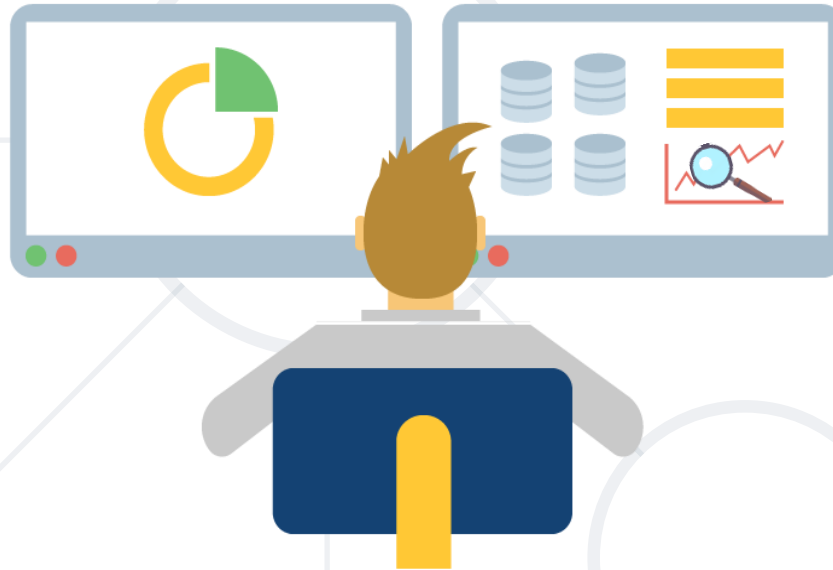


# Test Monitoring and Control



**SoftUni Team**  
**Technical Trainers**



**SoftUni**



**Software University**

<https://about.softuni.bg>

# Table of Contents

1. Test Progress Monitoring
2. Test Metrics and KPIs
3. Monitoring Tools Demo
4. Test Reporting
5. Test Control
6. Risk Management in Testing
7. Test Closure
8. Grafana



# Have a Question?

**sli.do**

**#QA-Fund**



# Test Progress Monitoring

# Test Progress Monitoring

- Testing work needs to be tracked
- **Test** progress **monitoring** is a test **management task**
  - Periodically **monitors the status** of a test project
  - Uses **metrics** for measuring actual **progress** against planned milestones
- Gives **visibility** and **feedback** on test **activities**



# Status of Test Activities

- Defined by a number of **factors**:
  - The level of test plan and **test case completion**
  - The tested object and its **pass**, **fail** and **blocked** data
  - The quantity of **incomplete tests**
  - Number of **open defects**
  - Amount of **retesting** and **regression** testing required





# Test Metrics and KPIs

## Measuring Testing Effectiveness

- **Test metrics**
  - Quantitative measurements
  - Gain insights into testing effectiveness, efficiency, and progress
  - Help to make informed decisions
  - Cover aspects like test progress, coverage, defect metrics, and performance
- **Key Performance Indicators (KPIs)**
  - Critical metrics aligned with testing goals, providing actionable insights
  - Inform project management, risk analysis, and stakeholder communication
- Performed throughout software testing lifecycle **from planning to closure**



# Common Test Metrics (1)

- **Defect density**

- The number of defects identified in a component or system divided by the size of the component or system
- Expressed in **standard measurement terms** such as lines of code, number of classes, or function points
- The formula for calculation of defect density is the **number of defects** divided by **some function points** tested



# Common Test Metrics (2)

- **Failure rate**

- It can be defined as the **ratio of the number of failures** of a given category to a given unit of measure
- For example failures per unit of time, per number of transactions, and per number of computer runs
- Test case failure rate is calculated as some failed test cases divided by the number of test cases executed



# Common Test Metrics (3)

- **Test coverage**

- Covers the extent of coverage achieved against requirements, risks or code
- Higher coverage leads to better quality of testing

Number of **requirements tested**

-----

**Total** number of requirements



# Common Test Metrics (4)

- **Percentage of test case preparation**
  - Helps the manager identify the extent of preparedness for testing

$$\frac{\text{Test cases prepared}}{\text{Total number of planned test cases}}$$



# Common Test Metrics (5)

- Percentage of **test environment preparation**
  - It is a helpful indicator to gauge the preparedness of testing effort

**Test environment** preparation complete

-----

**Total** amount of preparation required



# Common Test Metrics (6)

- **Percentage of test case execution**
  - This is an indicator of the amount of test execution progress achieved

Number of **test cases executed**

-----

**Total number of planned test cases to be executed**



# Common Test Metrics (7)

- **Defect information**

- Defect density, defects found, open and fixed defects are useful metrics for evaluating the software stability and production readiness
- Defect metrics are also used as an indicator to gauge the overall health of the development process

total number of **defects**

-----

total number of **modules**



# Common Test Metrics (8)

- **The confidence level of Testers**

- The confidence level of Testers in the application or product can be captured through surveys or voting

- **Test milestones dates**

- Test milestones dates are set as a part of the test plan
- These need to be monitored to measure any schedule slippages





# Common Test Metrics (9)

- **Testing costs**

- Testing costs include cost compared to the benefit of finding the next defect or to run the next test
- The amount of cost spent in testing including the cost of manpower, environments and associated costs
- Costs should constantly be monitored to ensure that testing does not exceed the budget or to evaluate when to stop testing



# Example: Test Metrics

QA Test Execution as on 11-Jan												
OVERALL Execution Status	Total # of Conditions	% Completed	Test Conditions Executed							% of Test Conditions Deferred	% of Test Conditions On Hold	% of Test Conditions On Failed
			As of Today									
			Planned	On Hold	Executed	Pass	Fail	In Progress	Deferred			
Underwriting	953	98.9%	953	9	923	898	25	1	20	2.1%	0.9%	2.6%
Billing	1438	98.1%	1411	0	1399	1397	2	0	12	0.8%	0.0%	0.1%
Claims	695	93.6%	653	0	611	608	3	0	42	6.0%	0.0%	0.4%
Finance	306	96.1%	296	0	272	250	22	1	23	7.5%	0.0%	7.2%
Printing	2776	76.2%	2379	0	1361	1307	54	27	991	35.7%	0.0%	1.9%
Conversion	671	80.7%	568	18	511	504	7	1	38	5.7%	2.7%	1.0%
Premium rater	1342	79.5%	1067	0	1067	1050	17	0	0	0.0%	0.0%	1.3%
Interfaces	792	84.7%	701	7	555	546	9	2	137	17.3%	0.9%	1.1%
Other Customizations	279	55.6%	197	0	109	93	16	5	83	29.7%	0.0%	5.7%
Underwriting End to End	1222	74.6	918	6	912	816	96	0	0	0.0%	0.5%	7.9%
Total	10474	84.6	9143	40	7720	7469	251	37	1346	13	0	2

- The concept of **KPIs** are **specific metrics** used to gauge the **performance** and **progress** of the testing process against defined objectives and goals
- **Benefits** of Test Metrics and KPIs:
  - Objective measurement of testing progress and effectiveness
  - Early identification of potential issues or bottlenecks
  - Data-driven decision-making for process improvement
  - Improved visibility and communication of testing outcomes to stakeholders



# Most common KPIs





# Test Reporting

# Test Reporting

- **Test reports**
  - Submitted for the testing period at the end of each phase or test project
- Test leaders generate multiple reports during the test planning, design and execution phase
  - They report the progress of test activities
- Reports
  - Keep all **stakeholders informed**
  - **Help** the **Test Leader** get **attention** or **resources** to resolve project risks



# Test Summary Reports

- Test team maintains different reports like **daily**, **weekly**, **monthly** and even **quarterly** status reports
- Test summary report should have **recommendations** and **decisions** for **future actions**, based on the metrics collected
- Include **lessons learned**, which helps in **preventing** repetitive mistakes for future phase





# Example: Test Report

Test Report

Test Cycle

System Test

EXECUTED	PASSED			130	
	FAILED			0	
	(Total) TESTS EXECUTED (PASSED + FAILED)				130
PENDING					0
IN PROGRESS					0
BLOCKED					0
(Sub-Total) TEST PLANNED					130
(PENDING + IN PROGRESS + BLOCKED + TEST EXECUTED)					

Functions	Description	% TCs Executed	% TCs Passed	TCs pending	Priority	Remarks
New Customer	Check new Customer is created	100%	100%	0	High	
Edit Customer	Check Customer can be edited	100%	100%	0	High	
New Account	Check New account is added	100%	100%	0	High	
Edit Account	Check Account is edit	100%	100%	0	High	
Delete Account	Verify Account is delete	100%	100%	0	High	
Delete customer	Verify Customer is Deleted	100%	100%	0	High	
Mini Statement	Verify Ministatement is generated	100%	100%	0	High	
Customized Statement	Check Customized Statement is generated	100%	100%	0	High	





**Test Control**

# Test Control

- **Test control** is a test management task
  - It deals with development and application of a **set of corrective actions**
  - It is done to get a test project on track when monitoring shows a **deviation** from plan
  - Actions may **cover any test activity** and may affect other software life cycle activity or task



# Example: Test Control

An organization usually conducts performance testing on weekday evenings, during off-hours, in the production environment. Due to unanticipated high demand for products, the company has temporarily adopted an evening shift that keeps the production environment in use 18 hours a day, five days a week. This increase in production time reduces the time available for conducting performance testing. This is a risk for the performance testing team. Test control in this case may lead to rescheduling the performance tests to the weekend to ensure zero impact on testing schedule. Regular monitoring of risks and test metrics, therefore helps the project remain on track to meet the test objectives.





# Risk Management in Testing

The process for handling risks

# Risk Management in Testing

- Involves the following tasks, associated with testing activities:
  - **Identifying risks**
  - **Analyzing risks**
  - **Mitigating risks**
- Ensures that **potential issues** are addressed

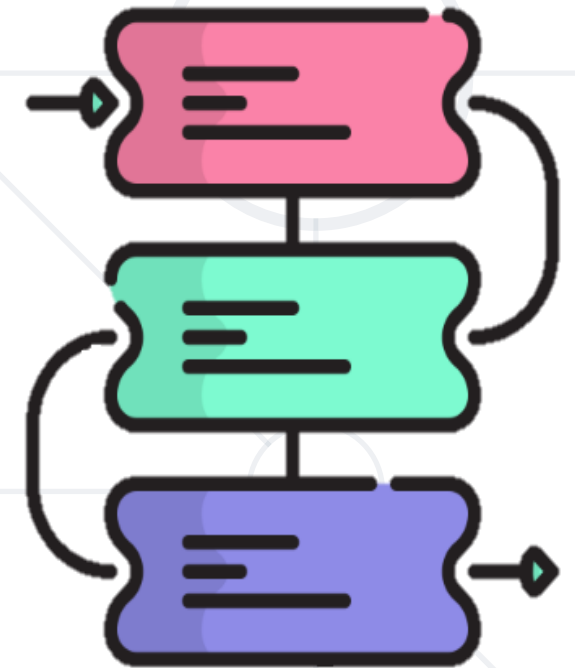


# Risk Identification

- Identifying risks early in the testing process
- Various sources **can introduce risks**
  - Project **complexity**
  - Requirements **volatility**
  - Resource **constraints**
  - Technology **dependencies**
- Significance of **involving stakeholders**, including business analysts, developers, and testers, in the risk identification process



- Explain the **process of analyzing and evaluating** identified risks
- Discuss techniques such as **probability and impact assessment**, risk matrices, and risk categorization
- Emphasize the **importance of prioritizing risks** based on their potential impact on the project and testing objectives





# Test Closure

The formal process of completing testing activities



# Test Closure

- **Involves**
  - Finalizing testing activities
  - Evaluating the test effort
  - Gathering valuable insights for **process improvement**
- Ensures that **objectives are met, risks are mitigated, and quality standards are achieved**



# Objectives of Test Closure

- The primary objectives of test closure include:
  - Ensuring that all **planned test activities** have been **completed**
  - **Collecting and analyzing** test-related data and artifacts for **reporting and decision-making**
  - Assessing the **effectiveness** and **efficiency** of the **testing process**
  - Capturing **lessons learned** to improve future **testing efforts**



# Test Closure Activities

- **Finalizing and documenting** test results - Consolidate and document test outcomes (test case execution status, defects, relevant metrics)
- **Analyzing test coverage** - **Review** test coverage achieved and assess if requirements and risks have been **sufficiently covered**
- **Conducting** a post-mortem review - **Evaluate** the testing process, identify strengths and weaknesses, gather **feedback** from team members through a retrospective meeting
- **Archiving** test assets - **Preserve** test documentation, scripts, data, and artifacts for future reference and audits



# Test Completion Checklist

- **Comprehensive document** that verifies all essential testing activities have been completed before moving forward with the software release
- Typically **includes** the following items:
  - **All Test Cases Executed:** Ensuring that planned test cases have been executed and their results recorded
  - **Defect Resolution:** Verifying that reported defects have been fixed, verified, and closed appropriately
  - **Test Environment Cleanup**
  - **Documentation:** All test-related documentation, such as test plans, test cases, and test reports, are complete and up to date
  - **Test Closure Activities:** Verifying that all Test Closure activities have been performed
- The Test Completion Checklist acts as a **final validation** to ensure that all necessary testing tasks have been addressed and software is ready for deployment



# Checklist Example

Exit Criteria	Status
100% Test Scripts executed	Done
95% pass rate of Test Scripts	
No open Critical and High severity defects	
95% of Medium severity defects have been closed	
All remaining defects are either canceled or documented as Change Requests for a future release	
All expected and actual results are captured and documented with the test script	Done
All test metrics are collected based on reports from HP ALM	
All defects are logged in HP ALM	Done
Test Closure Memo is completed and signed off	



# Test Summary Report

- Formal document - **overview** of the testing effort and its outcomes
- Serves as a **communication tool** for stakeholders. Includes:
  - **Introduction:** An overview of the testing objectives, scope, and timelines
  - **Test Progress:** A summary of test progress
  - **Defect Summary:** An overview of defects found during testing, including their severity, priority, and status
  - **Test Coverage:** Details of test coverage - requirement, code, and risk coverage
  - **Test Metrics:** Key test metrics and KPIs used to evaluate the testing process's effectiveness
  - **Recommendations:** For process improvement or further testing actions
- The Test Summary Report provides stakeholders with a **clear understanding** of the testing effort's results and helps in decision-making regarding the software's release and overall quality





# Grafana

## Live Demo

<https://play.grafana.org/d/0000000012/grafana-play-home?orgId=1>

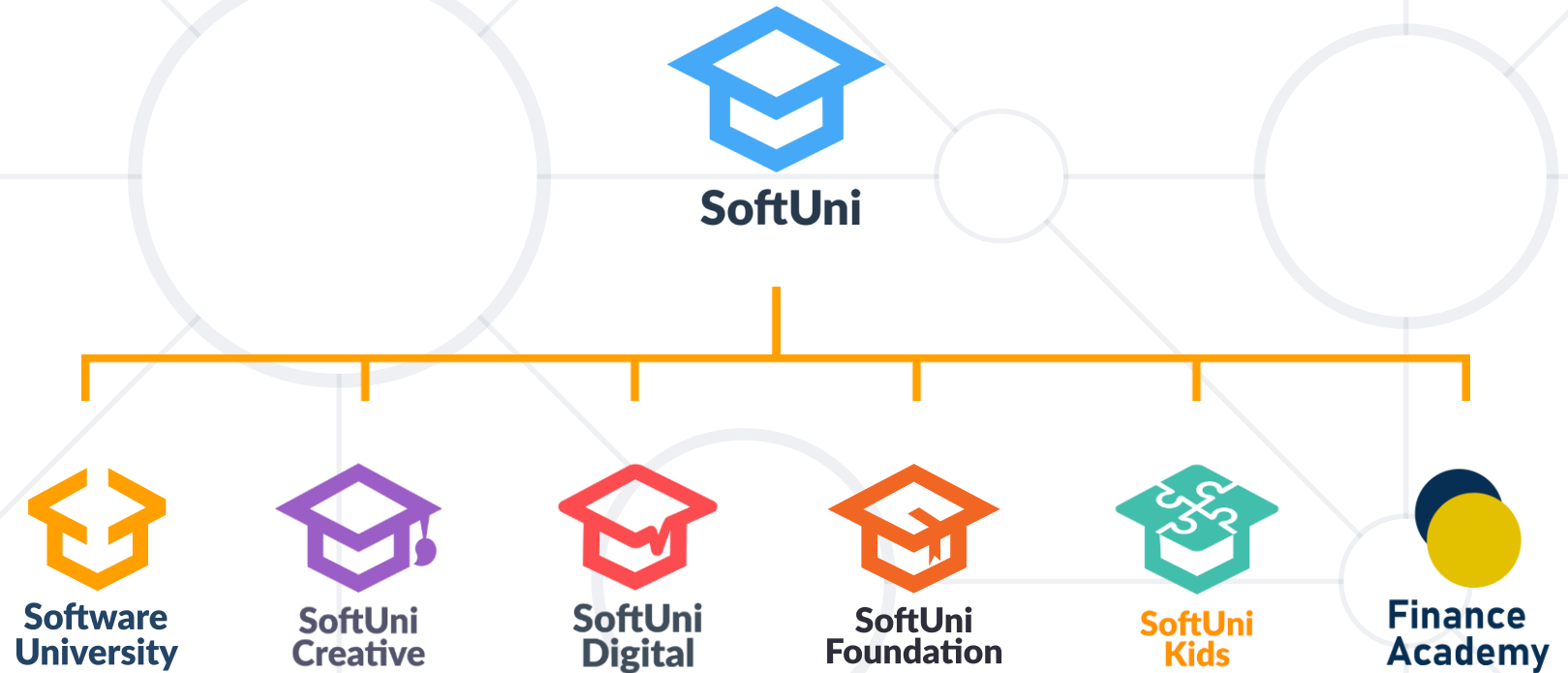
- An open-source **analytics** and **monitoring** platform
- **Interactive dashboards** to monitor and track key test metrics
- Dashboards can display information such as **test progress**, **test coverage**, **defect trends**, and more
- Can be **integrated** with various **data sources**, including testing tools, databases, and other monitoring systems, to collect and visualize test-related data
- Extensive **customization options**, allowing users to tailor the dashboard's appearance and layout to meet specific project requirements
- **Real-time data** updates



- **Test progress monitoring** and **test reporting**
- Understanding **test metrics** and **KPIs**
- How to conduct **control over test process**
- What is **risk management**?
- **Test Closure** – checklist, summary, activities
- **Grafana** – powerful **analytics** and **monitoring** platform



# 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

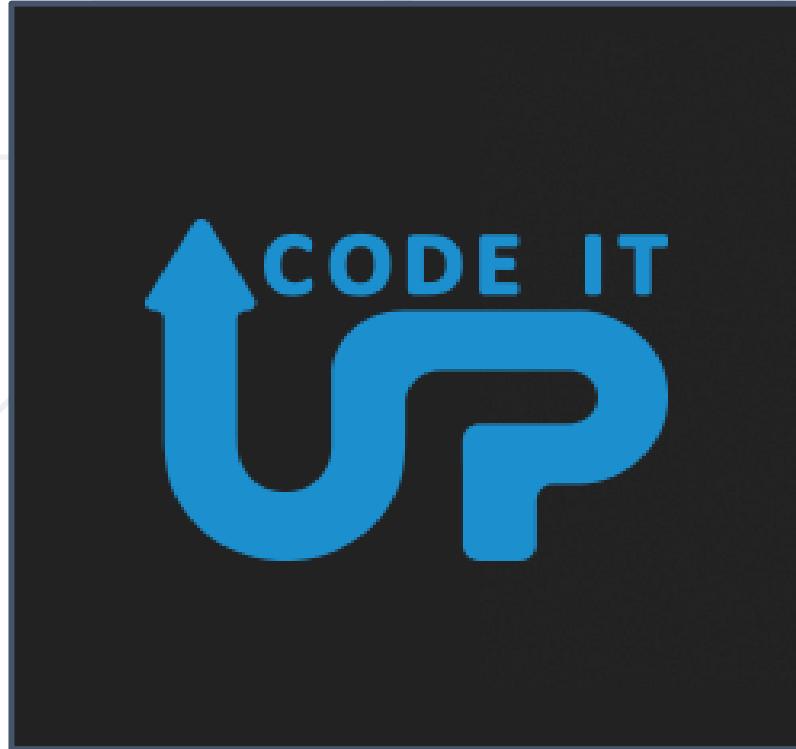


**BOSCH**

**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)

