

# Test-Driven Tuesday

## Acceptance Testing

# What are Acceptance Tests?

- Unit Tests help to build the system right
- Acceptance Tests help to build the right system

# How does it work?

# It starts with a story....

**As a** microblogger,  
**I want** to find out as soon as others mention me,  
**So that** I can choose to respond while topics are current.

# It starts with a story....

**As a** microblogger,  
**I want** to find out as soon as others mention me,  
**So that** I can choose to respond while topics are current.

- A User Story is a “letter of intent” for developers and customers to work on a problem together.  
It’s a commitment to a future conversation.
- The result of this conversation is a shared understanding of the expected behaviour of the system, expressed using Acceptance Tests.  
Those form a “contract”.

# Acceptance Tests

As a  
I wa  
So t

**Feature:** Notifies microbloggers when they're mentioned

As a microblogger,  
I want to find out as soon as others mention me,  
So that I can choose to respond while topics are current.

**Scenario:** Notifies logged-in microbloggers using web UI

**Given** I'm logged in as '@JanMolak'

**When** someone mentions me in their post

**Then** I receive a notification

**Scenario:** Notifies registered microbloggers by email

**Given** I'm registered as '@JanMolak'

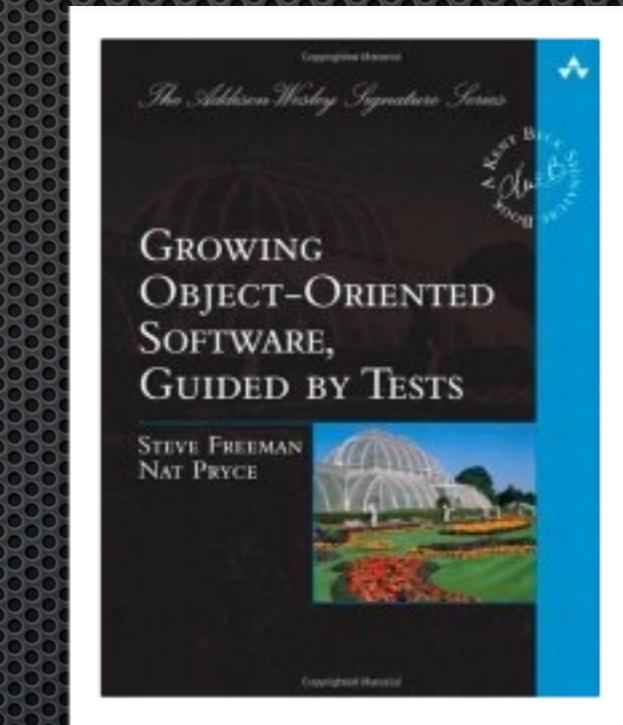
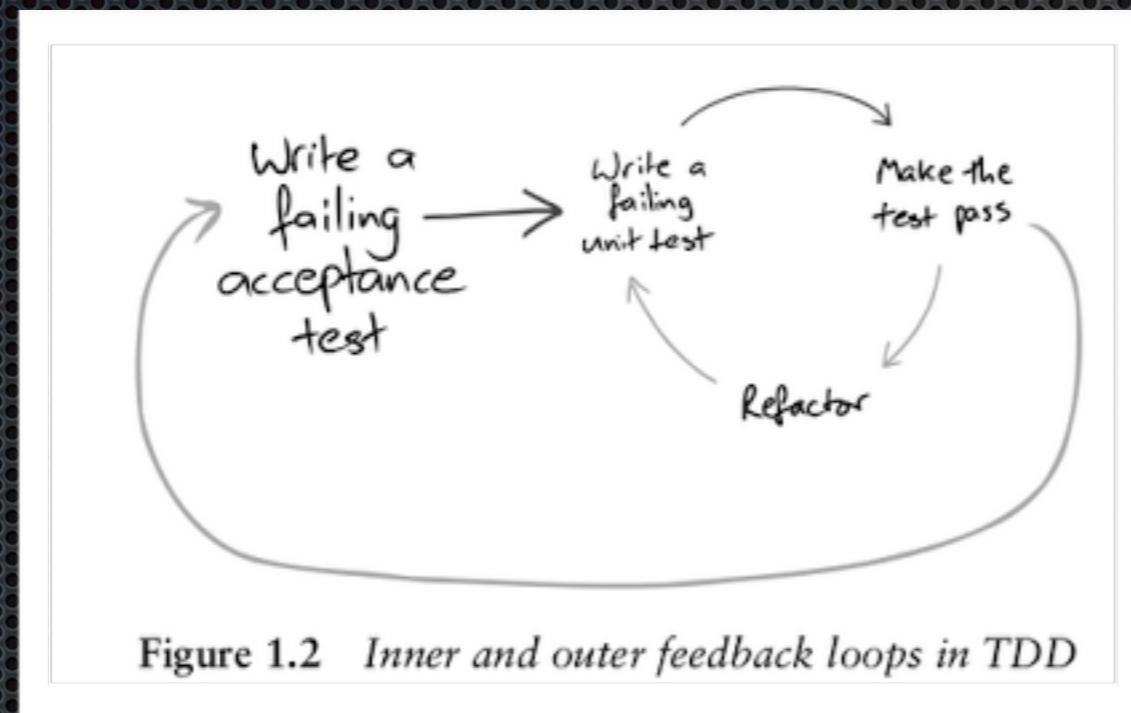
**When** someone mentions me in their post

**Then** I receive a notification by email

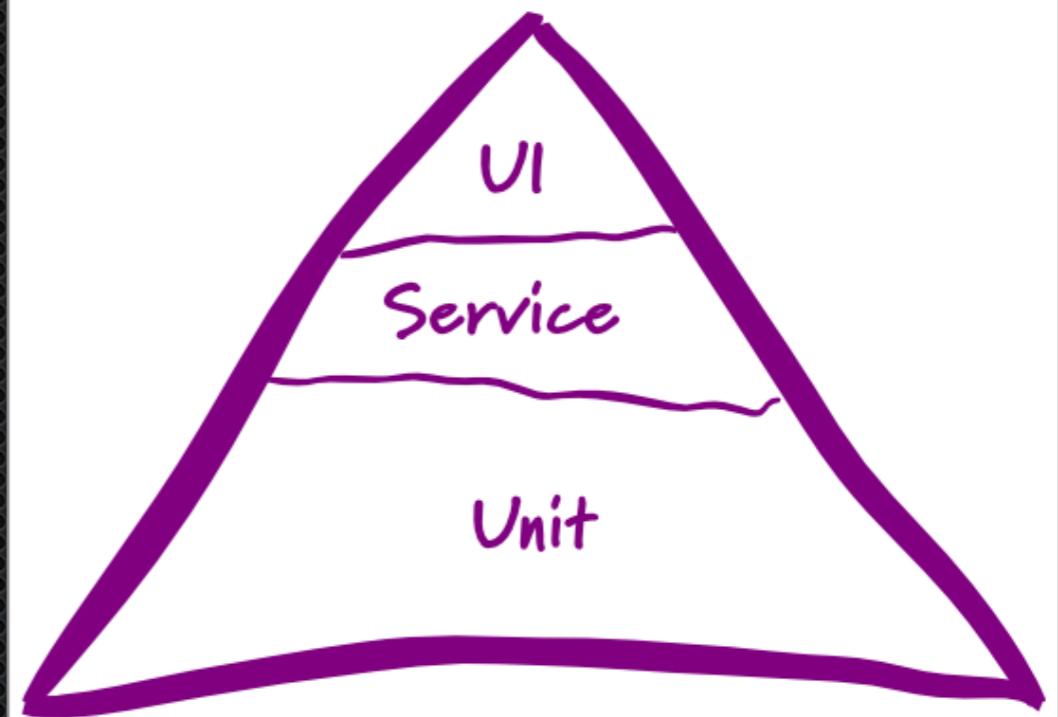
# Why is Acceptance Testing important?

- Helps capture user requirements as they change and evolve.  
And they will. They always do ;-)
- Automated Acceptance Tests can be validated directly:  
If the test passes the system meets user requirements  
specified in the test.
- Gives system coverage, helps find bugs unit tests could miss.
- Tells you when you're done - you're done when the system is  
ready for release, it's ready for release when acceptance tests  
deemed by the customer "must-have" pass

# Can acceptance tests drive the design process?



Do they have to  
interact with  
the User Interface?



# Do they have to be written in English?

```
test
public void displays_potential_failure_cause() {
    given.I(
        defineABuildFailureCause(
            "Shell script failure",
            "A shell script failure happened",
            "Build step 'Execute shell' marked"
        ),
        createABuildMonitorView("Build Monitor"),
        that(includesAllTheJobs())
    ).

    When IHave(
        aFreestyleProject("example-acceptance")
        .configuredToRun(aFailingShellScript)
        .executed()
    );

    // then ...
    assertThat(
        buildMonitorView("Build Monitor")
        .possibleFailureCause(),
        is("Shell script failure")
    );
}
```



# Coding Dojo

<https://github.com/jan-molak/animal-quiz>