COMP1531 Requirements - Use Cases, User Stories

Lecture 8.3

Author(s): Hayden Smith



(Download as PDF)

In This Lecture

- 🕨 Why? 🤔
 - Requirements can be very human, and express complex flows. We need ways to model this.



- User Stories
- User Acceptance Tests
- Use Cases
- Use Case Representations



Overview

User Stories are a method of requirements engineering used to inform the development process and what features to build with the **user at the centre**.



Structure

When a customer tells you what they want, try and express it in the form **As a < type of**

user >, I want < some goal > so that < some reason >

- E.G. They say:
 - A student can purchase monthly parking passes online
- But your story becomes:
 - As a student, I want to purchase a parking pass so that I can drive to school



Attributes

- Are written in non-technical language
- Are focused on keeping the customer at the core of the experience
- Are user-goal focused, not product-feature focused
 - User stories ideally describe problems, not solutions



Activity

Let's write out some user stories for a to-do list!



Good properties of user stories:

- Independent: Can be developed/delivered independently
- Negotiable: avoid too much detail.
- Valuable: must hold some value to the client
- Estimable: we'll get to this in a later lecture
- Small: user story should be small
- Testable



Further Reading

• Atlassian User Stories

We need a method to help us "test" weather a user story has been satisfied.

- Break down a user story into criteria that must be met for the user, or customer, to accept.
- Written in natural language.
- Can be refined before implementation.

For example: As a user, I want to use a search field to type a city, name, or street, so that I can find matching hotel options.

User Acceptance Criteria:

- The search field is placed on the top bar
- Search starts once the user clicks "Search"
- The field contains a placeholder with a grey-colored text: "Where are you going?"
- The placeholder disappears once the user starts typing
- Search is performed if a user types in a city, hotel name, street, or all combined
- The user can't type more than 200 symbols

For example: As a user, I can log in through a social media account, because I always forget my passwords

User Acceptance Criteria:

- Can log in through Facebook
- Can log in through LinkedIn
- Can log in through Twitter

- Acceptance criteria should not be too broad (but nor should they be too narrow).
- Minimise technical detail. They can be more technical than the story itself, but client still needs to understand them.
- While they can be updated during development, they should first be written before it starts.
- Acceptance Tests are tests that are performed to ensure acceptance criteria have been met.
- Not all acceptance criteria can easily be mapped to automated acceptance tests.
 Sometimes you need to setup User Acceptance Testing environments for real users to manually use it and provide feedback.
- Acceptance tests are black-box tests.

🐳 Alternative: Scenario Oriented

- The Acceptance criteria from before are often referred to a **rule-based AC**
- Sometimes it is preferable to have AC that describe a scenario
- This can be done in the Given/When/Then format:
 - Given some precondition
 - When I do some action
 - Then I expect some result



Example: As a user, I want to be able to recover the password to my account, so that I will be able to access my account in case I forgot the password.

User Acceptance Criteria:

- Scenario: Forgot password
- **Given**: The user has navigated to the login page
- When: The user selected forgot password option
- And: Entered a valid email to receive a link for password recovery
- Then: The system sent the link to the entered email
- **Given**: The user received the link via the email
- When: The user navigated through the link received in the email
- Then: The system enables the user to set a new password



- Rule-based acceptance criteria are simpler and generally work for all sorts of stories
- Scenario-based AC work for stories that imply specific user actions, but don't work for higher-level system properties (e.g. design)
- Scenario-based AC are more likely to be implementable as tests



- Represent a dialogue between the user and the system, with the aim of helping the user achieve a business goal.
- The user initiates actions and the system responds with reactions.
- They consider systems as a black box, and are only focused on high level understanding of flow.



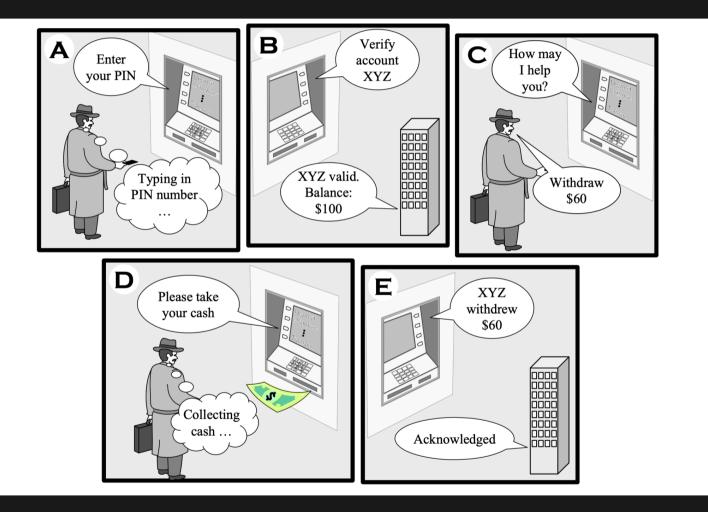
- Generally you can represent use cases as:
 - Informal list of steps (written)
 - Diagramatic (visual)
- There is a range of different approaches that can be taken too, e.g. Cockburn style (not required reading)



Template for providing background.

- Use Case: [the name should be the goal as a short active verb phrase]
- Goal in Context: [a longer statement of the goal, if needed]
- **Scope**: [what system is being considered black-box under design]
- **Preconditions**: [what we expect is already the state of the world]
- Success End Condition: [the state of the world upon successful completion]
- Failed End Condition: [the state of the world if goal abandoned]
- **Primary Actor**: [a role name for the primary actor, or description]
- **Trigger**: [the action upon the system that starts the use case, may be time event]







We can provide a use case in written form.

- Step 1. ATM asks customer for pin
- Step 2. Customer enters pin
- Step 3. ATM asks bank to verify pin and account
- Step 4. Bank informs ATM of validity and balance of account
- Step 5. ATM asks customer what action they wish to take
- Step 6. Customer asks to withdraw an amount of money
- Step 7. ATM Dispenses money to customer
- Step 8. ATM informs bank of withdrawal



Let's take the opportunity to build our requirements for a UNSW monorail. Ensure some of the requirements are expressed in terms of user stories and/or use cases.



- If you wish to know more about use cases, see here:
 - Software Engineering Ivan Marsic (Chapter 2, Section 4)
 - http://www.cs.otago.ac.nz/coursework/cosc461/uctempla.htm
 - Writing Effective Use Cases Alistair Cockburn
- https://www.mountaingoatsoftware.com/blog/the-two-ways-to-add-detail-to-userstories
- https://www.altexsoft.com/blog/business/acceptance-criteria-purposes-formats-andbest-practices/
- https://dzone.com/articles/acceptance-criteria-in-software-explanation-exampl





Or go to the form here.