Risk-Based Testing & Test-Driven Development

Minsoo Ryu

Hanyang University
What is Risk?

- A factor that could result in a future negative consequence; usually expressed as impact and likelihood (ISTQB Glossary)
Risk-Based Testing

- Identify risks
  - What the risks are
  - Where they are

- Analyze the risks
  - Impact
  - Likelihood

- Build and perform tests
  - Based on the risk analysis

- Monitor and report
Why Risk-Based Testing?

- Because there is **never enough time to test everything**
- Test is at the end of the project (“caboose effect”)
- There are always compromises about testing
- The task of testing is to measure and fight risk

- Test shall find **IMPORTANT** defects
  - It means important enough to fix them, i.e. they must belong to important risks
Risk Identification

- Risk identification involves collecting information about the project and classifying it to determine the amount of potential risk.

- The risk could be related to system complexity, new technology or methodology involved that could cause problems, limited business knowledge or poor design and code quality.
  - Brainstorming is an effective way for risk identification.
  - Maximum number of appr. 35 risk items.

<table>
<thead>
<tr>
<th>Risk item 1</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Risk item 2</td>
<td>Security</td>
</tr>
<tr>
<td>Risk item 3</td>
<td>Functionality</td>
</tr>
<tr>
<td>Risk item 4</td>
<td>Interoperability</td>
</tr>
</tbody>
</table>
Risk Analysis

- Risk = impact x likelihood
  - What is the impact for the business?
  - What is the likelihood that there are defects?

- Determine factors based on previous projects, e.g. defect patterns
Risk Analysis

- Likelihood
  - Complexity
  - New development (level of re-uses)
  - Interrelationship (# interfaces)
  - Size
  - Technology
  - Geographical spread
  - Inexperience (of development team)

- Impact
  - User importance (“selling item”)
  - Financial (or other)
  - Damage (e.g. safety)
  - Usage intensity
  - External visibility
  - Cost of rework
Stakeholder Involvement

- Stakeholder identification
  - Internal (likelihood) and external (impact)
- Individual stakeholder scoring
- Consensus meeting

<table>
<thead>
<tr>
<th></th>
<th>Likelihood</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Complexity</td>
<td>New development</td>
</tr>
<tr>
<td>Item 1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Item 2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Item n</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Risk Matrix

Focus on Development Testing

Focus on Acceptance Testing

Likelihood

Impact

STA (Severe Test Area)
STTA (Strong Test Area)
ITA (Intensive Test Area)
FTA (Fundamental Test Area)
Low Level Test Example

- Focus on Development Testing

![Diagram showing test coverage and inspection methods.]

- Statement Coverage 70%
  - Pair Inspection
  - Boundary Test
  - Full Code Inspection

- May not be tested
  - Statement Coverage 70%
High Level Test Example

- Focus on System/Acceptance Testing

Use Case Testing (main only)
Equivalence Partitioning

Use Case Testing (main & alternative)
Decision Table Testing
Boundary Analysis

Use Case Testing (alternative only)

Use Case Testing (main & alternative)
Equivalence Partitioning
TestLink

- Test Case Management
Redmine

- Project management web application
  - Test Scheduling and Management
Bugzilla

- Bug tracking system

![Bugzilla interface]

- Reporter: mizusuru@soberit.hut.fi
- Version: other
- Product: TestProduct
- Component: TestComponent
- Platform: PC
- Priority: FS
- OS: Windows 2000
- Severity: HTML
- URL: http://
- Summary: Splash screen has typo "Copyrigt" should be "Copyright"
- Description: Splash screen displays text "Copyright ...", should be "Copyright"

Only users in all of the selected groups can view this bug:
(Learn more about checking to make this a public bug)
- testing group

Commit  Remember values as bookmarkable template
Test-Driven Development

Minsoo Ryu

Hanyang University
What is TDD?

- Test-driven development (TDD) is a software development process that relies on the repetition of a very short development cycle:
  - first the developer writes a failing automated test case that defines a desired improvement or new function,
  - then produces code to pass that test and finally refactors the new code to acceptable standards

- Kent Beck, who is credited with having developed or 'rediscovered' the technique, stated in 2003 that TDD encourages simple designs and inspires confidence
What is TDD?

- Test-driven development is related to the test-first programming concepts of extreme programming, begun in 1999, but more recently has created more general interest in its own right.

- Programmers also apply the concept to improving and debugging legacy code developed with older techniques.
Requirements

- Test-driven development requires developers to create automated unit tests that define code requirements (immediately) before writing the code itself
  - The tests contain assertions that are either true or false
  - Passing the tests confirms correct behavior as developers evolve and refactor the code

- Developers often use testing frameworks, such as xUnit, to create and automatically run sets of test cases
Test-Driven Development Cycle
Write a Test

☐ In test-driven development, each new feature begins with writing a test
  ▪ This test must inevitably fail because it is written before the feature has been implemented
  ▪ If it does not fail, then either the proposed “new” feature already exists or the test is defective

☐ To write a test, the developer must clearly understand the feature's specification and requirements
  ▪ The developer can accomplish this through use cases and user stories that cover the requirements and exception conditions
  ▪ This could also imply a variant, or modification of an existing test
  ▪ This is a differentiating feature of test-driven development versus writing unit tests after the code is written: it makes the developer focus on the requirements before writing the code, a subtle but important difference