirements, and embed them in Analysis and Design.

Iterative development can be done by creating successively deeper designs, starting with flat screens and moving to more functional prototypes, until the final system is delivered. This is a standard Agile approach and has the benefit of allowing
clients and users to provide feedback on prototypes as development progresses.

Each time we perform an iteration from concept to functional system there is some form of evaluative process, which would appear acceptable in theory but there is a potential problem with this approach.

2.4 Key differences in Agile Iterative development

In traditional development the system is built in layers, with all the functionality delivered in the resulting product. A good analogy is building a house. You start with the foundation that encompasses a complete set of functionality and create successive layers from the ground up.

![Layered approach in traditional development](Image)

**Figure 3. Layered approach in traditional development**

Agile iterates differently, selecting core functionality to have several stand-alone deliverables, allowing the client to adjust scope at each iteration based on what has been delivered. In this way it aims to deliver something that’s working and complete early and often.

So in Agile we start designing early and spend less time documenting. The most important aspect is that the Specification (or Requirements) and Design phases are overlayed, they co-exist. Specifications themselves become less formal, easier to produce, are often presented as designs themselves, and develop as the designs become more detailed.

![Functional approach to Agile development](Image)

**Figure 4. Functional approach to Agile development**

2.5 Benefits

There are several important advantages of this approach: As requirements are usually expressed as documents, which are often inaccessible to clients, Agile means designers spend less time creating documents and more time building (sometimes disposable) prototypes that can be used to elicit requirements. Many of us will have been on one side or the other of this scenario. It’s not until designs are presented that the client is able to articulate what it is they wanted. No matter how carefully a specification is done, until the tangible designs are produced, the client has trouble providing feedback that is valuable in gathering system specifications. This is often due to the density of specification documentation which clients and business stakeholders find difficult to interpret.

An advantage of this approach is that it reflects practices in the real world. Many businesses are unsure of what functionality they require in a system and they do not have a clear idea of what the User Interface will ultimately look like, so Agile attempts to overlay time consuming requirements gathering with designing in order to elicit such system specifications as may be required.

This gives clients a feeling of security in that they can see system designs early on in the project and thus provide more tangible input throughout the life of the project. Clients get greater a sense of involvement in the project and a higher level of control of the direction of the project as a whole.

Agile also has the benefit of allowing the designers more freedom to experiment with their designs as they are less bound by a specification. This means there is more scope to be creative, allowing multiple sketch solutions to be put forward.

3. AGILE IN PRACTICE

3.1 Agile and team dynamics

Ultimately a successful project is about a deliverable piece of software which is created by people and key to that is successful people management. Everyone has a role to play, and Agile methods prescribe some significant changes to traditional team dynamics.

The client seeks the expertise to create a system that will meet his business goal. He has a view of what he wants and what users will do in his system, and starts his project with enthusiasm and energy. Then there is the project manager who represents the best interests of the development organisation, making sure the deliverables are met by those on his team, managing client expectations of scope, and generally distributing project plans as they are revised. He understands the business and technical constraints and is key to managing relationships. They are joined by Creative people, designers who are responsible for developing the user experience and creating exciting and dynamic visuals that meet client needs.

The team also consists of an outsource usability practitioner or interaction designer who provides an independent analysis of designs and makes structured recommendations based on user responses and against best practice and usability standards. He
is an advocate for the user and is frequently commissioned by the client. In addition there is the technical team without whom we wouldn’t have any HTML code or a functioning server. The key point is that all these people have different skills, goals and attitudes and getting them all to work together can sometimes pose quite a challenge.

In traditional Development, the project manager oversees the work of the designer and is in charge of the relationship with the client. Communication is principally done through the project manager.

In Agile, the client is in closer contact with the designer, and the project manager becomes the overseer of the process, less directly involved in the detail of the design. This structure affords greater client involvement and gives them the opportunity to take the role of advocate for the user group. The challenges for the project manager in this situation are to ensure the designer and client work well together and stay on scope.

4. USER CONSULTATION MODELS

In some organizations user consultation is done by the designer/developer by gathering business requirements from the client by eliciting “stories” which are the various tasks a user would undertake on the system. These are often written on post-it notes and become a sort of default set of user and functional requirements that can be embodied in designs. Clearly in this scenario it is still client conjecture as to what a user would want from the system which forms the basis of user requirements.

A better method is to include direct consultation with users, a sort of participatory design process, where the client as well as a group of users review designs and provide feedback. Now we have direct involvement of users, however two key things are missing from this scenario - an independent person to interpret and moderate the user feedback, and feedback from best practice perspective.

The final and best method for organising the team to produce high quality designs is to have an independent usability analyst provide input to the designs throughout and at key stages of the design process. The benefit of this is to ensure the maintenance of independence and usability standards. An experienced usability analyst will have had exposure to and interacted with many systems and can advocate for best practice and standards throughout the design process.

4.1 A working definition

In summary, Agile methods are ones that organise development teams differently to promote open communication and involve multiple development iterations. It anticipates that clients are uncertain of their requirements and that requirements will change throughout the life of the project. In Agile methods scope creep does not mean a poorly specified project, but is built into the development methodology. The process of designing earlier allows requirements to emerge, change or be abandoned. A key differentiator is in the way the system is iterated in groups of functionality. This allows the delivery of a working system, not just a prototype, which contains core functionality. This allows projects to be signed off in stages and scoping of the project to be simplified. This gives us a working definition:

Agile is a term used to describe a type of iterative development that relies on designing early as a means of discovering requirements, producing usable systems in stages of functionality designed to meet the uncertain needs of modern business systems.

5. ASPECTS OF UCD IN AGILE

This section discusses the role of usability in Agile development. Usability as a word has many interpretations – it’s a discipline, an industry, a method, but it can be thought of as a statement of quality about a system. We can say something is highly usable if users can complete their tasks with ease. In other words, it’s a measure of quality that is judged by the extent to which the design and functionality of a system meet user requirements, whether these requirements are explicit or not.

5.1 Usability in theory

In User Centered Design, the evaluation process clearly involves an evaluation of designs against user requirements, and this is typically done by an independent analyst.

When we undertake usability testing there are two main sets of benchmarks against which the designs are judged. The first is the needs, requirements and perceptions of users. They are given common tasks to perform and the degree to which they can complete their goal, and their satisfaction in doing so, forms much of the basis of a study. It mixes observation on tasks with interview and questionnaire, and this provides a highly contextual view of the system and how users respond to it.

However, usability testing is not all about what users say and do, there is a significant component of expert analysis, particularly when it comes to making recommendations. The usability analyst will have undertaken many usability studies and be aware of best practice and well as guiding principles relating to consistency, feedback and other principles. Issues are identified and analysed against a set of criteria, described in standard terminology such as navigation, page layout, aesthetic design and so forth. It is knowledge and understanding of these heuristics by the outsource usability analyst which highlights their value.
One way to look at usability in practice is in terms of perspectives and benchmarks. Perspectives are like layers of quality. Yes the system needs to be usable, but also users must be motivated to use it. This is called user acceptance.

Benchmarks are an essential part of any usability evaluation. These are the standards by which the quality of the design is judged. These are informed by design principles and usability principles first stated in Neilsen’s 10 usability heuristics - things like the visibility of system status. And of course, best practice is derived from these benchmarks and conducting many studies and seeing what works in practice.

5.2 Usability in the real world

Usability Companies such as UX Research typically offer a range of quality assurance services across the development lifecycle, from the requirements gathering phase through to design and build. This gives the impression that they are busily engaged across the whole project doing various good things to make designs more usable and ultimately serve customers.

The reality is somewhat different - our experience is that external usability testing tends to be at one, possibly two places in development lifecycles. The larger, higher profile projects we have worked on have tended to increase the numbers of users in testing, rather than the number of test cycles. In addition, despite the acknowledgement that usability evaluation is invaluable in the early design phase, in our experience it is placed much later in the process when most of the design is locked down. This reduces the usability study’s focus to detail elements at a page level, rather than more fundamental aspects such as functionality or information architecture.

The reasons for this are twofold. It’s partly that some see it as a validation exercise, which is fine if user centred design has been undertaken in development. It is also that project development teams have found it difficult to engage external usability people. It is seen as quite a chunk of time taken away from activities directly related to development, and developers don’t want a critique of their work until they are certain they are ready for it.

If the purpose of usability testing is validation, this attitude makes some sense. This mindset is however slowly changing and developers are starting to see the value of external usability at different stages of development. Depending on the stage at which it takes place, usability evaluation is different. Early in the development lifecycle it will be more exploratory, and contribute more directly to designs, while later on its more a matter of validating the quality of the user experience for business owners. It will almost always have a strategic component, offering advice to business owners that will improve market positioning or customer retention.

5.3 Usability in Agile is a challenge

It is no surprise that organisations are struggling to see how usability services can fit into Agile development lifecycles. Many were just becoming comfortable with incorporating usability into normal development methods and with the tendency to perform larger external testing later in development, it is not surprising that development companies are having difficulty fitting external testing into their project plans and budgets.

These are the challenges for usability in Agile. Now there is a changed team dynamic, traditional team roles have shifted, and the client, occasionally supported by consultation with real users, acts as the user advocate. Since user consultation is now more embedded in the development process, it is more difficult to recognize the benefit of external usability consultation. It becomes even more problematic to decide when to undertake external usability testing, as there are many iterations, in what can appear to be an ill-defined process.

6. STRUCTURING USABILITY SERVICES TO BE SUITABLE FOR AGILE

6.1 Principles and assumptions

In this section we discuss ways that usability practitioners can respond to this new development environment. We begin with an analysis of the characteristics of usability services that will fit with Agile development methods. Firstly, as the process in its initial stages is creative and flexible, it will be important to be able to call on usability analysts in a flexible way, which is dependant on the needs of the project and the issues that arise. This means having an arrangement with a usability provider already in place, knowing their process and methods and costs, and being able to commission them to undertake usability work quickly.

Secondly, such advice will need to be at a detail level, frequent and as needed by the team. This advice will be related to principles and standards and draw on the expertise of the analyst who has a knowledge of current and best practice.

Further, due to the fact that Agile methods involve multiple iterations, its likely that usability advice will be required multiple times, and testing will be executed quite quickly and for less cost. Project teams will not have the luxury of time to wait for a detailed report, and will seek input more directly into design decisions.

In forming a response to how the usability services of an external organisation can fit into Agile development methods, we start with the basic assumptions that external usability testing must be employed at some point as it’s a fundamental part of User Centered Design. We know it works, as it is a method that has been used for decades though it appears in various guises such as Usability 2.0. We know as providers of these services that we can add value in each and every instance but in making external usability testing faster and cheaper we cannot use machines or substitute junior staff as the quality of usability insights gained through testing is always going to rely on the experience of the analyst to make the appropriate judgments in each case. We think that the provision of usability services in an Agile environment is a matter of making usability testing more efficiently delivered.

6.2 A UCD approach for Agile

With a limited budget for usability testing on any project, traditional development typically included one or two rounds of user testing. In Agile development we aim to include expert advocacy and more rounds of smaller test events as required.
Expert advocacy is being available as a usability expert to comment on designs, solve design problems, be an advocate for the user, and ensure standards are being met. This is best done in short meetings usually without any report or making a comment or opinion by email. This has the benefit of solving simple problems, which do not require user testing to be identified and addressed. It also provides the usability analyst with more involvement in the design process allowing them to better plan for usability test events.

A good example of how advice can reduce time spent in formal testing is recently one of our consultants was meeting with a client to discuss what to include in the next round of testing. There was debate around some screen elements, but to the consultant the answer was clear, as the same issue had presented itself in a previous study for a different client. We convinced them to adopt these suggestions, several of them, feeling confident that we knew what would work. This occurred with a number of design elements, until it was realised that many of the issues on the list that were thought to require testing were now resolved. It was decided to delay user testing until some other aspects of the design had progressed. This example shows the value of time flexible advice closer to the design team.

Usability testing itself has to be shorter and faster, and that means cheaper. The whole lifecycle for usability testing can be as short as a week or even less, including five days to recruit the participants. Testing can be done in one day with up to eight sessions, using a portable usability lab on the client premises, with a report delivered to the client two days later.

Testing on the client site has its challenges but a test configuration can be created quite simply. The advantage this has is that business people have more opportunity to participate by watching the testing by dropping into the observation room when they can. Other organisations are making sessions more accessible through the use of various video streaming technologies, such as those in Morae.

The usability testing itself is also different. It is more exploratory at earlier stages with a more open test script and incorporates usability standards. In many ways the users are there to help the analyst find problems - for instance if we know a button is poorly labeled we won’t spend any time discussing that with user, but create tasks that direct them to other areas of the site to explore new problems. Including business objectives in user testing is also important. If a site’s business goal is to generate signups the focus of the testing is to find the points that cause most users to drop out of the flow.

The test sessions in Agile must be hosted in a flexible manner also, sometimes with a very consultative style that treats the user as a partner in discussing issues, which is a less pure hosting protocol that challenges users to confirm or deny mental models. In some testing protocols you might say that such questioning is leading, however the justification is that you only have a short time with the participants and you need to get as much feedback from them as possible.

In Agile, often there will be multiple designs to test, and so comparative testing becomes part of the process. In fact comparative testing is an important component as it allows the users to judge designs against each other rather than some ill-defined absolute standard. It is advisable to include some comparative components in all Agile testing, particularly in early iterations.

Finally, Agile demands that usability is closer to the design process while remaining independent of it. This is so user testing can add more direct value and have immediate input to designs. This can be done in a number of ways: The usability consultant as a design team member, called upon for occasional consultation as an advocate; Faster testing with a participatory design component aimed at eliciting requirements; testing on client premises, and making usability outputs more ‘design friendly’ with reports centred on recommendations supported by observation.

7. CONCLUSION AND RECOMMENDATIONS

In conclusion we offer a set of recommendations for those organisations wishing to embark on Agile development projects and concerned about how to include usability testing in what is a challenging and exciting approach to development.

Firstly, it is important to constantly remind ourselves of the first principle of User Centered Design – to involve users at several stages of development. If the design process involves the designer working closely with the client, then the use of external users as a sounding board for designs will obtain valuable insights. Give users the tasks they would undertake in the system and gauge their reactions. User involvement can be informal, but it must exist.

Secondly, we recommend the engagement of an external usability specialist to give advice as required. They can help solve design issues, particularly minor interface issues, speed development and save user testing time that can be used in areas that are more problematic. It will be important to consult with the usability consultant about the timing and scope of the testing that is required rather than commissioning testing against a set project plan.

Further, if the project prescribes one round of testing, ask for two smaller ones. If your usability provider suggests two rounds with 12 users in each, ask for 3 rounds of 8.
Lastly, when usability testing is undertaken, ask your analyst to gather requirements as well as test designs. Expect a more open and exploratory sessions early in development, and more validation later in development. Work the test script through with the analyst and endeavor to involve the business in observing sessions, which may be facilitated by testing on the client premises, to help engage the business. A report should be available two day after testing, which will allow designing to proceed. The report will always offer both strategic and design recommendations.

8. ACKNOWLEDGMENTS
The research in this paper is based on a series of industry seminars about Usability and Agile methods that were given by UX Research in May and June 2008, and at STANZ in August 2008. These were sponsored by The Web Industry Professional’s Association (WIPA) and Software Testing Australia New Zealand (STANZ) respectively.

9. REFERENCES