

INF 117 Project in Software Engineering

Lecture Notes ~Spring Quarter, 2008

Michele Rousseau

Set 5 - Presentations, Peer Evals, Test Plans

What's Next APRIL 2008

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
13 Week 3	14	15 Taxes Due	16 Stud. Pres-Des Order 1,2,3,4 Team App: #1 Peer Eval: #1	17 Req. Iter: #2 Test-Plan It. #1	18 Team Log #1	19
20 Week 4	21	22 Earth Day	23	24 Req. Iter: #3 Des. Iter: #1 Project Plan #2	25 Cust. Milestone Req. Approved	26
27 Week 5	28	29	30 Stud. Pres-Des Order 2,3,4,1 Team App: #2 Peer Eval #2	1 Des. Iter: #2 Test-Plan It #2 (Incl Des)	2 Team Log #2 Course Log #1	

Set 5

2

Announcements

- ↳ Should send Reqs to client for feedback
 - Give them time to review things
 - Don't assume they will always be available
- ↳ Don't have to use templates provided
 - But... use some template
 - Why is the template you picked the best?
- ↳ Iteration #2 of Reqs should be complete

Set 5

3

Today's Class

- ↳ Presentations
- ↳ Peer Evaluations
- ↳ Acceptance Test Plans

Set 5

4

Presentations

- ↳ Time
 - 10-12 Minutes
 - + 5 Min Q/Comments
- ↳ Format
 - Project Plan
 - Introduction
 - Describe what is happening in that phase

Set 5

5

Peer Evaluations

- ↳ For each presentation
- ↳ Evaluate on
 - Technical Aspects
 - Style
 - ▣ How are the slides?
 - ▣ How was the presentation overall?
 - ▣ How was the presenter?
 - Project, Process, Progress
 - ▣ Overall assessment of their project
 - ▣ Are they following a good S/E Process?
 - ▣ Are they sticking to their project plan?

Set 5

6

Acceptance Test Plans

Testing at the requirements level

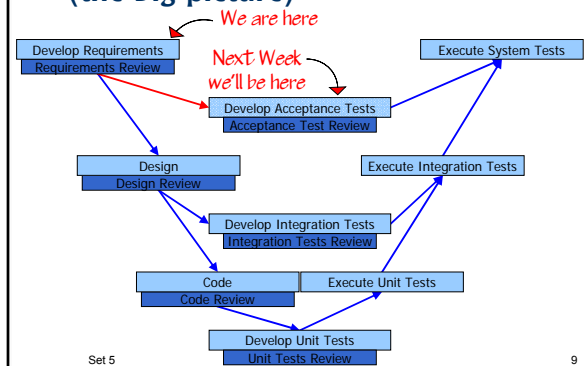
Requirements Qs to Answer

- K Is the requirements specification *complete*?
- K Is each of the requirements *understandable*?
- K Is each of the requirements *unambiguous*?
- K Are any of the requirements *in conflict*?
- K Can each of the requirements be *verified*?
- K Are all terms and concepts *defined*?
- K Is the requirements specification *unbiased*?

Set 5

8

V-Model of Development & Testing (the big picture)



Set 5

9

Verification & Validation

K Verification

- “Are we building the product right?”

K What about validation?

- “Are we building the right product?”
- Ensure software meets customer’s intent
- External consistency

K How can we do it?

- Prototypes
- Observing the customers

Set 5

10

Acceptance Test Plans

a.k.a. System Test Plans

- K Tests the system from a user’s perspective
 - Is this white or black box testing?

- K What happens if your system does not pass one of these tests?

- K Should have SPECIFIC TESTS

- K What are you testing it and how...
 - What is your approach?

Set 5

11

Acceptance Test Plan

- K Accompanies a requirements specification

- K Specifies → in an operational way

- Validates consistency between req. spec. & the system
- Captures user requirements in a *verifiable* way

- K Exposes problems that unit tests miss

- K Binds a customer to accept the delivered system

- if it passes all the tests
- Provides a definition of how “done” the system is

- K Covers all aspects of the requirements specification

Set 5

12

Motivation

- Test planning reveals ambiguities & defects early
 - Many defects originate in requirements phase
 - Much less costly if caught early
 - Must verify requirements document
- System/acceptance testing
 - Specify what is required of the system
 - Based on scenarios/flows

Some argue Acceptance Tests are more important than the spec.

Set 5 13

Where Use Cases Are Used

- Requirements:
 - Collect
 - Clarify
 - Validate
- Analysis & design
 - Object modeling
 - Interface design
 - Object interaction diagrams
- Test
 - Verify needs are met

Use cases are often used to describe possible user interaction (input)

Set 5 14

Use Case Based Testing

- Develop a naming convention
 - Match the test to the use case to test cases
 - Name the actors
- For each test case
 - Identify the actors involved
 - Identify the use case it covers
 - Identify pre-reqs and inputs
 - Remember to test the boundary cases!
 - Define expected output
 - Based on the scenarios!

Set 5 15

Use Case Based Testing

- There is often tool support
 - Automatically generate test suites with specified coverage criteria
- Enhance use case diagrams with
 - Inputs from actors
 - Output to the actors
 - How the system's state changes
 - Can also describe flows between use cases
- Useful for integration, system testing

Set 5 16

ReadySET

- Open source S/E tools
 - <http://www.tigris.org>
 - <http://requirements.tigris.org>
 - <http://readyset.tigris.org>

Set 5 17

ReadySET: Test Case Format

Unique test case ID: Test Case Title	
Purpose	Short sentence about the aspect of the system
Prereq	Assumptions that must be satisfied prior to running the test case
Test Data	List of variables and their values. Can be exact values or ranges.
Steps	Steps to carry out the test (brief list here but detailed explanations below)
Notes and Questions	Extra notes or questions

Set 5 18

ReadySET: Example Steps

K Login Test

1. visit LoginPage
2. enter userID
3. enter password
4. click login
5. see the terms of use page
6. click agree radio button at page bottom
7. click submit button
8. see PersonalPage
9. verify that welcome message is correct

Set 5 `username` 19

ReadySET: Example Steps

K login [as ROLE-OR-USER]

- Log into the system with a given user or a user of the given type. Usually only stated explicitly when the test case depends on the permissions of a particular role or involves a workflow between different users.

K visit LOCATION

- Visit a page or screen. For web applications, LOCATION may be a hyperlink. The location should be a well-known starting point (e.g., the Login screen), drilling down to specific pages should be part of the test.

Set 5 20

ReadySET: Example Steps

K enter FIELD-NAME [as VALUE] [in SCREEN-LOCATION]

- Fill in a named form field. VALUE can be a literal value or the name of a variable defined in the "Test Data" section. The FIELD-NAME itself can be a variable name when the UI field for that value is clear from context, e.g., "enter password".

Set 5 21