



# **QA&QC DURING THE SOFTWARE DEVELOPMENT LIFE CYCLE**

**PRESENTERS:**

**ROXANA SOPORAN**

**ALEXANDRU ANDREI**

# AGENDA

---

- INTRODUCTION
- QUALITY ASSURANCE & QUALITY CONTROL (QA & QC)
- QUALITY MANAGEMENT SYSTEMS (QMS)
- CONTINUOUS IMPROVEMENT (CI)
- REQUIREMENTS/ DESIGN
- TEST PLANNING
- TEST DESIGN
- TEST EXECUTION
- TEST REPORTING

---

**WHAT DO YOU KNOW ABOUT ENDAVA?**



# WE DELIVER GLOBAL TRANSFORMATION

## OFFICES

### AMSTERDAM

Laapersveld 43, Hilversum

### ATLANTA

715 Peachtree Street N.E. Suite 2046

### BELGRADE

9đ, Milutina Milankovića St.

### BOGOTA

Calle 96 No. 10-38, Edificio BOX, 7th & 8th Floor, Bogota D.C.

### BRASOV

5 Turnului Street, 4th Floor



## PHILOSOPHY

# OUR PHILOSOPHY IS SIMPLE

"We focus on helping people succeed. The people who work for us, the people who engage with us, and the people who use the systems and applications we design, build, and operate."

**JOHN COTTERELL, CHIEF EXECUTIVE, ENDAVA**





# BE MORE

we're prodigies and hard workers, peace lovers and fighters,  
we're designers and project managers, automation testers and architects,  
we're leaders and rebels, skaters and bikers,  
but above all, we are people who believe they can be more

[See Videos >](#)



# QA & QC

---

QUALITY **ASSURANCE** AND QUALITY **CONTROL**

WHAT IS THE **DIFFERENCE?**

# QA & QC

---





# QMS

---

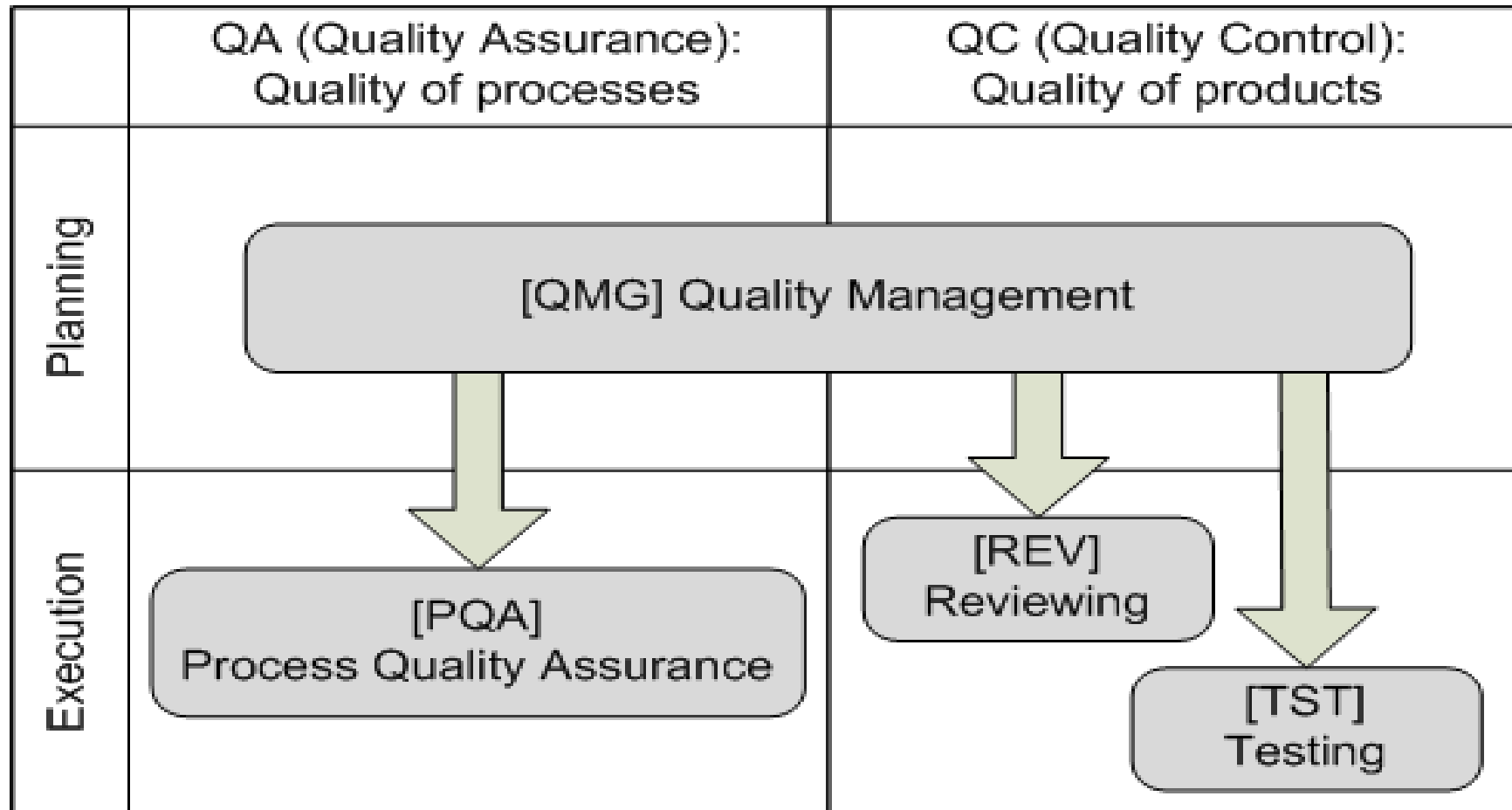
**A QUALITY MANAGEMENT SYSTEM (QMS)  
IS A SET OF **POLICIES, PROCESSES** AND **PROCEDURES**  
REQUIRED FOR **PLANNING** AND **EXECUTION**  
(PRODUCTION/DEVELOPMENT/SERVICE) IN THE CORE  
BUSINESS AREA OF AN **ORGANIZATION****

# QMS

---



# QMS



# MEASURING QUALITY

---

- IF YOU CANNOT **DEFINE** IT, YOU CANNOT ACHIEVE IT

- IF YOU CANNOT **MEASURE** IT

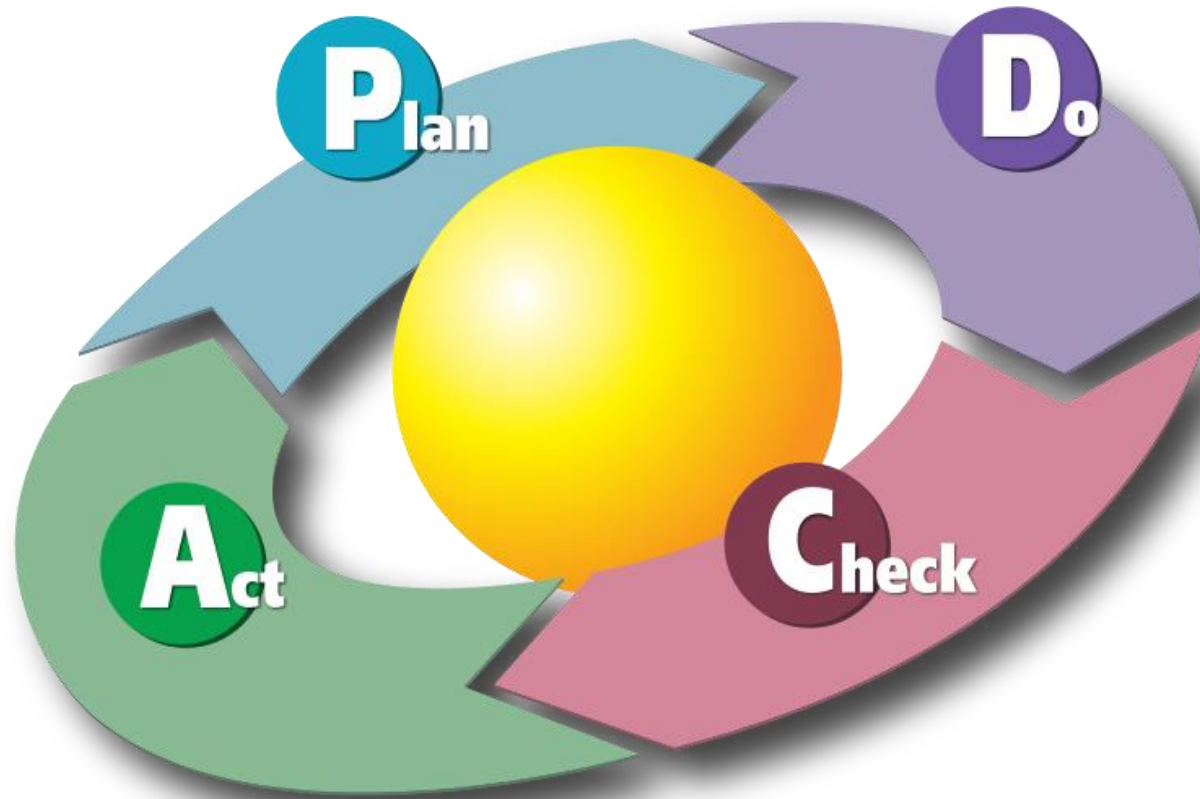
You do not know how you are progressing

You do not know when you have arrived

You cannot demonstrate it

# CI

---



# CI

---

## ■ MAIN DRIVER WAS THE VERY DIVERSIFIED WAY OF WORKING:

- Multiple technologies
- Multiple methodologies
- Various level of client control
- Locations
- Company growth, etc

## ■ MAIN GOAL IS TO ENSURE COMPANY-WIDE SPREAD OF

- Lessons learned
- Best practices
- Ensure a uniformed way-of-working between projects

# SUMMARY

---

- QA & QC GIVE THE **BEST RESULTS** TAKEN TOGETHER
- **CUTTING CORNERS** IS NOT AN OPTION
- **DELIVERING QUALITY IS EVERYBODY'S JOB**
- YOU HAVE TO DEFINE **"WHAT IS QUALITY"** IN ORDER TO ACHIEVE IT
- **CONTINUOUS IMPROVEMENT**
- **QA PROCESSES** MUST BE PRESENT IN EACH STAGE OF THE **DEVELOPMENT LIFECYCLE**

# ASK QUESTIONS

---

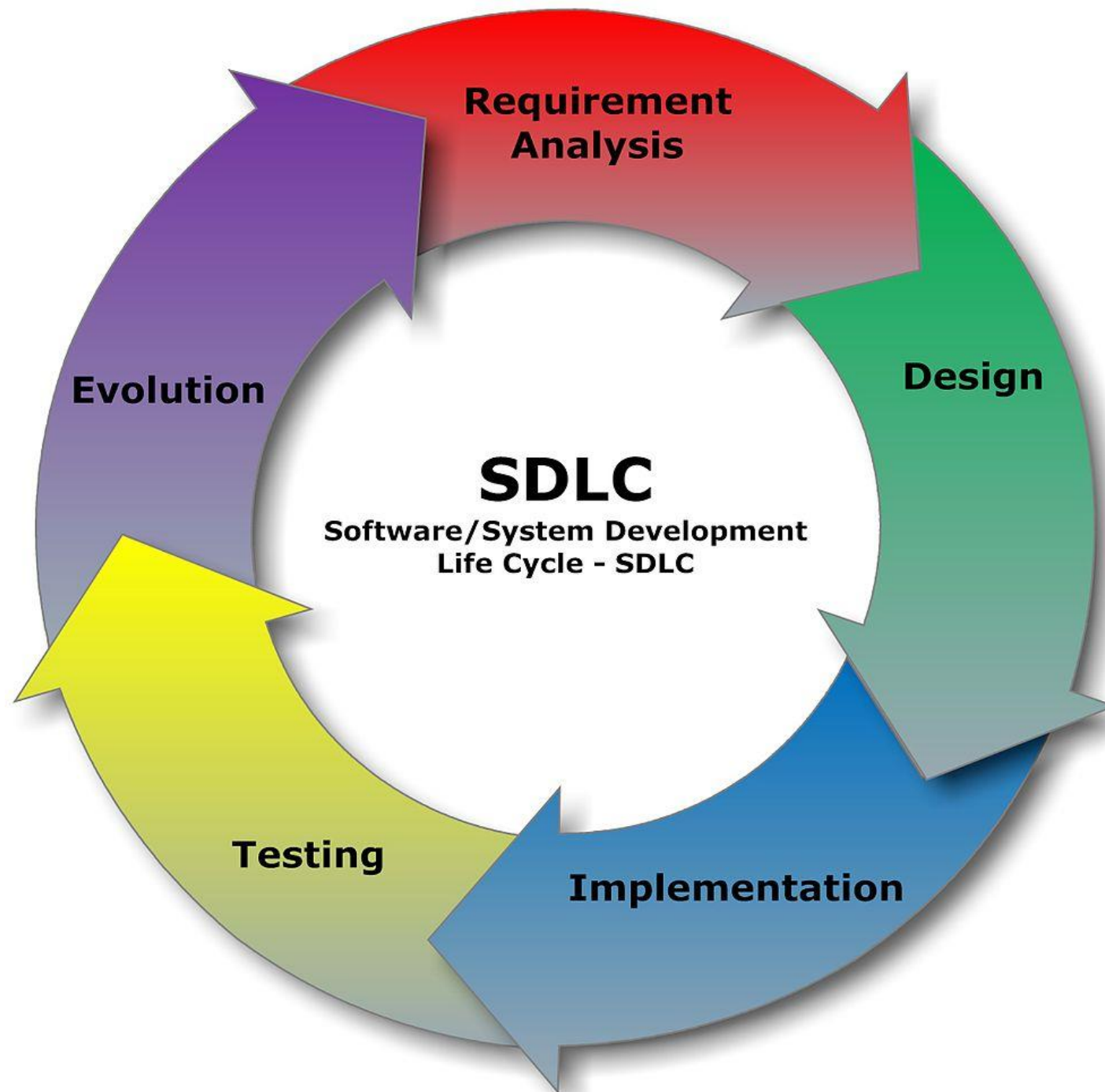


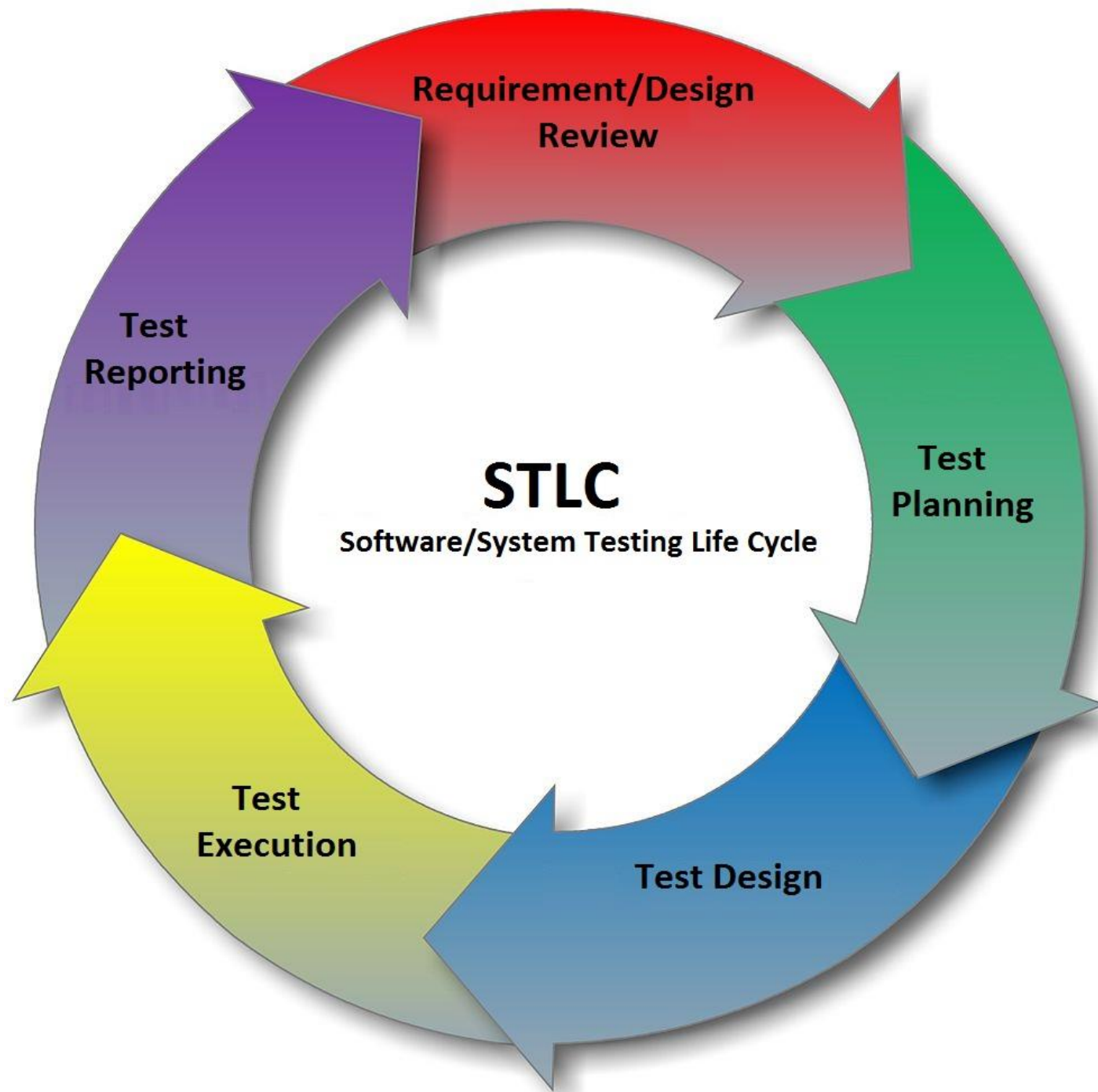


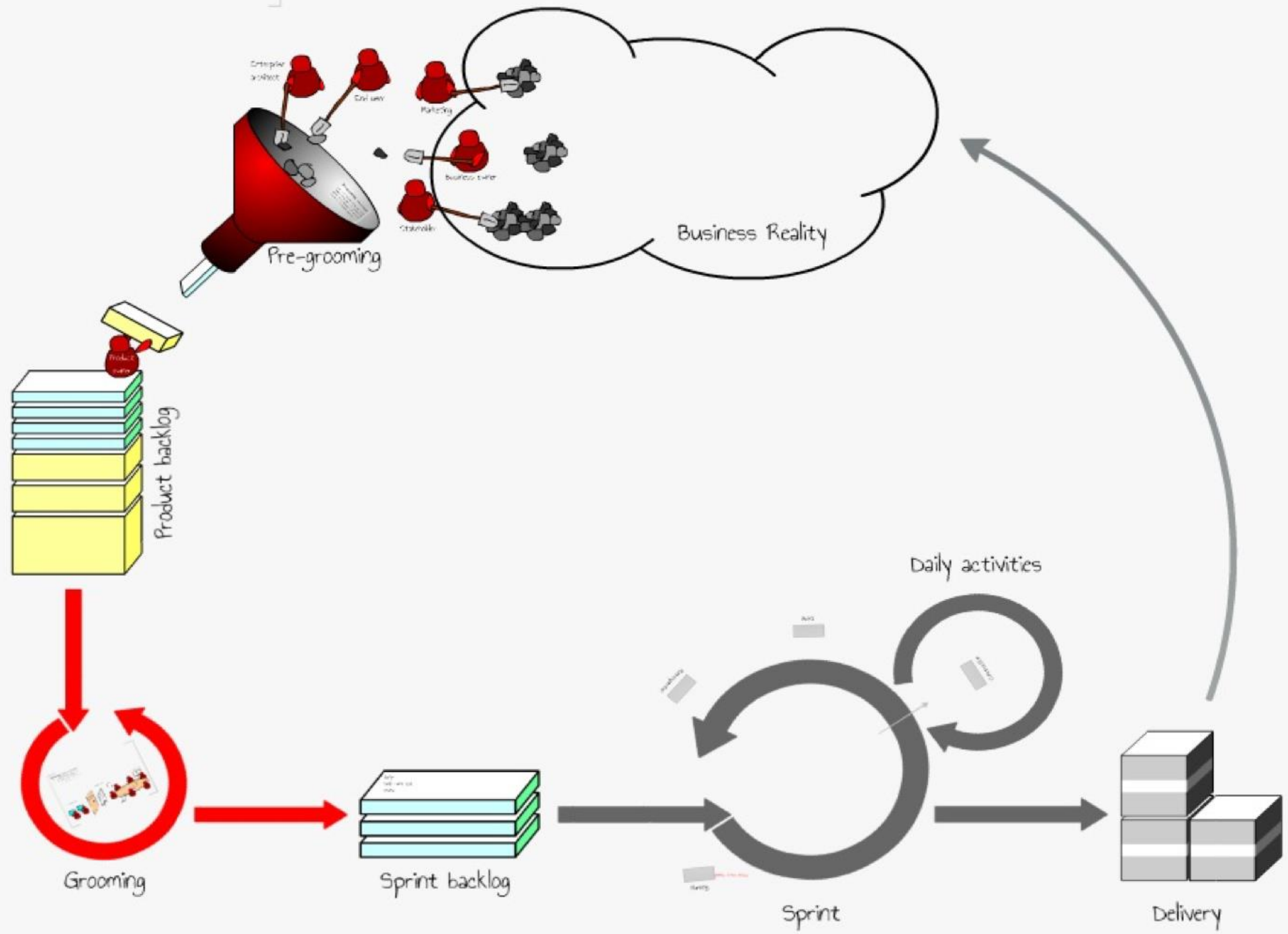
---

# WHY IS SOFTWARE TESTING NECESSARY?



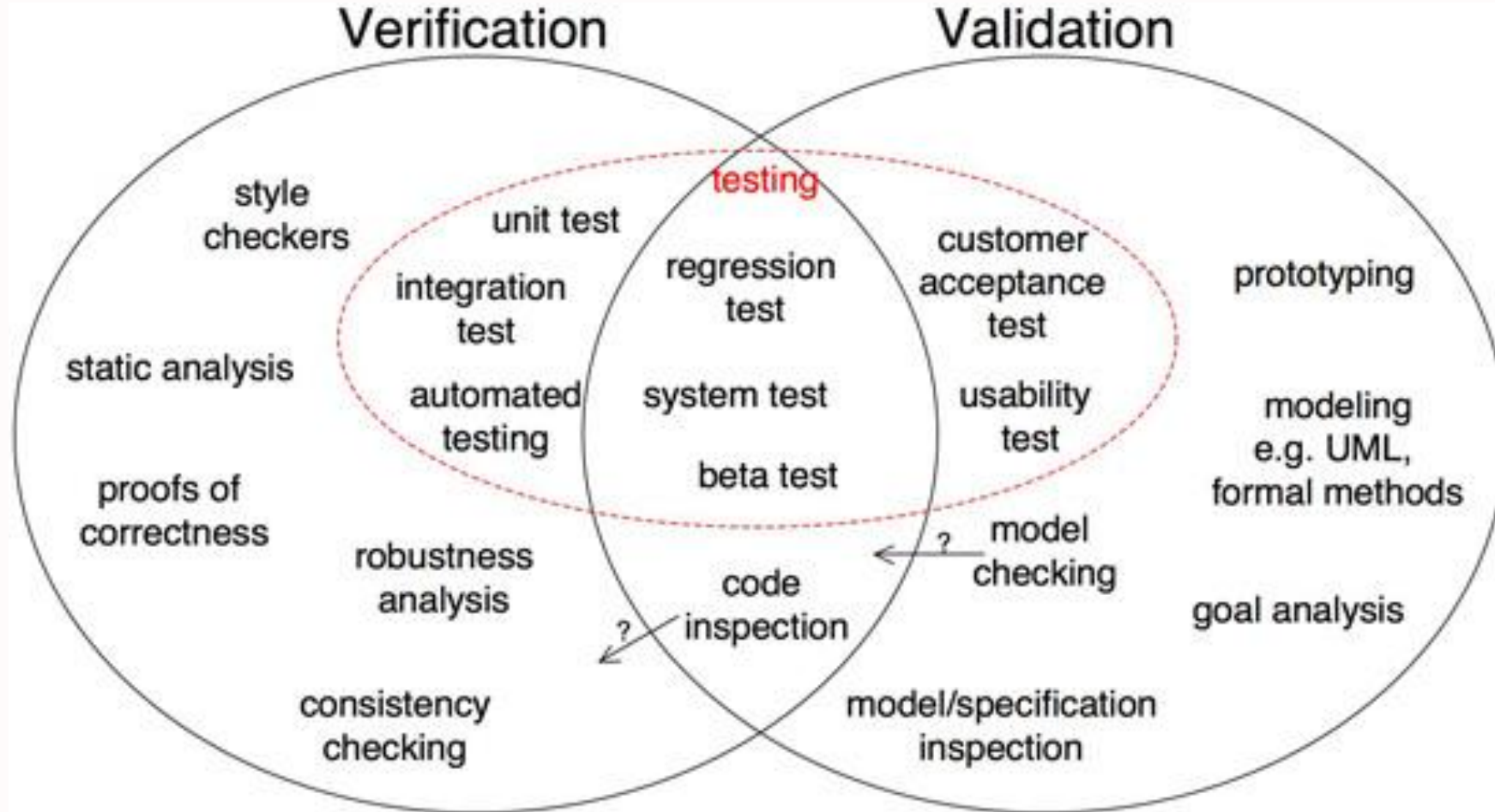




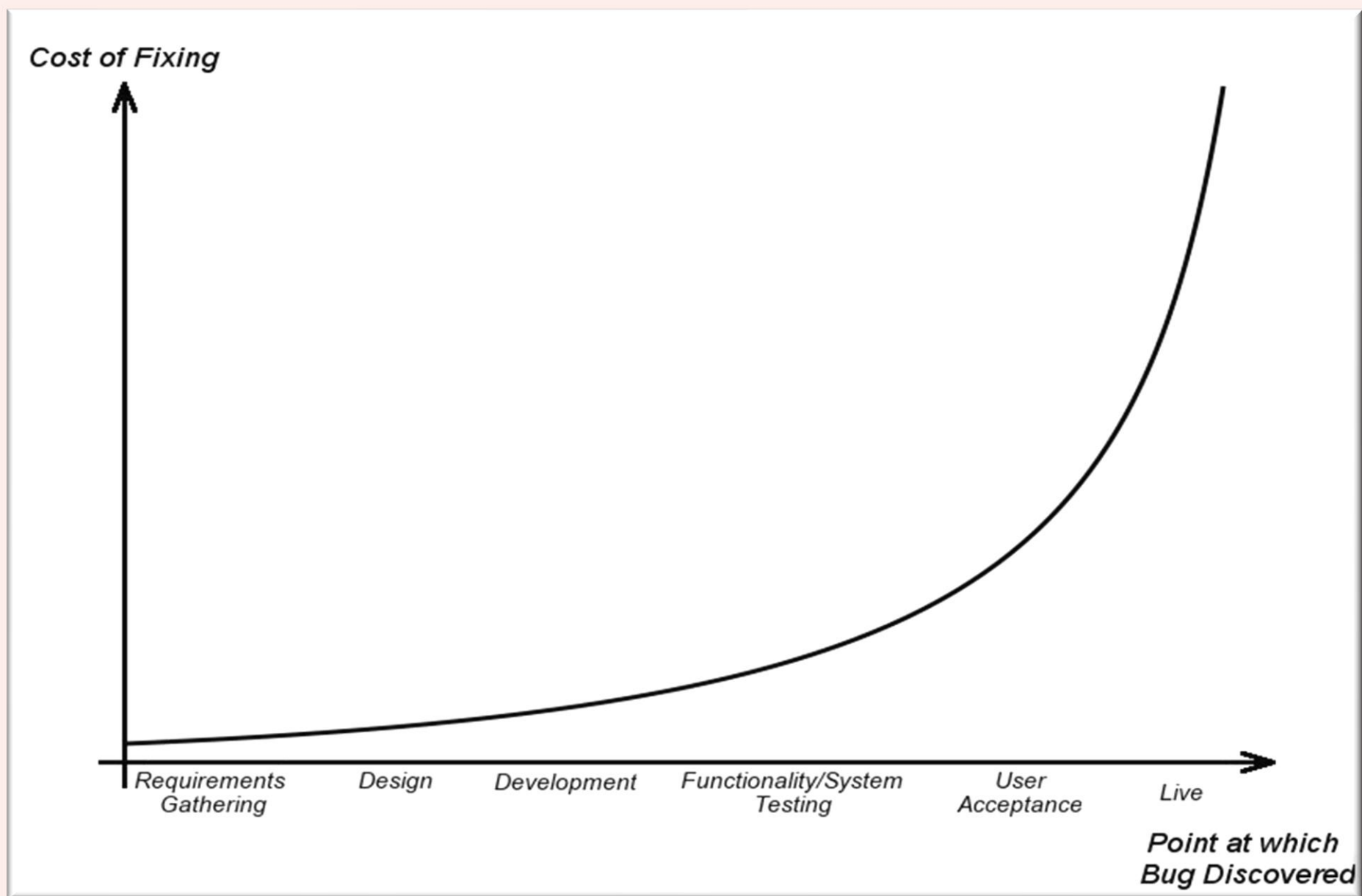


# VERIFICATION & VALIDATION

---



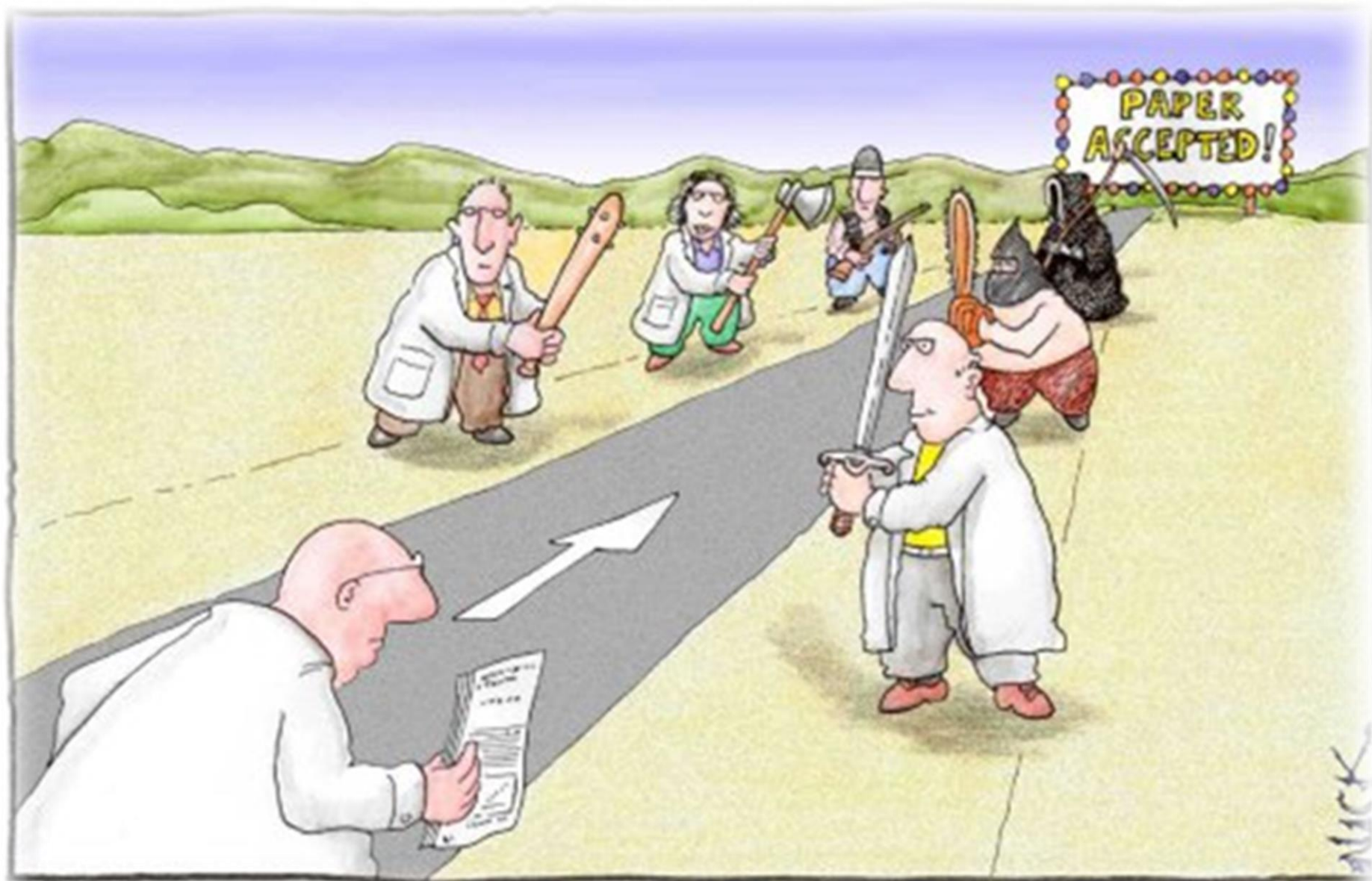




---

# WHY IS SOFTWARE TESTING NECESSARY?





# REQUIREMENTS/ DESIGN/ REVIEW

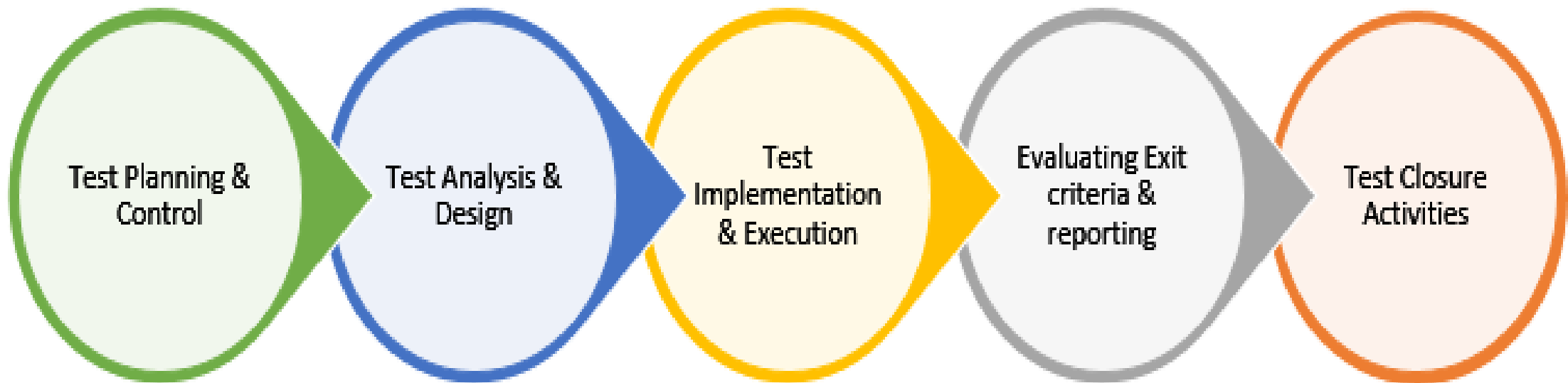
---

**WE ARE FINDING A DEFECT IN REVIEW 9 TIMES FASTER THAN IN TESTING.**

**WE ARE SOLVING A DEFECT FOUND IN REVIEW 5 TIMES FASTER THAN A DEFECT FOUND IN TESTING.**

# THE FUNDAMENTAL TEST PROCESS

---



[www.letzdtesting.com](http://www.letzdtesting.com)

# TEST PLANNING AND CONTROL

---

## TEST PLANNING

### Activities:

- Determine the scope and risks and identify the objectives of testing
- Determine the test approach (techniques, test items, coverage, identifying the testing team, resources)
- Schedule test analysis and design tasks, test implementation, execution and evaluation
- Determine exit criteria

**Deliverables:** Test Policy, Test Strategy, Test Plan

## TEST CONTROL

### Activities:

- Compare actual progress against the planned progress
- Monitor and document progress, test coverage and exit criteria and provide information on testing through reports
- Initiate corrective actions, if necessary

**Deliverables:** reports, test data

# TEST ANALYSIS AND DESIGN

---

The phase where general testing objectives are transformed into tangible test conditions and test designs.

**Activities:**

- Review the test basis examining the specification for the software that we are testing
- Identify test conditions
- Design the high-level tests
- Evaluate testability of the requirements and the system
- Design the test environment setup and identify any required infrastructure and tools

**Deliverables:** test conditions, test suites

# TEST IMPLEMENTATION AND EXECUTION

---

## TEST IMPLEMENTATION

### Activities:

- Develop and prioritize our test-suites
- Create scenarios and test-cases derived for efficient test execution
- Implement / configure and verify the test environment

**Deliverables:** test data, test cases, test scripts

## TEST EXECUTION

### Activities:

- Execute the test-suites and test-cases
- Compare actual result with expected results
- Log the outcome of the test execution, report discrepancies (bugs, defects, error reports)
- Re-execute the tests that previously failed in order to confirm the fixes

**Deliverables:** test-execution reports, issues, issue-reports

# EVALUATING EXIT CRITERIA AND REPORTING

---

Evaluating exit criteria is the activity where test execution is assessed against the defined objectives.

**Activities:**

- Check the test logs against the exit criteria specified in test planning phase
- Assess if more tests are needed or if the exit criteria specified should be changed
- Write a test summary report for stakeholders

**Deliverables:** Test Summary Report, other reports

# TEST CLOSURE ACTIVITIES

---

Closing the corresponding testing activities.

**Activities:**

- Check which planned deliverables are actually delivered and to ensure that all incident reports have been resolved
- Finalize and archive testware such as scripts, test environments, etc. for later reuse
- Handover the testware to the maintenance organization. They will give support to the software and make any bug fixes or maintenance changes.
- Evaluate how the testing went and learn lessons for future releases and projects

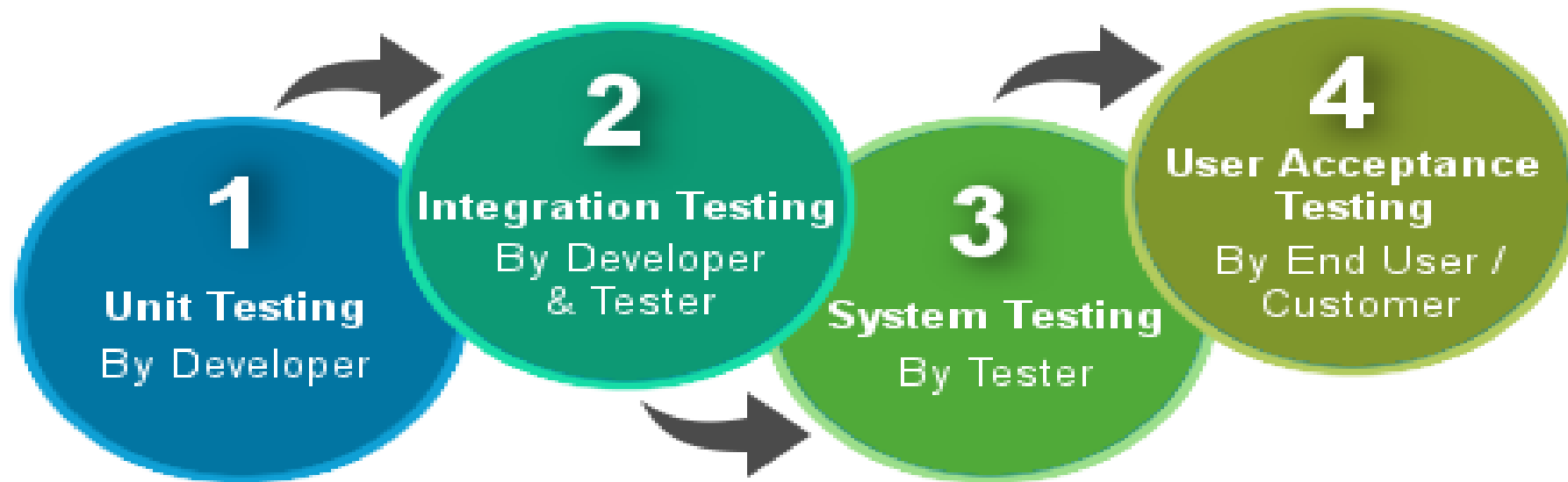
**Deliverables:** testware, archives with testware; process improvement suggestions



# TEST LEVELS

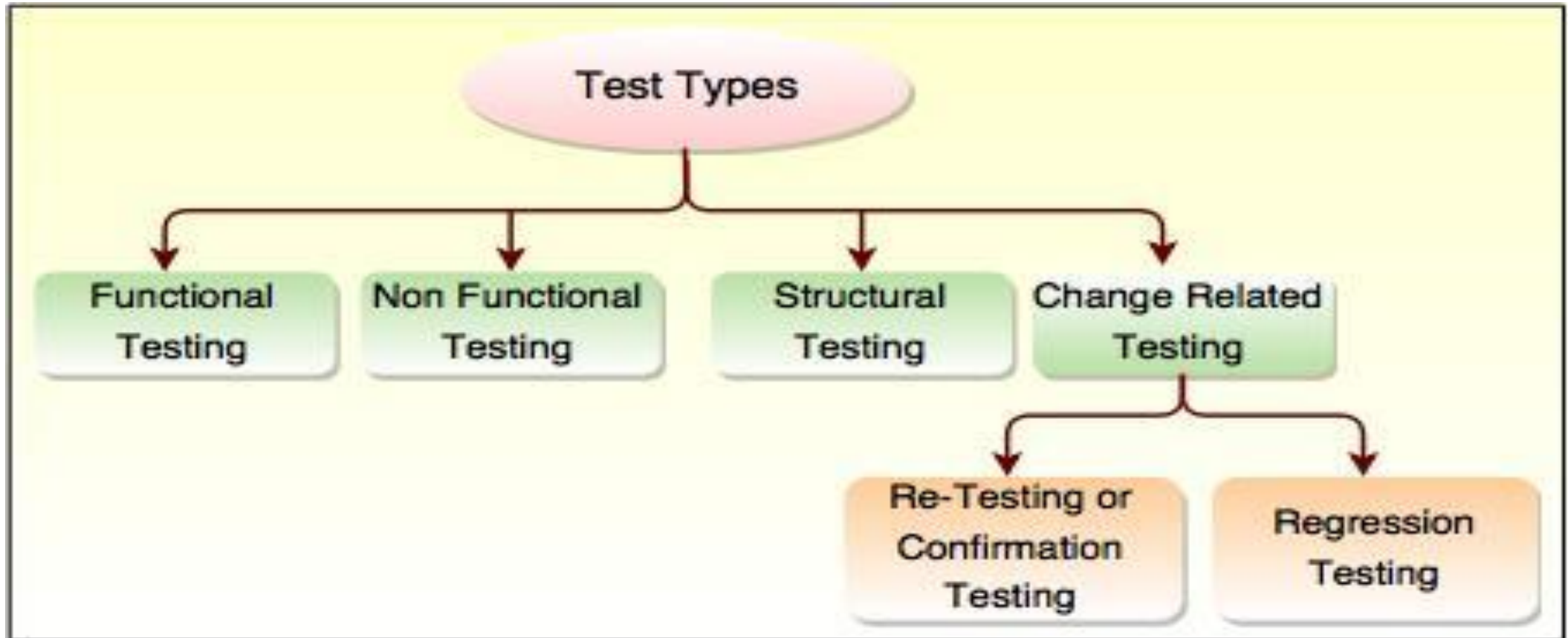
---

## Levels of Testing



# TEST TYPES

---



# HOW DOES ENDAVA DO IT?

---

## AGILE SOFTWARE DEVELOPMENT

- The majority of the projects in Endava follow the **ITERATIVE** software development model -> the software delivery is divided into increments or builds, each increment adding new functionality.
- To be able to deliver value faster, with greater quality and predictability, Endava successfully implemented **AGILE** software development, approach that builds software incrementally through collaboration between clients and cross-functional teams.
- **SCRUM** and **KANBAN** are two of the most widely used AGILE methodologies in Endava.

# AGILE SOFTWARE DEVELOPMENT USING SCRUM

---

## SCRUM METHODOLOGY

The **SCRUM methodology** is a framework for project management where the main iterations (features) of a project are broken down into small short iterations called sprints.

After the end of each sprint, a *build* of the project is delivered to the client.

# AGILE SOFTWARE DEVELOPMENT USING SCRUM

---

## SCRUM METHODOLOGY

A **SCRUM Team** is usually formed by 5 to 9 members working together to deliver the required product increments.

The common **roles** within a SCRUM Team are:

- **Product Owner** – “translates” the customer needs to the whole team; refines the Project Backlog
- **Scrum Master** – ensures that the process runs smoothly, removes impediments, organizes critical events and meetings
- **Business Analyst** - gathers, documents, and analyzes business needs and requirements; solves business problems and designs technical solutions
- **Developer** – writes the code for the project
- **Tester** – tests the code

# AGILE TESTING

---

Agile testing is a continuous process rather than being a sequential one.

- Agile testing begins at the start of the project and there is ongoing integration between testing and development; the common objective of development and testing being to achieve a high quality product.
- The testers participate at every Scrum meeting and they constantly give their input, having in mind all the Fundamental Test Process activities.
- **Jira Software** is the most common **AGILE Project Management tool** that is designed to support any AGILE methodology, be it Scrum, Kanban, or something else. From sprint boards to reports, the team can plan, track, and manage all their software development projects within a single tool.
- But there are others: **Rally, Azure, Helix.**

# AGILE TESTING WITH JIRA/AZURE

---

## JIRA WORKFLOW

- At the beginning of a **sprint** (which can last between 1 to 3 weeks), a list of **Backlog Items** (also called “user stories”) is created and it is put in the **Sprint Board** in Jira.
- Every user story from the sprint board is then picked-up by a Developer, coded, reviewed and, tested by a Tester and then put on DONE.
- When doing the testing, the Tester has in mind follows all the Fundamental Test Processes.
- After every user story from the sprint is put to DONE, a code merge is made and, at the end of the sprint, a functional piece of software is delivered to the client.
- Then another sprint starts and so on, until the project is fully delivered.

# HOW DOES A SPRINT BOARD LOOK LIKE IN JIRA?

BO2 Team

BO Team 2 Sprint 20.5 A

Do your best as usual!

QUICK FILTERS: To be merged Performance Tech refinement Only My Issues Bas Corina Franck Lucian Simon Roxana Adriana Hide Done Refinement Show subtasks Documentation Recently Updated

6 days remaining Complete Sprint Board

TO DO DEVELOPMENT REVIEW TESTABLE TEST DONE

Hide subtasks/defects 8 issues

DEF-1003  
[PERF]Transaction timeouts in job PaymentAgendaJob, PaymentAgendaService for contract  
In Development  
UNDER REVIEW  
None  
13

DEF-992  
Manually settling a paymentspec is not possible when there is more than one payment  
CMSP aftercare on aftercare  
Ready for Test  
UNDER REVIEW  
None  
3

DEF-987  
Close dossier page  
Benu sluitcodes CMSP  
In test  
UNDER REVIEW  
None  
3

DEF-981  
Benu sluitcodes: Service to retrieve fulfilmentcode for Benu  
Benu sluitcodes CMSP  
Done  
MERGED  
Defacto 1.150.0  
2



# HOW DOES A **SPRINT BOARD** LOOK LIKE IN AZURE?

The screenshot displays the Azure DevOps Sprints interface. The top navigation bar includes the Azure DevOps logo, the breadcrumb 'infomedics / Test automation / Boards / Sprints', a search bar, and utility icons. The left sidebar lists navigation options: Test automation, Overview, Boards, Work items, Backlogs, Sprints, Queries, Plans, Repos, Pipelines, Test Plans, and Artifacts. The main area shows the 'Test automation Team' sprint board for the period 'March 31 - April 27' with '12 work days remaining'. The board is divided into columns: 'New', 'In Progress', 'Review', and 'Done'. A 'Taskboard' tab is active, with options for 'Backlog', 'Capacity', and 'Analytics'. A '+ New Work Item' button and 'Column Options' are also visible. In the 'New' column, three work items are listed: '6305 Research: Claimfile generation solution' (Active), '6306 Library solution' (New), and '6307 Template solution' (New). In the 'Done' column, one work item is listed: '6309 DB update solution (after first claimfile upload)' (Closed). Each work item card shows the title, assignee, state, and a time box or label.

# WHAT IS A **USER-STORY**?

---


Is a small, self-contained unit of development work designed to accomplish a specific goal within a product.

- We can look at it as a (software) requirement formulated in everyday language so it can be easily understood by everyone who reads it; usually follows the simple format: -> “As a user I want to perform this action so that I can accomplish this goal.”
- It can represent a user's need, serve as a planning item in agile software development, or simply be used as a basis for a discussion.
- The user story content can be detailed to also contain a background, a more substantial description of the feature that needs to be coded, technical notes, testing hints, acceptance criteria, definitions of done, definitions of ready etc, depending on the type of project or team`s needs.

# HOW DOES A **USER-STORY** LOOK LIKE IN JIRA

Dashboards ▾ Projects ▾ Issues ▾ Boards ▾ Create

Search 🔍

 OHW / OHW-2

Install project on TST enviroment

Edit Comment Assign More ▾

Share

Details

Type: **+** User story

Priority: **🔴** Major

Status: **DONE** (View Workflow)

Resolution: Done

Labels: None

Story Points: 3

Fix Version/s: OnderHandenWerk 1.0.0

Epic Link: **Finance OHW**

Team: BO2 Team

Sprint: BO Team 2 Sprint 19.14, BO Team 2 Sprint 20.01, BO Team 2 Sprint 20.2 a

Merged: Yes

Database changes: No

Property Changes: No

Documentation Updated: No

Description: 

What is it that the PO wants?

The OHW module must be installed on a TST enviroment, Ron Boukens can help with this.

Why do we build this?


So we can test the module


When will it be good enough?

We can see and access the OHW-module

also the sftp server should be configured - just like it was for dcim

People

Assignee:  Adriana Oana Oros

Reporter:  Bas Brinkman

Watchers: **1** Start watching this issue

Dates

Created: 10-Dec-2019 13:39

Updated: 11-Feb-2020 14:50

Resolved: 03-Feb-2020 16:15

Development

Create branch

Agile

Completed Sprints: BO Team 2 Sprint 19.14 ended : BO Team 2 Sprint 20.01 ended : BO Team 2 Sprint 20.2 a ended

View on Board

43 <<

ndava

# WHAT IS A TEST SCENARIO? WHAT IS A TEST-CASE?

---

## TEST SCENARIO

- Any functionality that can be tested, also called Test Condition or Test Possibility.
- Gives a high-level idea of what we need to test.
- Is a collective set of test cases which helps the testing team to determine the positive and negative characteristics of the project.
- As a tester, you should put yourself in the end user's shoes and figure out the real-world scenarios and use cases of the applications under test.

## TEST CASE

Set of actions executed to verify a particular feature or functionality of the software application.

It contains:

- Input values
- Preconditions
- Steps to reproduce
- Expected results
- Postconditions

# HOW DOES A **SCENARIO & TEST-CASE** LOOK LIKE IN JIRA

Corina Cretu added a comment - 22-Apr-2020 14:50




Tested on: <http://buildserver.famed-linux.int:8080/job/DEF-982/11/>

Issue  OK


## Scenarios tested:

1. For a Benu nota in Credios /Bailiff trajectory a fulfilmentcode for contract 75000 is used when it is configured
2. For a Benu nota in Credios /Bailiff a generic Famed fulfilmentcode is used when a fulfillmentcode is NOT configured for contract 75000
3. For a Famed nota in Credios/Bailiff/Debt Surveillance trajectory a generic Famed fulfilmentcode (where contract\_id is null) is used or the configured fulfilment code is used when a code is configured for a specific Famed contract



## Test-case 1:

1. Import a Benu nota for contract 75000 (or 75001)
2. Add a new fulfillmentcode (eg: 853 REBATE) and configure it for contract 75000
-  I configured the fulfillmentcodes from Defacto UI from Beheer > Creditmanagement > Onderhouden incassocodes
-  the generic Famed fulfillment code 853 in defacto db is WRITE OFF when contract\_id = null
3. Get the Benu nota into 'Transferred to Credios" status, then import a closeInvoice file with configured fulfillmentcode 853
-  Expected and actual result -> a REBATE is performed (not a WRITE OFF) according to the fulfillmentcode configured for contract 75000


## Test-case 2:

1. Import a Benu nota for contract 75001 with no fulfillment code configured for 75000
2. Get the nota into 'Transferred to Credios" status then import a closeInvoice file with fulfillmentcode on MANUAL (eg: 828)
-  Expected and actual result -> the generic Famed fulfillment code is used and a workload task is created so the user can choose what to do further with the nota (rebate, write off, etc)

## Test-case 3:

1. Import a Famed nota for contract 88888
2. Add a new fulfillmentcode (eg: 828 DONE) and configure it for contract 88888
-  the generic Famed fulfillment code 828 in defacto db is MANUAL when contract\_id = null
3. Get the Famed nota into 'Transferred to Credios" status, then import a closeInvoice file with configured fulfillmentcode 828
-  Expected and actual result -> nota is closed with the configured code, no manual task is created

## Test-case 4:

1. Import a Famed nota for contract 88888 with no fulfillment code configured for 88888
2. Get the nota into 'Transferred to Credios" status then import a closeInvoice file with the generic Famed fulfillment code 828 which is on MANUAL
-  Expected and actual result -> a manual task is created so the user can choose what to do further with the nota (rebate, write off, etc)

# TEST-CASE X-RAY EXAMPLE

## Version 1

Created on 13/09/2012 09:01:58 by [REDACTED]

Last modified on 10/01/2013 14:34:14 [REDACTED]

## Summary

Verify that user can't create a new organization setting the flags Trust Center false; Scoring center nothing checked ; Health Self Management false ; Initial Call false ; [REDACTED]=0 but i can save if it creates and then adds a new default coach (DefCoach=1).

This are the settings for Moove organization

## Preconditions

Admin credential should be known before starting the test

#	Step actions	Expected Results	Execution		
1	Login using the admin credential Click on Control Panel Under "Portal" section click on "User and Organizations" link Click on Add button and select "Regular organizations" (from the dropdown) Fill in Name and press Save	Admin home page should load Liferay control panel page should load "User and Organizations" page should load Add "New Organization" page should load New organization should be created (a message should be displayed: "Your request completed successfully"); custom field should appear under "Organization Information" page	Manual	✗	+
2	Click on "Custom fields: under "Organization Information" section	Custom field page should load	Manual	✗	+
3	Set the fields Trust Center to false; Scoring center nothing checked; Health Self Management false ; Initial Call false and then press Save (at this moment there is no default coach present on that organization)	An error should be shown a this configuration can't be saved	Manual	✗	+
4	Under "Portal" section click on "User and Organizations" Click on the organization created before Add a new default coach for that organization	"User and Organization" page should load Organization page should load New default coach for that organization should be created	Manual	✗	+
5	Click on Edit organization button and then on "Custom Fields"	Organization's custom field page should load	Manual	✗	+
6	Set the fields Trust Center to false; Scoring center nothing checked; Health Self Management false ; Initial Call false and then press Save (at this moment there is a default coach present on that organization)	This configuration should be saved	Manual	✗	+

## 6284 Vecozo import



[Redacted]

0 comments

Add tag

State ● Design

Area



Reason 🔒 New

Iteration

### Summary ❗

*Click to add Summary*

### PreSteps

<https://tst->



### Steps

📎 📎 📎 | ↑ ↓ ✖ | [@] 📎 | **B** *I* U

Steps Action

Expected result

Attachments

1. Open the N Admin application web site: <https://tst->



Admin app is opened

2. Go to Importeren > VECOZO

Vecozo page is opened

3. Select the "Ingangsdatum" and put a start date for the import

Start date is selected

4. Press the button

Pop-up to select the file is opened

5. Select a file (excel) and click Open

The file name can be seen at "Bestandsnaam" column and has a green icon with "?" mark (if the icon is red, the file is malformed)

6. Press the "Verwerk verzekeringen" button

Import is done - A green feedback message appears in the upper-right corner of the page and the file now have green icon with checked/done mark

7. Check the content of the page

Bestanden table is displayed with the following columns: Uploadada



# TEST-CASE EXCEL EXAMPLE

	A	B	C	D	E	F	G	H	I	J	K
1	Test Case ID	BU_001	Test Case Description	Test the Login Functionality in Banking							
2	Created By	Mark	Reviewed By	Bill	Version	2.1					
3											
4	QA Tester's Log	Review comments from Bill incorporate in version 2.1									
5											
6	Tester's Name	Mark	Date Tested	1-Jan-2017	Test Case (Pass/Fail/Not	Pass					
7											
8	S #	Prerequisites:			S #	Test Data					
9	1	Access to Chrome Browser			1	Userid - [REDACTED]					
10	2				2	Pass = [REDACTED]					
11	3				3						
12	4				4						
13											
14	Test Scenario	Verify on entering valid userid and password, the customer can login									
15											
16	Step #	Step Details		Expected Results		Actual Results			Pass / Fail / Not executed / Suspended		
17											
18	1	Navigate to http://demo.guru99.com		Site should open		As Expected			Pass		
19	2	Enter Userid & Password		Credential can be entered		As Expected			Pass		
20	3	Click Submit		Customer is logged in		As Expected			Pass		
21	4										
22											
23											



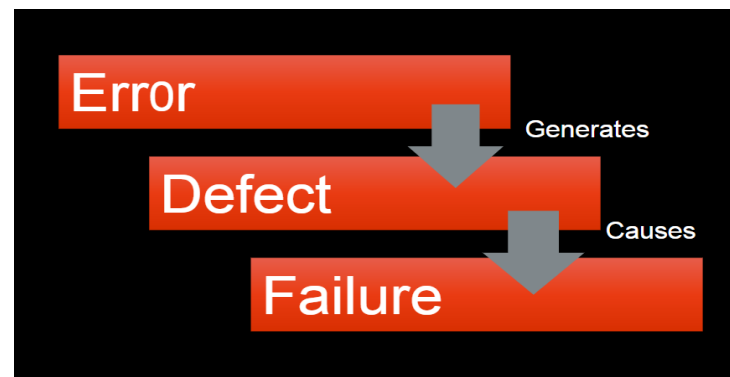
1	Test cases																		
2	Content	Test cases	Remark 1	Formulate everything as clear as possible	Date review														
3	Date	xx/xx/200x			Reviewed by														
4	Authors		Remark 2	Complete every test case as much as possible, status etc.	Date last adjustment														
5	Project				Adjustment by														
6	# Test cases		Remark 3	Write down everything you observe, even when its not specified by the test case															
7																			
8																			
9	General remarks																		
10																			
11	Start	Prepare the environment for the execution of the test cases																	
12				Pre-condition	Action	Location	Value	Expected result	Status	Issues	Remarks								
13																			
14																			
15																			
16	TS	TC	Iterati	Priority	Pre-condition	Action	Location	Value	Expected result	Status	Issues	Remarks							
41		TC_1.2.4.7		Medium		Zoom in and out the page.			Design stays in place.										
42	TS_1.3 Delete a message (Client side)			High		Delete message?			Message deleted?			!!! Recomendation							
43																			
44	TS_1.3 View Messages List and details (Specialist side)																		
45		TC_1.3.1.1		High	Access Quarant application.	Go to user Online02 -> [REDACTED]			Message list appears.										
46		TC_1.3.2.1		High		Select a message.			At the bottom of the list the message (or the conversation between the client and the specialist) will appear).										
47		TC_1.3.2.3		High		Close the message.			Message closed.										
48		TC_1.3.4.1		Medium		Observe all the labels. The size and font of the text; poztion; etc.			Correct design.										
49		TC_1.3.4.2		Medium		Zoom in and out the page.			Design stays in place.										
50																			
51	TS_1.4 Create new message (Specialist side)																		
52		TC_1.4.1.1		High	Access Quarant application.	Go to user Online02 -> [REDACTED]			Message list appears.			IMPORTANT! The Specialist can receive only messages from his clients and he can see only the messages send by him.							
53		TC_1.4.1.2		High		Create a new message with a new topic,using [REDACTED] button from the top menu bar.		subject: Subject message: Text message											
54		TC_1.4.1.3		High		Press 'Opslaan' button from the top menu bar/ Press 'Annuleren' button.			Message sent to the client (add this message to the dossier)/ Cancel the action.										
55		TC_1.4.1.4		High		Check in the frontend if the message is shown also, for the client.			Message created.			IMPORTANT:The specialist should be notified that he received a new message							

# WHAT IS A DEFECT?

---

A human being can make an **error** (mistake) which produces a **DEFECT** (or a BUG, a flaw) in the code.

- If the defect in the code is executed, it can cause a **failure**, but not all defects result in failures.
- In other words, when **actual result** deviates from the **expected result** while testing, then it results into a defect. Hence, any deviation from the specification mentioned in the product functional specification document is a defect.
- Depending on the organization, the defect is called differently like: **bug**, **issue**, **incident** or **problem**.



# HOW DO WE REPORT A DEFECT?

---

## When reporting a defect a tester should:

- Have a clear goal in mind, know exactly what he wants to transmit
- Provide developers, managers and others detailed information about the behavior observed and the defect itself
- Be precise, concise, clear, unambiguous
- Attach screenshots if you can
- Express the observations in a neutral tone, fact-focused and impartial

## If no bug reporting tool is available, a defect report should mainly contain:

- **Defect ID** - Unique identification number for the defect
- **Severity/priority**
- **Version** - Version of the application in which defect was found.
- **Date Raised** - Date when the defect is raised
- The **name of the person** who found the defect
- **Reference** to the documents like requirements, design, architecture to help others understand the defect
- **Status** of the defect
- **Description** of the defect with:
  - Steps to reproduce
  - Expected result
  - Actual result
- **Screen shots, logs, videos** which capture and prove the defect found



## If "declaratiowaarde" is empty for one DBC, the operation should be rejected

Edit Comment Assign More ▾

### Details

Type: Bug  
Priority: Major

Status: **DONE** (View Workflow)  
Resolution: Done  
Fix Version/s: [REDACTED]

Labels: None  
Story Points: 0.5  
Epic Link: Finance OHW  
Team: BO2 Team  
Sprint: BO Team 2 Sprint 20.2 b  
Merged: Yes  
Database changes: No  
Property Changes: No  
Documentation Updated: No  
Description:

#### Steps to reproduce:

1. Import file [REDACTED]nedeclaratiowaarde\_0000.asc with 1 patient having 2 operations (DBC's)
2. Modify 1st operation to have on "declaratiowaarde" field (which is a M field for [REDACTED] "000000" value -> which is interpreted as being empty, because it is a numeric field on Vektis
3. The 2nd operation has on "declaratiowaarde" field [REDACTED] value
4. The line 99 has on [REDACTED] select [REDACTED] amount

Actual result -> the file is processed, but the 1st operation (the one with empty value) is PREFINANCED



Expected result -> the file should be processed, but the 1st operation should have the status REJECTED

### People

Assignee: Corina Cretu  
Reporter: Corina Cretu  
Watchers: 2 Stop watching this issue

### Dates

Created: 27-Feb-2020 13:13  
Updated: 10-Mar-2020 12:44  
Resolved: 10-Mar-2020 12:35

### Development

2 branches  
7 commits  
2 pull requests **MERGED**

Create branch

### Agile

Completed Sprint: BO Team 2 Sprint 20.2 b en  
View on Board

6667 Trying to make a payment, user is stuck

 Unassigned

0 comments

State ● New

Area	Test automation
------	-----------------

Reason  New

Iteration Test automation\TA-03

### Description

### General description

Using the Payment page, you cannot make a payment using the new simulators:

### Steps to reproduce

1. Open PP using an invoice id for which you can make the payment
2. Select a bank
3. Select any of the simulators in the list: Sim1 or Sim2
4. Press "make a payment" button.

### Current result

I am redirected to a blank page. The payment is not made.

**Expected result**

I should have been redirected to simulator page, and the payment should be completed.

Note: check attachments section.



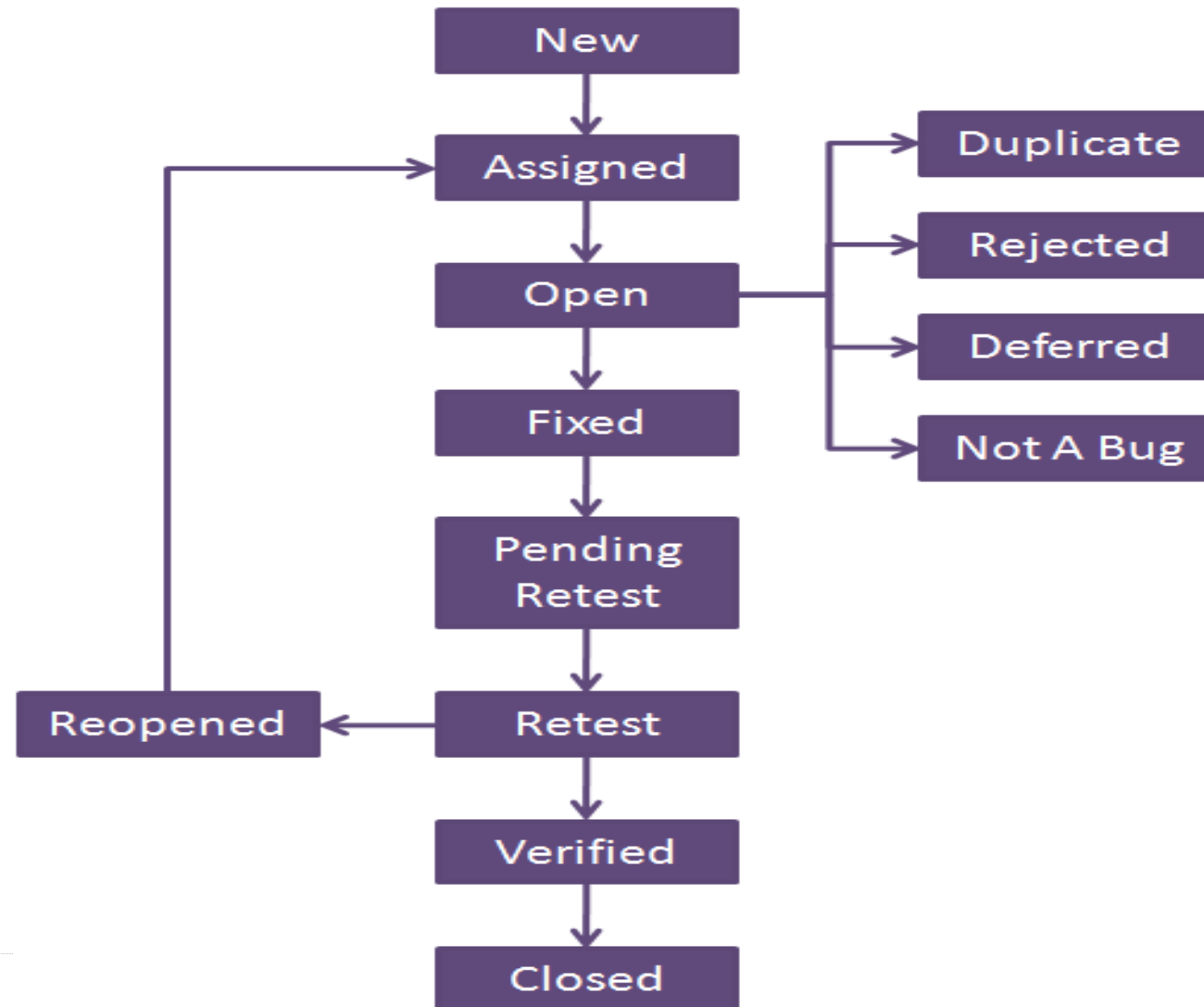
## Discussion



Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.

# DEFECT LIFE CYCLE EXAMPLE

---

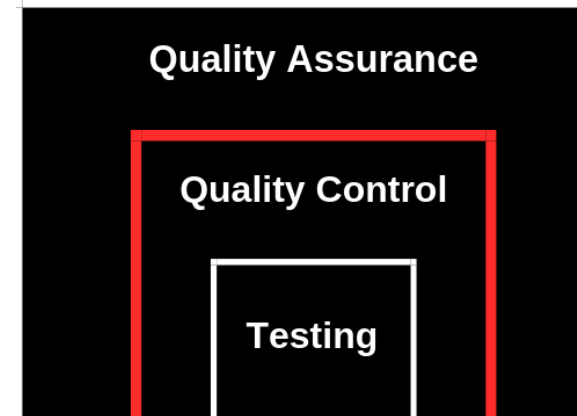


# TESTING & QUALITY

---

Testing is **part** of quality assurance and it contributes to higher quality.

- Testing helps in **measuring** the quality of software in terms of the number of defects found, tests run, test coverage.
- Testing **identifies defects as early as possible**, before the software is in use, in this way avoiding unwanted cost of the stakeholders.
- Testing activities and QA activities complete each other:
  - the goal of a **Tester** is to find bugs, find them as early as possible and make sure they get fixed.
  - the goal of a **QA person** is to create and enforce standards and methods to improve the development process and to prevent bugs from ever occurring.
- Testing **gives confidence** in the quality of the software, because when testing find defects, the quality of the software increases if those defects are fixed.
- Testing **proves** that the software products are fit for purpose.





# NEWS AND DIRECTIONS IN TESTING AREA

---

## AUTOMATION TESTING



# PANDEMIC SITUATION...

---

**WFH**









# PANDEMIC SITUATION...

---

**ONLY A VERY SMALL PERCENT OF  
CLIENTS REDUCED COSTS**

# PANDEMIC SITUATION...

---

## CONFERENCES, WORKSHOPS AND EVENTS

# PANDEMIC SITUATION...

---

**NEW TEAMS,  
TEAM BUILDINGS  
AND CLIENT RELATIONSHIP**

# PANDEMIC SITUATION...

---

**MORE TIME WITH FAMILY AND FURRY  
FRIENDS**











# RESOURCES & BIBLIOGRAPHY

---

## USEFUL SITES:

<https://artoftesting.com/>  
<https://www.atlassian.com/agile>  
<https://www.softwaretestingtricks.com/>  
<https://www.guru99.com/>  
<http://tryqa.com/>

## BOOKS:

*Foundation of Software Testing* – Dorothy Graham

*Agile Testing. A Practical Guide For Testers and Agile Teams* – Lisa Crispin & Janet Gregory



**QUIZ TIME! LET`S PLAY...**



# QUIZ TIME! LET`S PLAY...

- ✓ Access: [menti.com](https://menti.com)
- ✓ Introduce the [code](#) mentioned in the course!
- ✓ Make sure you fill in your entire name and group number, e.g. [RoxanaSoporan\\_231](#)





**ROXANA SOPORAN**  
**ALEXANDRU ANDREI**

roxana.soporan@endava.com

alexandru.andrei@endava.com