Agile Software Development
Agile UX Work

Kati Kuusinen
Researcher @ TUT / Pervasive / IHTE
kati.kuusinen@tut.fi
Contents

1. Introduction / Motivation
2. Agile software development
3. User experience work as integral part of SW development
4. How is it done in practice – experiences from industrial development projects
What does **AGILE** mean in software development?

How can you produce good **User Experience** in an Agile software project?
Can you make money with it?
Will people want it?
Can you deliver it?

What is desirable?

What is feasible?
Technology

UX

What is viable?
Business

Product triangle edited from Cooper.com and Hyysalo, S. 2009
Will people want it? – How do we know?

- Let’s research it thoroughly and design it well
  - Big design upfront

- I have an idea – let’s just implement it and see
  - No design upfront

- Let’s do some research and test it soon in practice – we can always change it
  - Some/little design upfront
Waterfall Development

- Sequential
- Phase ready and correct when moving to next one
- Big design upfront: basically everything designed before implemented
Agile Methods

• Born in 1990’s when noticed that in some cases, waterfall is not an ideal model
  – You cannot get everything right beforehand
  – More lightweight methods were needed
  – Waterfall is still a good model for e.g. in industries where iterative work is costly
Agile Manifesto

...we have come to value:

- Individuals and interactions
- Working software
- Customer collaboration
- Responding to change

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

TRUE:
✓ Iterative
✓ Incremental
✓ Cooperative
✓ Short feedback cycles

FALSE:
✗ No planning
✗ Just coding
✗ No rules
✗ No documentation
Scrum Framework

Four ceremonies

Three artifacts

Sprint planning meeting

Tasks generated from sprint backlog

New functions that can be demonstrated

Sprint review

Sprint

24h

4 weeks

Daily Scrum

Product backlog

Sprint backlog

Katu guest lecture – Kati Kuusinen – Spring 2014

21-March-2014
Scrum Team Members
(adopted from Marie-Elise Kontro)

- **Product Owner (PO)**
  - owns the product (backlog)
  - Scope vs. schedule

- **Scrum Master (ScM)**
  - owns the process
  - "Team servant"

- **Development Team**
  - Delivers product increments
  - Cross-functional and self-organizing

- Other roles are stakeholders
Lean Development

Maximize the ability to create customer value, work less

Seven principles:
1. Eliminate waste
2. Amplify learning
3. Decide as late as possible
4. Deliver as fast as possible
5. Empower the team
6. Build integrity in
7. See the whole
Kanban

- Visualize
- Limit WIP
- Manage Flow
- Make policies explicit
- Implement feedback loops
- Improve collaboratively, evolve experimentally
Standard for HCD

Vision
Concept creation
Evaluation
Design & Development

Plan the human-centred process

Understand and specify the context of use

Evaluate design against requirements

Specify the user and organizational requirements

Produce design solutions

System meets requirements?

ISO 9241-210:2010
Contents

1. Introduction / Motivation
2. Agile software development
3. User experience work as integral part of SW development
4. How is it done in practice – experiences from numerous development projects
Agile User Experience Development in General

- Agile UX is user experience work that is conducted in accordance with Agile (and Lean) principles and methods, thus integrating (or merging) UX work and human-centered design practices with Agile (and Lean) development practices.

- The goal is to adapt an efficient way of working that constantly leads to desired user experience of the outcome.
  - The desired level of outcome quality and project effort naturally varies.

- Business, user, and technical requirements should be in line and support each other.
Agile User Experience Development in General

- UX work aims to Lean development; less unneeded features, less expensive correcting work
- Leads to cumulative savings in time, in maintenance costs, in development costs, training and manual costs
- Leads to better usability, efficiency, effectiveness and satisfaction
- Leads to better reputation, increased customer loyalty
- Systematizing UX work is not a cost, it is an investment
- Less repeated work by templates, guidelines, heuristics, metrics…
- Systematic and early work decreases development and maintenance costs (rework, refactoring, maintaining unneeded features etc.)
- Systematic and early UX work increases product quality
UX Work

- Is iterative
- Emphasizes early phases
- Is cooperative
- Quality over quantity
- Aims at lean process & outcome
- less wasteful features, less expensive wasteful work
- Aims at happy people :)

A Commonly Recommended Agile UX Model

Cycle 0
Plan and study

Cycle 1
Implement high dev cost low UI cost features
Design for cycle 2
Study for cycle 3

Cycle 2
Implement designs
Test cycle 1 code
Design for cycle 3
Study for cycle 4

Cycle 3
Implement designs
Test cycle 2 code
Design for cycle 4
Study for cycle 5

D. Sy 2007
U-Scrum – Involving a UX-PO
(Adapted from M. Singh 2008)
Scrum Process with UX Tasks

Vision V, Evaluation E, Concept creation C, Design and Development D&D, and Shipping S
Common Agile UX Practices

- Little/Some Design Up Front, LDUF (inherently an agile practice)
- Close collaboration
- Prototyping (particularly lowFi)
  - at early phases, also as a communication tool
- User stories
- User testing
- Inspection evaluation / usability inspection
- Refining the UI for the next iteration
- UX one or two sprints ahead of development
- Big picture of the project needs to be maintained
- Scenarios & Personas are commonly in use
Recommended Practices

- Cooperation is a key issue
  - ”All together from early on” (J. Coplien)
    - UX, architects, developers, customers, business…
  - Emphasizing early work, it is affordable
- Parallel design with short feedback cycles
  - Best way to have a good idea is to have lots of ideas, don’t fix with one idea too early
- UX team should be agile too, e.g. UX sprints
  - However, some upfront design is needed
  - Good UX comes from the process and cooperation. UX team should lead the UX work, not serve development on order-basis
- Keeping the big picture in mind is essential
General Advice

- Heavier studies conducted outside project work
  - Generalize & reuse
- Define & understand 'good user experience'
  - Easy to use, fun, efficient, aesthetic, convincing, reliable…
- Fail and learn quickly, waste cheap time
- Cooperate! Business, UX, architects, developers…
- Early and continuous feedback
  - From users, developers, customers, other stakeholders
- Many methods and ways of working may lead to excellent outcome, no one single right way exists
Contents

1. Introduction / Motivation
2. Agile software development
3. User experience work as integral part of SW development
4. How is it done in practice – experiences from numerous development projects
Some Tasks UX Specialists commonly Participate

- **Design**: determining HOW the system should work
  - not what it should look like (graphical design)
- Sales, budgeting / pricing
- Feasibility studies, roadmap planning
- Design upfront (planning, doing user studies and design work before implementation is started)
- Backlog and feature creation, clarifying user requirements
- Planning, conducting and analyzing user studies and tests
- Designing the user interaction or user flow (the core of UX designer’s work)
- Doing graphic design for the user interface
- Implementing the user interface
- Reviewing user interface designs or implementation
Communicating UX Design for Development

- High fidelity or photorealistic images of UI screens (such as PhotoShop)
- Wireframe images with explanation texts (interaction and navigation presented with e.g. text and arrows)
- References to style guides (guidance for selecting UI components and styles)
- Paper prototypes (physical low-fidelity models. E.g. cardboard, post-it)
- Low-fidelity software prototypes (sketches of screens, mockups, storyboards. E.g. Balsamiq)
- Mid-fidelity software prototypes (fairly detailed but simple and approximate model with simulated functionality. E.g. PowerPoint, MS Visio, Axure RP, Omnigraffle)
- High-fidelity software prototypes (detailed graphics with some actual functionality (often simulated back-end), such as AppSketcher, FluidUI, Adobe Flash Catalyst)
- Working software, source code (such as HTML and CSS)
Research Example

Multiple case study of three mobile development projects supervised by a single product owner in a company

Project team locations

Project 1
- Finland 1
  - PO, UXS
- China
  - 5 Devs, SM

Project 2
- Finland 1
  - PO
- Finland 2
  - 5 Devs, UXS
- China
  - 2 Devs

Project 3
- Finland 2
  - 3 Devs
- Finland 1
  - PO
- Latvia
  - UXS
Setting in project 1

- UX on product level
  - Early work
  - Has power of decision
- Learning between PO and UX-PO
  - PO able to do UX tasks
  - UX able to do PO tasks
- High overhead costs
  - Two POs coordinating a distributed team
- Whom to contact, PO or UX-PO
Setting in project 2

- Shared UX tasks
  - User studies (PO)
  - Visual design (UX)
  - Platform style (Developer)

- Everyone can be a UX person
  - Shared ownership
  - Everybody participated to ideation
  - Developers (WP and Android) had face-to-face communication with UX person
    - Easy to get immediate answer

- UX vision should be quite clear when using this approach
Setting in project 3

- Shared UX tasks
  - UX vision, user studies (PO)
  - Visual design and HTML style (UX)
  - Platform style (Developer)

- Design task allocation between developer and UX designer
- UX designer role needs to be stronger in distributed project
  - Ensuring UX implementation quality

- Higher cost of communication due to distribution of the team
Summary

- UX work should be included in software development practices
- Some design upfront (DUF) is usually needed
  - Studying and concepting before starting development
- UX, business, and technical goals should be aligned
- Agile principles should be applied on UX work also: short feedback cycles (iterations), constant communication, limited DUF etc.