Agile Project Management and the Real World

Emily Lynema

DLF Fall 2010

November 1, 2010
Outline

• Why care about project management?
• Traditional vs. Agile
• What is Agile?
• What is Scrum?
  • Agile case study: NCSU
• Making choices
• Resources
Why care?

• You have too much to do

• NCSU Libraries
  – 6 developers
  – 33 Digital Library staff
  – >250 library staff

• Core Information Systems
  – 3 full-time developer positions
  – 18 supported applications
  – 10 in active development
What makes it harder?

• Priorities change frequently
• Requirements change frequently
• No defined business analysts
• Emergencies happen every day
• Many projects across few people
• Everyone handles full life cycle
And it keeps going….

- IT black box
  - How long?
  - When will it be ready?
  - When will you work on my stuff?
  - Are you actually doing anything?
  - What do I have to do to get something done?
Traditional Project Management
Agile Project Management
team flexible just-in-time collaborative change iterate

people transparent adapt volatile communicate
What’s the same?

• A project is still a project:
  – Vision
  – Life cycle
  – Requirements
  – Schedule
  – Team
  – Communication mechanisms
Project Life Cycle

Agile: iterative
1. Envision
2. Speculate
3. Explore
4. Adapt
5. Close
6. Repeat 3 – 5 as necessary

Traditional: waterfall
1. Initiate
2. Plan
3. Define
4. Design
5. Build
6. Test

Taken from Highsmith, James (2010). Agile project management: creating innovative products
What’s different?

• Traditional
  – Plan all in advance
  – Work-breakdown structure
  – Functional specs
  – Gantt chart
  – Status reports
  – Deliver at the end
  – Learn at the end
  – Follow the plan
  – Manage tasks

• Agile
  – Plan as you go
  – Feature-breakdown structure
  – User stories
  – Release plan
  – Story boards
  – Deliver as you go
  – Learn every iteration
  – Adapt everything
  – Manage team
What is Agile?

“Agile development is a method of building software by empowering and trusting people, acknowledging change as norm, and promoting constant feedback”

What is Agile?

“The formula for success is simple: deliver today, adapt tomorrow.”

What is Agile?

- Response to waterfall approach
- Values:
  - Individuals and interactions
  - Working software
  - Customer collaboration
  - Responding to change

Manifesto for Agile Software Development. Accessible at http://agilemanifesto.org/
Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Manifesto for Agile Software Development. Accessible at http://agilemanifesto.org/
Agile Principles

2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

Manifesto for Agile Software Development. Accessible at http://agilemanifesto.org/
Agile Principles

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

Manifesto for Agile Software Development. Accessible at http://agilemanifesto.org/
Agile Principles

4. Business people and developers must work together daily throughout the project.

Manifesto for Agile Software Development. Accessible at http://agilemanifesto.org/
Agile Practices - Managerial

• Collocate team members and customers
• Allow team members to make decisions
• Maintain quality of work life
• Use information radiators for transparency and accountability
• Daily stand-up meetings
• Regularly evaluate processes
Agile Practices - Technical

• Build automation
• Automated deployment
• Continuous integration
• Simple design
• Collective ownership
• Refactoring
• Pair programming
Project Life Cycle

**Agile: iterative**
1. Envision
2. Speculate
3. Explore
4. Adapt
5. Close
6. Repeat 3 – 5 as necessary

**Traditional: waterfall**
1. Initiate
2. Plan
3. Define
4. Design
5. Build
6. Test

Taken from Highsmith, James (2010). Agile project management: creating innovative products
Envision

• **Initiate** project
• Develop project vision, objectives, and constraints
• Create a core team
• High-level feature list
Speculate

• **Plan and Define** project
• Gather initial broad requirements
• Create initial backlog of features with user stories
• Develop iterative high-level release plan
  – Velocity + story points
  – Must be adaptable over time!
What is a user story?

Agile Planning

• 5 levels of agile planning
  – Vision
  – Roadmap (2 years)
  – Release (2 months)
  – Iteration (2 weeks)
  – Daily
Explore

- **Design, build, and test** project
- **Iteration planning**
  - Commit to user stories for iteration
  - Create and estimate technical tasks
- **Monitor progress**
  - Daily stand-ups
  - Visual taskboard
  - Burndown chart
- **Working code = committed + tested**
Adapt

• Review everything!
  – Not part of traditional model
• Customer demonstrations
  – Feedback used to plan next iteration
• Technical review
• Team performance
  – Do need to change process?
• Project status
  – Do we need to re-align release plan?
Close

• Release (maybe)
• Celebrate!
Agile Development Methodologies

- eXtreme Programming
- Crystal
- Lean Software Development
- Scrum
- Feature-Driven Development (FDD)
What is Scrum?

- Focuses on iteration management
- Roles
  - Product Owner
  - ScrumMaster
  - Team
- Artifacts
  - Product Backlog
  - Sprint Backlog
A Scrum Sprint
A Scrum Sprint

• Sprint Planning
  – Commit to certain functionality & estimate
  – Produces Sprint Backlog

• Daily Scrum
  – 15 minutes @ start of day
  – What have you done since last Scrum?
  – What will do before next Scrum?
  – What obstacles?
A Scrum Sprint

• Sprint
  – Team does the work!

• Sprint Review
  – Show off completed functionality
  – Add new requests / changes to backlog

• Sprint Retrospective
  – What went well during the Sprint?
  – What could be improved for the next?
Agile Case Study: NCSU
Agile in Libraries

• What makes agile challenging to apply in libraries?
  – Small development teams
  – Responsible for both operational support and development
  – Often *many* smaller projects to handle
  – Fewer defined project roles
Why Agile @ NCSU?

- Tackle big problems in small pieces
- Be more transparent
- Be more adaptable
- Produce tangible results quickly and frequently
What is Agile @ NCSU?

- Loosely based on Scrum
- Iterative development cycles followed by release
- Ongoing, just-in-time planning & documentation
- Collaboration with customers
  - Cross-functional teams w/IT point person
  - Developers participate
Transition

• Migrating from 6 week to 3 week cycle
• Goals
  – Focus on fewer projects at a time
  – Increase collaboration and cross-training
  – Reduce complexity of planning
  – Easier to estimate and plan velocity
  – Easier to freeze requirements & projects
  – Technology spikes
Real world

• 3 week iteration
  – Speculate: release planning prior to start of iteration
  – Explore: 3 weeks development
    • Start with sprint planning
    • Re-align as necessary
  – Adapt
    • Expose to customers during cycle
NCSU Toolbox

- Requirements: Confluence + JIRA
- Product & Sprint backlog: JIRA
- Release planning: Google docs
- Sprint planning: Google docs + JIRA
- Daily Scrum
- Sprint review: Product Team meetings
- Sprint retrospective
Speculate
Release Planning

• Ongoing
  – Each stakeholder team works with IT representatives to lay out functional priorities for upcoming releases
  – This is very flexible and changeable!

• Core IT team prioritizes several projects each cycle
Other Future Issues

- Archive legacy holdings (need to determine criteria for doing this and interface implications) **NEEDS DEFINITION**
- Manual merge of collection and article database resources (EMAT-302) **NEEDS DISCUSSION**
- Re-evaluate user interface design around holdings **NEEDS DISCUSSION**
- first open source release?
- Include other record types (sertype blank)

2.5.0

- Automatic merging of records from SerSol and Sirsi based on SerSol identifiers (see Jacqui) - exclude MARC records with 'SS' in 001 by default; evaluate whether synchronizer can handle this as a new identifier.
- Create ability to migrate manually entered data in E-Matrix when SerSol changes database codecs for existing databases.
- Re-design license data entry form based on new mockups (fewer fields). May require mapping some old fields with data in the database to new fields.

  Use SOLR for begins with work - database web service for database list & quick search integration
  Define user stories and wireframes for phase 1 access verification system with E-Matrix Committee.
  Define necessary work for ISSN-L. This will likely require re-evaluating the entire way the synchronizer works, in addition to making it possible to import >2 ISSNs per record from Sirsi, and adding 022$8 to the Sirsi synchronizer.

2.6.0

- Create Unique Titles report?
- Phase 1 of access verification system

  - select collections / journals to check
  - click button to generate report / queue of items to check
  - provide some sort of queue (maybe one queue anyone can pull from)
  - basic screen for verification (what links should be on here?)
  - button to mark something done / checked
  - link to generate email / ticket to Remedy. There is a desire to have a form that would actually categorize the issue using Remedy API (see Troy's work on the acquisitions form) if that is possible in phase 1.
  - students need to be able to log into E-Matrix to do checking???

2.7.0

- ISSN integration for OCLC patient registration log
Release Planning

• 3rd week of previous iteration
  – High level overview of upcoming projects out 3 months
  – Prioritize projects for the next 3 week iteration with core IT staff
  – Goal: no more than 2 projects at a time
  – Just in time requirements gathering this week, if necessary
<p>| | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>June 14 - July 2</td>
<td>July 5 - July 23</td>
<td>July 26 - August 13</td>
<td>August 16 - September 3</td>
<td>September 6 - September 24</td>
<td>September 27 - October 15</td>
<td>October 18 - November 5</td>
<td>November 8 - December 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>E-Matrix</td>
<td>2.3.0 testing</td>
<td>2.3.0 finish</td>
<td>2.3.0 release</td>
<td>2.4.0 - cost data association;</td>
<td>2.4.0 user testing / debugging</td>
<td>SOLR begins with; include database list services</td>
<td>exclude serial records by October 21; evaluate serial merging identifier; 2.5.0</td>
<td>Licensing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Endeca / TRLN</td>
<td>shopping cart (AC out 6/28 - 7/5)</td>
<td>finish 3.0.0;</td>
<td>work on 3.0.0; release website redesign - look and feel (2.10.0);</td>
<td>finish and release 3.0.0</td>
<td>cover image URL caching; open mail relay;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ReservesDirect</td>
<td>roll out 1.5.0 / 1.6.0</td>
<td>website re-design (re-do ncsu stylesheets)</td>
<td>RD course instance audit phase 1</td>
<td>open source release</td>
<td>audit phase 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Journals</td>
<td>1.6.1 - website redesign - looks and feel</td>
<td>small UI changes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Databases</td>
<td>website re-design - look &amp; feel</td>
<td>subject URL bug fix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Sirsi</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>OLE ILLiad</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Journey</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Classmate</td>
<td>first use of classmate</td>
<td>catalog stats form to connect to google docs?</td>
<td>ClassMate for students</td>
<td></td>
<td>Journey bug fixes - 1.4.2</td>
<td>Journey 1.4.3 - language</td>
<td></td>
<td>ClassMate 1.0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Intranet Apps</td>
<td>student jobs 0.0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Sprint Planning

• Day 1 of iteration
  – 2-hour meeting
  – Include all development team members
  – Goal: utilize stories already entered in JIRA
  – Collaboratively create and estimate tasks for all stories
  – Collaboratively assign / volunteer for tasks
Sprint Planning

• Day 2 or 3
  – Add up task estimates across projects
  – Ensure that individual developers are not over-committed
  – Scope down at project level or divide work as necessary
<table>
<thead>
<tr>
<th></th>
<th>August 16 - September 3</th>
<th>September 6 - September 24</th>
<th>Sep 27 - October 15</th>
<th>October 18 - November 5</th>
<th>November 8 - December 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>E-Matrix</strong></td>
<td>59</td>
<td>63</td>
<td>24</td>
<td>84</td>
<td>97</td>
</tr>
<tr>
<td><strong>Endoca / TRLN</strong></td>
<td>44</td>
<td>12</td>
<td>12</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td><strong>ReservesDirect</strong></td>
<td>15</td>
<td>37</td>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Journals</strong></td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td><strong>Databases</strong></td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><strong>Summon</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Sini</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>OLE</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>ILiad</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Journey</strong></td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td><strong>Classmate</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Intranet Apps</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Data &amp; Authentication</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>SFX</strong></td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><strong>EZproxy</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Jira</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Virtual Browse</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Mobile Devices</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Umault</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Shopping Cart</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Portal Services</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>OLE</strong></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

| Assigned Hours   | 63 | 55 | 35 | 164 | 0   | 63 | 35 | 16 | 0  | 16  | 0  | 0  | 0  |
| Max Hours Allowed| -3 | 5  | 8  | 6   | 16  | -3 | 5  | 8  | 6  | 16  | -5 | 24 | -10|
| Unassigned Hours | -9 | -24| -10| -10 | 0   | -9 | 24 | -10| 0  | 5   | -24| 0  | 26 | 16 | 16 |
Explore
Development

• Get it done

• Daily scrum 10 – 15 minutes
  – Identify obstacles and priorities
  – Emphasize collaboration

• Weekly review
  – How does progress look for cycle?
  – Requires estimation and work logging

• Subversion -> JIRA integration
Testing

• “the weakest link”
• Manual testing throughout iteration
  – Utilize weekly product team emails
  – Demo at regular meetings
  – Need automated testing!
• Developers and IT product managers are first line of QA
Adapt
Weekly Review

• How much progress are we making toward sprint goal?
  – Things are harder than you expect
  – New requests come in
  – Emergencies crop up
Sprint Retrospective

• Last day of iteration
  – Did we accomplish what we wanted to?
  – If not, why not?
  – What went well?
  – What would we do differently next time?
Close
Release

• Release at end of cycle if approved
  – Stakeholders may prefer to wait for new features
  – Release can be delayed if testing is not performed during cycle
  – Need automated testing!

• If release is large or complex, may need an entire cycle to test prior to release
Things we’ve learned

• Prioritization difficult for library staff.
  – Work at release level or higher
• Reduce ‘churn’ by trying to focus on fewer projects within a given cycle.
• Limit the unknown – don’t combine new projects with new technologies.
• Difficult to freeze plans for 6 weeks.
Challenges

• Many small projects to support at once
  – Not traditional for Agile practices
  – Each iteration can be a release
• Difficult to estimate story points
  – Planning hindered with estimates at task-level only
• Tough to fit it all into 3 weeks
  – Develop a rhythm of staggered release?
Challenges

• Difficult to develop requirements in time
  – True customer not collocated with team
  – Teams of librarians work slowly

• Testing
  – How and when to automate for small projects?
  – No ‘QA’ experts

• Simultaneously handle support and development
Outcomes

• Positive movement across multiple projects
  – Individual development efforts timeboxed
  – In 2010, ~32 releases across 9 projects
  – Increased user satisfaction

• Increased flexibility to adapt to changing priorities and needs
Why Choose?

• Traditional
  – Change is very expensive
  – Well-defined requirements
  – Project is familiar territory
  – Customer is difficult to communicate with

• Agile
  – Change happens frequently
  – Requirements not well-defined
  – New technology or project domain
  – Customer isn’t sure of what is desired
Agile Tools

• JIRA + Greenhopper
  – 10 users for $20 (local installation)
  – http://www.atlassian.com/software/jira/
  – http://www.atlassian.com/software/greenhopper/

• Agile Zen
  – Free for one project (hosted)
  – http://agilezen.com/

• VersionOne
  – Free for one team (hosted)
  – http://www.versionone.com/
Online

- Agile Manifesto
  http://agilemanifesto.org/
- Agile for All blog:
  http://www.agileforall.com/blog/
- Succeeding with Agile:
  http://blog.mountaingoatsoftware.com/
Books

• Agile Project Management with Scrum
  ISBN: 073561993X
• Agile Software Development with Scrum
  ISBN: 0130676349
• Managing Agile Projects
  ISBN: 0131240714
• Agile Project Management: Creating Innovative Products
  ISBN: 0321658396
Questions

• Emily Lynema
• Associate Head of Information Technology, NCSU Libraries
• emily_lynema@ncsu.edu