

Software Testing

Lesson 8 – Test Management

Uwe Gühl Winter 2015 / 2016



Contents



- Test Management
 - Test Organization
 - > Test Organization and Independence
 - > Test Manager and Tester
 - More Testing Roles
 - Test Planning and Estimation
 - Test Planning
 - > Test Planning Activities
 - Entry Criteria
 - ≻Exit Criteria
 - Test Estimation
 - > Test Strategy, Test Approach

Contents



- Test Management
 - Test Progress Monitoring and Control
 - > Test Progress Monitoring
 - > Test Reporting
 - Test Control
 - Configuration Management
 - Risk and Testing
 - Project Risks
 - Product Risks

Improve effectiveness of finding defects by testing and reviews by using independent testers.

- No independent testers; developers test their own code
- Independent testers within the development teams
- Independent test team or group within the organization, reporting to project management or executive management
- Independent testers from the business organization or user community
- Independent test specialists for specific test types such as usability testers, security testers or certification testers (who certify a software product against standards and regulations)
- Independent testers outsourced or external to the organization

<u>}</u>

High



- Recommendation for large, complex or safety critical projects:
 - Multiple levels of testing
 - Independent testers for some or all of the levels
 ⇒ Development staff at lower levels
- Definition of test processes and rules
 - Good idea to be done by independent testers, but a clear management mandate required



- Benefits Independent testers
 - see other and different defects, and are unbiased
 - can verify assumptions people made during specification and implementation of the system
- Drawbacks
 - Isolation from the development team (if treated as totally independent)
 - Developers may lose a sense of responsibility for quality
 - Independent testers may be seen as a bottleneck or blamed for delays in release



- Who should / can do testing tasks?
 - People in a specific testing role
 - Alternatively:
 - Project manager
 - Quality manager
 - > Developer
 - Business and domain expert
 - Infrastructure or IT operations





- Synonyms: Test leader, test coordinator
- Role may be performed by a
 - Project manager
 - Development manager
 - Quality assurance manager
 - Manager of a test group
- Idea: Supporting test team members, so they could do a good job.



Main tasks:

 plans, monitors and controls the testing activities and tasks,
 e.g. as defined in the fundamental test process





Test Manager

- coordinates the test strategy and plan with project managers and others
- writes or reviews a test strategy for the project, and test policy for the organization
- contributes the testing perspective to other project activities, such as integration planning





A helping tool: RACI-Matrix (Responsible, Accountable, Consult, Inform) → To identify and to define roles of people involved in the project (stakeholder)

[Wik16b]

Responsible

Those who do the work to achieve the task

Accountable (also approver or final approving authority)

The one ultimately answerable for the correct and thorough completion of the deliverable or task, and the one who delegates the work to those responsible.

Consulted

Those whose opinions are sought, typically subject matter experts; and with whom there is two-way communication.

Informed

Those who are kept up-to-date on progress, often only on completion of the task or deliverable; and with whom there is just one-way communication.

Winter 2015 / 2016



A helping tool: RACI-Matrix (Responsible, Accountable, Consult, Inform)

Example

Role Task	Customer	Project sponsor	Project manager	Quality manager	Integration manager	Test manager
- Create project plan	С	Α	R	C	С	С
- Create test plan	-	I	С	A	С	R
- Create quality plan	-	Α	С	R	С	С
- Evaluate project result	R	Α	С	C	С	С
- Integrate system components			Α		R	С
 Write project closure report 		Α	R	С	С	С



Test manager

Consider effort for test planning itself – depends on project (size, risk, ...)

- plans the tests
 - selecting test approaches,
 - estimating the time, effort and cost of testing,
 - acquiring resources,
 - defining test levels, cycles, and
 - planning incident management.
- considers context, test objectives and risks.

Winter 2015 / 2016



Test manager

- initiates
 - specification of tests
 - preparation of tests
 - implementation of tests
 - execution of tests
- monitors the test results
- checks the exit criteria



Test manager

- adapts planning based on test results and progress (out of status reports)
 => take action to compensate problems
- sets up configuration management of testware for traceability
- introduces suitable metrics for
 - measuring test progress, and
 - evaluating the quality of the testing and the product.



Test manager

- decides concerning automation
 - what should be automated, to what degree?
 - when and how should automation been done?
- selects tools to support testing and organize any training in tool use for testers
- decides about the implementation of the test environment(s)
- writes test summary reports based on the information gathered during testing

Winter 2015 / 2016

Uwe Gühl - Software Testing 08



- Synonyms: Test Engineer, Test Designer
- Best people should test!
- Software Testers are real experts after end of the tests
 - They know the software:
 Strengths and weaknesses
 - They could support
 - as multiplier
 - for introducing
 - for training



- Qualification
 - Requirements Know-how
 - Modelling
 - UML (Unified Modeling Language) concerning use cases
 - agile methods concerning user stories
 - IT Know-how
 Data modelling
 - Test Know-how
 - Expertise about the subject to be tested



Tester

- reviews and contributes to test plans
- analyses, reviews and assesses user requirements, specifications and models for testability

 \rightarrow Points out faults / open issues

- Prepares and acquires test data
- Creates test specifications
 - creates test cases
 - creates test scenarios
 - combines test data with test cases / test scenarios



Tester

- reviews tests developed by others
- sets up and operates the test environment (often coordinated with system administration and network management).
 => could be a special role

Depending on effort / size of test environment



Tester

- implements tests on all test levels
 - executes and logs the tests
 - evaluates the results
 - documents deviations from expected results
- opens and retests defects after fix
- uses test administration or management tools and test monitoring tools as required
- automates tests (may be supported by a developer or a test automation expert)
- measures performance of components and systems, if applicable

Winter 2015 / 2016

Uwe Gühl - Software Testing 08



- Who could take the role of tester? Example:
 - Component and integration level:
 - Developers
 - Acceptance test level:
 - Business experts
 - ≻ Users
 - Operational acceptance testing:
 - Operators
- Specialization People who work on test analysis, test design, specific test types or test automation may be specialists in these roles

Winter 2015 / 2016



- Test Data Manager
 - Qualification
 - Data base expert (Data modelling know-how)
 - Test Know-how
 - Tasks
 - > Test data strategy / concept
 - Test data research
 - Test data generation
 - Mapping of Test data to Test cases / Test scenarios
 - During Test execution supporting with test data



- Defect Manager
 - Tasks: Choice of tool, defect collection, defect tracking, moderation of defect meetings, control of release management.
- Environment Manager
 - Tasks: Providing Test environment at a time for corresponding tests, accept software, installing it, running smoke test, keep the software "run capable".



- Non functional test manager / Non functional tester
 - Tasks: Defining of a strategy, planning, organizing, execution of performance test, load tests, security tests, breakdown tests
- Security tester
 - Tasks: Security test strategy, execution, consulting
- Test automation expert
 - Tasks: Test automation strategy, choice of tool, preparation and execution (scripting, delivering reports)



- More people could support in testing, especially stakeholder:
 - Customer
 - Requirements Engineers (know specifications best)
 - Users "Old stager" are very valuable, they know processes
 - Operation (Architectural requirements)
 - Software developer



- Important: Testing is not independent coordination with main project required
- Testing is related to software life cycle
 - acquisition
 - supply
 - development
 - operation
 - Maintenance

The goal of test planning is not the test plan but test planning

- Test planning is a continuous activity
- Regular update of test plan required Winter 2015/2016 Uwe Gühl - Software Testing 08



What to consider for Test Planning?



- > You tast botton when you learn about the product
 - You test better, when you learn about the product
 - Not all bugs found in the first cycle will be fixed after the first cycle
 - Not all bugs will be found in the first cycle
 - Side effects not considered

Winter 2015 / 2016

Uwe Gühl - Software Testing 08





- Planning may be documented in
 - master test plan
 - separate test plans for test levels such as
 - > system testing

> acceptance testing

- The outline of a test-planning document is covered by the 'Standard for Software Test Documentation' [IEEE Std 829-1998]
- Expected content: Time Schedule and Resource Plan



- Planning is influenced by
 - the test policy of the organization
 - the scope of testing
 - objectives
 - Risks \rightarrow they change
 - constraints
 - criticality
 - testability

Consider training / training on the job for

- business related know-how
- test tools

- the availability and know-how of resources



Prioritization ... is the basic of testing!

- Why?
 - Time problems
 - Focusing on critical areas
- Which criteria are important for prioritization?
 - Complexity
 - Importance
 - Specification coverage
- How prioritization should be done?
 - Identify most important business processes
 - Identify most important use cases
 - Identify highest risks

Winter 2015 / 2016



Prioritization ... is the basic of testing!





Prioritization ... is the basic of testing!

- Test the important scope first
- Achieve as early as possible a high test coverage
- Detect critical defects as soon as possible in testing critical business processes first
- Minimize the risk of not detected critical defects at the end of testing
- Support the defect fixing in the best way
Test Planning and Estimation Test Planning Activities



- Determining the scope and risks and identifying the objectives of testing
- Defining the overall testing approach including
 - definition of test levels
 - definition of entry and exit criteria

Test Planning and Estimation Test Planning Activities



- Making decisions about
 - what to test
 - what roles will perform the test activities
 - how the test activities should be done
 - how the test results will be evaluated
- Assigning resources for the defined activities.

Test Planning and Estimation Test Planning Activities



- Test documentation Defining the amount, level of detail, structure and templates
- Selecting metrics for monitoring and controlling
 - test preparation
 - test execution
 - defect resolution
 - risk issues
- Setting the level of detail for test procedures in order to provide enough information to support reproducible test preparation and execution

Test Planning and Estimation Entry Criteria



- Entry criteria define when to start testing
- Typically entry criteria:
 - test environment availability and readiness
 - test tool readiness in the test environment
 - testable code availability
 - test data availability

Test Planning and Estimation Exit Criteria



- Exit criteria define when to stop testing
- Typically exit criteria:
 - Thoroughness measures, such as coverage of code, functionality or risk
 - estimates of defect density or reliability measures
 - cost
 - residual risks, such as defects not fixed or lack of test coverage in certain areas
 - schedules such as those based on time to market



Goal: Identifying resources, draw up of a schedule

Approaches for the estimation of test effort

- Metrics-based approach
- Expert-based approach

Test Planning and Estimation Metrics-based approach (1/4)



- Estimating the testing effort based on
 - metrics of former or similar projects or previous cycles
 - typical values / constraints
 - Number of man days available for testing
 - Number of test cases to be executed
 - Complexity of test cases

Test Planning and Estimation Metrics-based approach (2/4)



- Functional Point Analysis [Alb79]
 - Measure of the amount of business functionality
 The higher the number of function points, the more functionality
 - Function points based on functional user requirements of the software, categorized into types: outputs, inquiries, inputs, internal files, and external interfaces
 - After a function is identified and categorized into a type, it is then assessed for complexity and assigned a number of function points

Test Planning and Estimation Metrics-based approach (3/4)

 Functional Point Analysis Function Point Model [Kus07]

EI = External Inputs

- EO = External Outputs
- EQ = External Inquiries
- ILF = Internal Logical Files
- EIF = External Interface Files



Measured application

Test Planning and Estimation Metrics-based approach (4/4)



 Functional Point Analysis Matrix to calculate Unadjusted Function Points

	Complexity weight	Low			Ave	rage		High	ı		
			х			х			х		Total
EI =	External Inputs		3	0		4	0		6	0	0
EO =	External Outputs		4	0		5	0		7	0	0
EQ =	External Inquiries		3	0		4	0		6	0	0
ILF =	Internal Logical Files		7	0		10	0		15	0	0
EIF =	External Interface Files		5	0		7	0		10	0	0
		Total Unadjusted Function Points									0

Test Planning and Estimation Expert-based approach



- Estimating the tasks based on estimates made
 - by owner of the tasks or
 - by experts
- Compare: Planning poker in Scrum [Wik16]



Image source: https://en.wikipedia.org/wiki/File:CrispPlanningPokerDeck.jpg

Uwe Gühl - Software Testing 08



- The testing effort may depend on
 - Characteristics of the product:
 - Quality of the specification and other information used for test models (i.e., the test basis)
 - Size of the product
 - Complexity of the problem domain
 - Requirements for reliability and security
 - Requirements for documentation



- The testing effort may depend on (cont'd):
 - Characteristics of the development process:
 - Stability of the organization
 - ≻ tools used
 - ≻ test process
 - skills of the people involved
 - ≻ time pressure
 - Outcome of testing
 - Number of defects
 - > amount of rework required



- Estimation based on fundamental test process:
 - Personal costs based on main tasks and deliverables during
 - 1. Test planning and control
 - 2. Test analysis and design
 - 3. Test implementation and execution
 - 4. Evaluating exit criteria and reporting
 - 5. Test closure activities
 - Material costs
 - Risk load



• Example:

	Test effort estimation project	t "KU Te	est Projec	t"									
	Test project over 8 weeks. 6 iterations planned with weekly deployments on Mondays from 3rd week on												
	Basic are requirements: User manu	al of old ve	ersion, requi	rements sp	ecification, system a	hes							
	Personal costs	Details				Plai	nning v	values		Real values			
ld	Task	Number	hours per item	intermedi ate hours	Comment	Cost	t / hour	hours	Costs	Cost / hour	hours	Costs	
1	Test planning and control			200		В	200	200	₿ 40.000				
2	Test analysis and design			440		В	150	440	₿ 66.000				
3	Test implementation and execution			224		В	150	224	B 33.600				
4	Evaluating exit criteria and reporting	6	8	48	8 hours / week	B	200	48	B 9.600				
5	Test closure activities			20	workshop / documentation	в	200	20	B 4.000				
	Sum							932	\$153.200			0 0	
										=			
	Material costs					Pla	nning	values		Real values			
ld	Item					Pric	e:	Quantity	Costs	Price	Quantity	Costs	
M1	Test Mgmt Tool incl. 5 licenses								B 20.000				
M2	3 test laptops					В	25.000	3	₿ 75.000				
M3	Load test tool, leasing for 4 weeks					В	1.000	4	в 4.000				
	Sum								\$ 99.000	-		0	
	• ·												
	Dverview					Pla	inning	values		Real values			
	Personal costs								B 153.200			0	
	Material costs								B 99.000			0	
	Risk load							20%	B 50 440				
	Overall result							2070	■ 302.640			0	
	overali result								# 002.040	-		V	



- Test approach
 - Implementation of the test strategy for a specific project
 - includes decisions made based on the (test) project's goal and risk assessment
 - Starting point for
 - planning the test process
 - selecting the test design techniques and test types to be applied
 - defining the entry and exit criteria



- The selected approach depends on the context and may consider
 - risks, hazards and safety
 - available resources and skills
 - the technology
 - the nature of the system (custom built or COTS)
 - test objectives
 - regulations
- Different approaches may be combined, for example, a risk-based dynamic approach

Winter 2015 / 2016



- Typical approaches include:
 - Analytical approaches
 - Risk-based testing where testing is directed to areas of greatest risk
 - Model-based approaches
 - Stochastic testing based on statistical information about
 - * failure rates (such as reliability growth models)
 - x usage (such as operational profiles)



- Typical approaches include (cont'd):
 - Methodical approaches
 - Failure-based (including error guessing and fault attacks)
 - Experience-based
 - Checklist-based
 - >Quality characteristic-based



- Typical approaches include (cont'd):
 - Process- or standard-compliant approaches
 - Industry-specific standards
 - > agile methodologies
 - Dynamic and heuristic approaches
 - ≻ For example:
 - Exploratory testing where testing is more reactive to events than pre-planned, and where execution and evaluation are concurrent tasks



- Typical approaches include (cont'd):
 - Consultative approaches
 Advice and guidance of technology and/or business
 domain experts outside the test team
 - Regression-averse approaches include
 - reuse of existing test material
 - > extensive automation of functional regression tests
 - standard test suites

Test Management Test Execution



- Goal: Deliver all necessary information as basis for decision concerning acceptance of the software
 Software Status Report / Quality Report
- Helping the project to achieve best quality in software
 - ⇒ Based on regulation of acceptance
 - Open defects, that are tolerable / accepted
 - High test coverage \rightarrow all critical areas covered
 - Documentation of status of quality criteria



Test Management Test Execution

Main test execution activities

- Test execution
 - Tests of new delivered features
 - Retests after fixes
 - Regression testing
 - Free testing exploratory testing
 - Specific tests (e.g. security, load and performance)
- Defect management
 - Inform about new defects
 - Discussion of defects (Severity, status)
 - Monitoring

Winter 2015 / 2016

Test Management Test Execution



Main test execution activities

- Regularly update of test suites Test cases, Test scenarios, and test data have to be added, updated, and removed because
 - of changes in the specification (change requests)
 - there are defects and faults in them
 - they were forgotten to create
 - there are more needed for specific areas for example to test more detailed

Test Progress Monitoring



- Test monitoring
 - Provide feedback and visibility about test activities
 - Used to measure exit criteria, such as coverage
 - Collected manually or automatically

Test Progress Monitoring



- Contents of common test metrics (1/2):
 - Percentage of work done in test case preparation (or percentage of planned test cases prepared)
 - Percentage of work done in test environment preparation
 - Test case execution, for example
 - Number of test cases run/not run
 - > Test cases passed/failed



Test Progress Monitoring

- Contents of common test metrics (2/2):
 - Defect information, for example
 - >Open defects (New, open, in work, fixed, retest)
 - Closed defects
 - Defect density
 - ≻ Failure rate
 - Re-test results
 - Test coverage of requirements, risks or code
 - Subjective confidence of testers in the product
 - Dates / Results of test milestones
 - Testing costs

Winter 2015 / 2016

Test Reporting



- Test Reports are the business card of the tester
 - Expected: Periodical statements concerning
 - >quality (of software, specification, test cases)
 - ≻ test progress
 - ≻ test coverage
 - status concerning critical areas, issues, and risks
 - Hint: Discuss reporting criteria in advance with audience – following importance
 - Customer (or principal)
 - Project manager
 - Software developer

- > Project sponsor
- Specification team
- > Operation

Test Reporting



- Test reports summarize
 - what happened during a period of testing, such as dates when exit criteria were met
 - analyzed information and metrics to support recommendations and decisions about future actions
- The outline of a test summary report is given in 'Standard for Software Test Documentation' [IEEE Std 829-1998]

Test Reporting Example (1/2)





Remark: According to our plan (1200 TC) we have executed 966 Test Cases. The gap is approx. 20 %. Reason of less increase in the amount of test execution is mainly the necessary retesting of fixed and delivered defects.

Test Reporting Example (2/2)



- Coverage
 - Delivered SR1 covers
 123 out of 124
 Use Cases
 - 966 of 1200
 Test Cases executed
- Most important statements
 ... Done / Planned
- Risks

Defects

- 303 open defects
 - 32 Severity Level 1,
 - 164 Severity Level 2,
 - 107 Severity Level 3.
- 642 final defect status
 - 60 Change Requests,
 - 427 Closed,
 - 110 Cancelled,
 - 35 Duplicated,
 - 10 Deferred

Test Control



- Test control: Any guiding or corrective actions based on information and metrics; like
 - Decisions based on information from test monitoring
 - Re-prioritizing tests, when an identified risk occurs, e.g. software delivered late
 - Changing the test schedule depending on availability of a test environment
 - Updating entry / exit criteria; e.g. developers have to re-test a fix before bringing to a build and delivering



- Motivation: Why do we need configuration / release management?
 - Expectation: Defect will be detected
 Which version was tested? How was the version built?
 What are the components of the build?
 Deviation to which requirement?
 - The software vendor would like to fix the detected defects \rightarrow We need a new software version
 - Discussion: What, if more than one defect gets fixed?
 How many got fixed? All in next release? Or in next but one?
 - What, if not all defects could be retested successfully?
 - Goal: The last delivered version should be in such high quality so that an acceptance is possible

Winter 2015 / 2016

Uwe Gühl - Software Testing 08



Purpose

Establish and maintain the integrity of the products (components, data and documentation) of the software or system through the project and product life cycle

 Definition: Configuration management are coordinated activities to direct and control configuration [ISO 10007:2003]

> Guideline how to use configuration management within an organization



- Configuration management should ensure for testing purposes:
 - Traceability throughout the test process:
 All items of testware are
 - ≻ identified
 - version controlled
 - tracked for changes
 - related to each other
 - related to development items (test objects)
 - All identified documents and software items are referenced unambiguously in test documentation



- For the tester, configuration management helps to uniquely identify (and to reproduce)
 - the tested item
 - test documents
 - the tests
 - the test harness(es)
- Configuration management procedures and infrastructure (tools) have to be chosen, documented and implemented
 Task during test planning


Release management

... defines the scope and the point in time of software deliveries.

... is the process of managing software releases from development stage to software release [Wik16].

- Release management topics
 - Coordination, when which version / release / patch gets delivered and deployed
 - Scope of a release for planning purposes
 - Release note
 - Description of contents / new functionality
 - Fixed defects
 - Name convention



- Release plan
 - Request: Short installation time
 - Example: Installing new release Friday afternoon, so testing could proceed on Monday
 → Weekend as backup if there are installation problems
 - Goal of release management from testing point of view:
 - Predictable releases to plan test resources optimal. It is costly if testers could not test, because the system is not available



- Smoke test (Synonyms: Sanity / Health check, confidence testing)
 - Typically first activity during / after delivery
 - Simple test as basics before doing "real testing"
 - Often automated
 - Covers e.g. installing procedures, login, basic functionality.
 - If smoke test **pass**:
 - Start testing the new release
 - If smoke test fails:
 - Rollback
 - Proceed testing the old release

Winter 2015 / 2016



- Name convention
 Example
 <Release>.<Version>.<Patch>.<Hot fix>
 - Software version 1.2.0.0 then means
 - Release 1
 - Version 2
 - Patch 0
 - Hot fix 0



• Example for a release documentation / plan

Planned EDC	Release /Patch/ Hot				
Deployment	Fix #	Content			
27.11.2016	1.2.0.0	Release of SR1 Wave 2 - 64 Use Cases			
04/12/2016	1.2.1.0	Patch for Defects resolved in past week			
12.11.2016	1.2.2.0	Patch for 4 Use Cases and 57 Defects resolved in past week			
12.12.2016	1.2.2.1	Hot Fix for Data			
13.12.2016	1.2.2.2	Hot Fix for Defect 596			
18.12.2016	1.2.3.0	Patch with 12 Use Cases, Incremental Test Data, and Defects			
		resolved during past week			
18.12.2016	1.2.3.1	Hot Fix for Client code along with DB scripts			
20.12.2016	1.2.3.2	Hot Fix for smoke test bugs during application matrix testing			
27.12.2016	1.2.4.0	Patch with Defects resolved during past week			
05/01/2017	1.2.5.0	Patch with Defects resolved during Christmas break			
05/01/2017	1.2.5.1	Hot Fix for Severity 1 Defect, if necessary			
13.01.2017	1.2.6.0	Patch for Change Request items - Delegation and LDAP and			
		Defects resolved during past week			
23.01.2017	1.2.7.0	Patch for Defects resolved in past week			
24.01.2017	1.2.7.1	Hot Fix for Severity 1 Defect, if necessary			
30.01.2017	1.2.8.0	Patch for Defects resolved in past week			





- Risk can be defined as the chance of an event, hazard, threat or situation occurring and resulting in undesirable consequences or a potential problem.
- The level of risk will be determined by
 - probability of an adverse event happening
 - impact: The harm resulting from that event
- Task of test manager concerning test related risks
 - Analyzing risks
 - managing risks
 - mitigating risks
- 'Standard for Software Test Documentation' (IEEE Std 829-1998): Test plans require risks and contingencies to be stated.



Risk and Testing Project Risks

- Organizational factors:
 - Skill, training and staff shortages.
 - Personnel issues.
 - Political issues, such as:
 - Problems with testers communicating their needs and test results
 - Failure by the team to follow up on information found in testing and reviews (e.g., not improving development and testing practices)
 - Improper attitude toward or expectations of testing For example: Not appreciating the value of finding defects during testing

Winter 2015 / 2016

Risk and Testing Project Risks



- Technical issues:
 - Problems in defining the right requirements
 - The extent to which requirements cannot be met given existing constraints
 - Test environment not ready on time
 - Late data conversion, migration planning and development and testing data conversion / migration tools
 - Low quality of the design, code, configuration data, test data, and tests



Risk and Testing Project Risks

- Supplier issues:
 - Failure of a third party.
 - Contractual issues.

Risk and Testing Product Risks



- Product risks
 ⇒ Risk to the quality of the product
 Potential failure areas in the software or system
- Risks are used to decide where to start testing and where to test more
- Risk-based approach to testing to reduce
 - the risk of an adverse effect occurring
 - the impact of an adverse effect

Risk and Testing Product Risks



- Examples:
 - Failure-prone software delivered
 - The potential that the software / hardware could cause harm to an individual or company
 - Software that does not perform its intended functions
 - Poor software characteristics
 Functionality, reliability, usability and performance
 - Poor data integrity and quality
 Data migration issues, data conversion problems, data transport problems, violation of data standards

Risk and Testing Product Risks



- In a risk-based approach the risks identified may be used to
 - determine the test techniques to be employed
 - determine the extent of testing to be carried out
 - prioritize testing in an attempt to find the critical defects as early as possible
 - determine whether any non-testing activities could be employed to reduce risk Example: Training for inexperienced designers

Risk and Testing Example



	Risk Description							cation		
ld 👻	Risk Identification	Potential Cause 🖕	Contact persor 🔻	Along w	Date 💂	₽ _.	-	Risl.	Statu-	Actions
R001	Example of a Risk Number 1 with low probability, but possible critical damage	Source 1	Uwe		01.02.12	1	3	3	in progress	2012-02-02 [Uwe] informed [Arnon]
R002	Example of a Risk Number 2 with high probability	Source 2	Arnon		01.02.12	3	2	6	done	2012-02-02 [Arnon] did some activities

Risk Index = P x T

- P = Probability of incidence:
- 3 = high
- 2 = possible
- 1 = low

- T = Estimated damage:
- 3 = very critical
- 2 = critical
- 1 = less critical

Sources



- [Alb79] A. J. Albrecht: "Measuring Application Development Productivity", Proceedings of the Joint SHARE, GUIDE, and IBM Application Development Symposium, Monterey, California, October 14–17, IBM Corporation 1979
- [IEEE Std 829-1998] IEEE Std 829™ IEEE Standard for Software Test Documentation, 1998
- [ISO 10007:2003] ISO 10007:2003 Quality management systems -- Guidelines for configuration management, 2003
- [ISTQB-CTFLS11] International Software Testing Qualifications Board: Certified Tester Foundation Level Syllabus, Released Version 2011, http://www.istqb.org/downloads/syllabi/foundation-levelsyllabus.html
- [ISTQB-GWP15] Glossary Working Party of International Software Testing Qualifications Board: Standard glossary of terms used in Software Testing, Version 3.01, 2015, http://www.istqb.org/downloads/glossary.html
- [Kus07] S. Kushal: "Calculating Function Points", 2007, http://www.codeproject.com/Articles/18024/Calculating-Function-Points
- [Wik16] wikipedia.org: Release Management; 2016; https://en.wikipedia.org/wiki/Release_management
- [Wik16a] Wikipedia.org: Planning poker, 2016, https://en.wikipedia.org/wiki/Planning_poker
- [Wik16b] Wikipedia.org: Responsibility assignment matrix, 2016, https://en.wikipedia.org/wiki/Responsibility_assignment_matrix,