

# The How-To Guide to Test Management

*(What The Heck is Test Management Anyway?)*

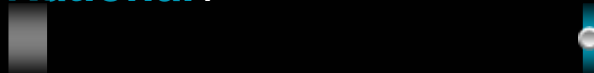
## QM1

*JohnWatkins*  
*IBM Rational Channel Technical Sales*  
*johnwatk@uk.ibm.com*

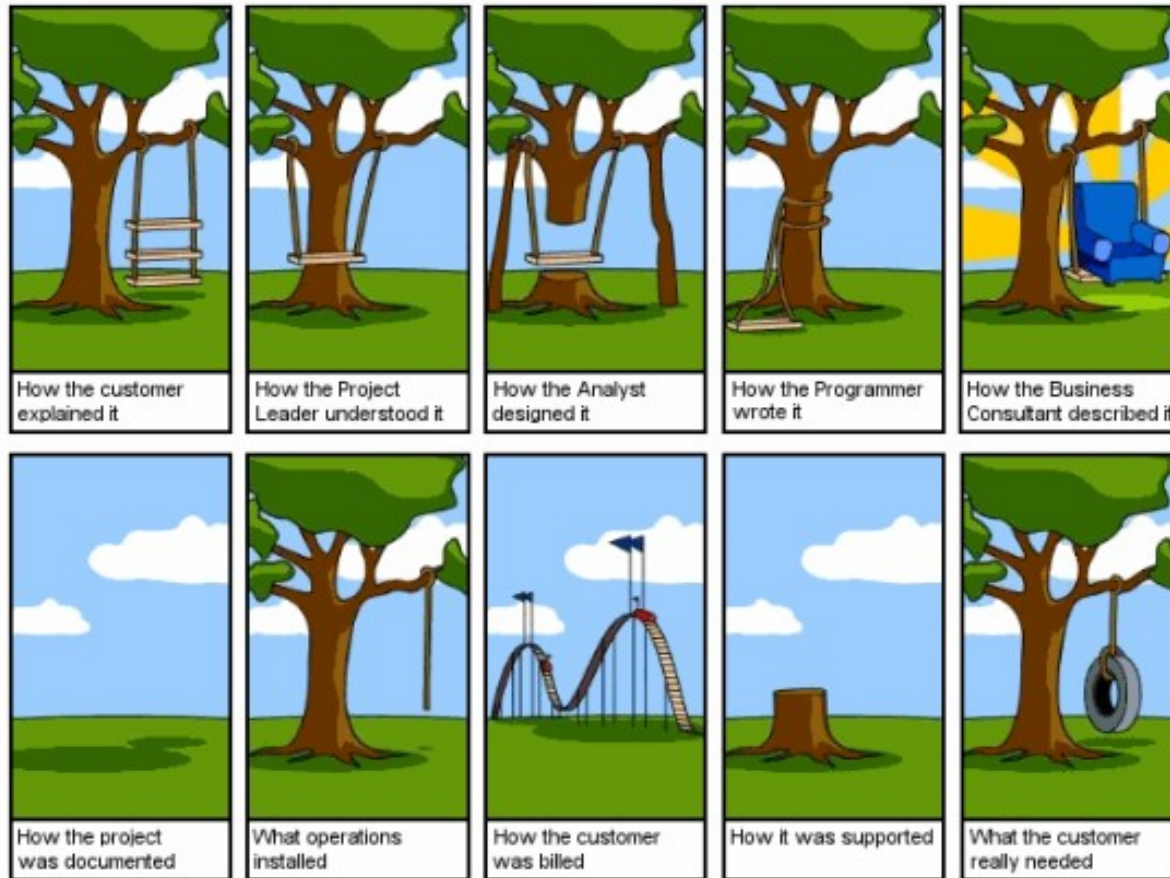
IBM Rational Software Development Conference 2007



▶ What keeps me **Rational**?



# Quality is a matter of perspective



- Customer perspective matters the most



## In the Beginning.....

- ▶ **Brief History of Test Management Solutions**
  - **Pure Manual Effort**
    - Test plans documented with paper and pen
    - IEE 829 created
    - Quality part of ISO 9000 efforts
    - Manually captured results
  - **Electronic Documentation**
    - Word Star/WordPro
    - Frame Maker
    - MS Word



## In the Beginning.....

- Test Automation, Record and playback
  - Many scripts
  - No means to manage scripts
- Testing requirements
  - Requirements drive testing
  - More requirements = more test scripts
- Test Management Tools
  - Primary goal initially to manage test scripts and provide coverage reports on requirements
- Test Management Solutions Designed from Bottom up



## Where have the solutions Evolved?

- ▶ Typical Test Management tools today
  - Create Test Cases
  - Associate Test Cases to Requirements
  - Execution Of tests
  - Associate defects to tests post execution
  - Coverage reports (maybe test dashboards)
    - What has been tested
- ▶ Bottom UP Architecture has not left much to build upon

## Industry has introduced new challenges

- Lab Management
  - Need to coordinate teams using test labs
- Globally Distributed Teams
  - The world is flat
- Accelerated release cycles
  - Takes 2 weeks to fully test and there is a release every week
- Centers Of Excellence
  - Test treated as its own project
- Quality requires process
  - Working with Iterative Development
  - Agile/TDD, Compliance, Governance



## Why is there poor quality – The Blame Game

### ▶ People

- Don't have enough testers
- Testers don't have the right skills
- Poor communication

### ▶ Time

- Testing got squeezed at the end

### ▶ Management

- They just don't see the value of testing

### ▶ Development

- Code was so bad I could not test it
- Development got it to test late



# Why is there poor quality – The Blame Game

## ▶ Tools

- We spend too much time trying to get the vendors tools to work
- We don't have tools
- We don't have the right tools

## ▶ Test Labs

- Could not get access to test environment in time
- Tested against the wrong configuration as we did not have the right configuration available in the lab

## ▶ Reporting

- Gave you all the data you did not understand it
- Poor visibility of progress to everyone





## Why is there poor quality – The Blame Game

### ▶ Requirements

- Not testable
- Poorly written
- No access or input to them

### ▶ Test Planning

- Didn't have a good Test Plan/Approach
- Didn't understand organizations Test objectives

### ▶ Process

- Right processes were not in place
- Spent too long in SVT
- Didn't have quality objectives defined



## We could keep going



- ▶ Not having a grasp on Quality Management has led us to where we are.

# What do Test Managers do anyhow (from job postings)

- Manage the test project and test teams
- Design and manage test program strategies.
- Develop Test Plans
- Track testing project against the test Plans and report status
- Coordinate testing project with development project
- Create Test methodology and best practices and drive test process improvements
- Develop and manage overall test schedule.
- Participate in design reviews to drive quality in design.
- Investigate, select and implements automation solutions for testing.
- Organize and implement Test Lab
- Coordinate review and approval processes
- Ensure Adherence to standards
- Review requirements and evaluate for Quality Concerns



## What do Test Managers do anyhow (from job postings)

- ▶ Project Management
- ▶ Test Leads
- ▶ Test Architects
- ▶ Deployment Managers
- ▶ Automation Experts
- ▶ SME's
- ▶ Compliance Experts
  
- ▶ SUPER MAN or Wonder Woman



## What needs to be Managed?

- ▶ Test Cases
- ▶ Test Scripts
- ▶ Test Plans
- ▶ Hardware
- ▶ Labs
- ▶ Builds
- ▶ Test Execution
- ▶ Reuse
- ▶ Risk
- ▶ Releases
- ▶ People
- ▶ Data
- ▶ Reports
- ▶ Test Logs
- ▶ Defects
- ▶ Quality
- ▶ Quality Process
- ▶ Software Versions



## Tools used Today?

- ▶ MS Word
- ▶ MS Excel
- ▶ Rational TestManager/ClearQuest Test Manager
- ▶ Mercury Quality Center/Test Director
- ▶ Compuware QA Director
- ▶ Borland SilkCentral Test Manager
- ▶ Home Grown Solutions
- ▶ Open Source (e.g. Fitnessse, QATraq)



## Which one meets our real Needs?

- ▶ If a true solution existed Word and Excel would not have been on the list



## Whatever happened to the simple view of Quality



- ▶ Or was it ever this simple?
  - Just never understood Quality



# The Importance of Test Management

- Would you go on a road trip without a map?
  - ▶ How would you know where to go?
  - ▶ How would you know how long it would take?
  - ▶ How would you reroute around problems?
  - ▶ How would you know how to do it better next time?
- Automated Test Management answers these questions...



# Test Governance and Control

## *Test Planning, Execution and Project tracking*

- Automated Test Management
  - ▶ Track your test plans and cases
  - ▶ Track your requirements and defects
  - ▶ Execute your tests
  - ▶ Measure your progress
- Automated Test Management is your Software Quality GPS
  - ▶ Know where you are
  - ▶ Know where you're going
  - ▶ Make your process predictable



# Process Management

## ▶ Quality Processes

- Tools/Solutions need to work in context of process
- Takes Time if you do it right
- Testing Cost More than expected (upfront cost)
- Test Management Needs to provide value
- Interfaces with Software Lifecycle
- Aligns with and validates organization policies

## ▶ A poor Quality Application

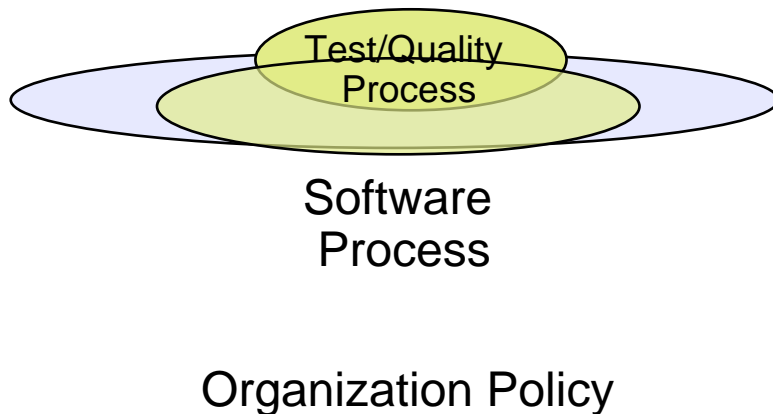
- Leads to a team with low morale
- Cost more over time (support, escalations, etc)



# Policies

## ► Organization Policies

- Standard software process defined
- Project process derived from organization process
- Software assets are maintained
- Testing only partially validates
- Test is a small part of overall effort

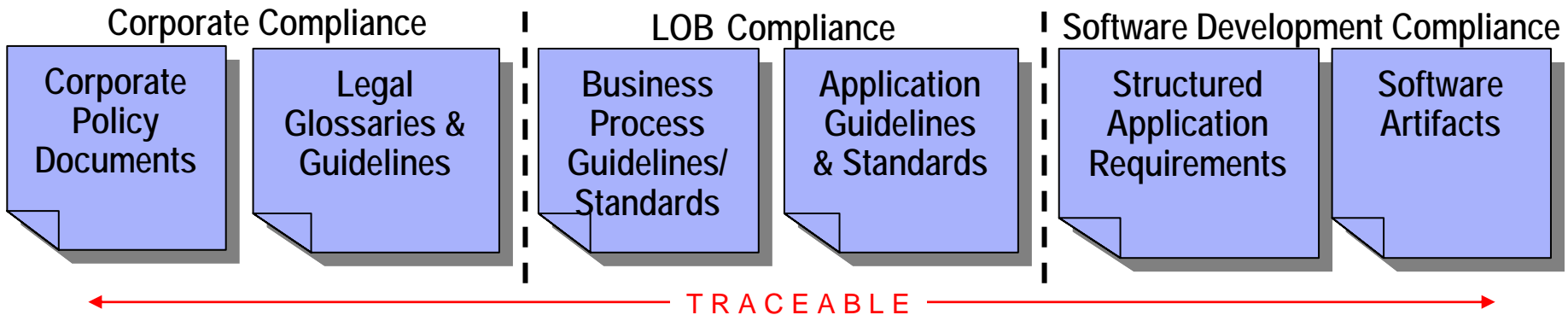


# Provision regulatory policies across the enterprise



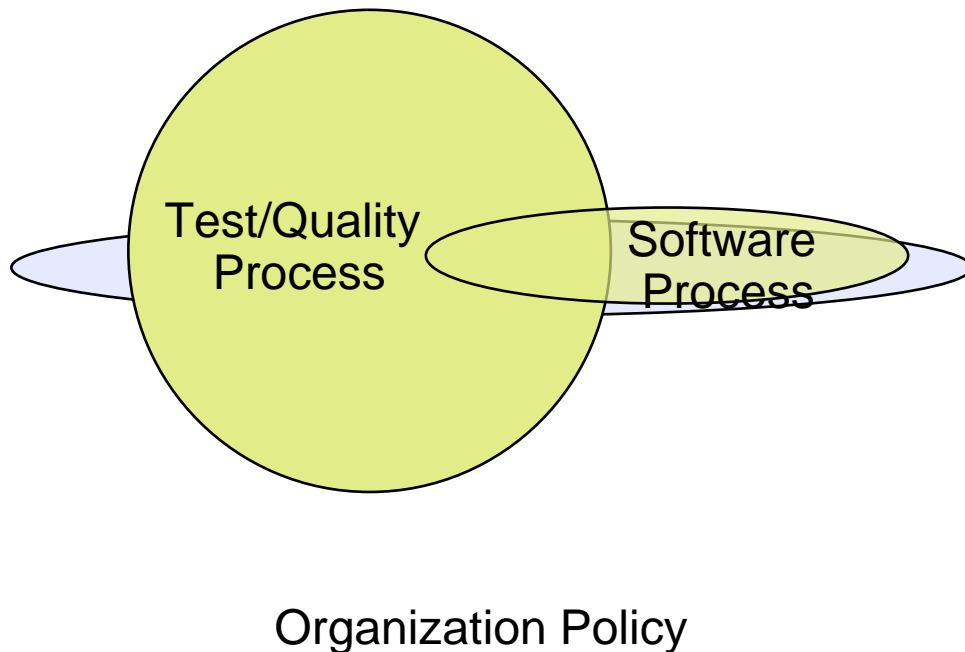
**Risk Officer / Analyst**

- Trace legislation to policy interpretation to application requirements
- Identify coverage gaps
- Understand the impact of changes to regulations, policies, or applications



# Quality Process vs Software Process

- ▶ **Software Process/Policies**
  - ▶ Builds software in accordance with organizational policies
- ▶ **Quality Process**
  - ▶ Validates software meets organization policies
  - ▶ Quality police come out to try to force standards for quality because of small overlap



- ▶ Quality is more than a small subset of software process



# Compliance (executives have interest in quality)



**Chief Risk Officer**

- **Establish policy guidelines**

- “We will adopt more conservative revenue recognition policies.”

- **Analyze and identify high-risk business processes; establish remediation processes and policies**

- Policy A: If credit score is “good”, then we will book revenue when contract is signed; else we will book revenue when cash is received.
- Policy B: We will no longer pre-announce new offerings more than 30 days in advance.



**Risk Analyst**



**IT Executive**

- **Track remediation progress**

- Policy A: Impacts 4 financial systems
- Policy B: Manual business control



**Test and Development Team**

- **Certify software remediation**

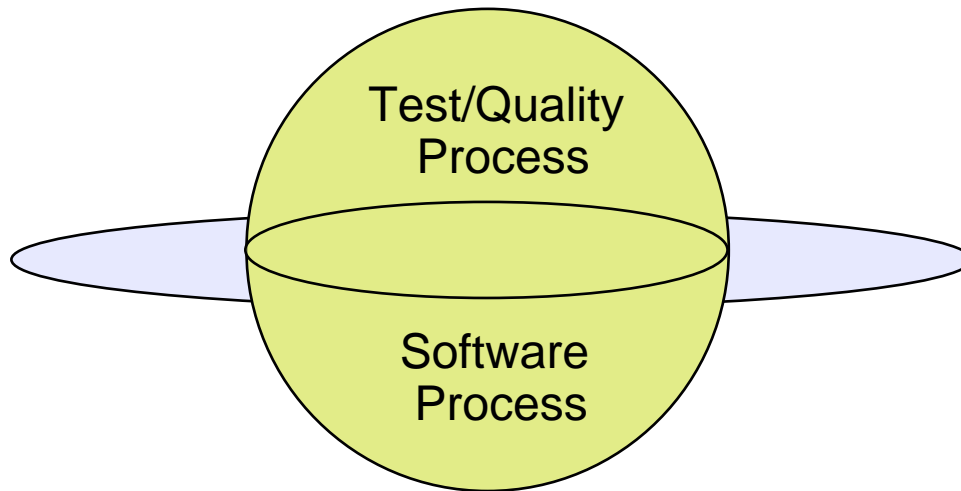
- Implemented Policy A
- Used appropriate controls throughout development
- Deployed to a secure and auditable environment



# How they merge is up to you..

## ▶ Organisational Policies

- ▶ Standard software process defined
- ▶ Quality procedures in place
- ▶ Quality measures explicitly defined
- ▶ Project process derived from organization process
- ▶ Software assets are maintained
- ▶ Quality validated



Organization Policy

- ▶ **Software Process = Quality Process**
- ▶ **Quality Process = Quality applications**

- ▶ Quality is not possible unless it is defined and measured in every facet of software development process





# Policy = Process = Quality

## 1. *WHAT* you build...



Are all compliance requirements accurately captured and implemented in key applications?

- Lifecycle traceability of requirements through test results
- Continuous validation

*Documents: Applications that meet compliance mandates*

## 2. *HOW* you build it...



Were all software changes performed for valid business reasons by authorized personnel?  
Was the software developed actually deployed?

- Auditable workflow management
- Verifiable software builds

*Documents: A secure, audit-ready development infrastructure*

## 3. *HOW* you manage it...



Can you demonstrate oversight over IT remediation projects?

- Compliance project tracking
- Metrics with drill-down analytics
- Periodic Validation of Processes

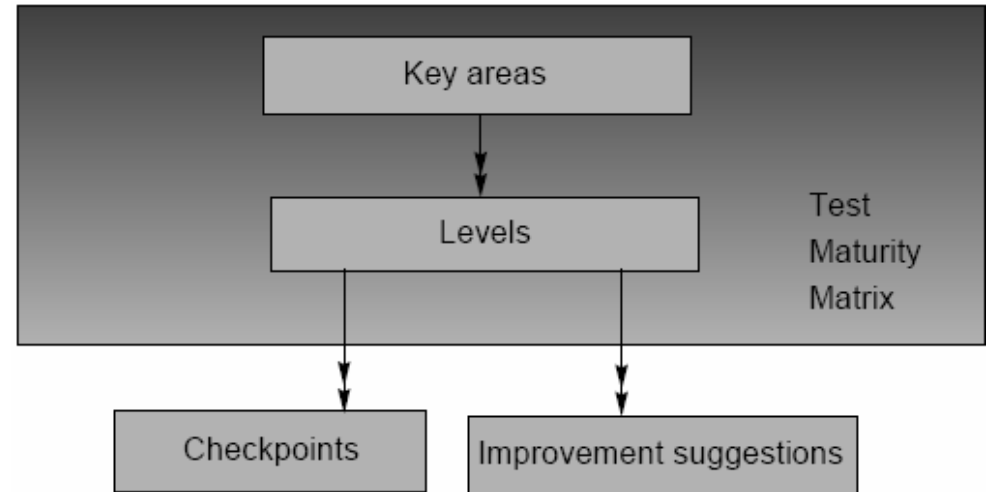
*Documents: Effective IT oversight*



# Test Process Improvement (TPI)

## ▶ Process needs

- Need a process
- Need to redefine process when policies not met



## ▶ Need more than improvement

- Everyone understanding of current Process
- Everyone collaborating and working in context on the current process
- Reporting and Metrics against the process
- Management of our QA/test effort in context of process
- Ensuring Organizational policies are met

## Quality Process

- ▶ Modifiable/Customizable
- ▶ Defines and drives usage model of tools
- ▶ No guess work on what needs to be done next
  - Or where I am now
- ▶ Learning a tool and process is the same
- ▶ Drives delivery of right information at the right time



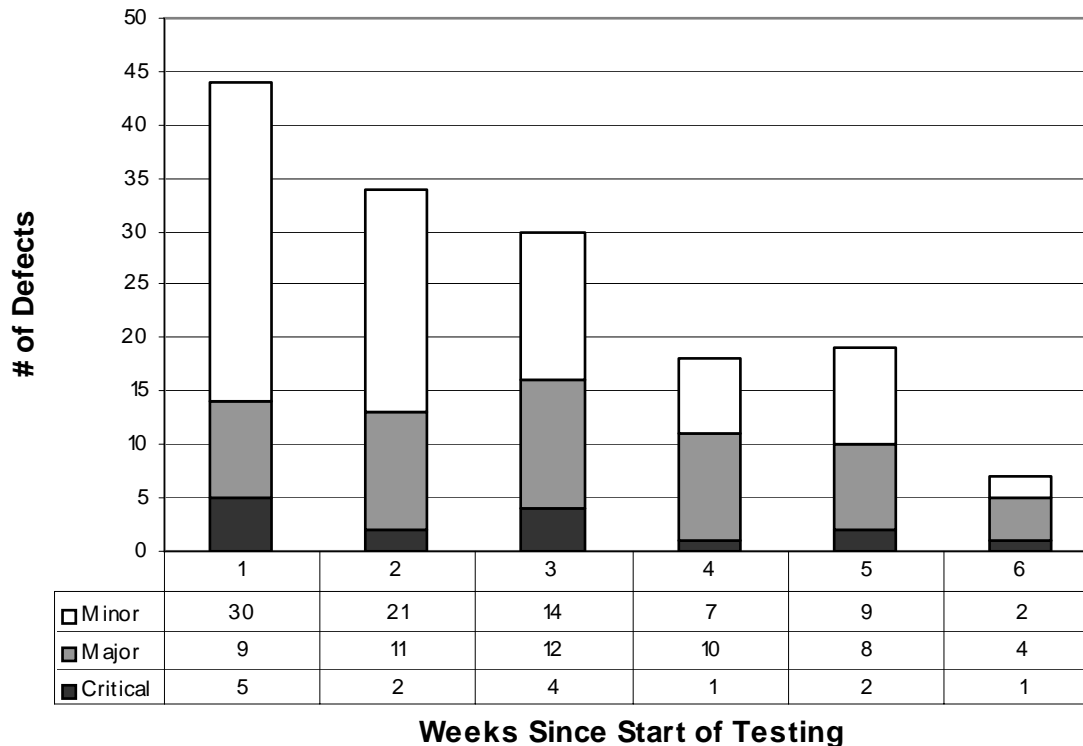
# Project Management

- ▶ Testing as its own project
  - Subproject in Software development Project
  - Separate project managed as part of a Center of Excellence
  - Outsourced Testing/Global System Integrators
- ▶ Communication of Results
- ▶ Scheduling and Project Tracking
  - Where am I and what do I need
- ▶ Can't control scope without treating testing as its own project
- ▶ Tight Integration and interfacing with Software Development



# Reporting and Metrics

Defect Detection Arrival Rate



▶ Begin with the end in mind

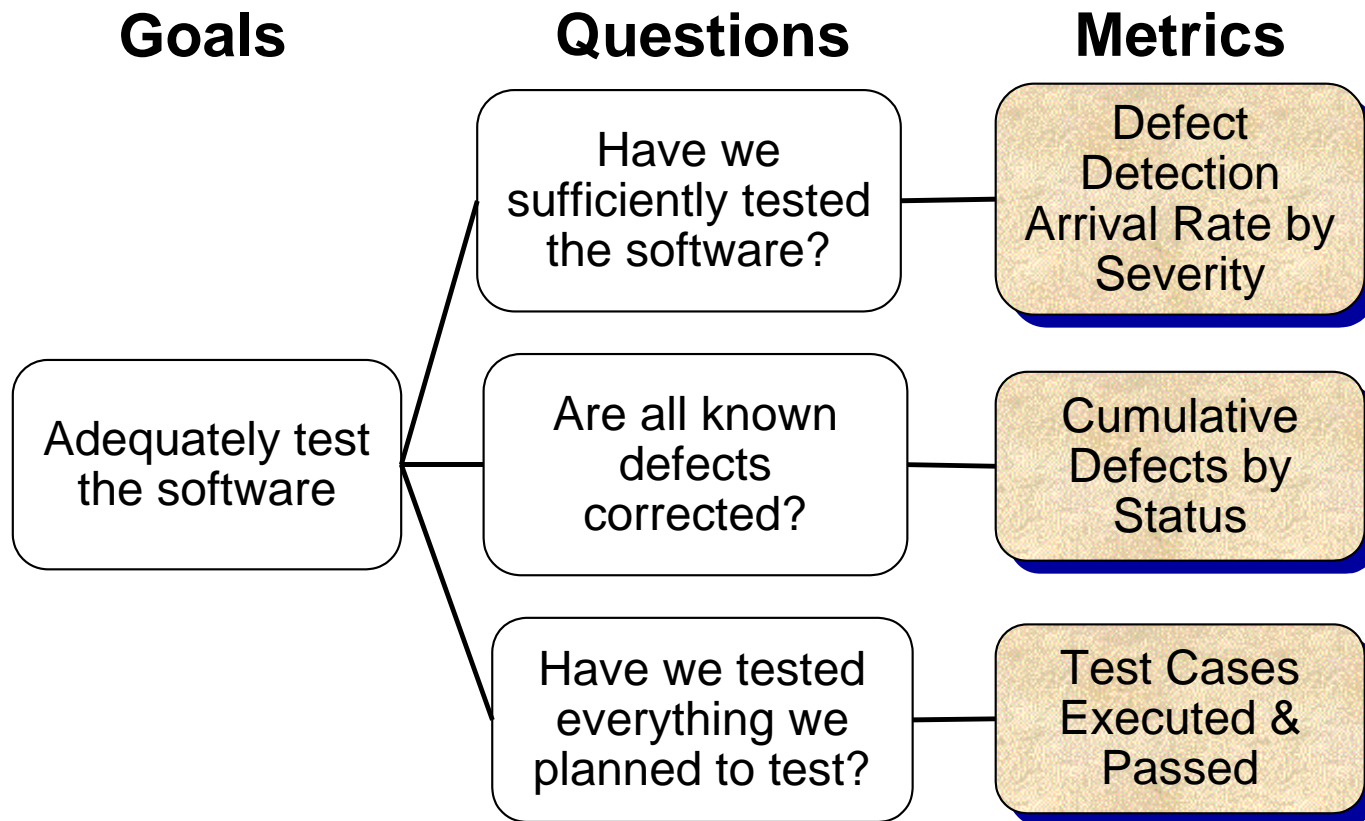


## Reporting and Metrics

- ▶ Begin with the end in mind
- ▶ Why do you test
  - We test to get data for our reports
- ▶ One of top questions we get asked as a vendor is
  - How can I create THIS report
- ▶ Customers tend to spend as much time creating reports to roll up to management as they do testing
- ▶ Reports help drive the Test Process
  - Your process and data collected feed your reports



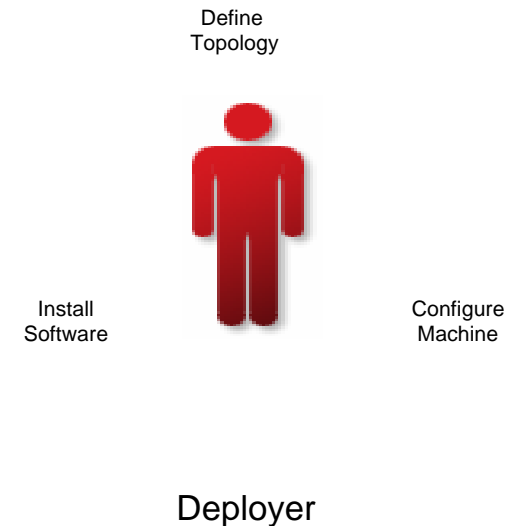
# Reporting and Metrics



# Test Labs

- Cost of Manually Configuring Machines for Test. A recent survey conducted within Rational revealed the following:
  - ▶ 19 people on the team
  - ▶ On average, 26% of their time is spent manually configuring machines to get ready to test.
  - ▶ The team is roughly split 50/50 at an annual cost\* of \$58k & \$170k
    - 10 people @ 170k = \$57
    - 9 people @ 58k = \$28 per hour
  - ▶  $10 \times \$57/\text{hr} \times 10.4 \text{ hours/wk} \times 52 \text{ wks/year} = \$308,256$
  - ▶  $9 \times \$28/\text{hr} \times 10.4 \text{ hours/wk} \times 52 \text{ wks/year} = \$136,282$
- NET: \$444,538 spent per year manually configuring machines for test

## Lab Management





## Distributed Teams

- ▶ Reducing costs,
- ▶ Decreasing time to market
- ▶ Leveraging specific skill sets



## Distributed Teams and Test Management Complexity

- ▶ Exposes issues between testing and development
- ▶ Status reporting on testing is even more time consuming
- ▶ Collaboration not done well
- ▶ Parallel Testing is difficult
- ▶ No CM typically in place for testing
  
- ▶ And all this needs to be managed

# Distributed Teams and Test Management Complexity

- ▶ Distributed Team Considerations
  - Planning
  - Authoring
  - Execution
  - Reporting



# Team Collaborations

## ► Types of Collaboration

- Desktop sharing
- Screen captures / video capture
- Whiteboard sharing
- Instant Messaging
- Review & Approval



## Team Collaborations

- ▶ What to collaborate upon
  - Review & Approval
  - Task assignments and Work Balance
  - Scheduling of Lab
  - Asset Sharing
  - Execution Results



# Integrations

- ▶ Quality effects everyone and everything
  - Requirements
  - Defects
  - Project Management
  - Lab Management
  - Development Environment
  - Source Code



## Who is interested in Quality

- ▶ Test Project Managers
- ▶ Test Leads
- ▶ Test Architects
- ▶ Project Managers
- ▶ Testers
- ▶ Developers
  
- ▶ Basically – Everyone
  - BUT IN DIFFERENT WAYS



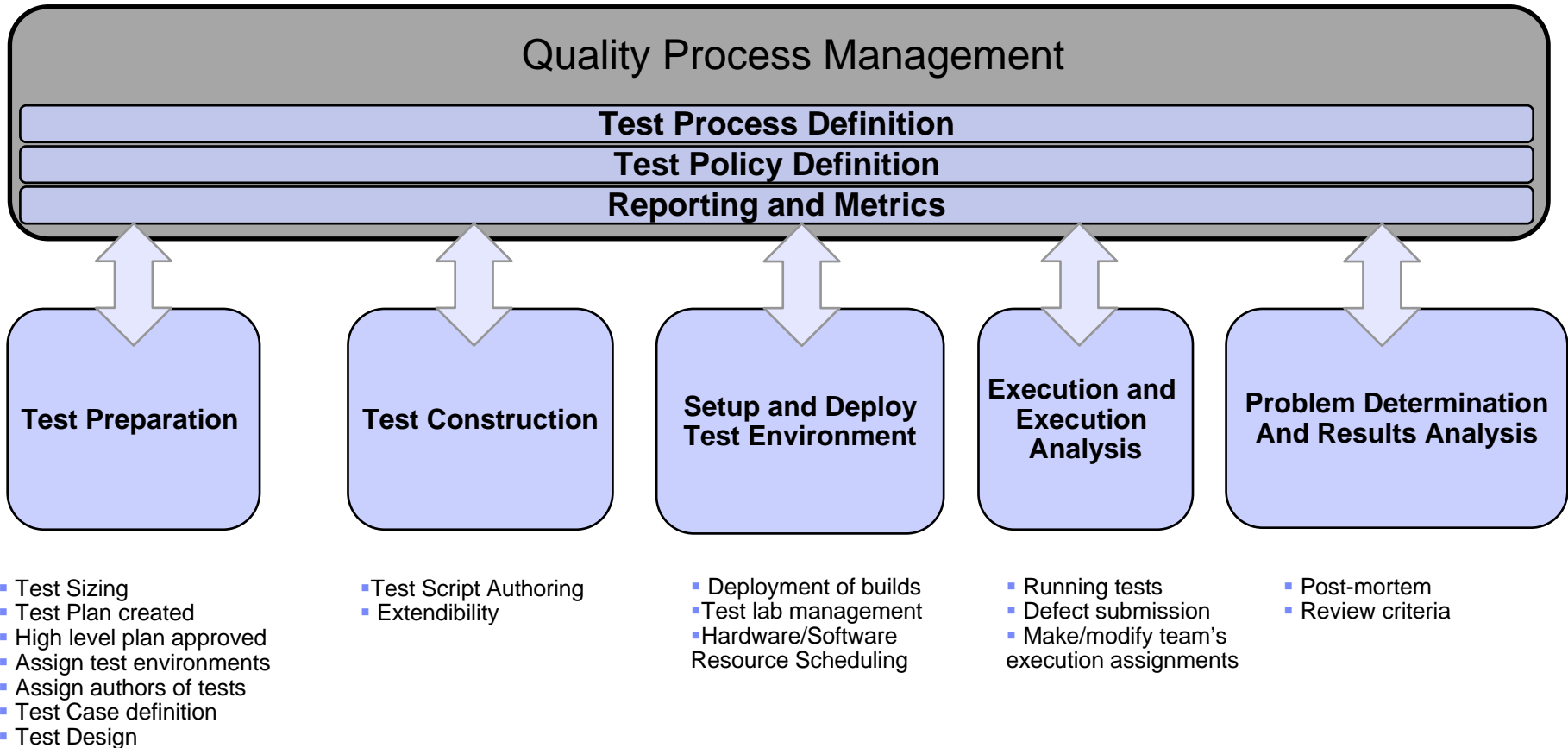
## Solutions need to be easy

- ▶ Not something that gets in your way
- ▶ Simplifies your job – not making it harder
- ▶ A tool isn't useful if it doesn't make a given task easier.
- ▶ Want testers spending the maximum amount of time testing the product
- ▶ Want testers to test and not waste time using an overly complicated test management tool
- ▶ Need is for a single unified comprehensive solution



# Quality Management Lifecycle

- Planning the plan
- Plan creation, review, approval
- Identifying test environments
- Establishing Acceptance Criteria



# My Vision of Quality Management



▶ Ahhh, now that seems easy..





# Questions





# Thank You

*JohnWatkins*  
*IBM Rational Channel Technical Sales*  
*johnwatk@uk.ibm.com*

