

# Test Management and Issue Tracking

# Test Roles, Test Strategy and Approach, Entry and Exit Criteria, Planning, Design, Execution, Monitoring, Closure



# SoftUni



Software University

<https://about.softuni.bg>

## SoftUni Team

## Technical Trainers



## 1. Roles and Responsibilities in Test Management

- Test Manager, Test Lead, Test Analyst, Test Engineer, Manual Tester

## 2. Components of Test Management

- Test Planning and Estimation
- Test Design
- Test Execution
- Test Monitoring and Control
- Test Closure

## 3. Test Management Tools

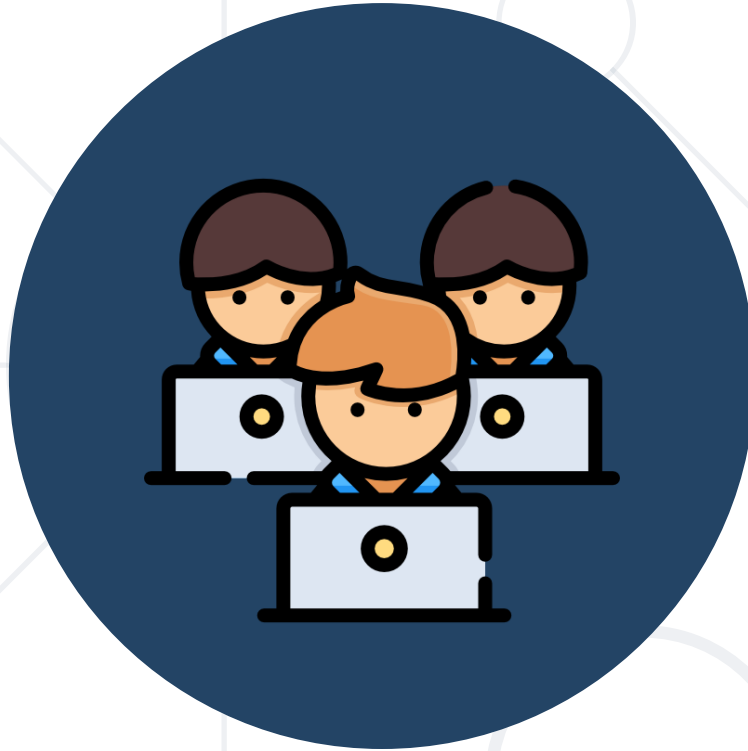
- TestRail, Jira



# Have a Question?

**sli.do**

**#QA-Fund**



# **Roles and Responsibilities in Testing**

Navigating the Testing Landscape

# Roles in Testing

- Most **common roles** in testing are:
  - Test Manager
  - Test Lead
  - Test Analyst
  - Test Engineer
  - Manual Tester
- **Other roles** can be found, depending on the organization, e.g. **Test Coordinator**



# Test Manager

- Responsible for **overseeing the testing process** and ensuring its successful execution
  - **Develops** the test strategy and test plans
  - Allocates resources, defines timelines, and **sets priorities**
  - **Coordinates** with stakeholders for clear communication and understanding of testing goals
  - **Monitors** the progress of testing activities and report on test metrics
  - **Manages** risks and issues related to testing
  - Provides **guidance** and **mentoring** to the test team
  - Ensures **compliance** to **testing standards** and best practices
  - **Collaborates** with project managers and stakeholders to align testing activities with project objectives



# Test Lead

- **Manages** the **testing team** and plays a crucial role in coordinating and driving testing activities
  - **Assists** in developing the test strategy and test plans
  - **Assigns** tasks to test team members and **monitors** their progress
  - Conducts **regular team meetings** to discuss test progress, challenges, and solutions
  - Provides **guidance** and **support** to test analysts and engineers
  - **Reviews** test artifacts such as test cases, scripts, and data
  - **Coordinates** with the Test Manager to ensure appropriate resource allocation
  - **Collaborates** with other leads and managers to ensure testing integration and alignment with project activities
  - **Identifies** and **resolves** issues or roadblocks that may impact testing



# Test Analyst

- Responsible for **analyzing requirements** and **designing** test scenarios and cases
  - **Reviews** system requirements and specifications to identify testable features
  - **Designs** test scenarios based on business and technical requirements
  - **Creates** and **executes** test cases, scripts, and data
  - **Identifies** defects, working with the development team to resolve them
  - **Participates** in test estimation and test strategy discussions
  - **Conducts** various types of testing
  - **Collaborates** with the Test Lead and Test Manager
  - **Provides input** for test automation opportunities and support automation efforts





# Test Engineer

- Focuses on the technical aspects of testing, including test automation and test execution
  - **Develops** and **maintains** automated test scripts
  - **Collaborates** with the Test Analyst to identify test cases suitable for automation
  - **Sets up** and **configures** test environments and test data
  - **Executes** automated test scripts and analyze test results
  - **Debugs** and **troubleshoots** issues in the test automation framework
  - **Creates** and **maintains** test documentation, such as test plans and reports
  - **Collaborates** with the development team to ensure proper test coverage
  - Continuously **improves** test automation processes and frameworks
  - Stays **updated** with industry trends and best practices in test automation



# Manual Tester

- Responsible for **executing tests manually** to ensure software quality
  - **Reviews** and **understands** software requirements and specifications
  - **Develops** test cases based on functional and technical specifications
  - **Executes** test cases manually to verify software functionality
  - **Identifies** and **reports** software defects using defect tracking tools
  - **Validates** and **verifies** bug fixes after they have been resolved
  - Performs **regression testing** to ensure existing functionalities are not impacted by changes
  - **Documents test results**, including defects and test execution status
  - **Collaborates** with the development team to resolve issues and improve software quality
  - **Provides feedback** and suggestions for improving test processes and test coverage





# Test Management

Coordinating Activities

# What is Test Management?

- **Integral part** of the Software Testing Lifecycle (**STLC**)
- Encompasses the **planning, coordination, and control** of testing activities within SDLC
- **Ensures** that **testing** is **carried out effectively, efficiently,** and in alignment with **project objectives**
- Plays a **vital role** in delivering high-quality software and mitigating risks
- Provides the **framework and guidelines** for managing testing activities within each phase of the STLC
- Ensures that **testing objectives, strategies, and approaches** are **aligned** with the specific phase of the STLC



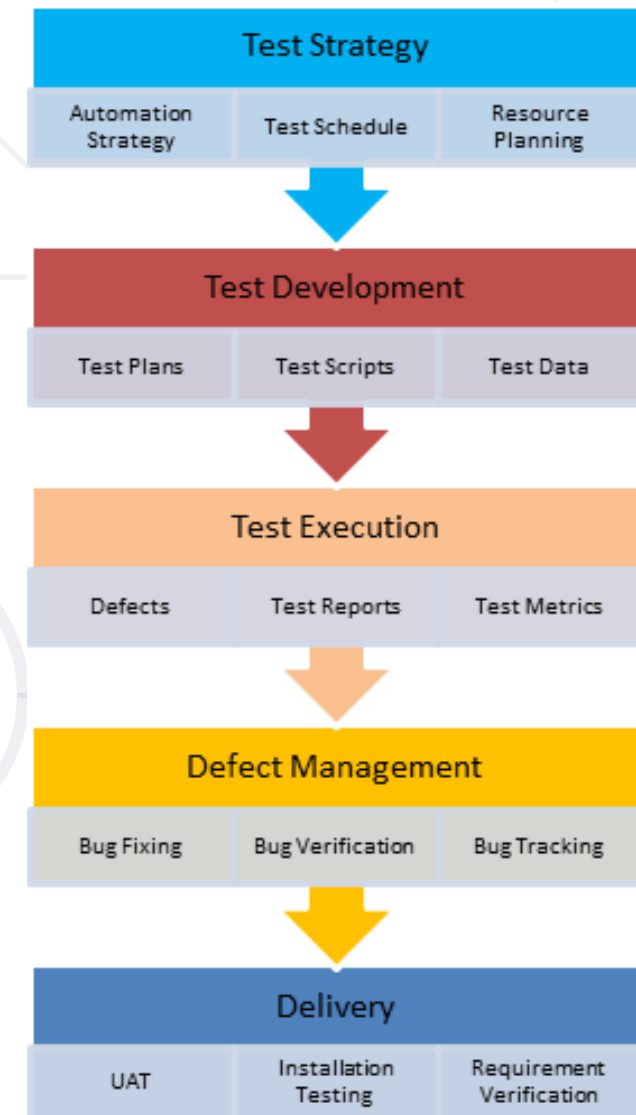
- **Test Strategy** sets the **overall direction, objectives**, and **scope** of the testing effort
- It defines the **approach to be followed**, considering factors such as test levels, techniques, and resource requirements
- The Test Strategy is **developed early** in the testing process, **aligning** with the **project's goals** and quality objectives
- It provides a **roadmap** for the testing team, ensuring that testing efforts are focused and effective
- The Test Strategy also takes into account factors like **risk assessment**, **timelines**, and stakeholder **expectations**

# How to Prepare a Good Test Strategy

- Every organization has their unique priority and set of rules for software designing, so do not copy any organization blindly, include the following in your Test Strategy:

- **Scope**
- **Test Approach**
- **Test Environment**
- **Testing Tools**
- **Release Control**
- **Risk Analysis**

[Sample Test Strategy Template](#)



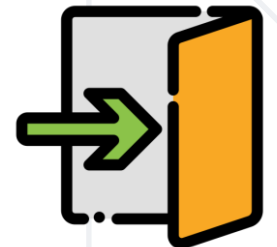
- **Test Approach** is a **more detailed** document **derived from the Test Strategy**, providing **specific instructions** for executing testing activities within each phase
- **It defines:**
  - Process of testing
  - Testing levels
  - Roles and responsibilities of each team member
  - Types of Testing ( Load testing, Security testing, Performance testing etc.)
  - Automation tools if applicable
  - Adding new defects, re-testing, Defect triage, Regression Testing and test sign off

# Different Test Approaches

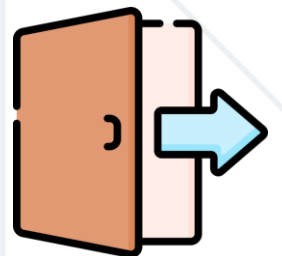
- **Analytical** - Focusing testing on the most critical functionality (risk based)
- **Model-based** - Stochastic or Monkey testing using random or statistical information (tool), Operational profiles
- **Methodical Testing approaches** - Failure based (error guessing and fault attack), Experience-based, Check-list based and Quality characteristic-based
- **Process- or standard-compliant Testing approach** - Industry-specific standards (e.g., medical, aviation), Various agile methodologies
- **Dynamic and heuristic approaches** - such as exploratory testing (more reactive approach than pre-planned approach), Execution and evaluation are concurrent tasks
- **Consultative approaches** - Test coverage is driven primarily by the advice and guidance of technology and/or business, Domain experts outside the test team
- **Regression-averse approach in Software Testing** - Includes reuse of existing test material, extensive automation of functional regression tests, and standard test suite



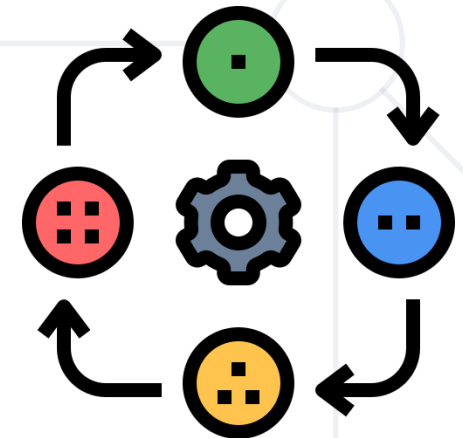
- **Entry Criteria** define the **prerequisite** to be **achieved** before starting the testing activity
- The main focus is to check whether a tester can perform the testing tasks on the software without major obstacles
- Areas to look at while **defining** entry **criteria**
  - Testing **environment** setup and **availability**
  - Availability of all **testing tools**
  - Accessibility of the **testable code**
  - Availability of the **test data**



- **Exit Criteria** define the conditions to be met before testing can be considered as **complete**
- Indicate that the software is up to the **required quality**
- Focus points for exit criteria are
  - Coverage of code, functionality, risk
  - Estimation of defect density or reliability measures
  - Cost or budget
  - Residual risks
  - Schedules like time for marketing



- Test Strategy sets the **overall direction**
- Test Approach offers **specific instructions**
- Entry Criteria establish **predefined conditions that must be met**
- Exit Criteria outline the **conditions that need to be fulfilled**
- Together these four, form the foundation for effective test management **throughout all phases** of the testing process





# Test Planning and Estimation

Strategizing Testing Efforts

# What is Test Planning and Estimation?

- Test planning is the **process of defining the approach, scope, and objectives** of testing activities for a specific project or release
- It involves **creating a comprehensive test strategy and test plan** to guide the testing efforts
- **Importance:**
  - Sets the foundation for **effective and organized testing activities**
  - Ensures clear **understanding** of testing goals and objectives
  - Helps **identify risks**, challenges, and dependencies upfront
  - Facilitates **resource allocation** and **time management**



- **Defining Testing Objectives:**
  - Identify goals and expectations, determine quality attributes to focus on, establish clear criteria for test completion
- **Test Scope and Coverage:**
  - Define boundaries and extent of testing, determine features, functions, and platforms to be tested, identify test environments and configurations
- **Test Strategy:**
  - Select appropriate test techniques, approaches, and levels, decide balance between manual and automated testing, outline test data and test environment requirements

- **Test Schedule and Timeline:**
  - Define testing milestones and deliverables, allocate time for different testing activities, consider dependencies and interdependencies
- **Resource Planning:**
  - Identify and allocate testing resources, determine roles and responsibilities, assess training needs and skill availability
- **Risk Assessment and Mitigation:**
  - Identify potential risks and their impact on testing, develop risk mitigation strategies and contingency plans, prioritize risks based on likelihood and impact

- **Test Documentation:**

- Create and maintain test plans, test cases, and test scripts, document test data, environments, and configurations, prepare guidelines and standards for testing activities

- **Communication and Collaboration:**

- Establish clear communication among stakeholders, coordinate with development and other teams, conduct regular meetings for alignment and issue resolution





# Test Design

From Requirements to Execution

# What is Test Design?

- The **process of creating test cases** and test **scenarios** based on the defined test objectives and requirements
- It involves **translating test conditions** into detailed test cases that validate specific functionalities and system behavior
- **Importance:**
  - Ensures **comprehensive test coverage**
  - Identifies **test conditions** and **test cases to** validate system behavior
  - Helps in **early detection** of defects
  - Enhances the **effectiveness** and **efficiency** of testing
  - Facilitates **traceability between requirements** and test **cases**



- **Reviewing Requirements:**
  - Understand the system requirements, identify functional and non-functional aspects to be tested, clarify any ambiguities or inconsistencies
- **Identifying Test Conditions:**
  - Identify specific test conditions, determine inputs, expected outputs, and system states to be tested, positive and negative scenarios
- **Creating Test Cases and Test Suits:**
  - Develop test cases based on identified test conditions, define steps for each test case, include preconditions and expected results

- **Test Data Preparation:**
  - Create test data to support test cases, consider various data combinations and boundary conditions
- **Test Case Prioritization:**
  - Determine the order of test case execution, prioritize based on risk, business impact, or dependencies, consider critical functionalities or frequently used features
- **Test Case Optimization:**
  - Identify redundant or duplicate test cases, optimize test coverage while maintaining adequate coverage

- **Traceability:**

- Between test cases and requirements, each requirement has corresponding test cases, tracking of test coverage and impact analysis

- **Documentation and Maintenance:**

- Document test cases, test case repositories for future reference, update and revise test cases as necessary



# Test Execution

Driving Quality Forward: Thorough Validation

# What is Test Execution?

- The phase where the planned test cases are executed, and the **actual testing takes place**
- It involves **running the test cases**, **capturing test results**, and comparing the **actual outcomes** with **expected results**
- **Importance:**
  - **Validates** the functionality and behavior of the software or system under test
  - **Identifies defects** and deviations from expected results
  - Enables **real-world simulation** to ensure software readiness
  - **Verifies** that the system meets the **desired quality** standards
  - **Provides feedback** on the overall product stability

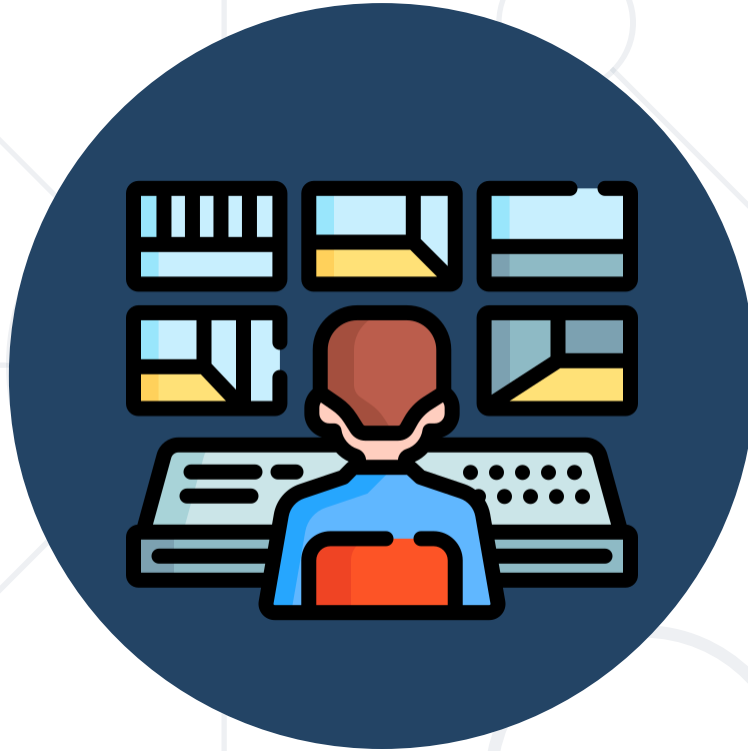


- **Test Environment Setup:**
  - Prepare the necessary test environment and configurations, install and configure the software or system under test
- **Test Case Execution:**
  - Run the test cases, follow the specified test steps and procedures, record the actual results and observations
- **Defect Logging:**
  - Log defects with detailed information, including steps to reproduce, observed behavior, and environment details



- **Defect Prioritization:**
  - Prioritize defects based on their severity, impact on the system, and business priorities
- **Defect Triage:**
  - Conduct defect triage meetings to review and prioritize reported defects, determine the appropriate course of action for each defect
- **Test Progress Monitoring:**
  - Track and monitor the overall progress of test execution, identify any gaps or missed areas, assess the test execution against the planned schedule

- **Regression Testing:**
  - Perform regression tests, address any issues or defects identified during regression testing
- **Test Execution Documentation:**
  - Maintain test execution logs, update test case statuses to reflect their execution status, document any issues
- **Defect Resolution and Verification:**
  - Re-test and verify the fixed defects to ensure they have been resolved satisfactorily
- **Defect Closure:**
  - Close defects that have been successfully resolved and verified, proper documentation



# Test Monitoring and Control

Keeping Testing on Track

# What is Test Monitoring and Control?

- Involves **overseeing** the **progress, quality**, and **effectiveness** of the testing activities throughout the testing lifecycle
- **Importance:**
  - Enables **effective tracking** and **control** of the testing process
  - Facilitates **timely decision-making** and **issue resolution**
  - Ensures **adherence** to **defined objectives** and **quality standards**



# Key Activities in Test Monitoring and Control

- Involves **overseeing and managing** the testing activities throughout the software development lifecycle
- The focus is on **monitoring the progress, quality, and effectiveness** of testing
- Ensures **keeping up** with **defined objectives, timelines, and quality standards**
- Includes **tracking test execution, managing issues and risks, monitoring the test environment, and communicating the testing status**
- Enables **timely decision-making and issue resolution**
- Helps ensure that testing efforts align with project goals and quality standards

\* This topic will be discussed in full in a separate lecture (14. Test Monitoring and Control) , delving into more detailed discussions and best practices



# Test Closure

Wrapping Up the Testing Journey

# What is Test Monitoring and Control?

- **Wrapping up** the testing activities and **documenting** the overall outcomes and lessons learned from the testing effort
- **Importance:**
  - Provides a **comprehensive summary** of the testing effort and its outcomes
  - Facilitates **decision-making** on the software's release readiness
  - Enables lessons learned and **continuous improvement** for future testing projects



# Key Activities in Test Closure

- Test Completion Evaluation
- Test Closure Reporting
- Defect Analysis and Metrics
- Lessons Learned Documentation
- Post-Implementation Review
- Archiving and Documentation

\* This topic will be discussed in full in a separate lecture (14. Test Monitoring and Control) , delving into more detailed discussions and best practices for effective Test Closure





# Test Management Tools

TestRail

- **TestRail** is a **test management tool**
  - Centralized test management
  - Efficient test case organization
  - Effective test execution and tracking
  - Comprehensive reporting and metrics
  - Seamless integration with other tools (e.g., Jira)
  - Customizable to fit unique testing workflows
- TestRail **simplifies testing processes, enhances collaboration, provides valuable insights**, and improves overall **efficiency** and **effectiveness**

- The **Primary Functions** of **TestRail**:
  - Document test cases with steps, expected results, screenshots, and much more
  - Organize test cases into test suites and sections
  - Assign test cases for execution and manage team workloads
  - Track the results of test runs in real-time
  - Review progress toward milestones
  - Generate reports on a variety of metrics
- TestRail supports every type of software testing
- Used to organize manual/script-based testing, schedule and report the results of exploratory testing, and integrate with the test automation tools
- Integrates with defect tracking tools out-of-the-box and includes an open API

# TestRail Example

OVERVIEW

TODOTODO

MILESTONES

TEST RUNS & RESULTS

TEST CASES

REPORTS

R16

Release 1.1: Run 1 (smoke test)

Reports

Rerun

Edit

92 Passed

39% set to Passed

29 Blocked

12% set to Blocked

25 Retest

11% set to Retest

9 Failed

4% set to Failed

39%

passed

80 / 235 untested (34%).

Sort: Section

Filter: None

+ Add Results

Assign To

Columns

Prerequisites

14

ID

Title

Status

T3571

Format table with built-in style

Passed

T3572

Add new review data point (including note)

Retest

T3573

Verify interoperability with system file dialogs

Failed

T3574

Verify title, sub title and heading styles

Blocked

T3575

Verify CSV import with enclosed test data files

Blocked

T3576

Add document footer with page numbers and author

Passed

T3577

Verify formula rendering with built-in print preview driver

Retest

T3578

Test menu keyboard bindings and shortcuts

Blocked

T3579

Verify interoperability with system file dialogs

Passed

T3580

Verify clipboard history with platform independent graphic file

Blocked

T3581

Add document header with document title

Untested

T3582

Add document header with document title

Passed

T3583

Verify interoperability with system file dialogs

Passed

T3584

Change text alignment in an inline table

Passed

T3575

Verify CSV import with enclosed test data files

X

Type

Other

Priority

Medium

Estimate

16 minutes

References

None

Automation Type

None

Expected Result

Etiama massa dolor, ornare sit amet, lacinia nec, bibendum ut, magna.

Nam feugiat, eros at commodo dictum,

Felis libero varius orci, in vulputate

Massa turpis scelerisque diam.

Nunc et felis est. Phasellus laoreet nibh vel augue

Faucibus at varius est pretium.

Quisque pellentesque mauris.

RESULTS & COMMENTS

HISTORY & CONTEXT

DEFECTS

Add a comment ..

Blocked

7/8/2019 8:24 AM

test@test.com

Elapsed

12m

Ac viverra tortor eros sit amet tellus. Massa sodales eget lorem dolor, lorem quam quis eget, tellus vestibulum rutrum:

Aliquam hendrerit mi posuere lectus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum enim wisi, viverra nec, fringilla in, laoreet vitae, risus.

Passed

7/7/2019 1:11 PM

admin

Elapsed

16m

Aliquam hendrerit mi posuere lectus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum enim wisi, viverra nec:

Felis libero varius orci, in vulputate

Massa turpis scelerisque diam.

Nunc et felis est. Phasellus laoreet nibh vel augue

Faucibus at varius est pretium.

Retest

7/6/2019 5:31 AM

Aliquam hendrerit mi posuere lectus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum enim wisi, viverra nec:

ADMINISTRATION

Created by test@test.com. Belongs to milestone Release 1.1.

Tests & Results

Activity

Progress

Defects

All

Prerequisites

Software & Versions

Hardware

Installation

Updates

Tutorial

Goals

Metrics

Login & Account

Reset Password

Feature 1

Feature 2

Feature 3

Feature 4

Feature 5

Administration

Projects

Settings

Users & Roles

Permissions

Groups

Search

Help & Documentation



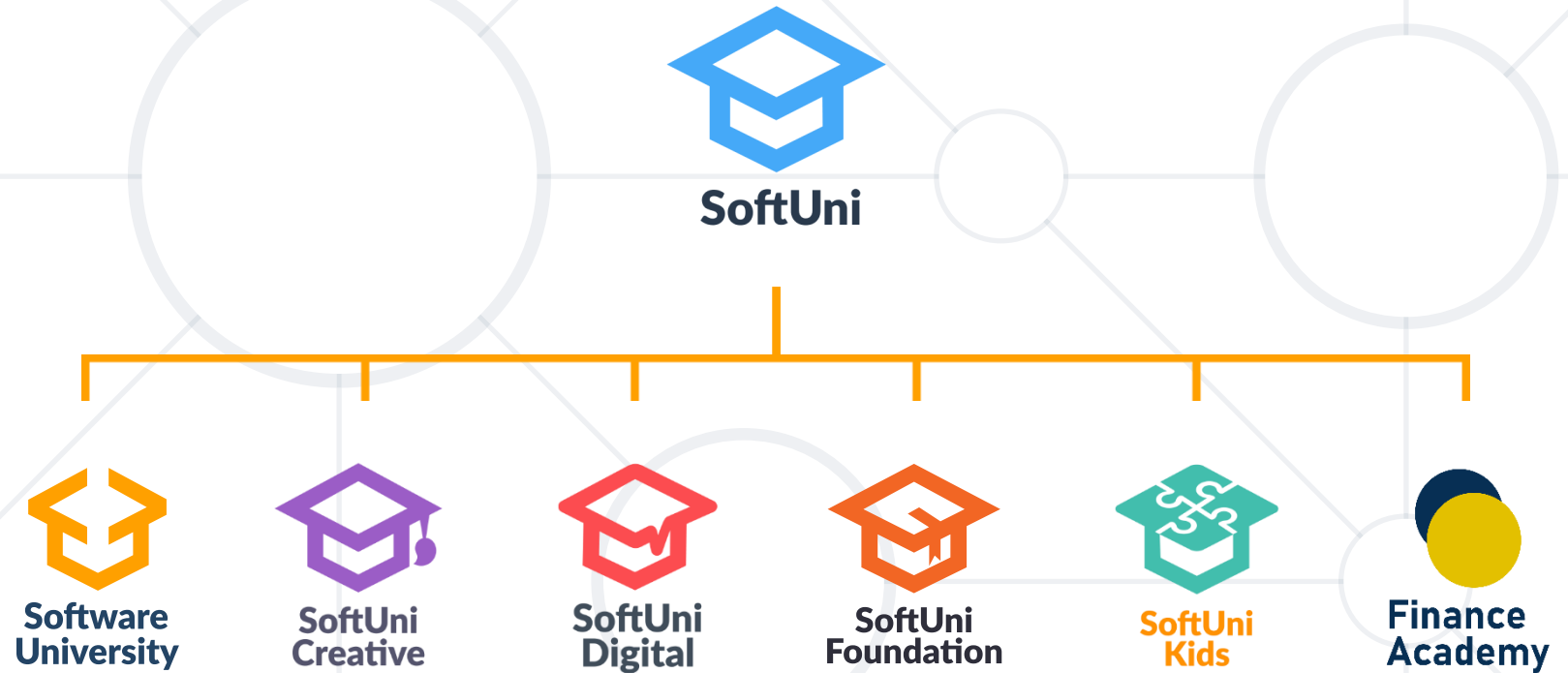
# Test Management Tools

TestRail Demo &  
connect with Jira

- What **Roles** there are in Test Management?  
What are their **responsibilities**?
- Test Strategy, Test Approach, Entry and Exit Criteria – **key components** of Test Management
- Phases in Test Management
  - **Test Planning and Estimation**
  - **Test Design**
  - **Test Execution**
  - **Test Monitoring and Control**
  - **Test Closure**
- Test Management Tools – **TestRail + Jira**



# Questions?



# SoftUni Diamond Partners

**SUPER  
HOSTING  
.BG**



**Coca-Cola HBC  
Bulgaria**



**POKERSTARS**  
POKER | CASINO | SPORTS  
a Flutter International brand

**INDEAVR**  
Serving the high achievers



**AMBITIONED**

 **DRAFT  
KINGS**



**SOFTWARE  
GROUP**

createX



**Postbank**

Решения за твоето утре



**BOSCH**

**DXC**  
TECHNOLOGY



**SmartIT**





- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg/>
- © Software University – <https://softuni.bg>



- Software University – High-Quality Education, Profession and Job for Software Developers

- [softuni.bg](http://softuni.bg), [about.softuni.bg](http://about.softuni.bg)

- Software University Foundation

- [softuni.foundation](http://softuni.foundation)

- Software University @ Facebook

- [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)

- Software University Forums

- [forum.softuni.bg](http://forum.softuni.bg)

