Agile Project Execution

CSCI 5828: Foundations of Software Engineering
Lecture 21 — 04/03/2012
Goals

• Review the material of Chapters 9-11 of the Agile text book
  • Iteration Management
  • Agile Communication Plans
  • Setting Up a Visual Workspace
Where are we?

• In our review of Agile life cycles, we have discussed
  • Agile Project Inception
    • How do we know if we have a viable project?
    • Do we have the right team?
    • Do we know what we’re supposed to build?
  • Agile Project Planning
    • What are user stories and how do we generate them?
    • How do we assign estimates and priorities to those user stories?
    • How do we create a plan and what do we do when things change?
What’s Next?

• Agile Project Execution
  • We have our goals, user stories, priorities, estimate and an overall plan
  • Now, we have to execute that plan to produce working software
  • We do this via iterations; But, how do we convert index cards to running code
    • First, we’ll need a way to do analysis that’s fast but accurate
    • Second, we’ll need solid development practices that produce good code
    • Third, we’ll need to integrate testing into the process from day 1
Agile Iteration

- An iteration is a time-boxed phase (one or two weeks) where
  - we convert our customer’s highest priority stories into working software

It is the engine that drives our project forward

© Kenneth M. Anderson, 2012
Iteration Break-Down

• Each iteration has the following breakdown
  • For each story assigned to this iteration
    • Perform analysis and design (get the work ready)
    • Implement the story (do the work)
    • Perform Testing (check the work)
  • At the end of the iteration
    • provide a demo to the customer
    • reevaluate priorities and generate new stories (if needed)
    • Plan next iteration; Note: we avoid changing stories mid-iteration
Example

• The book presents an example of an iteration that has been assigned two user stories for a website that allows customers to create construction work permits:

  • Create work permit

  • Print work permit

• We’ll draw on this example as we continue to discuss what to do on an iteration
Analysis and Design

• Different teams will need different artifacts but a general approach is the following
  
  • Take an index card and convert it into a single page of text that analyzes the feature in more depth, adds task info, and testing criteria
  
  • Perhaps create a flow-chart to understand the interactions that need to occur around the story

    • Who creates the permit? Who approves it? What can go wrong?

  • Create personas to describe the different type of users who interact with this story; use that information to update the flow chart

  • Experiment with different designs (UI layouts) by creating paper prototypes

  • Finally, define success for the story by writing acceptance tests (Use Cucumber!) (See book pages 165-170 for examples of the above steps)
Plug for User-Centered Design and OOA&D

• The book says (unfortunately)
  • “Remember, no one went to school to be taught how to do this stuff. Be creative. There is no one right way.”

• That’s not true!
  • Human-Centered Computing (HCI & CSCW & Ubiquitous Computing + ...)
    • has been developing techniques for designing interactive software systems for decades

• if you are interested in learning more about analysis and design techniques
  • Take CSCI 4839/5839: User-Centered Design
  • Take CSCI 4448/5448: Object-Oriented Analysis & Design
A&D: When do we do all this?

• The book recommends that analysis and design be done
  • by 1-2 team members plus the customer
  • on stories assigned to iteration \( n+1 \)
  • during iteration \( n \)
• As a result, you are sure to have the analysis and design information you need just when you need it
  • And you do the work close enough to when it will be implemented to cut down on the chance that too many things change to disrupt development
• How do we get started?
  • Iteration 0 is a special case that we’ll discuss in a moment
Implementation: Do the Work

• We’ll cover techniques that are useful when developing software in an iteration in subsequent lectures
  
  • Unit Testing
  
  • Refactoring
  
  • Test-Driven Development (related to Cucumber)
  
  • Continuous Integration
  
  • Configuration/Build Management
Implementation: Iteration 0

- The first iteration on an Agile project is typically not focused on user stories but on getting the team ready to run
  - Determine your version control system (git, mercurial, svn, etc.)
  - Determine your development environment (Eclipse, IntelliJ, XCode, ...)
  - Determine your automated build tools (mvn, ant, rake, make, etc.)
  - Determine your automated testing tools (JUnit, Cucumber, etc.)
  - Determine your issue tracking system (Bugzilla, FogBugz, etc.)
- Plus, you need to start performing analysis and design on
  - the stories assigned to iteration 1!
Testing: Checking the Work

- Once a story has been implemented
  - with unit test and integration tests written and tested along the way
- the work is not yet over
- You must also allow the user to perform user acceptance tests
  - That is, you need to show the implemented story to the user
  - Show them that it passes the tests they helped you to create (Cucumber)
  - And get feedback on how well it meets their expectations
    - Change requests lead to new/updated stories with updated estimates and priorities and are handled in subsequent iterations
Execution Requires Coordination

• A&D, implementation, and testing are fundamental parts of an iteration
  • you have to do each of them for each story distributed across 2 iterations

• But, how do you keep track of the state of your user stories?
  • How do you know what progress is being made?

• You have to plan the types of communication that will exist in your iteration and where/when they will occur
  • Communication via e-mail, social media, issue tracking, etc. will be constant part of daily life

• But there are other practices you can place in your normal iteration that represent team synchronization points
  • These practices help your team stay coordinated and focused
Types of Coordination/Communication

- Daily stand-up meetings (“so short, there’s no time to sit down”)
- Story planning meetings; Iteration planning meetings;
  - Showcases (demo); Mini-retrospectives

* SPM: Story planning meeting
* IPM: Iteration planning meeting
The stand-up meeting

- Held every day, at the beginning of the work day
  - Quick status reports covering
    - the work finished yesterday
      - (make sure the burn down chart is updated if you are updating it daily)
    - the work planned for today
  - any issues that need to be addressed
    - such as problems with the development environment, a problem with a tricky algorithm, a difficult UI interaction, someone is out sick, etc.
- These meetings are meant to be quick; establish a heartbeat for the project
Story Planning Meeting

• The story planning meeting is a meeting
  • that is schedule towards the beginning or middle of an iteration
  • to verify that good progress is being made on
  • the analysis and design of the stories assigned to the next iteration
• It is also a place where our iteration plan can be updated if
  • a story turns out to be too big for a single iteration, or
  • vice-versa, a story turns out to be smaller than planned
• Both of these situations, require the iteration plan to be updated
  • that is, what stories are assigned to which iterations
  • In addition, estimates and priorities for upcoming stories can be adjusted
The Showcase

• You need to schedule time to demo the software that you created
  • The book recommends scheduling an event called the showcase where
    • during iteration n, you demo the work you finished during iteration n-1
  • If you do it this way, you will be working right up to the end of an iteration to get stories completed
    • you then demo those stories during the next iteration
    • and respond to feedback in iteration n+1
  • Or, you can show your completed stories at the end of an iteration
    • if so, you need enough time to do the demo and then an iteration planning meeting afterwards
Iteration Planning Meeting (I)

• The iteration planning meeting is held towards the end of an iteration

• It is the place where you
  • review your team’s velocity
  • you review upcoming stories (as determined by priority)
  • and you plan out the next iteration

• You already had a sense for this because you were already doing analysis and design on the upcoming stories BUT

  • you have to be flexible, if something doesn’t go as planned during THIS iteration, that can affect your plans for the NEXT iteration
Iteration Planning Meeting (II)

• The IPM is also where you can discuss how things are going, in general,
  • and address any issues that may be causing problems
  • that is, issues that haven’t come up or been addressed by the daily stand-up meetings

• Update the burndown chart (if you’re updating each iteration rather than daily)
  • Discuss implications because
    • you’ll either be ahead of schedule, on schedule, or LATE

• This is the part of Agile that is both loved and hated
  • there’s no hiding the state of the project, progress is kept visible at all times
Here the team is falling behind, because the customer added new stories sometime during iteration 3 or 4.
Mini-Retrospective

- The final coordination practice to put in place is
  - holding a short meeting at the end of an iteration
  - to examine our process
    - is there anything about our life cycle that is holding the team back
    - if so, identify potential solutions, pick one, and deploy the change in the next iteration
      - agile requires active software process improvement on an iteration by iteration basis!
    - you can identify other issues as well that are related to the tools being used by the team, or for overall team issues (the number of bugs went up)
Flexibility: No One-Size-Fits-All

• Agile is not a one-size-fits-all software life cycle
  • while these meetings are recommended
    • stand-ups, SPMs, showcases, IPMs, mini-retrospectives
  • they are not required; use what works for your team
    • try things out and iterate until you find the right balance
  • that’s the essence of software process improvement
    • agility
      • the ability to do change how we do things on a weekly or biweekly basis
Your workspace

• If a team is co-located, there are things you can do to keep the progress of the project visible at all times

  • Our book calls this the visual workspace

    • It consists of

      • The inception deck (why are we doing this in the first place?)
      • The release wall (where are we?)
      • The storyboard (what are we doing right now?)
      • The velocity and burn-down wall (how fast are we going and when will we get there?)
you always know who the customer is, what they're after, and, most important, why we decided to spend money on this project in the first place.

Impressed, the executives lean closer and ask you where you are in the project. To answer that, you then direct their attention to your release wall.

The release wall is where you and the team keep track of what's been done and what's remaining. The left side of the wall shows those features that have been fully analyzed, developed, tested, and vetted by the customer (they are ready to be shipped). And the right side shows those stories still needing to be developed.

As far as what the team is working on this iteration, you draw management’s attention over to this iteration’s storyboard.

The storyboard

Not started

Done

In progress

Analysis done

Current iteration

Ready for test

State of this iteration’s user stories

Under construction

Ready for review

Blessed and seen

by our customer

Report erratum

this copy is (P2.0 printing, February 2011)

Prepared exclusively for Ken Anderson

Updated once per iteration at the end of the iteration
you always know who the customer is, what they're after, and, most important, why we decided to spend money on this project in the first place.

Impressed, the executives lean closer and ask you where you are in the project. To answer that, you then direct their attention to your release wall.

The release wall is where you and the team keep track of what's been done and what's remaining. The left side of the wall shows those features that have been fully analyzed, developed, tested, and vetted by the customer (they are ready to be shipped). And the right side shows those stories still needing to be developed.

As far as what the team is working on this iteration, you draw management's attention over to this iteration's storyboard.

The storyboard

Current iteration

State of this iteration's user stories

Not started  In progress  Ready for test  Done

Analysis done  Under construction  Ready for review  Blessed and seen by our customer

Report erratum this copy is (P2.0 printing, February 2011) Prepared exclusively for Ken Anderson
The storyboard tracks the state of this iteration's features (or what we call user stories). Features yet to be developed live on the left, while those that have been built and blessed by the customer live on the right. As a story gets more developed, it moves across the board from left to right. Only when it is fully developed, tested, and vetted by the customer does it get moved into the Done column.

Looking at their watches, they then cut to the chase and ask when you expect to be done. To answer that, you bring them over to the only two charts on your wall you haven't shown them yet—your team velocity and the project burn-down chart.

You explain that the team velocity is the closest thing you and the team have for measuring the team's level of productivity. By measuring how much the team gets done each week and using that as the basis of planning going forward, the team can accurately predict when they expect to be done. This is shown on the project burn-down chart.

The project burn-down (see above in Section 8.5, The Burn-Down Chart, on page 148) takes the team velocity and extrapolates the speed at which the team is "burning" through the customer's wish list. The project is done when the team delivers everything on the list or the project runs out of money (whichever comes first).

With the stage set, you now calmly point out what should already be obvious to everyone in the room. Halving the development team would effectively cut the team's productivity in half.
A Variant on the Story Board: The Big Board

• Head First Software Development presents a technique similar to the storyboard

  • It shows how to break the user story to a finer granularity to get a more detailed view of the progress made each day

  • In particular, user stories are decomposed into tasks

    • the tasks are shown on the storyboard as well and move around as they are worked on and completed

• Here’s a demo of the big board technique
Big Board: How to Use?

- The Big Board is a major feature of your team’s workspace.
  - It is updated at least once per day during the stand up meeting (discussed next).
  - But could be useful to update it more often than that.
- It is a one-stop shop for getting a “big picture” view of the current iteration.
<table>
<thead>
<tr>
<th>User Stories</th>
<th>In Progress</th>
<th>Complete</th>
<th>Burn Down</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Next</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Completed</td>
</tr>
</tbody>
</table>

**Start with this…**

<table>
<thead>
<tr>
<th>Title: Show Current Deals</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong> The website will show current deals to Orion’s Orbits users.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title: Book package</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong> An Orion’s Orbits user will be able to book a special package with extras online.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title: Pay online</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description:</strong> An Orion’s Orbits user will be able to pay for their bookings online.</td>
</tr>
<tr>
<td>User Stories</td>
</tr>
<tr>
<td>--------------</td>
</tr>
</tbody>
</table>
| **Title:** Book package  
**Description:** An Orion's Orbits user will be able to book a special package with extras online. | | | |
| **Title:** Pay online  
**Description:** An Orion's Orbits user will be able to pay for their bookings online. | | | |
| **Title:** Show Current Deals  
**Description:** The website will show current deals to Orion's Orbits users. | | | |
| And add user stories | | | Next |
| one per swim lane | | | |

**Completed**

**Completed**
### User Stories

<table>
<thead>
<tr>
<th>User Stories</th>
<th>In Progress</th>
<th>Complete</th>
</tr>
</thead>
</table>
| **Title:** Book package  
**Description:** An Orion’s Orbits user will be able to book a special package with extras online. | | |
| **Title:** Pay online  
**Description:** An Orion’s Orbits user will be able to pay for their bookings online. | | |
| **Title:** Show Current Deals  
**Description:** The website will show current deals to Orion’s Orbits users. | | |

Now add tasks each with a description and estimate

**Completed**
### User Stories

<table>
<thead>
<tr>
<th>User Stories</th>
<th>In Progress</th>
<th>Complete</th>
<th>Burn Down</th>
</tr>
</thead>
</table>
| **Title:** Book package  
**Description:** An Orion’s Orbits user will be able to book a special package with extras online. | ![In Progress Icon] | ![Complete Icon] | ![Burn Down Chart] |
| **Title:** Pay online  
**Description:** An Orion’s Orbits user will be able to pay for bookings online. | ![In Progress Icon] | ![Complete Icon] | ![Burn Down Chart] |
| **Title:** Show Current Deals  
**Description:** The website will show current deals to Orion’s Orbits users. | ![In Progress Icon] | ![Complete Icon] | ![Burn Down Chart] |

**Tasks that a developer is working on move to “In Progress”**
<table>
<thead>
<tr>
<th>User Stories</th>
<th>In Progress</th>
<th>Complete</th>
<th>Burn Down</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title:</strong> Book package</td>
<td><img src="progress.png" alt="Status" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description:</strong> An Orion's Orbits user will be able to book a special package with extras online.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Title:</strong> Pay online</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description:</strong> An Orion's Orbits user will be able to pay for their bookings online.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Title:</strong> Show Current Deals</td>
<td><img src="status.png" alt="Status" /></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Description:</strong> The website will show current deals to Orion's Orbits users.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you stop working on a task, it goes back to its user story.
<table>
<thead>
<tr>
<th>User Stories</th>
<th>In Progress</th>
<th>Complete</th>
<th>Burn Down</th>
</tr>
</thead>
</table>
| **Title:** Book package  
**Description:** An Orion’s Orbits user will be able to book a special package with extras online. |  |  |  |
| **Title:** Pay online  
**Description:** An Orion’s Orbits user will be able to pay for bookings online. |  |  |  |
| **Title:** Show Current Deals  
**Description:** The website will show current deals to Orion’s Orbits users. |  |  |  |

Tasks that get finished move to the “Complete” section of the swim lane and more tasks get started.

**Completed**
<table>
<thead>
<tr>
<th>User Stories</th>
<th>In Progress</th>
<th>Complete</th>
<th>Burn Down</th>
</tr>
</thead>
</table>
| **Title:** Book package  
**Description:** An Orion’s Orbits user will be able to book a special package with extras online. | ![In Progress](image) | ![Complete](image) | ![Burn Down](image) |
| **Title:** Pay online  
**Description:** An Orion’s Orbits user will be able to pay for bookings online. | ![In Progress](image) | ![Complete](image) | ![Burn Down](image) |
| **Title:** Show Current Deals  
**Description:** The website will show current deals to Orion’s Orbits users. | ![In Progress](image) | ![Complete](image) | ![Burn Down](image) |

Be sure to update the Burn Down chart as you make (or don’t make) progress.
Its okay to be working on more than one task for a single user story
<table>
<thead>
<tr>
<th>User Stories</th>
<th>In Progress</th>
<th>Complete</th>
<th>Burn Down</th>
</tr>
</thead>
</table>
| **Title:** Pay online  
**Description:** An Orion’s Orbits user will be able to pay for their bookings online. | ![Progress](progress1.png) | ![Complete](complete.png) | ![Burn Down](burn_down.png) |
| **Title:** Show Current Deals  
**Description:** The website will show current deals to Orion’s Orbits users. | ![Progress](progress2.png) | | |
| **Title:** Book package  
**Description:** An Orion’s Orbits user will be able to book a special package online. | ![Progress](progress3.png) | ![Complete](complete.png) | |

Eventually all tasks for a user story are complete; the whole story moves to the Completed section.
User Stories

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pay online</td>
<td>An Orion’s Orbits user will be able to pay for their bookings online.</td>
</tr>
<tr>
<td>Book package</td>
<td>An Orion’s Orbits user will be able to book a special package with extras online.</td>
</tr>
</tbody>
</table>

Next

<table>
<thead>
<tr>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Current Deals</td>
<td>The website will show current deals to Orion’s Orbits users.</td>
</tr>
</tbody>
</table>

Completed

If a user story gets bumped (for whatever reason); move it to the Next section
<table>
<thead>
<tr>
<th>User Stories</th>
<th>In Progress</th>
<th>Complete</th>
<th>Burn Down</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Title: | Book package |
| Description: | An Orion’s Orbits user will be able to book a special package with extras online. |

| Title: | Pay online |
| Description: | An Orion’s Orbits user will be able to pay for their bookings online. |

| Title: | Show Current Deals |
| Description: | The website will show current deals to Orion’s Orbits users. |

Keep working until all stories are complete or have been pushed to the next iteration.
<table>
<thead>
<tr>
<th>User Stories</th>
<th>In Progress</th>
<th>Complete</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Don’t forget to update the Burn Down chart with your final status</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Burn Down</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title: Show Current Deals</td>
</tr>
<tr>
<td>Description: The website will show current deals to Orion’s Orbits users.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title: Book package</td>
</tr>
<tr>
<td>Description: An Orion’s Orbits user will be able to book a special package online.</td>
</tr>
<tr>
<td>Title: Play online</td>
</tr>
<tr>
<td>Description: An Orion’s Orbits user will be able to play online.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title: Pay online</td>
</tr>
<tr>
<td>Description: An Orion’s Orbits user will be able to pay for their bookings online.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title: Show Current Deals</td>
</tr>
<tr>
<td>Description: The website will show current deals to Orion’s Orbits users.</td>
</tr>
</tbody>
</table>
Summary

- **Iteration Management**
  - for each story: Analysis & Design, Implement, Test; split across 2 iterations

- **Agile Communication Plans**
  - Stand-Ups, SPMs, IPMs, Showcases, Retrospectives

- **Setting Up a Visual Workspace**
  - Inception Deck, Release Wall, Storyboard (Big Board), Velocity/Burn-Down
Coming Up Next

• Lecture 22: Cucumber: Testing User Interfaces and More

• Lecture 23: TBD