System Development
Scrum 2

Datamatiker /Computer Science

2nd Semester

Spring 2017

Agenda for Scrum Day 2

- Product backlog
  - User stories
  - Done criteria
  - Estimation

Good User Stories

Stories – how to write GOOD stories!

1. Identify stories	= PO responsibility
2. Write stories	= PO responsibility
3. Estimate stories	= team responsibility

Apply INVEST criteria for each story

- I – Independent
- N – Negotiable
- V – Valuable
- E – Estimable
- S – Small
- T – Testable

Independent Stories

- Stories are easiest to work with if they are independent.
- We'd like stories to not overlap in concept
- We'd like to be able to schedule and implement stories in any order.

Negotiable… … and Negotiated Stories

- A good story is negotiable
- Story isn’t an explicit contract for features; Rather, details will be co-created by the PO and Team.
- A good story captures the essence, not the details

Valuable Stories 1

- A story needs to be valuable to the customer

- What about Tech Stories? (H. Kniberg p. 39)
  - Examples:
    - Install continuous build server
    - Write a system design overview
    - Refactor the data layer
    - Update bug tracking system

  - What do to?
    1. Transform into normal story
    2. Be a task in another story
    3. Define and keep in separate list
       - Let Product Owner see, but not edit
       - Negotiate with Product Owner

Valuable Stories 2

- Valuable – to who?
  - Customer (purchaser & user)
  - Secondarily developer

Examples:

Valued by purchaser, but maybe not the users:

"All configuration information is read from a central location"

"The development team will produce the software in accordance with CMM Level 3"

Valued by both customer and developer… if changed from

"All error handling and logging is done through a set of common classes"

into this text:

"All errors are presented to the user and logged in a consistent manner"

Estimable Stories 1

- A good story can be estimated
- We don't need an exact estimate, but just enough to help the Product Owner rank and schedule the story's implementation
- Being estimable is
  - partly a function of being negotiated, as it's hard to estimate a story we don't understand
  - Also a function of size: bigger stories are harder to estimate
- And of the team: what's easy to estimate will vary depending on the team's experience

Estimable Stories 2

- Why difficult to estimate stories?
  1. Developers lack domain knowledge
  2. Developers lack technical knowledge
  3. The story is too big

Solutions

1. Discuss with customer
2. Turn into two stories:
   a) a quick spike to gather information
   b) a story to do the real work.
3. Decompose into smaller, constituent stories

Small Stories

- Good stories tend to be small
- Stories typically represent <u>at most</u> a few person-weeks worth of work (that is actually long time)
  - Often teams try to restrict them to a day of work
- Above this size, it seems to be too hard to know what's in the story's scope

Testable Stories

- A good story is testable
- Writing “how to demo” accept criteria carries an implicit promise: "I understand what I want well enough that I could write a test for it."
- Will be used in sprint review – is the story done?

Done User Story

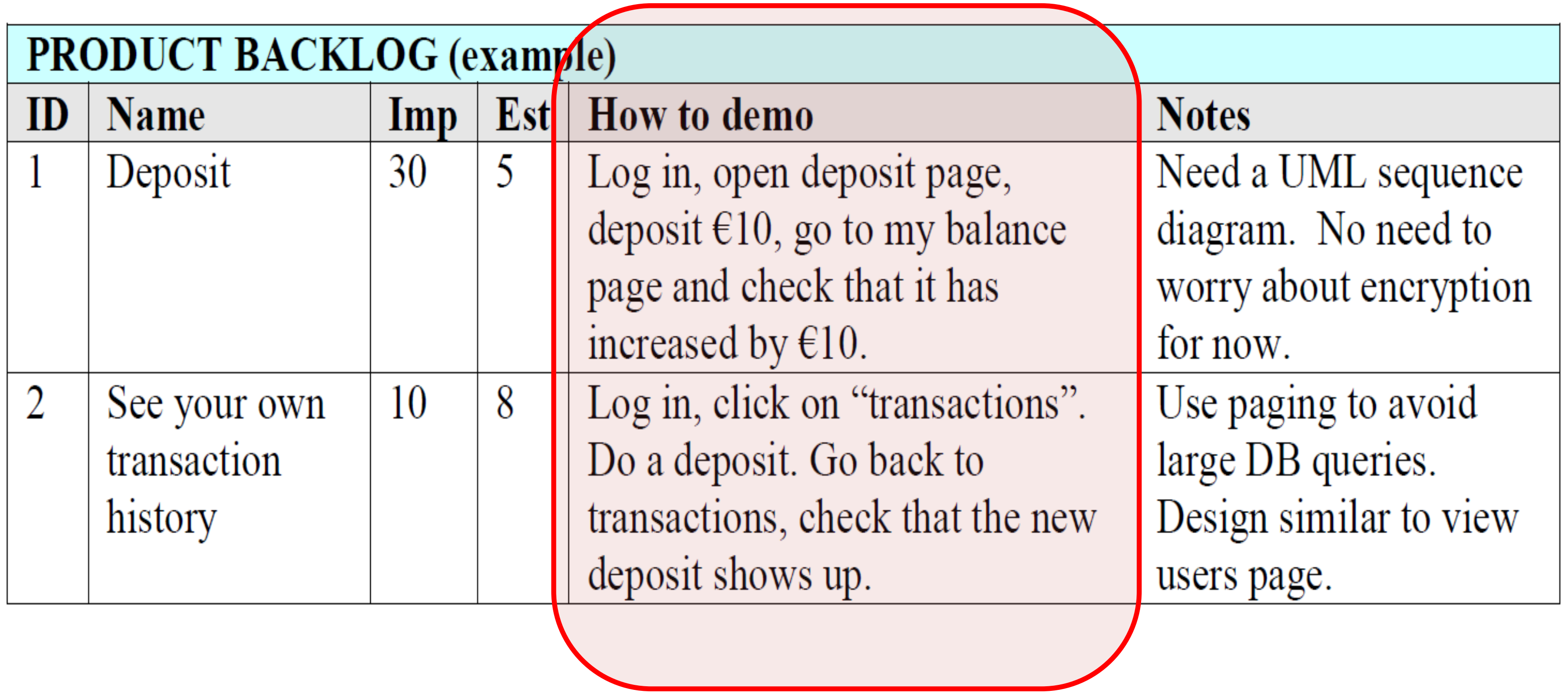
Acceptance Criteria …

- Bring the project from *"It Works as Coded"* to *"It Works as Intended"*
- Are *conditions* that a story must satisfy to be *accepted* by a user/customer/other stakeholder (PO in Scrum)
- Are a set of *statements*, each with a *clear pass/fail result*, that specify both functional and non-functional requirements
  - Functional example: *When a user clicks on the 'Reports' dropdown, a list of available reports will be displayed*.
  - Non-functional example: *Form edit buttons will be blue, and Form workflow buttons will be green.*

Source: http://www.seguetech.com/blog/2013/03/25/characteristics-good-agile-acceptance-criteria

Accept Criteria for Story - Example

Accept criteria:



PRODUCT BACKLOG (example)

ID	Name	Imp	Est	How to demo	Notes
1	Deposit	30	5	Log in, open deposit page, deposit €10, go to my balance page and check that it has increased by €10.	Need a UML sequence diagram. No need to worry about encryption for now.
2	See your own transaction history	10	8	Log in, click on "transactions". Do a deposit. Go back to transactions, check that the new deposit shows up.	Use paging to avoid large DB queries. Design similar to view users page.

User Story Estimation

Story Estimation Technique

- **S, M, L and XXXXL**
- Each estimator has four cards S, M, L and XXXXL (epic)
- Each estimator privately selects one card to represent his estimate for a story. All cards are revealed at the same time
- If consensus, that will be the estimate
- If not, discussion will lead to re-estimation until consensus
  - Possibly decompose stories into smaller stories

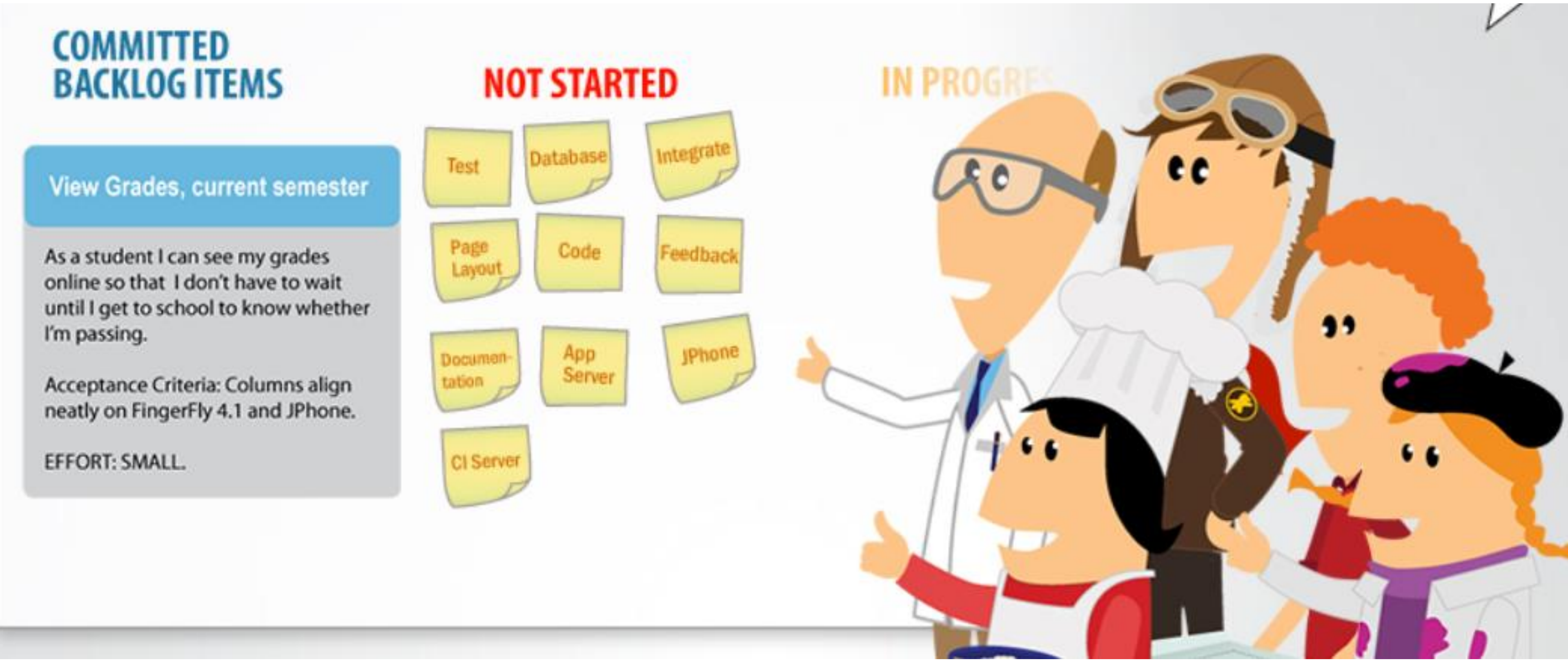
Story Estimation Technique

- A deck of **Planning Poker** cards with values like **1, 2, 5, 8, 13, 20, 40, 100** and ? (I don’t know), coffee cup (I want a break)
  - The values represent number of story points, ideal days, hours, or other unit in which the team calculates its estimations
- Each estimator privately selects one card to represent his estimate for a story. All cards are revealed at the same time
- If consensus, that will be the estimate
- If not, discussion will lead to re-estimation until consensus (Possibly decompose stories into smaller stories)

How to Estimate

- What is best approach to reach realistic estimate of a story?

Sprint Backlog Example


COMMITTED BACKLOG ITEMS
NOT STARTED
IN PROGRE
View Grades, current semester
As a student I can see my grades online so that I don't have to wait until I get to school to know whether I'm passing.
Acceptance Criteria: Columns align neatly on FingerFly 4.1 and JPhone.
EFFORT: SMALL.
Test
Database
Integrate
Page Layout
Code
Feedback
Documen-tation
App Server
JPhone
CI Server

See more in Sprint Planning Meeting video at http://scrumtrainingseries.com/

How to break stories into thin vertical slices

- Exercise that trains a software development team to better understand the benefits from dividing user stories into very thin but still vertical slices
- The idea is to learn how to do small work slices (of 15-20 minutes!), and continuously making progress while keeping confidence high ☺