

## Program szkolenia:

# Fundamenty testowania

### Informacje:

<b>Nazwa:</b>	<b>Fundamenty testowania</b>
<b>Kod:</b>	<b>QA-fun</b>
<b>Kategoria:</b>	Testowanie dla QA
<b>Odbiorcy:</b>	testerzy
<b>Czas trwania:</b>	2 dni
<b>Forma:</b>	50% wykłady / 50% warsztaty

---

Fundamenty są ważne gdy wdrażasz w organizacja testowanie automatyczne oraz gdy chcesz do nich wrócić gdy próbujesz uporządkować strategię testowania, która przez lata powstawała intuicyjnie.

Uczestnicy podczas szkolenia przejdą przez praktyczne przykłady najistotniejszych aspektów testowania automatycznego aby świadomie stworzyć strategię testowania w swojej organizacji.

### Zalety szkolenia:

- Zorientowanie na strategię tesowanie
- Najlepsze wzorce i praktyki
- Porządkowanie pojęć

## Szczegółowy program:

### 1. Introduction to testing

1.1. What is testing?

1.2. What is QA?

1.3. What is QA...not?

1.4. QA Roles in project

1.4.1. Tester

1.4.1.1. Can a tester be a Business Analyst?

1.4.2. Defect Manager and Quality Manager - Big and small projects

1.4.3. The Enemy of the Tester - Developer

1.5. Cooperation with non QA people - One Team

1.6. When testing is done?

1.6.1. Exit criteria

1.7. Testing for better Software

1.8. Testing Documents

1.8.1. Testing strategy

1.8.2. Testing policy

### 2. Levels of testing - Based on V Model

2.1. Unit Testing

2.2. Integration Testing

2.3. System Testing

2.4. Acceptance Testing

2.5. System Integration Testing

### 3. Types of testing

3.1. Functional vs Nonfunctional

3.2. Blackbox vs Whitebox

3.3. Manual vs Automated

3.4. Exploratory vs scenario

3.5. Regression Tests

#### 4. Testing techniques

4.1. Whitebox Techniques

4.2. Blackbox Techniques

#### 5. Defects

5.1. What is a defect

5.2. Cost of a defect and V- Model

5.3. Herd of defects

5.4. Defect Lifecycle

5.4.1. Basic statuses

5.4.2. Standard

5.4.3. Optional cases statuses

5.4.4. Acceptance Testing with internal confirmation statuses

5.5. A Perfect Defect Report

5.5.1. Severity vs Priority

5.5.2. Steps to reproduce

5.5.3. Expected result

5.5.4. Other fields

5.6. Defect Issues and special cases

5.6.1. Defect Ping-pong in Integration and Acceptance testing

5.6.2. Never to be solved low defects

5.6.3. Testability of application vs Defects

5.6.4. Specification Defects Gaps

5.6.5. Test Case Defects

5.6.6. Who creates a defect?

5.6.7. Repairing a defect without analysis

5.6.8. Defect Antipatterns

5.6.8.1. Defect in e-mail

5.6.8.2. Insulting

5.6.8.3. Personal opinion

5.6.8.4. Too much information

5.6.9. Acceptance testing: Who pays for defect and smuggling new functionality

5.6.10. Defect Decision Board - Sprint with defects

5.6.11. Sinking with defects - Bug busters week

5.6.12. Consultation hours

## 6. Metrics in scope of quality

6.1. Are metrics really useful when talking about quality

6.1.1. Quality of the system

6.1.2. Quality of the testers work

6.1.3. Quality of developers work

6.2. Examples of QA metrics

6.2.1. Count of defects

6.2.2. Number of reopens

6.2.3. Tests passed vs Test Failed - Greener is better

6.2.4. Coverage reports

## 7. A Good Test Scenario

7.1. Attributes of test scenario

7.2. Good Case vs Bad Case

7.3. Granularity - Level of details in steps

7.4. Number of steps in scenario

7.5. Expected Result

7.6. Can test be a specification?

## 8. Requirement Based Testing

8.1. Bottom2up

8.2. Up2Bottom

## 9. Risk Based Testing

## 10. Test Automation

10.1. Advantages of automation

10.1.1. Regression of automation

10.2. Disadvantages

10.2.1. Cost of automation

10.2.2. Maintenance

10.3. Web Testing - Page Object Model

10.3.1. DDT - Data Driven Testing - introduction

10.3.2. One test - one assert

10.3.3. One test for different Dumps DEV, INT, PROD

10.3.4. One test for different browsers

10.3.5. One test for different resolutions

10.3.6. Business scenario in code

10.4. Automation Tools

10.4.1. Junit/TestNg Framework

10.4.1.1. TestSuite vs Test

10.4.1.2. Setup, WrapUp, Before, After

10.4.1.3. Naming convention - Clean Code

## 11. Other QA Topics

11.1. Estimation in testing

11.1.1. Small changes in code, big effort to test

11.1.2. Big changes in code, hardly nothing to test

11.1.3. Risks, and increasing effort

11.2. Reviews

11.2.1. Spec Reviews

11.2.2. Code Reviews

11.2.3. Test Case Review