



# **Risk Based Testing**

Why We Need It & How To Do It

# Agenda

---

- What is Risk Based Testing?
- Functional Testing Total Coverage
- Risk Based Functional Testing
- Risk Based Non-Functional Testing

# What is Risk Based Testing?

---

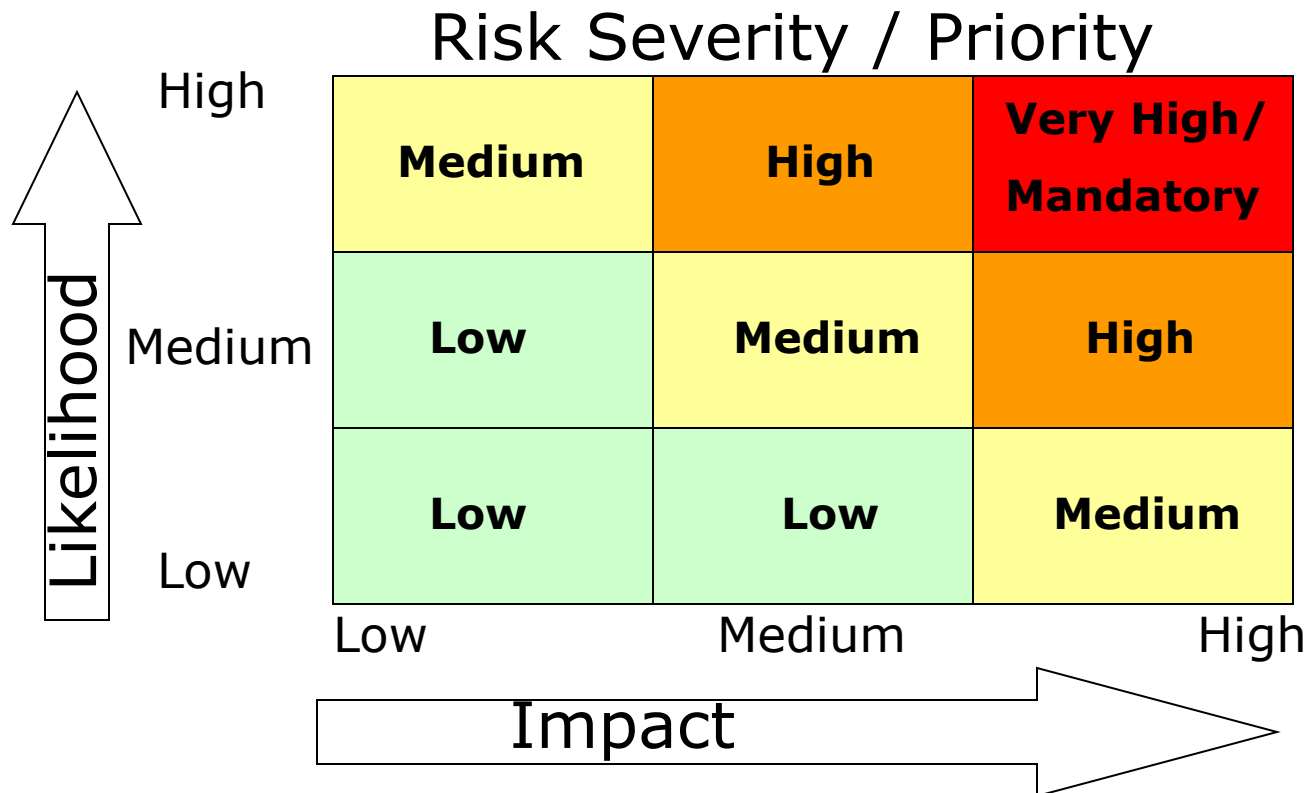
- A testing strategy to determine what to test, when, and how much, based on risk
- An opportunity to negotiate what “quality” means, and the acceptable level of quality for a project
- A way to confirm that risk remaining after testing is visible and acceptable to the business

# What is Risk?

Risk = Cost of Failure

= Likelihood of Occurrence \*

Impact of Occurrence



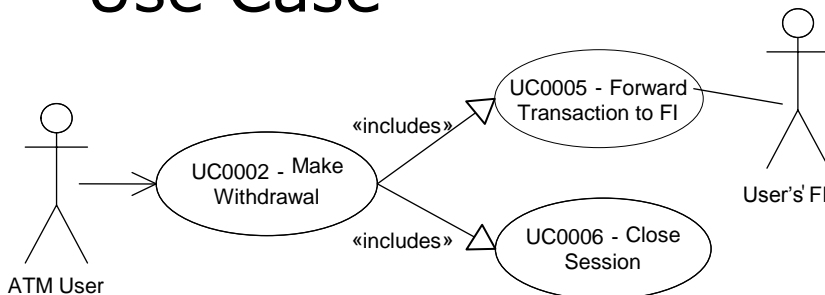
# Functional Testing Total Coverage

## How To Do It

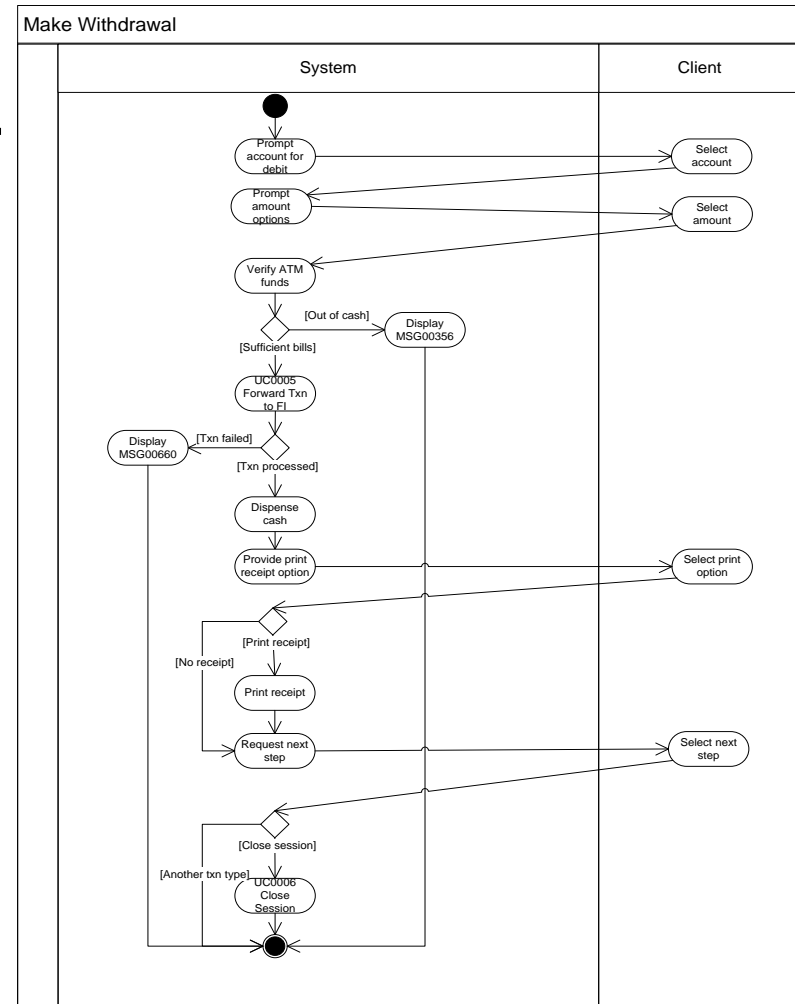
At least 1 test case for

- Every use case
- Every alternate path
- Every option
- Every error condition

## Use Case



## Activity Diagram



# Functional Testing Total Coverage

## Advantages

- Rigorous approach
- Complete coverage
- No surprises

## Disadvantages

- Large effort
- Long duration
- High cost

# Functional Testing Total Coverage – Example 1

Scenario: There are 10 account types

- For each account type there are 6 valid options
- For each valid option there are
  - 1 “happy path” and
  - 5 alternate / error paths

**Question 1: How many test cases do we need to write and run to achieve functional test total coverage?**

**Answer 1: Number of test cases =  
10 account types \* 6 valid options \* (1 + 5) paths = 360 test cases**

**Question 2: With 2 testers each able to run 8 test cases a day, how long will testing take?**

**Answer 2: Number of days =  
360 test cases / (2 testers \* 8 test cases / day) = 22.5 days**

# Functional Testing Total Coverage – Example 2

Scenario: There are 10 account types

- For each account type there are there are 6 valid options
- For each valid option there are
  - 1 “happy path” and
  - 5 alternate / error paths
- One set of code addresses 6 of the 10 account types
- Another set of code addresses the other 4 account types

**Question 1: How many test cases do we need to write and run to achieve functional test total coverage?**

**Answer 1: Number of test cases =**

$$\begin{aligned} & \mathbf{10 \text{ account types} * 6 \text{ valid options} * (1 + 5) \text{ paths}} \\ & \mathbf{= 360 \text{ test cases}} \end{aligned}$$

# Functional Testing Total Coverage

---

## Current Pressures

- Do it faster
- Spend less
- Do less
- Get it done on time
- Don't miss anything!
- Get it right!!

# Risk Based Functional Testing

---

How to do it

- Assign to each test case priority based on risk:  
Mandatory, High, Medium or Low
  - New product / platform / device / function / process
  - Impact / visibility: mobile banking vs. new layout
  - Previous initiatives' issues / defects
  - Previous testing phase experience / defects
- Schedule test case execution in priority order
  - Minimize exposure from curtailed schedule
  - Enable logical assessment of exposure
  - Make contingency plans; e.g., more testing during performance or business acceptance testing phases
  - Accept outstanding risk tradeoff
- Review and revise as conditions change

# Risk Based Functional Testing

---

## Advantages

- Systematic approach
- Accommodates time constraints
- May reduce effort, cost, duration
- Enables sizing remaining risk at any point

## Disadvantages

- May leave undetected defects

# Risk Based Functional Testing

---

Who determines what to test when?

- Overall Test Coordinator
- Test Phase Leads
- Information Sources / Reviewers
  - Risk Analysis => Key Issues / Concerns
  - Business Analysts & Subject Matter Experts => Detailed Business / System Requirements => Testing Requirements
  - Developers / Architects => Interdependencies
  - Business Stakeholders => Acceptable Risks

# Risk Based Functional Testing

---

## Process

- **Confirm Requirements – Detailed Requirements, Use Cases, Data Definitions, Error Messages, User Interface Design, etc.**
- **Define Testing Requirements**
  - **Requirements – Test Case traceability**
  - **Agree on test case priorities**
  - **Use project targets / constraints to set scope**
  - **Get approval of scope / omissions**
- **Maintain planned test cases based on risk**
  - **Reflect Change Requests**
  - **Reflect previous test phase results**
  - **Reflect previous project results**
- **Execute from high to low priority where feasible**
- **Quantify remaining risk before exit approvals**

# Risk Based Functional Testing – Example 1

**Scenario: There are 10 account types**

- For each account type there are there are 6 valid options
- For each valid option there are 1 “happy path” and 5 other paths
- One set of code addresses 6 of the 10 account types
- Another set of code addresses the other 4 account types

**Question 1: How would we select appropriate number of test cases to minimize risk?**

**Answer 1: Number of test cases =**

**For 1st 6 account types: (6 valid options \* 6 paths) +**

**For last 4 account types: (6 valid options \* 6 paths)**

**= 72 test cases**

**with variety of test account types proportional to number of production account types**

**Question 2: How long would testing take with 2 testers each able to execute 8 test cases per day?**

**Answer 2: Duration =**

**72 test cases / (2 testers \* 8 test cases / day) = 4.5 days**

# Risk Based Functional Testing – Example 2

**Scenario: There are 10 account types**

- For each account type there are there are 6 valid options
- For each valid option there are 1 “happy path”+5 other paths
- One set of code addresses 6 of the 10 account types, and 90% of the accounts are in this group
- Another set of code addresses the other 4 account types; and only 10% of the accounts are in this group

**Question 1: How would we select appropriate test cases to minimize risk if we had only 3 days to test and 2 testers each able to execute 8 tests per day?**

**Answer 1: In 3 days with 2 testers we can execute**  
 **$3 \text{ days} * 2 \text{ testers} * 8 \text{ test cases} / \text{day} = 48 \text{ test cases}$**   
**Run (90% of 48) or 40 - 45 test cases addressing the first set (90%) of accounts**  
**Run (10% of 48) or 3 - 8 test cases addressing the second set (10%) of accounts**

# **Risk Based Functional Testing – Example 2**

**Scenario: There are 10 account types**

- **For each account type there are there are 6 valid options**
- **For each valid option there are 1 “happy path”+5 other paths**
- **One set of code addresses 6 of the 10 account types, and 90% of the accounts are in this group**
- **Another set of code addresses the other 4 account types; only 10% of the accounts are in this group**

**Question 2: If 48 test cases cannot provide confidence that we have done sufficient testing / addressed risk appropriately, what do we do?**

**Answer 2: Lobby for additional testing time / resources and back up your concerns**

- **Identify / quantify the exposure (risk) from not executing more test cases**
- **Provide options / recommendations for approach to take**
- **Argue from the perspective of the decision makers**
- **Accept business decision and provide maximum value feasible**

# **Risk Based Non-Functional Testing**

---

- **Apply similar logic to non-functional testing based on risk**
- **Non-functional categories of testing include**
  - **Content**
  - **Usability**
  - **Efficiency**
  - **Reliability**
  - **Installability**
  - **Maintainability**
  - **Portability**
  - **Documentation**
  - **Platform**
  - **Performance**
  - **Capacity****and more**

# **Content Testing Risk Considerations**

---

- **How major are the content changes?**
  - **Complete redesign**
  - **A few words and pictures on a page**
- **How visible are the content changes?**
  - **Intranet pages used rarely; e.g., annual tax forms**
  - **Intranet pages used regularly; e.g., employee benefits**
  - **Intranet pages used heavily; e.g., Online Banking**
- **What is the life of the content changes?**
  - **Short campaign**
  - **Permanent content**
- **How important are the content changes?**
  - **Legal document?**
  - **Sensitive issue?**
  - **Only cosmetic?**

# Usability Testing Risk Considerations?

---

- **Frequency Of Use?**

- **Occasionally**
- **Regularly**
- **Daily**

- **Users?**

- **Small number of internal users**
- **High number of internal users on core function**
- **Small number of VIP customers**
- **Large number of customers**

- **Importance Of Application?**

- **Non-critical function**
- **Core function**
- **VIP customer function**
- **Basic need for everyone**

# Platform Considerations?

---

- **Type of Technology?**
  - **Proven**
  - **Brand new**
  - **History**
- **Time To Resolve Issues?**
  - **Immediate**
  - **Days**
  - **Outages required**
  - **Users Impacted**
- **Support Availability?**
  - **Immediate**
  - **On call with Service Level Agreement**
  - **Best Effort**
- **Importance Of Application**
  - **High**
  - **Low**

# Performance & Capacity Considerations?

---

- **Application Importance?**
  - **High**
  - **Low**
- **Number Of Transactions?**
  - **Low**
  - **High**
  - **Low with occasional peaks**
- **Peak Requirements?**
  - **Daily**
  - **Annually for 3 days**
- **Timing?**
  - **Peak days are next week or 10 months away**
- **Available Capacity?**
  - **Low – currently at limit or High – just upgraded**
  - **Vendor lead time on upgrades**
  - **Cost of upgrades**

# Dealing With Unrealistic Constraints

- Same for Functional and Non Functional Testing
  - Identify and Quantify / Qualify Risks
  - Provide / recommend options
  - Accept business decision
  - Do the most you can with what you have

# Risk Based Testing

---

## Questions & Answers