I’m here to talk about the role of usability in the Agile development process
I was the interaction designer and user experience person for a project to convert paper to electronic forms.
We used an iterative Agile development process for this project.
Project overview

- Banking Center (branch) employees fill out paper forms.
- Current paper-based process is expensive and wasteful, decides to use electronic forms
- Eforms platform chosen (Adobe LiveCycle)
- Agile development recommended by vendor
- Integrating user experience practices within Agile framework
- Build client trust
Project Goals

• Replace paper forms.
• Banking center Client: Intuitive system, easy to use, no written documentation necessary
• Enterprise Client: Solution portable to other domains
• As automated as possible (not always possible: i.e., hardware Signature pads not available)
• Fit well in existing workflows and systems
• Why do we want to do this? Saves money!
UX, meet Agile development

- How many people are familiar with Agile Development?
- We used a modified version of scrum
  - Daily standup
  - Weekly show-and-tell (to customer)
  - Timeboxed iterations/sprints
  - Planning poker
  - Burn-down charts
  - Story and task cards.
    - Story cards indicate a feature: Ex: “As a teller, I need to search for a customer to locate the correct customer record”
    - Task cards break out the steps needed to implement that feature, with time estimates.
    - I made sure to make task cards for UX activities, not just development/coding.
Typical agile iterative process

- Iteration 0 includes “spikes”—time-limited research development
- Client aka “Product Owner” involvement in each iteration
- Features developed and released at each iteration
Work ahead, follow behind method

- You can’t just do all the UI design up front and then leave, consultant-style.
- Ongoing process, interacting with developers
- Need to work ahead of development team
Work ahead, follow behind method

UX

Planning
User Research

Iteration 0

• Let’s step through what we did at each stage in the process:
• Iteration 0:
  • Review existing paper forms
  • User research activities
  • Initial designs
  • Business Analyst gathers requirements and documents process flows.
Paper has physical costs

- Raw material
- Printing
- Transport and stockpiling

We knew that paper forms cost the company a lot
Project was approved by executive council based on the cost savings for the physical items alone: Materials, Printing, transport, and stocking costs.
Paper has process costs

- Storage and retention
- Logistics for approval routing

- This is a “Forms tree”: It shows a form and its dependencies
- All sorts of regulatory requirements around retention and destruction of documents. Can’t be automated in paper format.
- Even the same form can have various levels of approvals based on amount. Even routing to multiple departments. Again, no automation possible in physical form.
- Since this was not considered in the project finances, it is a bonus savings
Paper costs in User Experience

• Delays due to errors
  — Wrong form filled out
  — Incorrect routing

• Delayed approvals waiting for people out of office

• Paper costs in terms of the customer experience as well.
  • Paper is slow
  • Paper is error-prone
  • Re-work
  • Miss-filed
Why so many mistakes?

What's so hard about this?
Why so many mistakes?
It’s complicated!

• This is a “swimlane” flow chart for one of the more complex forms, originally developed by the Business Analyst.
• For some forms, users have to gather information from multiple sources and systems in order to complete a single form.
• We also want to know what the user is really doing during this process. How do they work on the forms?
Involving actual users

- Not a core value at the bank. Very siloed, hierarchical, and rules-driven.

- Client survey ≠ user testing

- Understandably don’t like to have UX people hanging around the money

• Not a core value at the bank. Very siloed, hierarchical, and rules-driven.
  • Client management wrote and approved the procedures manuals for the forms
  • So, if users aren’t filling out the forms correctly, it must be a training problem. Why don’t they just read the procedures manual?
  • Client survey ≠ user testing
    • When I joined project, I was told there had been user testing.
    • Turned out users had been asked what features they wanted in a forms system. Gets you a laundry list of wants.
  • I asked about a branch visit. Understandably, they didn’t like the idea of having UX people hanging around the money
Since we couldn’t go to the mountain, we brought the mountain to us.

• Arranged to have a small sample of typical users come in to be interviewed and demo how they work
• We set up a conference room with live functioning terminals that they use to do their work on (mostly) paper forms now.
• When actual users demoed how they currently use forms, they weren’t always following procedures manual perfectly (gasp)
• Errors that aren’t considered errors: When interviewed, users initially said that errors on a particularly difficult form were rare. But they indicated that the back office often sent the form back for rework (that is to say, to correct errors). But this was so common that they didn’t consider it an error. It was just part of the routine.
• There is a tradition in the bank of doing a “Walk-A-Mile” exercise; that is, essentially shadowing someone while they do their job. This is used for cross-training, and also for direct training of backups.
• So, after we built some trust with the client, we leveraged this tradition to eventually score a visit to an actual branch (banking center). With the client along as adult supervision, of course.
  • (Sadly, it turns out Comerica Park is not a banking center.)
• A situation came up (not a robbery!) that necessitated calling the police. Ate an hour of the user’s time before the user (a branch manager) could return to her task.
  • So, Interruptions were common = we needed better status messaging
• We used the information we gathered from users to develop Personas, which are portraits of typical users and how they will be interacting with the system.
• We used this to help us make decisions about how the eforms product should work for users.
Work ahead, follow behind method

Iteration 0  Iteration 1

• Iteration 1
  • UX works on high-fidelity simulation of user interface, to be implemented in future iteration
  • Design built working closely with client/stakeholder representatives
  • Developers concentrate on tasks that have high complexity, but little or no User Interface elements
Simplified UX Agile Process

- Design UI elements for story
- Validate design with customer
- Frequent communication with developers
- Test developed feature
- Present developed feature to customer

- Design User Interface elements for story (often collaboratively with client)
- Validate design with client (iRise or wireframe)
- Communicate to/with developer (Not thrown over wall)
- Test developed feature
- Present developed feature to customer
Early interaction design: Hi-Fi

• **iRise simulator:**
  Almost looks like the real thing.

• **Meanwhile, Developers working on back-end search functionality**

- High-fidelity comps early on
  - iRise prototyping tool: I can’t believe it’s not eForms.
  - Provides deeper UI guidance for developers
  - While UX is building this, developers working on back-end integration for search etc.
Early interaction design: Hi-Fi

• Something to show-and-tell to product owners at early stages

• Buttons work, walk stakeholders through interactions

• Builds trust

• High(er)-fidelity comps early on
  • Something to show-and-tell to product owners and stakeholders at early stages
  • Stakeholders can see interface in action, be shown interaction as it progresses, clicks work, etc.
  • Builds trust, gets buy-in with clients stakeholders
Actual LiveCycle form

• Implemented code

• This is how the search looks in the actual LiveCycle environment, after it was developed in later iterations.
• Search is embedded in the top section of each form, where the customer information is eventually displayed. So the form name “CD Payment Maintenance” appears above the search area.
- iRise simulator on the left.
- Actual LiveCycle environment as implemented is on the right.
- The search area is replaced by the selected customer’s information in the top section of the form.
Work ahead, follow behind method

- Iteration 2
  - Validate search development from Iteration 1
  - Communicate designs with developers for iteration 2
  - Start designing for iteration 3
Later interaction design: Lo-Fi

- Developers have internalized design patterns
- Omnigraffle (or visio) with hand-drawn-look templates
- Developers have picked up speed, simulations take too long to build
- Developers have actual working code and forms to demo for customers

- Decreasing fidelity as development internalizes design patterns
- Hand-drawn look to help clients feel this is work-in-progress and not final
- Have already built client trust with earlier high-fidelity simulations, so they don’t need to see detailed comps at this stage
- Quicker design process to keep developers fed with fresh designs
- Show-and-tells are now using actual LiveCycle forms anyway
Work ahead, follow behind method

- Planning
- User Research
- Design for Iteration 2
- Develop high-complexity, low UI stories
- Implement designs
- Design for iteration 3, Validate iteration 1
- Implement designs
- Design for iteration 4, Validate iteration 2

• Iteration 3 and beyond: The pattern continues:
  • Design in low-fidelity and research for upcoming iterations
  • Validate previous iteration’s development
  • Clarify and communicate with developers regarding current iteration development
Pitfalls
Pitfalls

• Bandwidth (keeping the developers fed)
• Work ahead means you aren’t on the same page at the same time with developers
• UI cohesiveness
• Easy to miss later feature dependencies

• Bandwidth (keeping the developers fed with enough designs)
  • Getting ahead can be tough, unless you do a lot in Iteration 0
  • Not being on the same page as developers means you have to switch mental gears. Not without some grinding.
  • Slicing and dicing features into stories and tasks can make it tough to keep a cohesive UI design throughout.
  • Easy to miss later feature dependencies
    • i.e., multi-account selection in search, which didn’t come up until later
Pitfalls

- Ideally agile should be run with expert developers in a familiar tool.
- Rest of organization in waterfall mode

- We had a brand new tool, and hence no expertise.
- Rest of organization in waterfall mode
  - Infrastructure couldn’t grasp that we needed development, QA, and production servers all at once.
  - Wanted full requirements before giving us access to databases and resources
Success Factors
Success factors

- Very accessible client/product owner
- Dedicated team
- Team workroom
Success factors

• Experienced Business Analyst

• Experienced developers

• Good connections to other parts of IS

• Experienced Business Analyst worked with customers to get requirements ironed out
• We had experienced developers which really helped with integration into bank systems
• Good relationships with people in other parts of organization helped us get what we needed to make this work
• It was personally very satisfying to work on a project whose goals were both good for business and good for the environment.
• These are some of our experiences integrating User Experience practices with Agile.
• I hope you found some of it useful.
• I'd like to hear about your experiences with UX and Agile!
Thanks

John C. Rivard
jcr@jcrdesign.com
Twitter: @jcr

• Please feel free to contact me!