

INTRODUCTION

TEST

PBA SOFTWAREUDVIKLING /
BSC SOFTWARE DEVELOPMENT

Christian Nielsen cnls@cphbusiness.dk

Tine Marbjerg tm@cphbusiness.dk

SPRING 2019



TODAY'S TOPICS

- **Introduction to the course**
 - Course objectives (curriculum)
 - Semester plan
 - Exam
 - Literature
 - Assignments and activities
- **Introduction to testing**
 - Setting the scene
 - Introducing Developer Testing (Lit: Developer Testing by Alexander Tarlinder)
 - Testing Activities - in particular Developer Testing (chap 1)
 - Testing Objectives, Styles and Roles (chap 2)
 - The Testing Vocabulary (chap 3)

WHAT IS THE PURPOSE OF TESTING?

- What is the purpose of testing?
- What is the outcome of testing?



LESSON 1: TESTING CAN SHOW THE PRESENCE OF BUGS

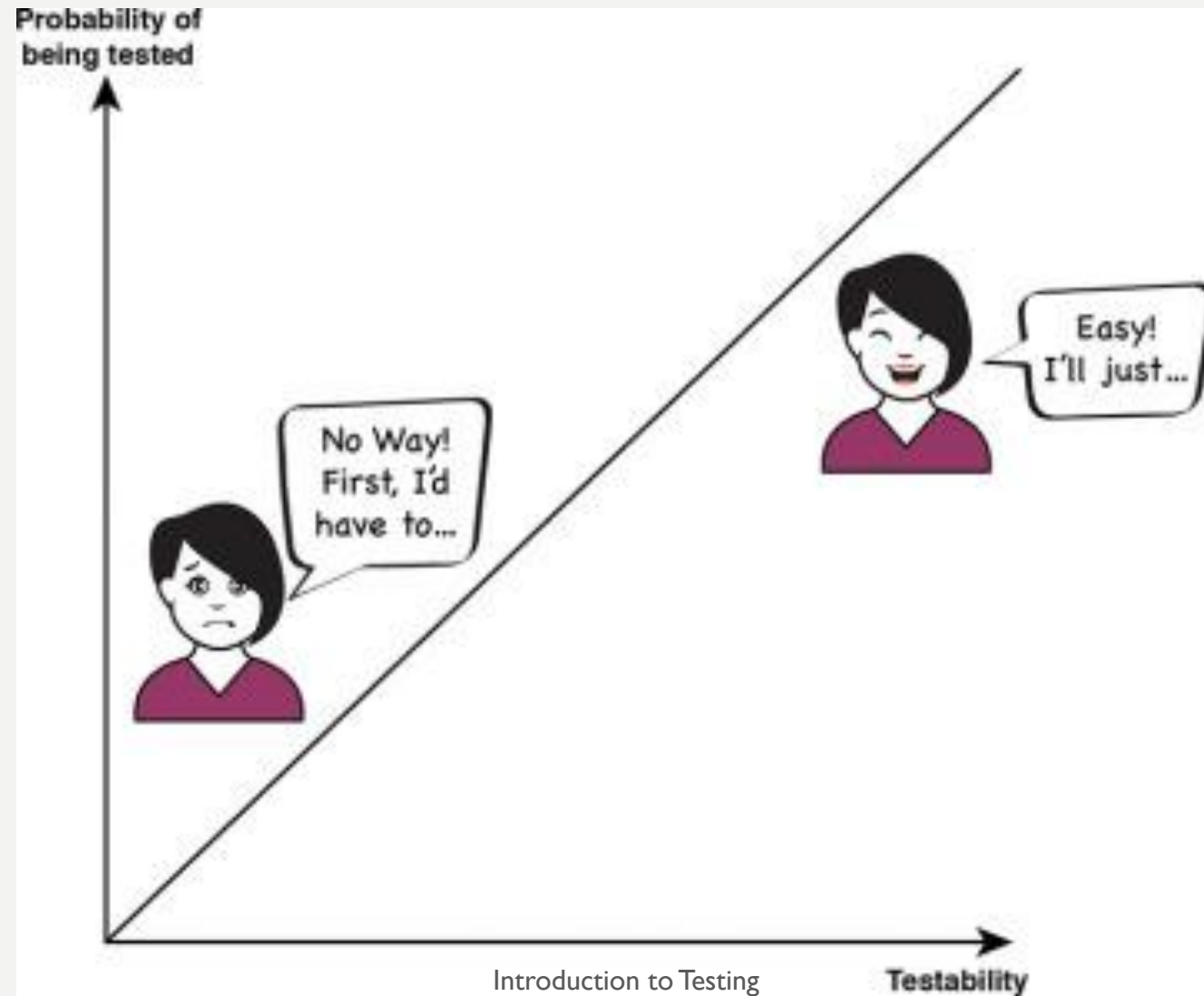
Never the absence!



LESSON 2: QUALITY MUST BE BAKED INTO SOFTWARE

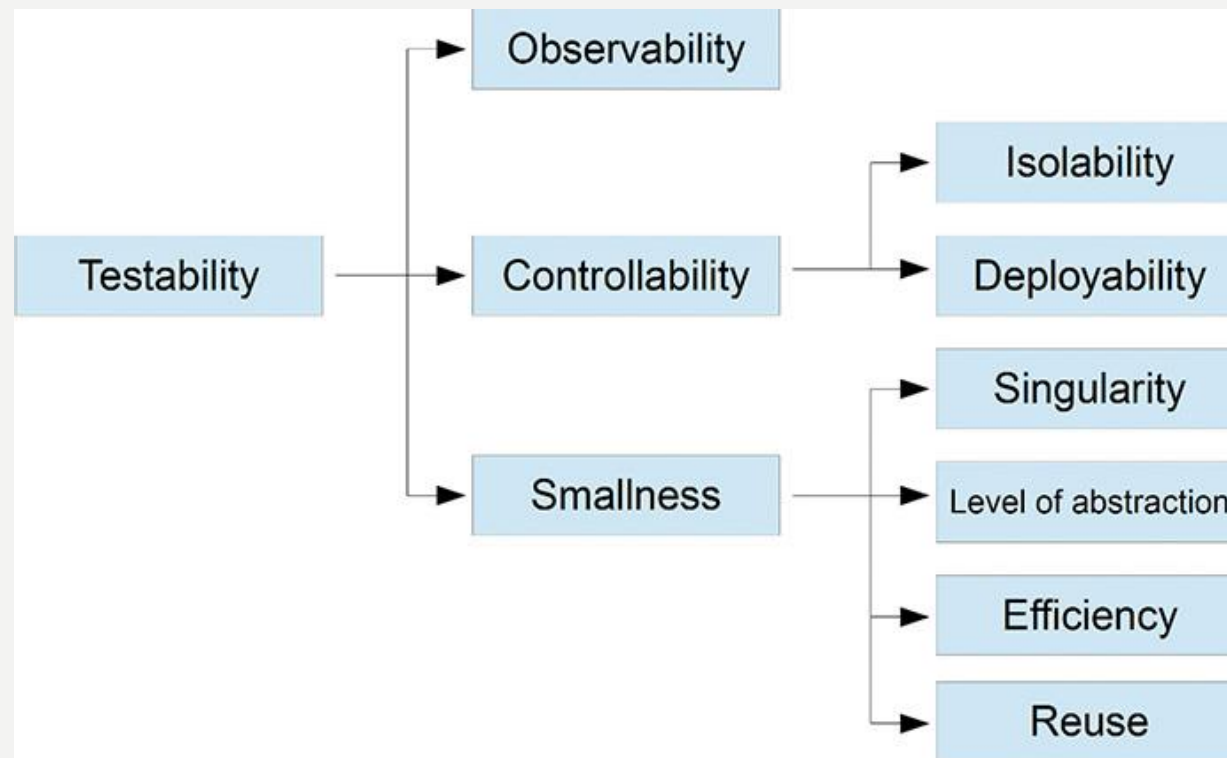


TESTABLE SOFTWARE

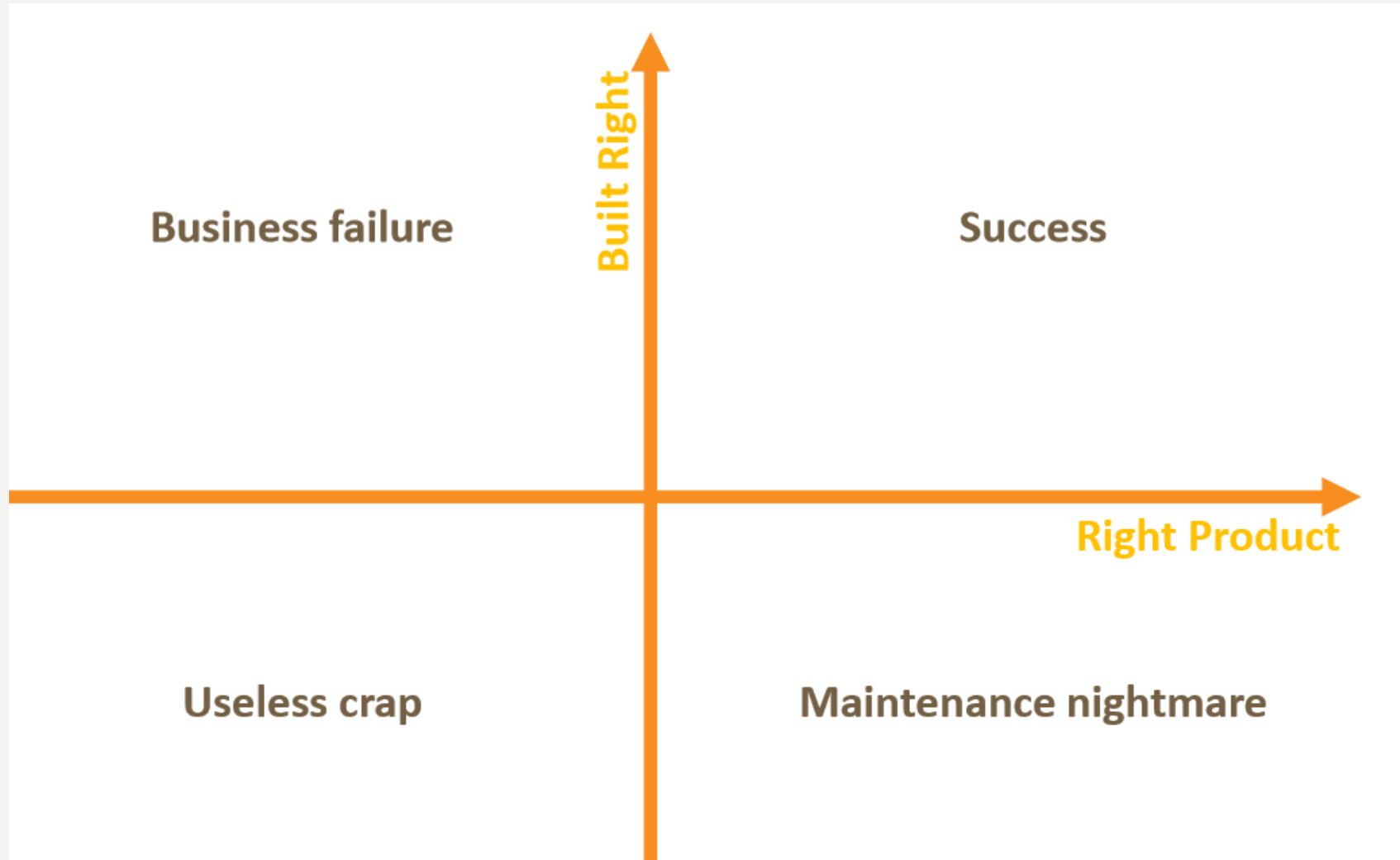


TESTABILITY DEFINED

- Testability is a quality attribute among other “ilities” like reliability, maintainability, and usability.
- **Observability** and **controllability** are the two cornerstones of testability



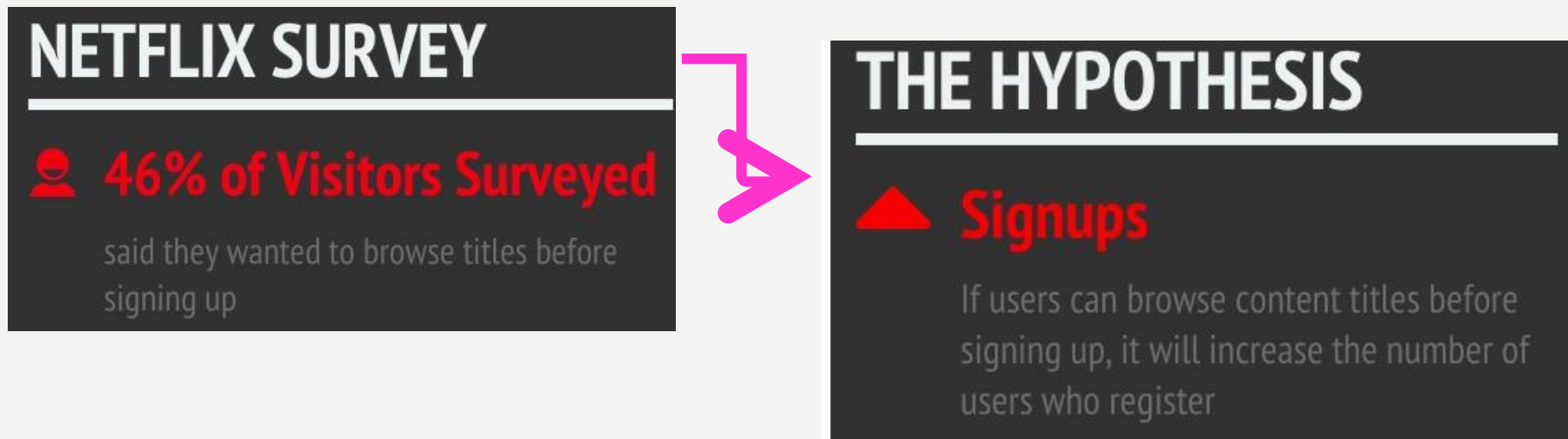
LESSON 3: RIGHT THING DONE RIGHT



Source: Adzic, Gojko - Specification by Example

NETFLIX EXAMPLE 1/3

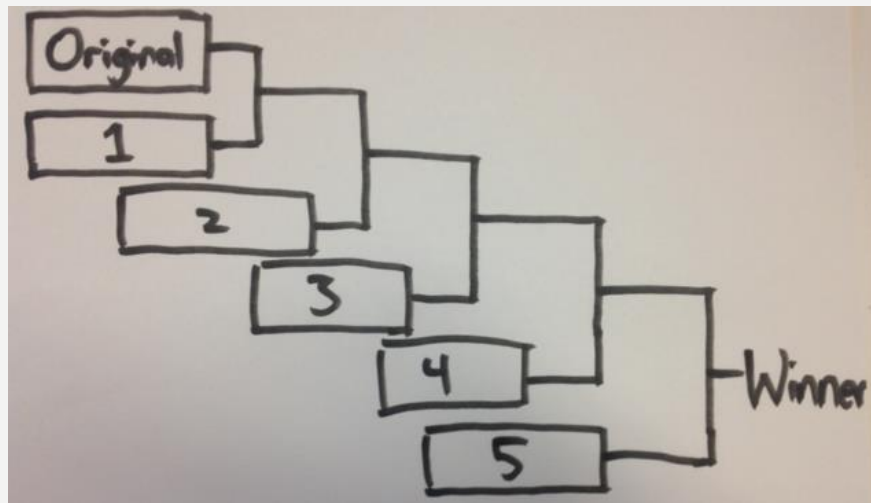
“Come on Netflix. Why can’t I browse titles before I sign up?”



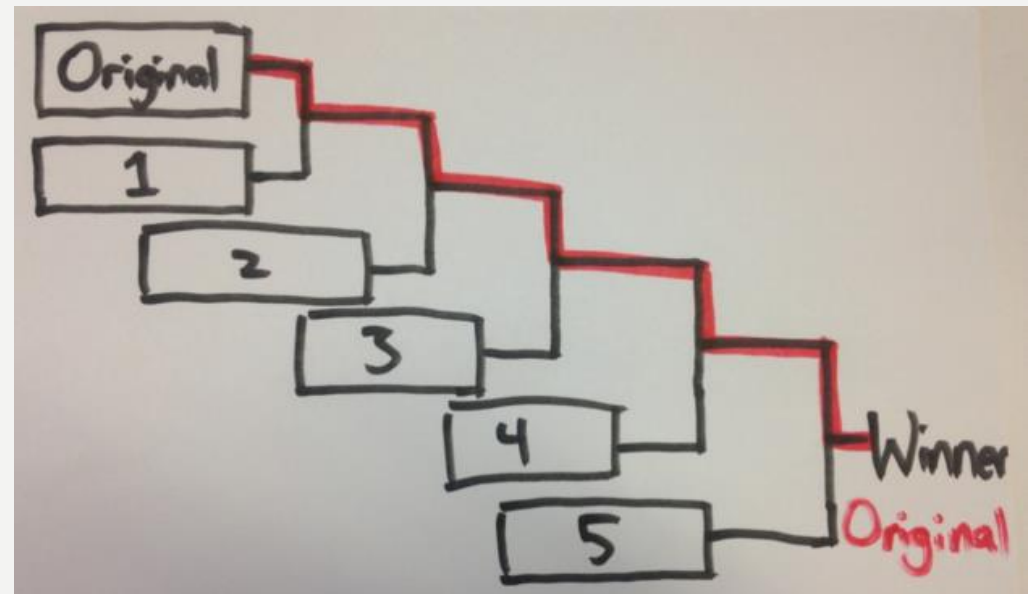
The Test Results Netflix Never Expected

NETFLIX EXAMPLE 2 /3

Netflix made experiments ...



As they ran the experiments, they found that the original page (without title searching abilities) beat all the other variants



NETFLIX EXAMPLE 3 /3

Conclusion 1

Don't Confuse the Meal With the Menu.

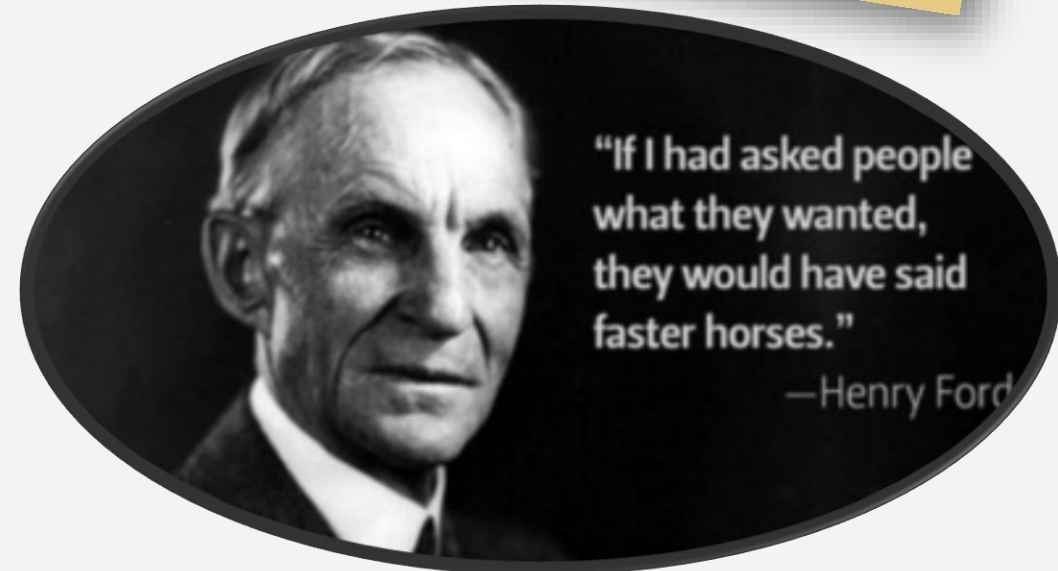
- *Netflix is all about the experience*
- *Simple choices*



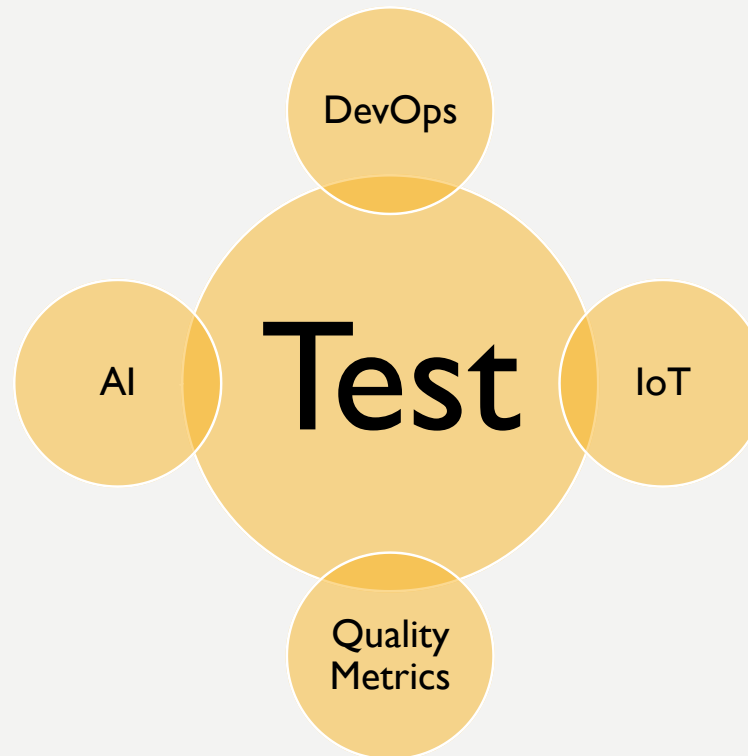
Conclusion 2

Users Don't Always Know What They Want

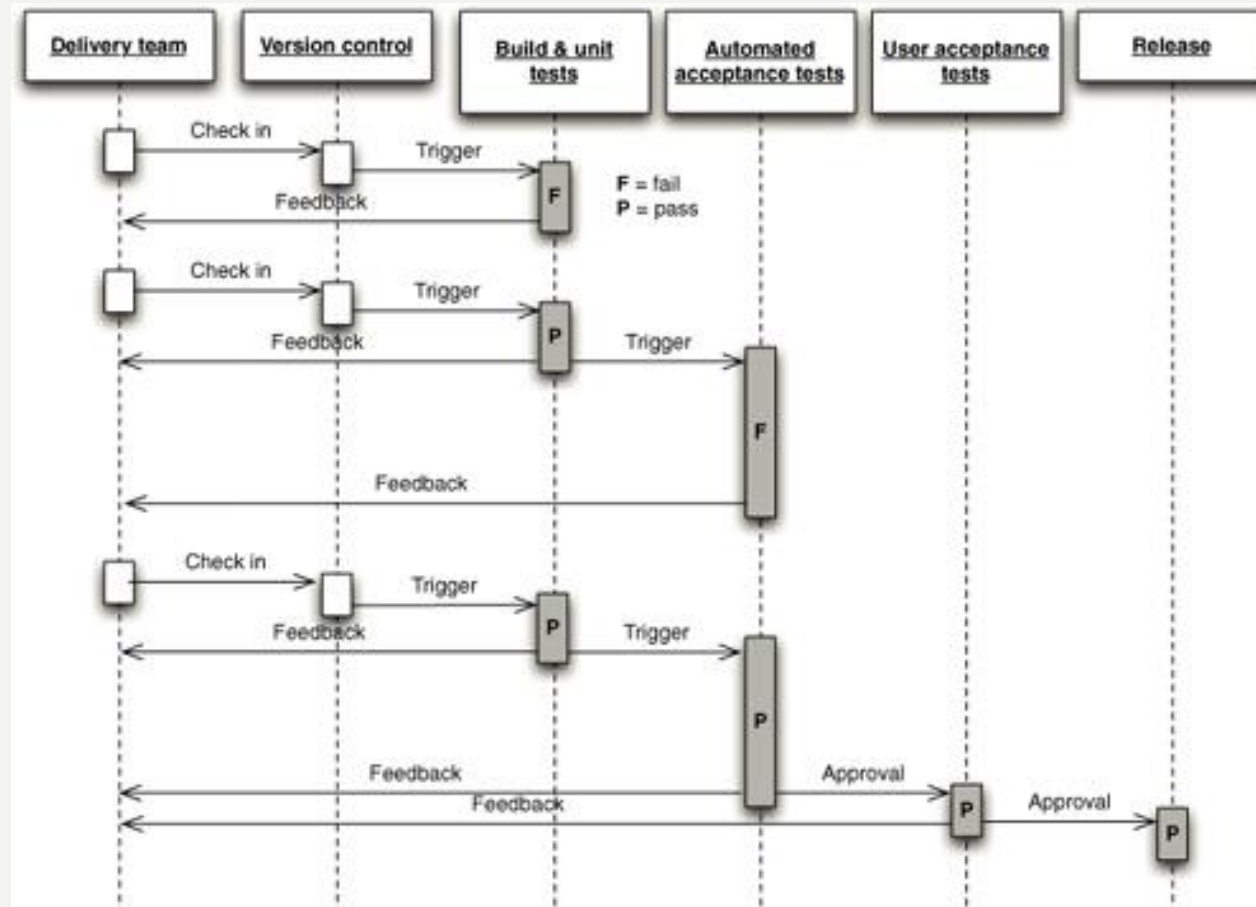
"Your assumptions are your windows on the world. Scrub them off every once in awhile, or the light won't come in." – Isaac Asimov



WHERE IS TEST HEADING / TEST CHALLENGES?



CHANGES MOVING THROUGH THE DEPLOYMENT PIPELINE



Source: Continuous Delivery by Jez Humble & David Farley

THE DEPLOYMENT PIPELINE



Source: Continuous Delivery by Jez Humble & David Farley

DEPLOYMENT ANTI-PATTERNS

- Deploying software manually
- Deploying to a production-like environment only after development is complete
- Manual configuration management of production environments

JOB ADVERTISEMENT YOU SEE (1 WEEK AGO)

*Your flair for **test automation** meets our passion for **Continuous Delivery***

- Are you motivated by the prospect of enabling Continuous Delivery through **Continuous Testing, test automation** and **Shift-left testing**?
- You will thrive in our **cross-functional teams** where you will be responsible for **automating and accelerating our testing processes**, paving the way for frequent, fast and focused deliveries.
- Ensuring tests are automatically executed as part of the deployment pipeline, your **integrated dashboards will give instant visibility into the risk inherent in any potential delivery**.

JOB ADVERTISEMENT CONT.

Responsibilities:

- Ensure that **acceptance criteria** of user stories are met prior to release
- Ensure that deliveries satisfy **non-functional requirements**
- Understand and **manage the risk** in every potential delivery
- Continuously improve and evolve our **test strategy**
- Ensure availability of up-to-date **test data**
- Implement **automated UI and API testing**
- Execute **manual** exploratory, usability and acceptance testing
- Support other Squad members with your QA expertise
- Build dynamic dashboards to **visualise quality metric**

CURRICULUMS IN GENERAL—

LEARNING OBJECTIVES HAVE DIFFERENT LEVELS

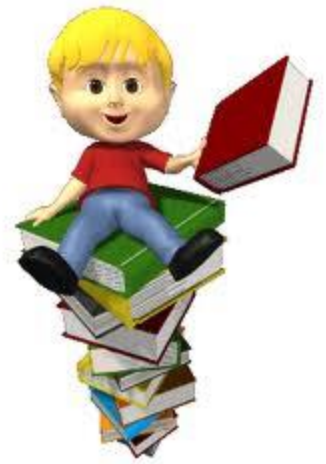


E X A M P L E

- Knowledge
 - **Know** about control structures as non sequential control mechanism
- Skills
 - **Design** a “for loop” from scratch to solve a particular problem
- Competences
 - **Evaluate** the quality (e.g. efficiency) of a “for loop” vs. other control structures and use appropriately

PBA Soft Curriculum on CphBusiness web site

TEST COURSE OBJECTIVES – KNOWLEDGE LEVEL



- Central test strategies and tests models and their role in the system engineering process
- Test as integral part of a development project
- Different test types

COURSE OBJECTIVES –

SKILL LEVEL



- Ensure traceability between system requirements and testing at all levels
- Apply black-box and white-box testing techniques
- Apply various criteria for the degree of test coverage
- Use techniques for verification and validation
- Use techniques and tools for automated testing
- Build systems to manage testing and incident management

COURSE OBJECTIVES – COMPETENCE LEVEL



- Define, plan and execute testing in a development project that matches the project's quality requirements
- Plan and manage the implementation of internal and external testing of software systems.
- Design for testability

SUMMING UP TEST COMPETENCES

- Craftsman skills
- Tests as driver of requirements
- Testing as integrated activity in software engineering
- Efficiency – we need automation

EXAM

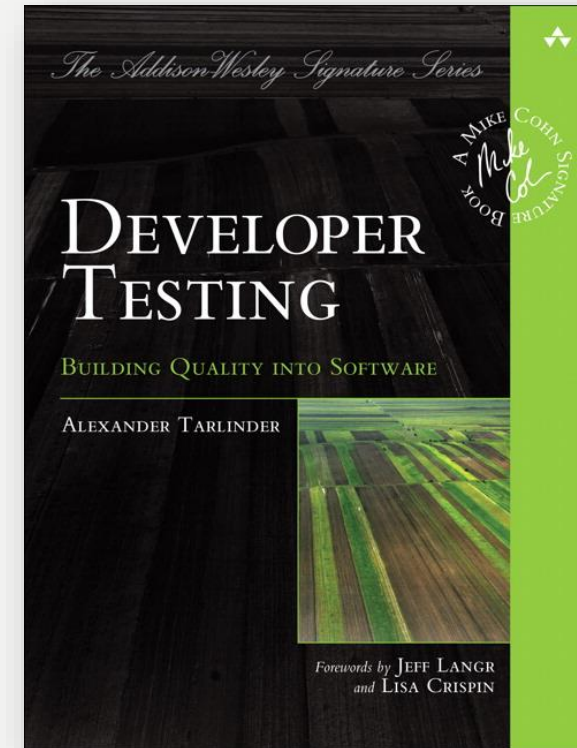


- 25 minutes individual oral exam
- No preparation time
- You pick a question which has both theoretical and practical perspective
- Each question relate to assignments or semester project

- *ISTQB certification at DIT (optional)*
 - Materials
 - Book

COURSE RESOURCES

- [Developer Testing](#) by Alexander Tarlinder
- Online resources



PBA SOFT TEST ON GITHUB

- <https://datsoftlyngby.github.io/soft2019spring/> (GENERAL)
- https://datsoftlyngby.github.io/soft2019spring/TEST_plan.html (TEST)

THE TEST CHAMPION (TESTER 2.0)



THE TEST CHAMPION EXPLAINED

Testing

- Quality Champions *must* have testing skills.
 - Testing techniques, tools, defect handling

Development

- The Quality Champion can have a deep expertise in the tools and frameworks the developers are using: What are their capabilities, limitations, and what will the quality of the developers' automated checking be?

Product

- It's not uncommon for the Quality Champion to be the team member who has the most time to spend with the product owner and other stakeholders. This results in deep understanding of the domain and its business rules.

6 QUESTIONS TO YOU 😊

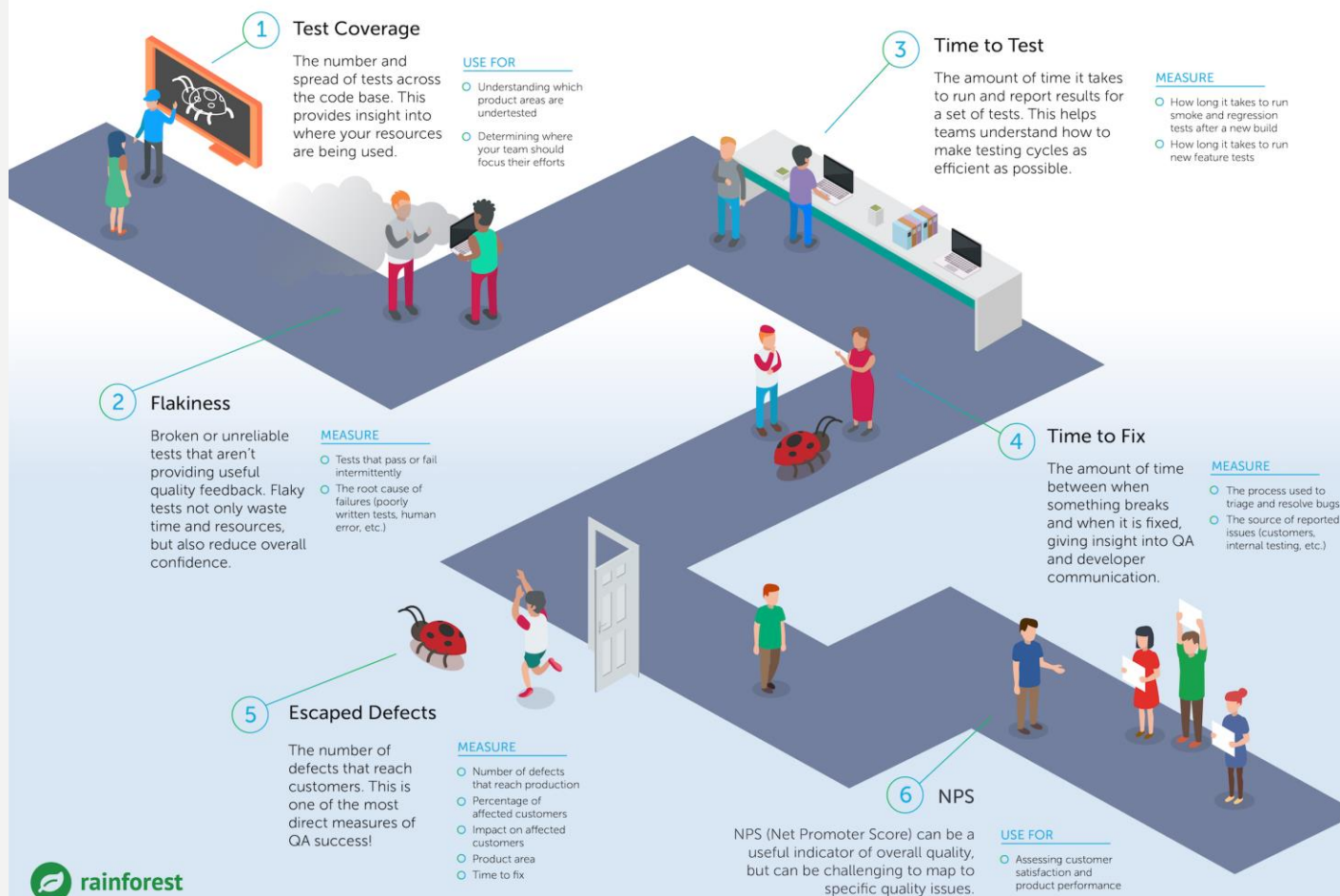
1. What is a bug (more formally called a defect)?
2. What is the difference between testing and Quality Assurance (QA)?
3. What is a smoke test?
4. What is a regression test?
5. What is a boundary test?
6. What does data driven testing mean?

QA METRICS EXAMPLE

Essential QA Metrics

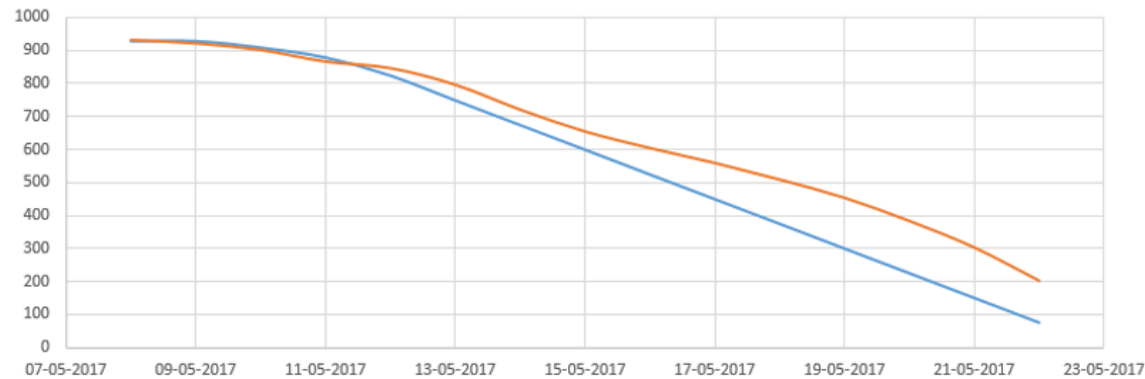
Strategic Measurements to Improve Quality Assurance Process

Collecting test metrics allows teams to use data to improve their software quality framework, understand product quality and increase the efficiency of software testing overall. Every team is different, but these 6 key metrics can help any team measure their quality goals more effectively.

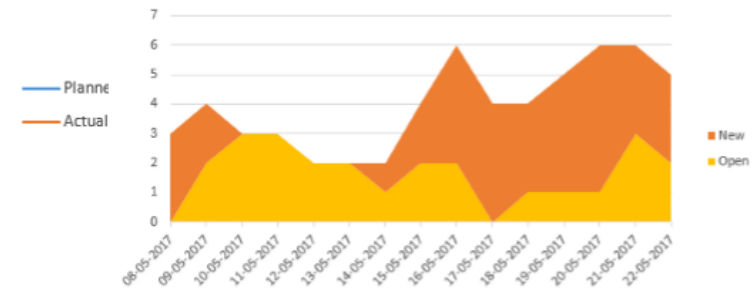


QA MEASURES – EXAMPLE (TESTHUSET)

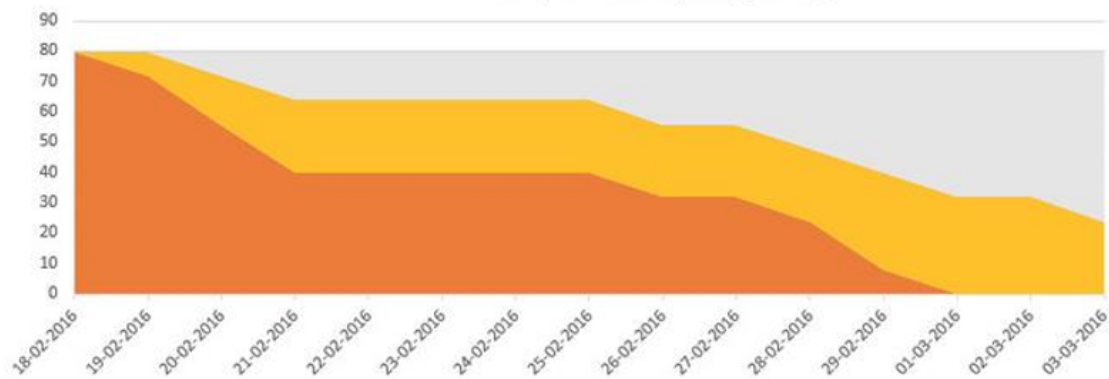
Story Burndown (estimated hours)



Unresolved bugs



Story burndown (story points)



Unresolved issues by type

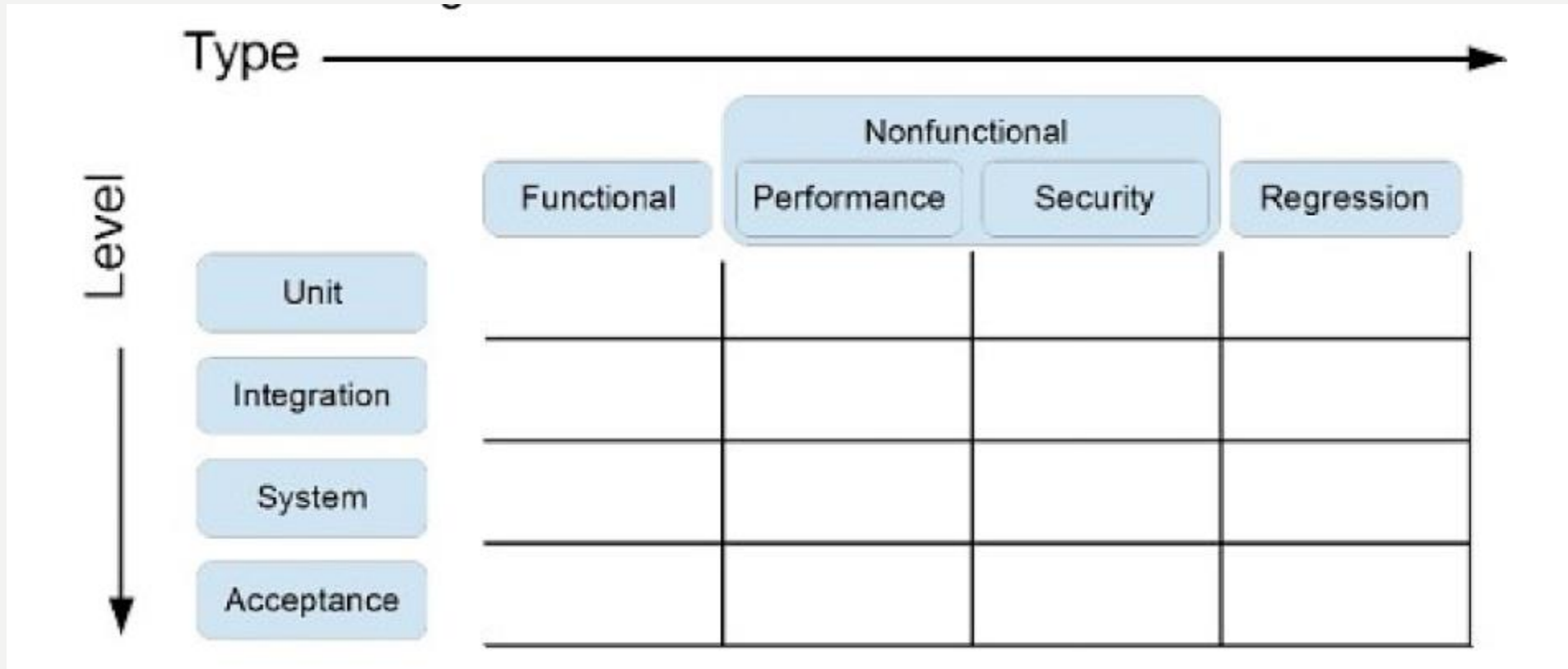


A decorative wavy line in yellow and white on the left side of the slide.

TEST COURSE MATERIAL

DEVELOPER TESTING CHAP 1-3

CLASSIFYING TESTS



Test levels & Test types (there are more non-functional test types)

MORE NON-FUNCTIONAL TEST TYPES

Testing of software characteristics

Typically how well or fast something can be done

- **U**sability testing
- **R**eliability testing (robustness & recoverability)
 - Stress testing
- **P**erformance testing (speed, throughout, scalability)
 - Load testing (scalability)
- **S**upportability
 - Portability testing
 - Maintainability testing

SOFTWARE LIFE CYCLE

- Test is not isolated activity in a software development project
 - The chosen software development model influences test organization
- Two camps of system development philosophy
 - Lightweight and fast methodologies (agile)
Time to market is central
 - Disciplined methodologies
Quality and reliability are central

WATERFALL

- Characteristics:
 - Development flows through distinct phases
 - No one has to deal with the consequences of their mistakes if different people in different phases
 - Testers have to make up lost time from previous phases
 - Time lag: it can take a long time for changes in requirements to filter through to the finished product

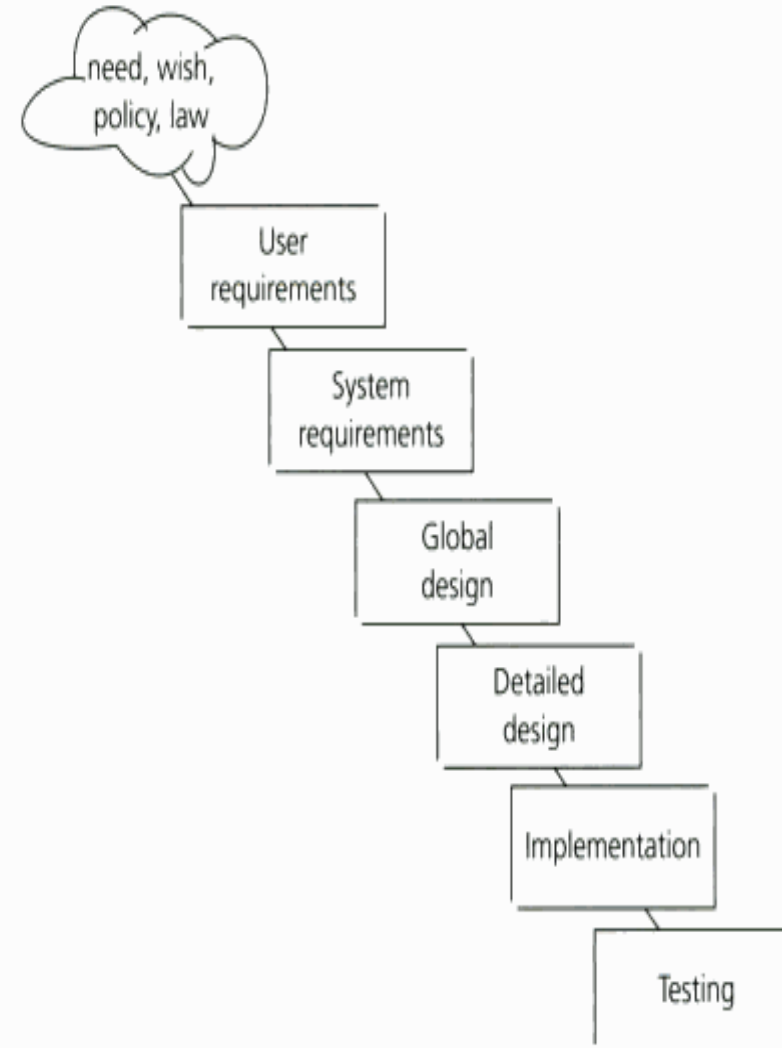


FIGURE 2.1 Waterfall model

V-MODEL: ENHANCEMENT OF WATERFALL

Characteristics:
- Earlier Planning
and Execution of
Tests

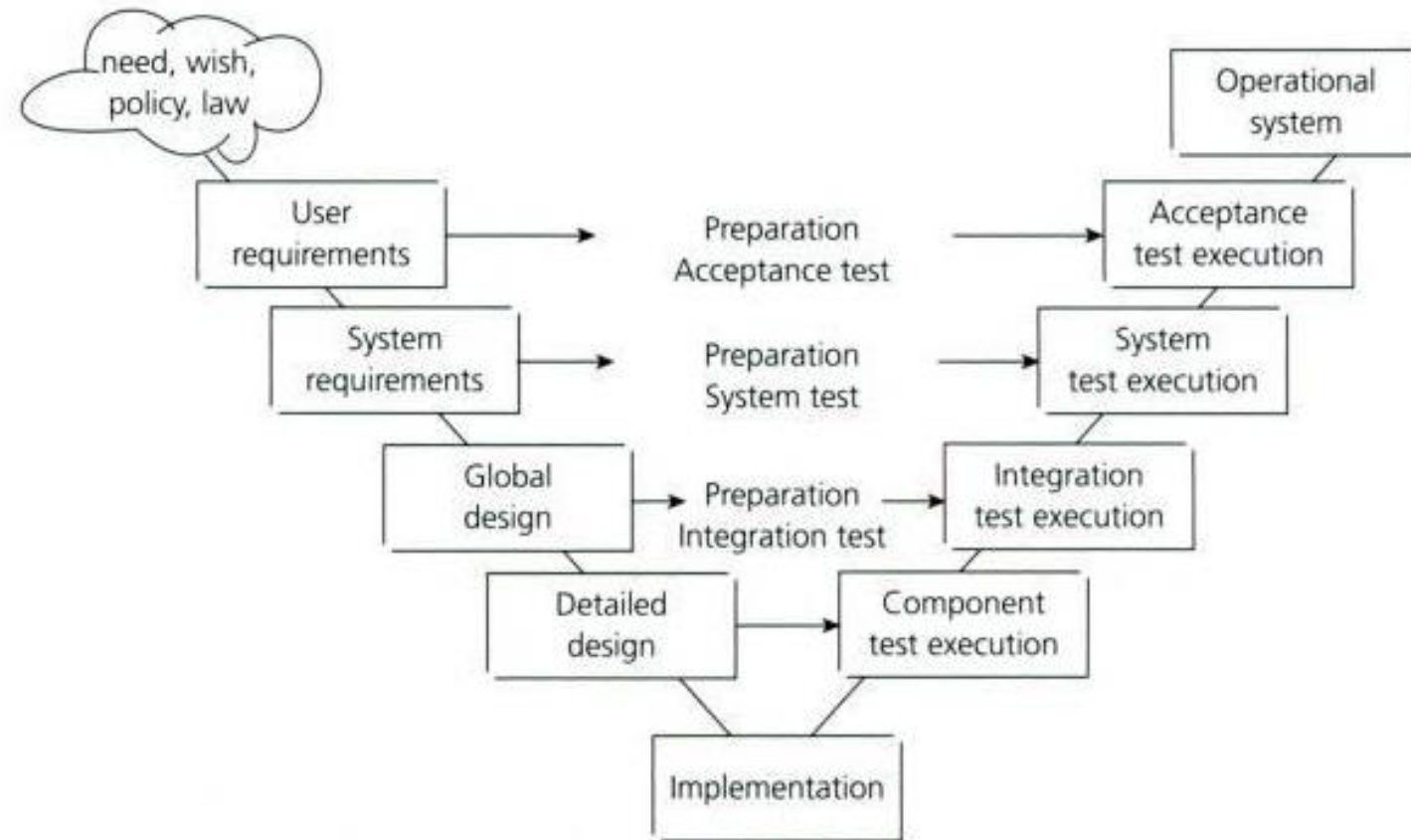
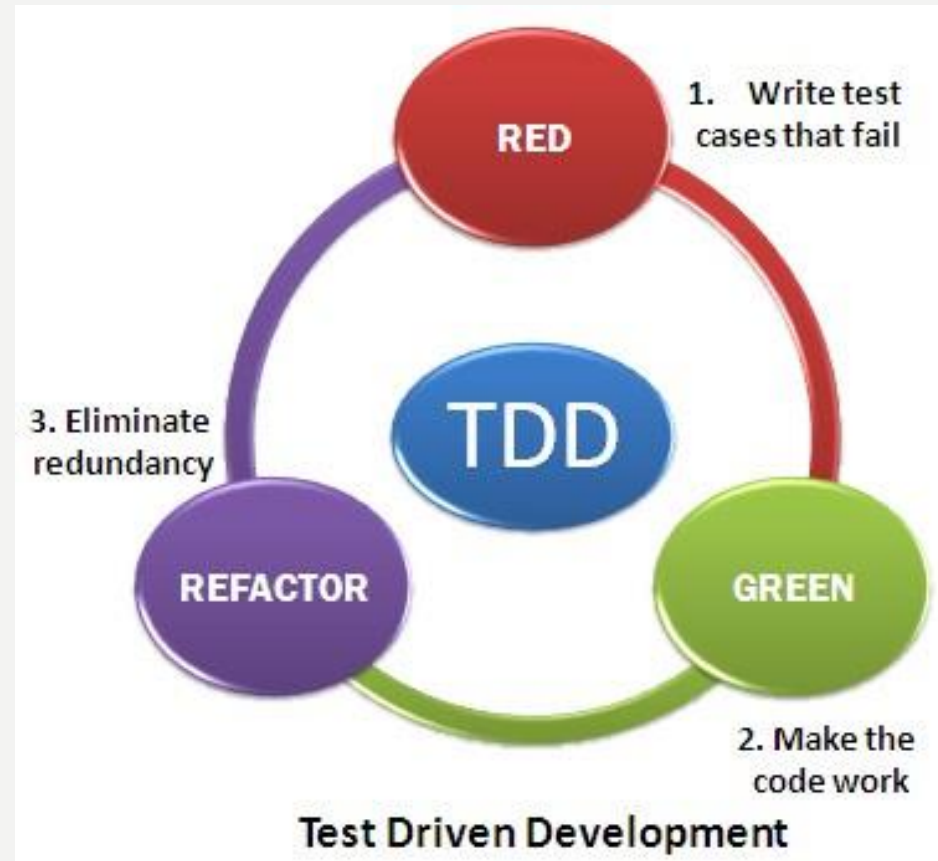
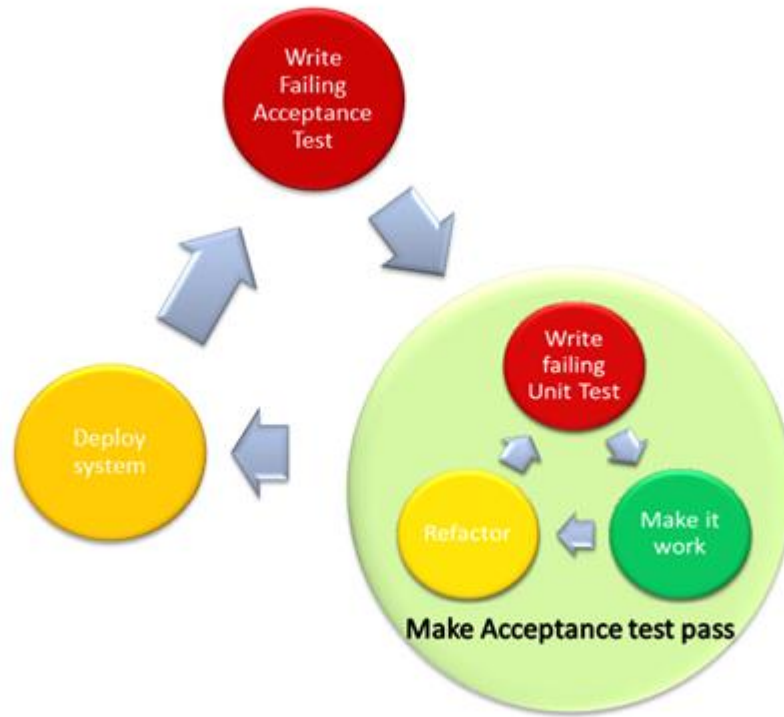


FIGURE 2.2 V-model

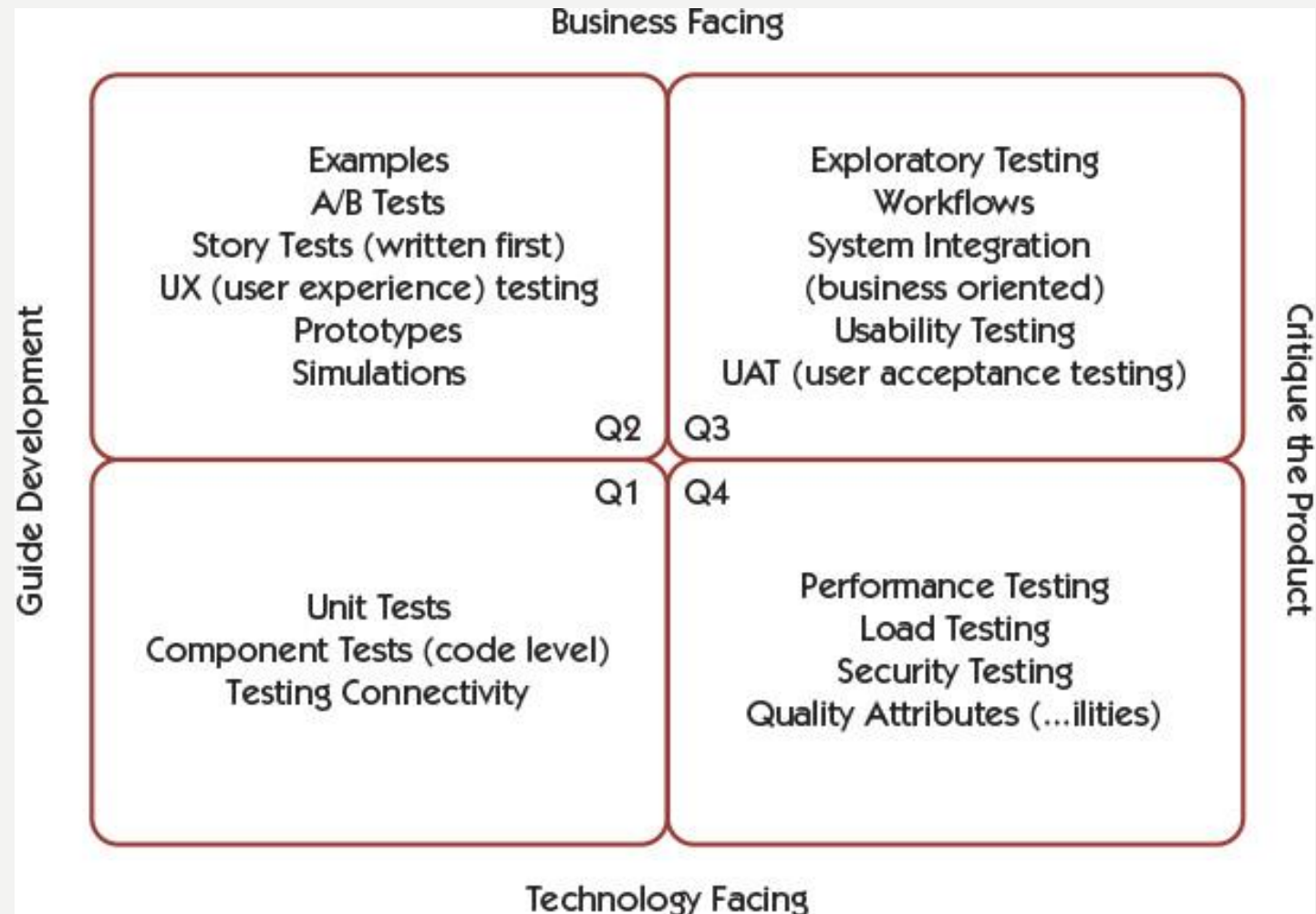
TEST DRIVEN DEVELOPMENT (TDD)



ACCEPTANCE TEST DRIVEN DEVELOPMENT (ATDD)



AGILE TESTING QUADRANTS



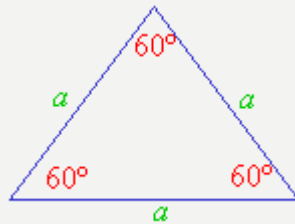
EXERCISE TIME



<code>

IMPLEMENT A TRIANGLE PROGRAM

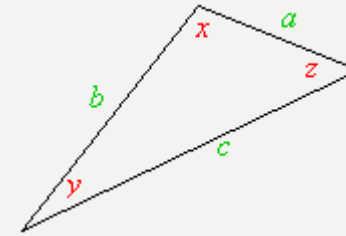
The program reads three integer values from an input dialog. The three values represent the lengths of the sides of a triangle. The program displays a message that states whether the triangle is scalene (ingen ens sider), isosceles (ligebenet), or equilateral (ligesidet)



An **equilateral triangle** has all three sides of equal length.



An **isosceles triangle** has two sides of equal length.



A **scalene triangle** has no sides of equal length.