

The Evolutif Tester Knowledge Base

Intranet Based Testing Resources for Practitioners - A White Paper

Systeme Evolutif Limited
Gloucester House
59 Gloucester Place
London
W1H 3PE
Telephone: +44 (0) 207 487 4705
Facsimile: +44 (0) 207 487 2960
Mail: paulg@evolutif.co.uk
Web: www.evolutif.co.uk
Version 1.1
6 September 1999

The Evolutif Tester Knowledge Base

Intranet Based Testing Resources for Practitioners - A White Paper

CONTENTS

Contents	i
1 Introduction	1
1.1 Purpose of this Document	1
1.2 TKB™ Technical Platform	1
1.3 Demonstration Version.....	1
1.4 The Testing Market	1
1.5 The State of Testing in the Industry	1
1.6 Improving Testing Infrastructure.....	3
2 The Tester Knowledge Base.....	4
2.1 Intranet-Based Testing Resource.....	4
2.2 TKB™ Major Components.....	4
2.3 Implementation of TKB™	6
2.4 Tester Knowledge Base Schematic.....	7
3 The Evolutif Testing Framework: The W model	8
4 Example Pages from the Knowledge Base	9
4.1 Foundation training Course	9
4.2 Feature and Behaviour Inventories	9
4.3 Behaviour Inventory - Template	10
4.4 Glossary of Testing Terms.....	10
4.5 Sample test Process Overview	11
4.6 TOM Questionnaire On-Line.....	11
4.7 Example E-Commerce Test Process Overview.....	12

1 INTRODUCTION

1.1 Purpose of this Document

This document provides an overview of the Systeme Evolutif Tester Knowledge Base product. It is not intended to be a detailed specification of the product, nor is intended to be complete or a commitment to build all of the features described. It is intended to present the thinking behind the Knowledge Base and provide an overview of its component parts. At the end of the paper are some sample screens of the Knowledge Base features to give an idea of its appearance.

1.2 TKB™ Technical Platform

The TKB™ static pages use HTML compatible with the HTML 3.0 standard.

The Test Process Manager is written in Active Server Pages 2.0, using VBScript hosted on NT Server 4.0 running IIS 4.0 and SQL Server 7.

1.3 Demonstration Version

A demonstration version of the Knowledge Base is available for use on the Systeme Evolutif web site <http://www.evolutif.co.uk/>

1.4 The Testing Market

Over the course of the last decade, the discipline of software testing has gained increasing attention in the software industry. Specialist testing services firms have flourished, the larger testing tools vendors depend on services for a significant proportion of their revenues and the testing tools market has grown considerably in size. The general perception of the market place is that it is not yet mature, but has nonetheless changed very considerably since the late 1980s.

Within the testing services sectors there are now specialist companies providing:

- Specialist testing resources including test management, test analysis, testers and automation technicians
- Services relating to test environment set-up and management
- Managed testing teams on or off-site providing outsourced testing services
- Consulting services including process improvement, strategy, audit, supervision and specification
- Testing training services covering accredited syllabuses leading to professional qualifications as well as management, methods and specialist techniques training.

The market is maturing and as testing companies evolve their services and customers recognise the possibility of outsourcing test activities, outsourced testing is emerging as an attractive offering to the service companies. As marketing efforts increase, potential customers are becoming aware of the flexibility outsourced testing companies can provide.

1.5 The State of Testing in the Industry

As consultants specialising in Test Process Improvement, our experience of test practices across the industry is depressingly consistent. Some progress is being made, for example, some test design techniques (available since the 1970s) are becoming established. However, there remain endemic problems in testing that block progress towards consistently better test practices.

We define testing as all those activities relating to the detection of faults in and the evaluation of software deliverables. This definition therefore includes all static test activities such as

The Evolutif Tester Knowledge Base

Intranet Based Testing Resources for Practitioners - A White Paper

reviews and inspections, static analysis of code as well as the traditional structured dynamic testing from component tests through link, system and acceptance testing.

One of the problems of software management is that the costs of testing are known to be high but few managers know where the costs of test activities actually occur. This is mainly because these activities are spread throughout the development lifecycle, and performed by most, if not all, software practitioners as part of their job. Usually, a relatively small number of project testing tasks are identified. Most test activities are deemed to be part of some other 'development' activity. Requirements specification, design and coding activities on project plans often include 'invisible' static test activities. 'Sign-off' may appear on plans as a brief stage, but with a minimal or zero effort assigned to it.

Test activities, particularly static testing and component level testing do not receive the management attention, because they are not visible enough. Late test activities are deemed problematic because the major faults that should be captured early only come to light at the end of the project. As a consequence, late testing is seen as a traumatic, stressful, negative activity that is impossible to estimate, manage and control. No wonder it is difficult to get people to volunteer for testing.

Many companies find it difficult to build permanent test teams. Few staff volunteer because testing is perceived to be a career backwater. Few managers extol the virtues of testing as a career path. Permanent staff often look for opportunities to get back into mainstream development roles. Consequently, many test teams rely on unwilling volunteers, and contract staff. Few have been trained, outsiders tend to stick to ways they used elsewhere and so, ad-hoc test teams rarely work effectively *as* teams.

Few senior managers are ex-testers. Few were ever trained; few are aware of, or take an active interest in recent testing advances. As a consequence, many non-tester's view of testing is that it is negative, boring, repetitive, clerical and of interest only to dullards. This view is perhaps the single greatest barrier to improved practices in many organisations. Perhaps one reason why tools sell so well is that managers believe testing is a robotic activity that can be delegated to unintelligent tools.

Our premise is that:

- The most effective test activities are not visible to management.
- The most visible test activities happen at the end, so get squeezed.
- To most managers, improving testing means adding resources or buying tools for system and acceptance testers.
- Managers believe testing does not add value and is always a problem.

We believe that testing body shopping is a popular 'solution' because companies are just giving up on doing testing themselves.

1.6 Improving Testing Infrastructure

Evolutif has been conducting test process improvement projects since 1991. Our strategy is always to concentrate on improvements that have the greatest benefit to clients. In our experience, no two organisations are alike and our proposed solutions, methodologies, training schemes have always been custom-built, but always based on a consistent set of principles, standards, methods and techniques.

To help us ascertain the underlying reasons for poor test practices, Evolutif devised the Test Organisation Maturity model, TOM™ and built an assessment questionnaire based on the model. We use TOM™ to generate prioritised lists of suggestions for improving an organisation's test processes and practices. Improvement areas normally span the following areas:

Methodology - covering risk assessment, high level test planning, test specification, implementation, execution, incident management and reporting.

Techniques guidelines - detailed processes for static testing, test design and measurement, management and control.

Training - at all levels, staff would be given accredited training, leading to certification, where possible. Additional training in areas such as testing in complex environments, technical reviews, test design and measurement techniques, test automation is often required.

Tools - having established requirements for software testing tools, covering, for example, testware management, test design support, incident management, test execution and test data preparation tools, implementation must follow a proven method to ensure successful use.

Consultancy support - improvement projects normally require support from specialist testing consultants backed by an extensive library of testing resources and network of partner organisations.

2 THE TESTER KNOWLEDGE BASE

2.1 Intranet-Based Testing Resource

Paper-based process, guideline and training resources work effectively for practitioners who are learning new skills and finding their way around a comprehensive methodology. However, when the time comes to apply these skills in a project, paper-based material becomes cumbersome and difficult to use. Methodologies, guidelines and training materials may cover hundreds of pages of text. Testing templates are normally available on a LAN so are not integrated. Most practitioners end up copying the small number of diagrams required to understand the method and pinning this on their wall. Other resources are unevenly distributed across a LAN for which no one has responsibility for maintaining.

The Internet (and Intranets) offer a seamless way to bring these diverse materials together in a useful resource, available to all practitioners. Evolutif now offer to build test infrastructure on Intranets or host it on our own web site. Comprising a large volume of standard reference material, the intention is to build a front-end to the product that supports project risk analysis and the generation of a comprehensive project test process, without the need for consultancy or specialist skills. The test process is built up from standard test types that reference standards, templates, methods, tools guides and training material as required.

The Tester Knowledge Base or TKB™ is a flexible but comprehensive resource for use by practitioners assembled from your existing methods and guidelines, our standard techniques and tools guides all fronted by a risk-based test process manager. The intention is for TKB™ to be a permanently available and useful assistant to test managers and practitioners alike.

2.2 TKB™ Major Components

Test Process Manager

The Test Process Manager is mainly used at the start of the project to help testers to create an inventory of risks from which an effective test process can be based. The Test Process Manager supports Master Test Planning (MTP) whose aim is to create a custom test process from pre-existing test templates. Master Test Planning has three stages: Risk Analysis, Test Objectives and Process Generation. Test Process manager is an active server pages based application supported by a SQL Server database.

The generated Test Process is considered as a table of sequenced test stages, and test types within each stage. Every test stage has the same general format and defines:

Test objectives	The purpose of this stage of testing is defined in terms of risks to be addressed, benefits to be demonstrated, requirements to be validated or specifications to be verified.
Exit criteria	Objective criteria to be met before the test stage ends.
The baseline document(s)	These define the component or system requirements to be met and are used for test design and to prepare expected results.
Test stage deliverables	These must be created during the stage. Standard templates for each deliverable are available on-line for every deliverable.
The process to be used	The step by step tasks to be used to execute the stage of testing. Specification, preparation, execution, analysis and management tasks with responsibilities are fully documented.

The Evolutif Tester Knowledge Base

Intranet Based Testing Resources for Practitioners - A White Paper

Test techniques	The specific methods to be used to plan, specify, prepare, execute and analyse the stage tests.
Test tools	Where the tests can be supported by test automation, standard tools which fall into usable categories can be looked up and the supplier identified. Where in-house tools are already available, links to standard tools guidelines are provided.
Test environment(s)	The test environment where the testing is to take place is identified as part of the test stage definition.

Test Guidelines

All test stages, tasks, techniques and standards have guidelines covering background, purpose and detailed explanation of the various components of the test process. Each guideline is linked from the sections in the Test Process Manager and test process definitions.

Testing Training

Testing training materials, consisting of presentation slides, narrative text associated with each slide bullet, graphics as well as interactive questions and answers have been converted to the web format. The ISEB Foundation course and other course materials are available. Where appropriate in the guidelines, links to sections of the training materials are available.

Test Automation

The CAST Report, comprising a handbook of automated testing guidelines and a tools directory is also available. Where tool support is available, the test process will link to tools that are appropriate to the test types. If tools are already available in-house, links to internal tools guidelines exist.

Standards

All documentation deliverables have on-line standard template documents available. By default, these are in Rich Text Format that is compatible with most word processors. Where your own standards exist, these will be substituted in your own preferred format.

Process Improvement

The Test Organisation Maturity model, TOM™ is also available for completion on-line. This can be used to generate recommended process improvements, based on responses to standard questionnaires. Responses can be compared with the analyses across industry sectors and also used to track process improvement progress over time.

2.3 Implementation of TKB™

A Framework, not a Fixed Package

It is not expected that TKB™ would be implemented as a fixed package. Rather it should be viewed as a framework within which client organisations can assemble a unique combination of TKB™, in house and custom-built components, using either industry-standard or your own testing terminology.

Modular Construction

In our experience, some organisations wish to implement improved test practices for only a part of their development processes. For example, it is common for us to implement methodologies covering Front-Loaded testing (evaluation of requirements and design documents), early test preparation, system and acceptance testing. This is appropriate if most development work is outsourced, for example.

Test Process Manager is Optional

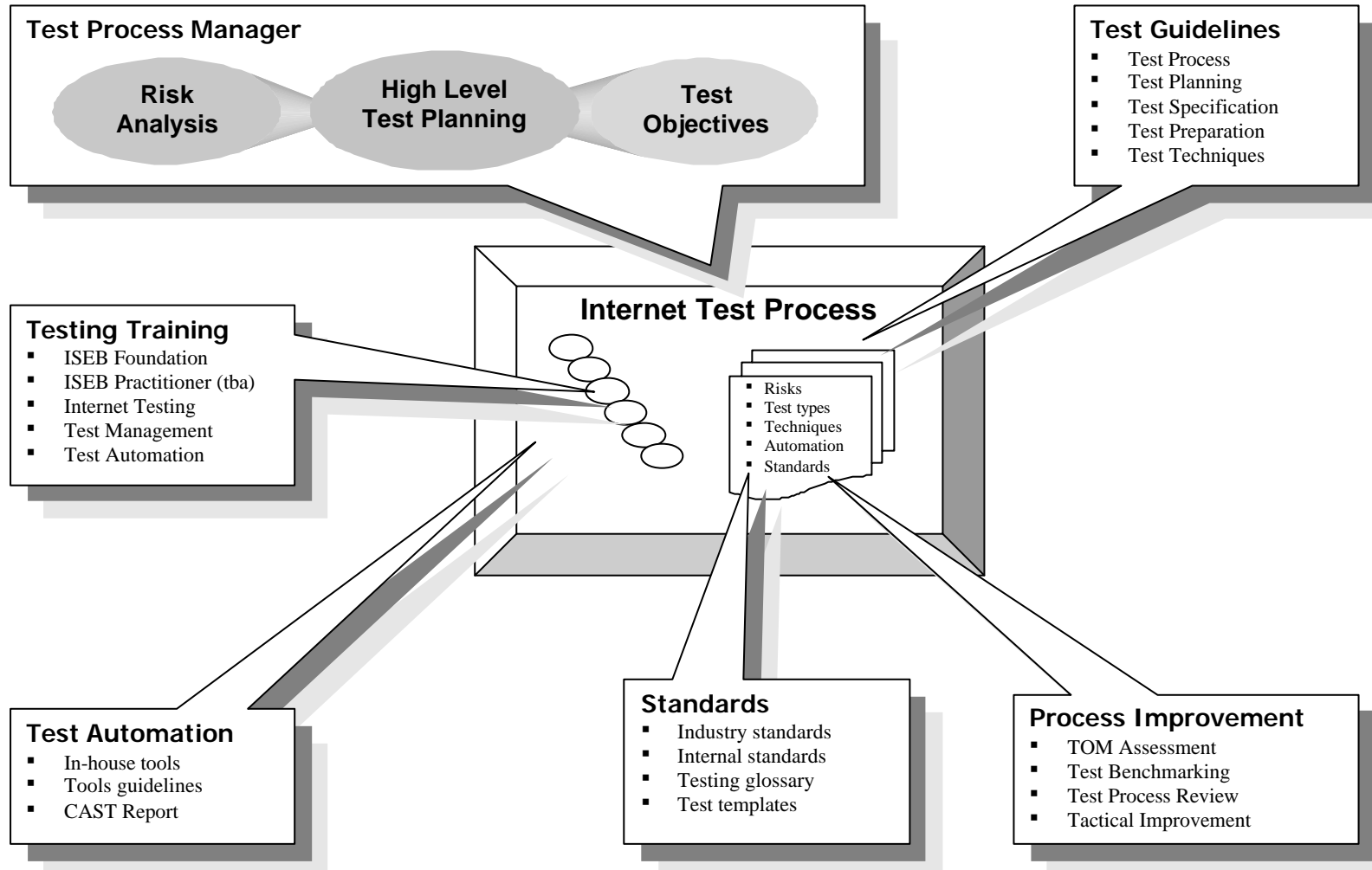
If you already have a well established methodology, with fixed paths through a defined test process, the Test process Manager is less likely to be of benefit. Your test process can be substituted into TKB™, but all the referenced guidelines, glossaries of terms and standard templates can be linked, in just the same way. The Test Process Manager is the only active application, as the remainder of TKB™ is built from static HTML pages.

The schematic on the following page provides an overview of the structure of the product.

The Evolutif Tester Knowledge Base

Intranet Based Testing Resources for Practitioners - A White Paper

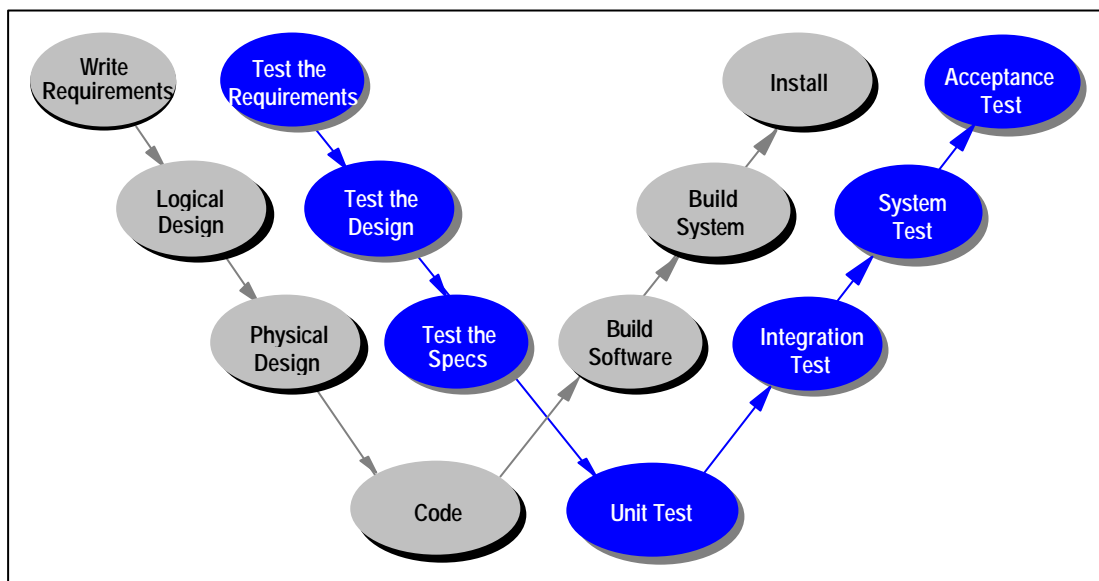
2.4 Tester Knowledge Base Schematic



3 THE EVOLUTIF TESTING FRAMEWORK: THE W MODEL

To many software engineers and development managers, testing is the last phase of the development life cycle. However, waiting until after executable software has been built is the most costly and least effective way of performing testing.

The belief that tests can only be run against executable software is, frankly, wrong. There are test techniques that can be applied throughout the entire development life cycle, even as early as the requirements gathering stage. These techniques are not only cheaper than dynamic testing, but they also avoid wasting money on building the wrong system. Fault prevention is cheaper than fault correction.



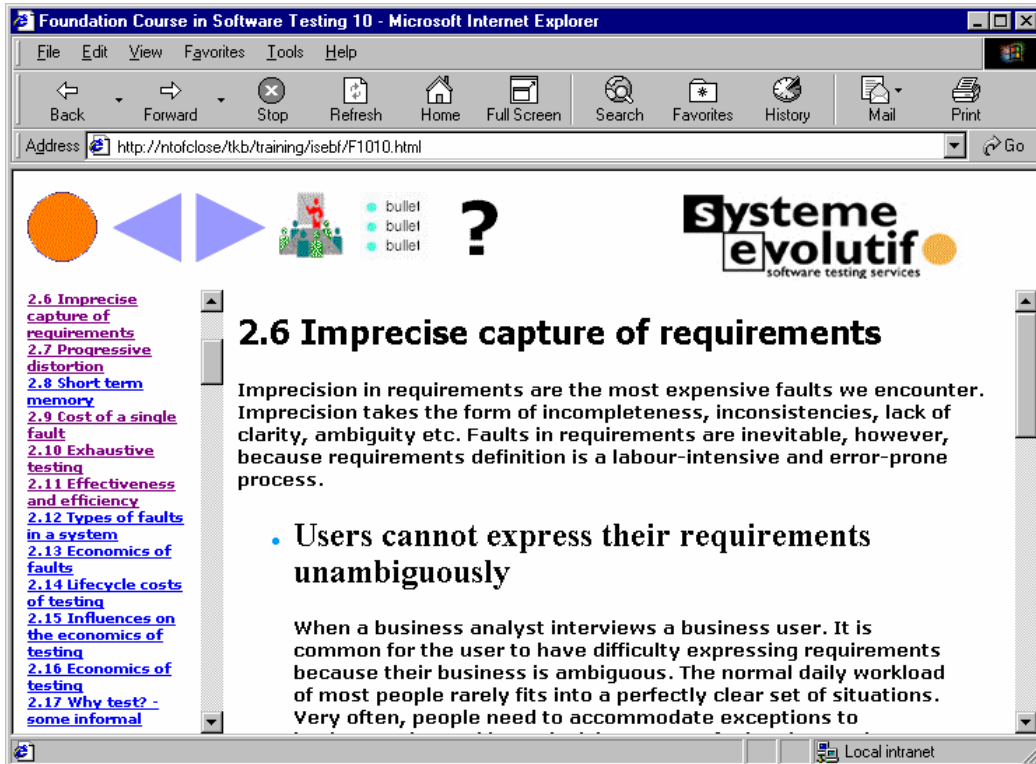
The W Model

Testing is least costly and most effective if it is performed throughout the whole life cycle, in parallel with every stage of development. This strand of testing in parallel with development is represented in the W model.

For those who are familiar with the V model, the W model is a natural evolution. The V model illustrates the layered and phased nature of software testing, but lists only dynamic test stages like unit and system testing. Some variations also include early preparation of test cases. The W model, by contrast, supports testing of all deliverables at every stage of the development life cycle.

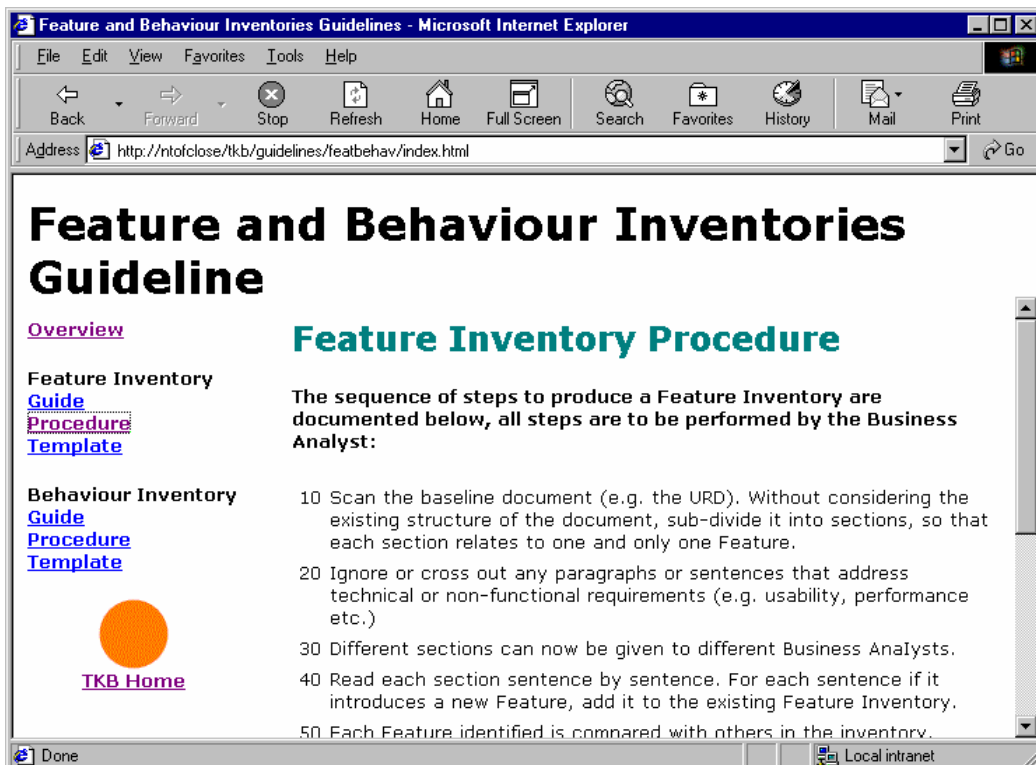
4 EXAMPLE PAGES FROM THE KNOWLEDGE BASE

4.1 Foundation training Course



The screenshot shows a Microsoft Internet Explorer window with the address bar displaying <http://ntofclose/tkb/training/isebf/F1010.html>. The page content includes a navigation menu on the left with links such as [2.6 Imprecise capture of requirements](#), [2.7 Progressive distortion](#), [2.8 Short term memory](#), [2.9 Cost of a single fault](#), [2.10 Exhaustive testing](#), [2.11 Effectiveness and efficiency](#), [2.12 Types of faults in a system](#), [2.13 Economics of faults](#), [2.14 Lifecycle costs of testing](#), [2.15 Influences on the economics of testing](#), [2.16 Economics of testing](#), and [2.17 Why test? - some informal](#). The main content area features the heading **2.6 Imprecise capture of requirements**, followed by a paragraph: "Imprecision in requirements are the most expensive faults we encounter. Imprecision takes the form of incompleteness, inconsistencies, lack of clarity, ambiguity etc. Faults in requirements are inevitable, however, because requirements definition is a labour-intensive and error-prone process." Below this is a bulleted list item: "• Users cannot express their requirements unambiguously". A second paragraph explains: "When a business analyst interviews a business user. It is common for the user to have difficulty expressing requirements because their business is ambiguous. The normal daily workload of most people rarely fits into a perfectly clear set of situations. Very often, people need to accommodate exceptions to". The Systeme Evolutif logo is visible in the top right corner.

4.2 Feature and Behaviour Inventories




The screenshot shows a Microsoft Internet Explorer window with the address bar displaying <http://ntofclose/tkb/guidelines/featbehav/index.html>. The page content includes a navigation menu on the left with links such as [Overview](#), [Feature Inventory Guide](#), [Procedure](#), and [Template](#). The main content area features the heading **Feature and Behaviour Inventories Guideline**, followed by a sub-heading **Feature Inventory Procedure**. Below this is a paragraph: "The sequence of steps to produce a Feature Inventory are documented below, all steps are to be performed by the Business Analyst:". A numbered list follows: "10 Scan the baseline document (e.g. the URD). Without considering the existing structure of the document, sub-divide it into sections, so that each section relates to one and only one Feature.", "20 Ignore or cross out any paragraphs or sentences that address technical or non-functional requirements (e.g. usability, performance etc.)", "30 Different sections can now be given to different Business Analysts.", "40 Read each section sentence by sentence. For each sentence if it introduces a new Feature, add it to the existing Feature Inventory.", "50 Each Feature identified is compared with others in the inventory." The TKB Home logo is visible in the bottom left corner.

4.5 Sample test Process Overview

Sample Test Process for Project X

[Process Summary](#)
[Test Deliverables](#)
[Process Overview](#)



[TKB Home](#)

ID	Activity	Description
6	System Test Design	Review Behaviour Table (if changes required) and prioritise
7	Build System Test Design Specification	Compile Test Specification from Behaviour Inventories and System Test Condition List
8	Prepare System Test Data	
9	Prepare System Test Procedures with Instructions and Expected Results	
10	Execute System Tests (Run Tests)	Execute System Test Procedures for changed/new functionality and regression tests

4.6 TOM Questionnaire On-Line

Test Organisation Maturity

[TOM Assessment Overview](#)
[The TOM Form](#)
[TOM questionnaire On-line](#)
[TOM Surveys: June 1999](#)


[TKB Home](#)

1 - this is a LOW priority - not a problem for us
 5 - this is a HIGH priority - it hurts!

Ref:	Symptom	Score	Priority
		1 - LOW 5 - HIGH	1 - LOW 5 - HIGH
S01	Testing has no deliverables of value	3	4
S02	Testing has little purpose	4	3
S03	Test objectives are less important than development objectives	2	2
S04	Testing is expensive, but the costs are not visible	4	5
S05	Of the errors that are found, there is a		

The Evolutif Tester Knowledge Base

Intranet Based Testing Resources for Practitioners - A White Paper

4.7 Example E-Commerce Test Process Overview

Test Type	Test Priorities				Static/ Dynamic	Test Types Mapped to Usual Test Stages				
	Smoke	Usability	Performance	Functionality		Desktop Development	Infrastructure Testing	System Testing	Integration Testing	Post-Deployment Monitoring
Static Testing										
HTML testing	Y				S	A/M				
Browser syntax compatibility	Y				S	A				
Visual browser validation		Y			D	M		M		M
Test Browsing										
Link checking	Y				D			A		A
Object load and timing		Y	Y		D			A		A
Transaction verification	Y				S	A/M		A/M		
Functional Testing										
Browser page testing	Y				D	A/M				
CGI component testing	Y				D		A/M			
Transaction Testing				Y	D			A/M		
Application testing				Y	D			A/M		
Internationalisation		Y			D	A/M		A/M		
Non-Functional Testing										
Configuration testing	Y				D	M		A/M	M	
Performance			Y		D		A	A		A
Soak Testing/reliability	Y				D	A	A	A	A	
Availability					D					A
Usability		Y			S/D			M		
Security				Y	D		A/M	A/M	A/M	A
Large Scale Integration										
External links/legacy system integration				Y	D		A/M		A/M	
End to end functionality	Y				D				A/M	A