Software Test

Lesson 5
Test Strategy
v1.0b

Uwe Gühl, Jittat Fakcharoenphol

Fall 2007/ 2008

Contents

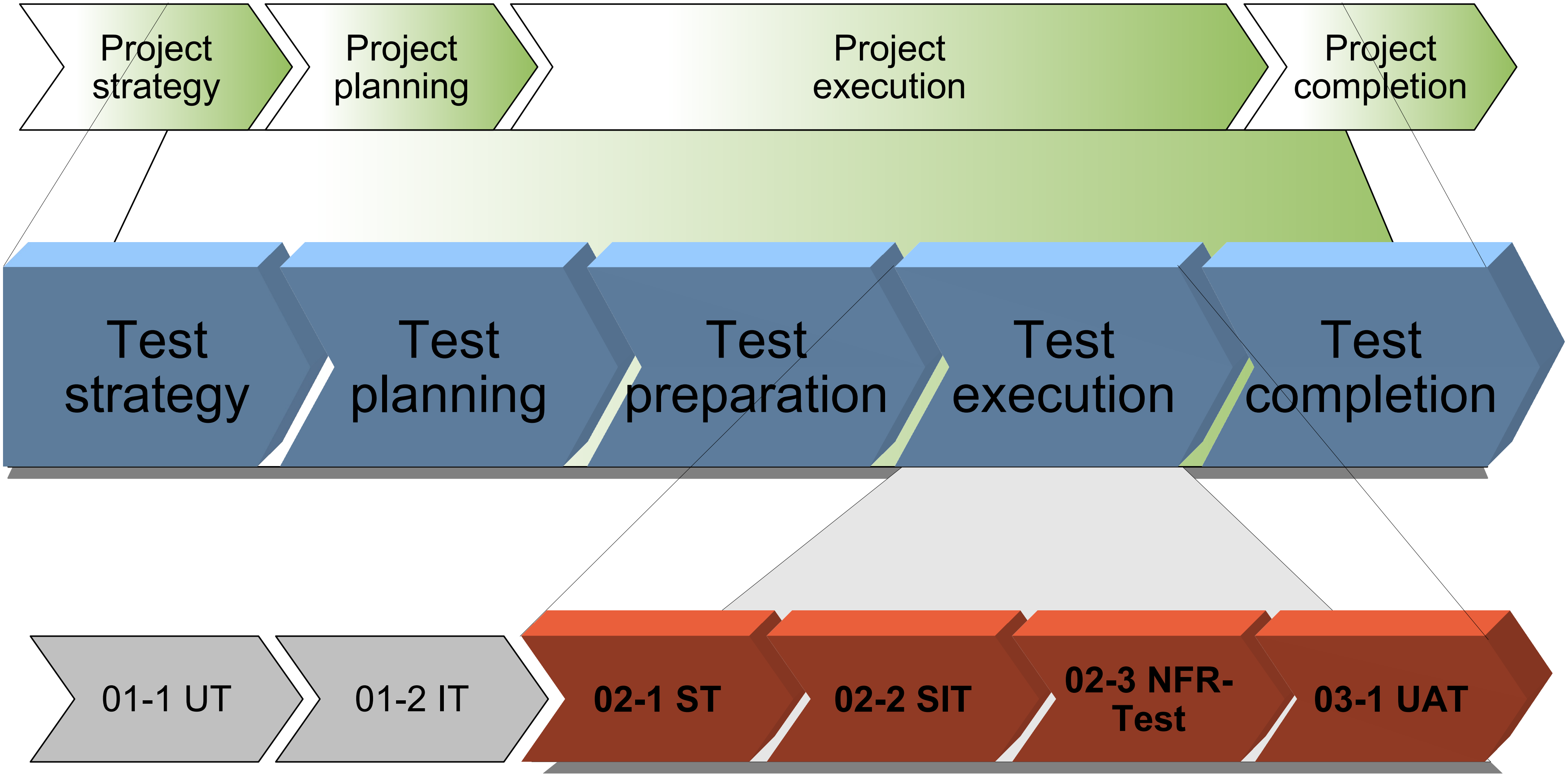
- A sample Testing Cycle
- Test Strategy during Software-Development
  - Introduction
  - Checklist

Hint: Focus of this lecture is Software Testing form the customers point of view, that's why only some words to Test Strategy in Software Development

Contents

- Test Strategy
  - Introduction
  - Philosophy
  - Test as a project in the project
  - Factor time
  - Goal
    - General
    - Contents
    - Summary
  - Links

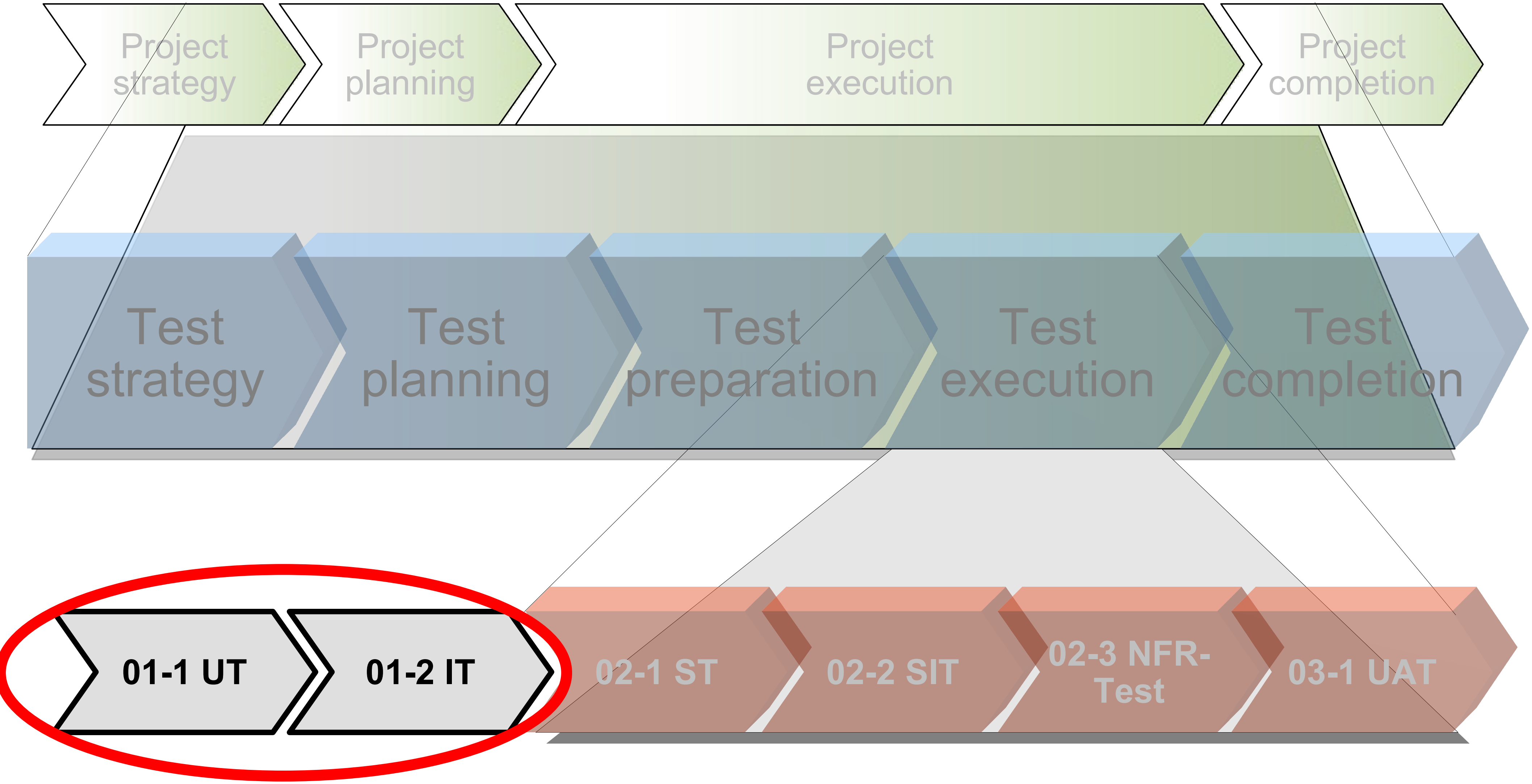
A sample Testing Cycle



A sample Testing Cycle

- In the lectures we follow now an ideal test process to give you (hopefully good) ideas how to organize testing in the best way
- Of course there are many influences in “real test project life” which hinders this ideal model
- But in my opinion only if you know and describe what you would like to achieve you can succeed...

A sample Testing Cycle
Test Strategy during SW-Dev.



Project strategy

Project planning

Project execution

Project completion

Test strategy

Test planning

Test preparation

Test execution

Test completion

01-1 UT

01-2 IT

02-1 ST

02-2 SIT

02-3 NFR-Test

03-1 UAT

Test Strategy during SW-Dev. Introduction

- Question: How do you plan testing during Software Development (SW-Dev.)?
- Which testing process do you follow?
- Ideas:
  - Software Quality with high quality coding (Pair programming, Programming Guidelines, ...)
  - Effective Review Techniques (peer review, ...)
  - Following a Software Development Process with Milestones that care about quality

Test Strategy during SW-Dev. Checklist

Avoid blunders – following [CCM07]

- Blunder # 1: Building Infrequently
  - Consequences: Painful software build process
  - How to avoid?
    - Continuous Integration
    - Daily builds
    - Build Automation

Test Strategy during SW-Dev. Checklist

- Blunder #2: Not Validating Changes Against Target Environments
  - Consequences: Customer hears “Well, it worked on my machine”
  - How to avoid?
    - validate code changes regularly against all of the environments the product should support
    - automate

Test Strategy during SW-Dev. Checklist

- Blunder #3 and #4: Broken Builds, not testing the builds
  - Consequences: Developers, Testers wait ...
  - How to avoid?
    - Use Version Control
    - Cascading the integration:
      1. Successfully building own code
      2. Pretesting build
      3. Integrate
      4. Fixing problems until code integrates
    - Automated Build Validation Tests

Test Strategy during SW-Dev. Checklist

- Blunder #5: Infrequent Check-Ins
  - Consequences: Code could be lost
  - How to avoid? Use Version Control and check in!
- Blunder #6: Not publishing Builds
  - Consequences: Nobody knows status
  - How to avoid?
    - Build and Distribution Management
    - Simply determine a well-known location, define a self-describing numbering schema for files, and publish all builds there so everyone can access them

Test Strategy during SW-Dev. Checklist

- Blunder #7: Interpreting Code Coverage in Isolation
  - Consequences: Misinterpretation, because: High code coverage does not guarantee quality
  - How to avoid?
    - Work carefully, always new tests for new code
    - Define a target for code coverage that’s properly aligned with
      1. the complexity of the product being developed (more complex / more Test Cases)
      2. how often that code is accessed by the application (more often / more Test Cases)

Definition: Code coverage a metric that tells you how much of the source code has been tested

Test Strategy during SW-Dev. Checklist

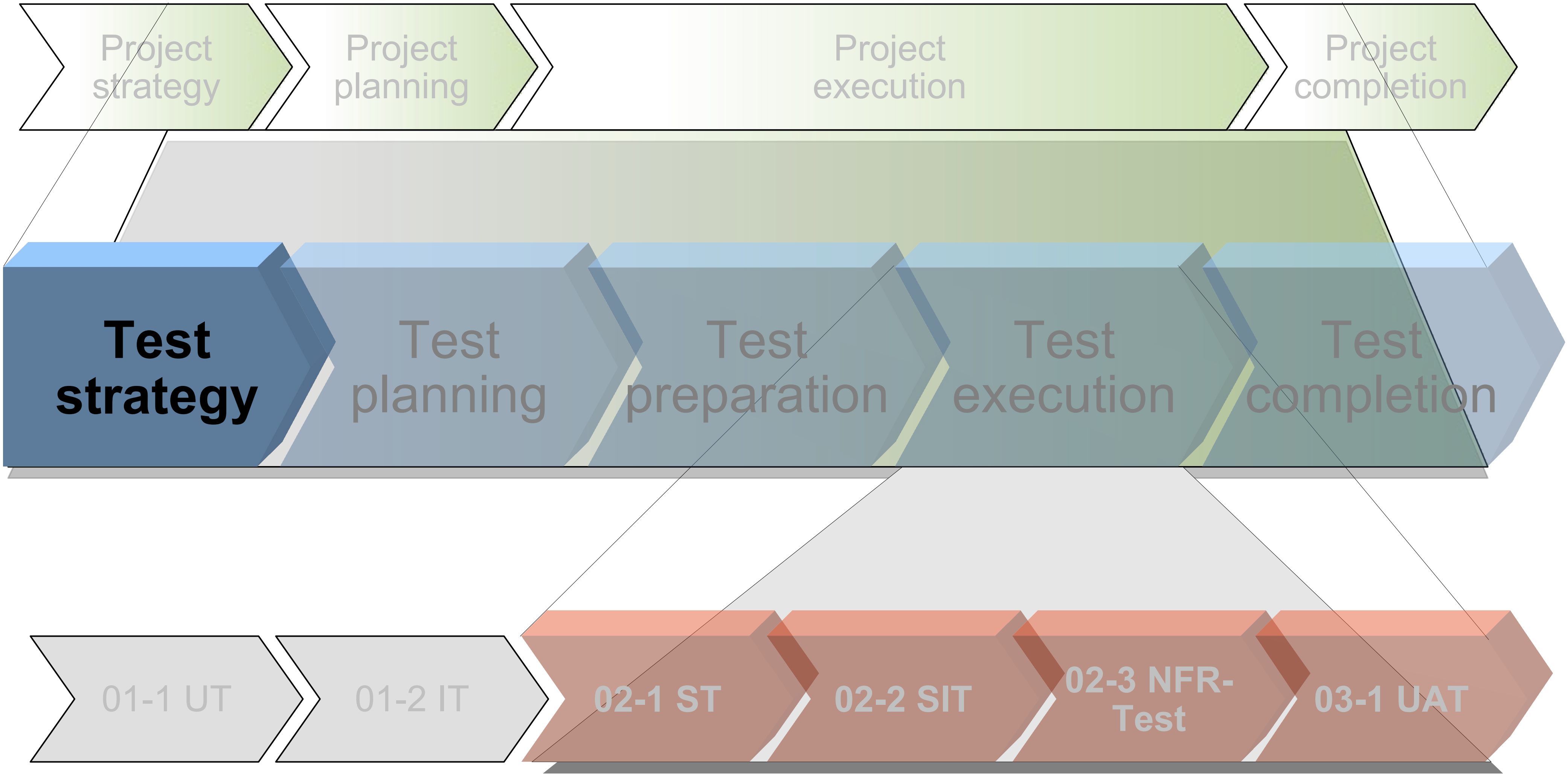
- Blunder #8: Lack of Visibility
  - Solution: Visibility
- Blunder #9: Relying Only on Unit Tests
- Blunder #10: Ignoring History
  - Solution: Capture and Mine Data

Next page

Test Strategy during SW-Dev. Checklist

- Blunder #9: Relying Only on Unit Tests
- Consequences: Even if code compiles successfully, your executable binary could be full of hidden problems
- How to avoid?
  - Regularly additional Tests like
    - Static analysis tools, e. g. FindBugs
    - Functional tests
    - Security analysis

A sample Testing Cycle
Test Strategy



Test Strategy Introduction

- Some words at the beginning...
  Some questions to ask...
  - The Test Strategy is always depending on the Software Project to test for – and its requirements:
    - How much time do we have?
    - What specification is available?
    - What is the requested quality?
    - How much budget is available, how many and which people?
    - What is the status of the delivered software?
    - Is it the first test?
    - Is there already a test process?

Test Strategy Introduction

- Questions to be answered by the Test Strategy
  - Expectations concerning test?
  - When will there be an acceptance of the software?
  - What is there?
    Specification, older versions, Documents, Prototypes
  - How to test, when to test, how to prepare, how to execute?
  - What kind of testing?
    - structured tests (test scenarios, test cases)
    - free tests (walk through, prototype test, desktop test)

Test Strategy Introduction

- Learning – Learn yourself! Ask around
  - Who has tested already?
  - Which experiences exist in the company, where you are working?
    - How did they test?
    - What went fine?
    - What should be improved?
  - Which tools are in use, experience?

Test Strategy Introduction

- Learning – Improve the organization
  - Discuss and decide with the Test Team how to ensure and improve quality of work like Test Cases, Test Data, and Test Scenarios
    - Support quality increasing techniques
  - Motivate the team to learn!
    - Establish "learning sessions"
    - Plan training

Test Strategy Introduction

- Learning – Improve the organization
  - Test Tandems
    - during Test Design, esp. Test Case Creation – one writing, one observing
    - during Test Execution
  - Reviews of people concerned
  - Plan rework – overwork
  - Establish a living “quality handbook”
  - Organize regularly Lessons learned workshops

Test Strategy Introduction

- Learning – Improve the processes
  - Determination of goals
    - The process should run faster and with reduced effort or
    - A higher test coverage should be achieved
  - Determination of strengths and weaknesses of the established test processes.
  - Definition of nominal condition concerning the defined goals and a definition of improvement action to achieve the nominal condition
  - Establishing and control

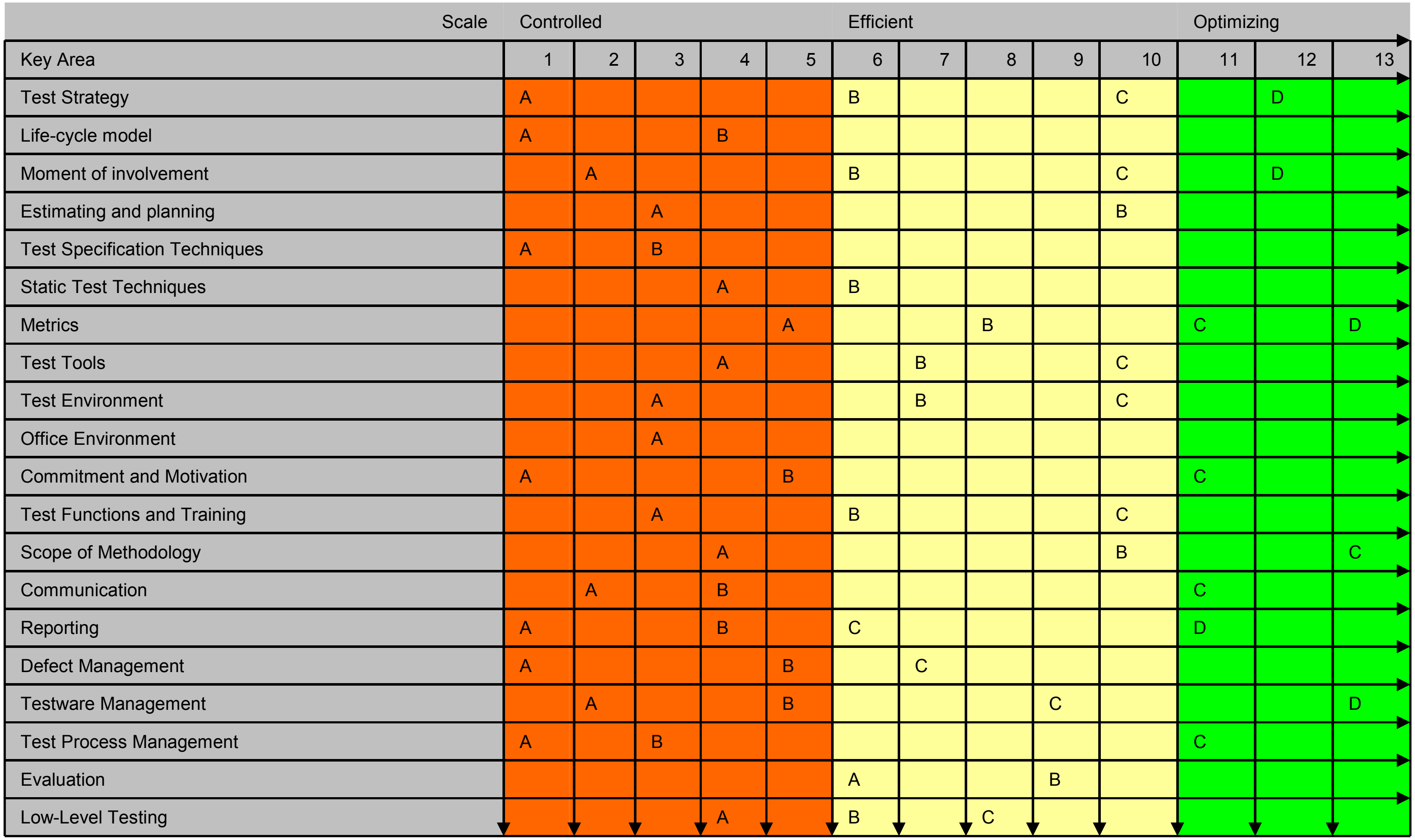
Test Strategy Introduction

- Learning – Improve the processes ... with TPI (Test Process Improvement)
  - 20 Key areas with up to 4 ascending Levels, e. g.:
    - Key area “Reporting”
      - Level A: Defects
      - Level B: Progress (status of tests and products), activities (costs and time, milestones), defects with priorities
      - Level C: Risks and recommendations, substantiated with metrics
      - Level D: Recommendations have a Software Process Improvement character
  - Checkpoints determine requirements to a Level
  - In a Test Maturity Matrix the organisation “wanders from left to right”

Source: [GM05]

Test Strategy Introduction

- Learning – Improve the processes ... with TPI



Scale	Controlled					Efficient					Optimizing		
Key Area	1	2	3	4	5	6	7	8	9	10	11	12	13
Test Strategy	A					B				C		D	
Life-cycle model	A			B									
Moment of involvement		A				B				C		D	
Estimating and planning			A							B			
Test Specification Techniques	A		B										
Static Test Techniques				A		B							
Metrics					A			B			C		D
Test Tools				A			B			C			
Test Environment			A				B			C			
Office Environment			A										
Commitment and Motivation	A				B						C		
Test Functions and Training			A			B				C			
Scope of Methodology				A						B			C
Communication		A		B							C		
Reporting	A			B		C					D		
Defect Management	A				B		C						
Testware Management		A			B				C				D
Test Process Management	A		B								C		
Evaluation						A			B				
Low-Level Testing				A		B		C					

Source: [GM05]

Test Strategy
Introduction

- Learning – Improve the processes ... with TPI – does it help?
  TPI Survey 2004 [GM05]
  - 61% of the people responded that implementation of TPI lead to better software quality e.g. less failures in production.
  - 83% of the people responded that implementation of TPI lead to (much) better testing control.
  - 92% of the people responded that the contribution of the TPI-model to the improvement process is (very) good.

Test Strategy Philosophy

- In Testing everything comes together:
  Tester are the first people, who see the finished product
- The finished product is the result of all project participants!
- That's why:
  One of the most important things is communication!
- Tester could be the headlights of the project

Test Strategy Philosophy

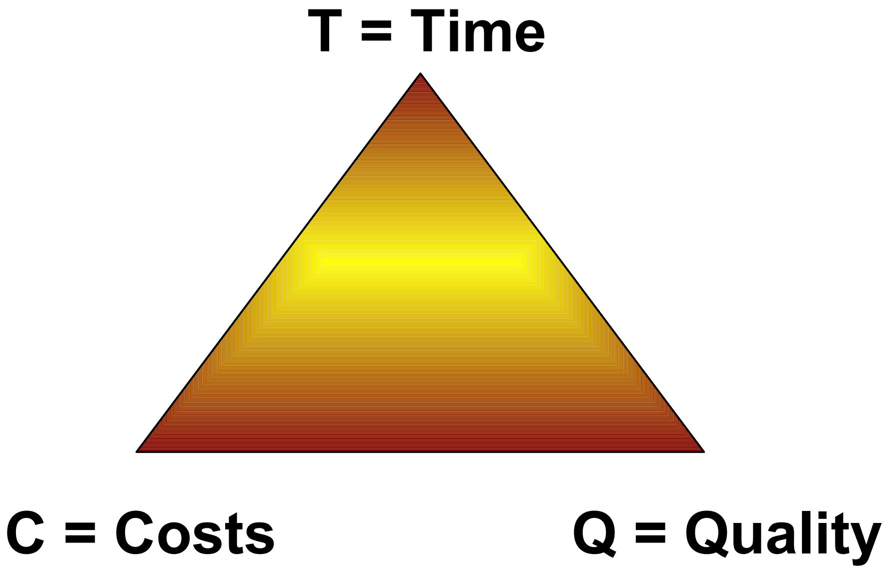
- Golden rules
  - Start with test as early as possible
  - Love defects
  - Plan Milestones
  - Integrative approach:
    Make people concerned (passive) to people involved (active)
  - Establish a service culture

Test Strategy Philosophy

- Ask better
  - Whom?
    - Test Case Designer
    - Software Developer
    - People who overwork the specification
  - NOT: Until when will you be ready (90 % trap)
  - BUT: What will be done until date X?

Test Strategy
Test as project in the project

- Project Management methods
- There is a start and an end
- Environmental analysis – communication strategy
- Magical triangle
  - **C**osts
  - **Q**uality
  - **T**ime



Test Strategy
Test as project in the project

- Important: Test is a part of an overall project, so test should follow the project ...
  - in strategy
  - in time schedule

Test Strategy
Factor Time

- If there is a time problem in the overall project, test is mostly in the focus of discussion
- Question: How could Testing help in reducing the expenditure of time?
- Bad idea: Reduce testing
- Good idea: Everything what could be done in advance should be done in advance
  ⇨ Especially administrative stuff

Test Strategy
Factor Time

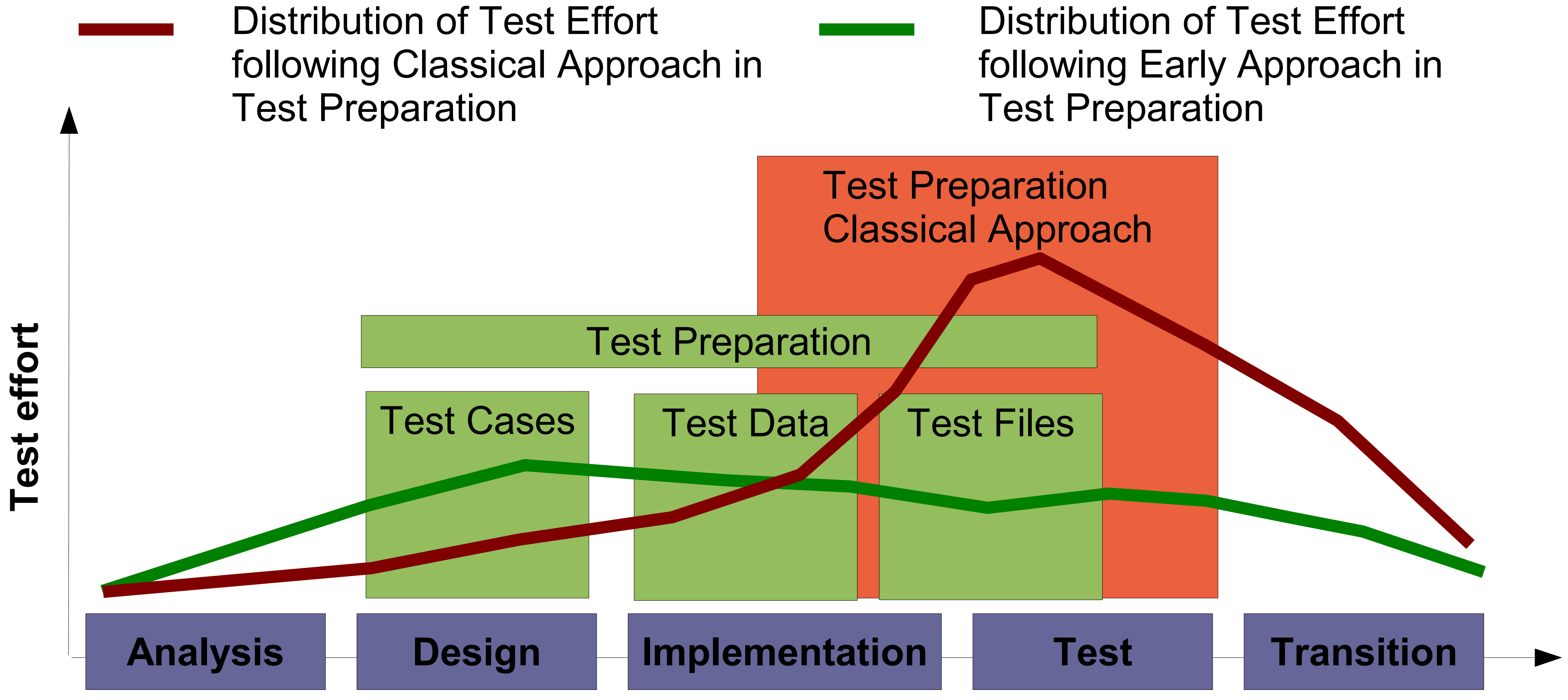
- Idea:
  As soon as the first software release could be delivered, everything should be prepared to start testing immediately in the best way
- Hmmm, how to do and what's this?

Test Strategy
Factor Time

- Start with test preparation as soon as possible
- Plan buffer (you will need it – anyhow ☺)
- What should be done before the first delivery?
  - Test lab and Test environment are ready
  - Processes are established
    (how are we going to start?)
  - Testers are trained
  - (Testing) Tools are working

Test Strategy
Factor Time

The earlier the test preparation starts, the lower is the total expenditure in testing!



Source SQS, following [Ueb06]

Test Strategy Goal

- Goal: Development of a proposed resolution and getting a decision of management concerning proceeding in testing
- Contents of the proposal should be:
  - Test Basics
    - Agreement of arrangement of acceptance
  - Test Strategy
    - Roles and responsibilities
  - Time schedule (has to be consistent with overall project)
  - ... and more

Objective of Testing

- Test Basics
  - Acceptance
  - Effort
  - Specification
- Test Strategy
  - People
  - Tools
  - Procedures
- Time Schedule
  - Stages
  - Release Mgmt.
  - Entry / exit criteria
- ...

Test Strategy
Goal – General

- Example for requirements to Test Strategy [GM05] – Discussion
  - The Aim of the Test Strategy is
    “Finding the most important defects
    as early and as cheaply as possible”
  - The Test Strategy defines which requirements and (product) risks are covered by what tests.
    The better each test level defines its own strategy and the more the different test level strategies are adjusted to each other, the higher the quality of the overall test strategy.

Test Strategy
Goal – General

- Define your goals depending on demands and constraints of the project (compare to definitions of testing)
  - Coverage
    - **Must** – what could be guaranteed to be tested?
    - **Should** – what is the main goal?
    - **Could** – what to test in remaining time?
  - Detection of defects – Severity / Priority levels
  - Statements concerning Software Quality
- Discuss expectation on testing

Test Strategy Goal – General

- Main task of Test Management is to
  - define a Test Strategy
  - synchronize Test Strategy with project stakeholders
  - transform the defined Test Strategy during
    - Test Planning
    - Test Preparation and
    - Test Execution
  - adapt the Test Strategy, if necessary

Test Strategy Goal – Contents

- Test Basics
  - Acceptance
    - Who (Roles) accepts
      when (Stages / Phases)
      what (Quality)?
    - What are the criteria to accept the software?
    - What is the demanded software quality?
    - Which defects are tolerated?
    - What is written in the contract?

Test Strategy
Goal – Contents

- Test Basics
  - Effort
    - How much budget?
    - How many people?
    - Which locations? Is there a Test lab?
  - Specification
    - Clarify: What is the basic for testing? Which documents do exist?

Test Strategy
Goal – Contents

Digression: Specification – Discussion [Mey99]
"Is the Moon an Enemy Rocket?"

- Specification
  - A system able to detect and identify flying objects.
  - Objects not identifiable as own airplanes are enemy targets.
- The system identified the moon as an enemy target. Was there a software defect?
  - "Soft" / user perspective: yes
  - "Hard" / developer perspective: no - specifications are satisfied

Test Strategy
Goal – Contents

- Test Strategy
  - People – Who should test?

·**Software developers**
- Conflict of interests
- Onesided perspective

·**Test specialist**
- Test Know-how
- Additional perspective
- Additional review of the specification.
- Mediator between software developers and customer

·**Customer**
- Knows what he wants
- Not a test expert
- Capacity problem possible

Test Strategy Goal – Contents

- Test Strategy
  - People – Test Team
    - Best people should test
    - Mixture
      - Experienced / novices
      - Male / female
      - International
      - Test experts / business area experts
      - Different characteristics
    - Roles – Responsibilities, Competence, Tasks
    - Location
      - All members of a test team should work together at one location

Test Strategy
Goal – Contents

- Test Strategy
  - People – Cooperation with Stakeholders
    - Identify roles and responsibilities:
      Who is going to do what?
    - RACI-Matrix*
    - Integrative approach
      Collaboration with customer, specification creators, operation, software vendor, contact people for connected systems
    - Communication model

* Responsible, Accountable, Consult, Inform
To identify and to define roles of people involved in the project

Test Strategy
Goal – Contents

- Test Strategy
  - People – Special cooperation with Software vendor
    - Which test processes does the vendor use?
      - Exchange of testing results?
    - How to exchange?
      - Test Cases / Test Scenarios: Review, first Testing
      - Test Data: Delivery, review
    - Collaboration in testing
      - Release Management
        - Release Notes
        - Smoke Test
      - Defect Management

Test Strategy
Goal – Contents

- Test Strategy
  - Tools
    - Which test tools to use?
      - Corporate strategy
      - Different topics (Test Management, Test Automation, Load tests, Performance Tests)
    - Proceeding in selection

Test Strategy
Goal – Contents

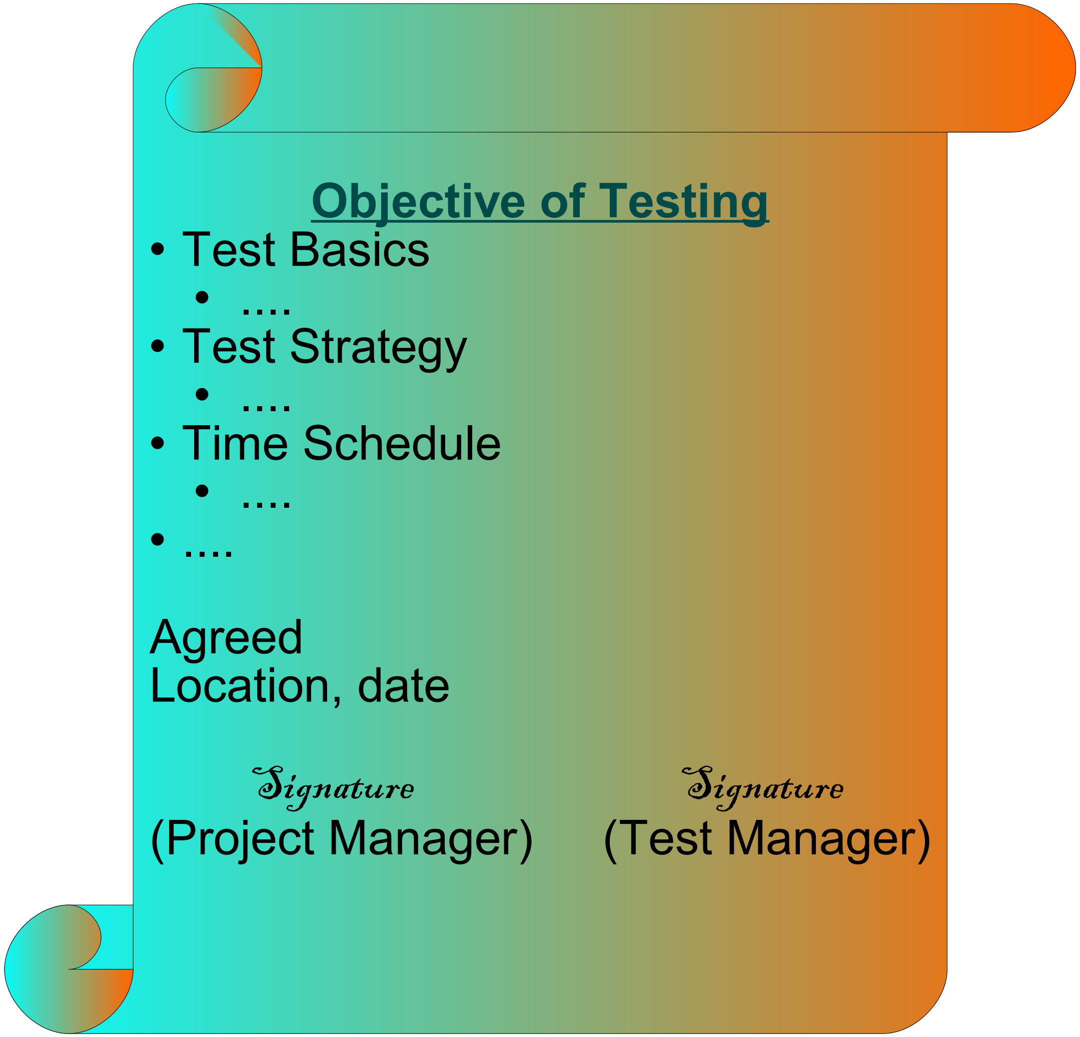
- Test Strategy
  - Procedures
    - Definition of the test process
    - Test Cases / Test Scenarios
    - Test Data
    - Interfaces
    - Test Environment (Hardware, Software, other systems)
      - Different Test Environments for different Tests
    - Milestones in content
    - Defect / Change management (from specification to software)
    - More: Smoke Test, Regression Test, ...

Test Strategy Goal – Contents

- Time Schedule
  - Develop first Time Schedule: When to do what with which resources?
    - Test planning
    - Test preparation
    - Test execution with Stages
      - Functional Test
      - System Integration Test
      - NFR Test
      - UAT
  - For each stage: Entry / exit criteria
  - Release Management

Test Strategy
Goal – Summary

- Document the above in an “Agreement on Objectives”
  - has to fit to the overall project strategy
  - to be discussed with and agreed by all project participants
  - to be decided by Project Manager



Objective of Testing

- Test Basics
  - ....
- Test Strategy
  - ....
- Time Schedule
  - ....
- ....

Agreed
Location, date

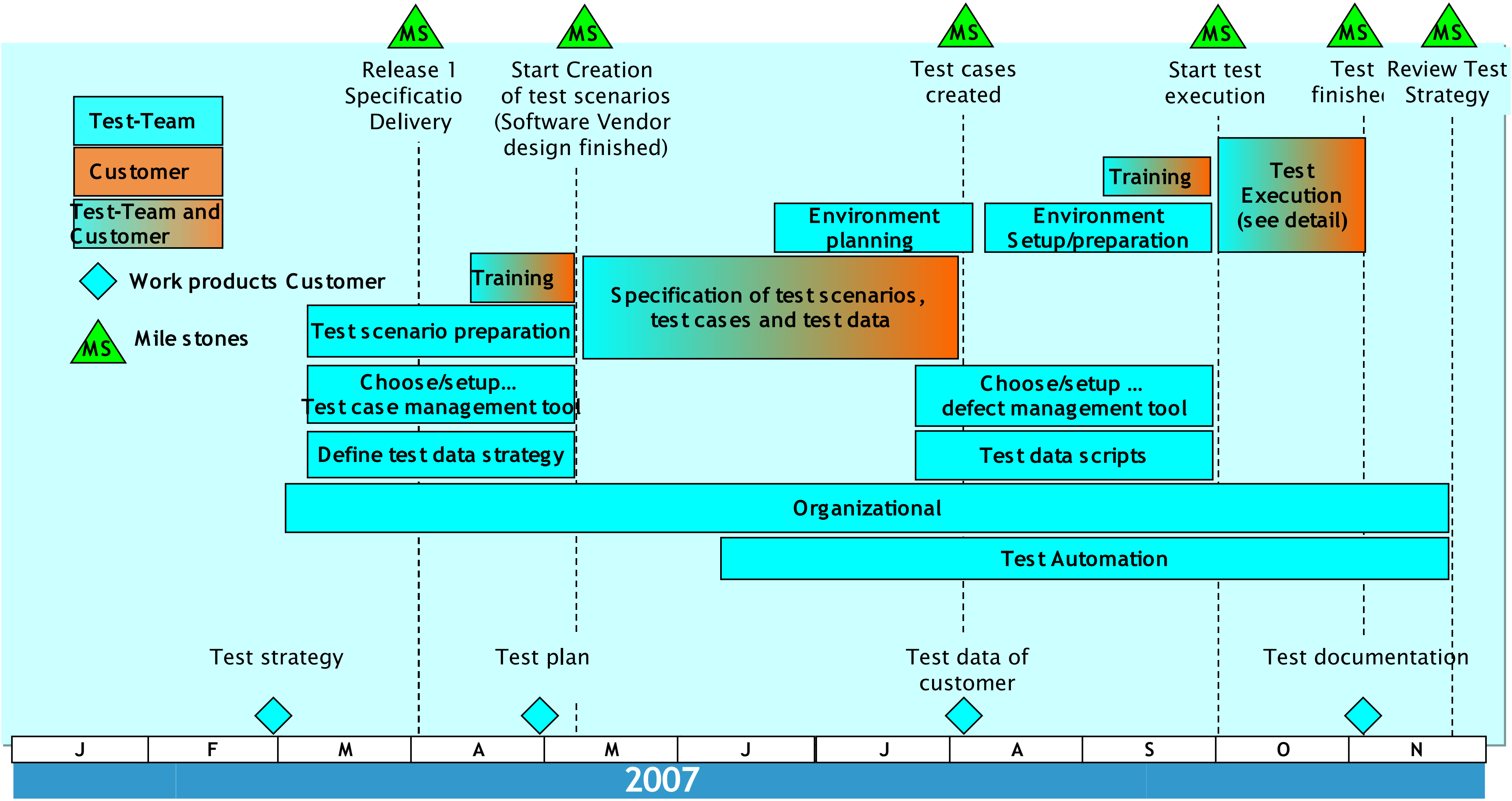
Signature	*Signature*
(Project Manager)	(Test Manager)

Test Strategy Goal – Summary

- Example
  - For a template study [Tur06]

Test Strategy
Goal – Summary

- Example first Time Schedule:



MS Release 1 Specificatio Delivery

MS Start Creation of test scenarios (Software Vendor design finished)

MS Test cases created

MS Start test execution

MS Test finishe

MS Review Test Strategy

Test-Team

Customer

Test-Team and Customer

◇ Work products Customer

MS Mile stones

Training

Test scenario preparation

Choose/setup... Test case management tool

Define test data strategy

Specification of test scenarios, test cases and test data

Environment planning

Environment Setup/preparation

Training

Test Execution (see detail)

Choose/setup ... defect management tool

Test data scripts

Organizational

Test Automation

◇ Test strategy

◇ Test plan

◇ Test data of customer

◇ Test documentation

J	F	M	A	M	J	J	A	S	O	N

2007

Test Strategy
Links / Sources

- [CCM07] Rob Cheng, Burke Cox, John Minnihan: The Top 10 Blunders In Integration And Testing ... And How To Avoid Them, White Paper, http://www.borland.com/resources/en/pdf/products/gauntlet/top10blunders.pdf, 2007
- [Mey99] Myers, Glenford J.: Methodisches Testen von Programmen; München, Wien, Oldenbourg; 1999
- [GM05] Mattias Göransson, Theis Meggerle: Test Process Improvement Based on the TPI Model, testway http://www.testway.se/seminars_e.html, 2005
- [Tur06] Neville Turbit: Developing a Test Strategy, http://www.projectperfect.com.au/info_test_strategy.php, 2006
- [Ueb06] Stefan Ueberhorst: Wer zu spät testet, verschleudert Geld; Computerwoche http://www.computerwoche.de/produkte_technik/software/581145/index.html ; 08.09.2006