

Software Development Processes: *User Stories*

Software Engineering
Millersville University

The Planning Game: User Stories

- Write on index cards (or on a wiki)
 - meaningful title
 - short (customer-centered) description
- Focus on “what” not the “why” or “how”
- Uses client language
 - Client must be able to test if a story is completed
- No need to have all stories in first iteration

Example: Accounting Software

- CEO: “I need an accounting software using which I can create a named account, list accounts, query the account balance, and delete an account.”
- Analyze the CEO’s statement and create some user stories

User Stories

Title: Create Account

Description: I can create a named account

Title: List Accounts

Description: I can get a list of all accounts.

Title: Query Account Balance

Description: I can query account balance.

Title: Delete Account

Description: I can delete a named account

User Stories

Title: Create Account
Description: I can create a named account

Title: List Accounts
Description: I can get a list of all accounts.

How is the list ordered?

Title: Query Account Balance
Description: I can query account balance.

Title: Delete Account
Description: I can delete a named account

User Stories

Title: Create Account
Description: I can create a named account

Title: Query Account Balance
Description: I can query account balance.

Title: List Accounts
Description: I can get a list of all accounts. I can get an alphabetical list of all accounts.

How is the list ordered?

Title: Delete Account
Description: I can delete a named account

User Stories

Title: Create Account

Description: I can create a named account

Title: List Accounts

Description: I can get a list of all accounts. I can get a list of

Can I delete if a balance is not zero?

Title: Query Account Balance

Description: I can query account balance.

Title: Delete Account

Description: I can delete a named account

User Stories

Title: Create Account

Description: I can create a named account

Title: List Accounts

Description: I can get a list of all accounts. I can get a list of

Can I delete if a balance is not zero?

Title: Query Account Balance

Description: I can query account balance.

Title: Delete Account

Description: I can delete a named account if the balance is zero.

User Story?

Title: Use AJAX for UI

Description: The user interface will use AJAX technologies to provide a cool and slick online experience.

User Story?

Title: Use AJAX for UI

Description: The user interface will use AJAX technologies to provide a cool and slick online experience.



Not a user story

Customer Acceptance Tests

- Client must describe how the user stories will be tested
 - With concrete data examples,
 - Associated with (one or more) user stories
- Concrete expressions of user stories

User Stories

Title: Create Account

Description: I can create a named account

Title: List Accounts

Description: I can get a list of all accounts. I can get an alphabetical list of all accounts.

Title: Query Account Balance

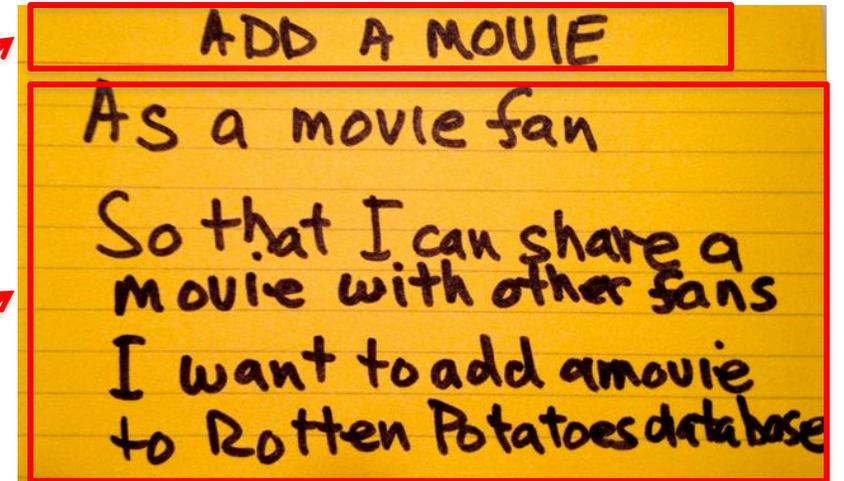
Description: I can query account balance.

Title: Delete Account

Description: I can delete a named account if the balance is zero.

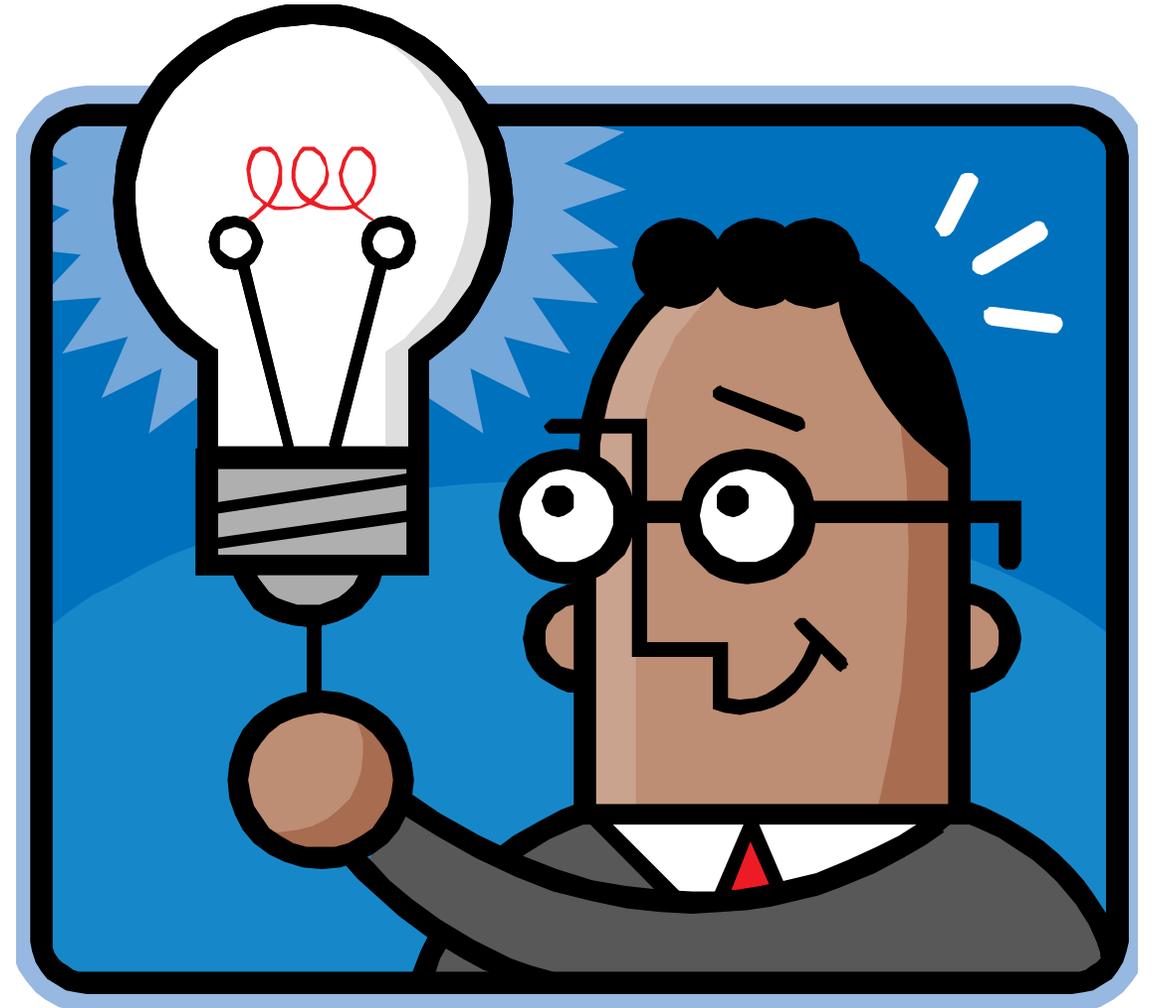
User Stories

- 1-3 sentences in everyday language
 - Fits on 3" x 5" index card
 - Written by/with customer
- “Connextra” format:
 - Feature name
 - **As a** [kind of stakeholder],
So that [I can achieve some goal],
I want to [do some task]
 - 3 phrases must be there, can be in any order
- Idea: user story can be formulated as *acceptance test before* code is written



SMART Stories

- **Specific**
- **Measurable**
- **Achievable**
(ideally, implement in 1 iteration)
- **Relevant**
("the 5 why's")
- **Timeboxed**
(know when to give up)





Specific & Measurable

- Each scenario testable
 - Implies known good input and expected results exist
- Anti-example:
“UI should be user-friendly”
- Example: Given/When/Then
 1. *Given* some specific starting condition(s),
 2. *When* I do X,
 3. *Then* one or more specific thing(s) should happen

Achievable



- Complete in 1 iteration
- If can't deliver feature in 1 iteration, deliver subset of stories
 - Always aim for working code @ end of iteration
- If <1 story per iteration, need to improve point estimation per story

Relevant – Business Value

- Discover business value, or kill the story:
 - Protect revenue
 - Increase revenue
 - Manage cost
 - Increase brand value
 - Making the product remarkable
 - Providing more value to your customers

Timeboxed

- Stop story when exceed time budget
 - Give up or divide into smaller stories or reschedule what is left undone
- To avoid underestimating length of project
- Pivotal Tracker tracks velocity, helps avoid underestimate



User Stories => Acceptance Tests?

- Wouldn't it be great to automatically map 3x5 card user stories into tests for user to decide if accept the app?
- How would you match the English text to test code?
- How could you run the tests without a human in the loop to perform the actions?

Behavior Driven Development: Big Idea

- Tests from customer-friendly user stories
 - Acceptance: ensure satisfied customer
 - Integration: ensure interfaces between modules consistent assumptions, communicate correctly
- Meet halfway between customer and developer
 - User stories are not code, so clear to customer and can be used to reach agreement
 - Also not completely freeform, so can connect to real tests

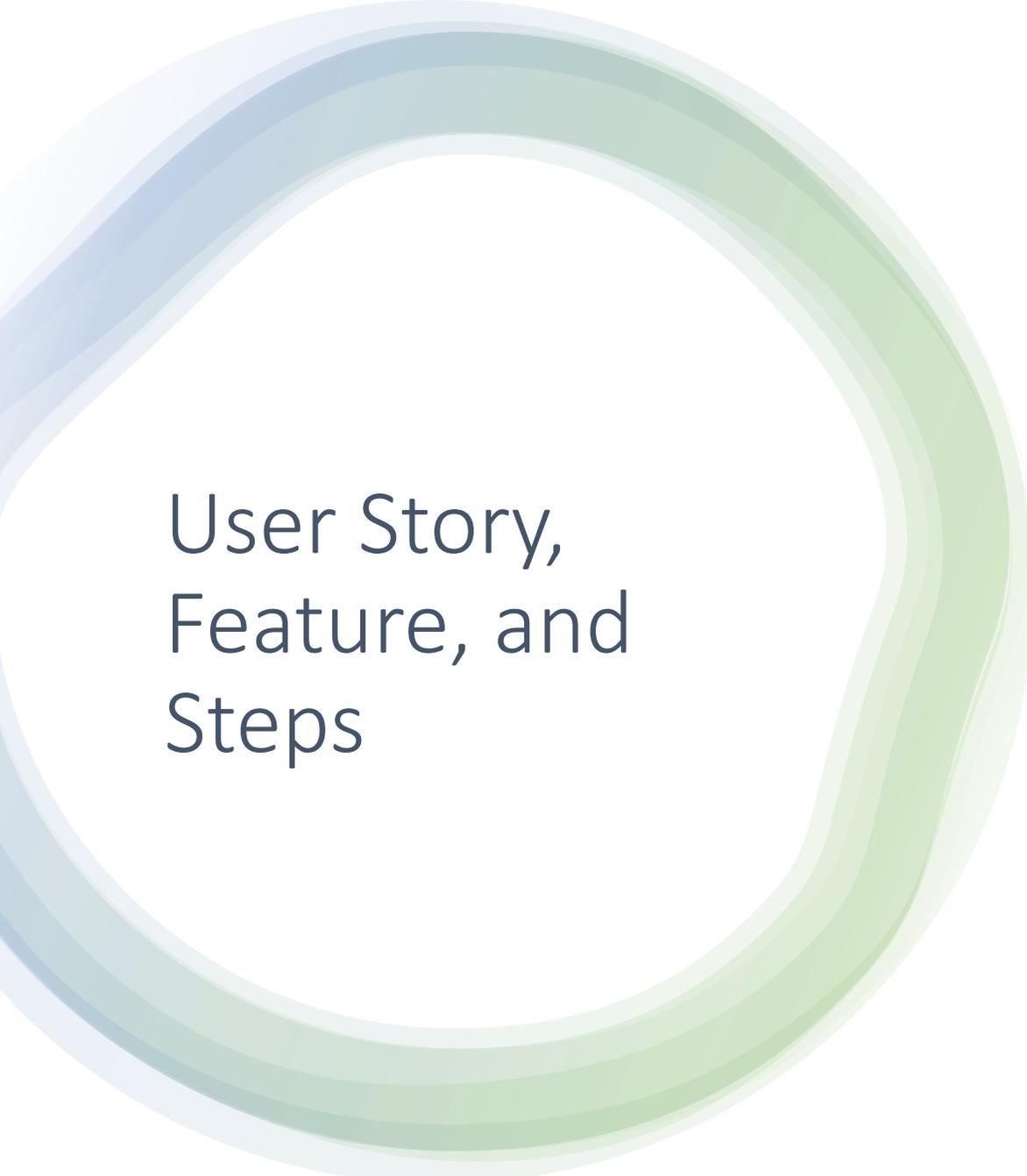
Example User Story

Feature: User can manually add movie 1 Feature

Scenario: Add a movie ≥ 1 Scenarios / Feature

Given I am on the RottenPotatoes home page
When I follow "Add new movie"
Then I should be on the Create New Movie page
When I fill in "Title" with "Men In Black"
And I select "PG-13" from "Rating"
And I press "Save Changes"
Then I should be on the RottenPotatoes home page
And I should see "Men In Black"

3 to 8 Steps / Scenario



User Story, Feature, and Steps

- **User story:** refers to single feature
- **Feature:** ≥ 1 scenarios that show different ways a feature is used
 - **Keywords** Feature and Scenario identify respective components
- **Scenario:** 3 - 8 steps that describe scenario
- **Step definitions:** code to test steps

5 Step Keywords

1. **Given** steps represent state of world before event: *preconditions*
2. **When** steps represent event
 - e.g., simulate user pushing a button
3. **Then** steps represent expected *postconditions*; check if true
4. / 5. **And & But** extend previous step

