Software Test

Lesson 15
Test Completion – Outlook
v1.1

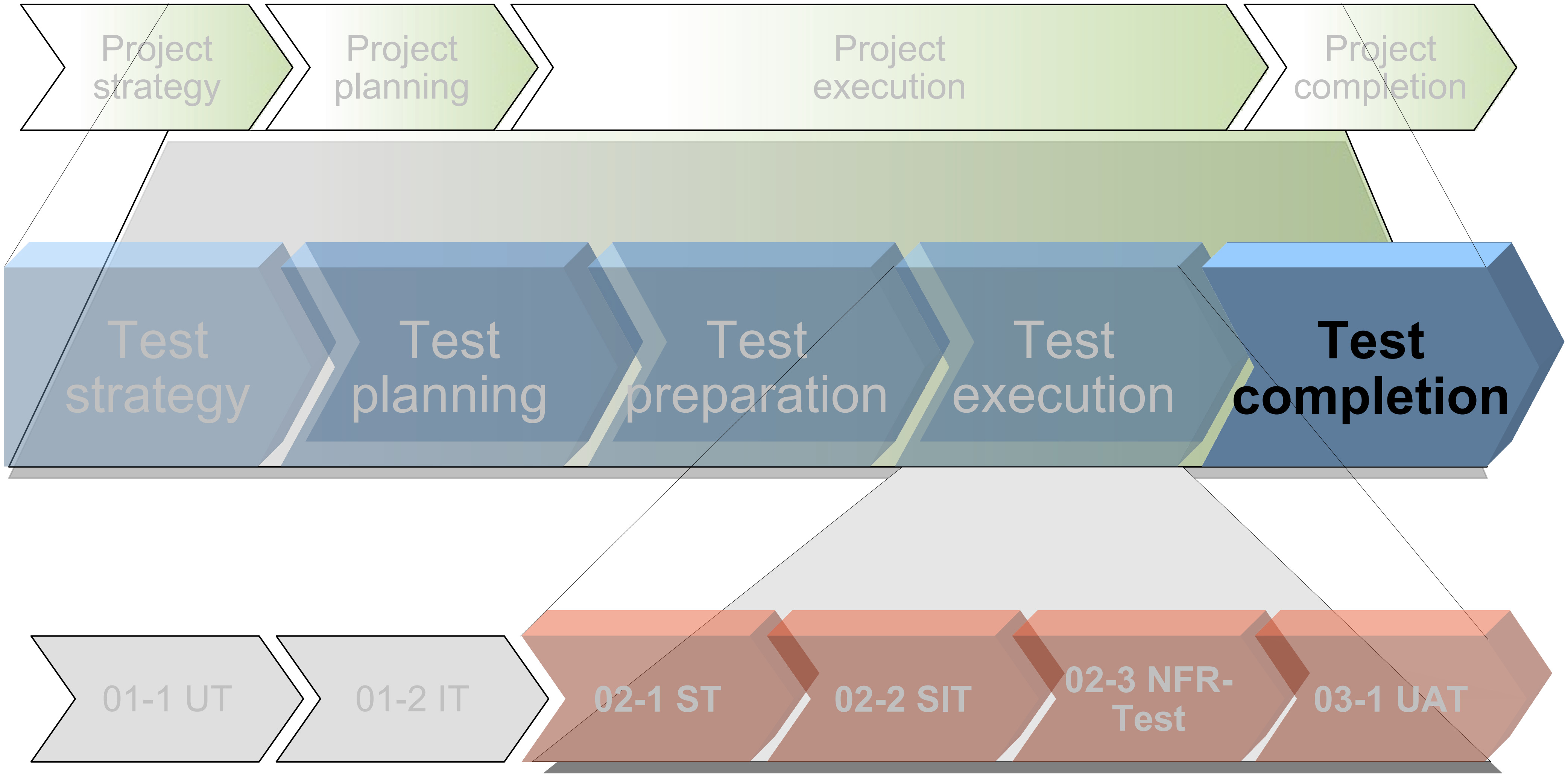
Uwe Gühl, Jittat Fakcharoenphol

Fall 2007/ 2008

Contents

- Test Completion
  - Goal
  - Documentation
  - Formal acceptance of software
  - Lessons learned
  - Finalization
- Outlook
  - Trends in Testing
  - Get Tester!

A sample testing cycle
Test Completion



Test Completion Goal

- Documentation – Final report
- Acceptance of Software
- Lessons Learned Workshop
- Finalization of Test Project
- Assuring results

Test Completion Documentation

- The final report
  - should give a complete big picture of the test
  - is based on the weekly test reports
  - should be well arranged, so that know-how could be reused in the future

Test Completion Documentation

- The final report – main contents (proposal)
  - Goal: Did we achieve our goal?
  - Test management
    - Organizational – org chart, assignment of responsibilities
    - Complete description of the test from start to end
    - Important decisions and reasons for them
    - Test progress – what has been tested and how?
    - Comparison target / actual
    - Resources needed compared to results
    - Statements: Confidence in quality of testing

Test Completion Documentation

- The final report – main contents (proposal)
  - Test Preparation
    - Results: Test Cases, Test Scenarios, Test Data, test environment ...
  - Test Execution
    - Test coverage
      How much of which areas has been tested?
    - Defect management – description of process
    - Defects and their status
      How many defects found? Final status: open / closed defects, severity / priority level
    - Software quality statements, esp. NFR (performance, security)

Test Completion Documentation

- The final report – main contents (proposal)
  - Summary, outlook
    - What worked out? Where are improvements necessary?
    - Suggestion for improvements for future test projects

Test Completion Formal acceptance of software

- Results to customer
  - Software delivery
  - Final testing report
    - Testing work order
    - Management Summary
    - Testing results
      - Defects
      - Test coverage
      - Software quality statements
    - Testing process
      - Did we reach the goal? Plan / Target – Results
      - Spent effort (Time, resources)

Test Completion
Formal acceptance of software

- Discharge of the Testing Team
  - Project leader accepts the results of the Testing Team
  - Project leader accepts the delivered software
    - not
    - partially
    - complete

Test Completion
Formal acceptance of software

- Based on the final test report the acceptance takes place
  - Complete acceptance
    - Ideal: Everything okay
    - Additionally: Future collaboration, software extension, ...
  - Partly acceptance
    - Agreement on rectification of defects with time schedule
    - Shortage of payment
  - Refusal of acceptance
    - Software does not fulfill any requirement, is crashing, ...
    - Possibly conflict management necessary

Test Completion Lessons Learned

- Lessons Learned Workshop
  - Goal:
    - Making things better in the future!
    - Share experiences
    - Mutual benefit for all participants

Test Completion Lessons Learned

- Lessons Learned Workshop
  - Proceeding (Proposal)
    - Feedback
    - Collecting of problems
    - Structuring found problems
    - Establishing small groups to develop proposals
    - Discussion, collection of tasks to do
    - Assignment, responsibilities and time schedules of tasks

Test Completion Finalization

- Documentation should be collected and shared
- The Testing gets stopped
- The Testing Team, boards get demobilized
- Closing of accounts
- Transition to operation
  - Definition of new processes e. g. concerning detected defects
- Party

Outlook
Trends in Testing

- Early Testing
  - especially in agile processes
- Test Case Creation parallel to creation of specification
  - Cost benefit analysis necessary
  - Example: All Test Cases have to be updated, if the basic requirement out of the specification is changing

Outlook
Trends in Testing

- Test Driven Design
- Continuation: “Behavorial Driven Testing” or Behavorial Driven Design
  see
  http://en.wikipedia.org/wiki/Behavior_driven_development

Outlook
Trends in Testing

- Behavioral Driven Design
  Result of a what a test produces* (successfully)

```
A grader engine
- should grade normal submission
- should just return nil when there is no submission
- should produce error message when submission cannot compile
- should produce timeout error when submission runs forever
- should produce timeout error correctly when submission runs
  slower
than expected in less than a second


Finished in 12.779491 seconds


5 examples, 0 failures
```

looks like a spec!

* this and following pages stolen out of an email from Jittat from 4th March 2008

Outlook
Trends in Testing

- Behavioral Driven Design (BDD)
  - Unit test is great. You test your unit thoroughly before you integrate. But what to test?
  - For each spec, you should have at least one test case for it. But this is not visible under the standard unit testing framework.
  - What behavioral-driven testing style is doing is to encourage you to think that way: exactly that way; and make it explicit.

Outlook
Trends in Testing

- Behavioral Driven Design (BDD)
  - When you write your test, you have to name each test so that it's readable, when the test runs, it produces a readable document showing which part of the spec you have covered.
  - Yes, you can go with unit test, but BDD gives explicit feedback, and to me it really encourages good test design (as we may like).
  - There's a lot more with it I'm sure, but that should give you the idea.

Outlook
Trends in Testing

- Behavioral Driven Design (BDD) – Tools
  - see http://behaviour-driven.org/Implementations Extract:
    - for Java: JBehave, JDave, beanSpec, Instinct
    - for Ruby: Rspec
    - for C++: CppSpec
    - for Python: Specipy, spec plugin for nose

Outlook
Get Tester!

- Professional possibility: Get professional tester!
- Get your qualification, e. g.:
  - QAI (Quality Assurance Institute Worldwide, USA) [QAI08a]
    - Professional Certifications
      - CSTE – Certified Software Tester / Test Engineer
      - CSQA – Certified Software Quality Assurance
      - CSPM – Certified Software Project Manager
    - Advanced Certifications
      - CMST – Certified Manager of Software Testing
      - CMSQ – Certified Manager of Software Quality

Outlook
Get Tester!

- Get your qualification, e. g.:
  - ISTQB (International Software Testing Qualification Board) [IST07]
    - ISTQB®-Certified Tester Foundation Level
    - ISTQB®-Certified-Tester Advanced Level
      - Module Test Manager
      - Module Functional Tester
      - Module Technical Tester
      - Full Advanced Level (after passing the above modules)
    - ISTQB®-Certified-Tester Expert Level (in preparation)

Outlook
Get Tester!

- Get your qualification, e. g.:
  - Information Systems Examinations Board (ISEB) – part of British Computer Society (BCS) [BCS07]
    - Foundation Certificate in Software Testing
      - Good overview of the basics of software testing.
    - Intermediate Certificate in Software Testing
      - Next level of knowledge and practical expertise, it also covers the key topics which are needed for the practitioner level exams
    - Practitioner Certificates in Test Management
      - High level qualification which examines knowledge and skills required for the management of Software Testing
    - Practitioner Certificate in Test Analysis
      - High level qualification which examines knowledge and skills required for the technical analysis of Software Testing.