



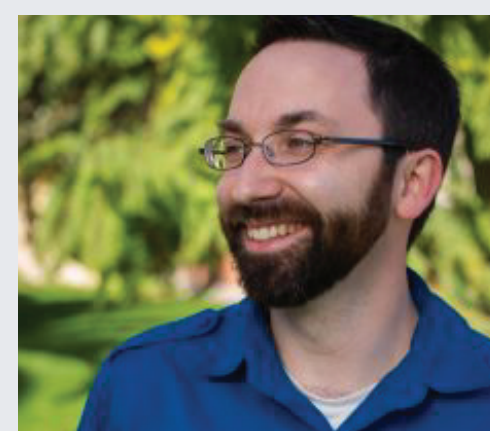
Iterating on Agile: Making It Work in Real Life

Who we are:



Rebecca Blakiston
@blakistonr
Web Design & User Experience Lead

A user experience librarian, Rebecca focuses on designing outstanding experiences for our students, faculty, and staff. She loves user research, content strategy, and questioning our assumptions. She authored two practical guides for librarians: *Usability Testing* and *Writing Effectively in Print and on the Web*. She was a *Library Journal* Mover and Shaker in 2016.



Mike Hagedon
@michaelhagedon
Software & Web Development Lead

A software engineer, Mike focuses on building outstanding products that meet the needs of our users. He loves creating elegant code and effective, transparent processes. In 2013, he won the ACRL IS Innovation Award and ALA Cutting-Edge Technology in Library Services Award for creating the open-source instructional tool now used around the world, *Guide on the Side*.

What's working for us:

Valuing the team

Individuals and interactions over processes and tools

- Daily standups for syncing our work
- Pairing for collaborative learning
- Swarming for building momentum and shared goals

Building working software

Working software over comprehensive documentation

- Simple and quick deployment for immediate user impact
- Version control for reduced stress on the developers
- A flexible code review process for increasing developer confidence
- Iterative and frequent deployment for user impact and developer sanity

Partnering

Customer collaboration over contract negotiation

- Product teams for shared ownership
- Acceptance criteria for getting on the same page
- Behavior-driven development scenarios for better decision-making
- Joint standups and Slack channels for frequent communication

Responding to change

Responding to change over following a plan

- User stories to focus at the right level, and plan just in time
- Prioritized, single-file project backlogs to allow scope-cutting and follow-through
- Flexible task status and project workflows

What didn't work:

Timeboxing two-week sprints

We failed most of our two-week sprints, mostly due to distractions and unrealistic estimates.

Creating individual lanes

Personalized, individual lanes on the Kanban board helped people track work, but discouraged collaboration.

Sizing stories

Sizing stories using story points and other techniques took a lot of time and was rarely accurate.

Reviewing all tasks

Tasks got caught up in the review column, and once we had established trust in our teams, rigid review requirements didn't always provide value.

Our teams:



Tools:

- Whiteboards, markers, magnets, and sticky notes
- Slack for asynchronous communication and goofing off
- Git for version control
- Redmine for issue tracking and documentation
- Jenkins for continuous integration
- Phing for building and deployment
- Behat and Selenium for automated testing

What we're going to try:

Improving process for receiving and responding to requests and ideas

It's not clear to our stakeholders how we prioritize our work. We shared a schedule of anticipated work for the year, but those who depend on us need more clarity and better response time.

Estimating time based on data

Over-promising is a chronic problem in web and software development, and we're not immune! We've begun tracking metrics like task time and throughput to help everyone plan better.

Implementing a flexible project management framework

We've made a project management checklist that we've found helpful this past year, but it's just a start. We'd like to build on this framework and use it to empower more project managers inside and outside our department.

Establishing meaningful impact metrics

In addition to time-related metrics, we want to demonstrate the importance of what we do. We hope to do that by attaching metrics to our library's strategic goals.