2.0 workflow
merging agile and UCD processes

Presented by Kelly Goto
kelly@gotomedia.com

A BIG BALL OF MUD is haphazardly structured, sprawling, sloppy, duct-tape and bailing wire, spaghetti code jungle. We've all seen them. These systems show unmistakable signs of unregulated growth, and repeated, expedient repair ...

...Programmers with a shred of architectural sensibility shun these quagmires.

Brian Foote / Joseph Yoder - PLoP ’97
Forces Creating the Big Ball of Mud

- Time
- Cost
- Experience
- Skill
- Visibility
- Complexity
- Change
- Scale

The Cure

a flexible, adaptive, feedback-driven development process in which design and refactoring* pervade the lifecycle of each artifact, component, and framework, within and beyond the applications that incubate them.

Refactoring: “Cleaning up” without affecting the functionality
Rich App/2.0 Web Development

Who Are You?
(and why are you here at the last session of the last day?)
Some areas to address:

• ‘Agile’ sounds great in theory - how to truly implement?
• Want to be more ‘user-centered’ but lack time, budget and expertise
• Too many project to manage at once - no ability to prioritize or scope properly
• Inconsistent processes (varies each project)
• No centralized communications, lack of templates and overall process
• We don’t have the right team or resources in place to make this work

A Bit of Background

Agile Manifesto

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others 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.
Agile Manifesto

Principles behind the Agile Manifesto

We follow these principles:
- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

UCD (User Centered Design)

“User Centred Design (UCD) is an approach that supports the entire development process with user-centred activities, in order to create applications which are easy to use and are of added value to the intended users...”

- usabilitynet.com

UCD (User Centered Design)

There are four important UCD principles:

- A clear understanding of user and task requirements
- Incorporating user feedback to refine requirements and design
- Active involvement of user to evaluate designs
- Integrating user centred design with other development activities
non-integrated process

traditional
- functional requirements (business needs)
- use cases (assumed tasks)
- functional specifications (assumed tasks)
- prototype (testing)

UCD
- contextual observation (needs analysis)
- contextual interviews (concept feedback)
- assessment testing (usability testing)
- verification testing (usability testing)

integrated process

Agile + UCD
- concept (what can you build in a sprint)
- build (start with a single feature)
- test & iterate (code, test, iterate)
- release (assess & plan for next sprint)

iterative, dynamic process

Cycle Planning
Structure/Content
Scope Planning
User Needs
Testing/Refinement
Design/Coding
Release
Flavors of Agile

**PREDICTIVE**
- Disciplined
- Planned

**ADAPTIVE**
- Agile
- Iterative

Waterfall Methodology
Traditional capture, analysis, design, coding and testing in strict sequence. First published in the 1970's.
Cowboy Coding

Process Spectrum

What we’ve learned

• Focus less on documentation and more on communication
• Break it down - iteration is key
• Change is part of the process
Merging Agile + UCD

Agile + UCD Basics
Determine Gap & End Goal

Design Process Diagram

Determine Gap & End Goal

Current & Future State Analysis

1. Have objectives been achieved?
2. How successful was your customer's idea?
3. Are the requirements met on the design/strategy plan?
4. How well did you meet your customers' needs?
5. Is your site meeting your customer's needs this time next year?
6. What are your marketing objectives for the next year?
Set Release Cycles

- Fast, iterative cycles during design process

Establish realistic cycle duration
- 3 - 4 weeks; 3 - 4 months

Release Cycles

- Concept
- Iteration 1
- Iteration 2
- Iteration 3
- Iteration 4
- Iteration 5
- MONTH 1
- MONTH
- MONTH 3
- MONTH 4

Cycle / Sprint

Determine “Sprint” Cycle for each iteration something manageable that can be agreed to by team and client.

- Web Sites = Release Quarterly
- Applications/Widgets = Release every month
Set Clear Milestones

Projects need clear milestones during project lifecycle that all members, client and team alike, can agree to.

AVOID BLACK BOX SYNDROME!

Establish Reporting Frequency

Teams need to be self-running (within reason) but each member needs to report in (recommended) every 24-hours with their progress.

Short Cycle/Sprint = Frequent Reporting

Structured Teams
Team Dynamics

• Non-traditional Project Management
• Less focused on documentation - more on communication
• Manager/Lead is part of the team (sub-managers)
• Team is self-governing within stated guidelines
• Team has a system of self-check points and milestones

Challenges

• Small, focused teams per ‘sprint’ require more resources & depend on specific areas of expertise (i.e. action script coding)
• Agile projects operate under compressed and often unrealistic timeframes, with little time for usability testing or feedback cycles
• Full project scope is often difficult to assess until further in the project lifecycle (budget and timeframes are sometimes finite)
**Approach** (process)
- Project goals (complexity & definition)
- Team (skillset & platform)
- Process (daily flow, communication)
- Release cycle (break it down)

**Approach** (audience)
- Who are they? (contextual studies, online surveys)
- What do they do? (usability studies, interviews, observation)
- What do they need? (iterative feedback, test early & often)
- Is this working? (keep it simple!)

**Output** (initial deliverables)
- Features list (ranked by task, in order of priority & release)
- Project spec (wiki-style, dynamic & communicative)
- Wireframes & flows (starting with main feature/task)
- Design & prototype! (low to high fidelity)

---

**Emotional vs. Practical**

- **Qualitative** lifestyle research
  - Field Studies
  - In Depth Interviews
  - Usability Testing
  - Card Sorting
  - Intercept Surveys
  - Stats/Tracking

- **Quantitative** statistical research

---

**Conduct Audience Research**
Conduct Audience Research

<table>
<thead>
<tr>
<th>Why</th>
<th>Understand context, lifestyle and usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>How</td>
<td>Take several hours to a day to observe and brainstorm with participant</td>
</tr>
<tr>
<td>When</td>
<td>During conceptual phase, or during a refinement stage</td>
</tr>
<tr>
<td>Reality</td>
<td>Time consuming and expensive, lack of internal expertise</td>
</tr>
<tr>
<td>Result</td>
<td>Helps to create ‘stories’ and ‘use cases’</td>
</tr>
</tbody>
</table>

Contextual Inquiry

Persona Development

Joseph Kidman

Quick Priorities

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>Priority Level</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Key Traits

- Needs a stable, reliable connection
- Prefers text over images
- Values privacy and security
- Requires responsive support
- Wears glasses
- Is a parent

WANTS TO BE...

- A social media expert
- A data scientist
- A product designer

Osaka, Japan

Age: 30 years

Job: Software Systems Engineer

Skills: Python, Java, SQL, Django

Personal Interests: Travel, Music, Reading

Profile:

Joseph Kidman

A young professional with a passion for technology and design

Adept at creating solutions that meet user needs and exceed expectations.

Quick Priorities

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

WANTS TO BE...

- A data scientist
- A product designer
- A social media expert
Audience Segmentation Grid

Create Task-Oriented IA

<table>
<thead>
<tr>
<th>Why</th>
<th>Traditional IA in software focuses on process, not people</th>
</tr>
</thead>
<tbody>
<tr>
<td>How</td>
<td>Create diagrams based on tasks and mental model for each user type</td>
</tr>
<tr>
<td>When</td>
<td>During IA process - before specifications</td>
</tr>
<tr>
<td>Reality</td>
<td>Some may view this as an unnecessary step</td>
</tr>
<tr>
<td>Result</td>
<td>Creates an alternate 'real' view of tasks on a product or site</td>
</tr>
</tbody>
</table>
Conduct Needs Analysis

Why
Find out what people really want/desire

How
Informal sessions with sticky notes-concurrent with in-person interview

When
As early in the process as possible

Reality
Difficult to find time/budget in the beginning, usually work with assumptions

Result
Cuts down on feature creep and focuses on the target audience’s real needs
Needs Analysis

Card Sorting

Affinity Diagram
Prioritize Features

**Why**
Effective means of locking down features for 'sprint' or launch

**How**
Evaluate features by user need, business value and technical feasibility

**When**
Prior to initial sprint/release

**Reality**
Features are often added with no sense of priority and end user need.

**Result**
Helps to control feature creep and project scope

Prioritize Features

---

Prioritize Features

---

Prioritize Features

---
Prioritize Features

<table>
<thead>
<tr>
<th>Desired Functionality</th>
<th>Customer Need</th>
<th>Business Value</th>
<th>Technical Feasibility</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario: Feature</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenario: Feature</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenario: Feature</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenario: Feature</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenario: Feature</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scenario: Feature</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

MyCompanyX - Requirements for Changes to Planned MyCompanyX Product Launch

Agile + UCD Case Study
seiko epson: UI & testing

Competitive Research