

Software Development Methodologies, Agile Development and Usability Engineering

David Parsons¹, Ramesh Lal¹,
Manfred Lange², Hokyoung Ryu¹

¹Massey University , ²FirstData Corporation



Research Question

- How can agile software development methods be integrated with the usability engineering lifecycle?



The Agile Manifesto

- “we are uncovering better ways of developing software by doing it and helping others to do it. Through this work we have come to value:
 - *Individuals and interactions* over processes and tools.
 - *Working software* over comprehensive documentation.
 - *Customer collaboration* over contract negotiation.
 - *Responding to change* over following a plan.
- That is, while there is value in the items on the right, we value the items on the left more”
 - Highsmith, J. 2002. *Agile Software Development Ecosystem*, Boston, Addison-Wesley.



Individuals and Interactions

- Generalists rather than specialists
- Strengths (or preferences) may be of benefit
- External experts or support contract with vendors or service providers can be used.
- Individuals will learn and master different technical skills by interacting with other team members when developing code
 - co-location
 - pair programming
 - regular reflection workshops



Working Software

- Working software can be used for conferences, pre-sales, or other customer related demonstrations
 - Nothing convinces more than running software
- Complementary to a shift in the nature of software architectures away from large monolithic systems towards systems built from small software components, (e.g. SOA)
 - Smaller teams work independently while collaborating on the common interfaces
 - Software architecture and methodology can interact and enable each other



Customer Collaboration

- Discount contract negotiation and emphasize collaboration among the stakeholders
- Collaboration:
 - Enables software to be developed quickly and delivered on a regular basis to the customers
 - Helps to make decisions based on consensus
 - Assists learning and knowledge sharing among team members for improvement and quality purposes



Responding to Change

- Meticulously following a plan will lead to an obsolete application
- Deviating from the plan as a result of new information is normal practice
- Customers (or customer proxies) are responsible for *what* gets implemented, e.g. features
- The engineering team decide *how* things get implemented
 - Conversation between them may include these topics, for example a product manager may suggest the use of a specific technology



Usability Engineering

- Usability engineering provides structured methods for achieving usability in the system development process
- The usability engineering approach generally consists of several procedural phases:
 - Requirements analysis
 - Design
 - Development
 - Testing
 - Installation.
- Understanding the user and the tasks, contexts, and environment in which the application will be used
- Iterative but not agile



Limitations of the Usability Engineering Model

- Usability engineering closely resembles the traditional software engineering approach in its formality and insistence on up-front analysis and design
- If usability engineering is integrated into a software engineering approach, the usability aspects of that system will still suffer from the same problems as the methodology into which it is integrated



Prescriptive Roles

- The usability engineer is a prescriptive role
- From the agile perspective there are no distinct roles, rather different activities within the same role
 - Merging of roles also leads to a more democratic structure
- One consequence of merging roles may be a significant reduction in specialist roles such as tester, architect and, of course, usability expert



Integrating Agile Methods and HCI concepts

- If traditional software engineering methods have not always integrated usability engineering, the same may also be said for agile methods
- It is not explicit which agile process would ensure the usability of a story or feature that would be implemented as a function



Real Customer?

- Geographical or financial limitations may mean the 'customer' is not actually the real end user of the application being developed
 - May be a quality assurance person, an agile coach, a product manager, a software engineer or a marketing analyst
- The major challenge, then, of an agile approach is how to identify the requirements of a system as accurately as possible from a customer who is not the actual end user and who will also lack the in-depth business knowledge that the actual users have.



“Minimal” Design

- Incremental UI design
 - Requirement: interface for entering an address that may need a field for the country
 - Iteration 1: simple text entry field.
 - Iteration 2: drop-down list
 - Iteration 3: clickable map of the world
 - Iteration n: enter a Google Maps link / send a text message from a GPS phone
- In contrast, usability engineering requires various upfront modelling and design alternatives before a best design is selected for implementation
 - Agilists believe that these artefacts do not deliver any value to the customer



Integrating HCI Practices with an Agile Approach

- The participatory and iterative aspects of usability engineering are similar to the agile approach
- Collaborative aspects of agile methods integrate well with user-participatory design practices
- Agile methods lack guidance on requirements elicitation, user interaction, and user interface
 - an agile customer equipped with a user-participatory design approach should be able to more comfortably guide the agile development process



Lessons From the Literature

- Software development processes can be improved by building on the best practices of different approaches.
- Agile methods are not a *set* of practices but a *system* of practices that support/facilitate/mitigate each other
 - e.g.
 - simple design / minimum upfront design require refactoring
 - Refactoring, in turn, requires a test-driven approach
- To integrate aspects of usability engineering into an agile approach, identify the points of synergy
 - e.g. integrating a user-participatory design approach into an agile method mandates a process that involves real users, which most agile methods lack



Experience Report - UtilSoft

- Development methodology is a combination of techniques from Scrum and XP
- User-participatory design integrated into development approach
- An improved UI may reduce the average call handling time by around 1 second.
 - Economics of scale can result in savings worth one full time employee's annual salary for some customers.
- A UI that “sells” is an important factor for the company's success, too



UI Expert Role

- User interface expert co-located with the development team (cross team role)
 - components for interaction, when to use Ajax, domain knowledge, customisation
- Product manager serves as the customer proxy
- Stories map to UI 'screen flow' sequence
- Engineering team + UI expert + product manager = consistent vision of what is both desirable and economically reasonable.



Outsiders and Co-location

- Outsiders
 - Regular assessment by an external consulting company which specialises in HCI
 - Per-customer customization.
- Co-located
 - product manager
 - UI expert
 - engineering team

Combining Approaches

Usability Engineering	Combined Approach	Agile Methods
<ul style="list-style-type: none"> Waterfall type approach, with analysis, design, implementation and deployment phases, but iteration within/across phases 	<ul style="list-style-type: none"> Use iterative development throughout 	<ul style="list-style-type: none"> Iterative approach based on working software
<ul style="list-style-type: none"> Documenting user scenarios (or user profiles and personas) in the analysis phase 	<ul style="list-style-type: none"> Merge user scenarios (or user profiles and personas) with stories throughout the cycle 	<ul style="list-style-type: none"> Requirements encapsulated in user stories
<ul style="list-style-type: none"> Specify full design (or partial prototypes) in design phase 	<ul style="list-style-type: none"> Allow working prototype to emerge 	<ul style="list-style-type: none"> Design emerges
<ul style="list-style-type: none"> Begin design with low fidelity prototypes including paper prototypes 	<ul style="list-style-type: none"> Use software for prototypes 	<ul style="list-style-type: none"> Prototypes are architectural and embedded in code
<ul style="list-style-type: none"> Assemble a multidisciplinary team to ensure complete expertise 	<ul style="list-style-type: none"> Ensure team includes UI expertise among other skills 	<ul style="list-style-type: none"> Business people and developers must work together daily throughout the project
<ul style="list-style-type: none"> Use surveys, field studies and usability testing in the testing phase 	<ul style="list-style-type: none"> Test the UI in the user context regularly, throughout all phases 	<ul style="list-style-type: none"> Drive development with testing and integration
<ul style="list-style-type: none"> Requires usability specialists 	<ul style="list-style-type: none"> Use usability specialists across multiple projects, either in house or by using external consultants or contractors 	<ul style="list-style-type: none"> Requires generalists



Summary

- Agile methods have not typically incorporated HCI and usability techniques and tools into their software development processes.
- Incorporating HCI and usability processes is important, because usability of the software product is an essential quality.
- While the traditional approach to usability engineering is incompatible with agile methods, it is possible to integrate certain practices from usability engineering into an agile approach.



Conclusion

- Usability engineering needs to adapt to the agile context, for example by using a more iterative approach, and by testing throughout the project lifecycle
- Agile projects need to adopt aspects of usability engineering, for example by incorporating user scenarios and including UI specialists in the team