Agile Methods

Module 3: Executing Agile Projects

- User Stories
- Estimating & Tracking
- Scrumboards
User Stories

What are User Stories?

– Written on 4 x 6 index cards
  • If your card is too crammed, consider breaking out detail into additional cards with sub-stories

– Focus on capturing a conversation
  • User Stories are meant to be torn up
  • “Give the user what they need, not what was written down”

– Broaden by identifying User Roles & Acceptance Tests
  • User Roles: Who is the conversation with?
    – “As an Accountant, I can run an expense report for all employees of a given Business Unit”
  • Acceptance Tests: How do I know what done looks like?

“User Stories are a placeholder for a conversation”
User Story Best Practices

Guidelines for Writing Good User Stories (Cohn, 2004)

1. Start with Goal Stories
2. Slice the Cake – end-to-end functionality; resist desire to decompose
3. Put Constraints on Cards – like UC “biz rules” or NF requirementss
4. Size the Story to the Horizon – More detail the closer to implementation
5. Keep the UI out as long as possible
6. Some things aren’t Stories
7. Include User Roles in Stories – helps specificity
8. Write for One User – not to a user “class”; eliminates ambiguity
9. Write in Active Voice – e.g. “A Job Seeker can post a resume”, not “A resume can be posted by a Job Seeker”
10. Customer Writes
User Story – “A placeholder for a Conversation”

Short Title

The “Story”

As a <user> I want to <do something> so that <the benefit>

Add “cancel” button to undo changes and return to previous page.

TC: Check that browser returns to correct previous change.

Estimated: 2h
Actual:

Estimates and Actuals, usually in “story points”

User Acceptance
Test Description

ID & Date

Story 26
05-Dec-01
Commitment: Negotiating Stories

Business stakeholders are responsible for Scope, Priority, and the Composition of releases – whether a release meets an acceptance criteria.

Developers are empowered – they do estimates, determine tradeoffs, decide on processes. Accept responsibility for a deliverable but not how to do it!

Ron Jefferies 3 Cs*:

1. Card - “token representing the requirement”
2. Conversation - “exchange of thoughts, opinions, and feelings”
3. Confirmation - through acceptance tests
Agile Relationship

Customer → Define Value
Agile Relationship

CUSTOMER → DEFINE VALUE → PROGRAMMER → ESTIMATE COST
Agile Relationship

- Customer
  - Define Value
  - Programmer
    - Estimate Cost
  - Choose Value
  - Customer

ARIZONA STATE UNIVERSITY

8
Agile Relationship
Agile Relationship

“Lather, Rinse, Repeat” Every Iteration
Agile Methods: Planning Poker

- Agile version of Wideband Delphi
- Every developer participates, & maybe other team members
- Product Owner conducts the session.
- Script:
  1. Each estimator is given some cards with estimates on them
  2. Product Owner selects a story card and reads aloud
  3. Each estimator selects an estimate card and puts it face down
  4. When ready, all estimators “roll-up” their cards at the same time
  5. High & low estimators have a short discussion of the varying estimates
  6. Return to step 3 and continue until consensus is reached

- Issues:
  - Having the “right amount of discussion”
  - Getting assumptions out in the open
Scrumboards: *Information Radiators*

Maps out stories & tasks for all to see

Variants exist, but work goes through at least 3 states

– *To-do* – no one has picked it up on the team yet

– *In-process* – assigned to a team member
  – *Done* – assigned member completed it
    – I also like “Verify” but your mileage may vary

– When all tasks are done the story should be done

– When all stories are done the Sprint Goal should be met
Interactive: Create Backlogs

1. Create a Product Backlog for a common domain of your choosing
   - Some ideas might be MOOC site, ATM, museum kiosk, …
   - Create User Stories for the description
   - Put in your Product Backlog in the backlog area of your SB

2. Planning Poker
   - 1 team member stay as Product Owner, rest go to other tables as devs Following the Planning Poker script, estimate your stories

3. Construct Initial Tasks and Estimates

4. Negotiate and decide on Sprint 1 and Tasks (To Do)
Burndown charts

Burndown ("up") charts are basically earned values
- Team capacity set ahead of the iteration and called it’s “velocity”
- Plot iteration expectations via “story points” (business value)
- When a user story is completed subtract it from the total left to do

Unlike EVA, burndown charts can reflect change in scope

This chart, taken from an Alistair Cockburn paper, shows a burndown chart with allocations for increases and decreases in project scope
Example 1 (help.accept360.com)
Example 2 (quotient.net)
Why is it hard to do?

Estimating

Tracking