Introduction to Software Testing
Acceptance TDD Explained (KO Ch. 9)

Software Testing & Maintenance

SWE 437

http://go.gmu.edu/swe437

Dr. Brittany Johnson-Matthews

(Dr. B for short)

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

Today we're gonna talk about:

User stories

From user stories to acceptance tests

The overall process

More on 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

"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

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

Acceptance tests (9.2)

Create tests based on user stories

Properties of acceptance tests include:

- Owned by customer
- Written together with the 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 Exercise

Discuss

acceptance tests

Consider the 3 user stories on previous slide (pg. 326)

Discuss whether and how they satisfy these properties

Example acceptance 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 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

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

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

- Too much **HOW** for users
- Not in users' **vocabulary**

Trimmed to focus on **WHAT**

1. Try creating new order
2. Try deleting an order
3. Try deleting multiple orders

Acceptance tests – what vs. how

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

Fig. 9.1

Too detailed

- 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 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

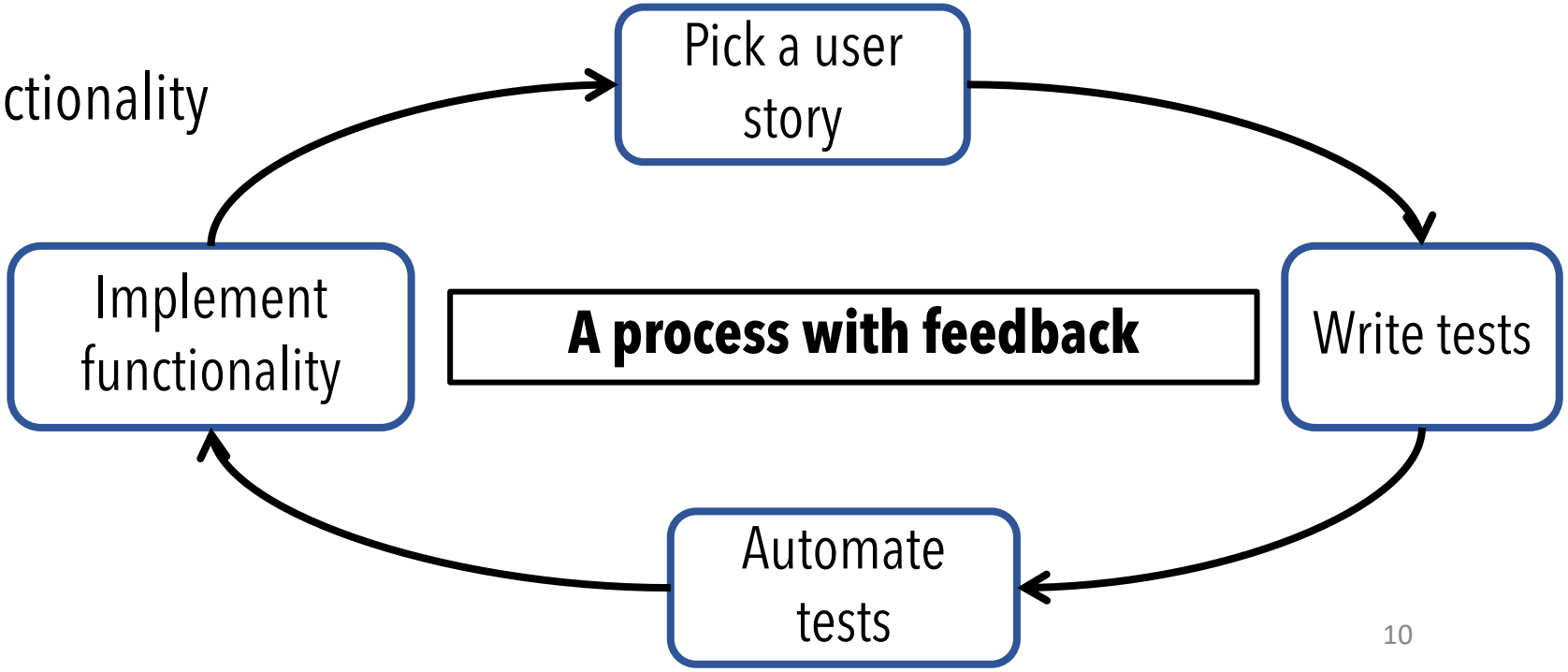
Trimmed version of tests

1. Valid account number
2. Non-existing account number
3. No account number provided

Understanding the process (9.3)

The acceptance TDD cycle

1. Pick a story
2. Write tests for the story
3. Automate the tests
4. Implement the functionality



Step 1: Pick a story

The acceptance TDD cycle

1. Pick a story *(which story?)*
   - Most important
   - Business value
   - Technical risk
   - Amount of programming

2. Write tests for the story
3. Automate the tests
4. Implement the functionality

Step 2: Write tests

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

Step 3: Automation

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
4. Implement the functionality

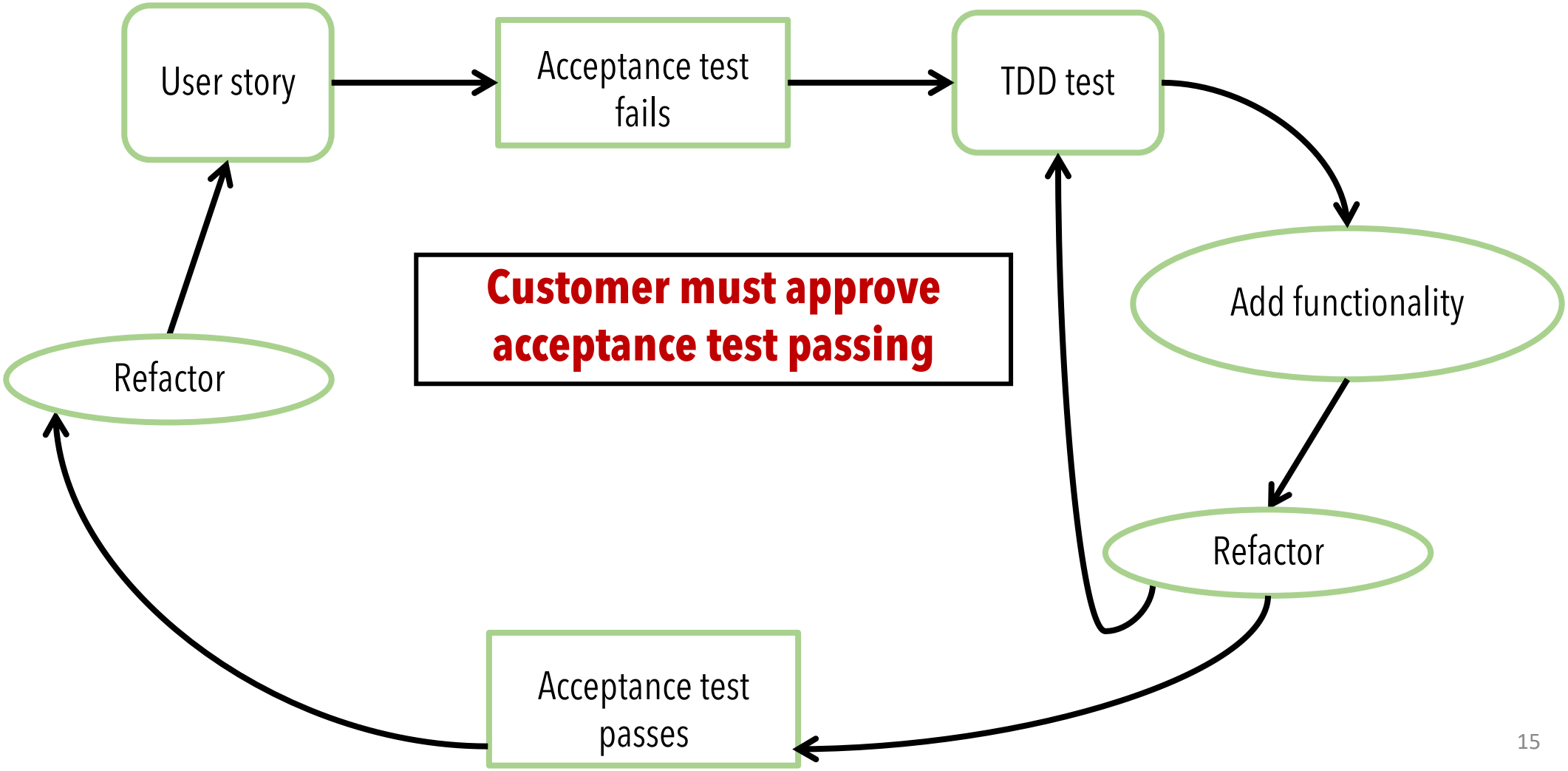
Action	Parameters
Place call	555-1234, account 123456
Accept call	555-1234
Verify text	123456
Verify text	Cory Customer

Step 4: Implementation

The acceptance TDD cycle

1. Pick a story
2. Write tests for the story
3. Automate the tests
4. Implement the functionality
   - This is done using TDD
   - Each A-TDD test leads to multiple small tests
   - As we get small tests to pass, we're closer to A-test passing

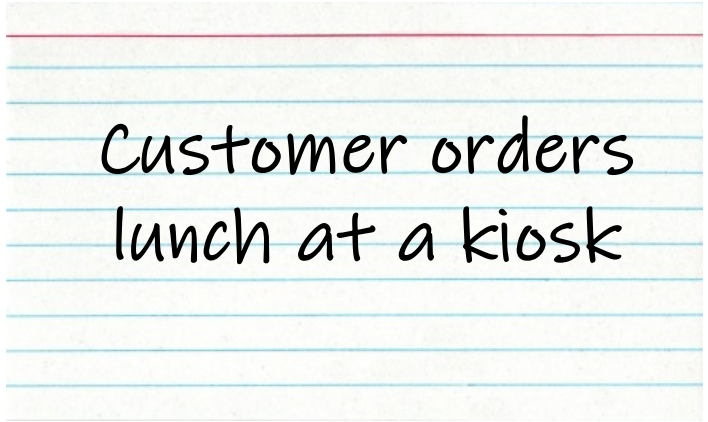
Acceptance tests in agile



In-class Exercise

Write two or three acceptance tests for the following user story

Follow the guidelines in Chapter 9



Acceptance testing as a team activity

Defining the customer role

- Representative of end users
- Possible 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

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 are needed!

What exactly are we testing? (9.6)

Should we test against the UI?

- Do whatever is easier long term
- UIs are often in the way
- Good tools can automate tests through and 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.