User Stories

Putting Stories to Work

User Stories

- Front of the card:
  - User story
  - Estimate in story points (more on this later)

A recruiter can post a new job (3 points)

- Back of the card:
  - Captures the expectations of the user in the form of acceptance tests
  - These tests will help confirm that a story is complete

Try it with an empty job description (fail)
Try it with a really long job description (fail)
Try it with a six-digit salary (pass)

User Stories: Estimation

- The cost of a story is the estimate given to it by the developers
- Each story is assigned an estimate in story points
  - This indicates the size and complexity of the story relative to other stories
  - For example, a story estimated at four points is expected to take twice as long as a story estimated at two points

User Stories: Estimation

- Each team defines the meaning of story points:
  - One team might treat a point as an ideal day of work
  - Another team might define a story point as an ideal week of work
  - Yet another team might treat a story point as a measure of the complexity of a story
- **Recommendation for the project**: treat one story point as an ideal evening of work for one developer (ideal = no interruptions, good focus/concentration)

User Stories: Acceptance Tests

- The back of each card contains acceptance tests
  - They capture important aspects about stories
  - They verify that a story was developed to work exactly the way the customer team expected it to work
  - They run frequently – ideally automated
  - They are intentionally left short and incomplete
  - The team can add/remove tests at any time
- Goal is to convey additional information so developers will know when the story is done

User Stories: Acceptance Tests

- Story: *A user can pay for the items in a shopping cart with a credit card*

Test with Visa, MasterCard and American Express (pass).
Test with Diner's Club (fail).
Test with a Visa debit card (pass).
Test with good, bad, and missing security codes.
Test with expired cards (fail).
Test with different purchase amounts (try going over card limit)

- By providing tests to the developer early, the customer team has:
  - Stated their expectations
  - Reminded the developers of a situation they might have forgotten

Story-Driven Development

- The customer team and developers choose length of each iteration of development of the project
  - Might be anywhere from 1 to 4 weeks
  - Same iteration length used for entire project
  - At the end of each iteration, developers are responsible for delivering fully usable code for some subset of the application
  - The functionality captured by the stories in the iteration just completed should now be fully usable in the application

Story-Driven Development: Release Planning

- Developers estimate how much work (in story points) they'll be able to complete each iteration to set the project's velocity
  - First estimate will be wrong, but this is useful to roughly determine the iteration schedule
- To plan releases, we sort stories in piles representing iterations, where the stories in each pile add up to no more than the estimated velocity
  - Highest-priority stories go in the first pile (iteration 1)
  - Next highest-priority stories go in the second pile (iteration 2)
  - …

Story-Driven Development: Release Planning

- Before each iteration starts, the customer team can make mid-course corrections to the plan
- As we complete iterations, we can determine the development team's actual velocity
  - We can work with this instead of the estimated velocity
  - This means that each pile of stories may need to be adjusted by adding or removing stories

Story-Driven Development: Release Planning

Stories

Story	Story Points
Story A	3
Story B	5
Story C	5
Story D	3
Story E	1
Story F	8
Story G	5
Story H	5
Story I	5
Story J	2

Release Plan
Estimated velocity of 13

Iteration	Stories	Story Points
1	A, B, C	13
2	D, E, F	12
3	G, H, J	12
4	I	5

Story-Driven Development: Prioritization

- To plan releases, customer team must prioritize stories:
  - Desirability of a feature to a broad base of users/customers
  - Desirability of a feature to a small number of important users/customers
  - Cohesiveness of a story in relation to others
    - For example, a zoom out story might not be high priority on its own, but might treated as such because it is complementary to zoom in, which is high priority

Story-Driven Development: Prioritization

- Developers have different priorities for many stories:
  - They may suggest a story’s priority be changed based on technical risk or because it is complementary to another

- Customer team listens to their opinions, but ultimately sets priorities in a manner that maximizes value delivered to the organization

BDUF vs. Story-Driven Development

- Traditional waterfall model:
  - Write requirements, analyze them, design a solution, code said solution, test solution
  - Customers involved at beginning to write requirements
  - Customers involved at the end for acceptance testing
  - Between requirements and acceptance: customers often disappear
- Story-Driven Project:
  - Customers and users involved throughout duration of the project
  - Not allowed to disappear in the middle of the project