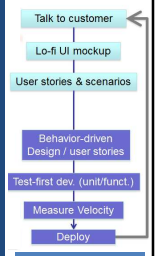
TCSS 360: SOFTWARE DEVELOPMENT AND QUALITY ASSURANCE

User Stories, Customer Meetings, Pivotal Tracker

Wes J. Lloyd
Institute of Technology
University of Washington - Tacoma



OBJECTIVES

- User stories – review
- Customer meetings
- Pivotal tracker

BDD: USER STORIES

- From human computer interaction (HCI) community
- Originally written on index cards
- 1 to 3 sentences
- Capture who, what, why
- Non-technical language
- Can be written by customers (users) or developers
- Cards: easy to rearrange (prioritize), promote brainstorming

```
Feature name: [descriptive name]
  As a [kind of stakeholder],
  So that [I can achieve some goal],
  I want to [do some task]
```


SMART USER STORIES

- **S**pecific, **M**easurable, **A**chievable, **R**elevant, and **T**ime-boxed
- **Specific** vs. vague features
  - Vague feature: User can search for a movie
  - Specific feature: User can search for a movie by title
- **Measurable**
  - A specific and measurable feature will be testable
  - Are the following features testable?

SMART USER STORIES - 2

- Testable = Specific + Measurable
  - Feature: RottenPotatoes should have a good response time
  - Is this testable?
  - How would you test it?
  - Feature: When adding a movie to RottenPotatoes, 99% of new movies added should be accessible to everyone within 3 seconds
  - Is this testable?
  - How would you test it?

SMART USER STORIES - 3

- **Achievable**
  - User stories should be scoped so that they are achievable in one agile iteration
  - If team completes less than 1 user story per iteration, user stories need to be decomposed into smaller stories
- **Relevant**
  - User stories must have business value to one or more stakeholders
  - A feature has business relevance if you can drill down and answer at least 5 "Why" questions

SMART USER STORIES - 4

- Relevance example
  - Feature: When adding a movie, the user will be able to post a link to the their new movie review to their facebook wall.
  1. **Why add the facebook feature? More people will go with friends and enjoy the movie.**
  2. **Why does it matter if people enjoy the movie? We will sell more tickets**
  3. **Why sell more tickets? Because the theater will make more money**
  4. **Why make more money? So the theater doesn't go out of business**
  5. **Why does it matter that the theater is in business? If not, I have no job**
  - If five WHY questions can't be answered, it may not be important

SMART USER STORIES - 5

- **Timeboxed**: User stories have time limits, when exceeded:
  - Give up
  - Divide story into smaller user stories
  - Reschedule remaining functionality using a new estimate
  - Reduce scope: ask customer to identify highest value parts of story, than can be done quickly
- Motivation
  - Extremely easy to underestimate length of a software project
  - Without careful accounting entire project could be late and fail
  - When exceeding a story time budget, refactoring the user story helps continually refine the project scope and maintain "tractability"

PREPARING TO MEET THE CUSTOMER

- Agile: customer collaboration over contract negotiation
- Never commit to delivering features X, Y, Z, by date D
- Initial contact: 30 to 60 minute phone call
- Explain the agile process
- Agile works on a time and materials basis, not a fixed bid basis
- Agile team asks customer for high level description of the system to be built
- If agile team and customer are a good fit, typically a 90-minute in-person scoping meeting is scheduled

MEETING THE CUSTOMER - 2

- Meeting attendees:
  - Customer: product manager, lead developer, designers
    Developer: two engineers
- Artifacts:
  - Existing designs (if any), and anything that may help clarify what system is to be built, etc.
- Meeting
  - Engineers ask series of questions to identify risks, external integrations, etc.
  - Seek to identify things leading to uncertain estimates

MEETING THE CUSTOMER - 3

- If clear definition: e.g. finished design, no external integrations
  - Agile team produces a tight-scoped estimate (e.g. 20-22 weeks)
- If product lacks definition: e.g. lots of external integrations, uncertainty
  - Agile team's estimate will have a wider range (e.g. 18-26 weeks)
- Agile team delivers
  - Findings, estimate, identification of risks to sales staff
  - Sales staff sends project proposal to client
- Cost estimation
  - Follows notion of Brook's Law: diminishing returns on team size
  - Agile team wants to advise the client about the ideal team size to avoid diminishing returns (higher cost, less output)

FOLLOW UP MEETINGS

- After 1st iteration, have working prototype
- Prior to meeting, customer must be able to play around with the app. This means:
  - App is deployed to the public cloud
  - App has (if appropriate) some "fake" data in it so customer can test the various behaviors
  - If app requires a login/password to do anything, setup & give credentials to customer.

FOLLOW UP MEETINGS - 2

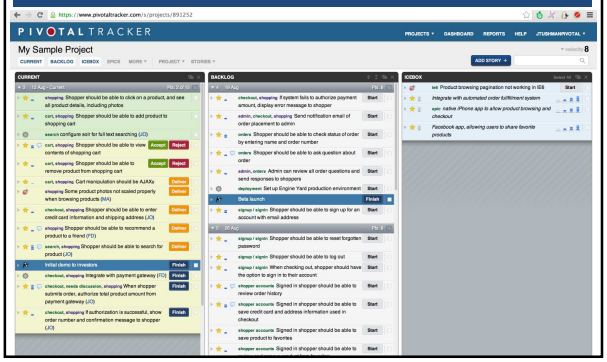
- Deploy early enough for customer to test out before meetings
  - At least 1-2 days in advance
    - Deploying and emailing the customer 11:30pm the night before your morning meeting is too late
- At the meeting: the customer drives and interacts with App
- Do not argue with customer feedback
  - OK to ask clarifying questions
  - OK to say if feasible given time budget
  - Don't argue if they prefer feature X over Y
  - Don't argue if they say UI experience is confusing

GOOD MEETINGS: SAMOSAS

(Photo by K.S. Poddar. Used by permission under CC-BY-SA-2.0.)

- Start and stop meeting promptly
- Agenda created in advance; no agenda, no meeting
- Minutes recorded so everyone can recall results
- One speaker at a time; no interrupting talker
- Send material in advance, since reading is faster
- Action items at end of meeting, so know what each should do as a result of the meeting
- Set the date and time of the next meeting
- Minutes and action items record results of meeting, start next meeting by reviewing action items

PIVOTAL TRACKER



PIVOTAL TRACKER

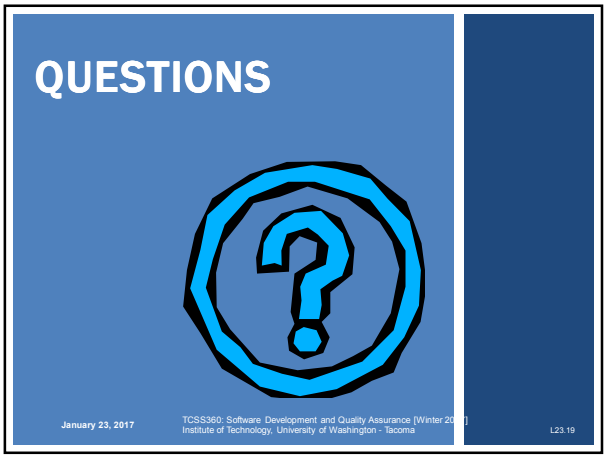
- SAAS tool for tracking user stories and measuring agile velocity
- Task prioritization: current panel, backlog, ice box
- Ordering within panel defines relative priority
- Completed stories go into Done panel
- Backlog priorities are suggested next
- Ice box: unprioritized, unscheduled user stories

PIVOTAL TRACKER - 2

- Release points: markers added to prioritized lists to indicate when releases will occur (relative to user story completion)
- Spike: short investigation into a technique or problem (e.g. prototype activity, exploration)
  - Enables tracking design exploration, etc.
  - Spikes not incorporated into delivered system - usually throw-away prototypes
- Epic: grouping of related user stories
  - Grouped in own panel, w/ separate progress bar

PIVOTAL TRACKER - 3

- User, not developer, accepts stories as completed
  - Developer presses deliver button, product owner performs acceptance testing, and hits "Accept" or "Reject" button
- Documents and UI sketches can be attached to stories
- Alternative to Pivotal Tracker:
  GitHub repositories includes a project Wiki, and separate logging to track project tasks/issues/defects providing project management support


QUESTIONS
January 23, 2017
TCSS360: Software Development and Quality Assurance [Winter 20
Institute of Technology, University of Washington - Tacoma
L23.19