

# Effective Test Strategies! Are They Meeting Stakeholder Needs?

Clyneice Chaney  
CQMgr/OE, PMP  
Clyneice.chaney@gmail.com

# Test Strategy

What are the most important things to do ?

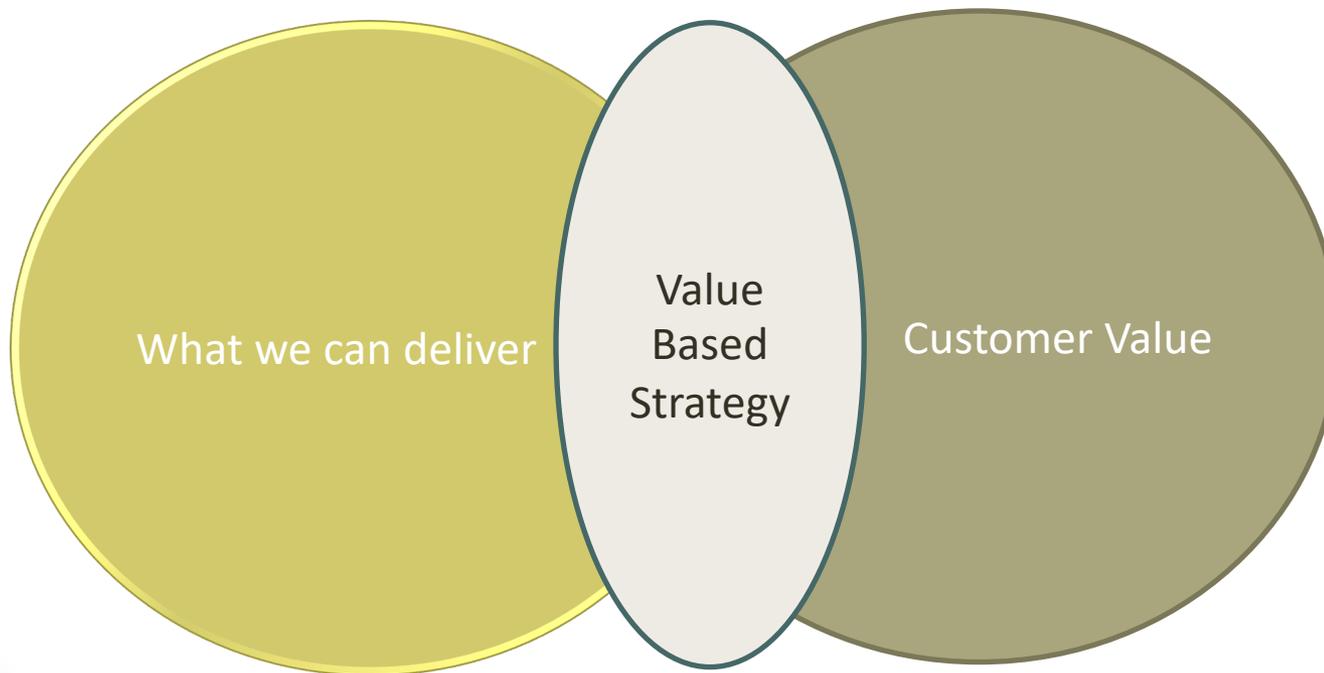
How are we to achieve the goal?

What is necessary to achieve goal?

# Deciding What to Test and How Much



# Do You Know What is Valued?



# Do You Know What is Valued?

## Understanding Your Own & Your Stakeholder's Values

- End user- customer
- Product Business Owner
- Project Manager
- Development Team
- PMO
- Quality Manager

# Using VOC For Insight

- End user- customer
  - *CTQ: Working product*
  - *CTQ: Fast performance*
- Product Business Owner
  - *CTQ:*
- Project Manager
  - *CTQ:*

- Development Team
  - **CTQ:**
  - **CTQ:**
- PMO
  - **CTQ:**
- Quality Director/ Manager
  - **CTQ:**

# What are **YOUR** Value Gains from Testing?

- Testing time to market
- Cost of quality
- First time right index and estimated cost saving
- End user satisfaction levels
- Delivered Defect Ratio
- Defect Predictability
- Test Coverage

# Getting to Good Test Strategies



Testing Techniques

Quality Targets

Product Risks

# Quality Targets

- Pre-negotiated quality targets
- Manage defect rates against the quality target
- Collect metrics
- Change quality targets as needed
- Report quality target compliance
- Include quality targets in project reviews

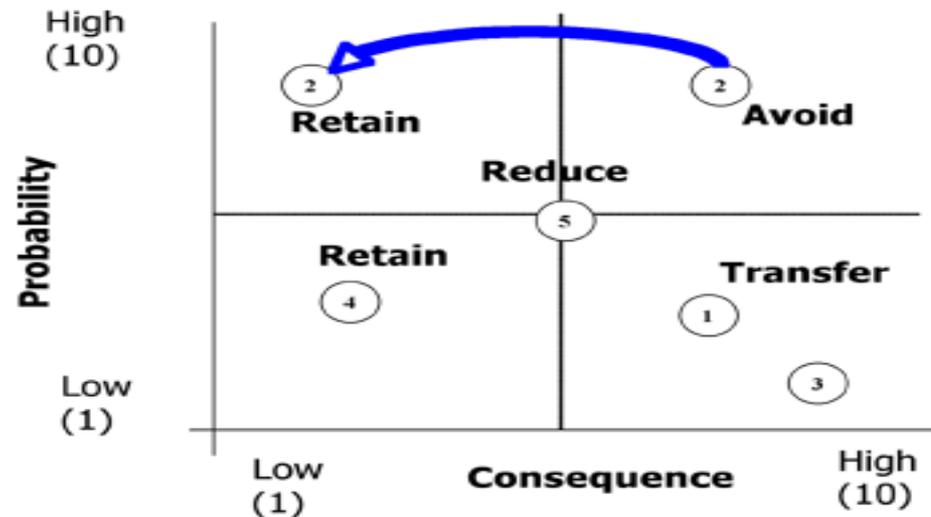
# Quality Objectives

| Attribute       | Description   | Measure and Target  | Priority  |
|-----------------|---|---|-----------|
| Correctness     | Features and functions work as intended   | 100% completion of agreed features<br>Severity 1 defects = 0<br>Severity 2 defects = 0<br>Severity 3 defects < 5<br>Severity 4 defects < 10 | Must Have |
| Integrity       | Ability to prevent unauthorized access, prevent information loss, protect from viruses infection, protect privacy of data entered | All access will be via HTTPS over a secured connection.<br>User passwords and session tokens are encrypted.                                 | Must Have |
| Maintainability | How easy it is to add features, correct defects or release changes to the system.   | Code Duplication < 5%<br>Code Complexity < 8<br>Unit Test Coverage > 80%<br>Method Length < 20 Lines  | Must Have |

# Considering Risks

| Risk of Failure                         | Approach  |
|---|---|
| High Probability/ impact of failure     | How Many Test Cases?<br><br>How Multiple approaches?<br>Coverage? |
| Moderate Probability /impact of failure | Coverage  |
| Low probability /impact of failure      | As needed<br><br>Requirement sampling                             |

# Risk Based Test Strategy



Designed to....

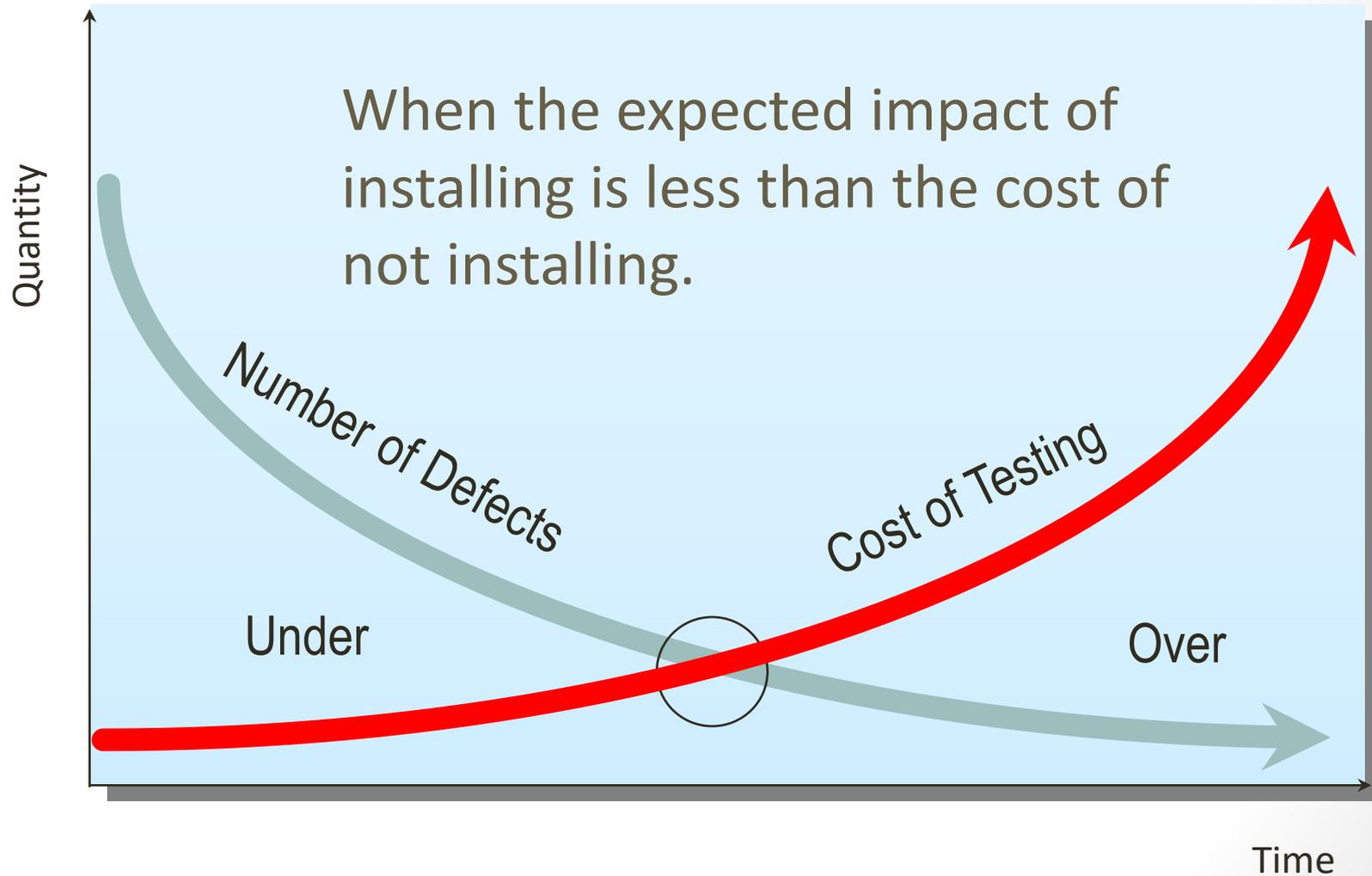
- Identify software product risks of concern
- Specifically address product risks through test activities
- Activities are structured into optimal risk-based test process
- Tests are designed and planned to achieve risk-based test objectives
- Reports are provided for release decision based on risks and benefits
- Incorporate some variations to make use of exploratory or situational reasoning to focus on important but unanticipated problems. Use- exploratory; checklists; scenario; data
- Focus most on areas of identified risks, with some effort on low risk areas .. Just in case

# Coverage

*Is there enough?*

A man in a dark suit and a bowler hat is seen from behind, holding a large black umbrella. The umbrella is open and covers most of the upper half of the frame. The text "Is there enough?" is written in a white, cursive font across the center of the umbrella. The background is a clear blue sky with some light, wispy clouds near the horizon. The overall mood is contemplative and somewhat somber.

# Does Your Strategy Help You to Know When Do To Stop Testing?



# Evaluating Your Strategy

## Test Strategy Heuristics

Is it optimized to find important problems fast, rather than finding all problems with equal urgency

Does it focus most effort areas of potential risks, with some effort on low risk areas

Does it address test platform configuration, how the product will be operated and observed.

Is it diversified in terms of test techniques and perspectives.      Need multiple dimensions of coverage

Does it specify how test data will be designed and generated.

Does it incorporate reasonable variations (exploratory) not all pre-specified.

# Test Strategy Template

## Handout D Risk Response Planning Worksheet

Finding Critical Problems Quickly: Techniques to be used.

Targeted areas & functionality

Area 1:

Why?

How Tested? Technique/ Tool

Targeted areas & functionality

Area 2:

Why?

How Tested? Technique/ Tool

Targeted areas & functionality

Area 3:

Why?

How Tested? Technique/ Tool

Test Types:

When Used?

Test phases:

# Sample Test Strategy

- **Test Phase/Activity**
- Event Management Operational Testing
- **Description:**
- Operational testing of event management activities is designed to provide confidence that event management process is executed and performs within require service levels.
- 
- Event management operational testing will occur in three phases:
- Phase 1 verifying installation and availability of all hw/sw and procures
- Phase 2 verify the operational interaction with other services
- Phase 3 verifies event management performance in a variety of scenarios
- 
- **Object Under Test:**
- Event Management Service
- **Test Objective:**
- Verify that hw/ sw and procedures that are necessary for event management services are:
- Event detection - verify that the monitoring tool has been configured correctly and is detecting the expected events.
- Event detection - verify that thresholds are configured and set appropriately, so as to increase the probability that action can be taken before an incident is generated.
- Event response - verify that events are responded to correctly and efficiently.
- Identify any process issues or bottlenecks; are processes meeting their Key Performance Indicators (KPIs)?
- **Phase Timeline/ Schedule**
- Phase 1--<dates> time period
- Phase 2 <dates> time period
- Phase 3 <dates> time period
- **Perquisite:**
- Event Management processes documented, including identification of when an event-driven alert notification results in an incident
- Service Level Agreements established
- Event Management tool configured for new service
- List of events that signify normal operation; list of events that signify unusual events but represent no exception to normal operation; list of events that signify exception to normal operation and require corrective actions
- Event Management training completed
- Event Management staff's accounts and licenses ready on Event Management tool
- Lists of events to be simulated

# Sample Test Strategy cont...

- **Success Criteria**
- Target service levels are being met, per SLAs:
- Event detection: \_\_\_ (95%?) of events detected and notified correctly
- Event response: \_\_\_ (95%?) of events responded to accurately and efficiently
- Other
- Event Management procedures being followed
- **Test Environment: Setup Requirement:**
- Procedures ready and available for user
- Team trained/ familiar with procedure manual
- Ability to set up triggered events <This would need more extensive information which I don't really have at this point>
- **Measures/data collection:**
- Number of events by category
- Number of events by significance
- Number and percentage of events that required human intervention and whether this was performed
- Number and percentage of events that resulted in incidents or changes
- Number and percentage of events caused by existing problems or known errors
- Number and percentage of repeated or duplicated events
- Number of events/alerts generated without actual degradation of service/functionality
- Response time to events and event completion rates
- Percentage of Event Management staff with appropriate support skills
- 
- **Execution Strategy**
- 
- **Management: Signoff**
-

# Not Enough Time? Jump Start A Risk Based Test Strategy

- Which functionality is most important to the project's intended purpose?
- Which functionality is most visible to the user?
- Which functionality has the largest safety impact?
- Which functionality has the largest financial impact on users?
- Which aspects of the application are most important to the customer?
- Which aspects of the application can be tested early in the development cycle?
- Which parts of the code are most complex, and thus most subject to errors?
- Which parts of the application were developed in rush or panic mode?
- Which aspects of similar/related previous projects caused problems?
- Which aspects of similar/related previous projects had large maintenance expenses?
- Which parts of the requirements and design are unclear or poorly thought out?
- What do the developers think are the highest-risk aspects of the application?
- What kinds of problems would cause the worst publicity?
- What kinds of problems would cause the most customer service complaints?
- What kinds of tests could easily cover multiple functionalities?
- Which tests will have the best high-risk-coverage to time-required ratio?

# Evaluating Your Strategy

## Test Strategy Heuristics

Is it optimized to find important problems fast, rather than finding all problems with equal urgency

Does it focus most effort areas of potential risks, with some effort on low risk areas

Does it address test platform configuration, how the product will be operated and observed.

Is it diversified in terms of test techniques and perspectives.      Need multiple dimensions of coverage

Does it specify how test data will be designed and generated.

Does it incorporate reasonable variations (exploratory) not all pre-specified.

# The End



[Clyneice.Chaney@gmail.com](mailto:Clyneice.Chaney@gmail.com)