Software Testing

Lesson 9
Test Management – Test Execution
V1.1

Uwe Gühl

Winter 2013 / 2014

Contents

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

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 → all critical areas covered
  - Documentation of status of quality criteria

Test Management
Test Execution

Main test execution activities

- Test execution
  - Regularly tests of new delivered features,
  - Retesting 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.

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.

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.
  - Hint: Discuss reporting criteria **in advance** with audience – following importance
    - customer,
    - project manager,
    - software developer,
    - project sponsor,
    - specification team,
    - operation.

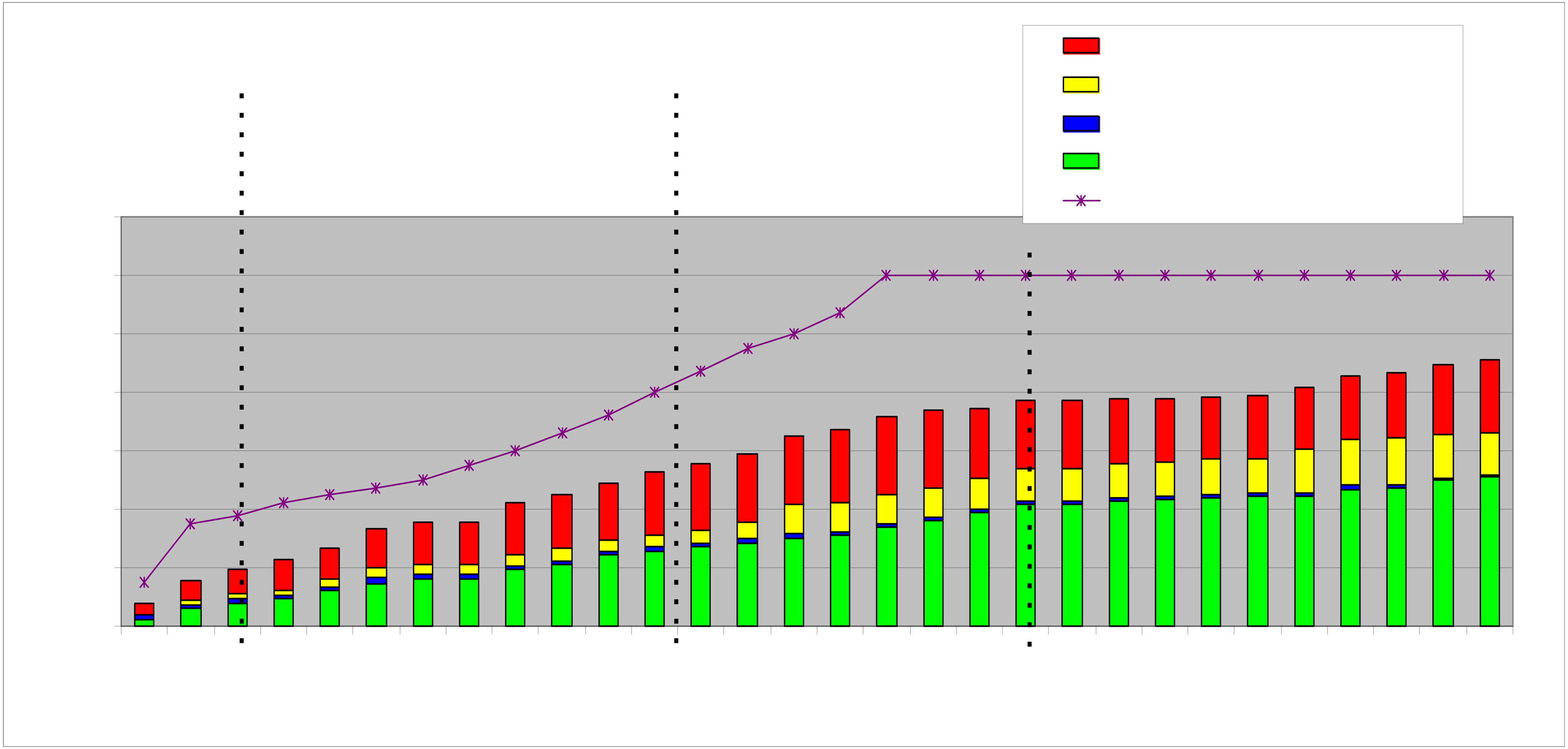
Test Reporting

- Test reporting is concerned with summarizing information about the testing endeavour, e.g.
  - 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, such as
    - an assessment of defects remaining,
    - the economic benefit of continued testing,
    - outstanding risks, and
    - the level of confidence in the tested software.

Test Reporting

- The outline of a test summary report is given in ‘Standard for Software Test Documentation’ [IEEE Std 829-1998].
- Metrics should be collected during and at the end of a test level in order to assess:
  - The adequacy of the test objectives for that test level,
  - the adequacy of the test approaches taken,
  - the effectiveness of the testing with respect to the objectives.

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 taken as a result of information and metrics gathered and reported.
- Actions may
  - cover any test activity and
  - affect any other software life cycle activity or task.

Test Control

- Examples of test control actions include:
  - Making decisions
    based on information from test monitoring.
  - Re-prioritizing tests
    when an identified risk occurs, e.g. software delivered late.
  - Changing the test schedule
    due to availability or unavailability of a test environment.
  - Setting an entry criterion
    For example: Developers have to re-test a fix before bringing to a build and delivering.

Configuration Management

- 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
    → 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.

Configuration Management

- 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

- 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, and
    - related to development items (test objects).
  - All identified documents and software items are referenced unambiguously in test documentation

Configuration Management

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

Release Management

- 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 [Wik14].
- 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 Management

- 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.

Release Management

- Smoke test
  - 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.

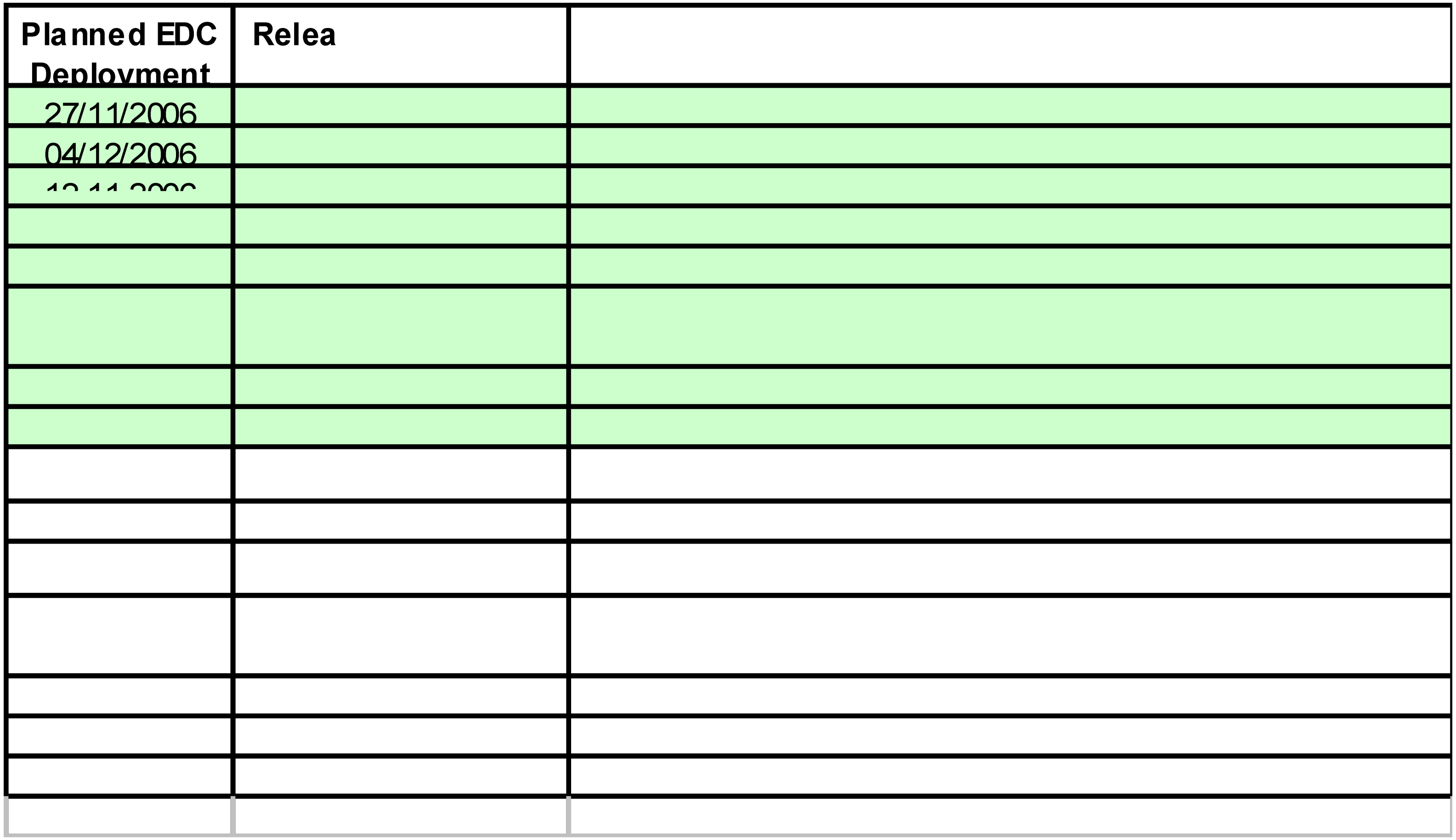
Release Management

- 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.

Release Management

- Example for a release documentation / plan



Planned EDC Deployment	Relea	
27/11/2006		
04/12/2006		
[illegible]		

Risk and Testing

- 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 and
  - impact
The harm resulting from that event.

Risk and Testing
Project Risks

- Project risks are the risks that surround the project's capability to deliver its objectives, like
  - organizational factors,
  - technical issues,
  - supplier issues.

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.

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
Project Risks

- Task of test manager:
  - Analyzing risks,
  - managing risks, and
  - mitigating risks.
- ‘Standard for Software Test Documentation’ (IEEE Std 829-1998):
  Test plans require risks and contingencies to be stated.

Risk and Testing
Product Risks

- Potential failure areas (adverse future events or hazards) in the software or system are known as product risks
  ⇨ Risk to the quality of the product.
- Product risks are a special type of risk to the success of a project.

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

- Risks are used to decide where to start testing and where to test more
- Testing is used to reduce
  - the risk of an adverse effect occurring, or
  - the impact of an adverse effect.

Risk and Testing
Product Risks

- Risk-based approach to testing
  … provides proactive opportunities to reduce the levels of product risk, starting in the initial stages of a project.

  … involves the identification of product risks and their use in guiding

  - test planning and control,
  - specification of tests,
  - preparation of tests, and
  - execution of tests.

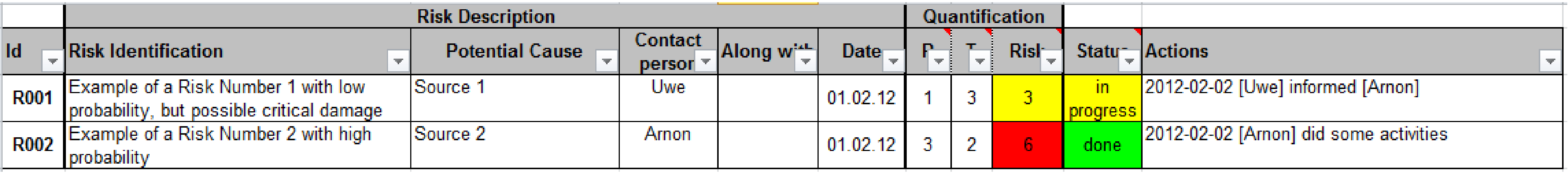
Risk and Testing
Product Risks

- 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.
    For example: Training for inexperienced designers.

Risk and Testing
Product Risks

- Risk management activities to minimize chance of a product failure
  - Assess what could go wrong
    ==> Reassess on a regular basis
  - Determine what risks are important to deal with.
  - Implement actions to deal with those risks.
- In addition, testing may
  - support the identification of new risks,
  - help to determine what risks should be reduced,
  - lower uncertainty about risks.

Risk and Testing Example



	Risk Description					Quantification				
Id	Risk Identification	Potential Cause	Contact person	Along with	Date	P	T	Risk	Status	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:	T = Estimated damage:
3 = high	3 = very critical
2 = possible	2 = critical
1 = low	1 = less critical

Sources

- [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-GWP12] Glossary Working Party of International Software Testing Qualifications Board: Standard glossary of terms used in Software Testing, Version 2.2, 2012, http://www.istqb.org/downloads/glossary.html
- [Wik14] wikipedia.org: Release Management; 2014; http://en.wikipedia.org/wiki/Release_management