Test Driven
Lasse Koskela
Chapter 9: Acceptance TDD Explained

Paul Ammann
&
Jeff Offutt

2018

Overview

"In the spacecraft business no design can survive the review process without first answering the question—how are we going to test this thing?"

—Glen Alleman

9.1 Introduction To User Stories

- Format of a story
  - Free form
  - Or structured: As a (*role*) I want (*functionality*) so that (*benefit*)
  - Often written on index cards
- Card, conversation, confirmation (CCC)
- Power of storytelling
  - User view of what is needed, but not how it is provided
- A user story represents a requirement, and creates a promise to communicate with the customer later

Storytelling reveals meaning without defining it
—Hannah Arendt

Example User Stories

Support technician sees customer's history on-screen at the start of a call

Application authenticates with the HTTP proxy server

The system prevents user from running multiple instances of the application simultaneously

State what, **NOT** how

Enabling value : A user story is *valuable* because it *enables* engineers to add functionality

9.2 Acceptance Tests

- Create tests based on user stories
- Properties of user stories
  - Owned by customer
  - Written together with customer, developer, and tester
  - Focus on the what, not the how
  - Expressed in language of the problem domain—user's vocabulary
  - Concise, precise, and unambiguous

In-class discussion:

- *Consider the 3 user stories on previous slide (pg 326)*
- *Discuss whether and how they satisfy these properties*

Acceptance Tests—Example Tests

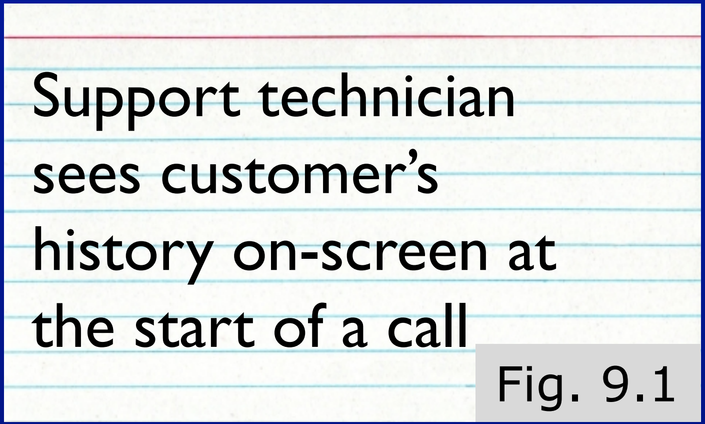

Support technician sees customer's history on-screen at the start of a call

Fig. 9.1

- Simulate a call with Fred's account number and verify that Fred's info can be read from the screen
- Verify that the system displays a valid error message for a non-existing account number
- Omit the account number in the incoming call completely and verify that the system displays the text "no account number provided" on the screen

Fig. 9.2

Acceptance Tests—What vs. How

- Go to the "new transaction" screen, fill in the required details, and save the entry; verify that the transaction shows up on the list
- Select the "delete" checkbox for the newly created entry, click "delete all marked transactions," and verify that they're gone
- Create multiple transactions, check several of them and delete; verify that all selected transactions were indeed deleted

Fig. 9.3

In-class discussion: What is wrong with these tests?

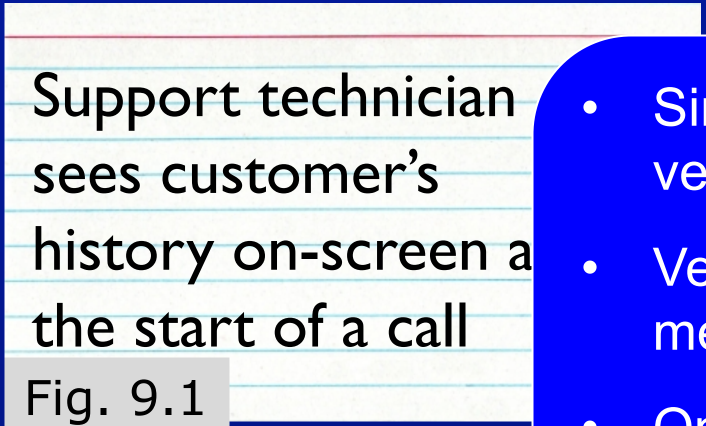
Too much HOW for users

<u>Trimmed to focus on WHAT</u>

- Try creating a new transaction
- Try deleting a transaction
- Try deleting multiple transactions

Fig. 9.4

Acceptance Tests—What vs. How



Support technician sees customer's history on-screen a the start of a call

Fig. 9.1

- Simulate a call with Fred's account number and verify that Fred's info can be read from the screen
- Verify that the system displays a valid error message for a non-existing account number
- Omit the account number in the incoming call completely and verify that the system displays the text "no account number provided" on the screen

Fig. 9.2

Too detailed

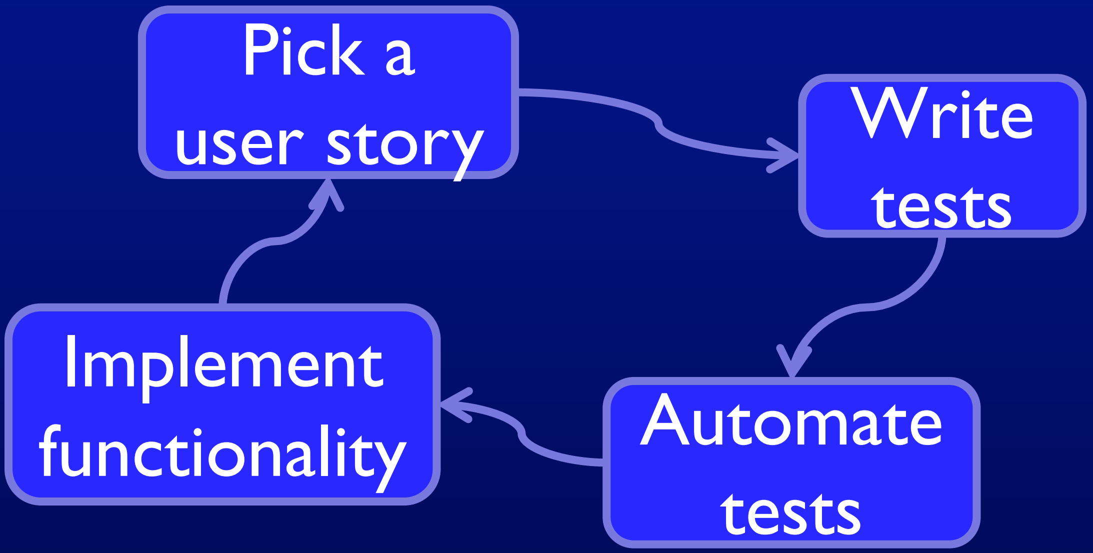
<u>Trimmed version of tests in fig. 9.2</u>

- Valid account number
- Non-existing account number
- No account number provided

Fig. 9.5

9.3 Understanding the Process

- The acceptance TDD cycle
  1. Pick a story
  2. Write tests for the story
  3. Automate the tests
  4. Implement the functionality



A process with feedback

A-TDD Process Step 1

- The acceptance TDD cycle
  1. Pick a story
     - Most important
     - Business value
     - Technical risk
     - Amount of programming
  2. Write tests for the story
  3. Automate the tests
  4. Implement the functionality

A-TDD Process Step 2

- The acceptance TDD cycle
  1. Pick a story
  2. Write tests for the story
     - Involve the customer
     - Iterate
     - Keep abstract as long as possible
     - Get ahead of refactoring
  3. Automate the tests
  4. Implement the functionality

A-TDD Process Step 3

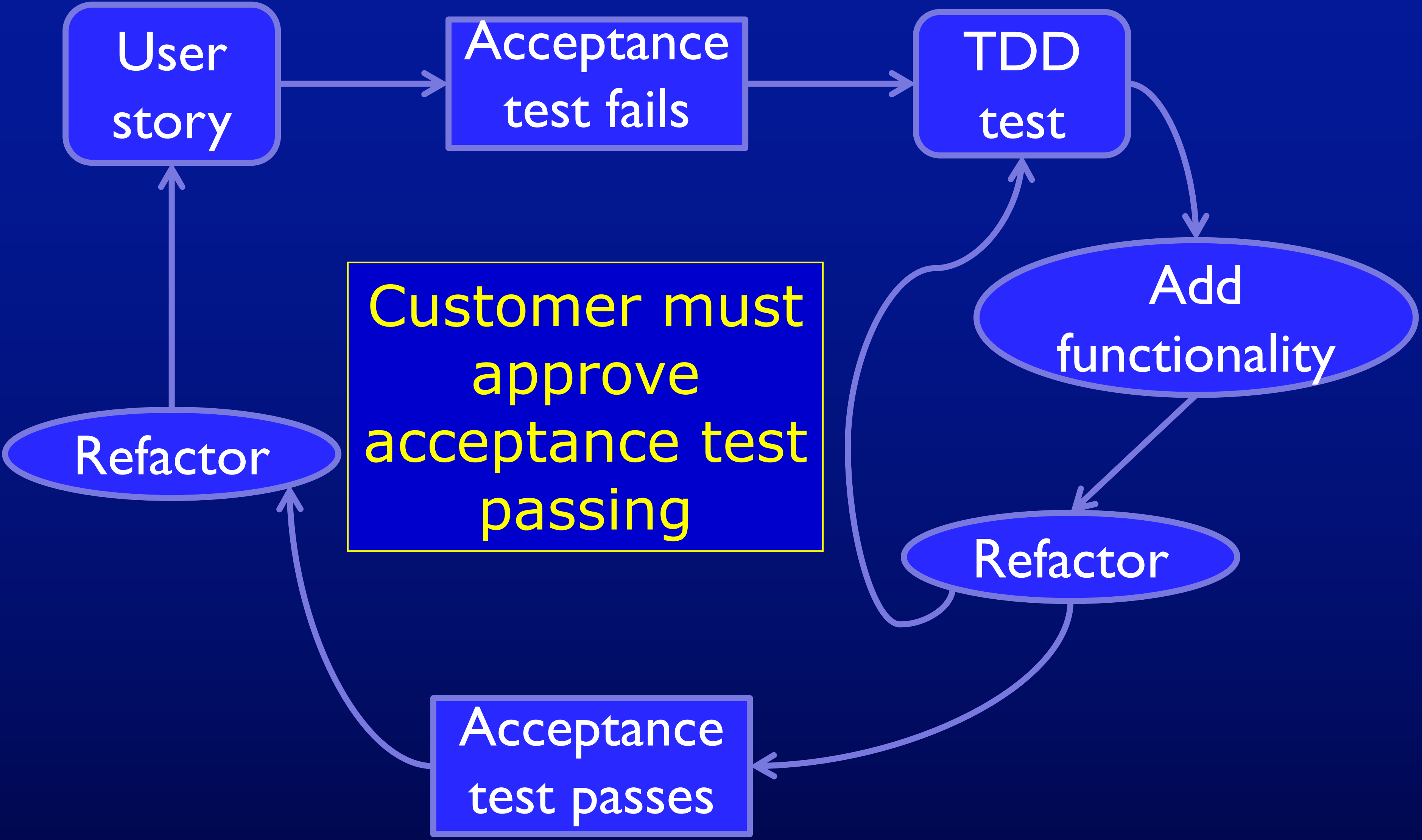
- The acceptance TDD cycle
  1. Pick a story
  2. Write tests for the story
  3. Automate the tests
     - Start with a table format
     - Translate to implementation
     - Postpone use of tools—tools steal focus from the topic
  4. Implement the functionality

A-TDD Process Step 4

- The acceptance TDD cycle
  1. Pick a story
  2. Write tests for the story
  3. Automate the tests
  4. Implement the functionality

- Each A-TDD test leads to multiple small tests

Acceptance Tests in Agile Methods



Customer must approve acceptance test passing

9.4 Acceptance Testing as a Team Activity

- Defining the customer role
  - Representative of end users
  - Possibly several people
- Characteristics of customer role
  - Shared interest in success
  - Authority to make decisions
  - Ability to understand implications
  - Ability to explain domain

Key is to verify against target domain

Acceptance Testing Team

- Who writes tests with the customer?
  - Tester ?
  - Developer ?
  - Requirements expert ?
  - Everybody ?
- How many testers do we need?
  - One or two developers per tester
  - Tester is a role, not a job title
  - All developers should be testers

More contributors is better

9.5 Benefits of Acceptance Testing

- Definition of "done"
  - Customer must agree it's done
  - Knowing where we are
  - Knowing when to stop
  - Test criteria satisfied
- Cooperative work
- Trust and commitment
- Specification by example
  - This is a big one!
- Filling the gap
  - Unit tests are not the same as acceptance tests

Both unit and acceptance tests needed

9.6 What Are We Testing, Exactly?

- Should we test against the UI?
  - Do whatever is easier long term
  - UIs are often in the way
  - Good tools can automate tests through or around the UI
  - Performance might matter
- Should we stub our system?
  - Sufficiently close to the real thing
  - Sometimes stubs are necessary
- Should we test business logic directly?
  - Of course—it's what the customer cares about

Tests are like votes—they need to run early and often