IT Quality and Software Test

Lesson 8
Test Management – Test Execution
V1.0

Uwe Gühl

Winter 2011/ 2012

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 informations 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
  - High test coverage → all critical areas covered
  - Open defects, that are tolerable / accepted
  - 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 (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, e.g. to test more detailed

Test Progress Monitoring

- Test monitoring
  - Provide feedback and visibility about test activities.
  - Information to be monitored may be collected manually or automatically
  - used to measure exit criteria, such as coverage
- Using metrics to assess progress against the planned schedule and budget.

Test Progress Monitoring

- Common test metrics include:
  - Percentage of work done in test case preparation (or percentage of planned test cases prepared)
  - Percentage of work done in test environment preparation

Test Progress Monitoring

- Common test metrics include (cont'd):
  - Test case execution, for example
    - Number of test cases run/not run,
    - Test cases passed/failed.
  - 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

Test Progress Monitoring

- Common test metrics include (cont'd):
  - 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
    - Customer
    - Software Developer
    - Specification Team
    - Operation

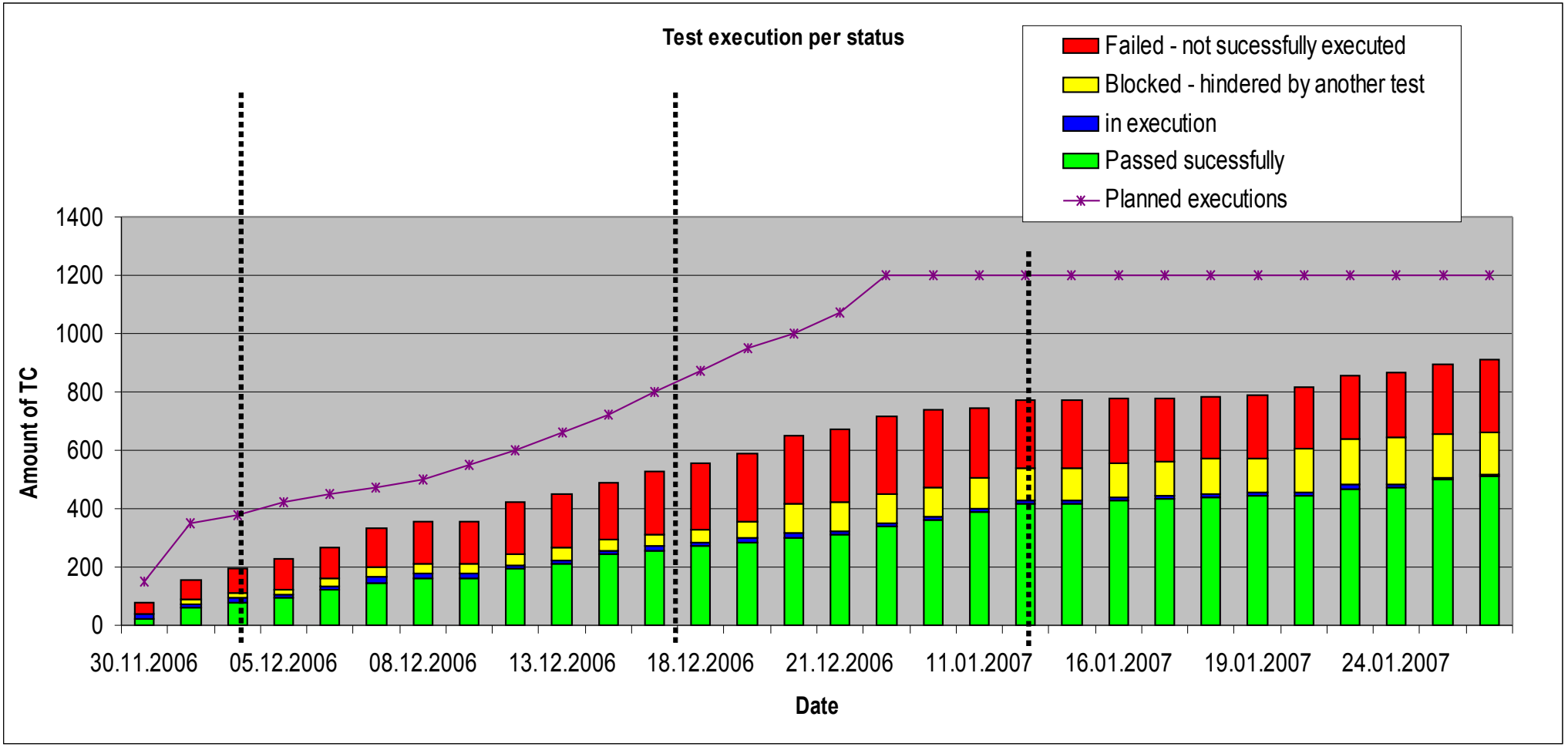
Test Reporting

- 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

  ...
- 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.
- ISO 10007:2003 defines configuration management as coordinated activities to direct and control configuration.

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
- Release Management
  - Agreement, when which version / release / patch gets delivered
  - Scope of a release
  - Release note
    - Description of contents / new functionality
    - Fixed defects
  - Name convention

Release Management

- Release plan
  - Request: Short installation time
  - Proposal: Installing new release Friday afternoon, so Testing could proceed on Monday
    → Weekend as backup if there are installation problems
  - Goal: Predictable release 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 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**: 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 plan

Planned EDC Deployment	Release /Patch/ Hot Fix #	Content
27/11/2006	1.2.0.0	Release of SR1 Wave 2 - 64 Use Cases
04/12/2006	1.2.1.0	Patch for Defects resolved in past week
12.11.2006	1.2.2.0	Patch for 4 Use Cases and 57 Defects resolved in past week
12.12.2006	1.2.2.1	Hot Fix for Data
13/12/2006	1.2.2.2	Hot Fix for Defect 596
18/12/2006	1.2.3.0	Patch with 12 Use Cases, Incremental Test Data, and Defects resolved during past week
18/12/2006	1.2.3.1	Hot Fix for Client code along with DB scripts
20/12/2006	1.2.3.2	Hot Fix for smoke test bugs during application matrix testing
27/12/2006	1.2.4.0	Patch with Defects resolved during past week
05/01/2007	1.2.5.0	Patch with Defects resolved during Christmas break
05/01/2007	1.2.5.1	Hot Fix for Severity 1 Defect, if necessary
13/01/2007	1.2.6.0	Patch for Change Request items - Delegation and LDAP and Defects resolved during past week
23/01/2007	1.2.7.0	Patch for Defects resolved in past week
24/01/2007	1.2.7.1	Hot Fix for Severity 1 Defect, if necessary
30/01/2007	1.2.8.0	Patch for Defects resolved in past week

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 the likelihood of an adverse event happening and the 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 (e.g., 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 concerning risks (project management principle)
  - analyzing,
  - managing and
  - Mitigating.
- ‘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.
  - Poor software characteristics (e.g., functionality, reliability, usability and performance).
  - Poor data integrity and quality (e.g., data migration issues, data conversion problems, data transport problems, violation of data standards).
  - Software that does not perform its intended functions.

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

- A risk-based approach to testing provides proactive opportunities to reduce the levels of product risk, starting in the initial stages of a project.
- It 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 riskbased 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 (e.g., providing training to inexperienced designers)

Risk and Testing
Product Risks

- To ensure that the chance of a product failure is minimized, risk management activities provide a disciplined approach to:
  - Assess (and reassess on a regular basis) what can go wrong (risks).
  - Determine what risks are important to deal with.
  - Implement actions to deal with those risks.

Risk and Testing
Product 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:

3 = high

2 = possible

1 = low

T = Estimated damage:

3 = very critical

2 = critical

1 = less critical

Sources

- International Software Testing Qualifications Board: Certified Tester Foundation Level Syllabus, Released Version 2011, http://istqb.org/display/ISTQB/Foundation+Level+Documents