



KOLLMORGEN



DATA
TJEJ

Welcome to our Software Quality Assurance Workshop!

We will start in a few minutes...

KOLLMORGEN



Welcome to a workshop about
Software Quality Assurance!

We will start in a few minutes...



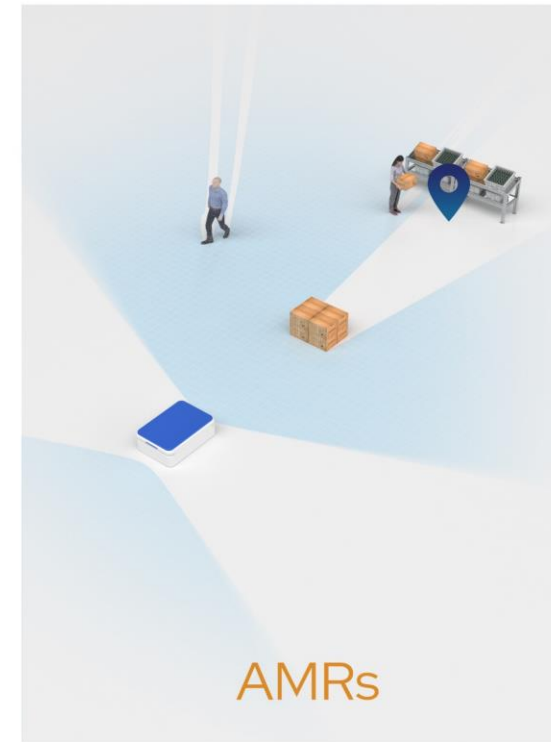
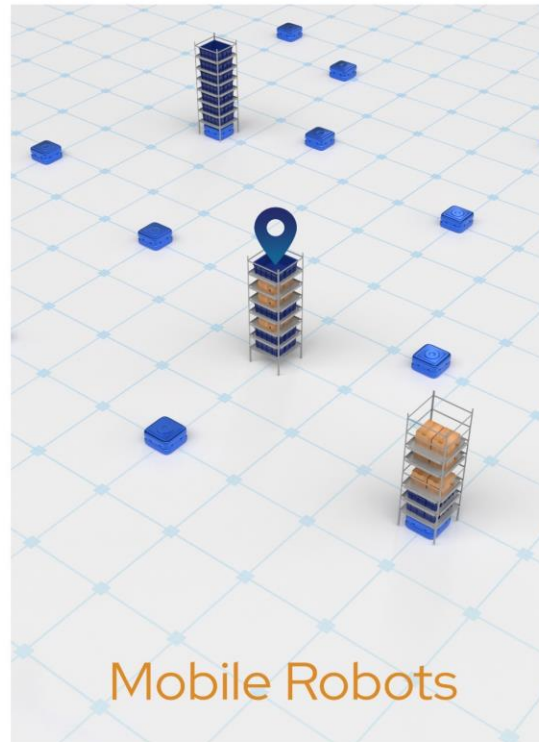
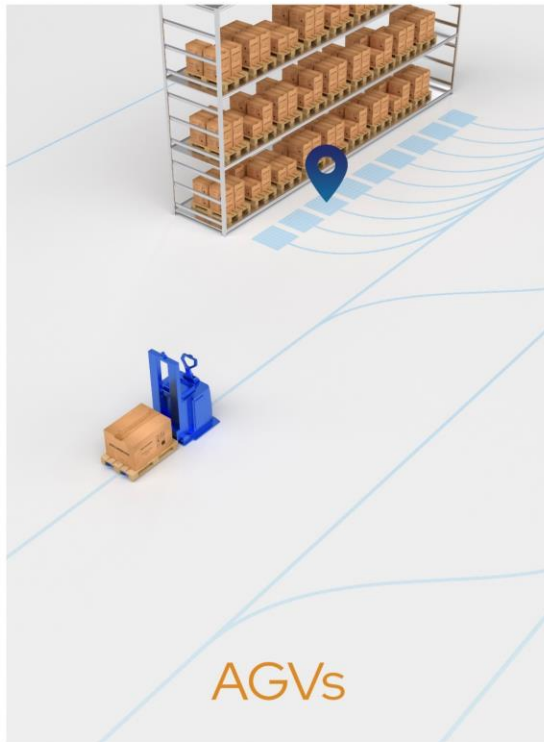
What do we do? 

We have provided technology for driverless material handling vehicles since 1972



KOLLMORGEN

The Right Solutions for Your Transport Needs



Kollmorgen is a global company with sites all over the world and HQ in Radford, USA



Kollmorgen Automation (AGV) are around 120 employees located in Sweden, China, USA, South Korea, Italy and Czech Republic.

HQ in Mölndal, Sweden where around 100 employees work within Product Development, Sales, Finance, Services & Support, IT and HR

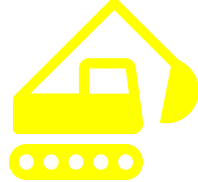
Scrum Teams

KOLLMORGEN



VEHICLE/Green

Software for vehicle motion control, sensors, and localization. Using C++, Python.



VEHICLE/Yellow

Hardware components such as vehicle controller, navigation sensors, and displays



STATIONARY/Red

Software and algorithms for fleet control, traffic control, and integration. Using C#, C/C++ etc.



TOOLS/Purple

Design, service and installation tools. Using C#, C++ etc.



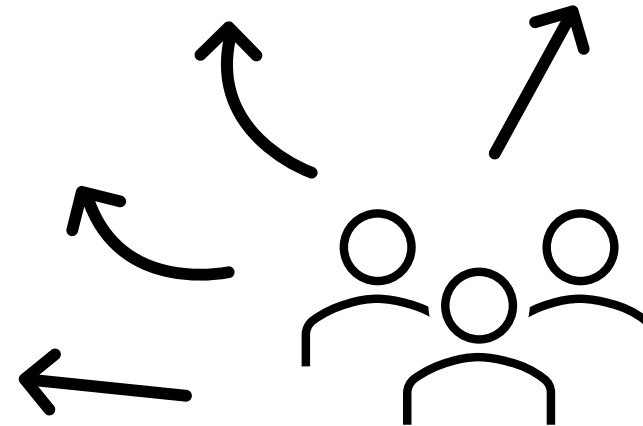
TOOLS/Deep Purple

Web based commissioning and operation tools. Using C#, Javascript etc.



SQA/Blue

Automatic regression testing and test platform. Using Python etc.

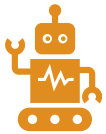


A part from developers, SQA engineers and System Architects, we do have shared resources such as Content Writers, UX, Product Owners & Product Managers for one or multiple teams

Today's Speakers

KOLLMORGEN

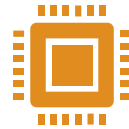
Pavithra Nagraj



SQA Engineer at
Kollmorgen



Computer Science,
India



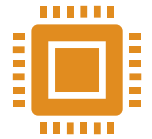
Frontend Developer,
React JS, Angular Js



SQA Engineer at
Kollmorgen, June, 2022



Summer Intern and Master
Thesis Student, Kollmorgen



Deep Purple team,
Web Based
Testing, DevOps



Purple Team,UI
Testing,Sikulix,Jenkins



K-drama,hiking



Tests
developing, executing and
maintaining tests, ensuring
quality of products



Fiction novels, French Com
edies and D & D

Today's Agenda

KOLLMORGEN

Introduction

Why is testing important?

- History of testing

- The impact of testing

- The future of testing

How to become a QA Engineer?

- Our Roadmap

- Proposed Roadmap

QA Fundamentals

- Testing Levels

- Testing Approaches

- Testing Types

Testing Quiz

Practical Task

- Case Study Introduction

- Exploratory Testing Time

- Report Defects

Introduction



Why is Quality Assurance Important

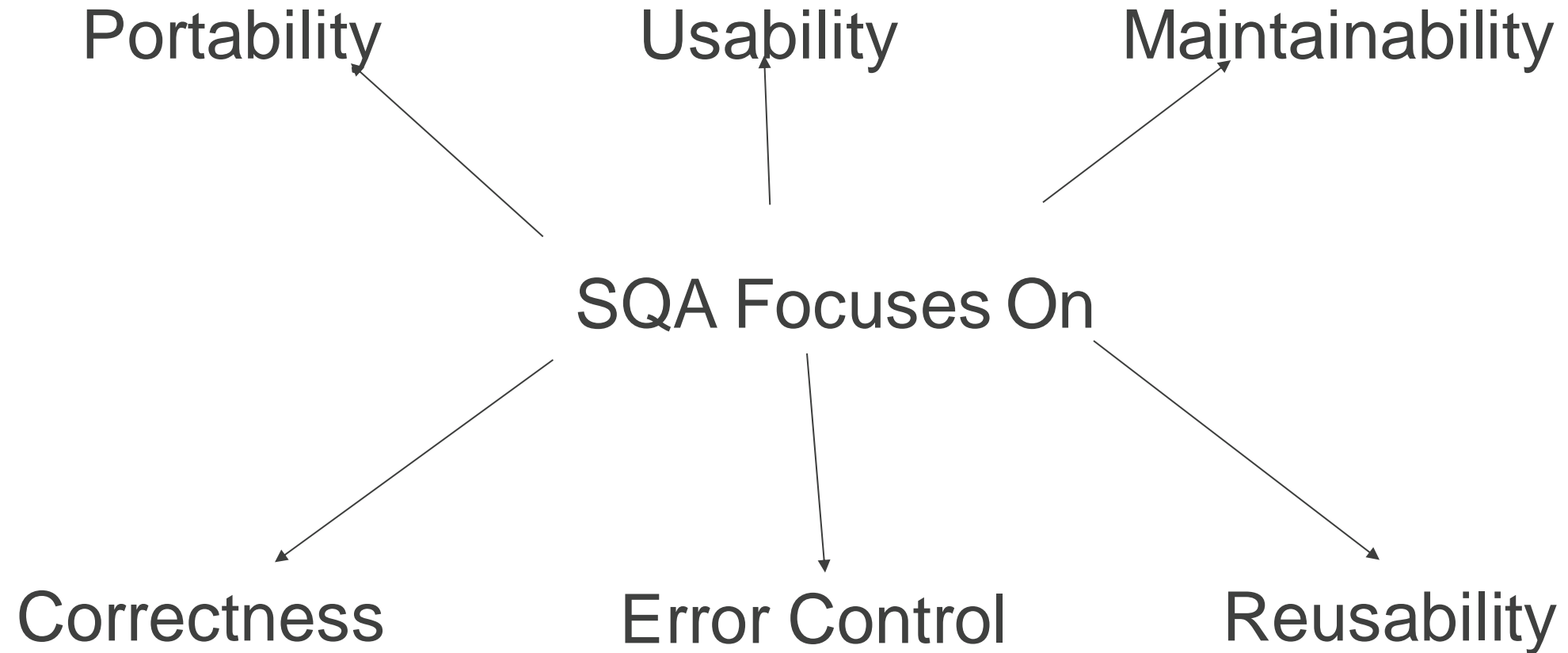


Quality assurance (QA) is a **systematic** process of determining whether a product or service meets **specified requirements**.

It is the set of activities which ensure **processes, procedures** as well as **standards** are suitable for the project and **implemented correctly**. Software Quality Assurance is a process which works parallel to development of software.

ISO 9000 is a series of three standards:





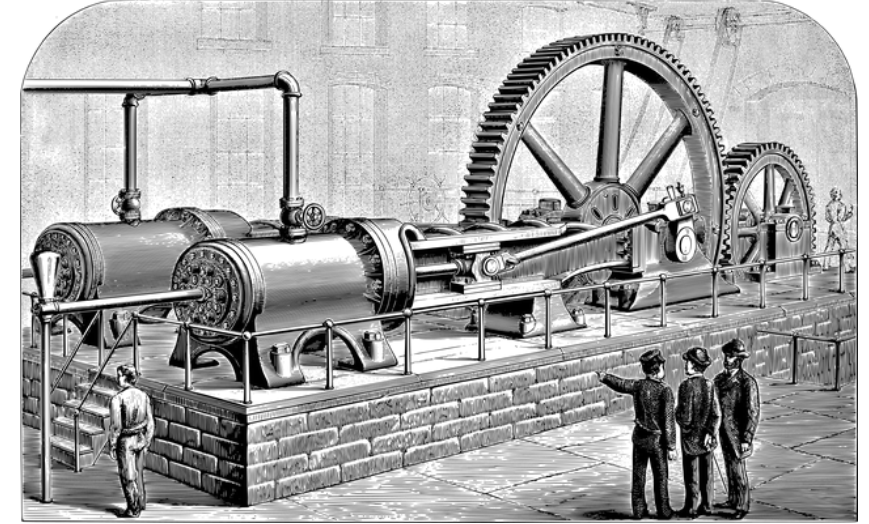
QA is always changing



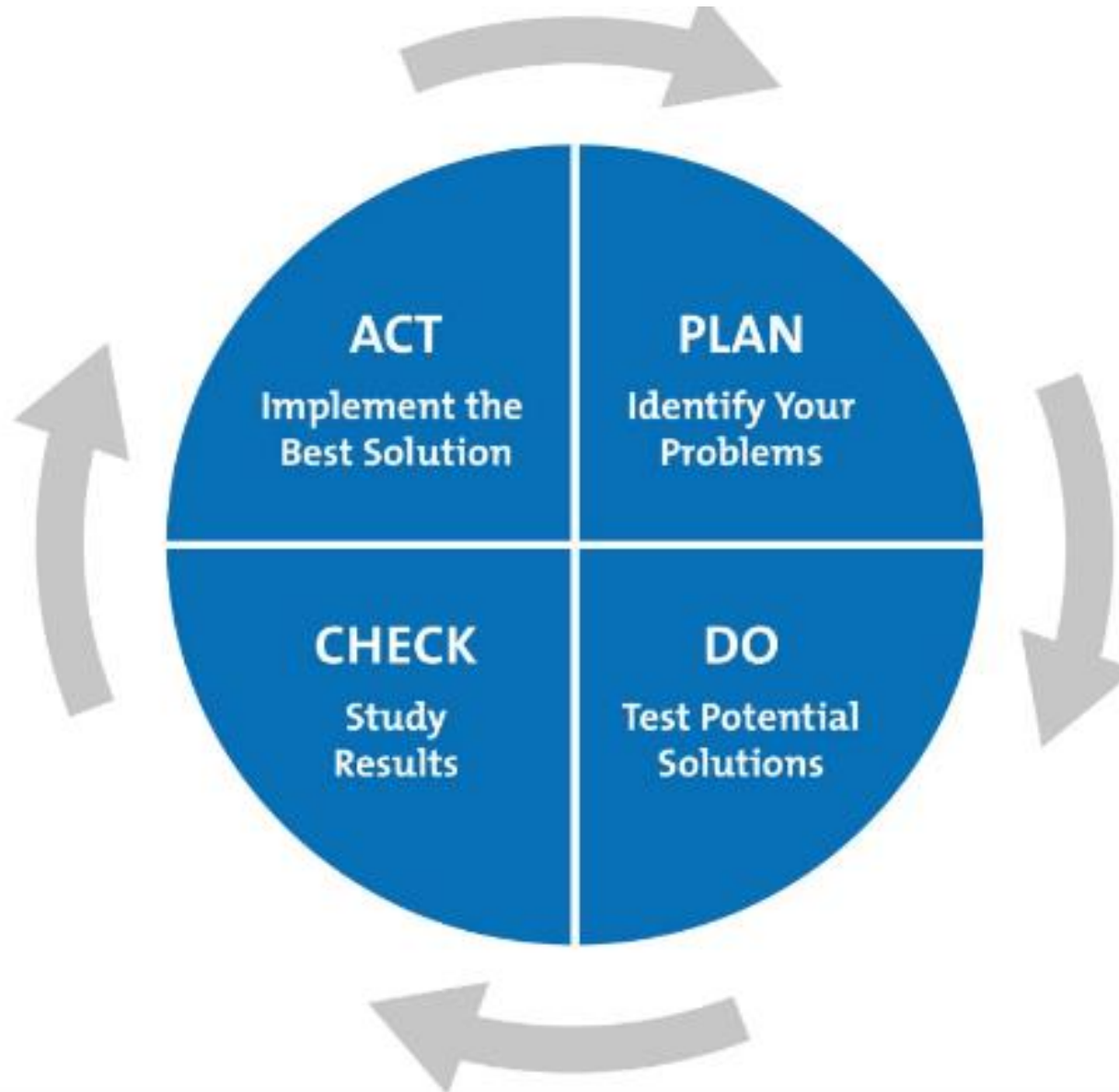
First Quality Assurance Attempts in Ancient Greece and Egypt



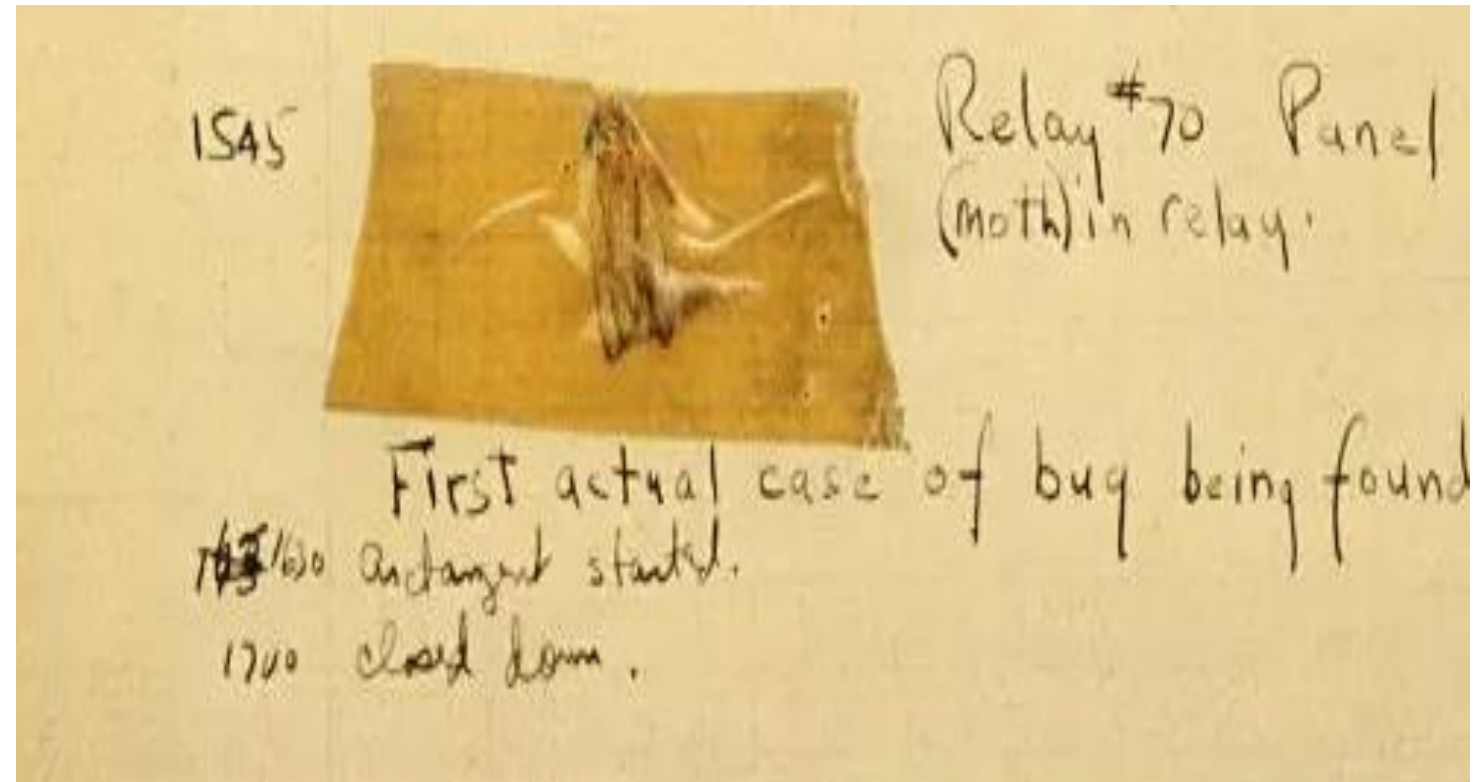
First Quality Standards, Medieval Guilds

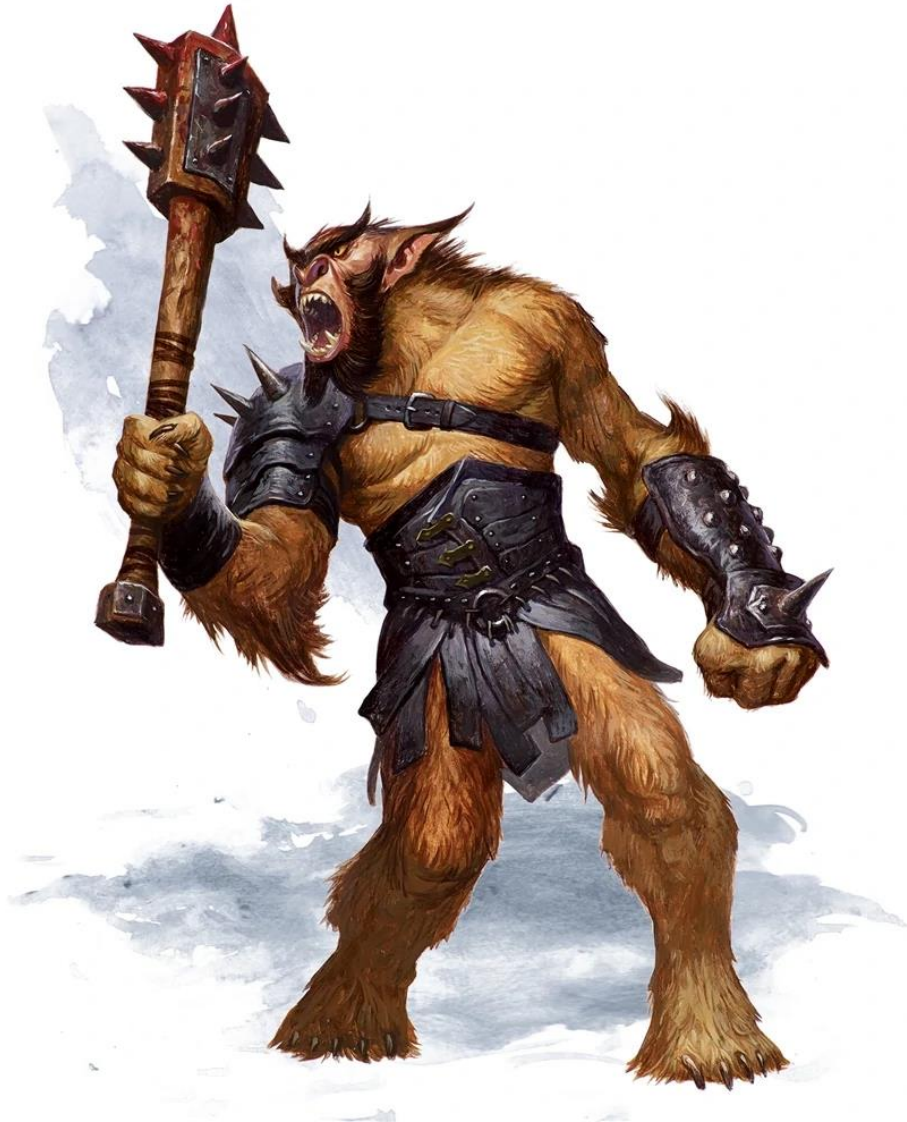


Industrial Revolution, mechanical engineering standards introduced by Frederick Winslow Taylor



In 1947, engineers working on the Mark II computer at Harvard University found a moth stuck in one of the components. This is just a fun story because bug comes from '**Bugbear**'





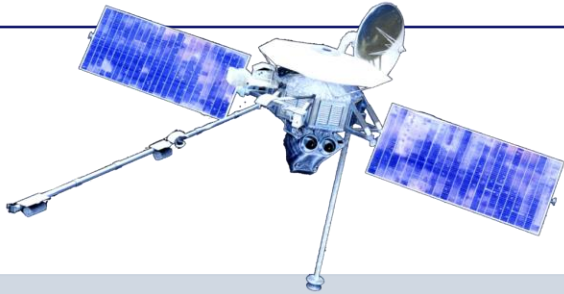
- Bugge is the basis for the terms "bugbear" and "bugaboo" as terms used for a monster.

A fault, defect, bug, error, failure?

Bug	Defect	Error	Fault	Failure
It is an informal name specified to the defect.	The Defect is the difference between the actual outcomes and expected outputs.	An Error is a mistake made in the code; that's why we cannot execute or compile code.	The Fault is a state that causes the software to fail to accomplish its essential function.	If the software has lots of defects, it leads to failure or causes failure.

Failures cost money, time and lives

KOLLMORGEN



The Mariner 1 Spacecraft, 1962, \$169 million



EDS Child Support System, 2004, \$950 million

1994

1962

2004

Pentium FDIV Bug, 1994, \$475 million



Quality Assurance Roadmap



Mihaela Grubii



QA Bootcamp

2017

- Testing Fundamentals
 - SDLC Model
- Practical Manual Testing
- Practical Automation Testing

B.S.E
Information
Technology

2019-2020

2019

- Develop and Maintain QA Framework
 - Reporting
- Monitoring and Logging
 - CI&CD

MS Software
Engineering and
Project Management

KOLLMORGEN



- Develop and Maintain QA Framework
- Monitoring and Logging
 - CI&CD
 - > DevOps.

SQA Engineer

QA Automation
Test Lead

SQA Engineer

Internship

2015-2019

2017-2019

2017

2020

- QA Fundamentals Course
 - Testing Approaches
 - Programing
- Project Management

- Frontend Automation Testing
- Backend Automation Testing
- Write and Maintain Test Scripts
- Identify and Report Defects

- Applied QA Fundamentals
 - Project Management
 - QA Teaching Assistant



Is testing really important?

Bachelor's in
Computer Science
India

2012- 2016

2017-2020

Xyram Software
Solutions
Frontend Development
India

Moved to Sweden

2020

Volvo Group
Connected Solutions
Buses and Trucks

2021

Kollmorgen
Automation AB
AMR(Autonomous
Mobile Robot)

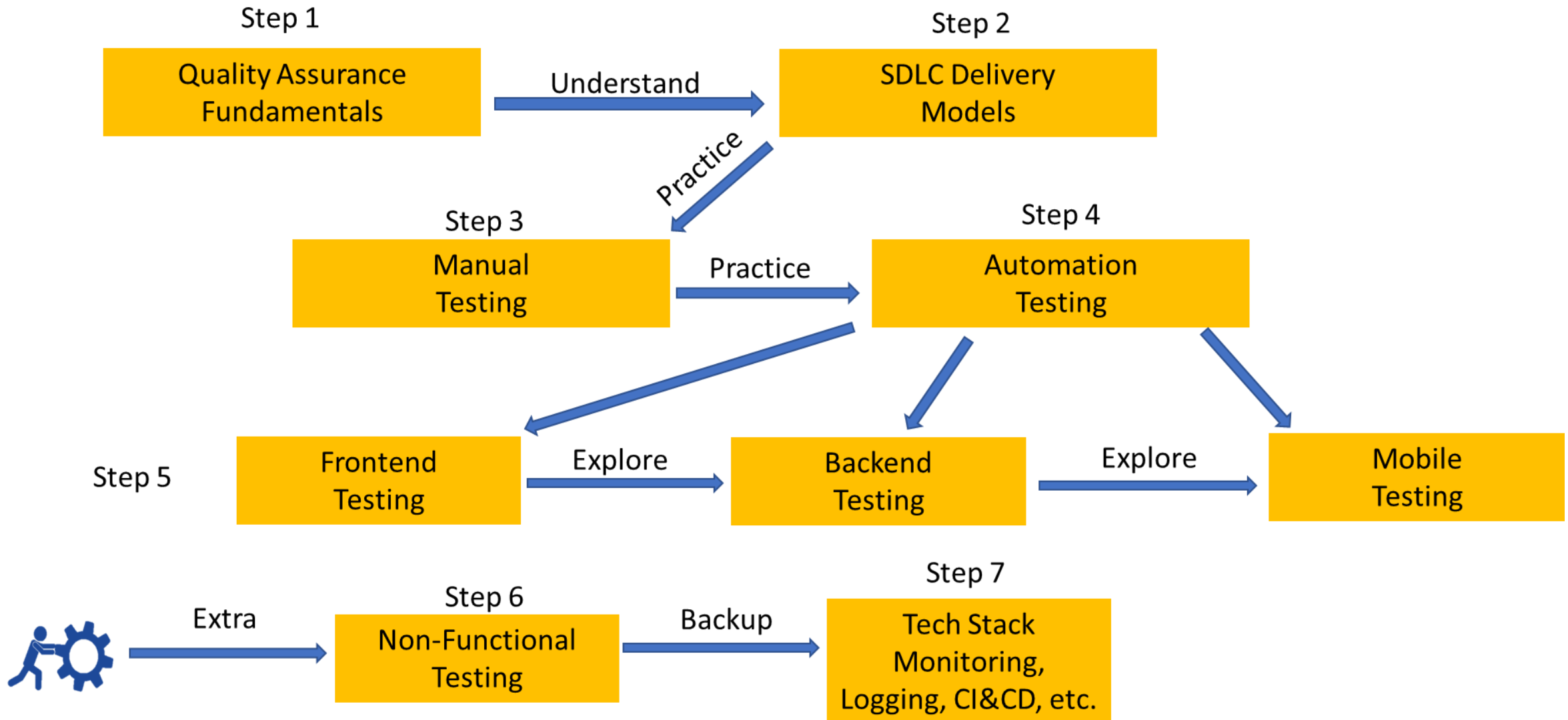
2021

Kollmorgen Automation
AB
SQA Engineer

2022

General Roadmap To Quality Assurance

KOLLMORGEN



Career Progression Opportunities

KOLLMORGEN



Product Manager



QA Manager



Senior Manager



DevOps



Customer experience leader



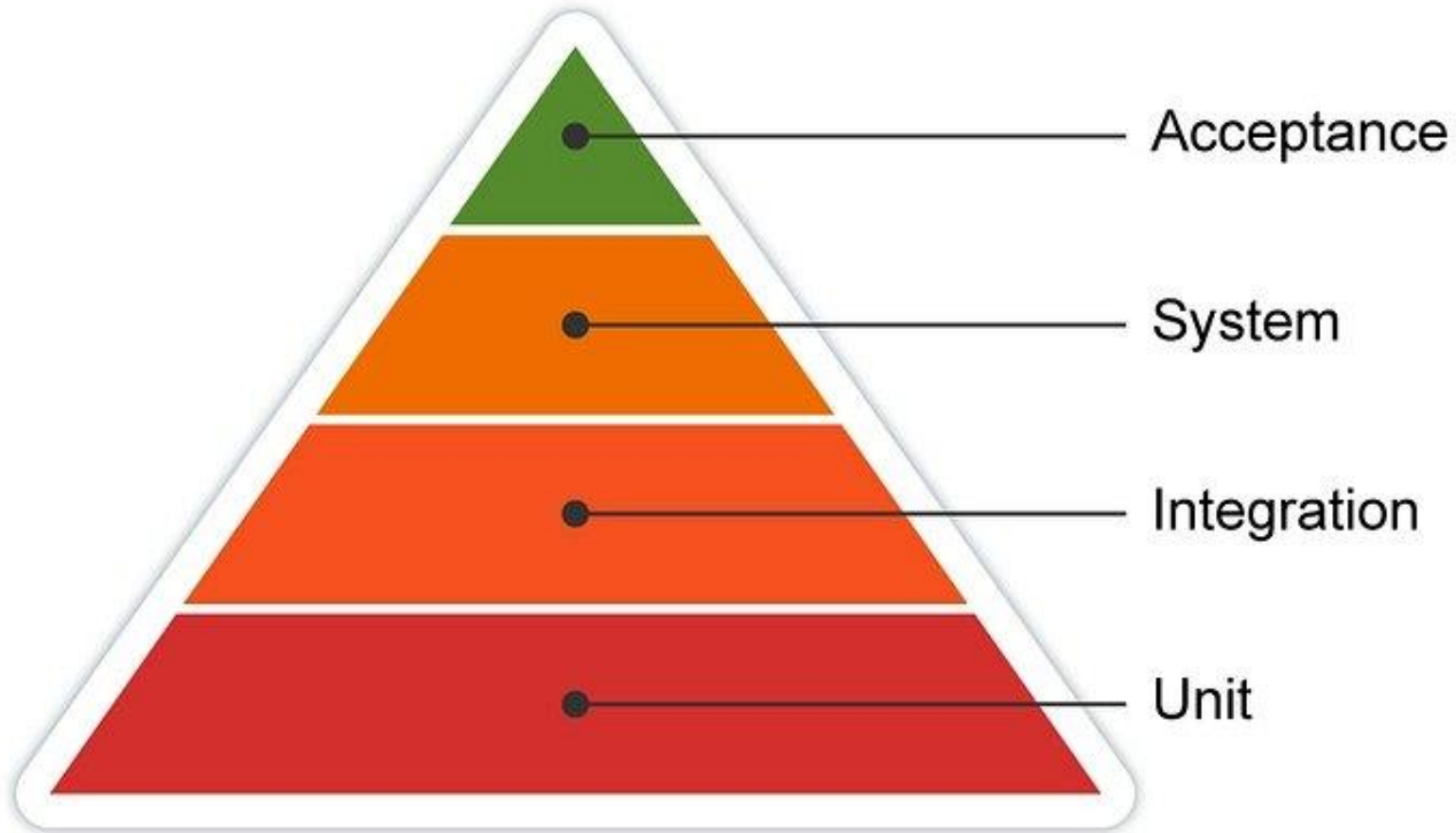
Enterprise architect

Quality Assurance Fundamentals



Testing Levels

KOLLMORGEN



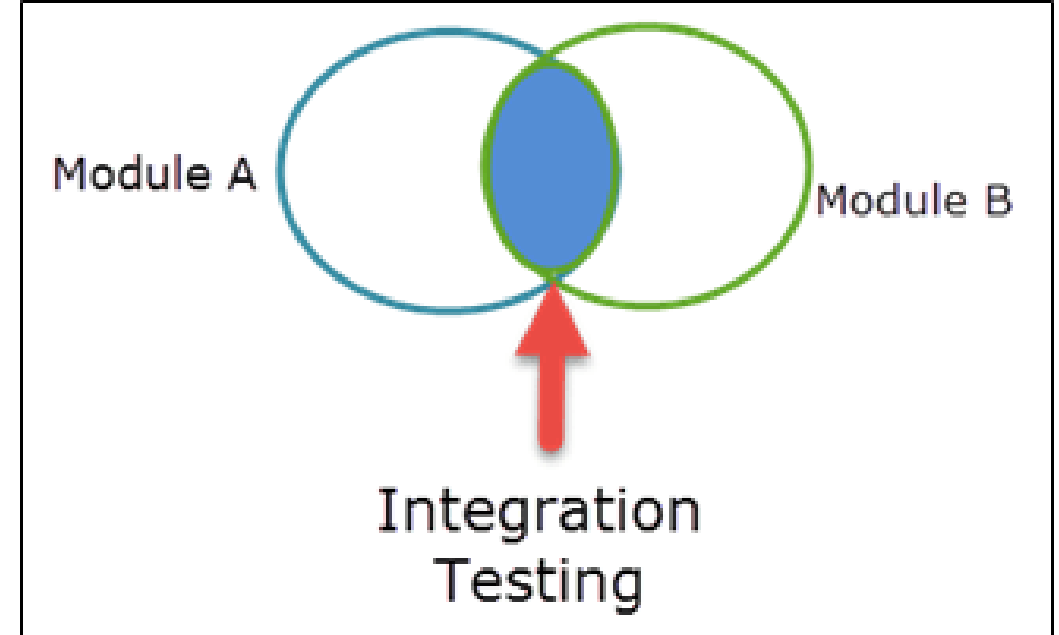
- A level of software testing where individual units(components) of a software are tested.
- Done during the development phase of an application by the **developers**.
- **Fix bugs early** in the development cycle and save costs.
- **White box** approach is used for unit testing.
- Example : For login page check for Name, password, email components
- Unit testing tools – JUnit, TestNG, PHPUnit.

Keep on a straight path with proper unit testing.

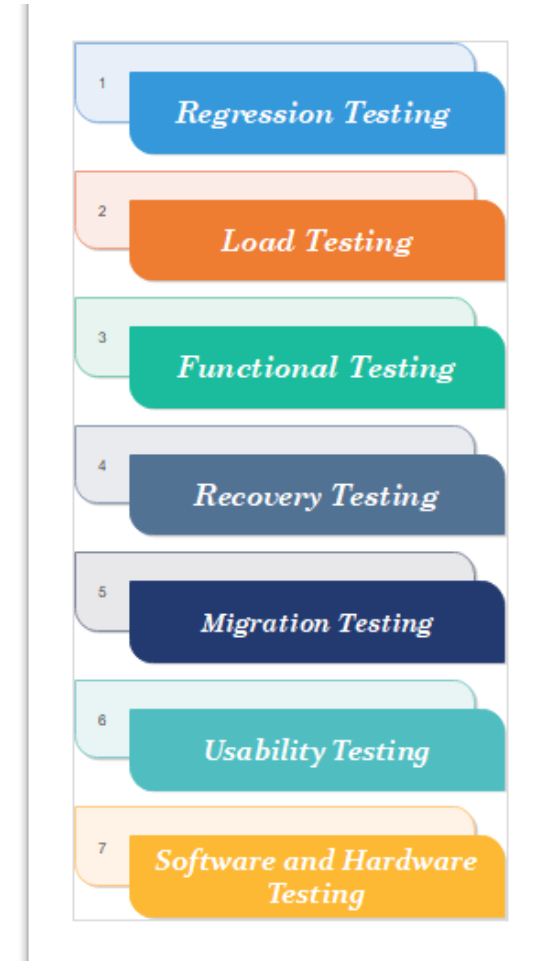


The more code you write without testing, the more paths you have to check for errors

- Software modules are **integrated logically** and tested as a group.
- Performed to **expose defects or faults** in the integration between software modules
- Example Check the interface link between the **Login** and **MailBox** module
- Integration Test approaches : **Big-bang integration, Top Down integration, Bottom Up Integration, Hybrid Integration**

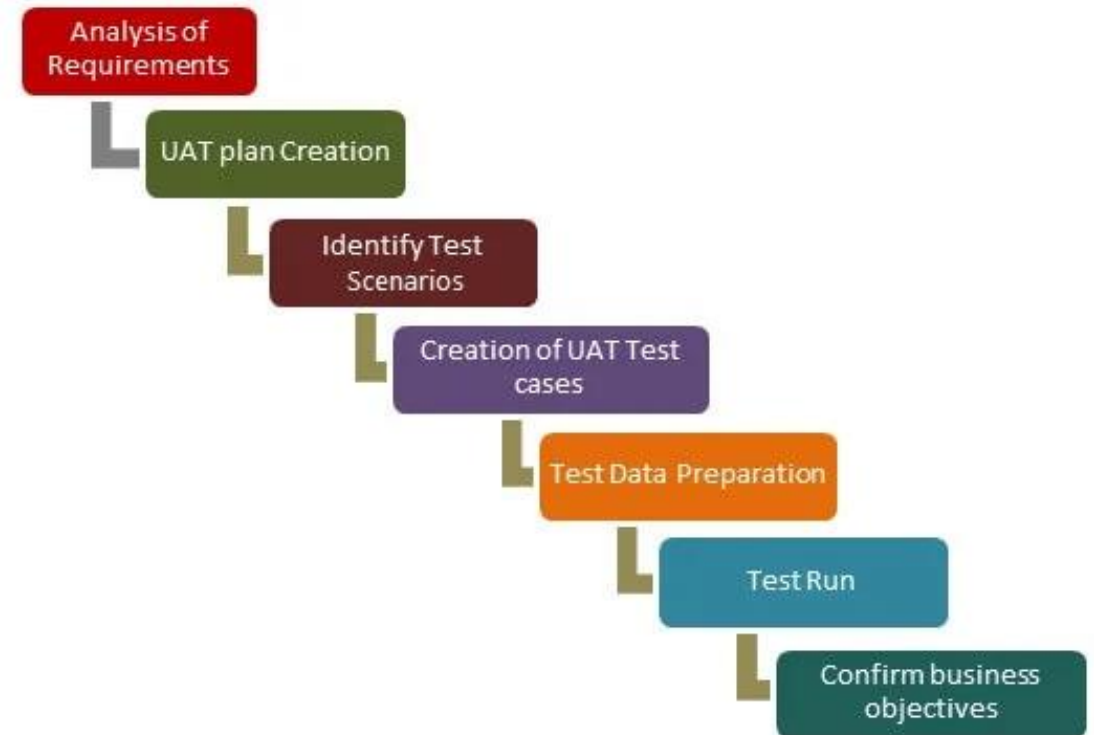


- Testing of a **fully integrated** software system.
- To check the end to end flow of an application or a software as a user is known as System testing.
- This is also called **End to End** testing scenario
- System testing falls under the **Black box testing**.
- Types of System testing: **Regression Testing, Load Testing, Usability Testing, Recovery Testing...**
- Tools used: **Selenium, JMeter.**



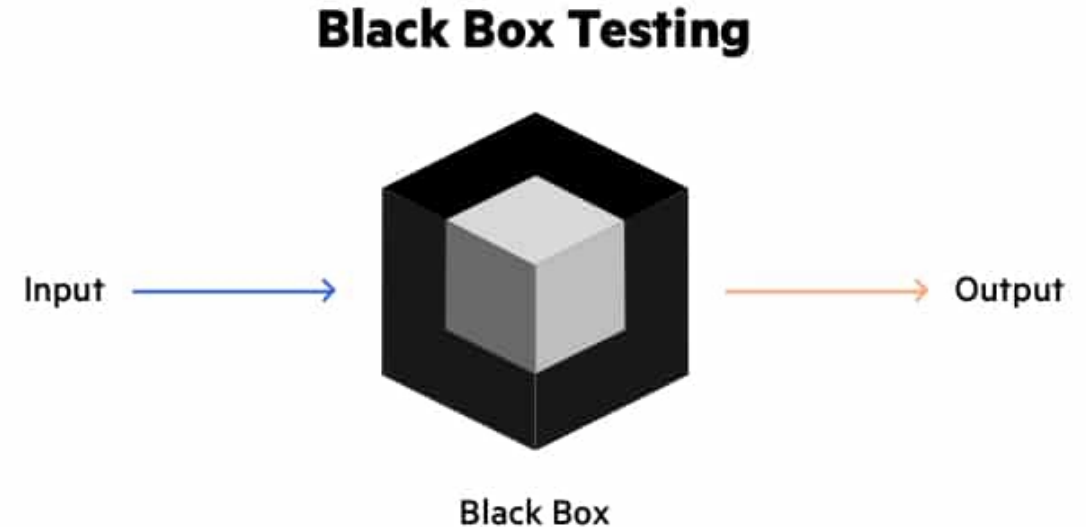
ACCEPTANCE TESTING

- Performed to determine whether or not the software system has met the requirement specifications.
- Test the **final system**.
- Evaluate the system's compliance with the **business requirement**.
- It is performed by client and end users.
- **Beta Testing**(Client location).

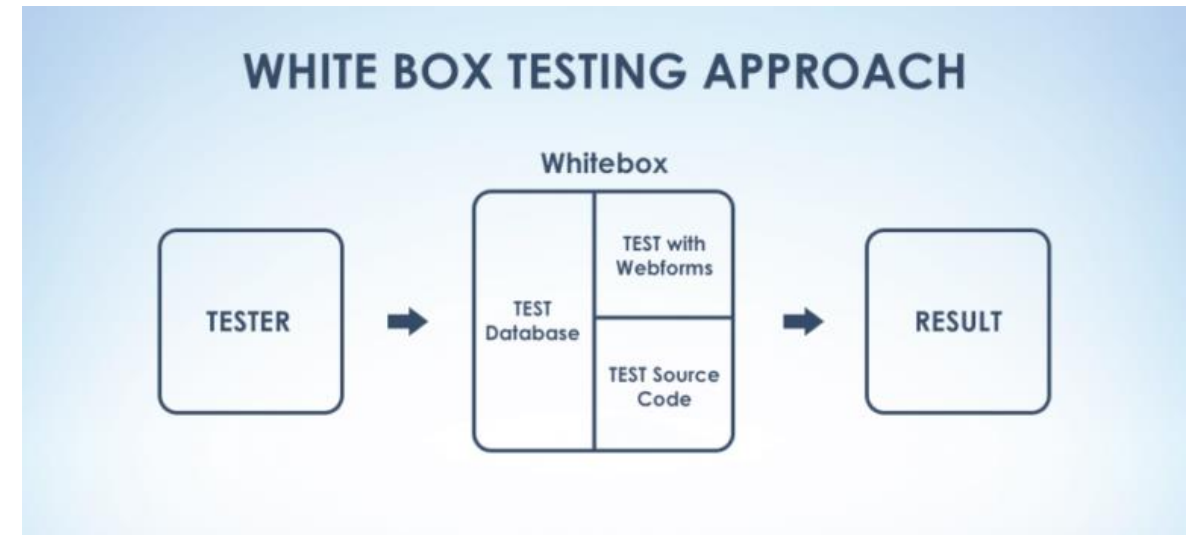


Testing Approaches

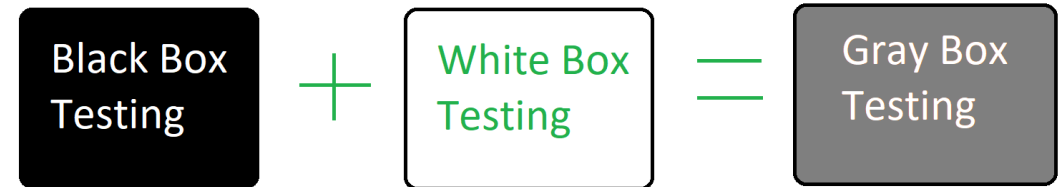
- Functionalities of software applications are tested **without having knowledge** of internal code structure, implementation details.
- Mainly focuses on **input and output** of software application.
- Goal of black box testing is to ensure that the application functions correctly and meets the requirements specified in the design.
- Also called **Behavioral testing**.
- Types of Black Box testing - **Functional, Non-functional, Regression**.



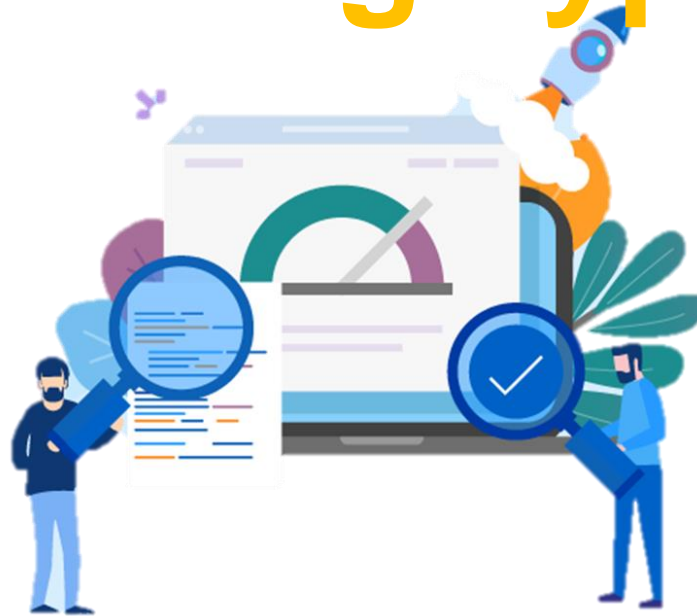
- Method of testing a software application with **knowledge of its internal structure and implementation.**
- Also known as **Open box testing, code based testing.**
- Goal of white box testing is to ensure that the application is **free of errors and efficient.**
- White box testing is more expensive and time consuming than black box testing, but it can help to identify bugs and issues that are not found through black box testing.
- It includes **statement coverage, branch coverage, path coverage, code review.**



- Method of testing a software application with partial knowledge of its internal structure and implementation.
- Goal of Gray box testing is to combine the benefits of black box and white box testing and provides more **comprehensive test coverage**.
- Allows tester to focus on specific area or modules of the application that contains bugs or issues.
- Gray box testing is much more effective in **integration testing and penetration testing**.



Testing Types



Software Testing Types

```
graph TD; A[Software Testing Types] --> B[Functional Testing]; A --> C[Non Functional Testing];
```

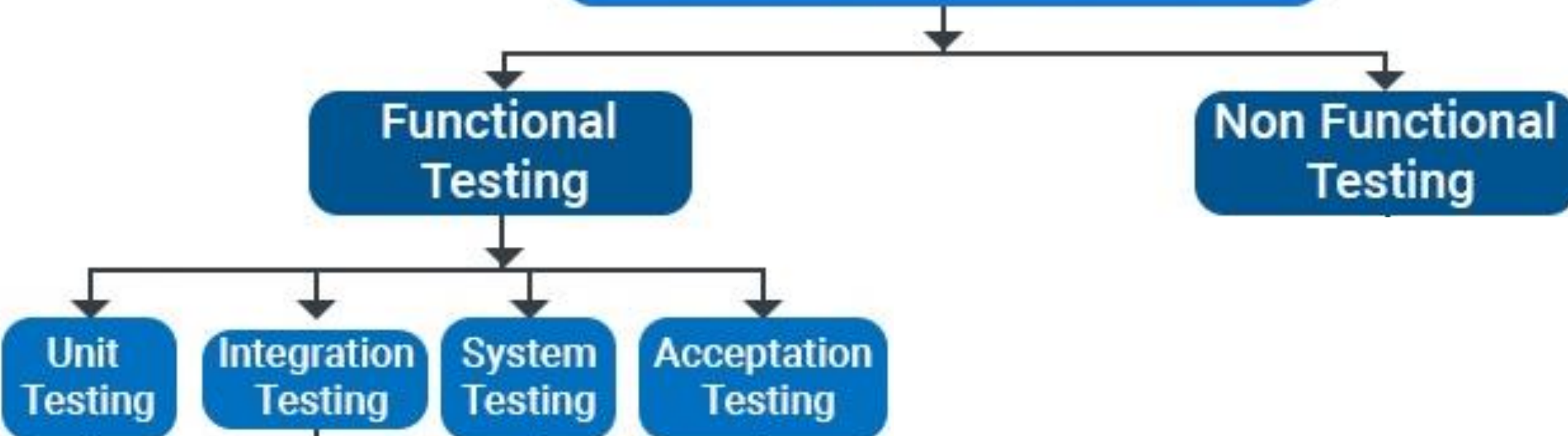
KOLLMORGEN

**Functional
Testing**

**Non Functional
Testing**

Software Testing Types

KOLLMORGEN



Software Testing Types

KOLLMORGEN

Functional Testing

Non Functional Testing

Unit Testing

- White Box Testing
- Gorilla Testing

Integration Testing

- Gray Box Testing

System Testing

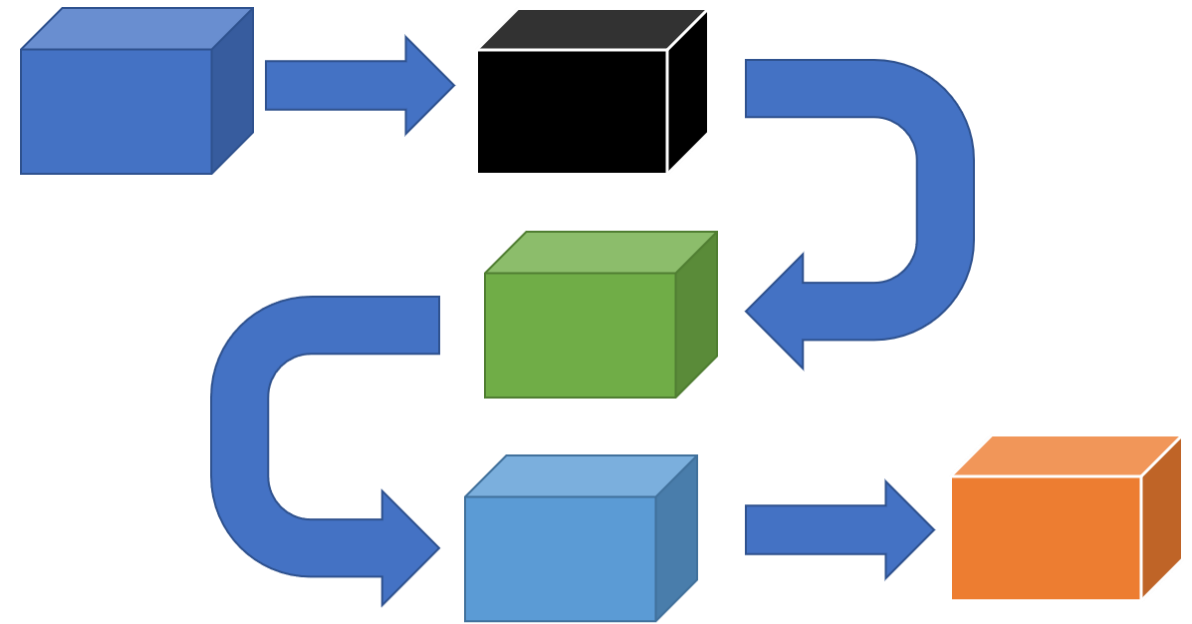
- End to End Testing
- Black Box Testing
- Smoke Testing
- Sanity Testing
- Happy path Testing
- Money Testing

Acceptation Testing

- Alpha Testing
- Beta Testing
- OAT

- performed on complex product in a start-to-finish process
- verifies that all components of a system are able to run and perform optimally under real-world scenarios.

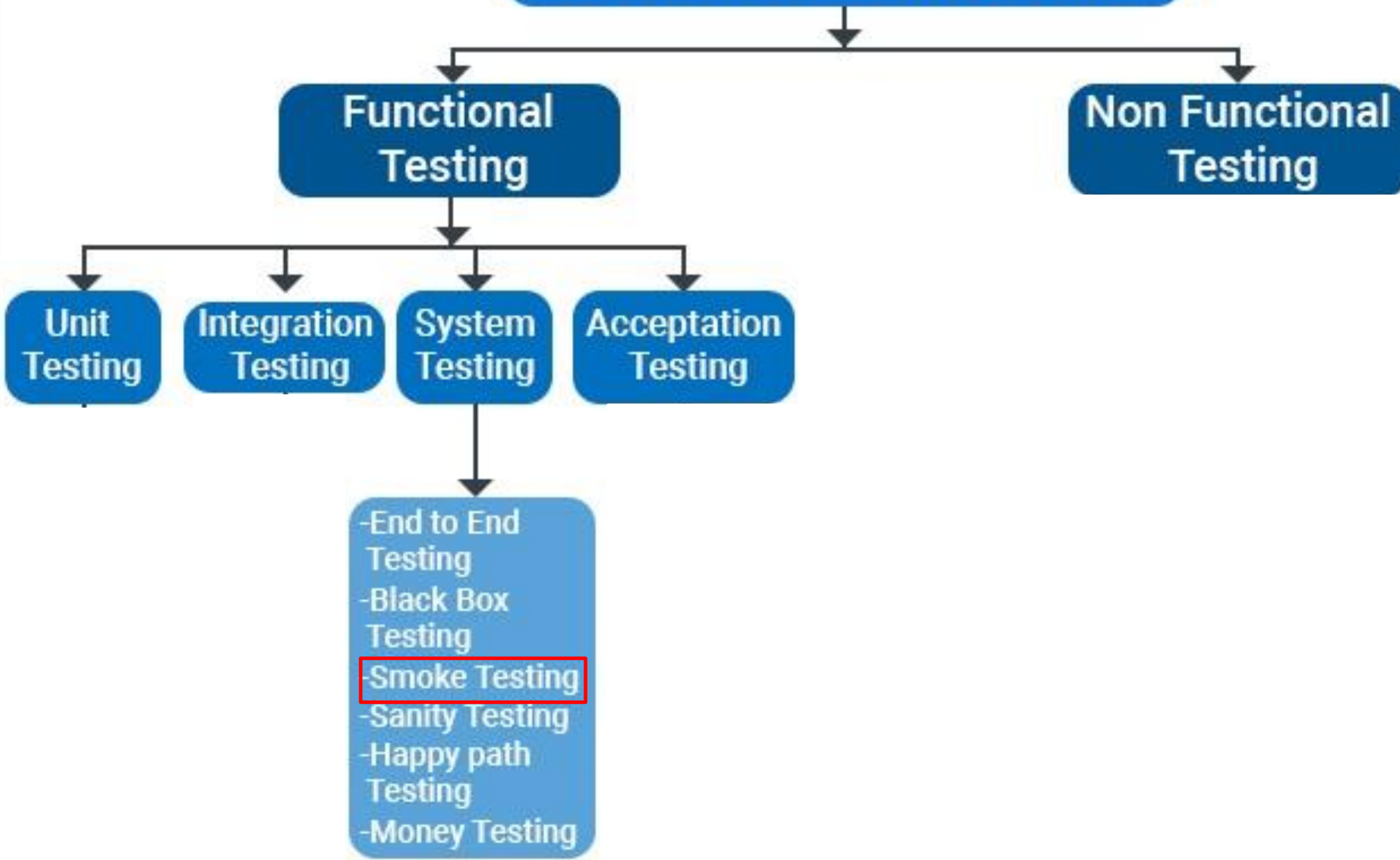
e.g: Playwright, Selenium, Puppeteer



End to End Testing

Software Testing Types

KOLLMORGEN



- ascertains the most crucial functions of a program
- does NOT delve into finer details
- preliminary check of the software
- run with a greater frequency



Software Testing Types

KOLLMORGEN

Functional
Testing

Non Functional
Testing

Unit
Testing

Integration
Testing

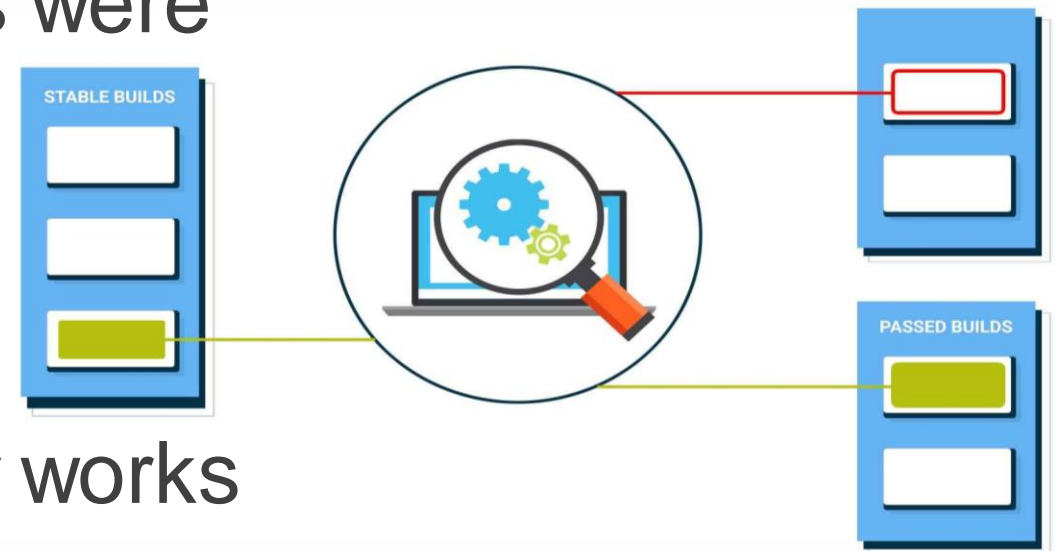
System
Testing

Acceptation
Testing

- End to End Testing
- Black Box Testing
- Smoke Testing
- Sanity Testing
- Happy path Testing
- Money Testing

Sanity Testing

- performed after small fixes or minor changes in code
- intended to verify that known bugs were fixed
- checks no further issues are
- introduced due to these changes
- checks that proposed functionality works roughly as expected.
- failed sanity test, results in rigorous testing is required



Software Testing Types

```
graph TD; A[Software Testing Types] --> B[Functional Testing]; A --> C[Non Functional Testing]; C --> D[Security Testing]; C --> E[Performance Testing]; C --> F[Usability Testing]; C --> G[Compatibility Testing];
```

KOLLMORGEN

**Functional
Testing**

**Non Functional
Testing**

**Security
Testing**

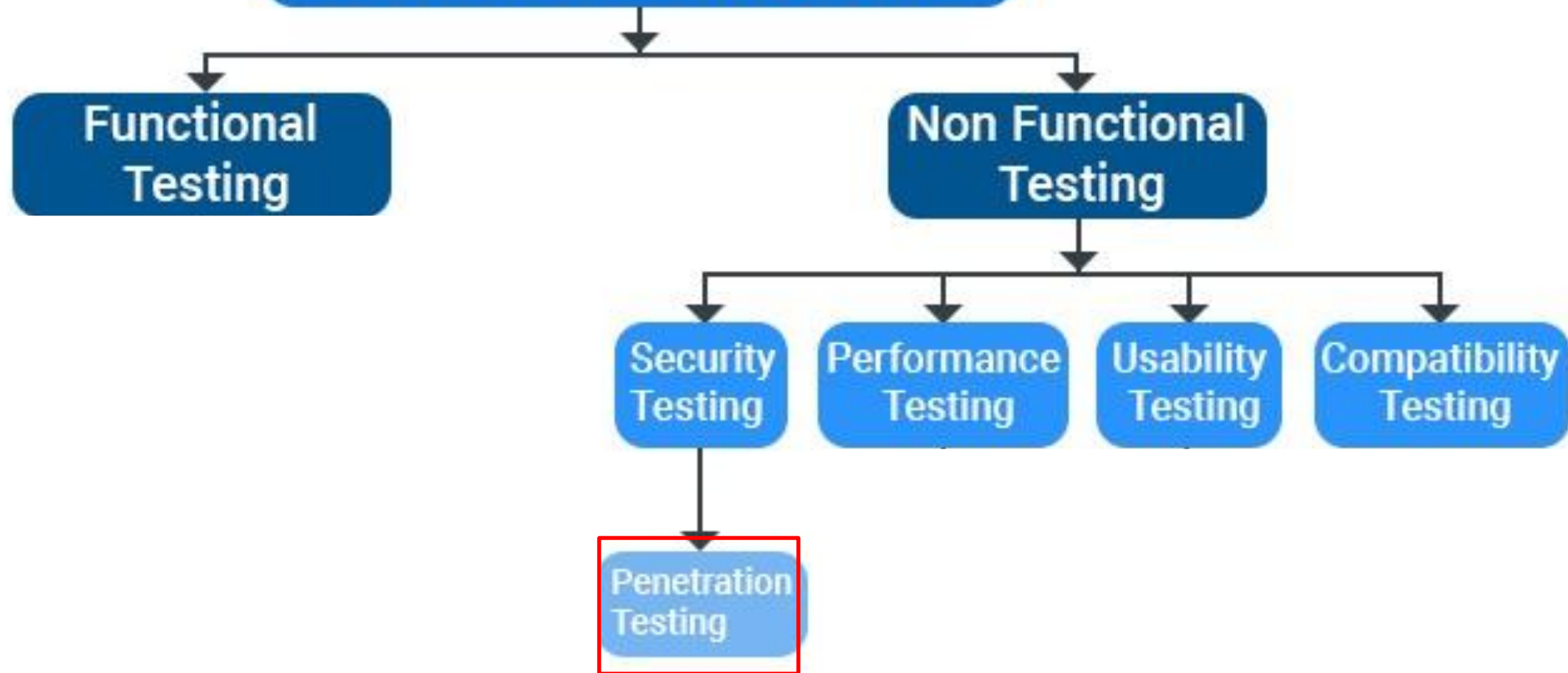
**Performance
Testing**

**Usability
Testing**

**Compatibility
Testing**

Software Testing Types

KOLLMORGEN



- uncovers vulnerabilities, threats, risks in a software application
- prevents malicious attacks from intruders
- identifies all possible loopholes and weaknesses

Penetration Testing simulates an attack from a malicious hacker and checks for potential vulnerabilities to an external hacking attempt.

Bug Bounty

e.g: Metasploit, Burpsuite

Vulnerability Scanning

Security Scanning

Penetration testing

Risk Assessment

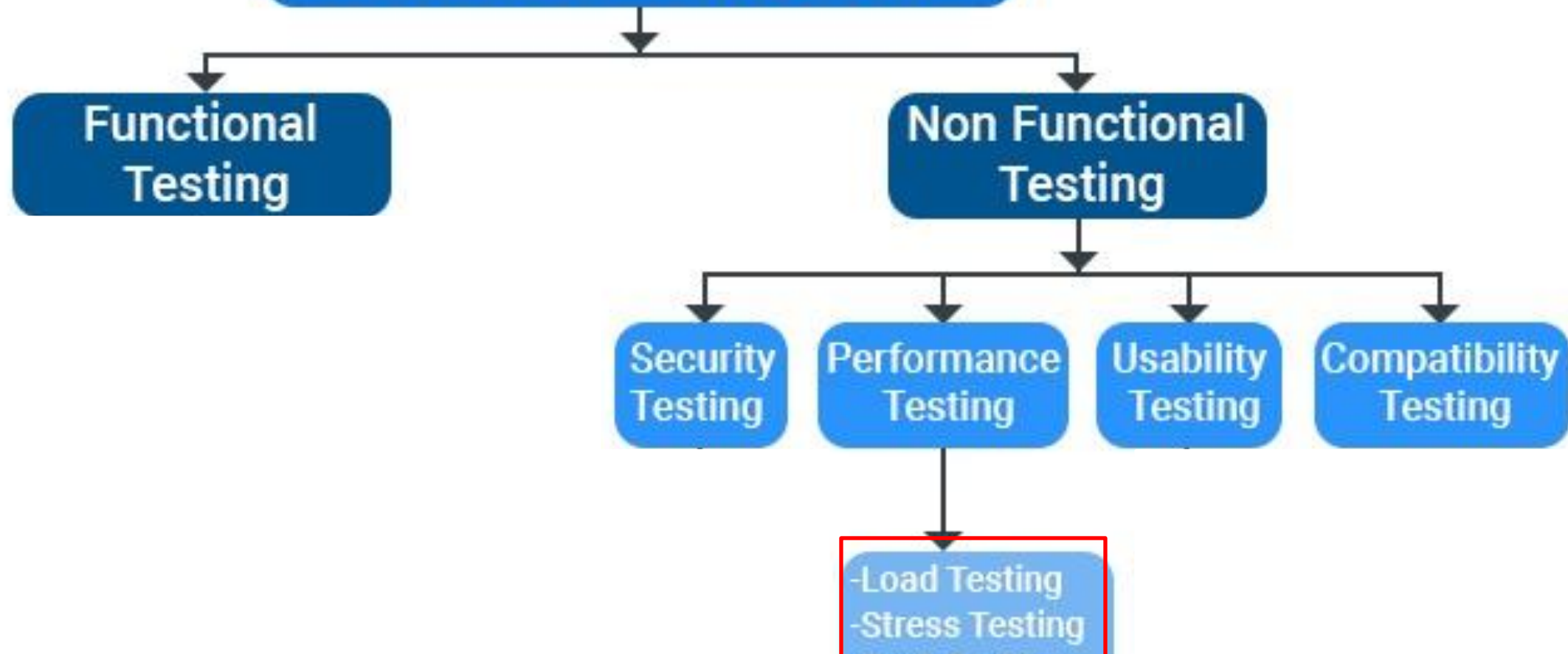
Security Auditing

Posture Assessment

Ethical hacking

Software Testing Types

KOLLMORGEN



Load Testing

- a process that tests the performance of a software application under a specific expected load
- determines how the software behaves while being accessed by multiple users simultaneously
- improve performance bottlenecks
- to ensure stability and smooth functioning before deployment



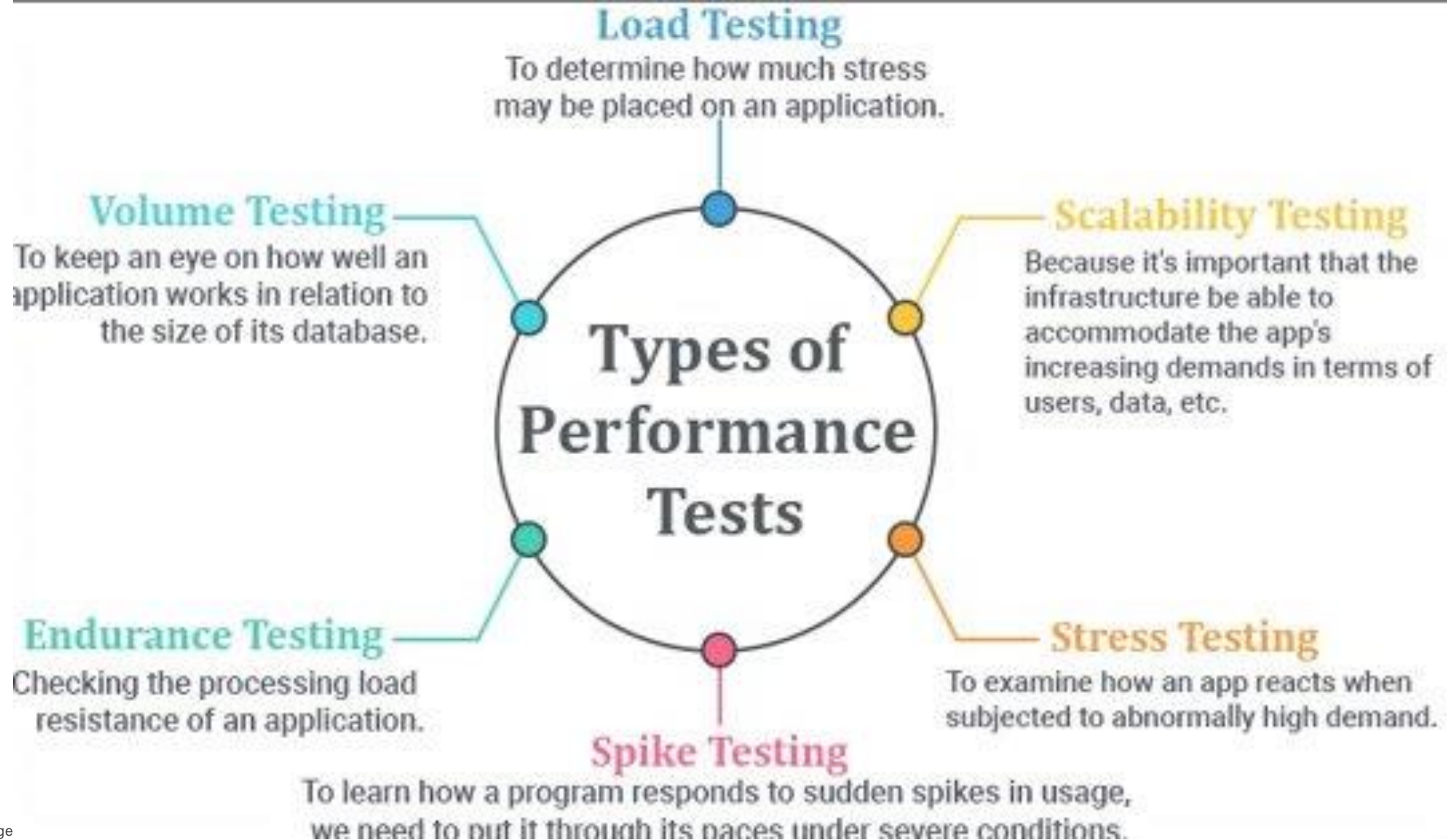
e.g: Apache JMeter,
LoadRunner, LoadNinja

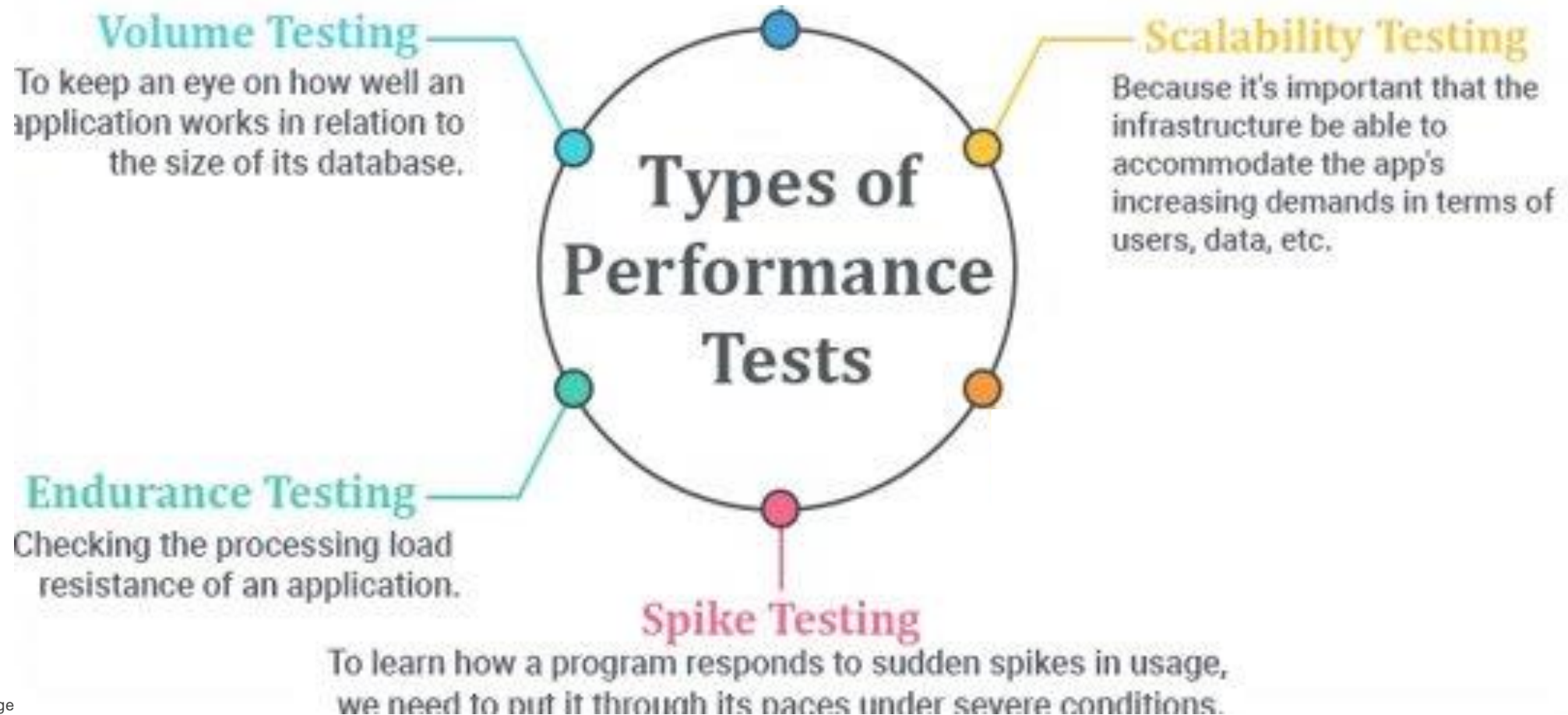
- verifies the stability and reliability of the system
- determines the system on its robustness and error handling
- extremely heavy load conditions
- ensures the feasibility of the software

e.g: WebLoad, LoadNinja, Apache JMeter



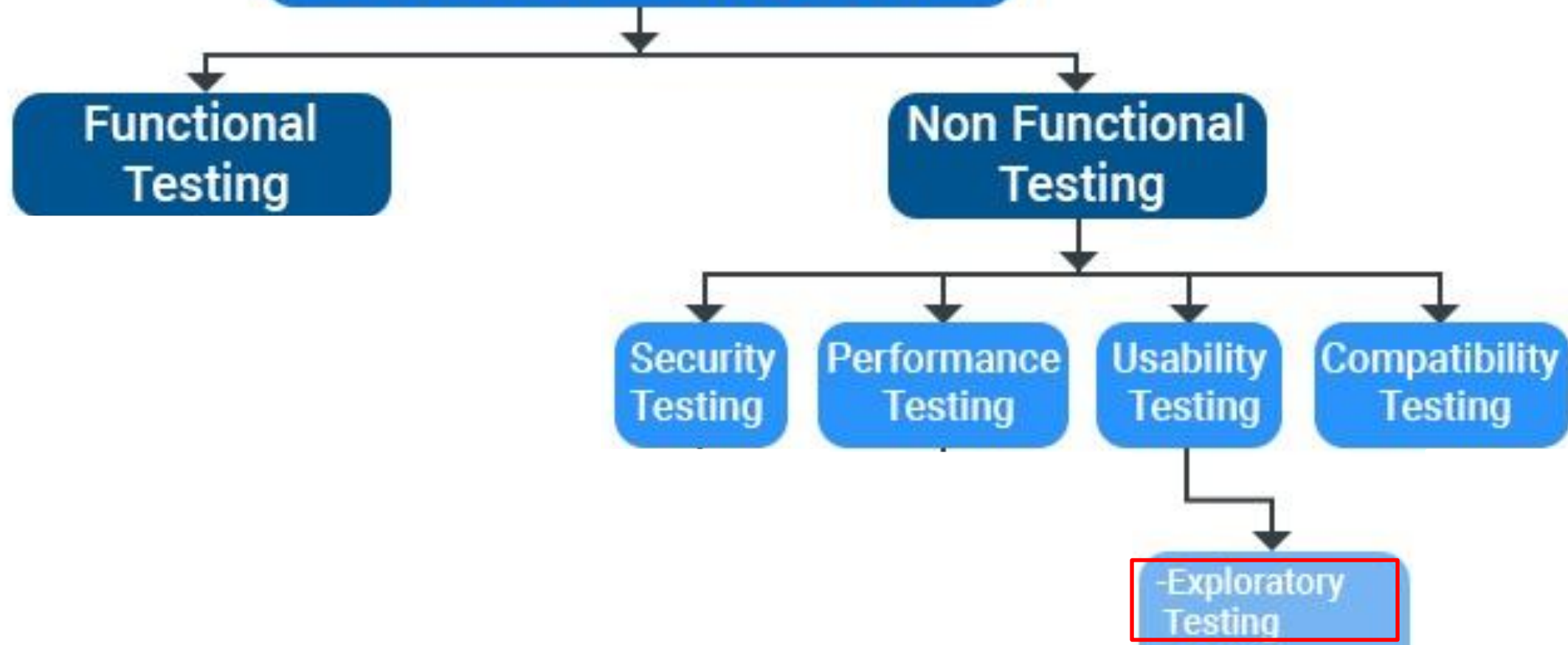
More Performance Testing Types



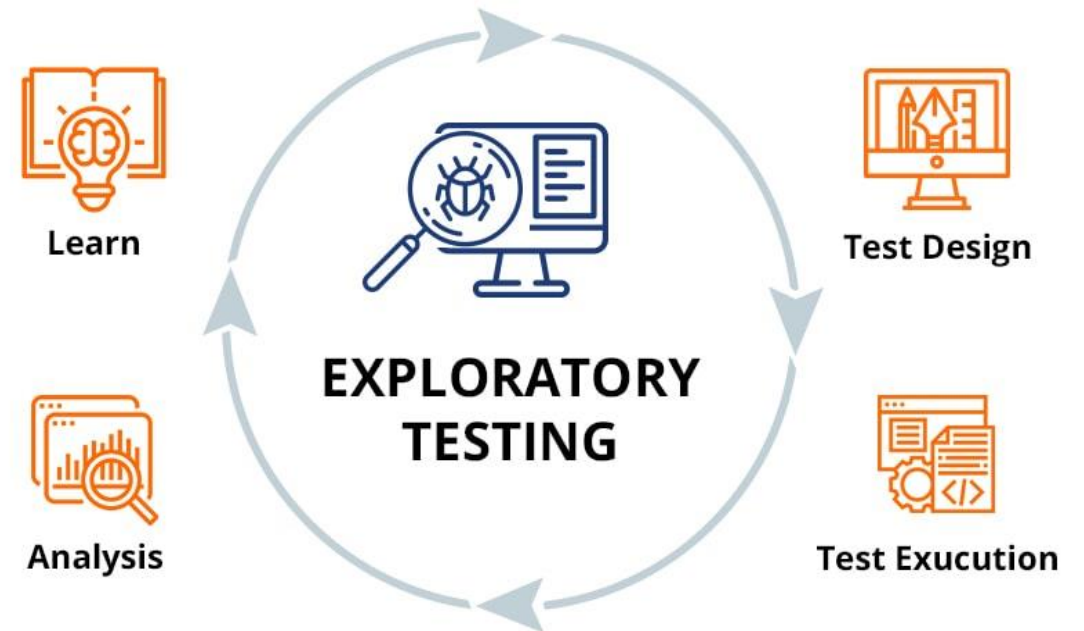


Software Testing Types

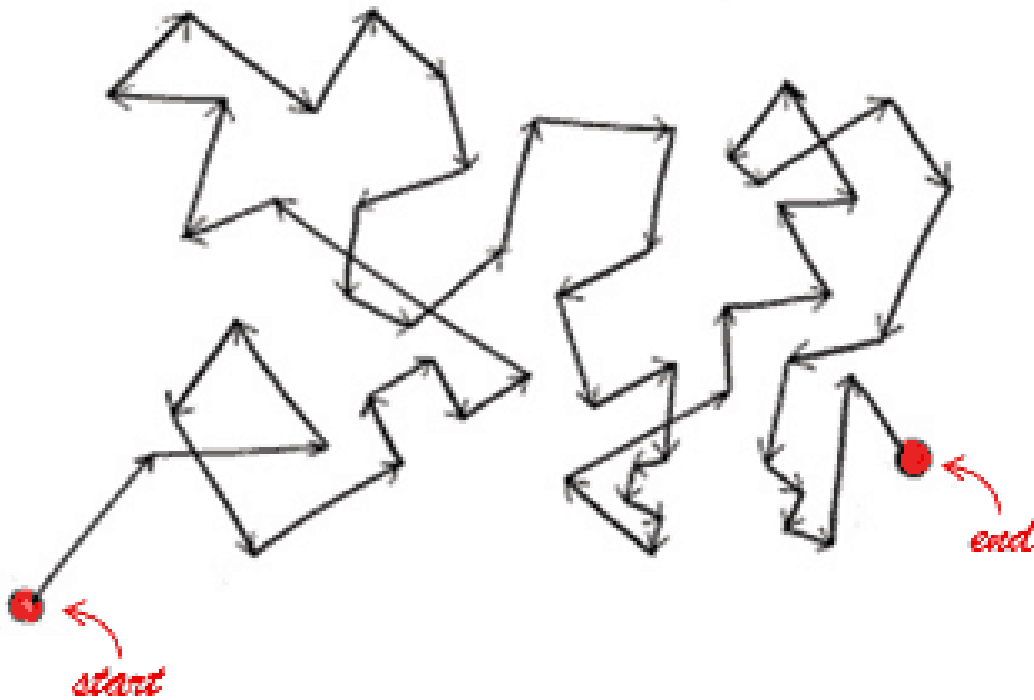
KOLLMORGEN



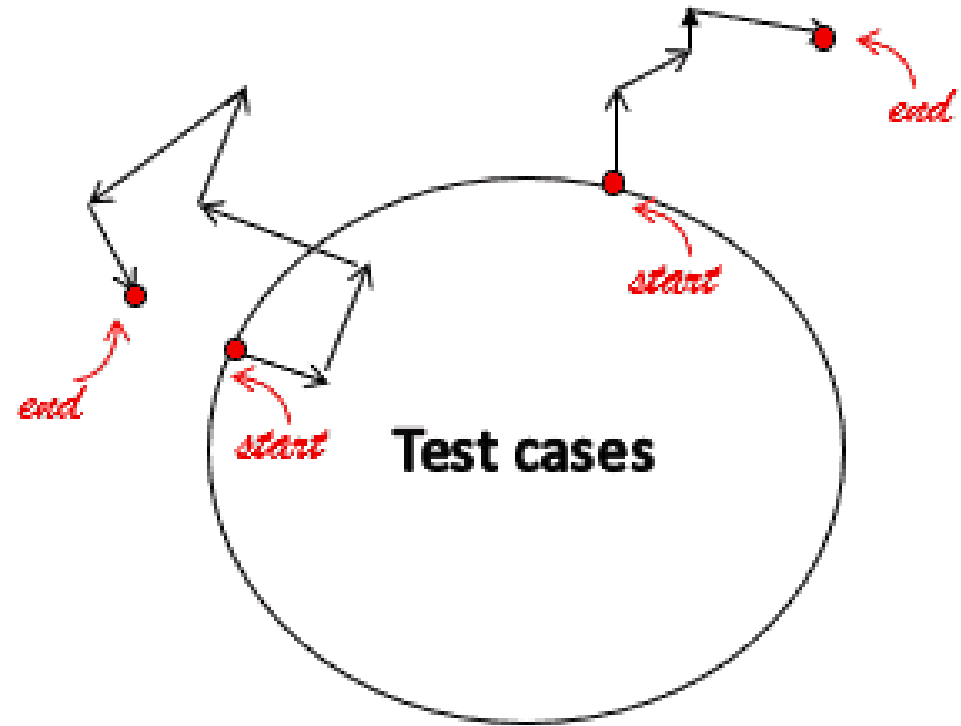
- **described as simultaneous learning, test design, and execution.**
 - focuses on discovery and relies on the guidance of the individual tester
 - targets area not covered by automated tests
 - captures non-obvious defects
- e.g: optional (Bug Magnet, Fiddler)



Ad hoc testing



Exploratory Testing



Quiz Time

www.kahoot.it



Please Ask Us Questions

- If you are interested what tools we use
- What topics you found insightful
- Interesting things in our daily work

Backstory: You are new SQA and are tasked to test an internal tool for meeting room management. You should identify the defects during some exploratory testing and report them to Jira.

1. Log In Into (1-6) mail.com (7-10) account.proton.me
2. Check Your Inbox and Create a Jira Account
 - Attlassian (verify link)
 - Jira (create your username: e.g Team6)
3. Check Your Inbox for GitHub link

- Jira: <https://datatjekollmorgen.atlassian.net/jira/software/projects/KMTOOLS/boards/1>
- GitHub: <https://github.com/KollmorgenWorkshop/DataTjejPracticTaskKollmorgen>
- Mailing service:
- <https://account.proton.me/login>
- <https://www.mail.com/>
- Java (Windows x64):
- https://cdn.azul.com/zulu/bin/zulu11.62.17-ca-jdk11.0.18-win_x64.zip
- Java (Windows x32):
- https://cdn.azul.com/zulu/bin/zulu11.62.17-ca-jdk11.0.18-win_i686.zip
- Java (Mac):
- https://cdn.azul.com/zulu/bin/zulu11.62.17-ca-jdk11.0.18-macosx_x64.zip