Coherent agile user-centered design
Big picture design, 2 weeks at a time

Desirée Sy
Media + Entertainment
Autodesk

What’s this about?
Examples of:
• Big Picture without Big Design
• Agile multi-sprint designs
What’s this about?

- Can’t see the big picture
- Unsure what to do with new backlog items
- Want to reduce unanticipated design re-factoring in later iterations
- Want to know when a design is “done”
- How to add user requirements to “done”

What’s this about?

Not today:
- Integrate basic UX activities in agile
- How to
Highly unscientific audience survey

Agile UX: the good

- Narrows the gap between finding and fixing issues
- Less “design drift”
- Enables requirements iteration
- Most important features are done first
- Contextual inquiry & usability testing on actual product
- User data has effect on current release
- Less wasted UX time
- Face-to-face is better than “over the wall”
Context

Alias (now Autodesk)
- products and users
- UX practices
- agile practices

You?

- Commercial software
- Website design
- Web applications

- Consultants
- Non-profit
- Government
- Internal IT
Agile looks more like a culture than a process
- Jeff Patton

Agile is about better project control, not speed
- Chris Nodder
Agile definitions

- Scrum coach/Scrum master
- Product Owner/Customer
- Retrospectives


Big picture vs. big design

Waterfall
Plan everything before coding starts

XP
Plan nothing before coding starts
Big picture before agile
Big picture before agile

Missing requirements

Time between the Big Picture thinking and coding meant we missed requirements:

- Change in the market
- Change in experience through use of product
Design issues

Trying to capture all requirements at detailed level for the whole product meant:

- A lot of missing detailed requirements
- A lot of unused design
- A lot of “overdesign”
- Long lapse between time of design and coding

Fix design issues
Big design

No big design, no big picture

UX
USER EXPERIENCE

Autodesk

Copyright © 2009, Autodesk. All rights reserved.
Fix design issues without Big Design

Levels of detail
Levels of detail

- Sprint
- Capability
- Release
- Product

Example

- SketchBook Pro
- Tablet PC + Wacom tablets
Set design goals

Design goals

- Applied to backlog, let you:
  - discard
  - sort
  - rank
- Focus UX investigations
- As requirements, they can define “done”
Product goals

• Product vision
• Who it’s for (and not)
• What it is (and isn’t)
• Design Principles
  • Define product characteristics to drive design decisions

In action...

• Product vision
  • For creative professionals
  • Sketching: responsive, light weight
• Drop: Image processing features
• Rank: Brushes, make it faster, increase flow
• Sort: Brush quality, Interoperability
In action...

- **Design Principles**
  - Elegant simplicity, Stylus-friendly, Self revealing, Maximum work area
  - Drop/Rank: Don’t add because we can
  - Design “Done”: All features must have access without a keyboard
  - Investigate: Discoverability, Clutter

---

In action...

- **Design and Engineering Principles**
  - Self revealing/Optimize (fast + small code)
  - Design had large set of icons, but that added to code weight. We redesigned.

- **Business Principles**
  - Enter broader market
  - We needed to add design and code for trial version
In action...

- TiVo Design Principle
- It’s entertainment, stupid
- “Lean back, not forward”
- Drop/rank: No keyboard entry
- “How (Not) to Destroy a Great User Experience”  UPA 2006
  Rich Fulcher, Rachel Garb, Alex Liston, Donna Slote

Release goals

- Aligns the team trajectory
- Guidance for course corrections
- Needs to be consensus between development, business, and design
- What the Product Owner uses
- Not the backlog (or a subset of it)
In action

• SketchBook Pro v2
  • “Remove barriers to purchase from trial”

• Investigate: Survey. Focused the ‘who’

• Drop/Rank: 200 > 25 > 10 (top 5)

• Drop: Saving colours

• Consensus: Dropped Rotate Canvas

In action

• Rare but powerful: Redefine the release

• SketchBook Pro v1.1
  • “Mac OS X port”

• Reset the alignment

• Promote: Add keyboard shortcuts
Capability/Sprint goals

- Articulate problems to solve for a workflow/user story
- Carry forward as sprint goals
- Defined through chunked research
- Used to chunk designs
- Used to chunk mini-releases
- Used to define “done”

In action

Brush Resize
In action

Brush Resize

- First 5 minutes: learn without documents
- Resizing without Brush Editor
- One control for size, not 2-5
- Keep focus in-canvas
- Fewer dialogues (covering the work)
- Stylus only (no keyboard)

Multi-sprint designs
Tear and build

- Design a capability over >1 cycle.
- Break a design into chunks.
- Mix and match chunks in investigations: mini-research, usability test and iterate on mini-prototype.
- Look at the design at the Capability level. Now break it into mini-specifications, to be coded over >1 sprint.

Caveats

- You’ll need to establish a buffer first
- Even with the buffer, you’ll still need to write designs for the next cycle
- Team will see it as Big Design if >1 = too many sprints
- Think about constantly revealing new capabilities
Design chunking

• What can you investigate over next few sprints?
• Look at the list of capability goals.
• How can you layer sprint-sized investigations and prototypes to meet goals?

In action

Brush Resize design goals:
• First 5 minutes: learn without documents
• Resizing without Brush Editor
• One control for size, not 2-5
• Keep focus in-canvas
• Fewer dialogues (covering the work)
• Stylus only (no keyboard)
In action

Brush Resize design chunks:
• Brush Resize with hotkey
• Brush Resize with stylus (interaction)
• Brush Resize with stylus (look)
• “Workflow” prototype

In action

Brush Resize with hotkey:
Disposable code prototypes

• Resizing without Brush Editor
• One control for size, not 2-5
• Keep focus in-canvas
• Fewer dialogues (covering the work)
In action

- Brush Resize with stylus (interaction):
  Whiteboard prototype

- Stylus only (no keyboard)
In action

- Workflow prototype: Disposable coded prototype
- First 5 minutes: learn without documents
- Combined with 2 other user stories: Brush Palette and Custom Brushes

Specification chunking

- Which users will see the next cuts and when?
- Look at the list of the capability goals.
- How can you layer the design so that each sprint delivers on key goals?
- Think worst-case scenario: is each incremental build shippable?
In action

Brush Resize implementation:
• Per-brush Property Editor, with Size control
• Brush Resize widget

In action

Per-brush Properties dialog with Size control:
• Overlap with Custom Brushes
• First 5 minutes: learn without documents
• Resizing without Brush Editor
• One control for size, not 2-5
• Stylus only (no keyboard)
In action

Brush Resize widget:

- Keep focus in-canvas
- Fewer dialogues (covering the work)

Usability acceptance criteria

- Sit with developers as they are turning the user stories into hard estimates.
- Make sure you understand how each of their pieces builds into the specification.
- If applicable, add code-testable usability criteria
In action

Heads up display

• Fixed decimal point

Working with the team
Course corrections

What to do at:

• “Scrum”/stand up meetings
• Daily contact with developers
• Capability discussions with developers
• Sprint/Release retrospectives
• Sprint Zero meetings

Putting it together
Coherent agile UCD

Requirements are still required:

- Product and release goals give you a big picture, and then inform capability goals
- Use capability goals to break multi-sprint designs into smaller pieces, both to design and implement
- Work closely with development

Thank you

desiree.sy@autodesk.com
@DesireeSy
#upa2009 #agileUX