

Are we there yet? Is Advanced Automation the pinnacle of Software Engineering?

Matthias Rasking
UCAAT 2013



High performance. Delivered.



**accenture**

consulting | technology | outsourcing



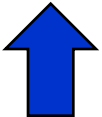
Introduction

Matthias Rasking

- **Senior Principal, Accenture GmbH, Kronberg, Germany**
- **EALA Platform Lead for Accenture Application Test Services**
- **Owner Accenture Delivery Methods for Testing and ADM for Testing Estimator**
- **Deputy Chair for Model Development and Model Maintenance for the TMMi Foundation, TMMi Lead Assessor**

How advanced automation sees the world

YACRDDQITTMRMWSFBC

Quality	Spec Defect Detection Rate – Test Design	SW Defect Detection Rate – Test Execution	Test Coverage
Your System			

Effort	Test Design effort	Test Execution Effort	Effort to train testers	Maintenance effort
Your System			Same as for traditional Testing	

Robert V. Binder at 2012 ETSI Model-Based Testing User Conference

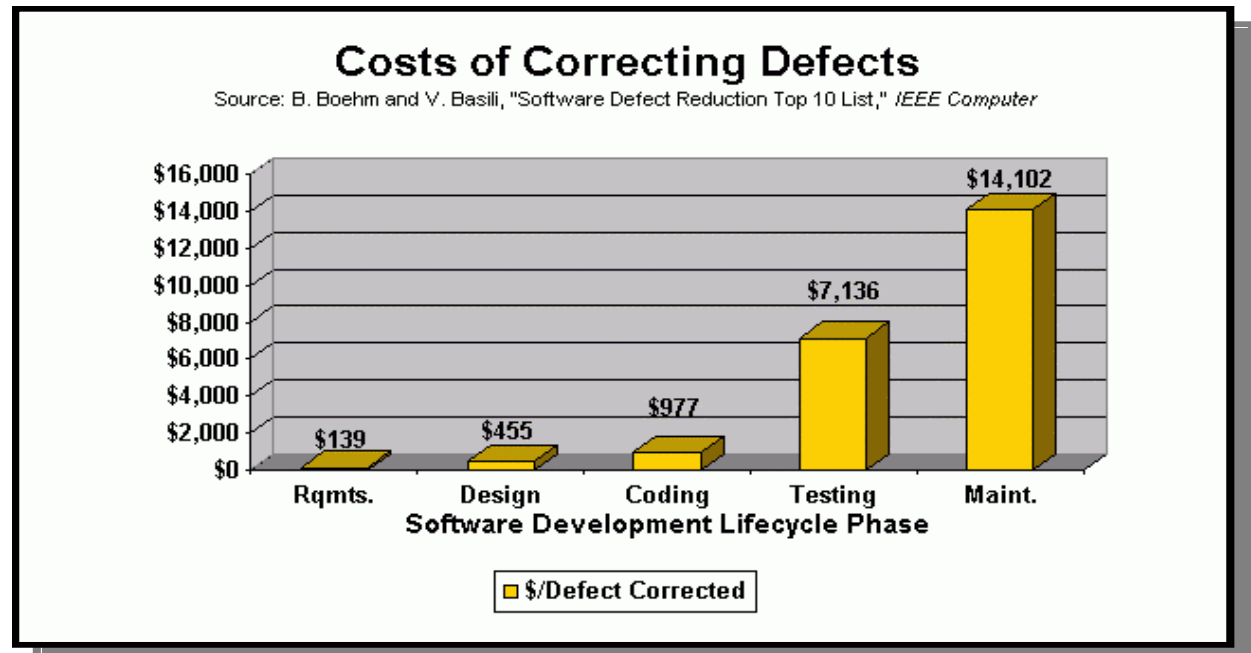
What have we learned?

- *Information Technology* is eating the world
- Common bug rates will **not** decline
- **More** new systems will have dragon kings
- Functional/model coverage MBT **will not** find dragon kings
- Multi-dimensional MBT **can** find dragon kings
- Opportunity for test engineering to **lead**

What have we actually achieved in 30+ years?

The concepts are not new...

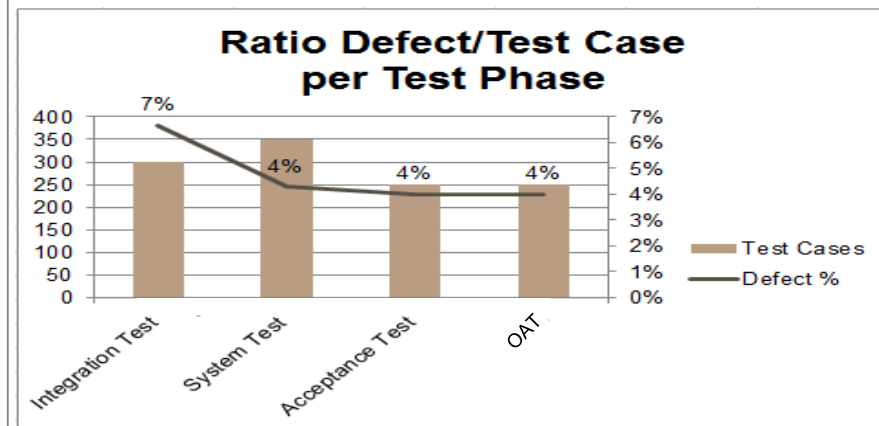
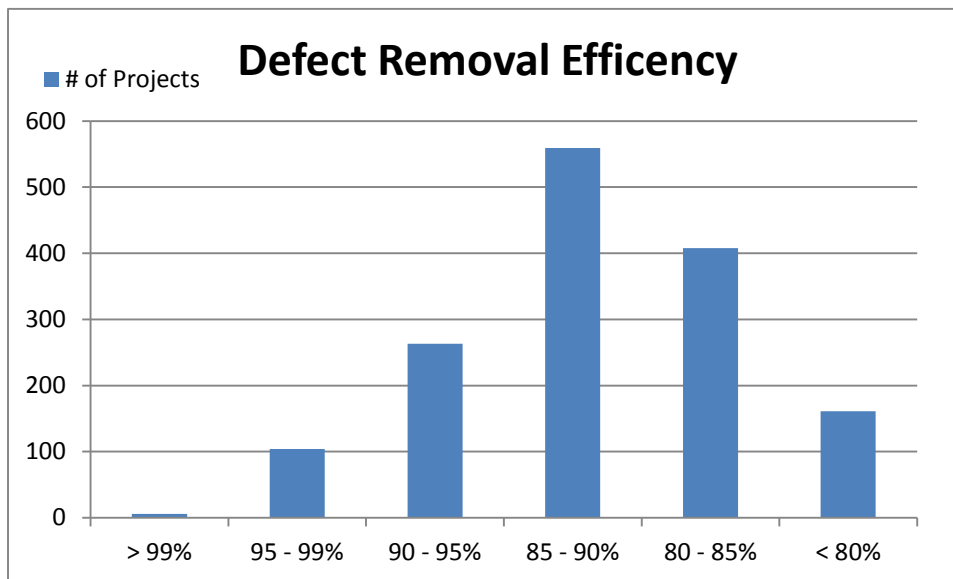
- Inspections: Known since at least 1976 to significantly increase productivity, quality and project stability
- Everybody knows Barry Boehm's work on the cost of defect (1975):



What have we actually achieved in 30+ years?

And still:

- Defect Removal Efficiency: > 15% found in Production
- Defect / Test Case Ratio: < 10%
- Defect Rejection Rate: > 25%
- Invalid definition of test coverage



Jones, C., Bonsignour, O., *The Economics of Software Quality*, 2012

Copyright © 2013 Accenture All rights reserved.

Let's start with some facts

- Requirements – Transformation of customer expectations into formal(ized) requirements
- Design – Transformation of requirements into rules and masks
- Development – Transformation of Design into Code
- Test – Understanding of all the assumptions of the previous teams and interpretation of the actual (implicit) wishes and expectations of customers and operations in order to be able to submit somewhat reliable evaluation of often undefined quality criteria. Nothing short of mind-reading, art and magic.

I promised to show you what Google thinks of Testing and Advanced Automation



why is testing

why is testing **important**

why is testing **important in software development**

why is testing **necessary**

why is testing **results important**



Advanced automation



[Advanced Automation Technologies](#)

www.aatinc-md.com/ ▼

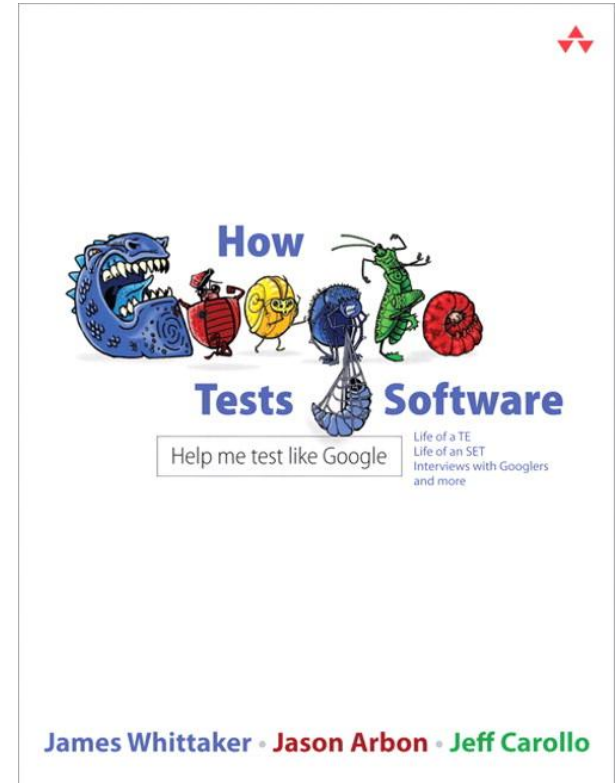
Welcome to **Advanced Automation Technologies, Inc. (AAT)**, a team of motivated and highly qualified professionals who are committed to a different and ...

Ok, seriously now

How Google Tests Software

“Quality is not equal to test. Quality is achieved by putting development and test into a blender and mixing them until one is indistinguishable from the other.”

“At Google, software testing is part of a centralised organisation called Engineering Productivity.”



DevOps

Continuous Integration and
Code Coverage Analytics

Crowd
Testing

Asset
Reuse

Automate
White-Box

Certify
Projects

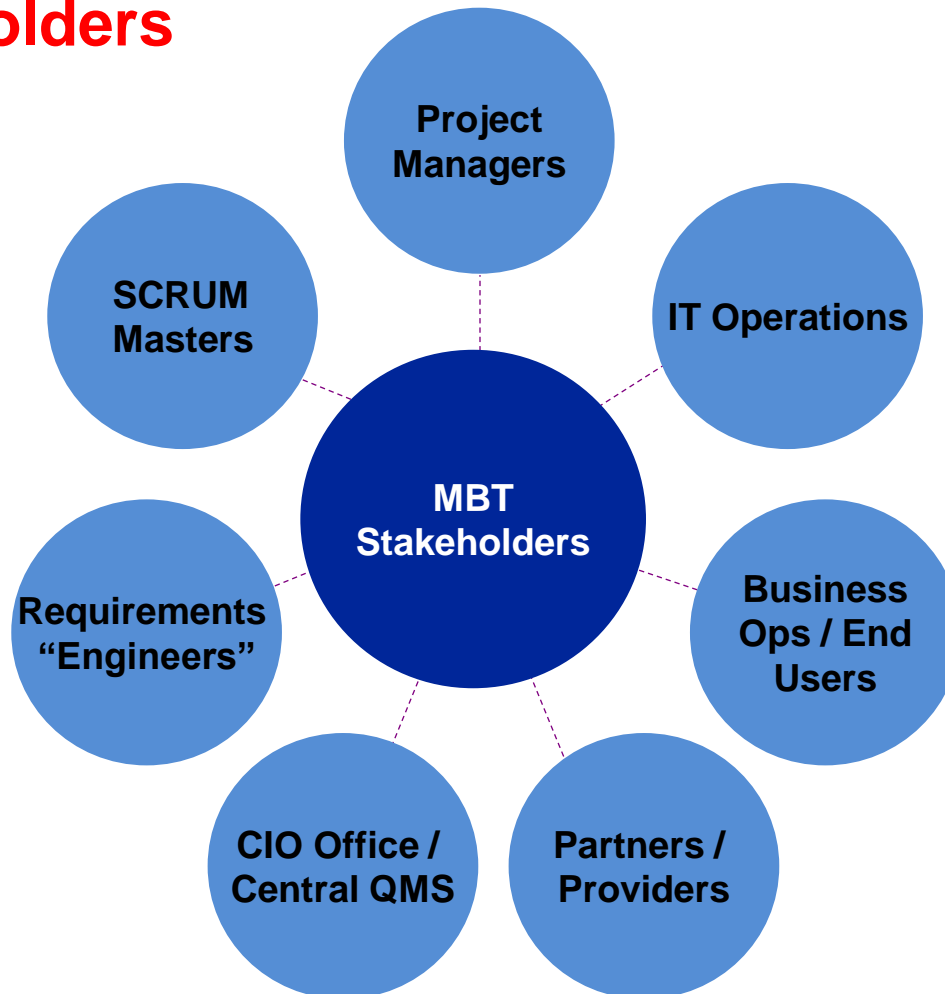
Dependency
Analysis

How do we motivate testers?



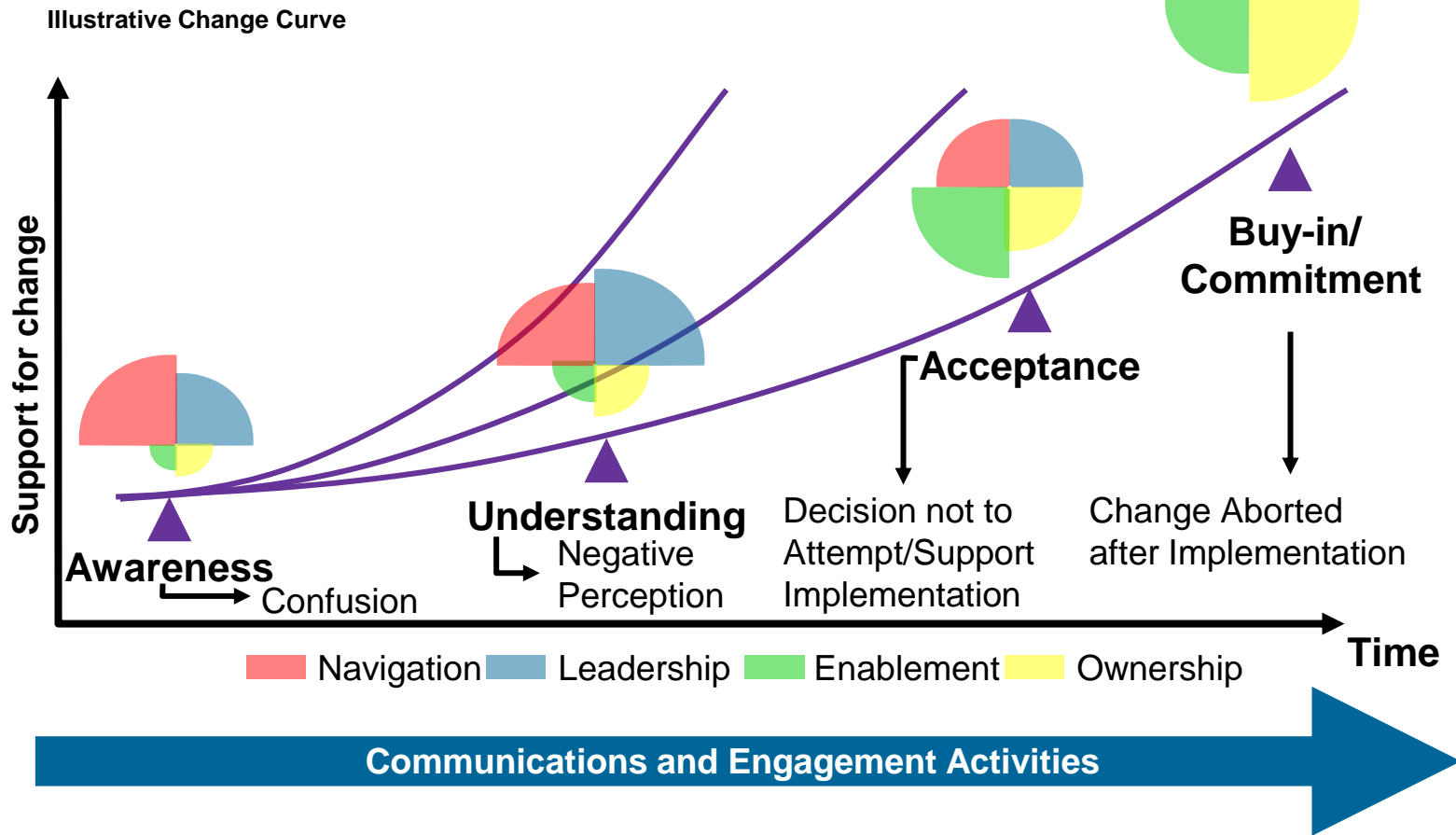
So we have motivated testers – what about our stakeholders?

MBT Stakeholders



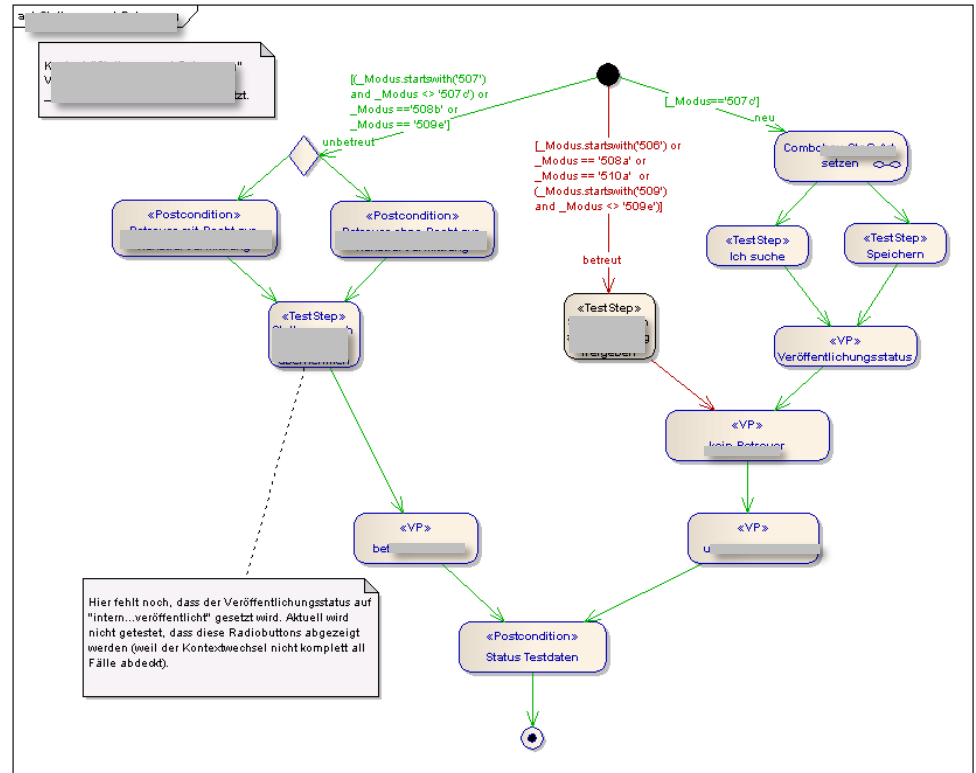
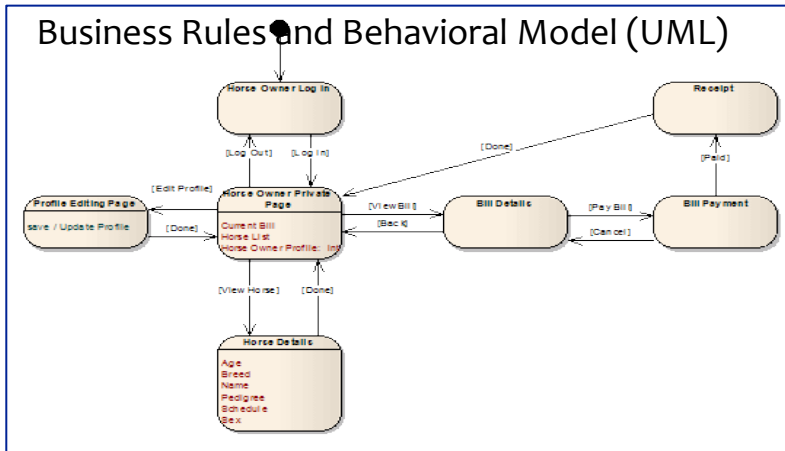
Manage the change before the ice melts

Change and Process Improvement



What does this mean now for the people involved?

- Requirements – Transformation of customer expectations into formal(ized) requirements



Conditions	Rules				
< 5 years	✓	✗	✗	✗	✗
>= 5 and < 18	✗	✓	✗	✗	✗
>= 18 and < 55 with concession card	✗	✗	✓	✗	✗
>= 18 and < 55 no concession card	✗	✗	✗	✓	✗
>= 55	✗	✗	✗	✗	✓

What does this mean now for the people involved?

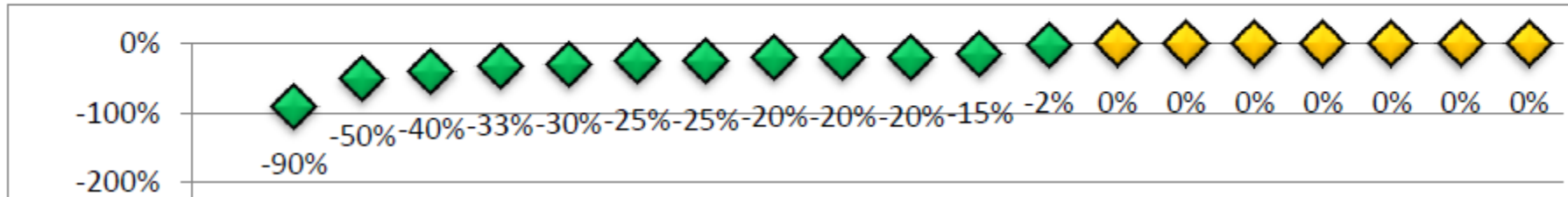
Implications for testers

- Modeling system behavior should not replace business and system requirements analysis – it's still a tester interpreting requirements
- Abstraction skills necessary – identify and stick to a specific level of abstraction
- Every good tester has been doing modeling already – just give them the right tool and motivation
- Review of generated tests is more important than ever
- For inexperienced teams, highlight the necessity to QA the code and test generators

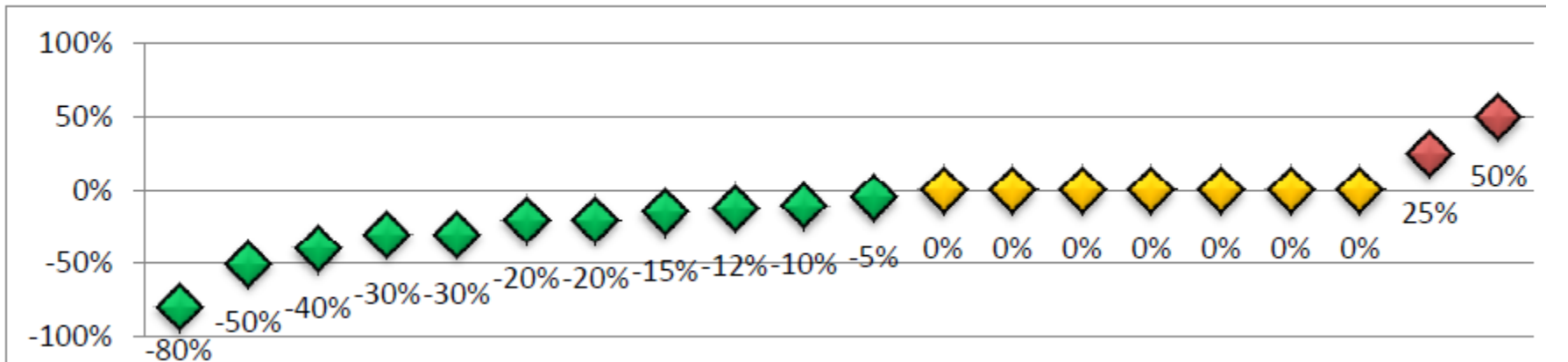
What does the industry say about benefits?

Why are we not convinced yet?

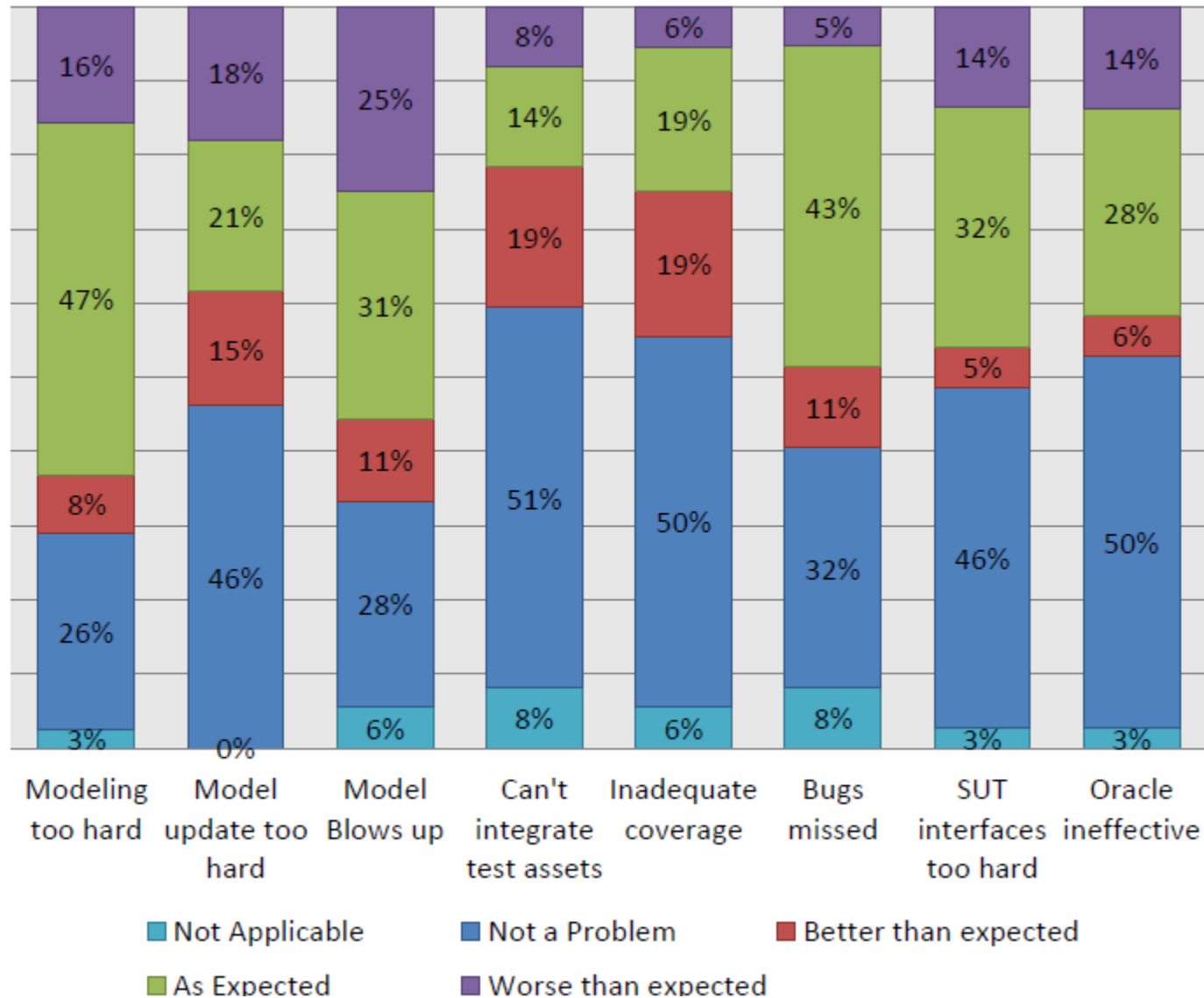
What is your best estimate of the percent increase or decrease in escaped bugs after testing with MBT?



What is your best estimate of the percent increase or decrease in testing costs with MBT?



and drawbacks



From “MBT in the Testing Ecosystem” to “MBT is the Testing Ecosystem”

Making MBT mainstream

- 1 ▪ UML Training or tools are not enough - motivation
- 2 ▪ Change Management, especially towards business
- 3 ▪ Focus on the ability to detect inconsistencies and conflicts in requirements early – but understand your stakeholders, not always is this transparency welcomed
- 4 ▪ For legacy applications build regression models first and use change requests coming in to build a broader model
- 5 ▪ Stay pragmatic in industries where this is feasible

But never forget:

“Every good tester has the heart of a developer.

In a jar.

On their desk.”