Test Management – Leading Your Team To Success

(sample of course slides)

Silverpath Technologies Inc.
Trevor.Atkins@silverpath.com

Thinking
Through
Testing

What is Quality?

Silverpath Technologies

- *"…conformance to requirements: meeting customer expectations, both stated and unstated."* – Philip Crosby, 1979

- *"…the degree to which a set of inherent characteristics fulfill requirements."* – PMI Project Management Body of Knowledge (2008)

- Quality can be:
  - Elegance
  - Correctness
  - Fitness-for-use (Joseph Juran, 1974)

- What is "good enough" quality?

…instead, focus on…

Attributes of Quality Software

Silverpath Technologies

- Satisfaction with the overall quality of the product is usually confirmed through customer surveys:
  - IBM monitors CUPRIMDSO
    (capability / functionality, usability, performance, reliability, installability, maintainability, documentation / information, service, and overall);
  - Hewlett-Packard monitors FURPS
    (functionality, usability, reliability, performance, and service)

– Stephen H. Kan, “Metrics and Models in Software Quality Engineering”

What is Testing?

- *"Testing is the process of trying to discover every conceivable fault or weakness in a work product"* - The Art of Software Testing, Glenford Myers, 1979

- *"Testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results"* - The Complete Guide to Software Testing, Bill Hetzel, 1988

…instead, focus on…

- As Testers, we reduce the likelihood of a serious failure in the field

- As Test Leaders, we give Testers the opportunity to succeed…

Example Testing Challenges

- Iterative project lifecycle
- Evolving product scope
- Arbitrary (unrealistic) ship dates
- Limited testability
- Limited or indefinite requirements
- Uncertain GUI/screens
- Varying expectations from stakeholder to stakeholder

  (Incl. more demanding users in terms of software quality)

- Project Constraints: Resources, Schedule & Budget

Opportunity for Improvement

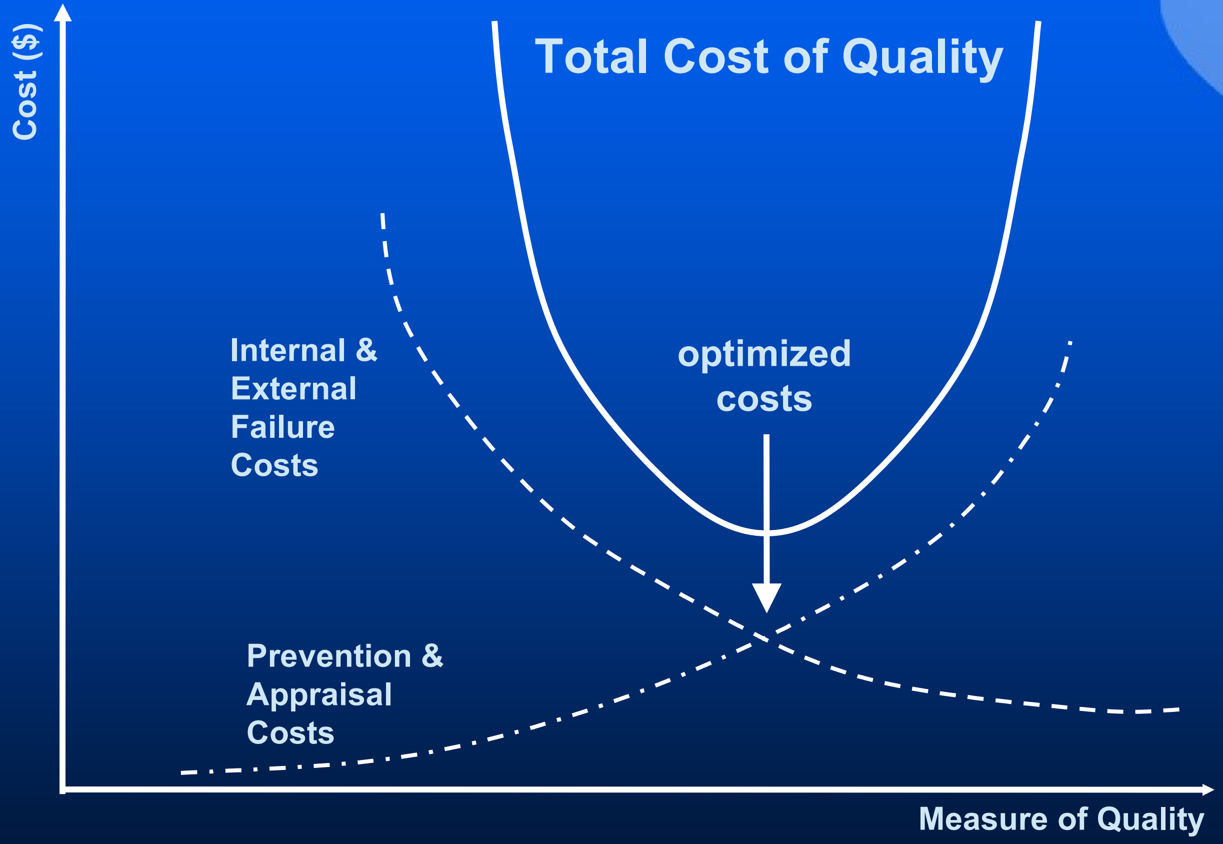
Silverpath Technologies

- According to a National Institute of Standards and Technology study:
  - Software errors cost the U.S. economy an estimated $59.5 billion annually, or about 0.6% of the GDP
  - 80% of the software development costs of a typical project are spent on identifying and fixing defects
  - About 1/3 of these costs, or an estimated $22.2 billion annually, could be eliminated by an improved testing infrastructure

- NIST 2002-10 Report

Optimal Level of Testing?

- Invest in risk mitigation that maximizes quality and ROI



- Continuous Quality Improvement and Outsourcing

The Role of the Test Lead(er)

❖ How do you make testing most effective?
  - ❖ Establish Agreement
  - ❖ Optimize Coverage
  - ❖ Minimize Rework
  - ❖ Mitigate Risk

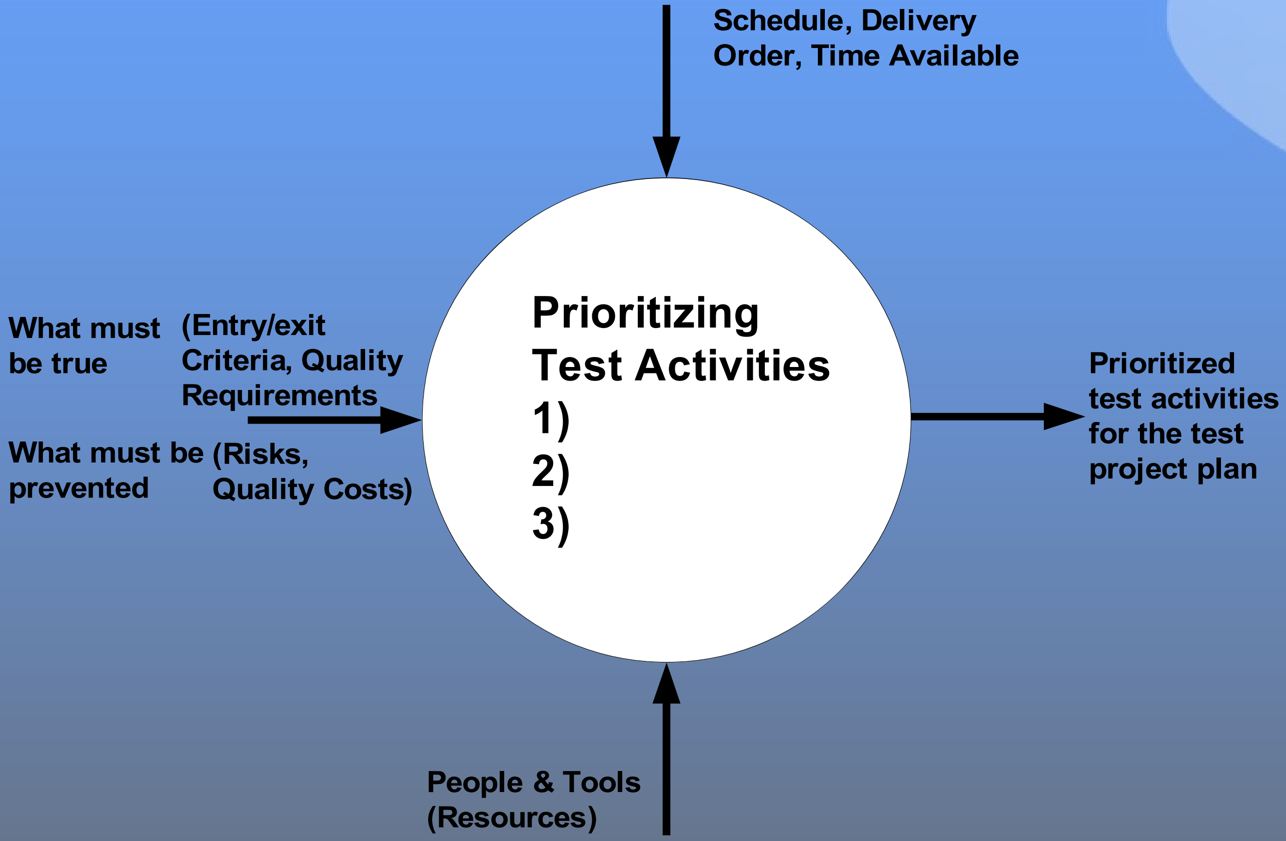
Risk Identification

- Common project concerns
  - Staffing or resource changes
  - Technology changes
  - Company or corporate changes
  - Marketing pressures
  - Over-allocation / Under-utilization of resources
  - Co-contractors or sub-contractors
- Common testing concerns
  - Requirements volatility
  - Requirements complexity
  - Use of new technologies
  - Estimated testability of the design
  - Number of defects corrected in the module since the last calculation
  - Importance of the module relative to the remainder of the project
- Consider risks to quality and the test effort

Deciding Your Testing Approach

- Own Your Approach
  - What are your risks?
  - Know your stakeholders and their expectations
  - Plan your testing by project phase
  - Attain and maintain agreement
- Define Scope & Set Expectations
  - Who will do What, Where, When and Why?
- Project constraints and quality requirements require an organized approach
- What is <u>your</u> “best-fit” test strategy?

Risk-Driven Testing



Schedule, Delivery
Order, Time Available

What must be true (Entry/exit Criteria, Quality Requirements

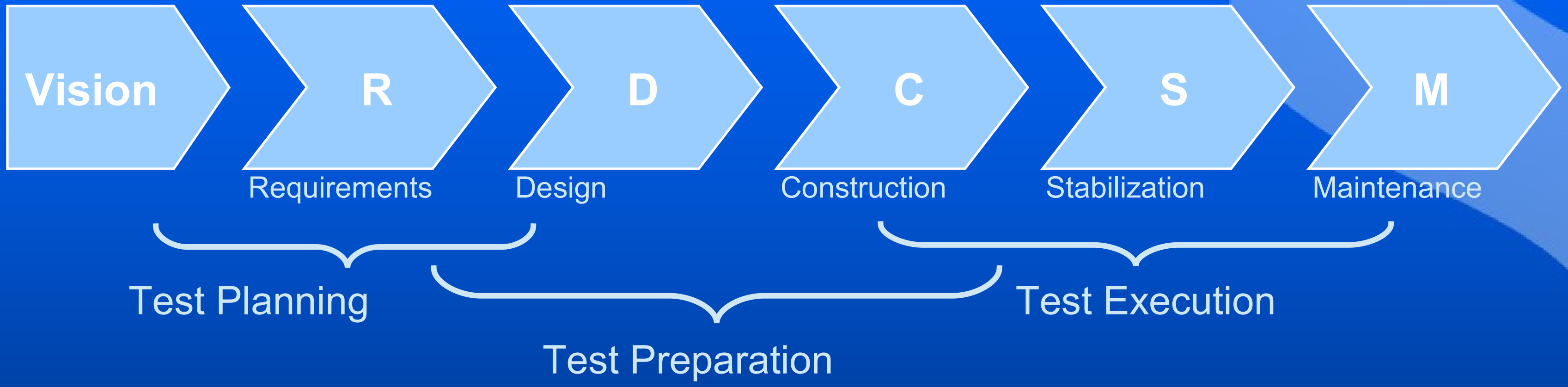
What must be prevented (Risks, Quality Costs)

Prioritizing
Test Activities
1)
2)
3)

Prioritized test activities for the test project plan

People & Tools
(Resources)

Testing Across any SDLC



Vision

R
Requirements

D
Design

C
Construction

S
Stabilization

M
Maintenance

Test Planning

Test Preparation

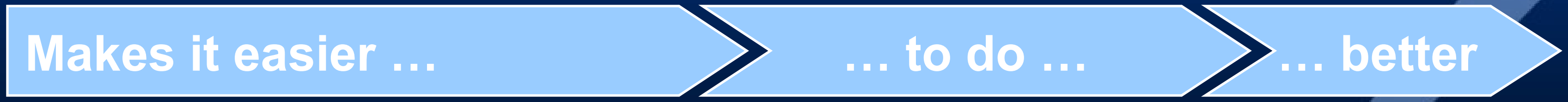
Test Execution

RISK ANALYSIS
EFFORT ESTIMATES
TEST STRATEGY
PROJECT PLAN

REQUIREMENTS REVIEW
TEST DESIGN
TEST DATA CREATION
TEST AUTOMATION

MANUAL TEST EXECUTION
AUTOMATED TESTING
PERFORMANCE TESTING

TEST RESULTS
DEFECT REPORTS
METRICS ANALYSIS



Makes it easier … **… to do …** **… better**

Test Strategy

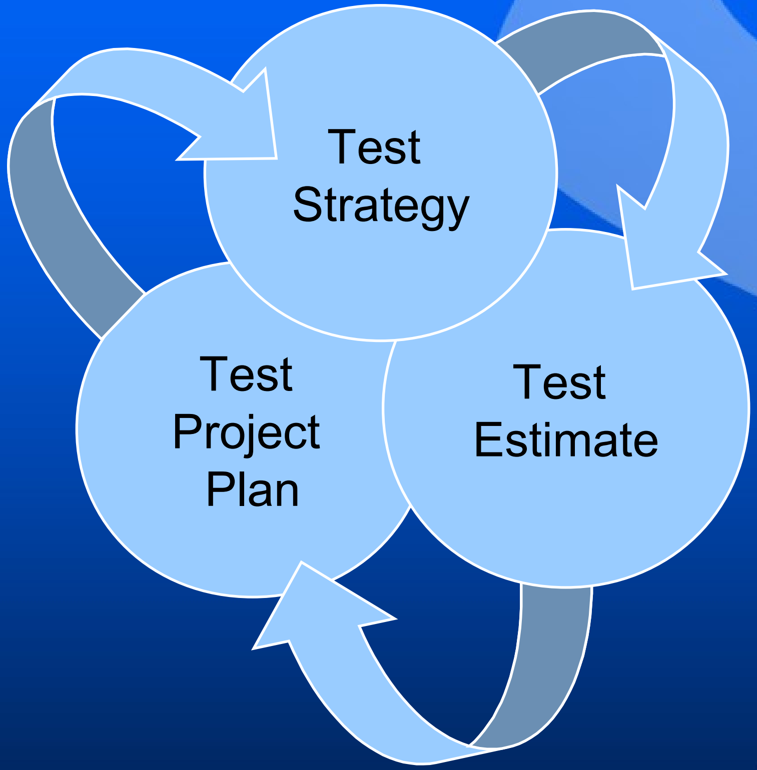
- Provides structure for organizing, scheduling and managing test effort
- Improves communication about test tasks and process
- Facilitates technical tasks of testing

- A test strategy is a high-level document that describes your plan
  - Purpose / scope
  - Quality requirements
  - Assumptions & constraints
  - Test approach
  - Inclusions / exclusions
  - Types of testing
  - Issues & risks

- Get agreement to the plan from stakeholders
- Keep plan & agreement alive as the project advances

Beware Cyclical Dependencies

- Review the project scope and delivery schedule
- Determine the stakeholders in the testing effort and their needs
- Identify the artifacts required to be produced
- Discover your available resources
- Outline a reasonable test strategy
- Distribute for review and acceptance by project team
- Adapt / maintain the plan as the project advances



The cyclical dependencies between the three artifacts requires that a change in any one should result in a review and possible update of the others.

Thinking Through Testing

Discussing "right-fit" approaches for software testing

For our latest updates:

❖ Visit ThinkTesting.com

❖ Follow @ThinkTesting

We are always sharing our ideas on crafting “right-fit” approaches to software testing. We are sure you will find something you can apply to your own projects and organizational environment.

Thinking

Through

Testing