

# TESTADVANCE TEST-SYSTEM ENGINEERING

**This document provides a comprehensive description of testadvance's Test-System Engineering services**

Positioning

Introduction and methodology of Test-System Engineering

Services

Detailed description of

- Test-System Specification services (TSS)
- Test-System Design services (TSD)
- Test-System Optimisation services (TSO)

Rev D.01 Last modified: 15/05/10 Author: Ralph Becker

## Table of Contents

1	Introduction.....	3
1.1	The System of Test.....	4
2	Test-Engineering (TE).....	6
2.1	Benefits from Test-Engineering Services.....	6
2.2	Key Benefits to Business.....	6
2.3	Key Benefits to Operations.....	6
2.4	Quality & Fulfilment .....	6
2.5	Description of Test-System Engineering .....	7
2.5.1	Structured Test-System Engineering.....	8
2.5.2	Test-Engineering Process (TEP).....	8
2.6	Test-Specification Services (TSS).....	9
2.6.1	Client Benefits from Test-System Specification Service.....	9
2.6.2	Test-System Specification Deliverables .....	9
2.6.3	Test-System Specification Process (TSSP).....	10
2.6.3.1	Requirements Analysis .....	10
2.6.3.2	Functional Requirements Design.....	10
2.7	Test-Design Services (TSD).....	11
2.7.1	Client Benefits from Test-System Design Service.....	11
2.7.2	Test-System Design Deliverables .....	11
2.7.3	Test-System Design Process (TSDP).....	12
2.7.3.1	System Concept.....	13
2.7.3.2	Test-System Architecture.....	13
2.7.3.3	Test-System Instrumentation and Test-Plan.....	13
2.7.3.4	Compile and verify.....	14
2.7.3.5	Optional elements.....	14
2.8	Test-Optimisation Services (TSO).....	14
2.8.1	Client Benefits from Test-System Optimisation Service.....	14
2.8.2	Test-System Optimisation Deliverables.....	14
2.8.3	Test-System Optimization Process (TSOP).....	15
2.8.3.1	Scope of Optimisation.....	15
2.8.3.2	Test-Analyses.....	15
2.8.3.3	Test-Optimisation: Design & Implementation, Benchmark & Review.....	15
3	Glossary.....	16
4	Index of Figures.....	17
5	Index of Tables.....	17

## 1 Introduction

Testing and verification, 'Test', is an integral part of the value chain. Businesses in the electronics industry need to ship products and provide services that satisfy customers and meet standards and regulations. Product and network design alone can not ensure that these criteria are met. Tolerances in components and variations production necessitate testing in manufacturing. Communications networks and in particular radio-networks are subject to variations in usage and deployment, demand and environment.

Businesses rely on their operations to deliver quality product and services, in the required volume respectively capacity, on schedule and demand and at acceptable cost. Operations in turn depend on test-departments to provide the required capability and capacity of testing. Test is a significant investment and cost. It is also a constraint. In essence, Test is an 'insurance' on the quality of the product and service. How much insurance is needed will always be a matter of business strategy. Test is also an asset. Test provides valuable information on the quality and integrity of the value chain. Test can help operations improve on their performance. Test is an important tool a business can use in delivering on its marketing promise.

Test plays a pivotal role in the operation's ability to meet production targets and schedules. The Test-Life-Cycle spans almost the entire product and service life-cycle, from design to support. Its purpose is to provide a reliable and reconcilable link from design, over build to the use of the product and service. Testing ranges from hands-on design-verification in the lab, over verifying prototypes, to production- and field-testing, service and maintenance. As testing matures with the life-cycle, it provides valuable information. Making good use of this information can help accelerate life-cycles and improve operational performance. The information and knowledge gained is invaluable in the execution of both present and future life-cycles.

Requirements and Design provide the foundation for test-requirement. Testing is initially an assortment of individual tests design-engineers use to verify the first designs. Testing becomes more integrated in line with the integration of assemblies and components into functional prototypes, respectively the compilation of environmental data and network concepts into a network topology. Design are thoroughly tested to ensure they meet use and regulatory requirements. The final stage in development is the release-to-manufacturing respectively model-verification and field-trials for networks.

Manufacturing (production) has it's own specific testing needs. Testing verifies that units are built and assembled correctly. Products may need to be adjusted or calibrated. The Test-Process defines where and how tests are executed on the production line. Testing typically coincides with key transitions in assembly, e.g. board test, functional test and phasing and final test. Re-work stations repair faulty products that are put back into the production flow. Special testing may be required or useful, such as 'burn in' or sample testing.

Network design and deployment has its own specific testing needs. Data used in modelling needs to be verified in the field, components and sub-systems themselves need to be tested. The Test-Process defines where and how tests are executed during the design and roll-out of the network. Testing typically coincides with key transitions in the design and build, e.g. propagation testing, functional test (network elements), build and install, drive-test and optimisation. Monitoring and maintenance tests need to be defined as well as diagnostics to determine and repair faulty elements. Special testing may be required or useful, such as 'load' or sample testing.

In an organisational or operational sense, test-engineers and managers provide Test-Capability and Test-Capacity. They develop the test-specifications on which the test-systems and procedures are based. Test-specifications provide a clear and traceable link between requirements, design and delivered product and service. Traceability needs to work in both directions, from requirements to delivery and thereafter, from operation back to requirement. Designers are primarily concerned with assuring adherence to requirements and regulations. Production-engineers, be it in manufacturing or networks, need to achieve a production targets within the given constraints, e.g. costs and schedule. They prefer as little testing as possible. Test-engineers though are concerned with meeting the best balance of all requirements. They need to deliver to schedule and in budget and understand the impact on both over all. They need to balance performance, cost, implementability and reliability, existing infrastructure and assets, users, utilisation and future needs.

Arguably the most crucial step in the Test Life-Cycle is the the release to production. Production can be the shipment of product in volume or the release and delivery of a service to a market. The objective is to create revenue. Businesses need to realise the projected returns on their investment in engineering and assets. Every delay is a delay in achieving profitability. Operations need to achieve the targeted production volumes and yields quickly. Time between the release-to-production and ramp-up must be as short as possible.

## Testadvance Test-System Engineering

This requires that Test Engineers and Managers ensure their design, implementation and release of test-systems and test-procedures is completed within the time allocated within the overall project or life-cycle.

Test-engineering needs to commence well before the development of the product or service, manufacturing or network is complete. Test-engineering is executed concurrently to design, build and deployment. Involving test-engineers in the design phases accelerates test-development and benefits design, build and deployment. Test-engineers structure testing within a Test Process. They define the tests executed within this Test Process. Each test-stage or phase typically has a specific system of test-procedures and -instruments associated with it. Test-systems integrate a number of instruments and components, interfaces, ancillary devices within a hardware and software platform. Test-procedures integrate individual test-methods and test-cases within one or more test-sequences. Modular structures in test-procedures and test-systems, with re-usable test-design enable companies to increase utilisation of assets and reduce costs. Structured methods shorten development and deployment, furthers re-use and minimises non-recurrent-engineering.

Engineering and deploying an efficient and effective test-solution is not a straightforward, sequential process. Nor is it solely a technical or engineering effort. Indeed, it is subject to a variety of outside developments and influences and subject to often contradictory needs and objectives. Test-engineers need to make qualified assumptions and work in and through iterations to achieve the required results. This Test-Life-Cycle is managed as part of the organisational life-cycle. Yet, test-departments and operations must also measure and improve over successive cycles and capitalise on the work and learning's made.

Test-engineers and managers plan, design, deploy and manage testing and verification (Test) for their company. They engineer and deliver test-solutions to schedules, budgets and targets. Their competence and work realises the Test-Capability and Test-Capacity the business needs to bring quality product and services to market, on time, in volume and at a profit.

Test-engineers and managers are 'knowledge workers'. They research, engineer, manage, support, collaborate and communicate. They stay abreast of developments in business, operations and technology. Test-engineers plan, specify, design, implement and deploy test-systems and test-procedures. They implement testing for new technologies, products and services. They liaise with R&D, manufacturing, contractors and vendors. Test-engineers maintain existing testing, implement changes and support operations. Test managers translate business and operational needs into projects and work-requirements. They allocate and manage resources, budget investments and departments and plan and control delivery. Staff, engineers and managers work together to measure and improve their work and its deliverables.

Test-Engineering and Test-Management requires diverse knowledge, skills and abilities. Test-engineers need to be competent in a variety of technologies, applications, systems and platforms. They need to apply that competence towards clear outcomes. They need to be proficient in procedures, processes and operations. Team leaders and managers need to manage projects, resources and workload efficiently. Test-engineers and managers need to be sufficiently self-reliant. They need to capture and apply learning's. They need to work effectively within their team and with their peers, suppliers and vendors.

Test-engineers and managers face considerable challenges. They need to manage workload and change, timeliness and urgencies. Product-Life-Cycles are ever shorter, reducing development time. Technologies are converging and products more complex. Shorter life-cycles lead to more frequent evolution and hence, more engineering. Technology is becoming more consumer oriented. Margins are shrinking and costs are a constant concern. Engineers and managers need to balance engineering effort against schedules and performance against costs. Care must be given to potential costs, e.g. delayed market entry, inadequate yields or quality issues. Test-procedures must be developed in unison with product and manufacturing, yet these are still in development themselves. Instruments must be sourced. They must be technically suitable, affordable, 'engineer able' and maintainable. Test-Engineers and Managers needs to bear all this in mind.

Managing – and utilising – competence is the key to overcoming these challenges. Seeking advice can only further performance.

### **1.1 The System of Test**

Testadvance Test-System Engineering applies 'Systems Thinking' and systems-engineering principle, from the initial discussions through to acceptance and review. Systems thinking applies that a system is best understood through the interactions with other systems in its environment, its elements and their relationships and interactions with each other.

'Test' itself is a system, embedded in the 'systems' of operations and business. It has its own internal sub-systems, is multi-layered, dynamic and distributed. It comprises procedures, data and processes, instruments, test-systems and infrastructure. It is subject to requirements, interactions and influences from across departments and operations. Test-departments use their competence to plan, specify, design, implement, deploy, maintain, upgrade, support and manage testing and test-infrastructure. Test is often technically complex. Yet Test also needs to accommodate a whole raft of influences and factors such as business strategies and budgets, marketing and production schedules, cross-departmental collaboration, market and technology developments. How well Test delivers depends on how well its elements perform, how well they work together and how well they interact with entities outside Test itself.

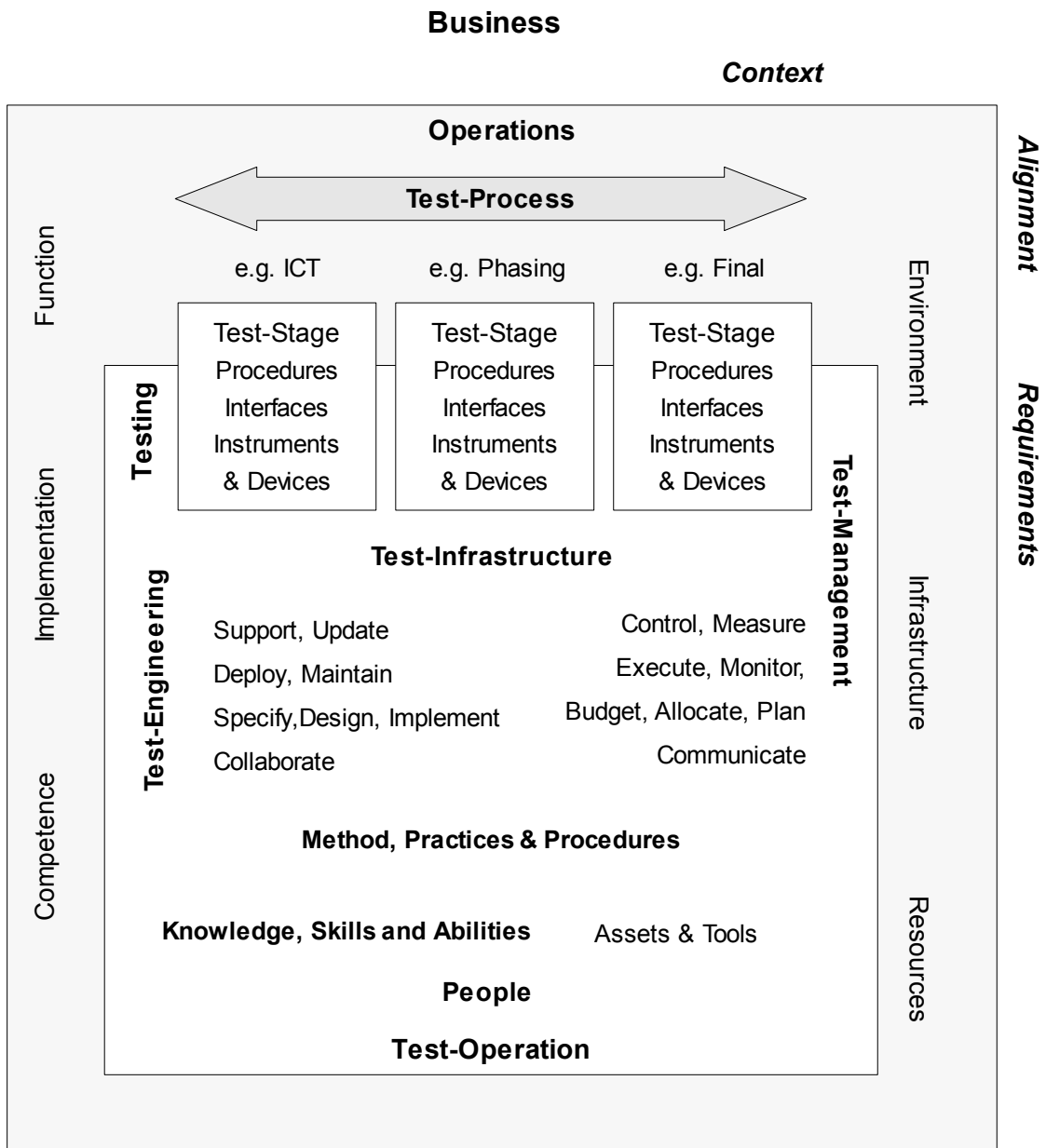


Figure 1 The System of Test

## 2 Test-Engineering (TE)

Test-Engineering provides the specification, design and optimization of a Test-System the testing and test-systems used. The services are based the learning's from over two decades of designing test-systems for clients in the wider electronics industry. The services use a structured and systematic approach based on clear and agreed needs and requirements, objectives and outcomes, budgets and schedules.

- Test-Specification: Requirements Analysis, Functional Design, System Requirements Specification
- Test-Design: System-Concept, Test-Architecture, System-Architecture, Test-Plan
- Test-Optimisation: Scope, Analysis, Measures, Execution, Review

### 2.1 Benefits from Test-Engineering Services

Testadvance Test-System Engineering helps engineers and managers:

- Deliver efficient, quality testing on schedule
- Achieve and exceed project and life-cycle requirements and objectives
- Relieve critical the path and/or resources
- Independent of product or vendor preferences
- Better plan and allocate resources, work and work-effort, deliverables, schedules and costs
- Reduce risk, effort and dependencies in sourcing and implementing test-solutions

### 2.2 Key Benefits to Business

- Achieve profitability earlier
- Reduce cost-of-goods/services sold
- Reduce capital investment and increase ROA
- Further quality and customer satisfaction

### 2.3 Key Benefits to Operations

- Accelerate product life-cycles
- Increase operational efficiency
- Reduce the cost-of-test, work-in-progress, re-work and inventory

### 2.4 Quality & Fulfilment

Each client, each operation and each situation is different, yet the three key criteria of timeliness, function and focus apply to all. It is how they combine that is unique.

- i. Scope: Needs, objectives and constraints and the agreed outcomes
- ii. Perspective: The client's operations, its elements and the influences on test
- iii. Focus: The specific areas to be addressed
- iv. Flexibility: Correct, concise and ready alignment of services with needs and deliverables
- v. Fulfilment: Delivery to needs and objectives, budgets and timelines

All services have quality measures and procedures embedded. A common and structured Engineering Process defines the engineering-work (service process) The 'delivery process' defines the phases and 'in- and outputs' (scope, needs/objectives, requirements, specifications and designs, schedules and outcomes). Delivery and service process have clear and defined links and milestones. Each phase of the service process is linked to 'in- and outputs' of the service process. Communication, Scope and Change Management are embedded in both the service and delivery process:

- Expectations, needs and objectives are understood and agreed, communicated and managed
- Requirements are correct and complete, during and upon completion of their development
- Specifications concur with requirements, technical standards, regulations, et al
- Acceptance covers Expectations, Requirements, Specifications and Performance criteria
- Open and comprehensive, clear and consistent documentation

## 2.5 Description of Test-System Engineering

The efficient and effective engineering and implementation of a Test-System requires understanding its function as an element within the overall 'System of Test'. A Test-System is a 'system within a system' that interacts with other sub-systems and in relation to the whole. The Test-System itself obviously a system in its own, with internal sub-systems that again interact with each other and with the system as a whole. While this indeed may sound confusing at first, bearing the principle in mind from initial discussions of needs and objectives through specification, design and deployment will expedite engineering and ensure its integrity.

	Product/Service	Test-System Engineering work-elements and deliverables	Production
▲ <b>Test-System / Test-Plan Optimisation</b>	Feedback  Test-data	Benchmark & Review Plan and execute implementation Design optimisation Analyse Test-System/-Stage Scope Optimisation Benchmark performance	Monitor, leverage Outcomes Measures Test-Plan, Test-Data Target, extent Metrics, requirements
▲ <b>Test-system / Test-Plan Design</b>	Product Test-specification Test-Methods & Procedures  Measurements & Design Test-Set-ups & prototype testing	<i>Develop Implementation briefs (opt)</i> Compile Test-Plan Design Test-Sequences, -Routines, -Cases et al Test-Layout Compile Test-System Specification Design Test-Architecture Design System Infrastructure/Interfaces Design Non-Measurement Subsystems Design Measurement Interface(s) Design Measurement Subsystem Design System Concept	<i>build, deploy, ramp-up</i> Manufacturing Test-specification  Test-Process  Production and Infrastructure Test & Assembly Production Test-specification
▲ <b>Test-system / Test-Plan Specification</b>	Product Test-specification Measurements Design-verification Technology, etc.	<i>Plan and budget (opt)</i> Compile System Requirement Specification Design Functional Requirements Analyse Requirements Framework Analysis	<i>sourcing</i> Manufacturing Test-specification Test-concept Volume, yields, cycle Line, Infrastructure Maturity, strategy, et al

Table 1 Test-System Engineering in operations

Testadvance Test-System Engineering employs proven engineering methods and expertise. Structured analysis and structured design are applied together with agile development methods. Where possible, we apply agile development principles and execute work concurrently to relieve the critical path. Each engagement provides only the work and deliverables needed and agreed with the client while maintaining the integrity of the outcome. We integrate work and outcomes with the client's projects, teams and operations. We ensure solutions are efficient and effective, upgradeable and reliable.

### 2.5.1 Structured Test-System Engineering

Testadvance Test-System Engineering is a structured portfolio of services grouped into the three types Specification, Design and Optimisation. Each service type corresponds to a key 'knowledge-work' phase of the Test-Life-Cycle.

- Test-Specification: System Requirements Specification
  - Requirements Analysis
  - Functional Requirements Design
  - Planning & Budgeting
- Test-Design
  - Test-System Concept
  - Test-System Architecture
  - Test-System Instrumentation
  - Test-Plan
- SW Functional Design
  - System Design Specification
  - Test-Plan Specification
  - Planning & Budgeting
- Test-Optimisation
  - Scope of Optimisation
  - Test-System/Stage Analysis
  - Optimisation Design
  - Implementation
  - Benchmark & Review

Table 2 Test-System Engineering Services

Test-System Engineering comprises a structured portfolio of pre-defined service-units and -modules. Units and modules are organised according to the typical work-structure and process for that specific phase of engineering. Units and modules can be delivered as individual engagements, e.g. a Test-Plan (unit), or a Test-Routine (module). Service-units can be configured into packages. A pre-defined service-package contains by default all service-units required to deliver the engineering work, e.g. a complete system requirements specification. Tailored packages are configured to a client's specific preferences. These packages provide only specified work-elements while retaining the integrity of the work-segment or project. The modular structure allows us to execute work concurrently. The clear delineation and 'interfaces' between modules ensures good communication with the client, simplifies integration and minimises risk. Modularity helps identify flaws prior to the system being deployed, reduce engineering overheads and furthers re-use. It helps teams more readily assess and implement changes.

### 2.5.2 Test-Engineering Process (TEP)

The Test-Engineering Process is common to all our Test-System Engineering. The TEP is adapted accordingly for each specific service and service type. The TEP has four fundamental phases: Scope, Analysis, Design and Execution. Key to the process is that all the detailed and iterative work is encapsulated within the appropriate phase. At the highest level, the TEP aligns the individual engineering services with the client's engineering and test life-cycle. At the service-level, the TEP organises and integrates the various service-units, -modules and work-elements required to deliver the expected outcome from the service, e.g. a SRS. Each phase is defined by its objective and outcome (input and output). These are determined in the Scope. Objective and outcome are broken down into specific requirements and metrics for a phase. All services include a dedicated review after delivery. It is important to understand that we define the TEP process for the 'worst-case' scenario: The process accommodates by default all units and work-elements that could be needed. In practice, engagements include only those units and work-elements actually needed.

<i>Client Test-Life-Cycle</i>				
<b>Plan/Specify ►</b>	<b>Design ►</b>	<b>Build</b>	<b>► Implement / Deploy</b>	<b>► Operate / Maintain</b>
▲	▲		▲	
<b>Test-Specification</b>	<b>Test-Design</b>		<b>Test-Optimisation</b>	
Execution	Execution		Execution	
Design	Design		Design	
Analysis	Analysis		Analysis	
Scope	Scope		Scope	

Table 3 Test-Engineering Process (TEP)

Each Test-Engineering service first and foremost considers the client's situation and their specific needs, objectives and constraints. Time-critical and specific needs are addressed immediately. For all other cases, it is important to establish the correct scope and focus. A dedicated Scope of Engagement defines the client's situation and timelines, the desired outcomes and projected returns. It identifies the areas and elements of Test-System Engineering concerned. The Scope is provided prior to a financial commitment from the client.

Delivery is managed using the common project management elements Scope of Engagement, Statement of Work, Work Breakdown Structure, Delivery and Acceptance schedules. Each is only as comprehensive as needed. Communication, change management are included within the delivery process. The service can provide for integrated teams. We coordinate work with the client's schedules and operations to minimise disruptions. All services include a dedicated review at key milestones and after delivery. Engineering work is executed as concurrently as feasible. We leverage practices from Dynamic Systems Development Method (DSDM), Rapid Application Development (RAD) and Structured Analysis and Design (SA/SD). We apply methods we have successfully developed and applied over the years. These include considerations for modularity, platform development and re-use, forward testing and integration; monitoring, and reliability, utilisation and 'up-time'.

## **2.6 Test-Specification Services (TSS)**

Test-System Specification services provide test-engineers and managers with an independent, timely and quality specification of Functional Testing. Engagements are fixed-price and outcomes are guaranteed.

### **2.6.1 Client Benefits from Test-System Specification Service**

Testadvance Test-System Specification services give test-engineers and managers:

- A complete, correct, pre-optimised and 'ready for design' System Requirement Specification (SRS)
- Reduced lead-times, overheads, risk, re-work and iterations in planning, budgeting and design
- The ability to remove research and assessment of requirements and solutions from the critical path
- Independent sourcing, cost and scheduling estimates and cost/benefit assessment
- Considerations for validation, verification, optimisation, reliability and re-use/leverage

### **2.6.2 Test-System Specification Deliverables**

Test-System Specification delivers a 'ready-for-design' System Requirement Specification (SRS). The SRS defines scope and deployment; measurements, functions, interfaces; attributes and performance; alternatives and cost/benefit assessments. It can provide planning and budgeting information and incorporate other work, outcomes and resources. The service provides only the work and deliverables needed and agreed while maintaining the integrity of an 'executable' SRS.

- Framework Analysis/Scope of Engagement
  - System Control and Automation
- Requirements Analysis Report
  - Attributes
- Functional Requirements Report
  - Alternatives and Cost/Benefit assessment
- Measurement Functions
- Non-Measurement Functions
- Interfacing, Input/Output
- Planning and Budgeting
- System Requirements Specification Document

#### **Table 4 System Requirements Specification**

The client receives revisions of the SRS throughout its development. Sections are provided at the end of each phase (segment) of the service, starting with the Scope of Engagement (SoE). The SRS includes relevant methods, calculations, underlying requirements, regulations et al; cost/benefit assessments and rationales; interim and final reviews and as applicable reference solutions, alternatives, commercial solutions and sources and references and implementation-briefs .

### 2.6.3 Test-System Specification Process (TSSP)

The Test-System Specification Process (TSSP) is a tailored version of the general Test-Engineering process. The TSSP applies the four common phases of Scope, Analysis, Design and Execution. The process organises the service-units and work-elements needed to develop the System Requirement Specification within the appropriate phase. Each phase determines the objectives of the subsequent phase. Emphasis is on the analysis and design phases. Analysis focuses on analysing requirements. Design pertains to the design of functionality and specification. Execution is limited to compiling an 'executable' SRS. The TSSP itself is encapsulated within the Plan/Specify phase of the Test-Life-Cycle.

- Framework Analysis
  - Define needs and outcomes
  - Analyse framework, target operation, et al
  - Define scope
- i. Requirements Analysis
  - i. Evaluate product and technology
  - ii. Define operational and operation's requirements
  - iii. Define required capabilities and capacities
  - iv. Define required performance
- ii. Functional Requirements Design
  - i. Specify measurement functions
  - ii. Specify non-measurement functions
  - iii. Specify Interfaces and in-/output
  - iv. Specify system control and automation
  - v. Specify attributes
  - vi. Consider alternatives
  - vii. Weigh cost/benefits
- iii. Plan/Budget design and implementation
- iv. Compile System Requirements Specification document

Table 5 Test-System Specification Process TSSP

The TSSP commences with the common Framework Analysis that sets the focus, scope and objectives. We discuss the client's needs, objectives and constraints, the target operation, schedules and budget. We review the client's installed base and infrastructure. We discuss principle cost/benefits and potential alternatives. We configure the service units and work-elements. Available work and outcomes are leveraged. We provide the this Scope of Engagement (SoE) prior to a financial commitment from the client.

#### 2.6.3.1 Requirements Analysis

The Requirements Analysis consolidates the various information and requirements into a structured requirements profile. The profile defines required capabilities and performance requirements. We focus on establishing requirements that apply testing at its boundaries, i.e. at the DUT. Capabilities are functions such as measurements, adjustments, interfaces, control, data-handling, etc. Performance is structured into functional performance and capacities. Functional performance is the performance of a specific function, e.g. a measurement uncertainty, an output power-level, etc. Capacities describe the performance of the system within the operational process, e.g. throughput, yield, repeatability. Example: A required capability is to measure the RF Power Output of a mobile phone. The functional performance would be the accuracy with which this has to be measured, the signal levels at the DUT, the required interfacing etc. The capacity would be to do this x number of times every y seconds while maintaining functional performance. Essentially, we create a matrix of all required capabilities and the performance they need to provide at the DUT. We compile and review the information in the Requirements Analysis.

#### 2.6.3.2 Functional Requirements Design

Here we define the functionality that will provide the required performance. We focus on defining how capabilities and capacities need to be provided. The focus here is to detail requirements within the boundaries of a potential test-system and interfacing. Using our above example: The RF Output Power measurement is executed via the RF interface (over the air). We determine that shielding and filtering are required and the path must be switched. This necessitates certain input and output parameters and can rule out certain measurement methods and instruments. We detail the individual entries in requirements profile. We add new requirements as they are identified. We review the now detailed requirement. We ensure the individual components we have defined combine correctly and deliver the required performance. Finally, we define the attributes required, e.g. reliability, environmental, instrument types, existing infrastructure, user and through-life needs etc.

Functional Requirements are defined in iterations, starting with the overall requirements at the system-boundaries. The number of iterations depends on the complexity of the requirement. These can be minimised depending on how much definition is already provided in e.g. a product test-specification. This method ensures an adequately structured requirement specification. It helps avoid stipulating requirements that later may need to be overturned or re-engineered. We suggest feasible alternatives. It is important to be clear that the Test-Specification process does this for the functional requirements and not for the design of the test-system itself. The key is that the process does not add unnecessary effort and in fact minimises ambiguity and effort in the subsequent Test-Design. It also provides for excellent modularity/separation re-use and traceability. We compile the information from the Requirements Analysis and Functional Requirements Design into the System Requirements Specification. We review the SRS with the client. If requested, we assess and estimate project costs, provide budgetary sourcing, planning and schedules.

## 2.7 Test-Design Services (TSD)

Testadvance Test-System Design services provide test-engineers and managers with an independent, timely and quality design of Functional Testing. Engagements are fixed-price and outcomes are guaranteed.

### 2.7.1 Client Benefits from Test-System Design Service

Testadvance Test-System Design services give test-engineers and managers:

- A complete, correct, pre-optimised and 'ready for build' Test System/Test Plan Design Specification
- A cost-, production- and 'engineering-efficient' test-design
- Reduced lead-times, overheads, risk, re-work and iterations in planning, budgeting and build
- Furthers re-use and leverage through of design-work
- Expedited integration, testing and 'de-bug' and rectification of flaws before deployment/ramp-up
- Considerations for diagnostics, maintenance, reliability, process control and feedback
- Assistance in procurement, planning, budgeting & implementation

### 2.7.2 Test-System Design Deliverables

Test-System Design delivers a 'ready-for-build' System Design Specification (SDS). The SDS provides the design of the Test-System and the Test-Plan. It includes specifications and information for build, integration, and verification. The SDS includes as feasible alternatives and cost/benefit assessments. It can provide planning and budgeting information. Implementation briefs on engineering work, costing, procurement, and uncertainty analyses are included as required. The service can incorporate other work, engineering and resources. It provides only the work and deliverables needed and agreed while maintaining the integrity of an 'executable' SDS. If requested, a SW Functional Design can be provided as a requirement specification and SOW for e.g. a sub-contractor.

- Framework Analysis
  - Measurement Interface
- Test-System Concept
  - Interpretation of SRS
  - Concept and Alternatives
  - Cost/Benefit Assessment
- Test-System Architecture
  - Test-Methods and Cases
  - Test-Set-ups
- Test-System Instrumentation
  - Measurement Subsystem
- Test-Plan
  - Test-Routines
  - Non-Test-Routines
  - Test-Sequences
- SW Functional Design
- Planning and Budgeting, Implementation Briefs

Table 6 System Design Specification

### 2.7.3 Test-System Design Process (TSDP)

The Test-System Design Process is a tailored version of the Test-Engineering process. The TSDP applies the four common phases of Scope, Analysis, Design and Execution. The process organises the service-units and work-elements needed to design the test-system and test-plan. Each phase determines the objectives of the subsequent phase. The TSDP itself is encapsulated within the Design phase of the Test-Life-Cycle.

#### Test-System Design Process

- Framework Analysis
  - Define needs and outcomes
  - Analyse framework, target operation, et al
  - Define scope
- i. Develop Test-System Concept
  - i. Evaluate and interpret System Requirements Specification
  - ii. Design the concept of the test-system and its operation
  - iii. Investigate alternatives
  - iv. Weigh cost/benefits, select concept
- ii. Design Test-System Architecture
  - i. Specify Test-Methods/cases from SRS
  - ii. Draft Test-Set-ups based on Test-Methods/cases and concept/probable instruments
  - iii. Draft Measurement Subsystem design
  - iv. Draft Measurement Interface design
- iii. Design Test-System Instrumentation
  - i. Compare designs and align with SRS
  - ii. Design Test-Methods, Test-Cases, Test-Set-ups
  - iii. Design Measurement Subsystem
  - iv. Design Measurement Interface
- v. Design Non-Measurement Subsystems
- vi. Design System Interfaces
- vii. Design System Infrastructure
- viii. Verify Subsystem designs against SRS
- iv. Design Test-Plan
  - i. Design Test-Routines
  - ii. Design Non-Test-Routines
  - iii. Specify Test-Layout in target operations and process
  - iv. Specify Test-Sequences and System Control
  - v. Verify Sub-system designs and Test-Routine/case integrity
  - vi. Verify Test-Routines and Cases et al with Test-Sequence
- v. Compile System Design Specification
  - i. Verify completeness and correctness
  - ii. Verify subsystem integration
  - iii. Verify system control and automation
  - iv. Verify system integrity
  - v. Verify integration in target operation
- vi. optional: Develop SW Functional Design
- vii. Develop implementation briefs
- viii. Plan/Budget design and implementation

Table 7 Test-System Design Process TSDP

We apply structured and modular design principles in our designs of test-systems and test-plans. This allows us to execute work concurrently. The approach furthers re-use and simplifies integration, verification and trouble-shooting. We segment the Test System Subsystems into groups of related and similar functions. We structure the test-system in the core Measurement Subsystem and Measurement Interface, Non-Measurement Subsystems, System Infrastructure and System Interfaces. Each subsystem is structured into functional modules.

The TSDP commences with the common Framework Analysis. We provide clients with a clear explanation of the design process and how it can be tailored to their situation. We discuss needs, objectives and constraints, the target operation and schedules, budgets and targets. We review the existing System Requirement Specification and agree on its applicability. We discuss which elements of the test-design are to be provided and how work is integrated with the client's engineering work and teams. The outcome is the Scope of Engagement (SoE). The SoE is provided prior to a financial commitment from the client.

### 2.7.3.1 System Concept

In the System Concept we identify the basic structure of the system, its sub-systems and their key functional modules. We first review the SRS and ensure each requirement is adequately specified. We then assign requirements and functions to the appropriate subsystem. Then we define the generic modules that make up the subsystem, e.g. a signal-analyser within the Measurement sub-systems. We at this stage do not specify instruments and elements in detail unless a definitive selection is sensible. We subsequently assign the functions and requirements within the sub-system to the appropriate module. Where requirement and assignment are unambiguous a single iteration is all that is needed. Where multiple solutions are possible, we execute as many iterations as there are feasible configurations. This approach removes any later ambiguity. It also provides the modularity, re-use and structure we seek. We will use this same approach in the detailed design. We compile the System Concept, assess the alternatives and weigh their cost/benefit. We review the outcomes with the client and agree on the System Concept.

### 2.7.3.2 Test-System Architecture

The Test-Architecture provides the structure of the Test-System. Here we first develop the actual tests from the functional requirements. We then apply these to the sub-systems and modules and specify the major elements of the test-system and its sub-systems. We start with measurement functions. We develop these first at the boundaries of the DUT. We determine set-up and access and define Test-Method, Test-Cases and test-criteria. Once defined, we further develop these tests from the boundaries of the test-system to the major modules of the sub-systems. We focus here on the Measurement sub-system and its instruments and signal-paths. We identify the test-instrument(s), devices and confirm/amend Test-Method and set-up. We define 'black box' substitutes for unknown areas, especially the interfacing between test-system and DUT. We take care to ensure that in/outputs are defined and realistic. In parallel we determine the main signal paths and their key characteristics. We verify each test and set-up against the SRS. We verify the integrity of key tests from DUT through to test-instrument and that the overall Test-Architecture concurs with the SRS. We review alternatives and weigh their cost/benefit. We compile the SDS and review it with the client.

### 2.7.3.3 Test-System Instrumentation and Test-Plan

Here we detail the design of the Test-System and Test-Plan. We develop the Test-System and Test-Plan concurrently to one another. Each area provides input and feedback to the other. Each iteration details the design further and tightens specifications. We ensure key objectives, detailed requirements and design are properly linked. We retain the overview needed to balance cost, timeliness and performance.

We first complete the Measurement Subsystem and Interface. (As discussed, their development begins in the Test-Architecture design). We detail elements and their specifications down to the most detailed level, e.g. components, switching and I/O ports. We specify their performance. We calculate uncertainties as required and we provide a complete measurement uncertainty analysis if needed. We develop the Test-Plan concurrent to the Measurement Subsystem/Interface. We detail and specify Test-Methods, cases and criteria. We design executable routines and specify specific system 'states', e.g. routing and switching definitions. In parallel we start to design the other subsystems, commencing with those that support the Measurement Subsystem and Interface. During the development of each subsystem we capture needs that determine the design of the System Infrastructure and System Interfaces. We design each sub-system individually and with the appropriate interface to ensure interoperability and integrity of the system overall. We develop for each subsystem a specific design and integration 'check-list' and procedure. The subsystems are then checked for their interoperability and integration in the test-system. We review the designs and their integration against the SRS. We compile all the sub-system and modules designs within the Test-System section of the SDS.

We design the Test Plan in three concurrent paths: Test-Layout, Test-Sequences and Test-Routines. The test-layout aligns testing within the operational test-process. Test-Sequences define the order of execution of Test-Routines, system-functions etc. Test-Routines (routines) are in essence functions. Test-Routines are specific to a test, typically measurements and set-ups. Non-Test-Routines are basically all other functions, such as system tasks, data-handling etc. Test- and Non-Test-Routines are use the same modular structure:

- Method defines the procedure by which the measurement or function is executed
- Set-up defines the devices and configuration with which the measurement or function is executed
- Cases define the conditions, variables, parameters, instrument settings, and device states, etc. used
- Criteria specify nominal values, tolerances, e.g. pass/fail tolerances etc.

Test-Routines are optimised according to their function, e.g. measurements for speed and accuracy. Special routines can be designed for monitoring and analysis, system self-test and for process and quality control. Each routine is checked with the elements of the test-system it uses. We verify the integrity of the routine and its correctness. We compile the Test-Sequences, -routines and sub-system and modules designs within the Test-Plan section of the System Design Specification.

#### 2.7.3.4 Compile and verify

We consolidate all the sections, designs and specifications within the System Design Specification. We then review the completed SDS. We verify its completeness and correctness. We verify that it fulfils both SRS and overriding, high-level objectives and needs. We take care to detach the verification from the design work. We check the functions the test-system provides and compare these to the requirements. We verify integration and control/automation by simulating/replicating key test-scenarios involving one or more sub-systems. We verify test-system integrity in the same fashion, with the focus on 'end-to-end' testing. We then check and verify the integration of the test-system within its target environment. The verification procedures and results can be provided as a draft integration and verification plan for the test-system build and deployment.

#### 2.7.3.5 Optional elements

If requested, we can provide planning and budgeting information including estimated project costs, budgetary sourcing, planning and schedules. We can also provide implementation briefs that describe integration and verification procedures and operation and user-documentation.

If requested, we can provide a functional SW design-specification. The SW-specification is based on the test-plan. We can provide the SW specification as a Requirement Specification, including e.g. pseudo-code to describe critical instrument control and measurement functions for SW developers.

## 2.8 Test-System Optimisation Services (TSO)

Testadvance Test-System Optimisation services provide cost-effective optimisation of a functional test-system or test-plan. Engagements are fixed-price and outcomes are guaranteed.

### 2.8.1 Client Benefits from Test-System Optimisation Service

Testadvance Test-System Optimization services give test-engineers and managers:

- Increased productivity of a test-system and/or test-plan
- Improved quality, throughput and yields and reduced work-in-progress (WIP)
- Measures and practices for continual test-reduction

### 2.8.2 Test-System Optimisation Deliverables

Test-System Optimisation delivers a defined and measured optimisation of a test-system/test-plan. Clients receive a clear scope of the effort and defined outcomes, prior to a financial commitment. The service provides an independent analysis of the target system/plan and its current and required performance. Measures are designed according to the scope defined. They are both targeted and balanced. All measures and their outcomes are weighed to ensure the best cost/benefit to the client. The service can incorporate client resources and work. Implementation is integrated with the client and executed to minimise disruptions.

- Framework Analysis
  - Test Data/Utilisation
- Scope of Optimisation
  - Benchmark
  - Requirements
  - Cost/Benefit
- Test-Analyses
  - System-Architecture/Test-Plan
  - Test-Optimisation Design
  - Test-Optimisation Delivery
  - Benchmark and Review

Table 8 Test-System Optimization Report TSOR

### 2.8.3 Test-System Optimization Process (TSOP)

The Test-System Optimisation Process is a tailored version of the general Test-Engineering process. The TSOP applies the four common phases of Scope, Analysis, Design and Execution. The process organises the service-units and work-elements needed to analyse and optimise a test-system and test-plan. Each phase determines the objectives of the subsequent phase. The TSOP itself spans the Build, Deploy and Maintain phases of the Test-Life-Cycle.

- Framework Analysis
  - Define needs and outcomes
  - Analyse framework, target operation, et al
  - Define scope
- i. Develop Scope of Optimisation
  - i. Benchmark Test-Stage
  - ii. Specify Requirements
  - iii. Define Optimization
  - iv. Assess Cost/Benefit
- ii. Analyses
  - i. Test Data
  - ii. Test-Architecture/Instrumentation
  - iii. Test-Plan
  - iv. Utilisation
  - v. Compile Performance Report
- iii. Design Optimisation Measures
  - iv. Assess cost/benefit
  - v. Implement
  - vi. Benchmark & Review

Table 9 Test-System Optimization Process TSOP

The process commences with the Framework Analysis. We provide clients with a clear explanation of the optimisation process and how it can be tailored to their situation. We discuss the client's needs, objectives and constraints. We review the test-operation, test-process and test-stage/-system. We review existing information and discuss how work is integrated with the client's operations and teams. We discuss the projected improvements. The outcome is the Scope of Engagement. The SoE is provided prior to a financial commitment from the client. As early as in the development of the SoE, we take care to ensure we do not 'over-optimize' and equally identify areas that can provide unforeseen benefits.

#### 2.8.3.1 Scope of Optimisation

Here we investigate test-system and testing in more detail. We establish its structure and elements and key performance indicators. We define and execute a benchmark. The benchmark delivers detailed performance criteria and metrics, e.g. test-times, yields, etc. We assess the results and specify the detailed requirements and metrics for the optimization. In parallel we assess the appropriate areas of the testing. We weigh cost/benefit and define the targets, extent and depth of the ensuing analyses and optimisation.

#### 2.8.3.2 Test-Analyses

The process provides for the analysis of Test-Data, Test-Architecture, Test-Plan and Utilisation. The full execution of all four individual analyses represents the 'worst-case' scenario. The previous scope of optimisation defines their extent and weighting. In Test-Data analysis we look for patterns and anomalies. In analysing Test-Architecture and Test-Plan we look for a variety of characteristics that help identify root-causes and areas to address. We focus on areas that provide the best cost/benefit from optimisation. We analyse Utilisation to determine any unrealised or imbalance in capabilities and capacities. We consolidate the results and specify Areas of Improvement (AoI). An AoI is an area that needs to be improved. We further define Areas of Opportunity (AoO) where performance is within expectation, yet where improvements can be made. We consolidate the results in the Performance Report and discuss the outcome with the client.

#### 2.8.3.3 Test-Optimisation: Design & Implementation, Benchmark & Review

Here we design the measures for the optimisation, e.g. a revised Test-Sequence. We assess each proposed measure for cost-benefit and develop a detailed design and specification. We then implement the measures according to an agreed implementation plan. In the implementation we leverage services and work-elements from our Test-System Engineering. We review key stages of the implementation. Upon completion of the optimisation, we jointly execute a detailed performance review and benchmark.

### 3 Glossary

## Alphabetical Index

Areas of Improvement (AoI).....	15
Areas of Opportunity (AoO).....	15
Case.....	7
Compile Test-Plan.....	7
Design Measurement Interface.....	7
Design Measurement Subsystem.....	7
Design System Infrastructure.....	7
Design Test-Sequences.....	7
Dynamic Systems Development Method (DSDM).....	9
Framework Analysis.....	12
Functional Requirements Report.....	9
Measurement Interface.....	11, 12
Measurement Subsystem.....	7, 11, 12, 13
Non-Measurement Subsystems.....	11
Non-Test-Routines.....	11, 12, 13
Performance Report.....	15
Product-Life-Cycle.....	4
Rapid Application Development (RAD).....	9
Requirements Analysis.....	9
ROA.....	6
Structured Analysis and Design (SA/SD).....	9
SW Functional Design.....	11, 12
System Design Specification (SDS).....	11
System Infrastructure.....	11, 12, 13
System Interfaces.....	11, 12, 13
System Requirement Specification (SRS).....	9
Systems Thinking.....	4
Test.....	3
Test Life-Cycle.....	3
Test-Capability.....	3, 4
Test-Capacity.....	4
Test-Case.....	12
Test-Engineering.....	4
Test-Engineering Process (TEP).....	8
Test-Layout.....	7, 12, 13
Test-Life-Cycle.....	3, 4, 15
Test-Management.....	4
Test-Method.....	7, 11, 12, 13
Test-Plan.....	7, 11, 12, 13, 14, 15
Test-Routines.....	11, 12, 13, 14
Test-Sequences.....	11, 12, 13, 14
Test-Set-up.....	7, 11, 12
Test-System.....	12
Test-System Architecture.....	12
Test-System Concept.....	12
Test-System Design Process (TSDP).....	12
Test-System Optimization Process (TSOP).....	15
Test-System Optimization Report TSOR.....	14
Test-System Specification Process (TSSP).....	10
Testing and verification.....	3
-Routines.....	7

## 4 Index of Figures

### Index of Figures

Figure 1 The System of Test.....	5
----------------------------------	---

## 5 Index of Tables

### Index of Tables

Table 1 Test-System Engineering in operations.....	7
Table 2 Test-System Engineering.....	8
Table 3 Test-Engineering Process (TEP).....	8
Table 4 System Requirements Specification.....	9
Table 5 Test-System Specification Process TSSP.....	10
Table 6 System Design Specification.....	11
Table 7 Test-System Design Process TSDP.....	12
Table 8 Test-System Optimization Report TSOR.....	14
Table 9 Test-System Optimization Process TSOP.....	15