Tools for Testing

Automatic for the People

John Fodeh
HP CI International Expertise Team
john.fodeh@hp.com

Contents

- Rationale for Automation
- Testing Software Quality
- The Testing Tool Landscape
- Some of the Challenges
- A Look Into the Future

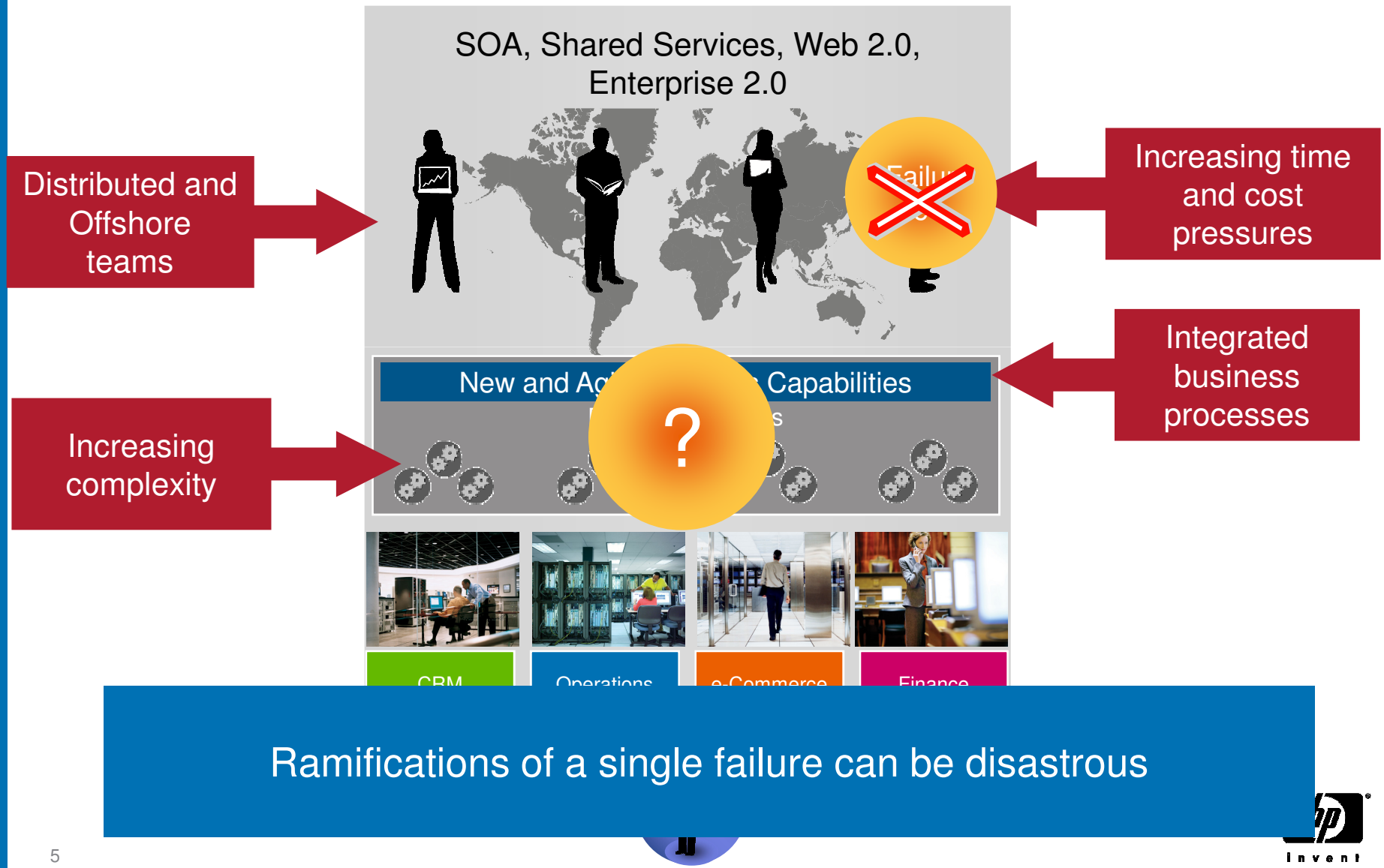
"The criticality of software to the business, the increasing complexity of software applications and systems, and the relentless business pressures for quality, productivity, and faster time to market have all been positive drivers in the past and will continue to be…"

IDC, 2006

Yesterday's Business Applications

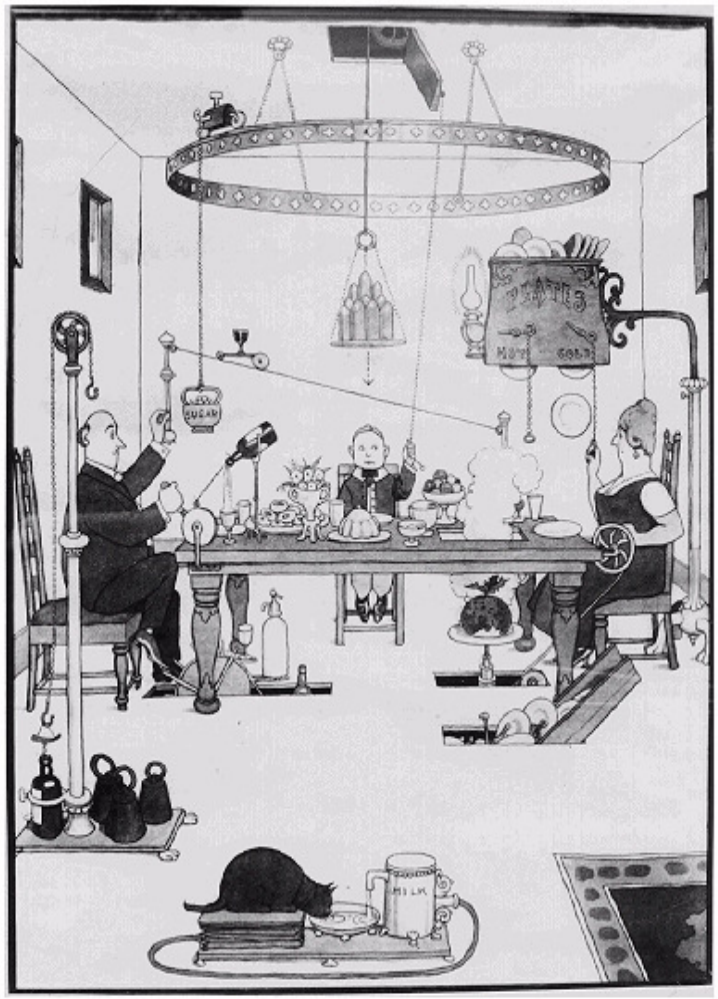
- Failures limited to a smaller set of applications
- Risks were limited, consequences of failure finite

Today: Relentlessly Increasing IT Complexity



Rationale for Test Automation

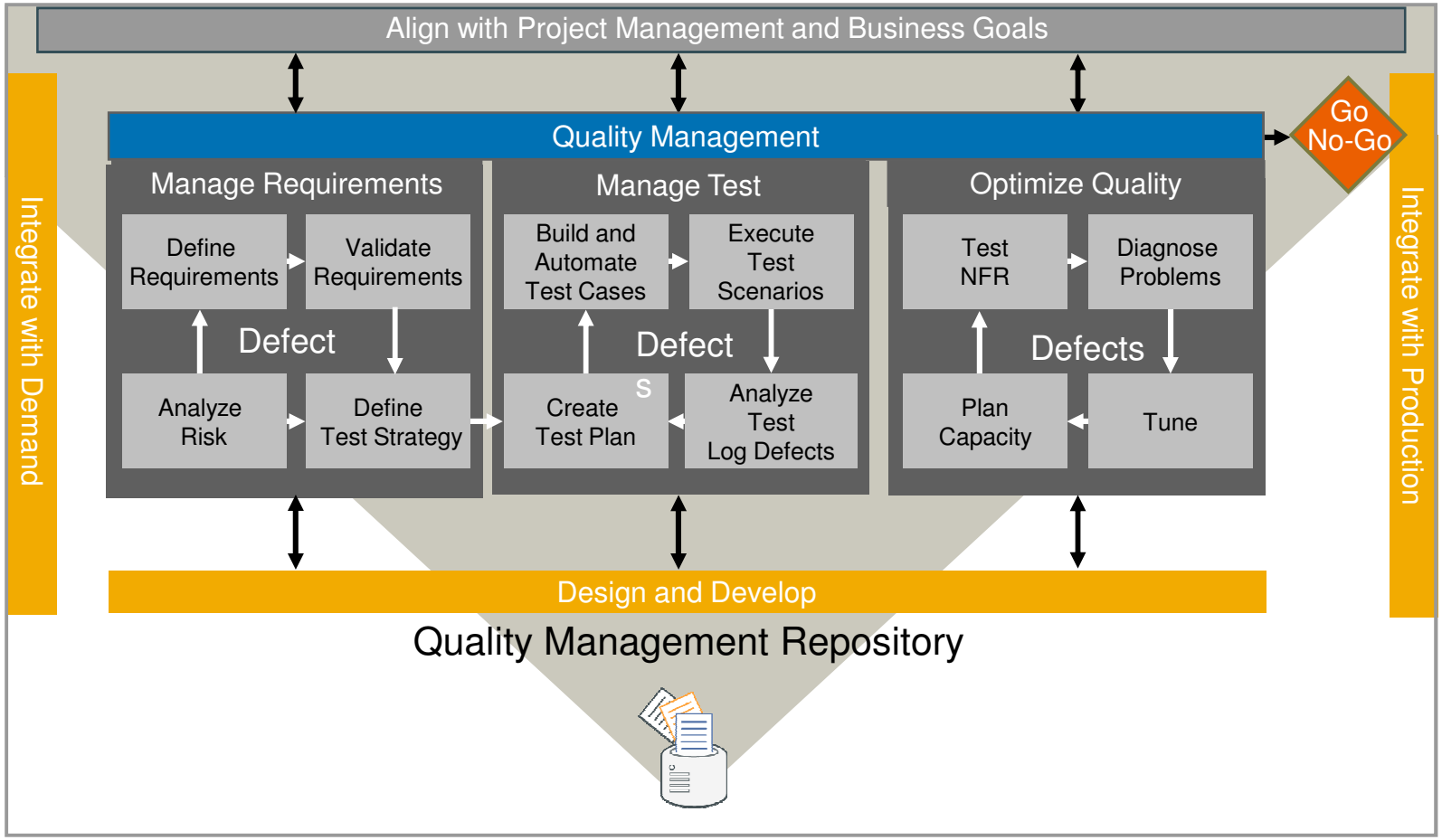
- Replacing repetitive, tedious and often error-prone manual testing
- Ensuring the consistency and repeatability of tests
- Performing tests that are difficult to run manually
- Enabling full regression cycle within a reasonable time frame
- Coping with new technologies
- Responding to new and iterative development models



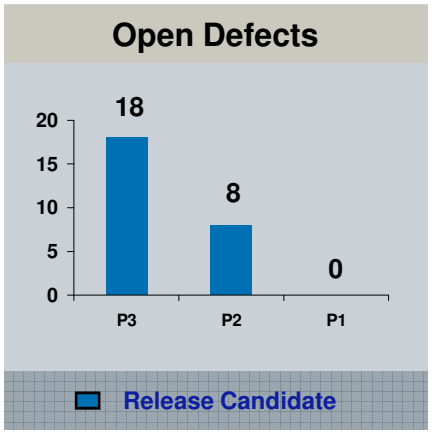
A Vision of Automation

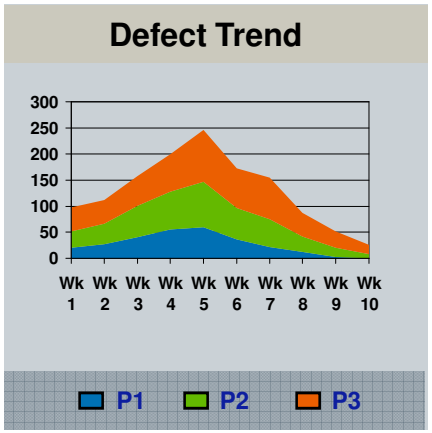
A hundred years ago, the concept of the future lacked one major ingredient... the computer! *(Image courtesy of Rosemont Engineering.)*

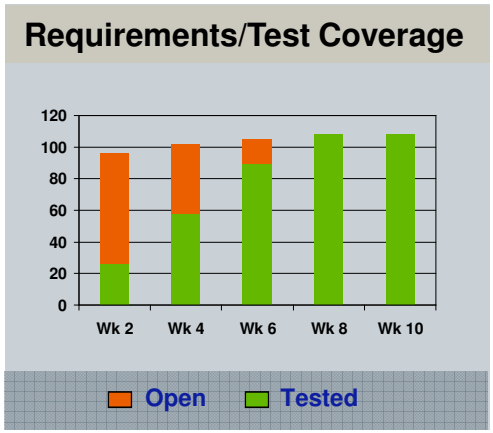
Testing is not an Island – Expanding the Territory

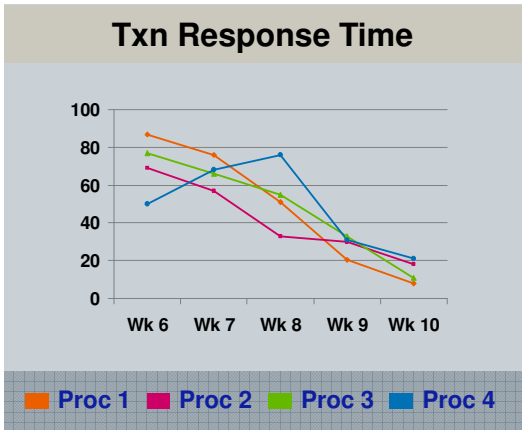


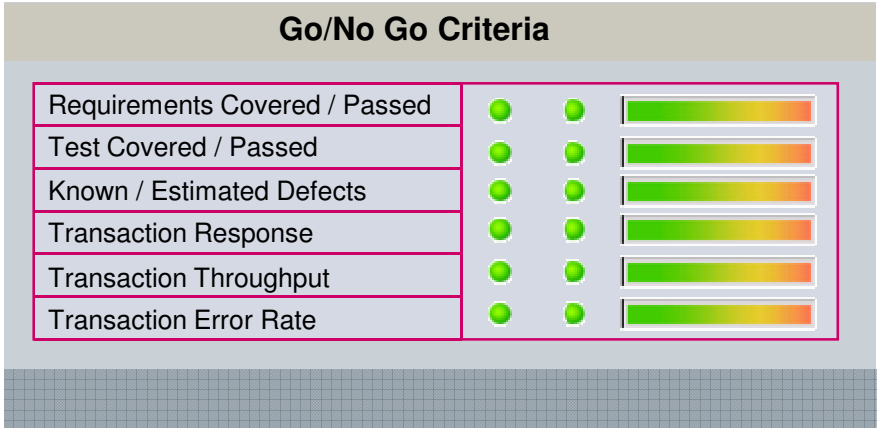
Continuous Insight Into Application State and Quality











What is Software Quality?

- “Conformance to requirements” [Crosby]
- “Fitness for use” [Juran]
- “Quality is pride of workmanship” [Deming]
- Value for money
- Quality is in the eye of the beholder

Quality is a multi-dimensional entity

ISO 9126: Quality Characteristics

Functionality

- Suitability
- Accuracy
- Interoperability
- Security

Reliability

- Maturity
- Fault tolerance
- Recoverability

Usability

- Understandability
- Learnability
- Operability
- Attractiveness

Efficiency

- Time behaviour
- Resource utilisation

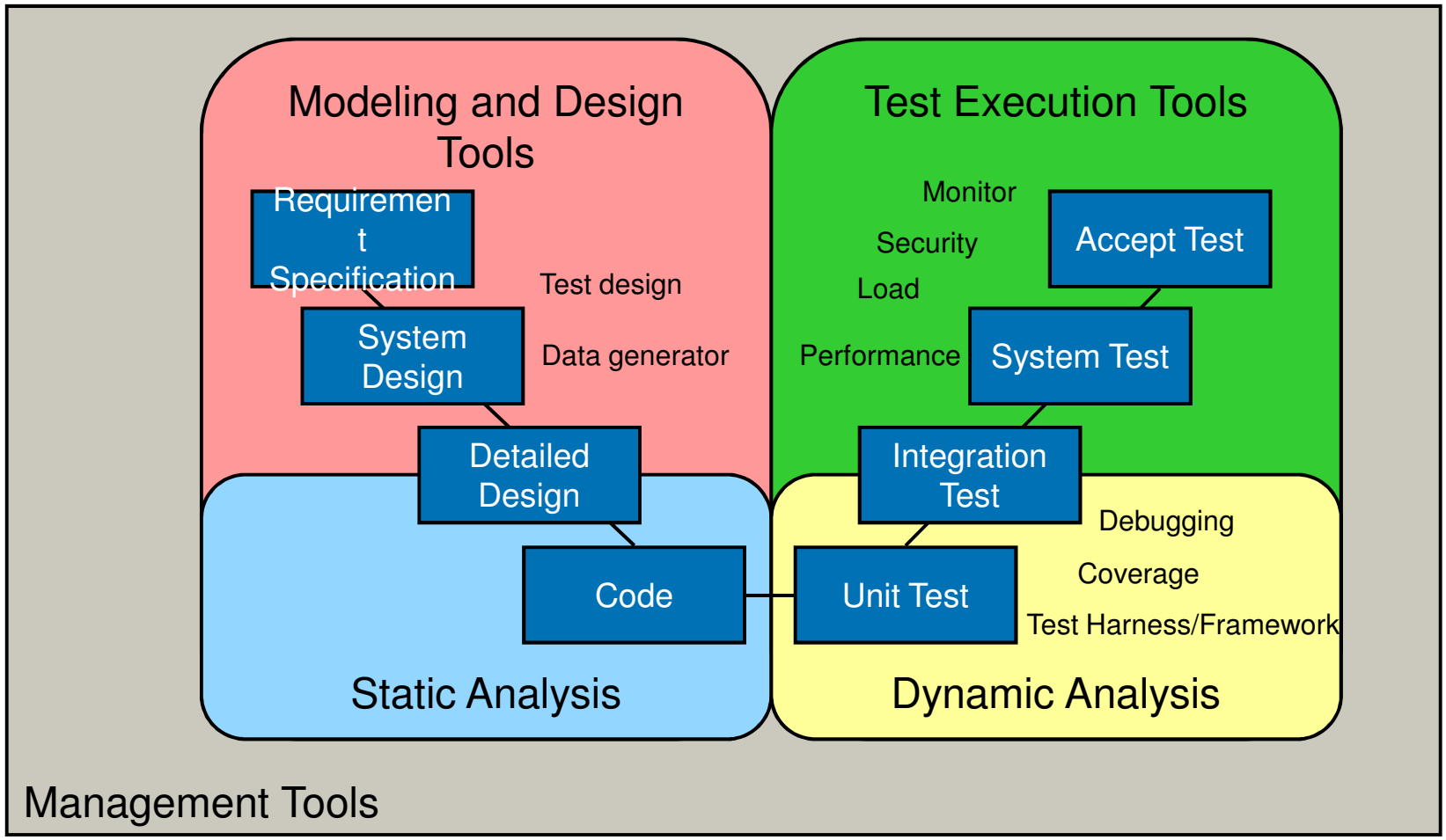
Maintainability

- Analyzability
- Changeability
- Stability
- Testability

Portability

- Adaptability
- Installability
- Conformance
- Replaceability

The Testing Tool Landscape



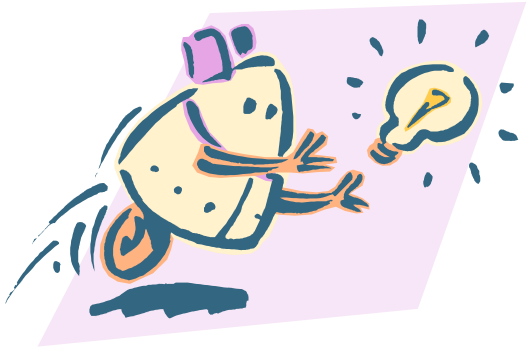
*Inspiration: Graham, Fewster

Tool Types - Management Tools

- Central repository for test assets
- Process workflow and guidance
- Governance and traceability
- Examples
  - Requirements management
  - Test management
  - Incident/Defect tracking
  - Configuration management
  - Review process tools

Tool Types – Modeling, Design and Analysis Tools

- Requirement validation tools
- Modeling
- Design
- Test data generator
- Static Analysis

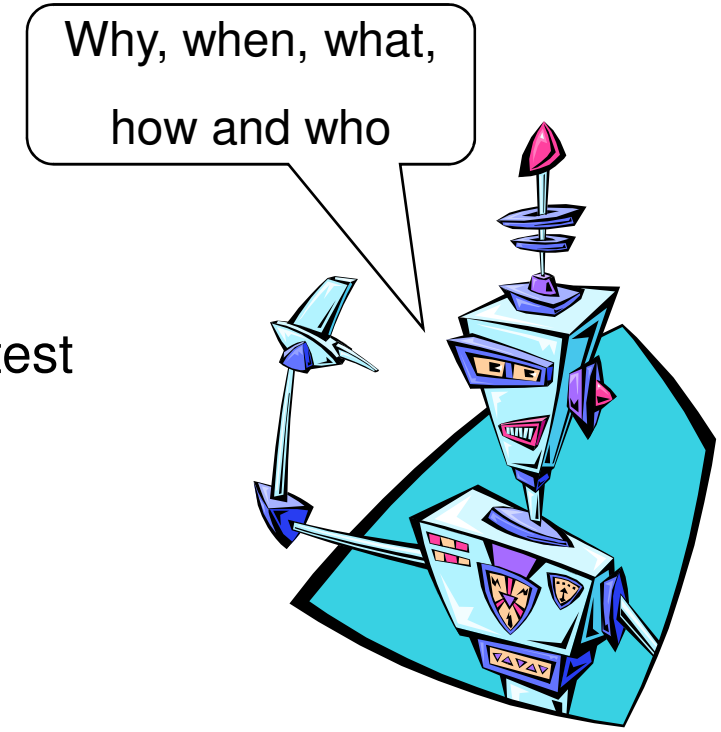


Tool Types – Dynamic Analysis and Execution Tools

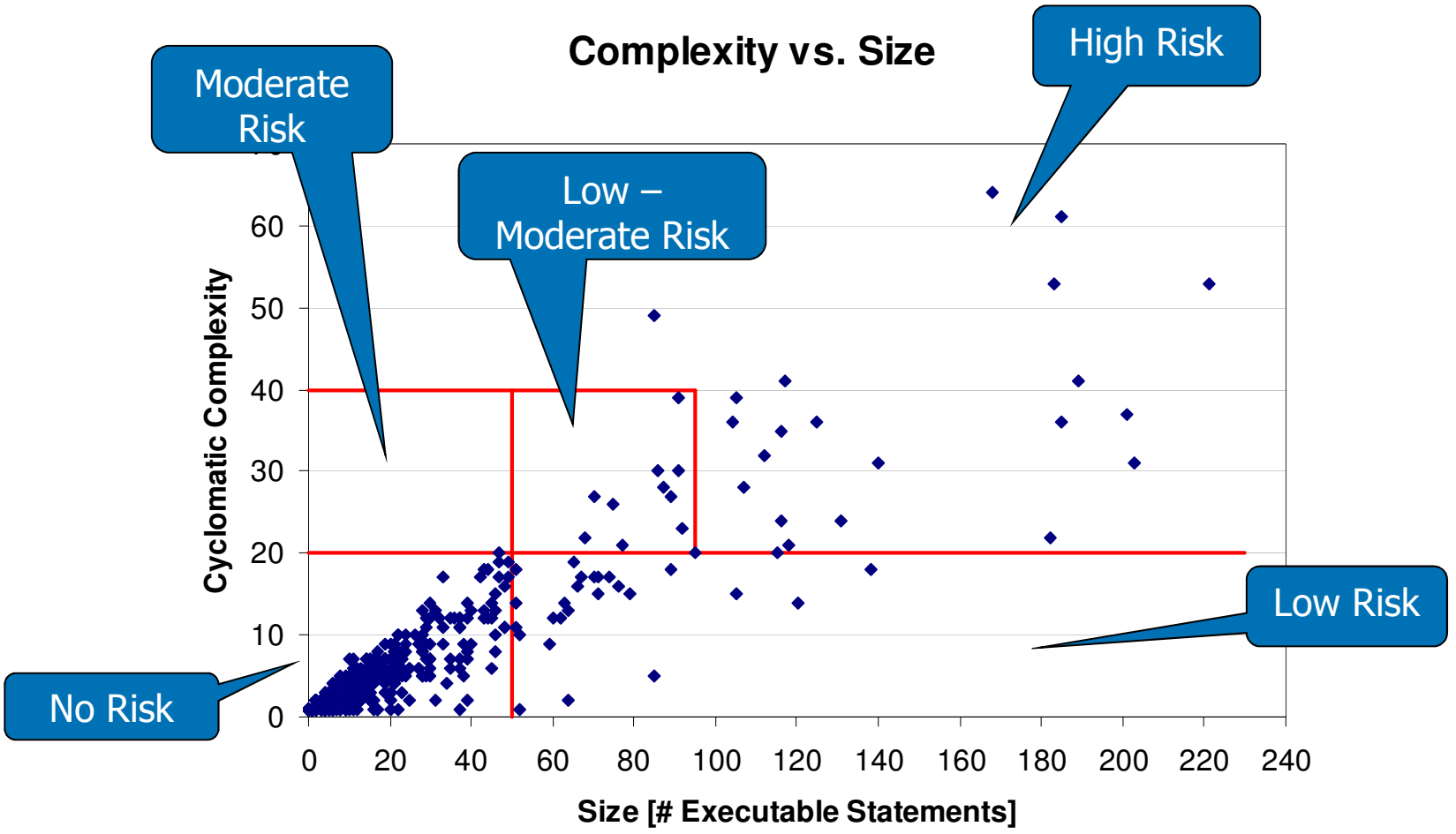
- Monitors
- Security
- Performance
- Load
- Test execution and comparison tools
- Dynamic analysis
- Test Framework/Harness

Have Clear Objectives

- Why
  - Overall aim of automation
  - Purpose of test
- When
  - Unit, integration, system or accept test
- What
  - Scope of test
- How
  - Automation approach
  - Input and reference data
  - Quality attributes (portability, scalability, robustness, etc.)
- Who
  - Responsible for design, development, test and maintenance

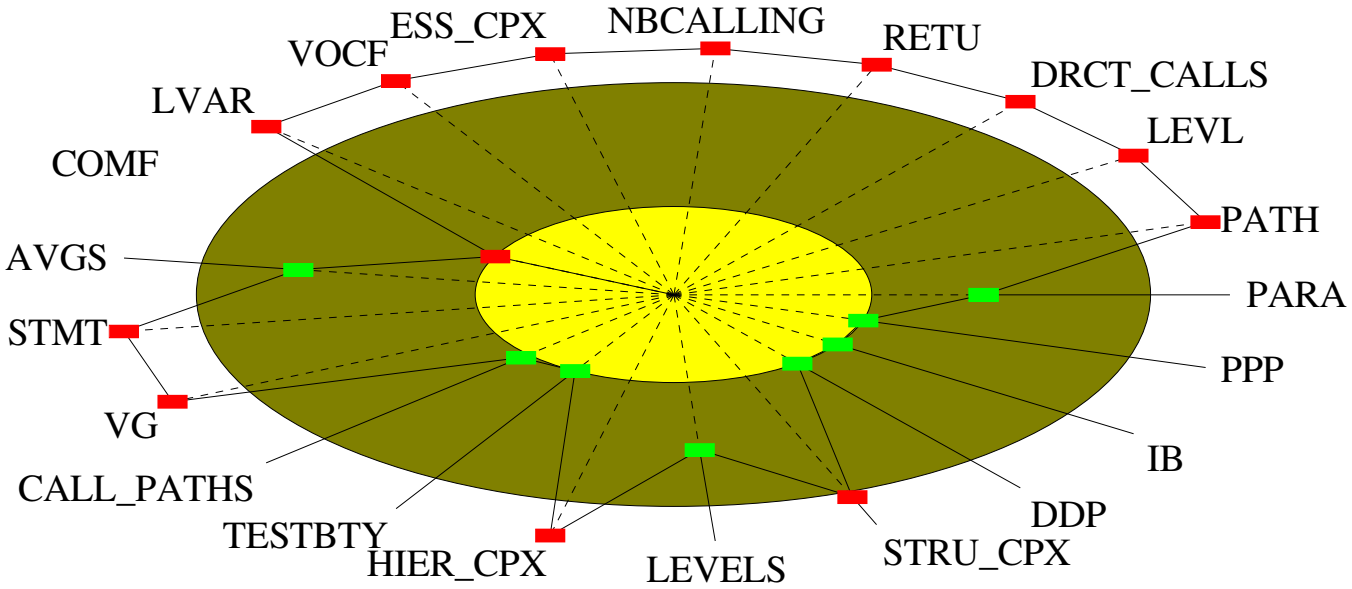


Static Analysis Quality Models



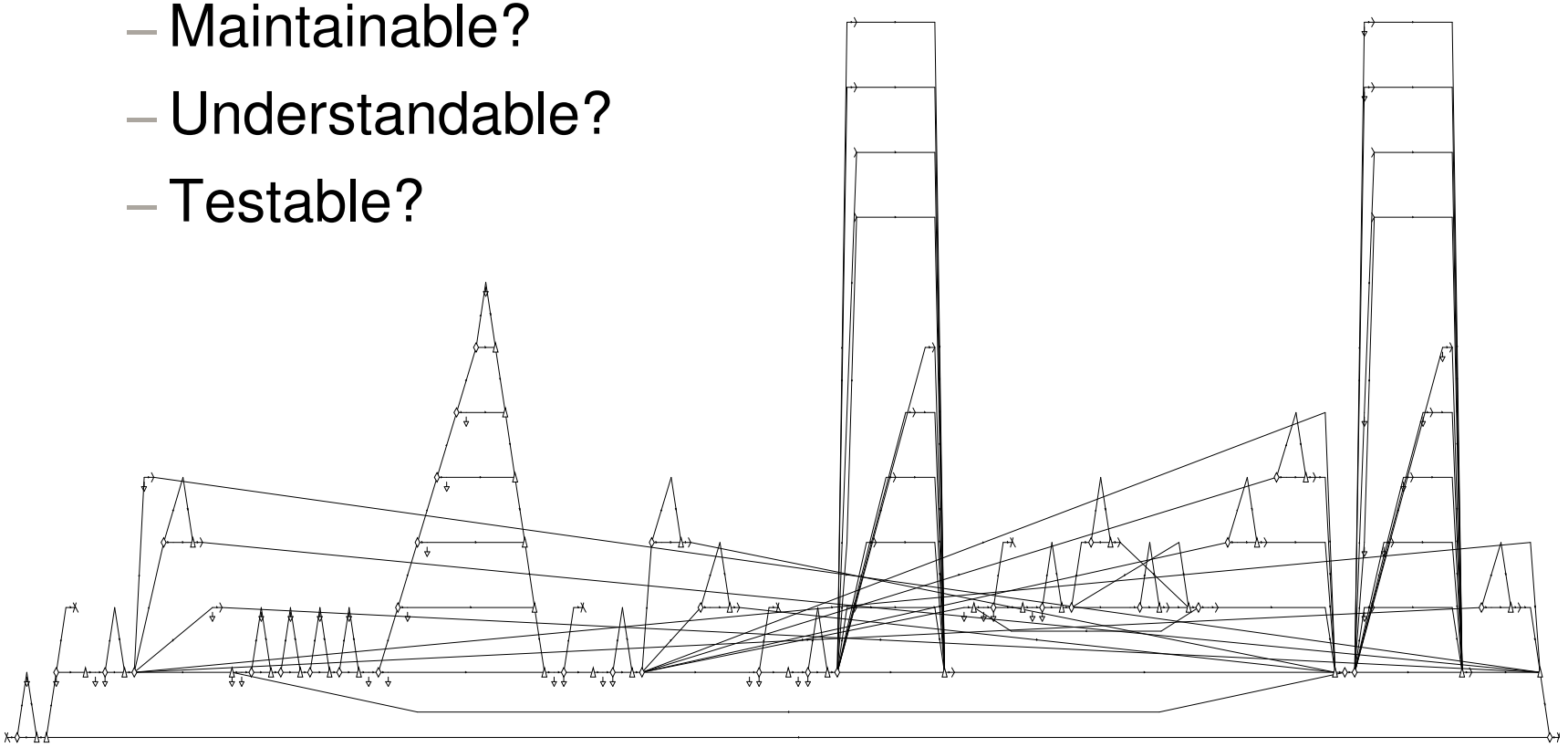
Static Analysis Quality Models (cont.)

- Identifying problematic areas

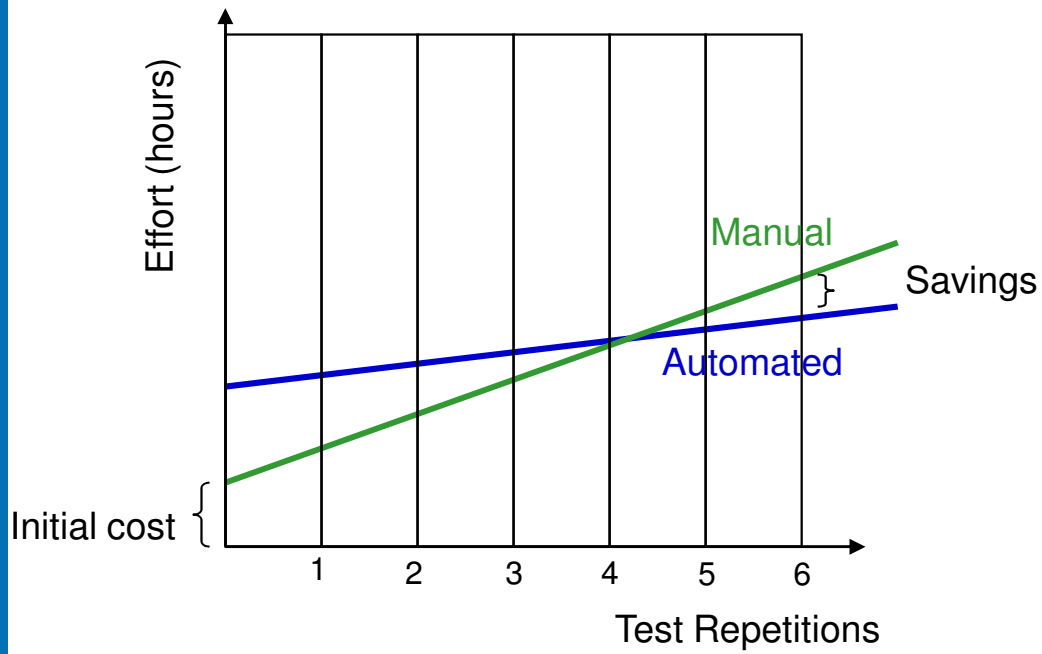


A Picture Speaks a Thousand Words

- To what extent is this code:
  - Maintainable?
  - Understandable?
  - Testable?



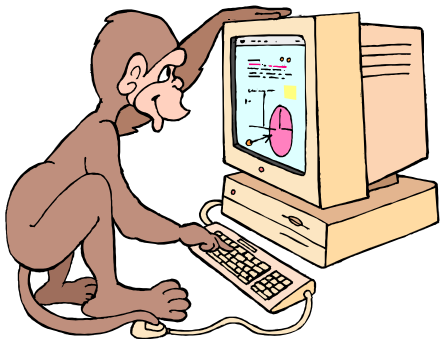
Automatic vs. Manual Test



But automatic tests

- are always different (timing, verification, way of operation)
- will only find what it's looking for
- may not be feasible
- requires more maintenance

Test Monkeys



Action List

```
# Script file: TEST4.ASC
KEY 5;
CP_KEY FOCUS_UP;
# KEY BOX_SIZEUP;
KEY 8;
TRACKBALL 0,64;
KEY SHIFT,R;
```

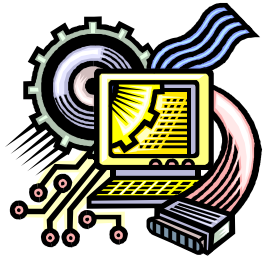
Random Test Tool

Application Under Test (AUT)

Log file

```
# Script file: TEST4.ASC
KEY 5,LONG;
CP_KEY FOCUS_UP;
# KEY BOX_SIZEUP;
KEY 8;
```

The Probe Effect



- Unintended alteration in system behavior caused by measuring that system [Wikipedia]
- Introducing a tool might have an effect on:
- System under test
  - E.g. code instrumentation might cause unwanted side-effects
- Existing processes
  - E.g. current practices and processes will most likely need adjustments
- People
  - Some might feel threatened by the tool

Refining the Tester Position

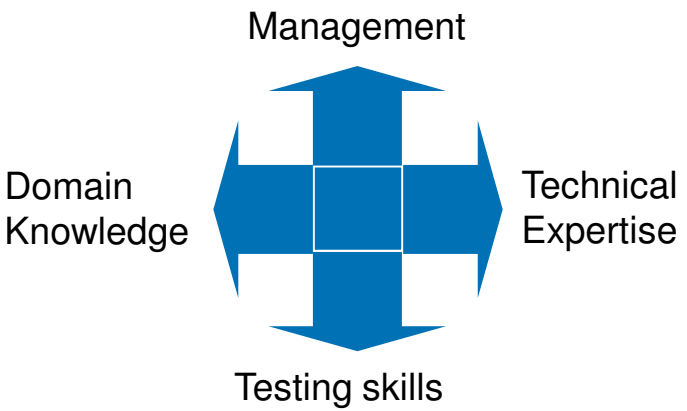
- Test manager/leads
- Test analysts/designers
- Test technicians

- Technical testers
  - Test engineers
  - Test architects
  - Test toolsmiths

- Skills:
  - Development, customization and use of advanced test solutions
  - Specialization in non-functional testing
  - Technical peers of programmers and system architects
  - Unit/integration test
  - System and database administrators
  - Release and configuration management

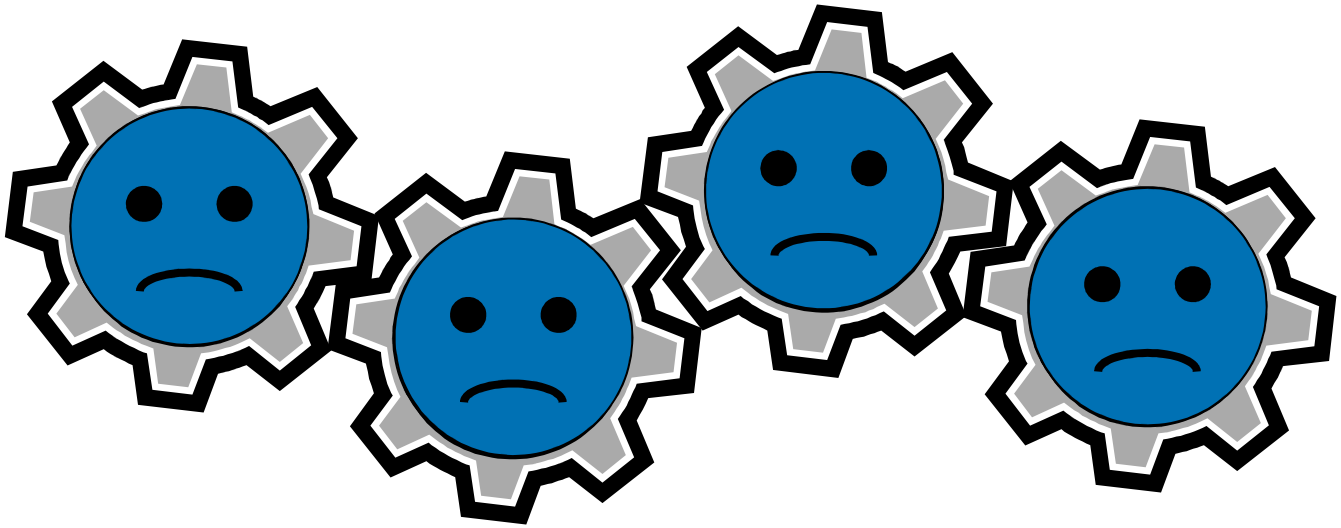
Get the Balance Right

- Your test team needs to have the right mix of roles, skills and competencies



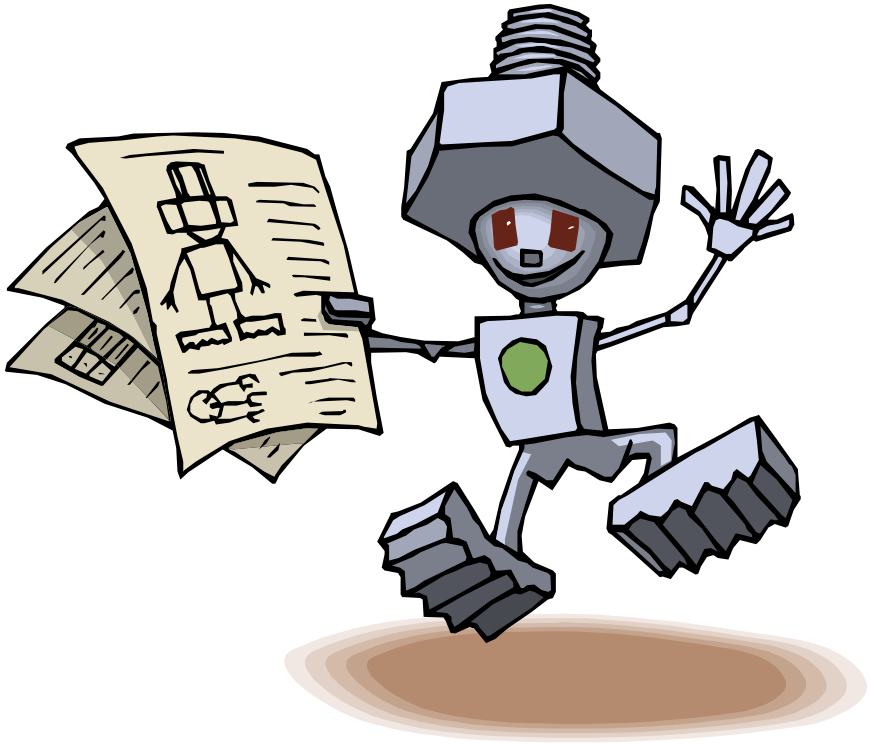
- Test Manager
  - Knows how to lead the team
- Domain knowledge expert
  - Understands the application and business requirements
- Technical expert
  - Handles technical issues
- Proficient Tester
  - Knows quality risks and test techniques

People Issues



- Testing is not replication
- You are not a machine part
- Your team members are not interchangeable machine parts

A Look Into The Future



Trends

- Model-Based Testing
  - Automatic generation of code and test
- Tool and AUT compatibility
  - Self-healing code-free scripts
- Agile development
  - Role of tester in Agile development is still evolving
- Holistic approach towards quality
  - More focus on technical testing
- Need for environment, data and configuration management
  - Virtualization to the rescue
- Professionalism of testing
  - Lack of skilled people – better paycheck ☺

12 Steps to Success

- Have clear objectives
- Define your processes
- Get management support
- Involve test- and development team early
- Acquire dedicated resources
- Provide training, mentoring and coaching
- Know the requirements
- Validate the concepts
- Promote application testability
- Apply software development ”best practices”
- Deploy in small increments
- Monitor, evaluate and optimize

Tools for Testing

Law I: A robot may not harm a human or, by inaction, allow a human being to come to harm.

Law II: A robot must obey orders given it by human beings except where such orders would conflict with the first law.

Law III: A robot must protect its own existence as long as such protection does not conflict with the first or second law.

Isaac Asimov's "Three Laws of Robotics"

See you at the Hague at EuroSTAR

Thanks for your attention