Certified Tester

Expert Level Syllabus

Improving the Testing Process
(Implementing Improvement and Change)

Beta Version 2009

FOR INFORMATION ONLY
COMMENTS FROM BETA REVIEW NOT YET IMPLEMENTED
LEARNING OBJECTIVES MAY BE ADJUSTED

International Software Testing Qualifications Board

Copyright Notice
This document may be copied in its entirety, or extracts made, if the source is acknowledged.

© International Software Testing Qualifications Board

Copyright © International Software Testing Qualifications Board (hereinafter called ISTQB®).

Expert Level Working Party: Erik van Veenendaal, Graham Bath, Isabel Evans 2006-2009.

© International Software Testing Qualifications Board

Revision History

Version	Date	Remarks
V2009-Beta	19.6.2009	Beta Release

© International Software Testing Qualifications Board

Table of Contents

Revision History 4
Table of Contents 5
Acknowledgements 8
1. Introduction to this syllabus 9
1.1 The International Software Testing Qualifications Board 9
1.2 Purpose of this document 9
1.3 The Certified Tester Expert Level in Software Testing 9
1.3.1 Level of knowledge 10
1.3.2 Examination 10
1.3.3 Accreditation 10
1.4 Normative versus informative parts 10
1.5 Level of detail 10
1.6 How this syllabus is organized 11
1.7 Terms and Definitions 11
1.8 Learning Objectives / Level of Knowledge 11
1.9 Expectations 12
2. The context of improvement 390 mins. 14
2.1 Why improve testing? 14
2.2 What can be improved? 16
2.3 Views of quality 16
2.4 The generic improvement process 17
2.4.1 The Deming Cycle 17
2.4.2 The IDEAL improvement framework 18
2.4.3 Fundamental Concepts of Excellence 18
2.5 Overview of improvement approaches 19
2.5.1 Overview of model based approaches 19
2.5.2 Overview of analytical approaches 20
2.5.3 Hybrid approaches 20
2.5.4 Other approaches to improving the test process 20
3. Model-based improvement 495 mins. 24
3.1 Introduction to model-based approaches 24
3.1.1 Desirable characteristics of test process improvement models 24
3.1.2 Continuous and Staged Representations 25
3.1.3 Assumptions in using models 26
3.2 Software process improvement models 26
3.2.1 CMMI 26
3.2.2 ISO/IEC 15504 27
3.3 Test process improvement models 27
3.3.1 The Test Process Improvement Model (TPI®) 27
3.3.2 The Testing Maturity Model Integration (TMMi) 28
3.3.3 Comparing TPI to TMMi 28
3.4 Content-based models 29
3.4.1 STEP 29
3.4.2 CTP 29
4. Analytical-based improvement 510 mins. 30
4.1 Introduction 30
4.2 Causal analysis 31
4.2.1 Cause-Effect diagrams 31
4.2.2 Causal analysis during an inspection process 32
4.2.3 Use of standard anomaly classifications 32

© International Software Testing Qualifications Board

4.2.4 Selecting defects for causal analysis ... 32
4.3 The GQM Approach ... 33
4.4 Analysis using Measures, Metrics and Indicators ... 33
4.4.1 Introduction ... 33
4.4.2 Test effectiveness metrics ... 34
4.4.3 Test efficiency / cost metrics ... 34
4.4.4 Lead-time metrics ... 35
4.4.5 Predictability metrics ... 35
4.4.6 Product quality metrics ... 35
4.4.7 Test maturity metrics ... 36
5. Selecting the approach for test process improvement 105 mins. ... 37
5.1 Selecting test process improvement approaches ... 37
6. Process for improvement 765 mins. ... 39
6.1 Introduction ... 40
6.2 Initiating the improvement process ... 40
6.2.1 Establishing the Need for Improvement ... 40
6.2.2 Setting objectives for test improvement ... 41
6.2.3 Setting the Scope for Improvement ... 42
6.2.4 Influence of people and culture on the initializing phase ... 43
6.3 Diagnosing the current situation ... 43
6.3.1 Planning the assessment ... 44
6.3.2 Assessment preparation ... 44
6.3.3 Performing interviews ... 45
6.3.4 Initial feedback ... 45
6.3.5 Analysis of results ... 45
6.3.6 Performing solution analysis ... 46
6.3.7 Recommending improvement actions ... 47
6.4 Establishing a Test Improvement Plan ... 47
6.4.1 Setting priorities ... 48
6.4.2 Developing an implementation approach ... 48
6.4.3 Planning the improvements ... 49
6.5 Acting to implement improvement ... 49
6.5.1 Selecting and executing a pilot ... 50
6.5.2 Manage and control the implementation ... 50
6.6 Learning from the improvement program ... 50
7. Organization, Roles and Skills 375 mins. ... 51
7.1 Organization ... 51
7.1.1 The Test Process Group ... 51
7.1.2 Test improvement with remote, off-shore and outsourced teams ... 52
7.2 Individual Roles ... 53
7.2.1 The Test Process Improver ... 53
7.2.2 Assessor roles ... 53
7.3 Skills of the Test Process Improver ... 53
7.3.1 Interviewing skills ... 54
7.3.2 Presentation and reporting skills ... 56
7.3.3 Analytical skills ... 56
7.3.4 Skills of persuasion ... 57
7.3.5 Management skills ... 57
7.3.6 Note-taking skills ... 57
8. Managing change 240 mins. ... 59
8.1 Introduction ... 59
8.2 Fundamental change management process ... 59
8.3 Human factors in the change management process ... 60
9. Critical Success Factors 225 mins. ... 62

© International Software Testing Qualifications Board

9.1 Key success factors 62
9.2 Setting a culture for improvement 63
10. Adapting to different Lifecycle models 60 mins. 65
10.1 Adapting to different lifecycle models 65
11. References 67
11.1 Standards 67
11.2 Trademarks 67
11.3 Books 67
11.4 Papers and Articles 69
11.5 Web references 70
12. Appendix A – Notice to Training Providers 71
12.1 Training Times 71
12.2 Standards used 72
12.3 Practical Exercises 72
12.3.1 Guidelines for practical exercises in the workplace 72
12.4 Generic guidance to training providers regarding criteria 73
12.4.1 Expert Level training provider criteria 73
12.4.2 Expert Level courses 73
12.4.3 Trainer entry criteria 73
Index 75

© International Software Testing Qualifications Board

Acknowledgements

This document was produced by a core team of authors from the International Software Testing Qualifications Board Expert Level Working Party for the module “Improving the Testing Process”:

Graham Bath (chair for this syllabus)

Isabel Evans

Erik van Veenendaal (ISTQB chair Expert Level)

The core team thanks the review team and all National Boards for their suggestions and input.

At the time the Expert Level Syllabus for this module was completed the Expert Level Working Party had the following membership (alphabetical order):

Graham Bath	Beata Karpinska	Klaus Olsen
Rex Black	Caroline Molloy	Meile Posthuma
Isabel Evans	Silvio Moser	Maud Schlich
Matthias Hamburg	Thomas Müller	Erik van Veenendaal
Kari Kakkonen	Ingvar Nordstrom	

The following persons participated in the reviewing, commenting and balloting of this syllabus (alphabetical order):

Graham Bath	Beata Karpinska	Klaus Olsen
Rex Black	Kari Kakkonen	Meile Posthuma
Sigrid Eldh	Marcel Kwakernaak	Stuart Reid
Isabel Evans	Caroline Molloy	Maud Schlich
Cheryl George	Thomas Müller	Neil Thompson
Derk-Jan de Grood	Silvio Moser	Erik van Veenendaal
Matthias Hamburg	Ingvar Nordstrom	Brian Wells

This document was formally released by the General Assembly of ISTQB® on xx 2009.

© International Software Testing Qualifications Board

1. Introduction to this syllabus

1.1 The International Software Testing Qualifications Board

The International Software Testing Qualifications Board (hereinafter called ISTQB®) is made up of Member Boards representing countries or regions around the world. More details on the structure and membership of the ISTQB may be found at [ISTQB-Web]

1.2 Purpose of this document

This syllabus forms the basis for the International Software Testing Qualification at the Expert Level for the module "Improving the Testing Process". The ISTQB® provides this syllabus as follows:

1. To Member Boards, to translate into their local language and to accredit training providers. National boards may adapt the syllabus to their particular language needs and modify the references to adapt to their local publications.
2. To Exam Boards, to derive examination questions in their local language adapted to the learning objectives for each module.
3. To training providers, to produce courseware and determine appropriate teaching methods.
4. To certification candidates, to prepare for the exam (as part of a training course or independently).
5. To the international software and systems engineering community, to advance the profession of software and systems testing, and as basis for books and articles.

The ISTQB® may allow other entities to use this syllabus for other purposes, provided they seek and obtain prior written permission.

1.3 The Certified Tester Expert Level in Software Testing

The Expert level qualification is aimed at people who have already achieved an advanced point in their careers in software testing and wish to develop further their expertise in a specific area. The modules offered in at the Expert Level cover a wide range of testing topics.

A testing expert is one that has a broad understanding of testing in general, and an in depth understanding in a special test area. An in-depth understanding means having sufficient knowledge of testing theory and practice to be able to influence the direction that an organization and/or project takes when creating, implementing and executing testing processes.

To participate in the Expert level exam in the module "Improving the Testing Process", candidates must hold the Advanced level certificate in test management.

In addition to passing the exam, proof needs to be provided of practical working experience in the testing field in general and specifically to the Expert level module that applies before the Expert Level certificate is awarded. In addition to passing the exam the following requirements apply:

- at least five years of testing experience in practice (CV needs to be submitted including two references)
- at least two years of experience in the Expert level module topic (CV needs to be submitted including two references)
- at least one paper written and published, OR a presentation is held at a testing conference regarding the Expert level module topic.

© International Software Testing Qualifications Board

Persons that formally comply with the defined exit criteria will receive the formal ISTQB Expert level certificate for the underlying module. Those that possess an ISTQB Expert level certificate will also be allowed to use the Certified Tester Expert Level (CTEL) acronym.

Holders of an Expert certificate in a particular module should regularly renew their certification by achieving a minimum number of credits within the ISTQB Certification Extension Process [ISTQB-CEP]. Further details of this process may be found at [ISTQB-Web].

1.3.1 Level of knowledge

Learning objectives for each chapter are divided such that they can be clearly identified for each individual module.

1.3.2 Examination

All Expert Level Certificate examinations shall be based on this syllabus, on the Advanced level syllabus, module test management (especially chapter 8 "Standards and Test Improvement Process") and on the Foundation Level syllabus. Answers to examination questions may require the use of material based on more than one section of this syllabus, the Advanced Level syllabus (module Test Management) and/or the Foundation Level syllabus.

The format of the examination is defined by the Expert Exam Guidelines of the ISTQB.

Exams may be taken as part of an accredited training course or taken independently (e.g., at an examination center). Exams may be taken on paper or electronically, but all exams must be proctored / observed (supervised by a person mandated by a National or Examination Board).

1.3.3 Accreditation

An ISTQB Member Board may accredit training providers whose course material follows this syllabus. Training providers should obtain accreditation guidelines from the board or body that performs the accreditation. An accredited course is recognized as conforming to this syllabus, and is allowed to have an ISTQB examination as part of the course.

1.4 Normative versus informative parts

Normative parts of the syllabus are examinable. These are:

- Learning objectives
- Keywords
- Required exercises in the work-place

The rest of the syllabus is informative and elaborates on the learning objectives.

1.5 Level of detail

The level of detail in this syllabus allows internationally consistent teaching and examination. In order to achieve this goal, the syllabus consists of:

- General instructional objectives describing the intention of the Expert Level
- Learning objectives for each knowledge area, describing the cognitive learning outcome and mindset to be achieved (these are normative)
- A list of information to teach, including a description of the key concepts to teach, including sources such as accepted literature or standards, and references to additional sources if required (these are informative)

© International Software Testing Qualifications Board

The syllabus content is not a description of the entire knowledge area of improving the test process; it reflects the level of detail to be covered in this Expert Level training course.

1.6 How this syllabus is organized

There are ten major chapters. The top level heading shows the time for the chapter. For example:

2. The Context of Improvement 180 mins.

shows that chapter 2 is intended to have a time of 180 minutes for teaching the material in the chapter. Specific learning objectives are listed at the start of each chapter.

1.7 Terms and Definitions

Many terms used in the software literature are used interchangeably. The definitions in this Expert Level Syllabus are available in the Standard glossary of terms used in software testing, published by the ISTQB

Each of the keywords listed at the start of each chapter in this Expert Level Syllabus are defined in the Standard Glossary of Terms used in Software Testing, published by the ISTQB.

1.8 Learning Objectives / Level of Knowledge

The following learning objectives are defined as applying to this syllabus. Each topic in the syllabus will be examined according to the learning objective for it.

Level 1: Remember (K1)
The candidate will recognize, remember and recall a term or concept.
Keywords: Remember, recall, recognize, know.
Example
Can recognize the definition of “failure” as:

- “non-delivery of service to an end user or any other stakeholder” or
- “actual deviation of the component or system from its expected delivery, service or result”.

Level 2: Understand (K2)
The candidate can select the reasons or explanations for statements related to the topic, and can summarize, differentiate, classify and give examples for facts (e.g. compare terms), the testing concepts, test procedures (explaining the sequence of tasks).
Keywords: Summarize, classify, compare, map, contrast, exemplify, interpret, translate, represent, infer, conclude, categorize.
Examples
Explain the reason why tests should be designed as early as possible:

- To find defects when they are cheaper to remove.
- To find the most important defects first.

Explain the similarities and differences between integration and system testing:

- Similarities: testing more than one component, and can test non-functional aspects.
- Differences: integration testing concentrates on interfaces and interactions, and system testing concentrates on whole-system aspects, such as end to end processing.

Level 3: Apply (K3)
The candidate can select the correct application of a concept or technique and apply it to a given context. K3 is normally applicable to procedural knowledge. There is no creative act involved like evaluating a software application, or creating a model for a given software. When we have a given model and cover in the syllabus the procedural steps to create test cases from a model, then it is K3.
Keywords: Implement, execute, use, follow a procedure, apply a procedure.

© International Software Testing Qualifications Board

Example
- Can identify boundary values for valid and invalid partitions.
- Use the generic procedure for test case creation to select the test cases from a given state transition diagram (and a set of test cases) in order to cover all transitions.

Level 4: Analyze (K4)
The candidate can separate information related to a procedure or technique into its constituent parts for better understanding, and can distinguish between facts and inferences. Typical application is to analyze a document, software, project situation and propose appropriate actions to solve a problem or task.
Keywords: Analyze, differentiate, select, structure, focus, attribute, deconstruct, evaluate, judge, monitor, coordinate, create, synthesize, generate, hypothesize, plan, design, construct, produce.
Example
- Analyze product risks and propose preventive and corrective mitigation activities.
- Describe which portions of an incident report are factual and which are inferred from results.

Level 5: Synthesize (K5)
The candidate can adapt an existing process to meet changing risks, business needs or stakeholder demands. Typical application is to combine and modify two processes in organizations working together, for example the test incident management system to be used by three organizations – a customer and two third party suppliers – during a joint project.

Keywords: synthesize, combine, integrate, modify, rearrange, substitute, plan, create, design, invent, what if?, compose, formulate, prepare, generalize, rewrite.
Example
- Adapt a current test process to meet tactical requirements of a particular project, for example, if a new type of project/technology is encountered.
- Show how a proposed process is modification of a standard process and where/why it has been adapted to a particular situation

Level 6: Evaluate (K6)
The candidate can provides reasoned evaluation of a scenario, together with recommendations at a level suitable for presentation to senior management within IT and customer/sponsor managers. Typical application is to evaluate a current process and recommend improvements and the mechanism for applying the recommendations
Keywords: evaluate: assess, decide, rank, grade, test, measure, recommend, convince, select, judge, explain, discriminate, support, conclude, compare, summarize
Example
- Evaluate a current test incident management process, and write a report comparing it to industry standards, identify improvements in the process for both effectiveness and efficiency including cost savings, and present the report and recommendations to a senior management panel. Use arguments in the reports that identify and *address the requirements of different stakeholders in the management meeting*

Refer to [Bloom] and [Anderson] for details about the cognitive levels of learning objectives.

1.9 Expectations

The Learning Objectives in this syllabus are intended to develop participants to fulfill the following expectations:
- To advise on test process improvement
- To ensure that the implementation of test process improvements within their organization or project takes place effectively and stands the best chance of success
- To fulfill the specific expert role within their organization or project.

© International Software Testing Qualifications Board

It is not intended that candidates who qualify at Expert Level should immediately be considered as "world experts" in test process improvement. The expectation is that the qualified ISTQB CTEL in Improving the Test Process will be able to provide specialist support within their organization or project to initiate, implement and support improvements to testing in that organization or project.

© International Software Testing Qualifications Board

2. The context of improvement 390 mins.

Keywords:

analytical approach, Deming cycle, EFQM, fundamental concepts of excellence, hybrid approach, IDEAL, manufacturing-based quality, model-based approach, product-based quality, retrospective meeting, SCRUM, software lifecycle, Software Process Improvement (SPI), standards, test tools, Total quality management (TQM), transcendent quality, user-based quality, value-based quality

Learning objectives for context of improvement

2.1 Why improve testing?

LO 2.1.1 (K2) Give examples of the typical reasons for test improvement
LO 2.1.2 (K2) Contrast test improvement with other improvement goals and initiatives
LO 2.1.3 (K4) Explain to all stakeholders the reasons for proposed test process improvements, show how they are linked to business goals and explain them in the context of other process improvements

2.2 What can be improved?

LO 2.2.1 (K2) Understand the different aspects of testing, and related aspects, that can be improved

2.3 Views of Quality

LO 2.3.1 (K2) Compare the different views of quality
LO 2.3.2 (K4) Relate the different views of quality to testing

2.4 Generic Improvement Process

LO 2.4.1 (K2) Understand the steps in the Deming Cycle
LO 2.4.2 (K4) Compare two generic methods (Deming Cycle and IDEAL framework) for improving processes
LO 2.4.3 (K2) Give examples for each of the Fundamental Concepts of Excellence with regard to test process improvement

2.5 Overview of improvement approaches

LO 2.5.1 (K2) Compare the characteristics of a model-based approach with analytical and hybrid approaches
LO 2.5.2 (K2) Understand that a hybrid approach may be necessary
LO 2.5.3 (K2) Understand the need for improved people skills and explain improvements in staffing, training, consulting and coaching of test personnel.
LO 2.5.4 (K2) Understand how the introduction of test tools can improve different parts of the test process
LO 2.5.5 (K2) Understand how improvements may be approached in other ways, for example, by the use of periodic reviews during the software life cycle, by the use of test approaches that include improvement cycles (e.g. retrospectives in SCRUM), by the adoption of standards, and by focusing on resources such as test environments and test data.

2.1 Why improve testing?

Systems in which software is a dominant factor are becoming more and more challenging to build. They are playing an increasingly important role in society. New methods, techniques, and tools are

© International Software Testing Qualifications Board

becoming available to support development and maintenance tasks. Because systems play such an important role in our lives both economically and socially, there is pressure for the software engineering discipline to focus on quality issues. Poor quality software is no longer acceptable to society. Software failures can result in huge business losses or even become catastrophic, e.g. loss of human lives.

Improving the test process should take place within the context of

- current business and organizational challenges,
- the maintenance challenges of currently delivered systems, and
- current testing and quality assurance challenges.

In this context the importance of the testing discipline, as one of the quality measures that can be taken, is growing rapidly. Often projects spend substantial parts of their budget on testing To meet these challenges the task of testing has now become a profession and should be treated as such.

Organizations face tougher business objectives everyday, e.g. decreased time-to-market, higher quality and reliability and reduced costs. We develop and manufacture products that consist of over 60% of software and this number is still growing. At the same time, options are now available for software development to be outsourced or co-developed with other sites. Together with the trend towards more re-use and platform-architecture, integration and testing are becoming key activities that directly influence not only the product quality but also the effective and efficient conduct of the entire development and manufacturing process.

Software is increasing in importance and size in our everyday-world. The amount of software in consumer products roughly doubles every 24 months, and this also holds for the complexity in professional applications; factors which influence the number of defects per ‘”unit” of software (e.g. Function Points). As the market is demanding better and more reliable products that are developed and produced in less time with a reduced amount of money, higher testing effectiveness and efficiency is no longer an option; it is an essential ingredient for success.

Delivered systems include other products and services as well as software code. The system may also include hardware, middleware and firmware. In some circumstances, the delivered service might include new buildings, changes to working practices and new infrastructure, so testing may extend to dress rehearsals of the transition and the first days in a full working organization in its new premises.

The scope of testing is therefore unlikely to be just the software system. Further, the people buying and using software don’t just need the code, they also need services and products such as business processes, training, user guides and support. The improvement of testing must be carried out in the context of the wider quality goals of the customers of testing – whether these are an organization, one or more customer organizations or one or more IT groups/teams.

The context within which test process improvement takes place includes any business/organizational process improvement and any IT or software process improvement.

Typical reasons for business improvements which influence testing are:

- A business may have a testing service that provides a high quality engineering approach which takes too long; if the goal is to reduce time to market but high quality must be maintained, then test improvement may be focused on
    - cutting the time for testing by increased efficiency without reducing effectiveness.
    - Increasing earlier testing activities (e.g. static testing) in order to reduce time taken to resolve defects later in the life-cycle.

© International Software Testing Qualifications Board

- Organizations may need to increase the quality of deliverables as a primary focus for test improvement; for them increased time and cost on testing may be a reasonable price for improved quality.
- Organizations may wish to increase ability of testers to provide predictability and to report well.
- Organizations that provide third party support may need to meet client requirements for their suppliers to be at a particular capability level (see examples of IT and software process improvement approaches below).
- An organization may have a goal of saving money and therefore wish to cut the cost of the testing.
- Organizations may wish to reduce the costs of failure and therefore improve their testing.

Test process improvement may take place in the context of organizational and business improvement. This may typically be managed via one of the following (note these are mentioned as examples only):

- Total Quality Management (TQM)
- ISO 9000:2000
- An excellence framework, for example the European Foundation for Quality Management (EFQM) Excellence Model™ or its equivalent in other continents
- Six Sigma

Test process improvement may take place in the context of IT/software process improvement. This may typically be managed via one of the following:

- Capability Maturity Model Integrated (CMMI). See section 3.2.1.
- ISO/IEC 15504. See section 3.2.2.
- ITIL [ITIL], [ITIL2]
- Team and Personal Software Process [Humphrey]

2.2 What can be improved?

Software Process Improvement (SPI) is often referred to as the continuous improvement of quality and/or efficiency of the process on the basis that an improved process leads to improved quality of the product.

Test Improvement is often referred to as the continuous improvement of the quality, and the efficiency of the testing process, in the context of the overall software process. This context means that improvements to testing may go beyond just the process itself, for example extending to infrastructure and testers' skills. Also, testing improvements may indicate that associated or complementary improvements are needed to requirements management and other parts of the development process. Conversely, testing improvements may be driven by overall SPI.

Testing goals must always be aligned to business goals. It is not always relevant for an organization or project to achieve the maximum levels of test maturity.

2.3 Views of quality

In a single project we may use several definitions of quality, perhaps inadvertently and unacknowledged by all the people in the project. It is important to realize that there is no “right” definition of quality. How we define “quality” for a particular product, service or project is situational. Improvement of the test process should consider which of the quality views discussed in this section are most useful for the organization.

Five views of software quality are explained with examples in [Trienekens and van Veenendaal 97], which is based on [Garvin Paper 84]. The five views are:

© International Software Testing Qualifications Board

- Product
- Manufacturing
- User
- Value
- Transcendent

In terms of the types, levels and techniques that may be used through the test process, the quality viewpoints may be addressed during static and dynamic test activities by use of roles and engagement of representative view points [see example in Evans04].

It is important to realize that there is no “right” definition of quality. How we define “quality” for a particular product, service or project is situational. Different industries will have different quality views. Safety critical products will require an emphasis on the product and manufacturing definitions of quality. Entertainment and games products will require the user definition and may also require product attributes not normally considered in other industries – for example Excitement as part of Usability. Software to launch an innovative new product and achieve “first mover” advantage requires the value-based definition because if we spend more to get a better product we may miss the market. For most commercial or bespoke software, the customer is best served by balancing the definitions: In “our particular project”, we should ask ourselves: What is the greatest number or level of attributes (product-based) that we can deliver to support the users’ tasks (user-based) while giving best cost-benefit (value-based) while following repeatable / quality assured processes within a managed project (manufacturing-based)?

The metrics that could be associated with these quality views are discussed in chapter 4, Analytical Methods and chapter 6, Process for Improvement.

2.4 The generic improvement process

2.4.1 The Deming Cycle

Continuous improvement involves the setting of improvement objectives, taking measures to achieve them and, once they have been achieved, setting new improvement objectives. Continuous Improvement Models have been established to support this concept.

The Deming-Cycle is a useful generic framework for achieving continuous improvement and consists of the following steps:

- **Plan:** targets are defined for quality characteristics, costs and service levels. The targets may initially be formulated by management as business improvement goals and successively broken down until individual "control points" which should be checked (in step 3) to see that the activities have been carried out. Defined objectives should be measurable (for further information, refer to section 4.4). An analysis of current practices and skills is performed after which improvement plans are set up for improving the test process.

- **Do:** After the plans have been made, the activities are performed to ensure that they are introduced. Included in this step is investment in human resources (e.g. training and coaching).

- **Check:** The control points identified in the planning step are tracked using specific metrics, and deviations observed. The variations in each metric may be predicted for a specific time interval and compared with the actual observations to provide information on both actual and expected deviations.

© International Software Testing Qualifications Board

- **Act** (sometimes referred to as “Analyze/Act”): Using the information gathered, opportunities for performance increase are identified and prioritized.

In the first two steps (“Plan” and “Do”) the sense of what is important play the central role. In the last two steps (“Check” and “Act”) statistical methods and systems analysis techniques are most often used to help pinpoint statistical significance, dependencies and further areas for improvement.

2.4.2 The IDEAL improvement framework

The IDEAL framework [IDEAL 96] is an instantiation of the Deming-Cycle mentioned above. It provides a process improvement framework with five stages and 15 sub stages that can be applied when improving the testing process.

- **I**nitiating
  - Stimulus for improvement
  - Set context and establish sponsorship
  - Establish an improvement infrastructure
- **D**iagnosing
  - Appraise and characterize current practice
  - Develop recommendations and document phase results
- **E**stablishing
  - Set strategy and priorities
  - Establish a Test Process Group (see section 7.1.1)
  - Plan actions
- **A**cting
  - Define processes and measures
  - Plan and execute pilots
  - Plan, execute and track installation
- **L**earning
  - Document and analyze lessons
  - Revise organizational approach

2.4.3 Fundamental Concepts of Excellence

The Fundamental Concepts of Excellence, as described for example on [EFQM-Web] are used in organizational excellence models globally, to measure organizations against the nine criteria which are listed below. The European Foundation for Quality Management (EFQM) provides an example, and the reference section of this syllabus provides pointers to equivalent models used outside Europe.

Other organizational quality and excellence initiatives, such as Six Sigma and Balanced Score Card also provide a way of discussing the goals for an organization, deciding how to achieve those goals, and measuring whether they have been achieved.

The Fundamental Concepts of Excellence are:

- **Results Orientation:** “Excellence is dependent upon balancing and satisfying the needs of all relevant stakeholders (this includes the people employed, customers, suppliers and society in general as well as those with financial interests in the organization)”

- **Customer Focus:** “The customer is the final arbiter of product and service quality and customer loyalty, retention and market share gain are best optimized through a clear focus on the needs of current and potential customers.”

© International Software Testing Qualifications Board

- **Leadership & Constancy of Purpose:** "The behavior of an organization's leaders creates a clarity and unity of purpose within the organization and an environment in which the organization and its people can excel."

- **Management by Processes & Facts**: "Organizations perform more effectively when all inter-related activities are understood and systematically managed and decisions concerning current operations are planned. Improvements are made using reliable information that includes stakeholder perceptions."

- **People Development & Involvement:** "The full potential of an organization's people is best released through shared values and a culture of trust and empowerment, which encourages the involvement of everyone."

- **Continuous Learning, Innovation & Improvement**: "Organizational performance is maximized when it is based on the management and sharing of knowledge within a culture of continuous learning, innovation and improvement."

- **Partnership Development:** "An organization works more effectively when it has mutually beneficial relationships, built on trust, sharing of knowledge and integration, with its Partners."

- **Corporate Social Responsibility:** "The long-term interest of the organization and its people are best served by adopting an ethical approach and exceeding the expectations and regulations of the community at large."

2.5 Overview of improvement approaches

2.5.1 Overview of model based approaches

To improve product quality, the software industry has focused on improving their development processes. A guideline that has been widely used to improve the development processes is the Capability Maturity Model. The Capability Maturity Model (CMM), its successor the Capability Maturity Model-Integration (CMMI) and ISO 15504 are often regarded as the industry standard for system and software process improvement.

Despite the fact that testing can account for substantial parts of project costs, only limited attention is given to testing in the various software process improvement models such as the CMMI. As an answer, the testing community has created complementary improvement models.

The ISTQB Advanced syllabus identifies test process improvement as one of the key areas within the testing profession and identifies two principal model-based approaches:

- Process models define generic bodies of testing best practice and how to improve different aspects of testing in a prescribed step-by-step manner. Examples are the Test Process Improvement (TPI®) model and the Test Maturity Model integrated (TMMi). Both are described in section 3.3. Other less used models are mentioned in the ISTQB Advanced syllabus.
- Content models are non-prescriptive; they do not require that improvements occur in a specific order. Instead, they define specific activities which can benefit a test process if applied well. The Systematic Test and Evaluation Process (STEP) and the Critical Testing Process (CTP) are two principal examples of this approach. Both are described in section 3.4.

© International Software Testing Qualifications Board

2.5.2 Overview of analytical approaches

Analytical approaches typically involve the analysis of specific measures and metrics in order to assess the current situation in a test process, decide on what improvement steps to take and how to measure their impact. The Goal-Question-Metric (GQM) approach is a typical example of an analytical approach and is covered in section 4.3.

Chapter 4 covers analytical approaches in more detail.

2.5.3 Hybrid approaches

A hybrid approach can be applied in which projects which have already been developed to a higher level of process maturity, (using either model-based approaches, analytical approaches or a mix of both), are used to set and measure improvement goals for other projects. This is a common sense approach which ensures that practices that work within a particular organization can be transferred to other similar projects without needing to rely entirely on the predefined best practices of a standard process model (see section 3.3).

2.5.4 Other approaches to improving the test process

Improvements to the test process can be achieved by focusing on certain individual aspects described below. Note that these aspects are also covered within the context of the models mentioned in section 2.5.4.

2.5.4.1 Test process improvement by developing people's skills

Testing is often regarded as a profession and the test group is often recognized as a necessity.

Improvements to testing may be supported by providing increased understanding, knowledge and skills to people and teams who are carrying out tests, managing testing or making decisions based on testing. These may be testers, test managers, developers and other IT team members, other IT managers, users, customers, auditors and other stakeholders.

Increase in skills and competence may be provided by training, awareness sessions, mentoring, coaching, networking with peer groups, using knowledge management repositories, reading and other educational activities.

Skill levels may be associated with career paths and professional progression, for example, the SFIA (Skills Framework for the Information Age) [SFIA-Web].

Skills and competencies which need to be improved may be in testing, other IT technical skills, management skills, soft skills or domain skills. For example:

- Test knowledge, e.g. test principles, techniques, tools, etc.
- Software engineering knowledge, e.g. software, requirements, development tooling, etc.
- Domain knowledge, e.g. business process, user characteristics, etc.
- Soft Skills, e.g. communication, accurate way of working, reporting, etc.

Skills for test process improvers are covered further in section 7.3. However, the skills described are needed not just in the improvement team but across the entire test team, especially for senior testers and test managers.

The focus for improvement teams is in improvements brought about by:

- increasing awareness of the benefits and limits of test activities / test improvements to the development process and for the business.
- increasing knowledge and skill levels to support activities in the existing or improved test processes.

© International Software Testing Qualifications Board

- increasing competencies of individuals to enable them to carry out the activities.
- improving the correlation between increasing competence and rewards, recognition and career progression.
- motivation of staff.

2.5.4.2 Test process improvement by using tools

Test improvements may be gained by the successful introduction of tools. These may be efficiency improvements, effectiveness improvements, quality improvements or all of these. For example:

- Test management tools align working practices regarding the documentation of test cases and logging defects
- Code coverage tools support the implementation of exit criteria at component level

Testing tools are implemented with the intention of increasing test efficiency, increasing control over testing or increasing quality of deliverables. Implementation of testing tools is not trivial and the success of the implementation depends on the selected tool addressing the required improvement, and the implementation process for the tools being successful. These areas were covered in the ISTQB Foundation and Advanced syllabi.

When examining potential process improvements in any part of the test process and at all points in the software life cycle, the test process improvement organization should consider whether the introduction of tools will support improvement. The quality of test tools has matured during the past number of years. Their scope, diversity and areas of application have increased enormously. By analogy with CASE (Computer Aided Software Engineering), CAST (Computer Aided Software Testing) should be seen as a list of available test tools, classified according to application and platform. The use of such tools may often bring about a considerable improvement in the productivity of the test process.

The use of tools to aid in gathering, analyzing and reporting on data, including statistical analysis and process modeling tools aids the process improver. These are not (necessarily) testing tools.

The focus for improvement teams is improvements gained by:

- The selection and implementation of tools to aid the improvement team in its work, for example statistical and process modeling tools.
- The selection of tools that provide appropriate support for an identified improvement for example static analysis tools to assess code quality during developer-led static testing.
- Improvement of the tool selection and implementation process for example following the causal analysis for problems during a tool implementation pilot.

2.5.4.3 Test process improvement by using life cycle phase end reviews

The test closure phase is when a retrospective or lessons learned review should take place.

Use of continuous improvement cycles is central to many improvement methods. Both sequential and iterative lifecycles may include Post Implementation Reviews, Phase End Reviews, Lessons Learned meetings and other opportunities to feedback and implement improvements.

In iterative methods with short iterations (for example Agile methods) the feedback loops will happen more frequently, and therefore the opportunities to implement improvements are more frequent. For example, Agile development life cycle models such as SCRUM expect a continuous improvement loop as part of the normal project process, with a retrospective and improvement of processes (including the test process) at the end of each sprint.

In exploratory testing, each test session is followed by an assessment of where best to focus the testing next. This allows for an improvement cycle at the end of each session.

© International Software Testing Qualifications Board

In scripted/structured approaches to testing, the effort made to draw up the strategy/plan/scripts may mitigate against a desire to implement improvements during the test project. However, it is possible to undertake a lessons learned or other process review at more frequent intervals and use this to re-focus and improve testing. In particular when following a risk based approach, the risk based tests will need to be changed (improved) to address the new/changed risks as the business, product and project risks change during the life cycle of the system.

2.5.4.4 Test process improvement in different test approaches

In exploratory testing, each test session is followed by an assessment of where best to focus the testing next. This allows for an improvement cycle at the end of each session.

In scripted/structured approaches to testing, the effort made to draw up the strategy/plan/scripts may mitigate against a desire to implement improvements during the test project. However, it is possible to undertake a lessons learned or other process review at more frequent intervals and use this to re-focus and improve testing.

2.5.4.5 Test process improvement related to adoption of standards

Process improvement may be dictated by standards. For example, the requirements of domain-dictated standards like, for example from the Food and Drug Administration (FDA) of USA, may mean that specific improvements are required to enable work to be performed in that domain.

Compliance to standards may be required for legal, regulatory, or commercial reasons or for improving communication across team, organizations, or national borders. Standards may also be used to set or measure in-house processes and improvements against benchmarks set by other organizations.

Test process improvement organizations may bring about improvements by selection of appropriate standards, (refer to the Advanced Level syllabus for details), and specification of how the standard is to be used. The standard may be used, for example, for the following purposes:

- To achieve compliance and consequently to meet the requirements of an audit process
- As a measurement benchmark for comparison with other organizations
- As a source of ideas and examples to aid choices in improvements
- As a source of standardized practices which may provide better interoperability of systems and processes within a changing network of partner companies.

2.5.4.6 Test process improvement focused on specific resources

The management of the test environment, test data and other technical resources may be outside the test team's control. If these areas are seen as a focus for improvement, the teams controlling these resources will need to be engaged in the improvement process.

Processes required to set up and manage the test environment include:

- requirements definition for the environment
- design, build and verification/test of the environment
- acceptance of the environment
- deployment process
- capacity planning
- configuration and change management for the environment
- access control processes
- booking/scheduling environments within and between teams
- retirement/dismantling of environments.

Processes required to support the design, acquisition and management of test data include:

- test analysis and design activities
- test implementation activities

© International Software Testing Qualifications Board

- backup, restore and archive processes
- configuration and change management on specific data sets
- applicable data security procedures (possibly required by law)

Improvements required at an organizational level for resources such as environments and data may include requests for savings in costs, energy usage to meet the social responsibility and environmental targets for the organization. These may be addressed by efficiencies in the deployment and use of the environments and by, for example, the virtualization of environments.

Some process models (see section 3.3) explicitly include these resources in the assessment and recommendations, but if for example an analytical approach such as root cause analysis is used (see chapter 4), they can be added as factors to be considered.

The focus for improvement teams is improvements made by:

- identifying improvement areas outside the test team's control.
- engaging with the controlling teams, if necessary by escalation through management.
- engaging with improvement teams outside testing to coordinate improvements.
- identifying and implementing improvements within the test team and across test teams for provision and management of resources.

© International Software Testing Qualifications Board

3. Model-based improvement 495 mins.

Keywords:

CTP, CMMI, continuous representation, GQM, ISO 15504, maturity level, model-based approach, software process improvement, staged representation, STEP, TPI, TMMi, content-based model, process models

Learning objectives for model-based improvement

3.1 Introduction to model-based approaches

LO 3.1.1 (K2) Understand the attributes of a test process improvement model with essential generic attributes
LO 3.1.2 (K4) Compare the continuous and staged approaches including their strengths and weaknesses
LO 3.1.3 (K2) Summarize the assumptions made in using models in general
LO 3.1.4 (K2) Compare the specific advantages of using a model-based approach with their disadvantages

3.2 Software process improvement models

LO 3.2.1 (K2) Understand the aspects of the CMMI model with testing-specific relevance.
LO 3.2.2 (K2) Contrast the suitability of CMMI and ISO/IEC 15504-5 with models developed specifically for test process improvement.

3.3 Test process improvement models

LO 3.3.1 (K2) Summarize the background and structure of the TPI test process improvement model
LO 3.3.2 (K2) Summarize the key areas of the TPI test process improvement model
LO 3.3.3 (K2) Summarize the background and structure of the TMMi test process improvement models
LO 3.3.4 (K2) Summarize the TMMi level 2 process areas and goals
LO 3.3.5 (K2) Summarize the TMMi level 3 process areas and goals
LO 3.3.6 (K2) Summarize the relationship between TMMi and CMMI
LO 3.3.7 (K4) Differentiate between the TPI and TMMi model and make a recommendation for which is appropriate in a given scenario
LO 3.3.8 (K3) Carry out an informal assessment using the TPI test process improvement model
LO 3.3.9 (K3) Carry out an informal assessment using the TMMi test process improvement model

3.4 Content based models

LO 3.4.1 (K2) Summarize the background and structure of the STEP content based model
LO 3.4.2 (K2) Summarize the activities, work products and roles of the STEP model
LO 3.4.3 (K2) Summarize the CTP content based model
LO 3.4.4 (K2) Summarize the critical test processes within the CTP content based model
LO 3.4.5 (K2) Summarize the role of metrics within the CTP content based model
LO 3.4.6 (K2) Compare the use of metrics in a content based and an analytic approach (chapter 4)

3.1 Introduction to model-based approaches

3.1.1 Desirable characteristics of test process improvement models

Test process improvement models can be characterized by the following attributes.

© International Software Testing Qualifications Board

- Easy to use
- Continuing publicity
- Publicly available
- Available support by consultants
- Not a marketing vehicle of a commercial organization
- Accepted by professional bodies
- Include provision for improvement
- Provide many small evolutionary improvements
- Sound basis meaning it is practical, empirical, theoretical, published and justified
- Provides details of how to assess, identify improvements and make improvements
- Quantifiable improvements
- Tailorable (project-specific)
- The degree to which the model prescribes the improvement activities to be performed
- Support for the order of improvement
- Whether the improvement is represented in a staged or continuous way
- Level of detail on testing content
- Richness and variety of suggested solutions to specific testing problems
- Level of formal accreditation required for assessors
- Certification possible for an organization

3.1.2 Continuous and Staged Representations

Process models show process maturity using either a staged or a continuous representation.

The staged representation offers a systematic “one step at a time” approach to improvement. The model architecture prescribes the stages that an organization must proceed through so that its test process improves in an orderly fashion. Achieving a stage ensures that an adequate level of process maturity is established (in TMMi, this is called a Maturity Level) before moving up to the next stage. The focus of improvement is on achieving the individual levels of capability for a predefined set of process areas (e.g. Test Planning and Test Environment in TMMi at level 2) which are allocated to a Maturity Level (e.g. TMMi level 4). A maturity level represents a well defined evolutionary plateau towards achieving improved organizational processes.

The advantages of a staged model lie mostly in the simplicity of its concept. It provides a maturity-level rating that is often used in external management communication and in qualifying companies (e.g. a customer company may require that all potential supplier companies achieve a minimum process maturity of, say, TMMi level 4). The problem with staged representation is its lack of flexibility. An organization may achieve relatively high levels of capability in many of the required process areas, but still fail to achieve an overall maturity level. A tendency to use this as an “all or nothing” approach or a “once and done” approach can result in not achieving the desired business goals.

Within the continuous representation there are no prescribed maturity levels which the development process is required to proceed through. The TPI model uses a continuous representation. An organization applying the continuous representation can select specific areas for improvement from several different categories according to the particular goals it wishes to achieve. Continuous representation permits individual capability levels to be achieved for each key/process area (e.g. in the TPI Model the maturity of the key area “Test Strategy” can be achieved at four increasing levels).
The advantages of the continuous representation lie mostly in its flexibility.

- An organization may choose to improve a single trouble-spot or group of process/key areas which are closely aligned to the organization’s business objectives.
- It is possible to improve different process areas at different rates.
- The “all or nothing” disadvantages of the staged model are avoided

© International Software Testing Qualifications Board

3.1.3 Assumptions in using models

There are a number of assumptions which are implicitly made when applying a model-based improvement:

- Models describe what their authors consider to be “best practice”. This term may actually be better described as “good practice” which has been shown to provide benefit for improving test processes. It is the responsibility of the test process improver to judge what “best” means for a particular project.
- The use of a model-based approach assumes that conforming to the “best practices” is required in order to improve.
- Models assume that there is such a thing as a „standard“ project and/or organization. Because all projects are in some way different, model-based approaches will not always apply equally well to all projects.

Using models cannot be considered to be a purely mechanical checklist-based procedure; experience and judgment must be properly applied to obtain maximum benefit. Generic models require interpretation in order to take specific project factors into account.

The following examples illustrate some of the factors which may require interpretation of the model:

- Life cycle applied (e.g. traditional V-model or Agile development process)
- Technology used (e.g. web, object oriented technology)
- System architecture (e.g. distributed systems, SOA, embedded systems)
- Risk level (e.g. for safety critical systems compared to business systems)
- Test approach (e.g. scripted versus exploratory)

There can be a number of problems with using models in general (i.e. not just the process improvement models considered in this syllabus). These include:

- Lack of knowledge by the originator of the model
- The model used may oversimplify causes and effects
- The model may be applied in an inappropriate context
- A model may be adopted without considering whether it could be incorrect to do this
- The model is applied without considering if this is useful. It is considered a goal in itself instead as an instrument for improving the process.
- Belief by the user that modeling implicitly means understanding.

3.2 Software process improvement models

SPI models typically only cover testing issues in general terms. For this reason test improvement models, like those covered in section 3.3 of this syllabus, have been developed specifically for the test process.

3.2.1 CMMI

The CMMI can be implemented via two approaches or representations: the staged representation or the continuous one. In the staged representation there are five “levels of maturity”, each level building upon the process areas established in the previous levels. In the continuous representation the organization is allowed to concentrate its improvement efforts on its own primary areas of need without regard to other areas.

The staged representation provides more focus for the organizations and has the largest uptake in the industry. It also ensure commonality with CMM and can be used to objectively measure organizations maturity level, while the continuous representation is generally considered to be more flexible.

© International Software Testing Qualifications Board

Within CMMI the process areas of Validation and Verification specifically reference both static and dynamic testing test process. However also the process areas Technical Solution and Product Integration deal with testing issues.

In addition to the testing related process areas a number of process areas can be distinguished that provide support towards a more structured testing process:

- Project planning
- Project monitoring and control
- Risk management
- Configuration Management
- Measurement and Analysis
- Causal Analysis and Resolution.

Though the relationship between software development and testing is addressed in the CMMI model, dedicated test process models like CTP, STEP, TMMi, and TPI provide more detail regarding testing and the test process.

The relationship between CMMI and testing is made more explicit within the TMMi model [TMMi-Foundation-Web].

3.2.2 ISO/IEC 15504

ISO/IEC 15504-5 is an international software process improvement standard that defines a number of process categories, including Engineering, Management, Organization, Customer-Supplier and Support. The “Support” process category (SUP) includes individual processes which are relevant to testing, including Verification and Validation. The capability level for each process is evaluated using a predefined set of process attributes and applies a continuous representation approach (see section 3.1.2).

3.3 Test process improvement models

An introduction overview of the various types of process improvement models is provided in section 2.5.1. In this section and in section 3.4 the major improvement models that are applied in practice are described in more detail.

3.3.1 The Test Process Improvement Model (TPI®)

The ISTQB Certified Tester Advanced Syllabus describes the TPI model and states learning objectives which are relevant for the test manager.

The TPI model is a process model which uses a continuous representation. Its approach is to distinguish all major aspects of a test process. Accordingly central elements of the TPI model, as outlined in [Koomen / Pol 99], are twenty key areas, each of which covers a specific aspect of the test process, such as test strategy, metrics, test tools and test environment.

A thorough analysis of key areas is supported by various maturity levels per key area, each defined by specific checkpoints. Dependencies between key areas reduce the risk of “one-sided” improvement (e.g. it is not possible to achieve high maturity levels in the metrics key area before certain levels of maturity have been achieved for the for the incident management and reporting key areas).

By means of a maturity matrix which covers all key areas, findings are summarized and visualized. When the outcome of the analysis is consolidated across all key areas a maturity level can be attributed to the whole test process. These maturity levels are named initial, controlled, efficient and optimizing.

© International Software Testing Qualifications Board

Numerous prioritized improvement suggestions reflecting best practice support the definition of a suitable development path. The definition of improvement objectives and its implementation can be tailored according to needs and capacity of the test organization.

The generic approach makes TPI independent of any SPI model. It covers both the test engineering aspects as well as support for managerial decision making.

The TPI model has been adapted for specific industries. An example of this is “Automotive TPI” which defines an extra key area (“integration”) and has been adopted by German car manufacturers.

Mapping exists between TPI and the software process improvement models CMMI and ISO/IEC 15504 mentioned in section 3.2.

3.3.2 The Testing Maturity Model Integration (TMMi)

The ISTQB Certified Tester Advanced Syllabus describes the TMMi model and states learning objectives which are relevant for the test manager.

TMMi [TMMi-Foundation-Web] has a staged architecture for process improvement. It defines the following levels or maturity levels through which an organization passes as its testing process evolves from an ad-hoc status:

- initial
- managed
- defined
- management and measured
- optimized

The five maturity levels in the TMMi prescribe a maturity hierarchy and an evolutionary path to test process improvement. Achieving each stage ensures that an adequate improvement has been laid as a foundation for the next stage. The internal structure of the TMMi is rich in testing practices that can be learned and applied in a systematic way to support a quality testing process that improves in incremental steps.

Each maturity level in TMMi has a set of process areas that an organization needs to focus on to achieve maturity at that level. For example,. the process areas at TMMi level 2 “Managed” are:

- Test Policy and Strategy
- Test Planning
- Test Monitoring and Control
- Test Design and Execution
- Test Environment

The structure of the TMMi is largely based on the structure of the CMMI. This is a major benefit because many people/organizations are already familiar with the CMMI structure. The CMMI structure makes a clear distinction between practices that are required (goals) or recommended (specific practices, typical work products, etc.) to implement. This aspect is also included in the TMMi.

3.3.3 Comparing TPI to TMMi

This section aims at briefly comparing the two test process improvement models covered in this syllabus. Depending on the specific project context and the improvement objectives to be followed, one of the two models could be preferred.

© International Software Testing Qualifications Board

Some differences between TMMi and TPI are shown in the table below:

Aspect	TPI	TMMi
Type	Continuous model – Bottom-up oriented	Staged model – Top-down oriented
Test methods	Uses generic TMap practices as terms of reference.	Test method independent
Terminology	Based on TMap	Based on ISTQB glossary
SPI	No formal relationship to a SPI model but mapping possible	Highly correlated to CMMI
Focus	20 key areas with test specific focus. Close-up view per key area, overview across the entire test process.	Detailed focus on limited number of process area per maturity level. Also focused on other testing issues such testability reviews, quality control, defect prevention and test measurement program.
Approach	Thorough, test engineering approach	Strong focus on management commitment

3.4 Content-based models

Content-based models allow testing processes to be improved by providing a structured description of good testing practices together with an overall approach to be followed. The content-based models described in this syllabus are STEP and CTP.

3.4.1 STEP

STEP [Craig02] does not require that improvements occur in a specific order. In some cases the STEP assessment model is blended with the TPI model.

The STEP methodology is based upon the idea that testing is a lifecycle activity that begins during requirements formulation and continues until retirement of the system.

The Advanced Syllabus provides details of the following:

- basic premises for the methodology
- examples of the quantitative metrics taken
- examples of qualitative factors

3.4.2 CTP

The basic premise of the Critical Testing Process [Black03] assessment model is that certain testing processes are critical. These critical processes, if performed well, will support successful test teams. The model identifies twelve critical testing processes.

A CTP assessment identifies which processes are strong and which are weak, and provides prioritized recommendations for improvement based on organizational needs. A number of quantitative and qualitative metrics are commonly examined during a CTP assessment.

Once an assessment has identified weak areas, plans for improvement are put into place. Generic improvement plans are provided by the model for each of the critical testing processes, but the assessment team is expect to tailor those heavily.

© International Software Testing Qualifications Board

4. Analytical-based improvement 510 mins.

Keywords:

analytical approach, causal analysis, cause-effect diagrams, cause-effect graph, Defect Detection Percentage, Failure Mode and Effect Analysis, Fault Tree Analysis, IEEE 1044, indicator, inspection, measure, metric, Pareto analysis

Learning objectives for analytical-based improvement

4.2 Causal Analysis

LO 4.2.1 (K2) Understand causal analysis using cause/effect diagrams
LO 4.2.2 (K2) Understand causal analysis during an inspection process
LO 4.2.3 (K2) Understand the use of standard anomaly classification for causal analysis
LO 4.2.4 (K2) Compare the causal analysis methods
LO 4.2.5 (K3) Apply a causal analysis method on a given problem description
LO 4.2.6 (K4) Identify and select test process improvement based on the results of a causal analysis
LO 4.2.7 (K3) Apply an approach to select defects for causal analysis

4.3 The GQM Approach

LO 4.3.1 (K2) Describe the Goal-Question-Metric (GQM) approach
LO 4.3.2 (K3) Apply the Goal-Question-Metric (GQM) approach to derive appropriate metrics from a testing improvement goal.
LO 4.3.3 (K3) Define metrics for a testing improvement goal
LO 4.3.4 (K2) Understand the steps and challenges of the data collection phase
LO 4.3.5 (K2) Understand the steps and challenges of the interpretation phase

4.4 Analysis using Measures, Metrics and Indicators

LO 4.4.1 (K2) Provide examples of the various categories of metrics and how they can be used in a test improvement context
LO 4.4.7 (K4) Recommend appropriate metrics and indicators for tracking improvement trends in a particular improvement situation.

4.1 Introduction

Analytical approaches are problem-based. Improvements to be introduced are based on actual problems and objectives rather than the generic model of ”best practice“ used in model-based approaches and content-based approached like those described in chapter 3.

Applying an analytical approach to improvement involves the analysis of a test process in order to identify problem areas and set project-specific goals. The definition and measurement of key parameters is required to evaluate whether improvement measures have succeeded.

Analytical approaches may be used together with a content-based approach to verify results and provide diversity, e.g. as is the case with Critical Test Processes. Also model-based approaches sometimes address analytical approaches as one or more separate key areas with the model, e.g. as is the case with TMMi.

Generally speaking, analytical approaches may also be applied to the task of test management. Test managers apply them at a project level, test process improvers apply them at the process level.

© International Software Testing Qualifications Board

4.2 Causal analysis

Causal analysis is the study of problems to identify their possible root causes. This allows identification of solutions which will remove the causes of problems and not just address the immediately obvious symptoms. If causal analysis is not used, attempts to improve test processes may fail because the same or similar problems may recur.

Many software process improvement models emphasize the use of causal analysis as a means of continually improving the maturity of the software process.

The following systematic methods for causal analysis are described below as examples:

- Cause-Effect diagrams (Ishikawa fishbone diagrams)
- Causal analysis during an inspection process
- Use of standard anomaly classifications

Note: other methods are available for causal analysis (see Advanced syllabus) and also checklists of common causes may be used as an input to the causal analysis, for example when carrying out causal analysis on defects.

4.2.1 Cause-Effect diagrams

Cause-Effect diagrams (also known as Ishikawa fishbone diagrams) were developed for the manufacturing and other industries [Ishikawa 91], [Juran] and has been adopted in the IT industry. These diagrams provide a mechanism to discuss and identify root causes under a number of headings.

The steps to apply are described by [Robson 95]:

1. Write the effect on the right hand side of the diagram
2. Draw in the ribs and label them. Labelling of the fishbone diagram may be dependent on the context for a particular group. Control of the fishbone is with the work group rather than with management.
3. Remind the group of the rules of brainstorming:
    - no criticism – all ideas are accepted at this stage
    - freewheel – random, crazy ideas are welcome as are ideas that build on other people's ideas
    - quantity of ideas – a large number of ideas is to be generated
    - record all ideas – including random, crazy and repeating ideas
    - don't evaluate the ideas now – having brainstormed, have a rest before evaluating the ideas
4. Use the brainstorming method - possible causes are brainstormed and added to the diagram. For each first level cause, checklists are used to identify the underlying root causes, which may be on a different part of the diagram.
5. Incubate the ideas for a period of time
6. Analyze the diagram to look for clusters of causes and symptoms. Use the Pareto idea (80% of the gain from 20% of the effort) to identify clusters that are candidates to solve.

Cause Effect diagrams may also be used to work from the effects back to the causes [BS7925-2] and [Copeland 03].

© International Software Testing Qualifications Board

4.2.2 Causal analysis during an inspection process

The inspection process is described in the ISTQB Foundations syllabus and expanded in the ISTQB Advanced syllabus. Using the software inspection process [Gilb and Graham] suggests a different approach to causal analysis.

The inspection meeting is a facilitated discussion which lasts two hours and follows a set timescale and format.

- The selection of defects to be considered may be taken by the inspection leader or in the context of a project retrospective. Alternatively the issues to be discussed may identified in the first part of the analysis.
- Part 1: Defect analysis (90 minutes) looks at specific defects and their specific causes
- Part 2: Generic analysis (30 minutes) focuses on identifying broad trends in the defects. The team looks for trends, commonalities and what has recently gone well, improved or gone wrong. Defect taxonomies (introduced in the Advanced Level syllabus) can be a valuable support this analysis. Generic analysis may also focus on particular defects found in dynamic testing to provide risk-based inputs for the future testing strategy.
- The discussion must ensure that the number and severity of defects considered attempts to maximize the return on the invested time.

In the defect analysis, each defect is categorized with

- A defect description: This is not the defect symptom but the defect itself
- A cause category: E.g. communication, oversight, education, transcription error, process
- A cause description: For the cause and any chain of cause events
- The process step when the defect was created: This is not always where it was detected
- Suggested actions to eliminate the cause: These must be specific and achievable.

4.2.3 Use of standard anomaly classifications

Standards such as [IEEE 1044] allow a common classification of anomalies allowing an understanding of the project stages when defects are introduced, the project activities when faults are detected, the cost of rectifying the faults, the cost of failures, and the stage where the defect was raised versus where it was found (Defect Leakage).

A common classification allows statistics about improvement areas to be analyzed across an organization. This classification must be introduced via training, tools and support so that any one using the incident management system understands when to use the classifications and how to interpret them. This information may be used in the improvement of the development and test process to identify areas where improvement will be cost effective and to track the success of improvement initiatives. The defects to be analyzed will be recorded from all life-cycle phases, including maintenance and operation.

4.2.4 Selecting defects for causal analysis

The application of methods described above may require the selection of specific defects for analysis from a potentially large collection. The following approaches can be taken in combination to selecting defects for analysis:

- Pareto analysis (20% of defects considered representative of all)
- Outliers in statistics (e.g. using values of statistical variation, such a 3-sigma, on a particular metric)
- Project retrospective meetings
- Use of defect severity categories

© International Software Testing Qualifications Board

4.3 The GQM Approach

The Goal-Question-Metric GQM approach [Basili-Papers], [Trienekens and van Veenendaal 97] applies the following steps to define suitable metrics:

- setting specific goals
- deriving questions that when answered will provide information about the achievement of the goals
- deriving metrics which, when measured, will provide the answers to the questions

Basili's approach with GQM is to provide a measuring mechanism for feedback and evaluation. GQM allows measurement to be:

- focused on setting specific goals.
- applied to all part of the software life-cycle products, process and resources.
- interpreted based on characterization and understanding of the organizational context, environment and goals.

Thus, the measurements appropriate in one organization may not be appropriate in another organization. The definition of the goals and questions allows the appropriate metrics to be selected or defined, and hence the appropriate data collected and analyzed.

The three levels in the GQM approach are:

1. Conceptual level – the GOALS for the organization, with regard to the quality of products, processes and resources (note: resources include people, offices, hardware, software) – this might include any of the definitions of quality from chapter 2, so productivity as well as manufacturing excellence could be a goal. Forms can be used to define goals.
2. Operational level – the QUESTIONS which characterize products, processes and resources with respect to their quality – Basili refers to these as the objects of measurement.
3. Quantitative level – the METRICS which may be objective (quantitative, factual) or subjective (qualitative, viewpoints).

Goals may give rise to one or more questions, and questions may give rise to one or more metrics. The goals, questions and metrics may be associated with the quality views described in chapter 2, and/or with the results measurement required by organizational and IT frameworks for improvement described in chapter 2.

4.4 Analysis using Measures, Metrics and Indicators

4.4.1 Introduction

Measures, metrics and indicators form part of all improvement programs. This applies regardless of whether these improvements are carried out formally or informally. It is also regardless of whether the data are quantitative or qualitative, objective or subjective. The feelings of the people affected by the improvement are a valid measure of progress with improvement.

Measures, metrics and indicators initially help to target areas and opportunities for improvement. They are required continuously in improvement initiatives in order to control the improvement process and to make sure that the changes have resulted in the desired improvements.

Measures, metrics and indicators can be collected at all stages of the software life cycle, including development, maintenance and use in production [Nance & Arthur 02]. They are also used for deriving other metrics and indicators. Note that for all items mentioned in section 4.4.2 which relate to defects it

© International Software Testing Qualifications Board

is important to make a distinction between the various priority and severity levels of the defects found. Specific measures, metrics and indicators may also be applied by test managers, in particular for the project level tasks of test estimation and for progress monitoring and control. Test process improvers will apply measures, metrics and indicators at the process level.

4.4.2 Test effectiveness metrics

4.4.2.1 Defect Detection Percentage (DDP)

The first indicator that can be used in almost any test improvement process, and is highly recommended by most test experts, is the Defect Detection Percentage (DDP). If you've found most (if not all) of the important defects during testing, and users/customers found few during real-life usage, your testing is good.

Defect Detection Percentage is defined as number defects found by testing divided by total known defects. The DDP can be calculated per test stage (e.g. integration, alpha testing, beta testing) or for the all test stages together. DDP is a calculable metric after a project is finished and some time (e.g. three or six months) has passed in which residual defects may be found.

4.4.2.2 Post-release Defect Rate

This indicator is defined as the number of defects found by customers during a certain period after release of the product per Kilo Lines Of Code (KLOC). If this rate decreases then the customers' perceived quality will increase.

Note: these defect metrics give information for the "manufacturing" view of quality (see chapter 2).

4.4.3 Test efficiency / cost metrics

4.4.3.1 Organizational cost of quality

To sell the idea of test improvement to others we need to show the benefit for them. To do this it is important to show the cost benefit of test activities and of improvement to testing, by measuring the cost of testing and improvement as well as the cost of not testing (i.e. the cost of failure).

4.4.3.2 Cost of Quality Ratio

Ratio of 'total effort spent on static testing (e.g. inspections and reviews)' and 'total effort spent on dynamic testing (e.g. unit, integration, system test)'. If this ratio increases then the efficiency of the defect removal process will increase.

4.4.3.3 Early Defect Detection

As stated at the previous performance indicator, the efficiency of the defect removal process will increase if defects are found earlier in the process. Not only is this applicable for static versus dynamic testing but also for unit and integration versus system and acceptance testing. The performance indicator 'Early Defect Detection' measures the total number of defects found during unit and integration testing (early dynamic testing) versus the total number of defects found during dynamic testing.

4.4.3.4 Relative Test Effort

A basic indicator is the ratio of the total test effort (or cost) versus the total project effort (or cost) pre-release. Post-release effort spent or activities such as maintenance and after-sales support is not taken into account.

4.4.3.5 Test Efficiency

This indicator is based on the number of defects found compared to the testing effort spent (per test level). The indicator is used to determine if the test effort is focused on the finding of high severity defects. Test efficiency may also be related to the size of the test assignment, such as test points.

© International Software Testing Qualifications Board

4.4.3.6 Automation Level
The ratio of the number of test cases executed automatically versus the total number of test cases executed (both manually and automatically).

4.4.3.7 Test Productivity
The total number of test cases (or test design) for the product related to the total test effort required. Of course this performance indicator can also be measured per test phase.

Note: these efficiency/cost metrics give information for the "value" view of quality (see chapter 2).

4.4.4 Lead-time metrics

4.4.4.1 Test Execution lead-time
The lead-time of testing is especially important during test execution, since test execution is on the critical-path of the project. Lead-time is defined as the period (in days or weeks) between two milestones that identify the start and end of one or more project activities. The test execution lead-time indicator to test the product should of course be related to the size of the product. This indicator can be measured per test phase, e.g. alpha or beta.

4.4.5 Predictability metrics

4.4.5.1 Test Execution Lead-time slippage
The difference between the actual and estimated test execution lead-time required for one or more test phases related to the estimated test execution lead-time.

For improvement purposes it is often interesting to measure test lead-time slippage against the estimation made at the beginning of the project and against the estimation made at the beginning of the test execution phase.

4.4.5.2 Effort slip (or cost)
The difference between the actual and estimated effort (or cost) required for one or more test phases related to the estimated effort (or cost):

4.4.5.3 Test Case slip
The difference between the actual and estimated number of test cases (or test designs) required for one or more test phases related to the estimated number of test cases (or test designs).

4.4.6 Product quality metrics

4.4.6.1 Metrics for quality attributes
A number of attributes are available with which product quality can be described. (e.g. functionality, reliability, usability, efficiency, maintainability, portability). These are documented in [ISO 9126] and its successor [ISO 25000]. The attributes and the indicators associated with them are described in the ISTQB Advanced Level syllabus. For example, indicators associated with the software quality attribute reliability may take the form of Mean Time Between Failures (MTBF) and Mean Time to Repair (MTTR).

The test process is one of the primary sources of information for measuring these software quality attributes. The capability of the test process to deliver meaningful and relevant product quality information may be considered an area for potential test process improvement .

4.4.6.2 Coverage indicators
The coverage of requirements and code achieved by testing may be used as indicators of product quality (assuming that higher product quality is related to higher coverage levels) during testing.

© International Software Testing Qualifications Board

Requirements coverage is defined as the number of requirements tested compared to the total number of requirements that has been defined. This can be refined by making a distinction between the number of requirements tested and the number of requirements tested *and* passed. If coverage increase testing is becoming better and it is expected that also product quality will increase.

Code coverage is defined as the percentage of the total software code that is executed during testing. Various levels of code coverage are described in the ISTQB Foundations syllabus.

These give information for the “product” view of quality (see chapter 2).

4.4.7 Test maturity metrics

These metrics represent the organizations’’ test maturity level in the terms used by models such as the Test Maturity Model (TMMi) or the Test Process Improvement (TPI) model. If the maturity increases then the risk of not satisfying test objectives regarding quality, lead-time and costs will decrease. Refer to the ISTQB Advanced Level syllabus for further details.

Please note that these metrics address the manufacturing, product and value quality views described in the chapter 2, but are not direct measures of the user view of quality. Test managers may therefore want to measure user quality by taking specific measures of user / customer satisfaction, measuring usability characteristics, (especially relating to task efficiency and effectiveness), or by qualitative measures of stakeholder views.

© International Software Testing Qualifications Board

5. Selecting the approach for test process improvement 105 mins.

Keywords:

(none)

Learning objectives

5.1 Selecting test process improvement approaches

LO 5.1.1 (K2) Summarize reasons for best applying a test process improvement approach
LO 5.1.2 (K5) Make a recommendation for a choice of approach(es) in a specific scenario and for a given improvement scope

5.1 Selecting test process improvement approaches

The choice of approach depends on:

- taking account of the critical success factors described in chapter 9.
- taking account of the general guidelines listed below.

The following lists are provided as a support to the decision process. They should not be taken as a list of mandatory requirements or unbreakable rules. Regarding test improvement models, it may also be helpful to consider the list of general model characteristics described in section 3.1.1 when making choices.

Process models (e.g. TMMi, TPI) are best applied when:

- a test process already exists (although they can also be useful for establishing test processes from new).
- comparisons or benchmarking is required between similar projects.
- compatibility with software process improvement models is required.
- company policy is to attain specific maturity levels (e.g. TMMi Level 3).
- a well-defined starting point with a predefined path of improvement is desired.
- a measure of test maturity is needed; e.g., for marketing purposes.
- process models are respected and accepted in the organization.

Content models (e.g. CTP, STEP) are best applied when:

- a test process needs to be established.
- an assessment to identify costs and risks associated with the current test process is needed.
- improvements do not need to be implemented in the order specified by TMMi or TPI, but rather in the order determined by business needs.
- tailoring is required to ensure the test process fits the company's specific context.
- discontinuous, rapid improvements and changes to the existing test process are desired or needed.

Analytic approaches are best applied when:

- specific problems need to be targeted.
- measures and metrics are available or can be established and taken.
- continuous, incremental improvement of the existing test process is desirable.

© International Software Testing Qualifications Board

- evidence for the need of a test process is required.
- agreement is needed about the reasons for change.
- the root cause of the problem is not necessarily in the control or influence of the test process owner.
- when a small scale evaluation or improvement pilot is required/budgeted for.
- when a pilot is required to see whether a larger scale investigation or improvement program is needed.
- to trial hypotheses and gather evidence about the causes, symptoms and effects of problems and of proposed improvements.
- the organization culture values/trusts internally developed analysis based on local evidence above externally built models (reference or content).
- stakeholders from many areas are to be involved personally in the analysis (for example in brainstorming sessions).
- the team has control of the analysis and the improvement process.

Mixed approaches may also be used, such as the use of analytical approaches within a process model or content model, for example:

- use of causal analysis during a TMMi test improvement program.
- use of metrics during a STEP test improvement program.

© International Software Testing Qualifications Board

6. Process for improvement 765 mins.

Keywords:

acting, assessment, assessment report, corporate dashboard, diagnosing, establishing, IDEAL, initiating, learning, scorecard, test improvement plan, test policy, ,

Learning objectives for the process of improvement

Note that individual skills shall be applied in achieving the learning objectives of this chapter. These skill-related learning objectives are covered in chapter 7. Information about change management (chapter 8) and key success factors (chapter 9) will also be needed.

6.1 Introduction

LO 6.1.1 (K2) Summarize a test policy
LO 6.1.2 (K3) Define a test improvement policy

6.2 Initiating the improvement process

LO 6.2.1 (K2) Summarize the activities of the Initiating phase of the IDEAL improvement framework
LO 6.2.2 (K4) Analyze business goals (e.g. using corporate dashboards or scorecards) in order to derive appropriate testing goals.
LO 6.2.4 (K3) Define an improvement strategy (including the scope of the test process improvement) for a given scenario

6.3 Diagnosing the current situation

LO 6.3.1 (K2) Summarize the activities of the Diagnosing phase of the IDEAL improvement framework
LO 6.3.2 (K3) Plan and perform assessment interviews using a particular process or content model in which an awareness of interview style and inter-personal skills are demonstrated
LO 6.3.3 (K3) Create and present a summary of the conclusions (based on an analysis of the findings) and findings from an assessment.
LO 6.3.4 (K2) Summarize the approach to solution analysis
LO 6.3.5 (K5) Formulate and document test process improvement recommendations on the basis of assessment results and the analysis performed

6.4 Establishing a test improvement plan

LO 6.4.1 (K2) Summarize the activities of the Establishing phase of the IDEAL improvement framework
LO 6.4.2 (K3) Prioritize recommendations using a given criteria list.
LO 6.4.3 (K2) Compare top-down and bottom-up improvement approaches.
LO 6.4.4 (K2) Summarize the typical contents of a test improvement plan.
LO 6.4.5 (K4) Create a test improvement plan.

6.5 Acting to implement improvement

LO 6.5.1 (K2) Summarize the activities of the Acting phase of the IDEAL improvement framework
LO 6.5.2 (K4) Select an appropriate pilot from a list of possibilities

6.6 Learning from the improvement program

LO 6.6.1 (K2) Summarize the activities of the Learning phase of the IDEAL improvement framework
LO 6.6.1 (K2) Understand the activities involved in learning from improvements

© International Software Testing Qualifications Board

6.1 Introduction

Test process improvement should be a stated objective within an organization's testing policy. (refer to the ISTQB Advanced syllabus for more details on Test Policy). An organizations test process improvement policy should be based on the overall test policy. Effective test process improvement requires a systematic process. In section 2.4 the generic improvement process was introduced by describing the Deming cycle and the IDEAL process improvement framework. As an example, in this chapter, the process for improvement is covered in more detail using the IDEAL improvement framework as a basis [IDEAL 96] . This approach can be applied to any life-cycle model. Each section described below relates to one of the principal IDEAL activities:

- Initiating
- Diagnosing
- Establishing
- Acting
- Learning

6.2 Initiating the improvement process

Initiating process improvement is perhaps the most important step in the process of test process improvement. Actions taken at this initial stage directly influence the final results of the improvement process. Poorly initialized improvement processes may deliver unsatisfactory results and significantly reduce the chances of being able to take any future improvement initiatives.

The IDEAL model describes the following high-level activities at the "Initiating" phase:

- Stimulus for improvement
- Set context and establish sponsorship
- Establish an improvement infrastructure (i.e. organization)

Based on the high-level activities of the IDEAL model, the following considerations must be taken into account at this phase:

- The actual need for improvement must be established
- Objectives need to be defined and aligned to the business needs
- The scope of improvement must be established
- An improvement strategy must be chosen
- The influence of people and culture must be taken into account

6.2.1 Establishing the Need for Improvement

As a first step in establishing the need for improvement, some awareness should be generated about process improvement. The more obvious needs for improvements to test processes arise from major software failures and several such failures have been attributed to poor test processes. There are, however, a wide range of different motivations for change. These may originate from one or more of the following stakeholders:

- Management /customer (e.g. more effective testing, less trouble in production)
- User (e.g. better usability)
- Developer (e.g. better support for defect analysis)
- Tester (e.g. establish more systematic testing)
- Maintenance (e.g. less time needed to test software changes)

Areas for improvement may be captured and defined based on preliminary analysis and interviews with stakeholders (refer to section 7.3 for required skills of interviewing and analysis). As part of the

© International Software Testing Qualifications Board

analysis it may be necessary to determine current indicators such as the total cost of quality, based on the total cost of failures in production and the total cost of testing (see section 4.4.3.1 Organizational cost of quality and section 6.2.2).

6.2.2 Setting objectives for test improvement

The main objectives of a test process improvement must always be established in relation to quality, cost, time, and business value. Setting objectives for test improvement requires three principal steps:

- establish a general vision for the (improved) future
- set specific objectives.
- aligning Testing Improvement to business goals

The objectives for test improvement are typically documented in the organization's test policy.

6.2.2.1 Establish a general vision for the future

Test process improvement must focus on the benefits required by a sponsoring stakeholder and a vision of the overall objectives to be achieved.

Improvement initiatives need such a vision of the future because:

- sponsors need to be convinced of the return on investment before committing resources.
- the management of any necessary changes must relate to agreed objectives.

Failure to define a common vision may result in failure of the proposed test improvements (see chapter 9)

- Poorly defined objectives may hide unresolved conflicts of interest
- We may focus on inappropriate or unachievable objectives. The result may be a waste of resources or a failure to improve.

6.2.2.2 Setting specific objectives

Objectives are needed for any test process improvement. These enable:

- the appropriate actions to be taken.
- the success (or failure) of the improvement efforts to be defined.

A number of possibilities are available to enable objectives to be represented:

- Qualitative objectives, perhaps supported by appropriate scales (e.g. from "very bad" to "very good" or "getting better" to "getting worse") or questionnaires.
- Quantitative objectives with metrics. For example, the Goal, Question, Metric (GQM) method (see section 4.3) enables metrics to be defined, and can be used to link objectives (goals) with measurable results.
- Objectives expressed as maturity levels. If test process improvement is to be conducted using a process model, objectives may also be represented in a form appropriate for the model in question. This typically involves defining specific levels of maturity to be achieved either for the test process as a whole or individual aspects of the test process.

6.2.2.3 Aligning Testing Improvements to the Organization

Test process improvements must be aligned to

- business goals.
- any software process improvements being performed.
- organizational improvements (see chapter 2).
- organizational structure.

Corporate dashboards and scorecards can be used by the organization to allow test improvements to be aligned to organizational improvement targets; such as:

© International Software Testing Qualifications Board

- Financial targets: for example, productivity improvements, improved revenue, improved profit, improved turnover, also aligned to the value view of quality.
- Improved product quality aligned to the product, manufacturing and user view of quality.
- Customer targets: for example improved market share, improved customer satisfaction, improved risk management process, also aligned to the user view of quality.
- Internal targets: for example greater predictability of project outcomes; reduced faults/failures during the software development; reduced project elapsed time; reduced effort/costs aligned to the manufacturing view of quality. Offshoring or outsourcing may also be considered as internal targets as this can be especially effective at reducing the cost of the testing process and enabling businesses to focus on their core areas of competence.
- Innovation and improvement targets: new marketplaces/industries, increased number of new products to market; speed to market; process/framework/standards accreditation (e.g. CMMI or industry standard) aligned to the value view of quality.
- People targets: for example job satisfaction, staff turnover, sickness and other absence reduced which may align to any of the quality views but will also affect the transcendent view of quality (trust, reputation).
- Social involvement/political targets: for example environmental impact of the organization; reputation; publicity which may align to any of the quality views but will also affect the transcendent view of quality (trust, reputation).

6.2.3 Setting the Scope for Improvement

The issues which must be addressed in order to set the scope of test process improvement include:

- General process scope: other processes which are in scope other than the test process.
- Test process scope: the parts of the test process to be addressed.
- Test levels; which test levels are within the scope of the improvement programme
- Project scope: are project(s) or the organization in scope?

6.2.3.1 General process scope

The scope for improvement may involve aspects of the software development process in general, such as project management, requirements management and configuration management. It is important to appreciate whether the stated improvement goals can actually be achieved by improving the test process or whether other processes (e.g. service management, development, supporting processes described in [ITIL] must also be improved. If this is the case, further resources and expertise may be required.

6.2.3.2 Test process scope

The focused improvement of individual areas of a testing process may be more cost-effective than considering all possible areas. It may be desired to improve all aspects of the test process on a broad front, or to only address specific aspects (e.g. test planning).

If the scope of test improvement is limited to specific aspects, care should be taken to consider all other dependencies. Improvement limited to only specific aspects may lead to sub optimization. Does it make sense, for example, to focus improvement efforts on establishing a testing metrics program when they are unused (e.g. not used in test reporting)?

6.2.3.3 Organizational Scope

A test improvement program may be organized in a program/project-centric or organization-centric manner.

Program/project-centric improvements are focused on an individual project or group of projects (i.e. program). Assessments are generally performed with project testing staff (e.g. testers, test managers). The scope of test process improvement considered may be limited to a relatively small set of process-related tasks, such as test planning or testing techniques. Such highly-focused

© International Software Testing Qualifications Board

improvement programs may be particularly cost-effective provided that the test process scope is carefully chosen (see section 6.2.3.2 above).

Program/project-centric improvements may produce relatively quick results but may fail to address problems at an organizational level. Improvements at an organizational level may be longer lasting and more widely beneficial, but generally take longer to achieve and cost more to implement.

Organization-centric improvements are focused on a testing organization, department or group. In addition to the assessment of individual projects the organization as a whole is in focus. Aspects of the test process which apply to all projects are especially in focus (e.g. training, organization).

In the IDEAL-model's "Initializing" phase a process improvement infrastructure is established [IDEAL 96]. This considers the structural organization of the test improvement program. Section 7.1 considers these issues in further detail.

6.2.4 Influence of people and culture on the initializing phase

A number of people-related factors can have an influence on achieving improvement objectives. At the initialization phase the following critical success factors are significant:

- Level of knowledge.
- Organizational culture (including off-shore issues)
- People culture
- Level of acceptance

Chapters 7, 8 and 9 have additional information concerning the soft skills needed by the improvement team and the affect of human factors on change management and organizational culture.

Recommendations for test Improvement should be sensitive to people issues and be able to suggest alternative strategies depending on the styles, culture and needs of the people in the organization.

6.3 Diagnosing the current situation

The IDEAL model describes the following high-level activities for the "Diagnosing" phase :

- Appraise and characterize current practice
- Develop recommendations and document phase results

The end result of this phase is typically a test assessment report.

Based on the high-level activities of the IDEAL model, the following considerations must be taken into account in this phase:

- Planning the assessment
- Preparing for the assessment
- Performing interviews
- Giving feedback
- Analyzing results
- Performing solution analysis
- Recommending improvement actions

The activities performed in this phase depend on the approach to be taken to test process improvement (see chapter 5).

© International Software Testing Qualifications Board

If an analytical-based approach is to be adopted (see chapter 4) the various causal analysis techniques (section 4.2) may be applied and metrics, measures and indicators (section 4.4) analyzed.

If a model-based approach is to be used (see chapter 3) then an assessment will be planned and performed. Sections 6.3.1 to 6.3.3 cover these aspects in more detail.

6.3.1 Planning the assessment

An Assessment Plan clearly identifies the following activities and schedules them:

- Assessment preparation
  - Preliminary analysis
  - Preparation of interview materials, such as checklists
  - Gather existing testware material (e.g. test plans, test specifications)
- Interviews with different roles involved in the test process:
  - Tester
  - Test manager
  - Developer
  - Project manager
  - Business owners
  - Business analysts (domain experts)
  - Specialists such as environment manager, defect manager, release manager, automation specialists
- Specific areas to be covered in each interview.
- Providing Initial feedback after completion of the assessment (dates, formats, expectations)
- Briefing the interviewee on the next steps to be taken

The Assessment Plan must ensure that all testing areas are covered according to the agreed objectives (section 6.2.2) and scope (6.2.3) agreed on in the initializing phase. For specific areas it may be advisable to cover the same subject in different interviews to enable consistency checks to be conducted.

6.3.2 Assessment preparation

A preliminary analysis of documents is often performed before interviews are carried out. If the assessment scope is at the organizational level then test policy, test process description, available templates and master test plans may be in focus. If the scope is at project/program level, then test plans, test specifications and test reports may be selected. Documents from other stakeholders (development, business etc.) may also be analyzed if they are relevant to testing.

The purpose of the analysis is:

- to gain insights into the current testing process prior to interviewing those involved.
- to prepare particular questions for the interview
- to perform formal elements of the assessment which do not require discussion. For example, the documents may be checked to ensure completeness and conformity to standards.

Prior to performing interviews it is recommended that a suitable environment should be made available. This includes:

- a comfortable setting
- freedom from disturbances
- privacy

© International Software Testing Qualifications Board

6.3.3 Performing interviews

Interviews are performed according to Assessment Plan (see section 6.3.1). The content of interviews is generally guided by the model being used (see chapter 3).

Interviews are mostly conducted on an individual basis to allow the interviewee freedom to express views and to ensure that a "secure" environment exists. The interviewer should always discuss the importance of confidentiality for interviewees and information providers prior to an assessment taking place, particularly if sensitive issues are expected. For this reason it is recommended that a person is not interviewed in the presence of their superior.

People who give information may be motivated to be honest if

- confidentiality is ensured.
- recognition for improvement ideas is given.
- no fear of punishment or failure exists.
- they know and understand how the information they provide will be used.

A wide range of skills are required for the conduct of a successful interview. These are described in more detail in section 7.3.1.

6.3.4 Initial feedback

Shortly after completion of interviews an initial feedback may be given, typically as a short presentation or walkthrough. This is particularly helpful in confirming initial assessment findings with the interviewee, for clarifying any misunderstandings, or for providing an overview of main points to stakeholders. Care should be taken to maintain the rules of confidentiality at all times and to avoid assigning blame for any problems identified.

6.3.5 Analysis of results

If an analytical approach to improvement is being used (chapter 4), the current situation may be analyzed by applying concepts such as

- Systems Thinking [Weinberg 92]
- Tipping Points [Gladwell].

Systems Thinking helps analyze the relationships between different system (process) components and to represent those relationships as stable ("balancing") loops or reinforcing loops. A reinforcing loop may have a negative effect ("vicious circle") or a positive effect ("virtuous circle").

Tipping Points help identify specific points in a system where a small, well-focused improvement may break a vicious circle and set off a chain reaction of further improvements.

If following a model-based approach (chapter 3) for test process improvement, a comparison is made between the process maturity of the current situation and the desired objectives defined at the initializing phase.

Where benchmarks are available, these should be used in evaluating results. The following benchmarks may be used, where available:

- Company-based, representing an entire organization
- Industry-based, where possible relating to the same business segment
- Project-based, where the comparison is made with a specific project regarded as meeting the desired objectives.

© International Software Testing Qualifications Board

Where key performance indicators have been established (see section 4.4 and section 6.2.2), these should be incorporated into the analysis. For example, if the Defect Density Percentage (DDP) has fallen below the required level, an analysis of faults found in production should be performed to evaluate their sources.

The result of the evaluation should provide sufficient information with which to define recommendations and support the planning process (see sections 6.3.7 and 6.4 below).

6.3.6 Performing solution analysis

Solution analysis is used to identify potential solutions to problems and then to choose between those solutions. Any chosen improvement(s) or solution(s) may be decided in a number of ways:

- **Pre-conception** of the right improvement solution – we have already decided what the solution is to the problem e.g. "let's automate everything". This means that a solution analysis stage is not carried out with the disadvantage that the preconceived solution may not address the root cause of the problem and may even make it worse.

- **Recommended solutions** may be built into the model used for the assessment as key practices/areas with the advantage of external evidence for a practice's usefulness being available. The disadvantage here is that the "next practice to adopt" in the model may not address the root cause of the problem in this circumstance and may even make it worse.

- **Requirements** from a customer or stakeholder, for example a customer may demand that all its suppliers are ISO9001:2008 accredited, with the advantage that the goal and focus are very clear but the disadvantage that the requested change in processes may not provide an improvement or may conflict with other planned improvements.

- Based on a **solution analysis** of information gathered about the problems, with the advantage that both the negative and positive consequences of each proposed solution are discussed, and consequently solutions selected which are positively beneficial and which have minimal negative side effects, but the disadvantage that the analysis process takes time, resources and money.

- A **tailored** method of selecting a solution, based on a mix of the above e.g. a stakeholder requests adoption of TMMi level 4 and a cost-benefit analysis is carried out resulting in a decision to aim for a partial adoption of TMMi level 3. The advantage of this is that the solution analysis is focused, and the disadvantage is that the analysis process takes time, resources and money.

The solution analysis process includes one or more of the following, depending on the method chosen:

- Prioritizing problems and root causes in order to choose which solutions will be developed.
- Identifying cost advantages and other benefits, including the risks of NOT implementing the solution.
- Identifying costs, risks and negative affects of implementing the solution.
- Identifying any constraints on implementing the solutions.
- Identifying conflicts, such as solutions which negate each other or are otherwise incompatible.
- Analysis of feedback loops (both virtuous loops and vicious circles).
- Assessing and prioritizing the solutions.
- Performing gap analysis on information and metrics gathered during earlier activities.
- Conducting cost-benefit analysis to provide estimate of return on investment.

© International Software Testing Qualifications Board

- Constructing “Reversed Fishbone” diagrams, where the fishbone used in Root Cause Analysis are reversed, and solutions brainstormed against the same fishbone headings for a specific root cause identified earlier. Additional headings for constraints may be added, for example budget, resource, time constraints.

6.3.7 Recommending improvement actions

An assessment report must relate results to the specified test improvement objectives. The report must be delivered as soon as possible after completion of the assessment, possibly as a preliminary version followed later by a complete version.

As a minimum, the assessment report must include:

- a management summary.
- a statement of scope and objectives.
- results of the analysis.
  - positive aspects
  - aspects which need improving
  - open issues
- a list of improvement recommendations.

Where possible, recommendations should be thought of as improvement requirements (e.g. “provide tool support for a defect tracking system”) which may be implemented in a number of different ways (e.g. “use tool XYZ and provide training”). The implementation task is covered in section 6.5.

An improvement recommendation should include the following information:

- A unique identifier (for traceability).
- The impact of the recommendation on one or more stated objectives (where possible using a scale to indicate the degree of fulfillment, such as “minimal”, “partial”, “full” or a percentage value).
- Estimate of cost and benefits.
- Implementation timescale (e.g. short-term, medium-term, long-term).
- Risks of implementation (e.g. high level of resistance expected).
- Dependencies and assumptions (e.g. assumes that another recommendation is also implemented).

To assist in the planning and tracking of high-level recommendations they should, where possible, be broken down into small steps of improvement with tangible results.

Some process models, such as TPI, include specific improvement suggestions to help in the task of creating recommendations.

6.4 Establishing a Test Improvement Plan

The IDEAL model describes the following high-level activities for the “Establishing” phase:

- Set priorities
- Develop approach
- Plan actions

The end result of this phase is typically a test improvement plan.

Based on the high-level activities of the IDEAL model, the following considerations must be taken into account in this phase:

© International Software Testing Qualifications Board

- Setting priorities
- Develop an implementation approach
- Planning the improvements

6.4.1 Setting priorities

The recommendations from the ”establishing” phase are prioritized according to a list of criteria, each of which may be weighted according to the needs of the improvement and the stakeholders involved.

At least the following criteria should be considered:

- Duration of improvement:
  A balance needs to achieved between short-term and long-term improvements. Short-term improvements (“Quick-Wins”) have the advantage of quickly showing return on investment and may have a strong motivational role on the implementing team. Long-term improvements may address some of the fundamental improvements in the testing process, including cultural and organizational issues.
- Implementation risk:
  Many improvements require a change to existing testing practices. There is a risk of failure associated with each of those improvements. The following factors will need to be considered:
  - Ability to return to an existing state if the improvement has to be abandoned.
  - Overall impact on the entire improvement program of “key” improvements, especially if other improvements are dependent on the success of this particular measure.
  - Ability to actually implement the improvement. Are sufficient resources available? Are key members of the improvement team likely to allocated to other tasks? Can risks be identified within the change process (see chapter 8), such as resistance to particular changes?
  - Cost/benefit of the proposed improvement (possibly expressed as a value of “Return on Investment”).
- Link to objectives:
  Can a clear association be made between the proposed improvement and the stated objectives of the business?
- Leverage:
  How much impact will this improvement have on specific objectives (e.g. high, medium, low)?

6.4.2 Developing an implementation approach

Recommendations for improvement are considered and prioritized by the stakeholder. A specific approach for implementing the change can then be selected which are closely related to the scope of improvement determined in the initializing phase. The two principal approaches are:

- top-down
- bottom-up

Features of the top-down approach are:

- The scope of improvement typically covers several projects or an entire organization.
- Ownership of the improvement process may be with a dedicated team.
- Detailed analysis of results is required in order to find commonalities (good and bad practices) between the different projects.
- Presentation and negotiation skills are particularly relevant in achieving consensus on objectives and recommendations.
- The following critical success factors are particularly relevant (see chapter 9 for details):
  - Management style
  - Obtaining sponsorship

© International Software Testing Qualifications Board

- Managing expectations

Features of the bottom-up approach are:

- The scope of improvement typically covers no more than one or two projects.
- The selected approach is often less formal. For example, individual projects may choose an analytical approach rather than applying a more formal model-based approach such as TPI..
- Ownership of the improvement process is typically within a project team.
- A typical objective is the prototyping of particular improvements in order to gain experience and build support.
- This approach may be adopted where funding for a test improvement program is limited and costs / benefits first need to be demonstrated
- The results of the bottom-up strategy may be used for a subsequent top-down roll-out of proven improvement measures.

6.4.3 Planning the improvements

The primary planning activities are:

- Agreeing on specific measures based on the recommendations.
- Establishing the performance indicators required (see chapters 2, 4 and 6).
- Prioritizing and combining groups of related improvements into packages (step-wise improvement).
- Setting up an improvements database to link the improvement issues to the recommendations and required performance indicators for achievement.
- Deciding on an approach for process improvement (see section 6.4.2).
- Scheduling the changes.
- Establishing groups or teams to implement the improvements (see section 7.1.1).
- Assigning tasks.
- Documenting the above-mentioned points in a test improvement plan.

The IDEAL model describes two possible action plans:

- A strategic action plan
  - has a timescale of three to five years.
  - covers the entire organization's test process improvements and integrates them with other software process improvement activities and any other total quality management (TQM) initiatives already planned or in process.
- A tactical action plan
  - typically has a short timescale (e.g. one year).
  - focuses on the detailed planning of activities of the Test Process Group (the team responsible for implementing the improvements). See section 7.1.1 for further details.

6.5 Acting to implement improvement

The IDEAL model describes the following high-level activities for the "Acting" phase:

- Create solution
- Pilot/Test solution
- Refine solution
- Implement solution

Based on the high-level activities of the IDEAL model, at least the following considerations must be taken into account in this phase:

© International Software Testing Qualifications Board

- Selecting and executing pilots
- Manage and control the change

6.5.1 Selecting and executing a pilot

Piloting a proposed improvement is an effective way of reducing the risk of failure, gain experience, build support and reduce the risk of implementation failure. This is especially important where those improvements involve major changes to working practices or place a heavy demand on resources.

Selection of a pilot should balance the following factors:

- Realism:
  Is the pilot representative of the "real world"? Care should be taken not to select a pilot which is unrealistic just because it offers the chance of a particularly quick or easy implementation.
- Scalability of solution:
  Can the results from the pilot be used all contexts? If the pilot is not representative of the complexity and size of real projects there is a risk that the implemented improvement will not scale.
- Impact on current projects:
  Pilots should not be performed on current projects unless the impact is acceptable. Particular care is required if existing practices are to be replaced by the improved practices for the duration of the pilot. A better solution is to run the new practices in parallel with the existing practices, although this may create a resource problem (we cannot expect project employees to perform the same task twice because of the pilot).
- Risk of failure:
  Even though the use of pilots is a risk-reduction measure, the risk of pilot failure must also be evaluated. The above-mentioned aspects are significant factors in the evaluation, which should consider both the financial and motivation risks of failure.

6.5.2 Manage and control the implementation

The implementation of the Test Improvement Plan is performed, monitored and progress towards achieving improvement goals is reported. The measures, metrics and indicators specified in the Test Improvement Plan are collected and compared to the established objectives.

If the analysis of lessons learned from performing a pilot yields positive results, the decision may be taken to roll-out the improvements to other parts of the organization and/or other projects. The roll-out of improvements follows a defined process, especially where an entire organization is affected by the change. Chapter 8 is devoted entirely to this critical aspect of test process improvement.

6.6 Learning from the improvement program

The IDEAL model describes the following high-level activities for the "Learning" phase:

- Analyze and validate
- Propose future solutions

After implementation of the test improvement plan a retrospective is performed with the stakeholders and the achieved objectives analyzed. The organization or person which has implemented the improvement usually manages the retrospectives, which are typically performed as workshops.

Depending on the results of this review further actions may be defined, which may include performing a new improvement cycle, e.g. initiating the diagnosing phase (section 6.2).

© International Software Testing Qualifications Board

7. Organization, Roles and Skills 375 mins.

Keywords:

assessor, codependant behavior, emotional intelligence, lead-assessor, Mind-Map, Test Process Group (TPG), test process improver, transactional analysis

Learning objectives for roles and skills for the improvement team

Training providers should consider the skill-based learning objectives (section 7.3) together with the learning objectives for chapter 6.

7.1 Organization

LO 7.1.1 (K2) Understand the roles, tasks and responsibilities of a Test Process Group within a test improvement program
LO 7.1.2 (K2) Compare the different organizational structures to organize a test improvement program
LO 7.1.3 (K2) Understand the impact of outsourcing or off-shoring of development activities on the organization of a test process improvement program
LO 7.1.4 (K4) Define an organizational structure for a given scope of a test process improvement program

7.2 Individual roles

LO 7.2.1 (K2) Understand the individual roles in a test process improvement program

7.3 Skills

LO 7.3.1 (K2) Understand the skills necessary to perform an assessment
LO 7.3.2 (K4) Evaluate test professionals (e.g. potential members of a Test Process Group / Technical Working Group) and identify their principal soft skill deficits to perform an assessment.
LO 7.3.3 (K3) Apply interviewing skills, listening skills and note-taking skills during an assessment, e.g. when performing interviews during “Diagnosing the current situation”
LO 7.3.4 (K3) Apply analytical skills during an assessment, e.g. when analyzing the results during “Diagnosing the current situation”
LO 7.3.5 (K2) Understand presentational and reporting skills during a test process improvement program
LO 7.3.6 (K2) Understand persuasion skills during a test process improvement program

7.1 Organization

Test process improvement should be organized to ensure correct implementation and “ownership” of the improvement process (see section 9.1).

Useful information regarding organization is provided by [Burnstein 03] and [IDEAL 96].

7.1.1 The Test Process Group

[Burnstein 03] describes a Test Process Group (TPG) which may cooperate with other quality-related organizations, such as a Software Engineering Process Group (SEPG).

© International Software Testing Qualifications Board

A TPG should be established as a permanent part of an organization and not just formed on an ad-hoc basis. This is principally because:

- The scope of ad-hoc improvement groups is often limited to a particular project or an individual problem area. A permanent TPG has a wider scope and can more easily recognize and propose improvement suggestions affecting the organization.
- The implementation of proposed improvement measures can be more affectively controlled by a permanent TPG. With ad-hoc improvement groups there is a risk that they disband before improvements are fully implemented. This absence of adequate control may result in failure to achieve or report on agreed objectives.
- The many and varied skills which are needed for performing effective test process improvement are best developed within a specialized group.
- A permanent TPG can function as the test process “owner” and provides an important channel of communication to principal stakeholders.
- Many maturity models (e.g. TMMi) include the permanent TPG as an indicator of higher test process maturity.

[Burnstein 03] mentions that an effective TPG is likely to include respected people from a wide range of backgrounds (e.g. management, development, testing) and who can be considered as practitioners. This not only provides valuable insights into the complexities of testing processes, but also increases the level of acceptance for any improvement suggestions proposed (respected practitioners are generally better accepted than pure theorists).

The IDEAL-model [IDEAL 96] describes the establishment of a process improvement organization with the following entities:

- Executive Council (EC)
  - Used in very large organizations
  - Deals with issues of strategy and direction
- Management Steering Group (MSG)
  - High-level managers from the organization’s existing management structure.
  - Sets goals, success criteria and priorities
  - Guides implementation activities in the organization
  - Supplies the resources
  - Sets up TWGs for specific aspects of process improvement
- Technical Working Group (TWG)
  - Exists only for the time needed to accomplish their specific goals
  - Research problem and propose solutions to the MSG
  - Perform prototyping activities
  - Revise tactical action plan with lessons learned from prototype

The IDEAL-model includes sample charters for each of these organizational entities.

The test process improvement organization is primarily concerned with process, but should also assume responsibilities for training; permanent process improvement can only be achieved when the people involved also improve continuously. The organizations’ goals, membership, roles, responsibilities and principal interfaces with other parts of the organization should be stated clearly and aligned with the Test Policy (see ISTQB Advanced Level syllabus).

7.1.2 Test improvement with remote, off-shore and outsourced teams

When teams involved in improvement are working in the same organization and on the same site, all part of the improvement process are easier to organize and carry out than when part of the team affected work in other organizations or on other sites.

© International Software Testing Qualifications Board

Where any part of the SDLC is out-sourced or off-shored, ease of communication of problems, root causes, solutions, evidence and plans for change are likely to be affected. The focus for improvement teams is on:

- Any political, cultural or contractual (mis)understandings that may need improvement, may block or encourage proposed improvements, or which may mean improvements are not understood or may affect sensitivities
- Timing of communications with regard to order that people are informed of change, but also local holidays, time zones and so on
- Informing and involving all parties in all stages from initiation of the improvement, through information gathering, choice of solutions, pilots and the rollout/change program
- Eliciting feedback from all parties on the success or otherwise of the improvement program.

One potential problem is where the on-shore test team has a different level of process maturity, efficiency and effectiveness compared with the off-shore/outsource test team. The expectations of the groups are widely different for process, communication and quality culture. Assumed processes may not be aligned. Any attempt at improvement or change on either side may cause cultural and communication clashes.

7.2 Individual Roles

7.2.1 The Test Process Improver

The test process improver must be capable of performing tasks in accordance with this syllabus.

There is a limit to the change that a test process improver can achieve. If, for example, the development managers or the customer control their own testing, this may be outside the influence of the test improver and necessary changes to their testing processes may be out of scope to the test process improver.

The test process improver should understand the scope of what they can improve and request need support from management and other groups as required. In performing their role, they may be limited to suggesting test process improvements rather than actually implementing them.

7.2.2 Assessor roles

Due to the many technical and soft skills required to perform assessments (see section 7.3), it is recommended that a specific assessor role is assigned.

If a model-based approach is adopted, assessments should be conducted by a person with specific knowledge of the model to be used. In some cases (e.g. TMMi, CMMI, ISO 15504, EFQM Assessor) this person may require formal training and certification. For example, the TMMi Foundation distinguishes between lead-assessors and assessors (see [TMMi-Foundation-Web] for more information on their roles and the requirements to become an accredited assessor).

7.3 Skills of the Test Process Improver

The success of conducting test process improvement is dependent on a wide range of skills. These may be technical skills, such as analysis, or non-technical, people-oriented skills, often referred to as „soft“ skills. The following skills are relevant:

- Interviewing skills
- Listening skills
- Presentational and reporting skills

© International Software Testing Qualifications Board

- Analytical skills
- Skills of persuasion
- Management skills
- Note-taking skills

All of these skills are required by any person performing an assessment. Particular skills may also be required for any member of a Test Process Group / Technical Working Group.

7.3.1 Interviewing skills

Interviewing can be represented by the following process:

- Opening — Clarify goals and targets
- Listening — Gather information (see section 7.3.1.4).
- Summarizing — Performed regularly on coherent parts of information
- Checking — Deepen he level of understanding between the interviewer and the interviewee by asking new questions based on the summary made so far.
- Closure — Explain next steps

Interviewing skills are essential for obtaining information and performing successful assessments. Good interview style is practiced by

- asking “open” questions which do not invite a “yes/no” answer.
- not reading out a questionnaire word for word.
- conducting a discussion in which the interviewer uses interpersonal skills (such as those mentioned in this section) to guide the conversation through the points to be covered.

Interactions between interviewer and interviewee are often complex and, without the necessary skills, can result in misunderstandings, the withholding of information or even the capture of incorrect or false information. Interviewers do not need to be psychologists, but they do need good interpersonal skills which come from an appreciation of concepts such as:

- Emotional intelligence [Mayer 04]
- Transactional analysis [Wagner 91]
- Co-dependant behavior [Copeland Paper 01]

7.3.1.1 Emotional intelligence

Emotional intelligence (EI) can “help one to make sense of and navigate the social environment [Mayer 04]. The ability-based model proposed by Salovey & Mayer proposes that individuals vary in their ability to process information of an emotional nature. For test process improvers this ability may be important when performing interviews, where factual information is often just part of the information being communicated.

The model proposes that EI includes 4 types of abilities:

- Perceiving emotions: The ability to detect and interpret emotions in an interviewee’s face and voice and also to identify one’s own emotions.
- Using emotions: The ability to harness emotions to enable thinking and analysis of problems. The emotionally intelligent test process improver can capitalize on changes of mood during an interview in order to obtain specific information or views from the interviewee.
- Understanding emotions: The ability to appreciate complicated relationships among emotions and how they might develop over time.
- Managing emotions: The ability to regulate emotions in both ourselves and in others. The emotionally intelligent test process improver can manage emotions to achieve the objectives of the interview.

© International Software Testing Qualifications Board

Measurement of EI using the ability-based model is possible, but this is not a skill expected of test process improvers.

7.3.1.2 Transactional analysis

The idea of transactional analysis within an organization or business context as described by [Wagner 91] is that each person is made up of six “inner people”. These are divided into three which are classified as “effective” and three which are “ineffective”.

Communication with “effective inner people” is generally considered positive and constructive.

- The Natural Child acts spontaneously, expresses feelings, and has need for recognition, structure and stimulation
- The Adult is logical and reasonable; it deals in facts rather than feelings
- The Nurturing Parent is firm with others, but also understanding, sensitive and caring

Communication with “ineffective inner people” is generally considered to be unhelpful for obtaining information.

- The Critical Parent uses body language, gesture and tone of voice to “tell others off” perhaps by sarcasm, pointing the finger, or raised voice.
- The Rebellious Child gets angry and stays angry, is very negative, does not listen, may deliberately forget things or procrastinate
- The Compliant Child blames itself, uses a soft voice, whines, is very careful and self protective.

Test process improvers with an understanding of transactional analysis should be able to improve their own interviewing skills and also to be able to distinguish informative from suspect information given by interviewees, suggesting rewording of questions or (in more extreme examples) revision of the interview approach.

7.3.1.3 Codependent behavior

The term "codependency" describes an undesirable pattern of human interaction in which one person’s shortcomings or weaknesses are compensated for by another person. In the words of Lee Copeland [Copeland Paper 01] “We do all the wrong things for all the right reasons." We have an unhealthy focus on the needs of others and may even start to assume the responsibility or “cover up” for the behavior of others.

An appreciation of codependency is important in the improvement of testing processes because these unhealthy codependent interactions can mask the true reasons for particular problems or could even be the problem themselves.

Lee Copeland provides some typical examples of codependency in the area of software development:

- Developers agree to implement systems without having a clear understanding user needs.
- If testers are given absurd schedules for their work by management they “do their best” by testing as much as possible in the time available.

In these examples a pattern exists of inappropriate behaviors (by development and management) and codependent responses. The short-term effect of these responses may be beneficial, but the long-term consequences may be damaging because the wrong messages are being sent out; “we don’t care about users” , or “our testing estimates are always inflated”.

Test process improvers should be aware of some typical indicators of codependence when conducting interviews:

- Responses which include “I do my best” (even though I know this is wrong)
- Responses which include “Never mind” (this needs correction, but I’ll pretend it’s OK)

© International Software Testing Qualifications Board

- Denial of risk (this could result in disaster, but I could suffer if I mention it)
- Responses where the interviewee tries to convince the interviewer that clearly incorrect behaviors are in some way “normal”

In the long-term, the codependent person may become the victim, feeling anger that they are constantly having to accept situations they know are wrong. A sense of resignation may set in as they begin tolerating abnormal, unhealthy, and inappropriate behaviors.

Test process improvers should appreciate that software testing professionals, of course, want to be helpful. At the same time they should carefully compare the short-term benefits of codependent behavior with the long-term difficulties that may arise. Improvement suggestions may need to focus on the long-term issues and on helping the codependent person out of their situation.

7.3.1.4 Listening skills

Listening skills are useful for extracting the information from what is being said and for preparing possible replies. “Active listening” is a technique which focuses on the person being spoken to and provides a structured approach for listening and responding to them. [Atwater 81]

7.3.2 Presentation and reporting skills

Presentation and reporting skills are important for

- obtaining „buy-in“ to test process improvement.
- clearly showing results to stakeholders.
- suggesting specific improvements.

Skills in presenting management summaries help to focus on key points at the right level of abstraction and without too many non-essential details.

- Be selective and choose only a few key ideas
- Be specific about how your ideas will work in your (or your customer’s) context
- Be realistic with timescales for improvements
- Talk the language of the managers
- Anticipate questions

Test process improvers are aware of their audience when presenting and reporting information. The views of quality presented in section 2.2. provide guidance on the type and depth of information presented. For example, management sponsors typically take a “value” view of quality and should therefore be presented with high-level information (e.g. improvement suggestions) which impacts their business.

Specific presentation and/or reporting skills (see [Tufte 90] and [Tufte 97]) include

- information design
- the ability to select appropriate media for the audience
- understanding of the appropriate use of evidence from metrics, statistics,
- understanding of the proper use of diagrams, charts and graphics
- speaking in public
- eliciting feedback from the audience and awareness of audience response.

7.3.3 Analytical skills

Chapter 4 describes a number of techniques relating to analytical improvement approaches. The general analysis skills required to apply these techniques are summarized below.

- Summary skills for gathered information

© International Software Testing Qualifications Board

- Ability to identify trends and patterns in information
- How to transform information into other formats e.g. text to process flow diagram, Mind-Map to presentation etc.
- Knowledge of when statistical analysis is appropriate and when not (see [Huff 93])
- Understanding and differentiating between: (1) cause and effect, (2) correlation and (3) coincidence when analyzing and reporting data
- If used, a proper understanding of the application and analysis of statistical information

7.3.4 Skills of persuasion

These skills are important where key stakeholders may need to be convinced of a particular improvement and when a future vision needs to be established. This may be the case, for example, where there is some initial resistance to change or where the person simply has limited time in which to understand issues and make decisions.

A useful technique which may be applied is described in [Frank 90]. This simple technique consists of the following steps:

- Set your objective
- Select your audience (unless this is obvious)
- Choose an approach
- Use a hook to get attention
- Know your subject
- Ask for your objective (or a next step towards achieving it)

Persuasion techniques are also part of sales and marketing techniques, as described by [Cialdini] and warned against by [Bernstein 03].

7.3.5 Management skills

A wide range of management skills are relevant to specific test improvement tasks but are beyond the scope of this syllabus. These include, for example,

- Planning
- Estimating
- Decision-taking
- Risk management

7.3.6 Note-taking skills

These skills permit the effective capture of relevant information. This may be particularly important, for example, when conducting interviews.

Mind-Maps [Buzan 95] are a natural and easy to acquire technique for preparing and controlling interviews, establishing connections between individual topics and for note-taking. They use principles of association between visual elements („doodles“) and text to provide the following benefits:

- The “big picture” can be captured effectively
- Note taking is quicker and better organized
- Presentation is easier to summarize
- The association of content to your ideas is easier to make
- It is easier to recall main points

Note that using Mind-Maps should remain the personal choice of the interviewer. Some interviewers may be inhibited during the interview by the need for premature analysis when constructing Mind-Maps. Alternatives include

© International Software Testing Qualifications Board

- Using natural text to preserve the actual interview responses
- Using short-hand forms of text such as key words
- Using diagrams and flowcharts

When note-taking the use of a laptop is sometimes considered a barrier between interviewer and interviewee which may even defeat the objective of the listening and interviewing skills. The decision to takes notes electronically should therefore be taken with care and in agreement with the interviewee.

Some interviewers record their interviewees by audio or video, and transcribe. This can be useful, but must only be done with the permission of the interviewee, and with agreement about how the recorded information may be used, and when / how the recording will be destroyed if necessary.

In some situations, such as group workshops, it may be better to collect and display the notes openly using, for example, flip charts, post-it notes or marked cards. When everyone in the room can see the notes and can contribute to them it is easier to reach agreement and obtain buy-in on the notes taken.

© International Software Testing Qualifications Board

8. Managing change 240 mins.

Keywords:

change management

Learning objectives for managing change

8.2 Fundamental change management process

LO 8.2.1 (K2) Summarize the fundamental change management process
LO 8.2.2 (K3) Address change management as part of the overall test improvement plan, with appropriate steps and actions

8.3 Human factors in change management process

LO 8.3.1 (K2) Summarize the role of human factors in the change management process
LO 8.3.2 (K4) Analyze people's attitude to change and relate them to the stage of grief
LO 8.3.3 (K4) Identify measure to create acceptance of the changes by the people involved

8.1 Introduction

Process improvement will not be successful without change management; the bulk of the investment in improvement is typically in deployment. In this chapter the process of managing change is in presented as a series of steps and activities.

8.2 Fundamental change management process

The changes required for a process improvement will almost certainly fail unless they are performed in the context of a change management process. The eight step process for change described in [Kotter & Rathgeber 05] can be applied to any IT-discipline, including test process improvement.

Set the stage

Step 1. Create a sense of urgency

- Establish the need for improvement (see section 6.2.1) preferably expressed by objective measurements and supported by a statement of risk (i.e. the risk of not implementing the proposed changes).
- Make it clear what changes will occur in what sequence and give broad timescales.
- Obtain visible management support and resources.

Step 2. Pull together the guiding team (e.g. Test Process Group, section 7.1.1).

- Engage natural early adopters as champions of change.
- Establish these people in the role of "multiplier" (i.e. a first level support person who passes on knowledge to others and motivates them).

Decide what to do

Step 3. Develop the change vision and strategy

- Manage expectations (clear objectives, what is in scope and what is not - "we are not changing the world").
- Establish a strategy (see section 6.2.4).

Make it happen

© International Software Testing Qualifications Board

Step 4. Communicate for understanding and buy-in

- Provide information (presentations, road-shows, newsletters etc) relating to
  - how measures that are being taken align with the goals set for the organization (strategy, policy, challenge, etc.).
  - how the measure will benefit employees and help them to improve their working.
  - previous successes and failures, indicating what will be different this time.
- Prototype ideas (initially, a bottom-up improvement strategy using a low-risk project may be supportive of prototyping).
- Motivate all those affected by the change. (See [Maslow] for further details of the "Hierarchy of Needs").

Step 5. Empower others to act

- Provide management support to enable change to take place (e.g. to help remove obstacles to change).
- Provide a feedback mechanism for those affected by the change. This may need to be anonymous depending on cultural issues.

Step 6. Produce short-term wins

- Go for quick wins – publish and reward them - gain momentum and motivate.
- Prioritize any quick wins that can reasonably be expected to stay in place and won't need to be undone in the near future in deference to broader considerations.

Step 7. Don't let up

- Ensure that top-down improvement strategies are supported by a separate improvement team, (e.g. the Test Process Group, section 7.1.1), which owns the improvement process and ensures that agreed changes are implemented according to the improvement plan.

Make it stick

Step 8. Create a new culture

- Roll-Out gradually using incremental steps and, where possible, avoid "big-bang" changes
- Determine whether the introduced changes have resulted in improvement. Consider each success as an opportunity to build on what went right and identify what can be improved further.
- Publicize the achievement of objectives ("do good things and talk about them").This can be done on the basis of quantitative metrics or using a qualitative scale (e.g. based on questions like " Are things getting better?').
- If any objectives have not been reached, provide analysis and evidence for the reasons and learn from them.
- Ensure that management support is available if problems occur with adopting changes.
- Install a culture of continuous improvement

8.3 Human factors in the change management process

This section addresses people's attitude to change and their learning needs. An individual or team's reaction to change depends on their previous experience of change implementation, their attitude to change, the level of trust in the organization and the extent to which the team own the change.

The change management process must allow for awareness, discussion, and differences in attitude to change when planning the improvement implementation.

The Elizabeth Kübler-Ross model [Kübler-Ross 07] examines stages of grief around bereavement and expected bereavement, and this has been used in business change as a metaphor for how people

© International Software Testing Qualifications Board

sometimes handle change to working practice. More recently, [Adams et al] added further stages (marked in the list with *). The stages are:

- Relief * : “at least I now know what’s happening”
- Shock and/or surprise* : a sense of disbelief
- Denial: total non-acceptance of the change and maybe proving to oneself that it is not happening and hoping that it will go away
- Anger: experiencing anger and frustration
- Bargaining: in an attempt to avoid the inevitable
- Depression: hitting the lows and responding with apathy or sadness
- Acceptance: reality of the situation is accepted
- Experimentation*: after having been very inward looking with acceptance, the idea arrives that perhaps there are things ‘out there’.
- Discovery*: the discovery that things may not be as bad as first imagined.

Note that although these are described as stages, they are not stages in a linear process. People confronted by change do not necessarily pass through all these stages in this order. They may also repeat stages or even miss some stages out. The “stages” are simply a description of emotional responses to change.

[Honey&Mumford 02], [Kirton web], and [Myers&Briggs 95] provide information about

- types of individual (e.g. Myers-Briggs Type Indicator).
- the motivation for an individual for change and improvement.
- whether change will be welcome or resisted.
- whether teams will be early or late adopters of improvement.
- whether they will be prepared to experiment and accept some failure in the changes or whether they will not be prepared to change until a “perfect” solution is provided.
- the different learning styles that individuals prefer and hence acceptable/engaging ways to present the proposed changes [Honey & Mumford].
- an individual’s preference for change by adaption of existing methods or by innovation of new methods [Kirton Web].

© International Software Testing Qualifications Board

9. Critical Success Factors 225 mins.

Keywords:

critical success factor, test process improvement manifesto

Learning objectives for critical success factors

9.1 Key success factors

LO 9.1.1 (K2) Explain the risks behind not considering the critical success factors.
LO 9.1.2 (K3) Use the critical success factors to perform a project risk analysis on a test improvement project
LO 9.1.3 (K4) Select appropriate measures to mitigate the projects risks identified

9.2 Setting a culture for improvement

LO 9.2.1 (K2) Understand the factors involved in setting a culture for improvement
LO 9.2.2 (K3) Demonstrate that cultural factors have been considered in a test improvement plan

9.1 Key success factors

In section 8 the process of change management was described and identified as a key success factor in introducing test process improvement. In this chapter a number of additional factors are discussed in two distinct sets.

"Getting started"

The first set of success factors is primarily related to the initial phases of an improvement project and can be linked to the "Initiating" and "Diagnosing" phases from the IDEAL improvement framework (section 2.4.2):

- clear and realistic objectives for the improvement process are set
- management commitment and sponsorship available
- test improvement organized as a formal project
- people involved have sufficient time scheduled for participation
- ambitions mapped to the maturity of the (development) organization
- change management process established (see chapter 8)

"Getting the job done"

The second set of success factors is related to the implementation phases of an improvement project:

- clear time scales for improvements and length of feedback cycles are defined. These are not too large so that momentum can be maintained
- clear and realistic improvement targets for every cycle
- process ownership identified and organized
- control and monitoring of all steps in the change management process (see chapter 8)
- test professionals involved when defining and implementing the improvements
- other stakeholders involved where problems lay outside the testing discipline, e.g. quality of specifications, change and release management processes
- resistance managed; marketing performed, e.g. level of resistance will depend on the success or failure of previous improvement efforts

© International Software Testing Qualifications Board

- existing practices used if already available; don't change for the sake of changing. If something which is available is not being used then first investigate the reasons why
- stable project team organized who work well together and buy into the change/vision
- tools to support and/or enable test improvements considered (see section 2.5.4.2)
- available knowledge and skills of people involved considered. This covers not just testing in general but also areas related to the improvement process and skills for the improvement approach(es) to be used (e.g. specific model, analysis technique)
- human factors such as their learning styles, personality types and attitudes consider
- external consultants involved as needed, e.g. for specific knowledge and skills, but do not let them manage the improvement project
- awareness of external standards, which may be mandatory, e.g. FDA for the medical industry
- overall process and terminology defined up front to ensure that the various components of the improvement strategy are aligned and part of an overall framework
- relationships built with all affected stakeholders, e.g. software process improvement officers, quality assurance and human resources
- progress clearly demonstrated
- internal approval and/or regulatory processes obeyed

9.2 Setting a culture for improvement

Improvement needs to be set in a cultural context, for example

- The management culture in the organization (command and control, consultative, team driven) will influence the acceptability of the approach suggested
- The geographical location of the organization (e.g. some models and improvement approaches are better accepted in the US, some are more accepted in Asia)
- The organization's goals, policy and strategy, and attitude to improvement (e.g. whether an improvement approach is already in use elsewhere in the organization and if that has been successful)
- Relationships between organizational departments, e.g. If two companies merge, then there may be resistance to changing to improved procedures if these are perceived as coming from the "other" organization.
- The life cycle model being used (sequential, iterative, Agile," home grown" or no process) will influence the frequency cycle for changes in process to be acceptable to projects
- The test approach being used (automated, manual, scripted, exploratory, mixed approach, ad hoc testing) will influence the acceptability of the type of change being suggested.

An example of an approach is the Test Process Improvement manifesto [van Veenendaal Paper 08] which models the Agile Manifesto and suggests that we should consider the following points:

Flexibility over Detailed Processes suggests that organizations will have to engage with change, and respond to those changes with a range of risk profiles. The need for flexibility also acknowledges that testers are knowledge workers and that they should be thinking, adapting and applying processes depending on the specific context for a project. Flexibility and freedom in process shows trust in people and will motivate those people to improve.

Best Practices over Templates says that templates are useful but examples are even better as they show how to use the template. Best practice examples need not be an absolute standard for the industry – they are simply what is best in the specific circumstance so one would expect several contrasted examples to choose from.

Deployment orientation over Process orientation. Building process is easy, the challenge in improvement is deploying processes so that they are used. Process improvement is all about change management; the bulk of the investment in improvement will be in deployment.

© International Software Testing Qualifications Board

Reviews over Quality Assurance (departments) says that communication and providing feedback are essential to project success. It is exactly what peer reviews do, if applied well. QA reviewers may be too far from the test team to give timely and valued feedback. Feedback loops are most effective when they are local and fast.

Business driven over Model driven reminds us that the improvement is to benefit the business not just to provide conformance to an external standard.

This approach to improvement would be acceptable in an organization sympathetic to team driven approaches, Agile life cycles, exploratory testing. It may be harder to "sell" in an organization with a command and control management style, strong reliance on detailed processes, and scripted tests.

© International Software Testing Qualifications Board

10. Adapting to different Lifecycle models 60 mins.

Keywords:

Agile, agile testing, extreme programming, lifecycle model, SCRUM, retrospective meeting, RUP

Learning objectives for adapting to different lifecycle models

10.1 Adapting to different lifecycle models

LO 10.1.1 (K2) Understand the factors that influence the way improvement is organized and that this is always context dependent
LO 10.1.2 (K2) Summarize the test improvement approach in agile environments
LO 10.1.3 (K2) Summarize the test improvement approach in iterative environments
LO 10.1.4 (K2) Give examples of where test process improvement models need to be adapted to be suitable for agile and/or iterative life-cycles

10.1 Adapting to different lifecycle models

The improvement methods described earlier in the syllabus are not specific to any particular lifecycle or methodology. However, improvement always needs to be set in a particular context, for example

- The management culture in the organization (command and control, consultative, team driven) will influence the acceptability of the approach suggested.
- The life cycle model being used (sequential, iterative, agile," home grown" or no process) will influence the frequency cycle for changes in process to be acceptable to projects.
- The test approach being used (automated, manual, scripted, exploratory, mixed approach, ad hoc testing) will influence the acceptability of the type of change being suggested.

For example, adoption of an Agile lifecycle means

- emphasis on self-managing teams, who can change their own processes at need.
- the relationship to Lean approaches at an organizational level need to be considered.

These factors may bias the choice of improvement methods to those favored in Lean manufacturing, for example short Deming cycles and use of cause-effect diagrams. This does not mean that these techniques cannot be used in other lifecycle models, nor does it mean that model-based approaches cannot be applied to organizations or projects using Agile lifecycles. The choice of life cycle should not dictate the choice of improvement method.

Any software process model or test process model can be used as a reference point on how to actually improve what has been judged as important to improve. In agile or iterative contexts, many alternative ideas lead to different improvement paths compared to the traditional lifecycle models, which principally have sequential emphasis.

In an iterative context you might use ideas from e.g. RUP (Rational Unified Process) or in an agile environment you might use ideas from e.g. SCRUM, such as retrospective meetings at the end of each sprint to provide very fast feedback loops and the opportunity to perform process improvement every few days. In an agile context many improvement paths of the content-based models with sequential emphasis have to be tailored extensively. Agile testing, in the context of defined in SCRUM or Extreme Programming) and other related developing testing practices can also provide testing structure more suitable to agile processes.

© International Software Testing Qualifications Board

Sequential models such V-model have feedback loops for checking product and process conformance and suitability (verification and validation) at all phase ends. From these phase-end reviews suggested improvements to the test process (static and dynamic testing) may be made via the use of any of the methods described in this chapter.

If a particular improvement model has been chosen, for example if a CMMi level is a target, that does not prevent the use of any particular approach to test improvements, nor does it dictate any specific life cycle model.

The focus for improvement teams is:

- Identify whether the chosen life cycle model predisposes particular improvement process choices
- Identify what is the correct fit of process improvement method for the context
- Identify the appropriate test structures and practices for the context.

© International Software Testing Qualifications Board

11. References

11.1 Standards

Identifier	Standard reference
[BS-7925-2]	BS 7925-2 Software Component Testing
[IEEE 1044]	IEEE Std 1044™ IEEE Standard Classification for Software Anomalies
[ISO 25000]	ISO/IEC 25000:2005 Software engineering. Software product quality requirements and evaluation (SQuaRE). Guide to SQuaRE
[ISO/IEC 15504]	ISO/IEC 15504-5 ISO 15504 - SPICE (Software Process Improvement and Capability dEtermination), Part 5, Assessment Model [1998]
[ISO 9126]	ISO/IEC 9126-1:2001, Software Engineering – Software Product Quality
[ISTQB]	ISTQB Glossary of terms used in Software Testing, Version 2.0, 2007

11.2 Trademarks

The following registered trademarks and service marks are used in this document:
CMM®, CMMI®, EFQM Excellence Model™,TMM^SM, TMMi®, IDEAL^SM, TMap®, and TPI®

CMM and CMMI are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

EFQM Excellence Model™ is trademark of the European Foundation for Quality Management.

IDEAL is a service mark of Software Engineering Institute (SEI), Carnegie Mellon University

TMM is a registered service mark of Illinois Institute of Technology.

TMMi is a registered trademark of TMMi Foundation

TPI is registered trademarks of Sogeti Nederland B.V.

11.3 Books

Identifier	Book reference
[Adams et al]	Adams, Hayes and Hopson, “Transition: Understanding & managing personal change”, 1976)
[Anderson 01]	Anderson, L. W. and Krathwohl, D. R. (eds) (2001). A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon.
[Atwater 81]	Eastwood Atwater, I Hear You. Prentice-Hall, 1981, ISBN 0-13-450684-7
[Bernstein]	Bernstein A. J. Emotional Vampires: Dealing With People Who Drain You Dry” McGraw-Hill Professional; ISBN-10: 0071381678, ISBN-13: 978-0071381673
[Black03]	Rex Black, “Critical Testing Processes”, Addison-Wesley, 2003, ISBN 0-201-74868-1

© International Software Testing Qualifications Board

Identifier	Book reference
[Bloom 56]	Bloom, B. S. (1956). Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain, David McKay, Co. Inc
[Burnstein 03]	Ilene Burnstein, “Practical Software Testing”, Springer, 2003, ISBN 0-387-95131-8
[Buzan 95]	Tony Buzan, The Mindmap Book, BBC-Books, 1995, ISBN 0-563-37101-3
[Cialdini]	Cialdini, R, “Influence: The Psychology of Persuasion” HarperBusiness ISBN-10: 006124189X, ISBN-13: 978-0061241895
[Craig02]	Craig, Rick David; Jaskiel, Stefan P., “Systematic Software Testing”, Artech House, 2002, ISBN 1-580-53508-9
[Copeland 03]	Lee Copeland, “A Practitioner's Guide to Software Test Design”, Artech House, 2003, ISBN 1-58053-791-X
[Evans04]	Isabel Evans, “Achieving Software Quality through Teamwork”, Artech House, 2004, 978-1580536622
[Frank 90]	Milo O. Frank, “How to get your point across in 30 seconds or less”, Simon & Schuster 1990, ISBN 0-552-13010-9
[Gilb & Graham]	Gilb, T., and Graham, D., Software Inspection, Addison Wesley 1993, ISBN 0-201-63181-4
[Gladwell]	Gladwell, M., The Tipping Point: How Little Things Can Make a Big Difference; ISBN-10: 0349113467, ISBN-13: 978-0349113463
[Honey&Mumford 02]	Honey, P. and Mumford, A., The Learning Styles Helper's Guide, Peter Honey Publications, 2002, from Peter Honey Publications Limited, 10 Linden Avenue, Maidenhead, Berks SL6 6HB or website [Honey-Web]
[Humphrey]	Humphrey, W. Introduction to the Team Software Process, Massachusetts: SEI, 2000 Introduction to the Personal Software Process, Massachusetts: SEI, 1997
[Huff 93]	Darrell Huff, How to Lie with Statistics, W.W. Norton, 1993, ISBN 0-393-31072-8
[ISTQB-CEP]	ISTQB Certified Tester Expert Level, Certification Extension Process, Version 1.0, June, 17th 2008. Available from [ISTQB-Web]
[ITIL]	ITIL, Best Practice for Service Support, Office of Government Commerce, 2002
[ITIL2]	ITIL, Best Practice for Service Deliver, Office of Government Commerce, 2002
[IDEAL 96]	Bob McFeeley/Software Engineering Institute (SEI), IDEAL: A User’s Guide for Software Process Improvement, CMU/SEI-96-HB-001, 1996
[Ishikawa 91]	What Is Total Quality Control? The Japanese Way, Prentice Hall Business Classics, ISBN-10: 013952441X
[Juran]	Quality Handbook (McGraw-Hill International Editions, Industrial Engineering Series) 0071165398
[Kotter & Rathgeber 05]	John Kotter and Holger Rathgeber, Our Iceberg is Melting, Pan Macmillan, 2005, ISBN 978-0-230-01420-6
[Koomen/Pol 99]	Tim Koomen, Martin Pol, “Test Process Improvement”, Addison-Wesley, 1999, ISBN 0-201-59624-5, is this the latest edition
[Kübler-Ross 07]	Elisabeth Kubler-Ross & David Kessler, On Grief and Grieving: Finding the Meaning of Grief Through the Five Stages of Loss, Scribner Book Company; Reprint edition, 5 Jun 2007, ISBN-13: 978-0743266291, ISBN-

© International Software Testing Qualifications Board

Identifier	Book reference
	10: 0743266293
[Maslow]	Maslow, A H, "Toward a Psychology of Being", ISBN-10: 0471293091, ISBN-13: 978-0471293095, Wiley 1998 "Maslow on Management", ISBN-10: 0471247804, ISBN-13: 978-0471247807, Wiley, 1998
[Mayer 04]	Mayer, J.D., Emotional intelligence: Key readings on the Mayer and Salovey model, 2004, ISBN 1-867943-72-2
[Myers&Briggs 95]	Myers, Isabel Briggs (1980). Understanding Personality Type. Davies-Black Publishing; Reprint edition , 1995, . ISBN 0-89106-074-X
[Nance & Arthur 02]	Managing Software Quality: A Measurement Framework for Assessment and Prediction, Springer, 2002, ISBN-10: 1852333936
[Page 08]	Page, A, Johnston, K, Rollinson B, How we test software at Microsoft pub Microsoft 2008 ISBN-13 978-0-7356-2425-2
[Robson 95]	Problem Solving in Groups, Gower, ISBN 0-566-07415-x
[Trienekens & van Veenendaal 97]	Trienekens and van Veenendaal, "Software Quality from a Business Perspective", Kluwer, 1997
[Tufte 90]	Tufte, E., Visual Explanations, Graphics Press, 1990, ISBN 0-961-39214-2
[Tufte 97]	Tufte, E., Envisioning Information, Graphics Press, 1997, ISBN 0-961-39211-8
[Wagner 91]	The Transactional Manager, Spiro Press, ISBN-10: 185835496X, ISBN-13: 978-185835496, 1996
[Weinberg 92]	Weinberg, G, Quality Software Management Vol. 1 [92), Dorset House, 1992, ISBN: 0-932633-22-6

11.4 Papers and Articles

Identifier	Paper / article reference
[Basili Papers]	• V.R. Basili, "Software Modeling and Measurement: The Goal Question Metric Paradigm," Computer Science Technical Report Series, CS-TR-2956 (UMIACS-TR-92-96), University of Maryland, College Park, MD, September, 992. • V.R. Basili, H. D. Rombach, "The TAME Project: Towards Improvement-Oriented Software Environments," IEEE Transactions on Software Engineering, vol.SE-14, no.6, June 1988, pp.758-773 • V.R. Basili, R.W. Selby, "Data Collection and Analysis in Software Research and Management," Proceedings of the American Statistical Association and Biomeasure Society, Joint Statistical Meetings, Philadelphia, PA, August 1984. • R. Basili, D. M. Weiss, "A Methodology for Collecting Valid Software Engineering Data," IEEE Transactions on Software Engineering, vol. SE-10, no.6, November 1984, pp. 728-738.)

© International Software Testing Qualifications Board

Identifier	Paper / article reference
[Copeland Paper 01]	Lee Copeland, “When helping doesn’t help”, SQE Magazine, January 2001
[Garvin Paper 84]	Garvin, D., “What does product quality really mean?”, Sloan Management Review, Vol. 26, No. 1, 1984
[Hackman and Oldman Paper 76]	J. R. Hackman, G. R. Oldham (1976). "Motivation through design of work". *Organizational behaviour and human performance* **16**: 250–279.
[van Veenendaal Paper 08]	van Veneendaal, E., Test Process Improvement Manifesto, Testing Experience, Dec 2008

11.5 Web references

Even though these references were checked at the time of publication of this Expert Level syllabus, the ISTQB can not be held responsible if references are no longer available.

Identifier	Web reference	Link
[ISTQB-Web]	Web site of the International Software Testing Qualifications Board. Refer to this website for the latest glossary, syllabi.	www.istqb.org.
[EFQM-Web]	Web site of the European Foundation for Quality Management. Note that links are provided on the EFQM web site to country/continent-specific models based on the fundamental concepts of excellence for other members of GEM – the Global Excellence Model group, which includes members in North America, Latin America, Australia, and Singapore. The EFQM model is only one manifestation of the use of these concepts and is referred to here as an example.	www.efqm.org
[Honey-Web]	Web-site of Peter Honey	www.peterhoney.com
[Kirton-Web]	McHale, J., Innovators Rule OK-or do they?	www.kaicentre.com
[SEI-Web]	Web site of the Software Engineering Institute, for CMM and CMMi and other Software Engineering Institute publications	www.sei.cmu.edu
[SFIAWeb]	Web site of the SFIA Foundation (Skills Framework for the Information Age)	www.sfia.org.uk
[TMMi-Foundation-Web]	Web site of the TMMi Foundation	www.TMMifoundation.org

© International Software Testing Qualifications Board

12. Appendix A – Notice to Training Providers

12.1 Training Times

Each chapter in the syllabus is assigned an allocated time in minutes. The purpose of this is both to give guidance on the relative proportion of time to be allocated to each section of an accredited course, and to give an approximate minimum time for the teaching of each section.

Training providers may spend more time than is indicated and candidates may spend more time again in reading and research. A course curriculum does not have to follow the same order as the syllabus.

It is not required to conduct the course in one continuous block of time.

The following guidelines have been used:

Learning Objective K-Level	Minutes (average)
K2	15
K3	60
K4	75
K5	90

The table below provides a guideline for teaching and exercise times for each chapter and shows separately the timing for exercises which may be conducted in the workplace (all times are shown in minutes). Note that exercises in the workplace may also be conducted as part of the course given by the training provider (see section 12.3.1 below).

Nr.	Chapter	Course teaching and exercises	Exercises in the workplace	Total
2	The context of improvement	390	0	390
3	Model-based improvement	495	0	495
4	Analytic-based improvement	435	75	510
5	Selecting the approach for test process improvement	15	90	105
6	Process for improvement	480	285	765
7	Organization, roles and skills	375	0	375
8	Managing change	180	60	240
9	Critical success factors	90	135	225
10	Adapting to different lifecycle models	60	0	60
	Total	**2520**	**645**	**3165**

The following table shows the total course times in days, based on an average of seven hours per working day.

Course element	Days
Course teaching and exercises	6
Exercises in the workplace	1,5
Total:	7,5

© International Software Testing Qualifications Board

12.2 Standards used

The syllabus contains references to established standards (see section 11.1), which must be used in the preparation of training material. Each standard used must be the version quoted in the current version of this syllabus. Other publications, templates or standards not referenced in this syllabus may also be used and referenced, but will not be examined.

12.3 Practical Exercises

Practical work should be included for all aspects where candidates are expected to apply their knowledge (learning objective of K3 or higher). The lectures and exercises should be based on the learning objectives and the description of the topics in the content of the syllabus.

12.3.1 Guidelines for practical exercises in the workplace

Certain learning objectives may be achieved by conducting practical exercises in the workplace. The following table shows the learning objectives which may be covered by these workplace exercises:

Subject area	**Relevant learning objectives**
Analytical-based improvement	LO 4.4.2 (K4) Recommend appropriate metrics and indicators for tracking improvement trends in a particular improvement situation.
Selecting test process improvement approaches	LO 5.1.2 (K5) Make a recommendation for a choice of approach(es) in a specific scenario and for a given improvement scope
Process for improvement	LO 6.3.2 (K3) Plan and perform assessment interviews using a particular process or content model in which an awareness of interview style and inter-personal skills are demonstrated
	LO 6.3.3 (K3) Create and present a summary of the conclusions (based on an analysis of the findings) and findings from an assessment.
	LO 6.3.4 (K2) Summarize the approach to solution analysis
	LO 6.3.5 (K5) Formulate and document test process improvement recommendations on the basis of assessment results and the analysis performed
	6.4.5 (K4) Create a test improvement plan.
Managing change	8.2.2 (K3) Address change management as part of the overall test improvement plan, with appropriate steps and actions
Critical success factors	9.1.2 (K3) Use the critical success factors to perform a project risk analysis on a test improvement project
	9.1.3 (K4) Select appropriate measures to mitigate the projects risks identified

The following guidelines apply:

© International Software Testing Qualifications Board

1. Training providers must publish any requirements for practical exercises in the workplace before the course commences.

2. The training provider must approve a proposal submitted by the participant before the practical exercise takes place.

3. The training provider must ensure that the relevant teaching has been provided before the participant performs the practical exercise.

4. Communication between the training provider and the participant must be made available for answering questions and checking on progress.

5. The results of the practical exercise must be submitted to the training provider. It is recommended that the results are presented or at least made available to other course participants.

12.4 Generic guidance to training providers regarding criteria

12.4.1 Expert Level training provider criteria

A training provider needs to submit an application form to the National Board. The National Board grants the training provider a right to provide ISTQB Expert Level courses on the Expert Level module based on following criteria.

- It is generally required that the training provider is also accredited to provide the following ISTQB courses:
  - Foundation level
  - Advanced Level, module Test Management
- Alternatives are allowed, if the reason is well-formulated to the board.
- Training providers should have at least two Expert Level trainers for the module approved by the National Board. Having only one trainer can be approved at the discretion of the board.
- Training providers have recognized status in testing (as a training provider or other), especially in ”Improving the testing process”

12.4.2 Expert Level courses

A training provider needs to submit an application form to National Board to have the course material approved. The National Board approves the course material based on the following criteria.

- Course material for an Expert Level module complies to the syllabus
- Timing complies to the values defined in section 12.1 “Training times”.
- The time spent on exercises discussions during the course and work-based assignments shall not be less than the timings given in section 12.1 “Training times”..
- Maximum class size 10 (to enable personal focus from the trainer)
- Needs to use at least the terms defined in the current ISTQB glossary.

12.4.3 Trainer entry criteria

At Expert Level there will be a formal accreditation of the lecturers. Trainers needs to submit an application form to the National Board. A trainer needs to be able to build on what has been taught on previous ISTQB levels. The trainer must also be able to convey that expertise to the participants in a manner which is effective and permits them to learn. This can be judged with following criteria.

- The trainer has passed at least the required ISTQB Advanced module(s) as defined by the entry criteria or equivalent (e.g. ISEB Practitioner Version 1.1 – 4th September 2001)
- It is highly recommended that the lecturers possesses the full ISTQB Advanced certificate.

© International Software Testing Qualifications Board

- It is highly recommended to have a university degree, have teaching experience, be recognized as an expert and have sound experience. As reaching all of these is not often practical, it will be possible to compensate one with another. Thus the trainer needs to meet additionally at least 2 of the criteria below:
    - The trainer has substantial teaching experience (5 years with at least 3 classes totaling 5 days per year taught) and teaching experience in the subject matter (3 or more classes taught in the subject area covered by the Expert Level module)
    - The trainer is a recognized testing industry leader, conference speaker, author or equivalent
    - The trainer has advanced university degrees related to testing (e.g. B.Sc, M.Sc. or Ph.D.)
    - The trainer has at least 6 years of real world experience in testing and 2 years expertise in the subject covered by the module

If the criteria are not fulfilled, the board can appoint the trainer to be a trainee trainer in Expert Level courses, provided they are certified to full ISTQB Advanced Level and fulfill at least one criterion from the additional criteria list. After 3 co-deliveries with an Expert Level trainer, the trainee trainer can resend his/her application and gain the full trainer status at the discretion of the board.

© International Software Testing Qualifications Board

Index

© International Software Testing Qualifications Board