

A large, semi-transparent watermark of the UCI logo is centered in the background. It features a stylized orange and blue crest with the letters 'UCI' in white.

Agile for a campuswide IT pilot project

UCCSC 2016

Kelsey Layos

Manager, Academic Web Technologies

Office of Information Technology

UC Irvine

Introduction

- Kelsey Layos, Manager: Academic Web Technologies, OIT, UCI
- We build web apps to support teaching and learning, including a homegrown learning management system started in 1996
- We use Agile project management methodologies (specifically Scrum) to get the best results we can in all we do
- Presented on our team's process back at UCCSC 2013

What's Agile?

“Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end.”

- Philosophy
- Focus on results, people
- Flexible processes; various practical implementations (we use Scrum)
- Creates **feedback loops** to facilitate change/evolution



UCI news items fit my theme nicely ;)



Creation of a sand mandala at UCI for the Dalai Lama's 80th birthday



[UCI & SIOC] Chemists find new ways to recycle plastic waste into fuel



Massive northeast Greenland glacier is rapidly melting, UCI-led team finds

Canvas Pilot

- Big
- Complex
- Novel
- 40,000 users
- Resistance to change
("if it ain't broke; don't fix it!")
- Changing landscape
 - High cost and risk to develop and host local system
 - Limited flexibility, adaptability to emerging needs & opportunities
- Costly and with potentially long-term impacts
- Great uncertainty about the future



The problem with pilots

- Asking users for time, work, and patience... for uncertain payoff
- Limited information, confidence
- Standing on moving ground
- **Basically: Just tons and tons of uncertainty**

The point of a pilot

A pilot is a real-world experiment

- No designer has perfect data
- No design is ever perfect
- In a pilot, the destination isn't clear;
How do you design a path to a destination you haven't pinned down?
- A pilot is a discovery process for critical data to inform next steps



Feedback loops

Think natural selection or scientific experimentation:

- Gotta do the thing to learn from the thing
 - Piloting service = testing hypothesis (“this app will benefit users”)
 - User feedback = experiment results
 - Fewer variables = more confidence in interpreting data
- Do the smallest / lowest cost thing to learn early and often

Don't just build to build; **build to learn**

Icebergs?



- How we conceptualize our project planning around the iceberg idea
- We spend the **most time and energy on the tip of the iceberg:** parts we see most clearly
- We **plan minimally for the deeper parts;** we plan for what we know and wait for better data if we can
- One way we refer to this is as “last responsible moment” decision making

Waterfall v. Agile

Waterfall	Agile
<ol style="list-style-type: none">1. Research what instructors need2. Design start-to-finish3. Build4. Test5. Launch6. Monitor	<ol style="list-style-type: none">1. Research what instructors need2. Identify Minimum Viable Product3. Design -and- build -and- test iteratively4. Launch5. Monitor
<p>Got it totally wrong?</p> <ol style="list-style-type: none">1. Live with it2. Start over3. Get the duct tape out, make it work, accept limitations/consequences	<p>Got it totally wrong?</p> <p>Since we only built MVP, more freedom to change or even start over with minimal loss</p>

Three Case Studies

- GrandCentral - Iterative development in response to user experience
- Turnitin.com - Flexibility in th terms of our experiement
- Grade Submission Tool - “Last responsible moment” decision making

Case Study: GrandCentral

The project:

- Utility for managing Canvas course spaces
- Gathered data for planning in “soft pilot” while provisioning on users’ behalf
- Built out in self-service provisioning ahead of Fall 2015

Case Study: GrandCentral

What happened:

- After Fall 2015 launch, solicited lots of feedback
- Monitored support
- Saw recurring issues, confusion around certain functions
- Completed major rewrite for Winter 2016 in response to user experiences

Case Study: GrandCentral

Why Agile mattered:

- MVP approach helped us avoid digging a hole we couldn't code out of
- Agile philosophy reminded us to listen to our users
- Rapid development cycle enabled us to prioritize work and redeploy for Fall

Case Study: Turnitin.com

The project:

- There wasn't one!
- We had no intention of including third party LTI integrations in the pilot

Case Study: Turnitin.com

What happened:

- People kept asking for us to add it
- We realized we were missing important data if we didn't
- We added it
- We saw significant confusion, issues with the way the integration works
- We learned to expect unpredictable quality with third party tool integrations
- We changed our language around third party integrations

Case Study: Turnitin.com

Why Agile mattered:

- Philosophically primed us to 1) listen to users and 2) change our minds

Case Study: Grade submission tool

The project:

- Transfer process to send final grades from Canvas to Registrar
- Highly effective function of homegrown system
- Had to be solid by Spring 2015 grade submission
- Critical to instructors and Registrar getting their work done
- Had to balance against many other priorities

Case Study: Grade submission tool

What happened:

- Devised low, medium, and high value approaches
- Planned for high value
- Identified 'point of no return' date
- Ordered our activities such that we started with the parts of the high value project that could be reused if we had to revert to the medium or low value
- Successfully completed high value -and- made iterative improvements

Case Study: Grade submission tool

Why Agile mattered

- Historic velocity data helped us estimate delivery timeline with confidence so we did not have to err on the side of starting excessively early (and accepting attendant opportunity cost in the form of other work we could not do ahead of the quarter as a result)
- MVP approach controlled cope and helped identify 'escape routes'
- Agile philosophy reminded us to listen to our users
- Rapid development cycle enabled us to prioritize work and redeploy for Fall

Validating this approach

- What makes us think this worked?
 - Solid pilot participation
 - Extensive assessment data
 - Clear signal from assessment data to support Canvas adoption
 - New connections/relationships across campus
 - Positive feedback

Conclusion

- Embrace *controlled* chaos & inevitability of change
- Design experiments, rather than outcomes
- Last responsible moment decision making (when you have the -most- info)



Resources

- <https://www.scrumalliance.org/>
 - <https://www.scrumalliance.org/why-scrum> - What/why Scrum? (incl. 30 sec video)
- Creation of a sand mandala at UCI for the Dalai Lama's 80th birthday
<http://bit.ly/29xNhAD>
- Massive northeast Greenland glacier is rapidly melting, UCI-led team finds
<http://bit.ly/29G2Bw6>
- [UCI & SIOC] Chemists find new ways to recycle plastic waste into fuel
<http://bit.ly/28R6x91>