

A structured approach to Test Engineering



Focus of this talk

- Test Development
- Achieving consistent efficiency, quality in Test System Engineering
- Building continuous and measurable improvement into test engineering

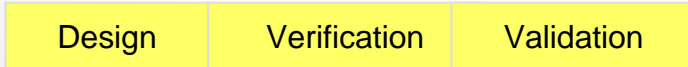
My background

- 1985 BSE (Hons) EE Comms University of Applied Sciences, Giessen (Germany)
- 1987 – 2003 HP/Agilent (Europe, Asia)
 - RF & μ W Systems Engineer, Project Manager
 - Wireless / Test System Consultant
- Founded testadvance in July 2003
- Knowledge Services for Test Engineering, Wireless:
 - Training, Test System Design, Test Optimization, Consulting

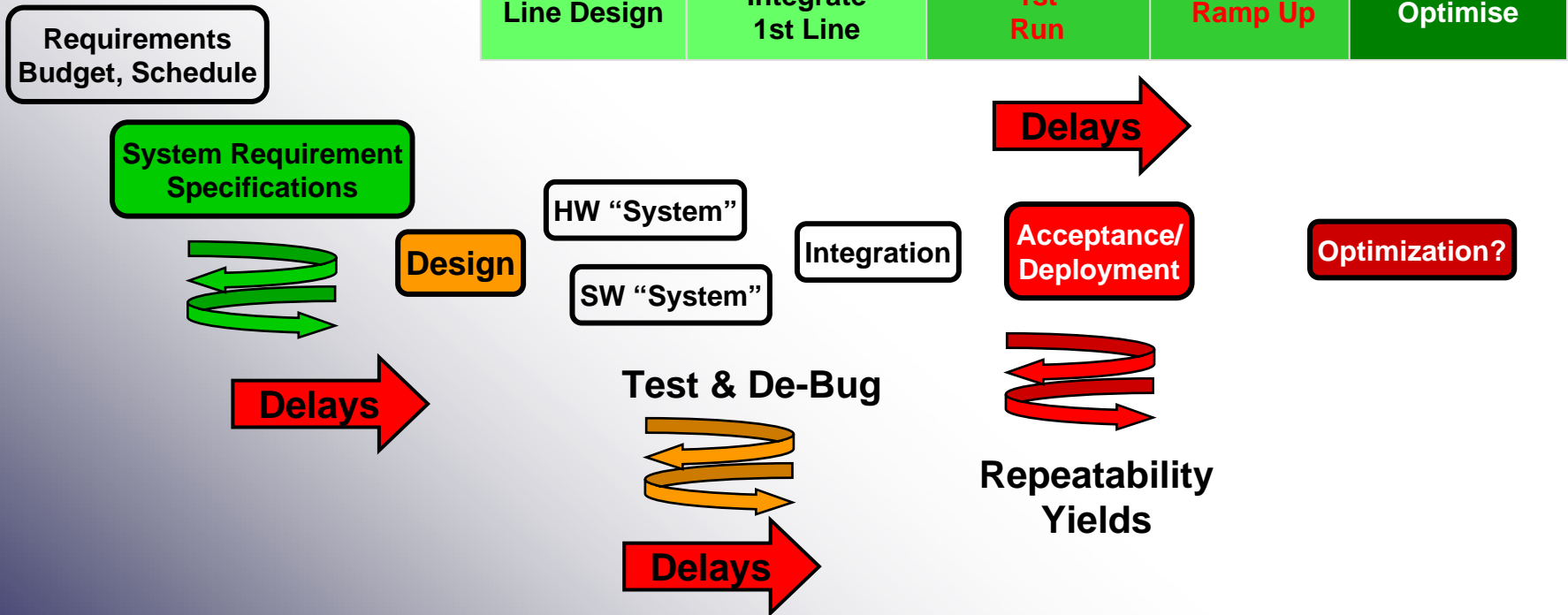
A quick look at a Test Development (Aka Test System Project)

A common implementation

A basic R&D Cycle



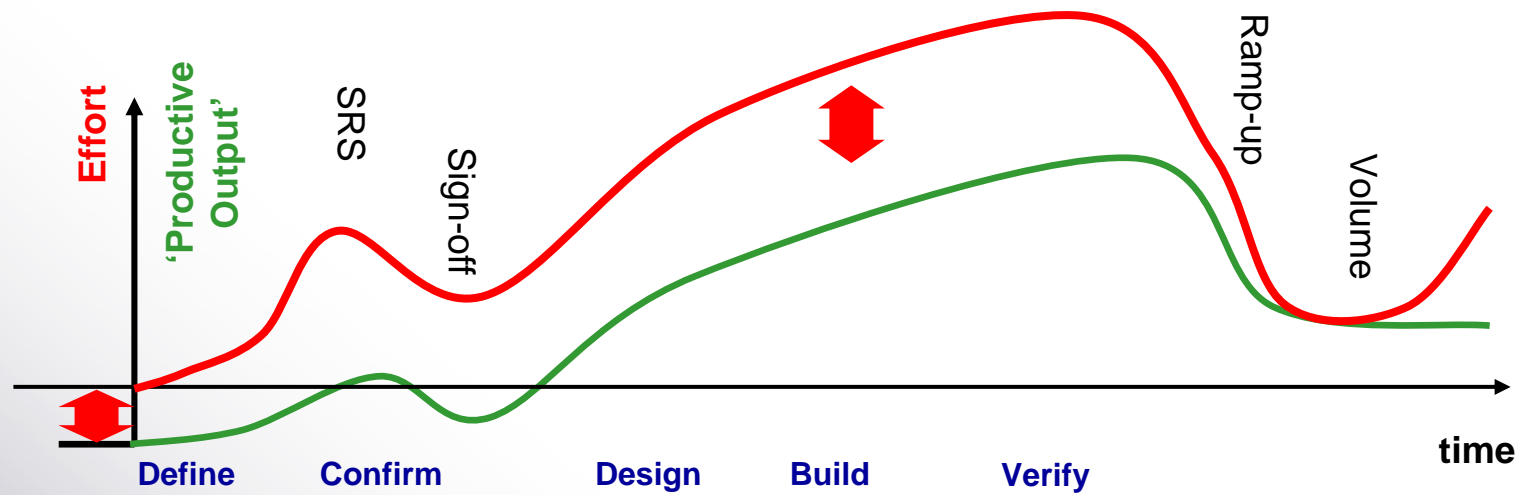
A (very) generic Manufacturing Cycle



What sometimes go wrong...

- Over budget, over schedule
- Unforeseen requirements, > changes
- Unplanned work, effort > delays
- Yields, throughput not achieved
- Costly delays, fixes
- **Two key concerns:**
- **Impact on manufacturing**
- **It happens again**

A qualitative view



- Three observations:
- Negative output, e.g. changes, 'fixing'
- Effort vs. Output
- Post deployment effort: Optimization or fix?

What are the underlying
issues, the root causes?

Some root causes

- Lack of time, skills, resources
- Diverse, complex technologies
- Multiple projects at the same time
- Non recurrent engineering (NRE)
- Time, schedule pressure
- Oversight/misunderstood requirements
- Planning effort, tasking, project-plan
- **Others?**

Critical areas

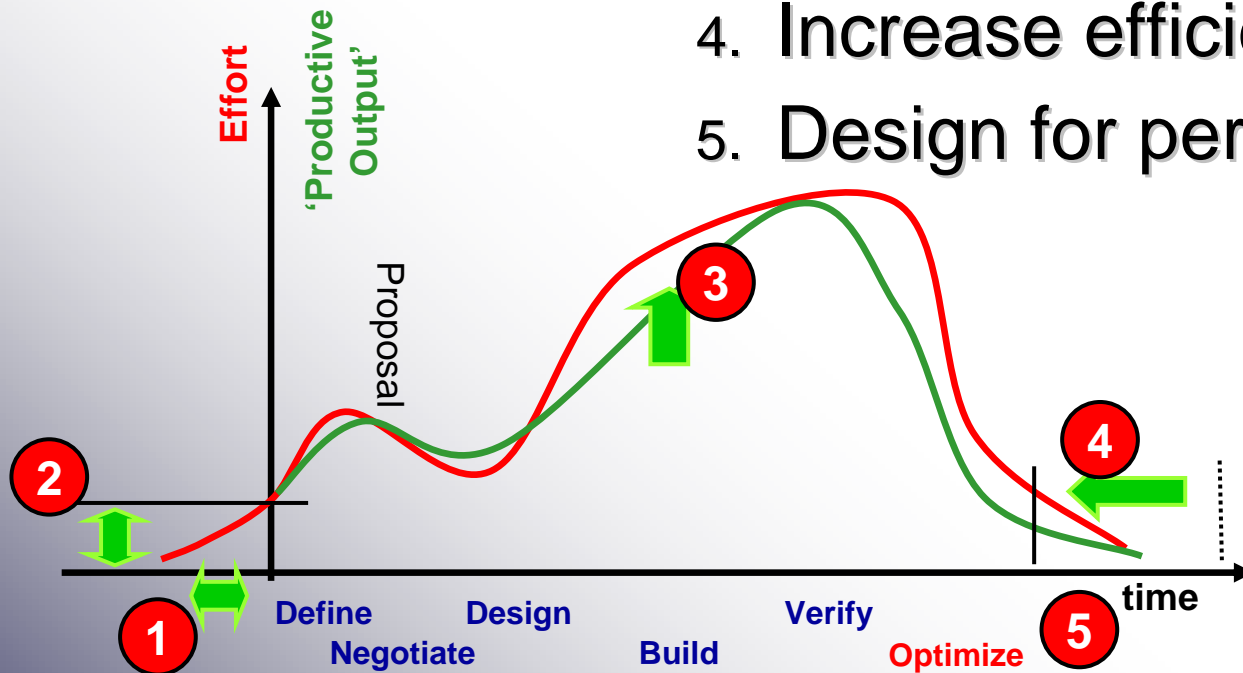
- Specification Phase
- Engineering Re-Use/NRE
- Training
- Preparation

So... what can we improve?

- Knowledge, skills, expertise
- Preparedness
- Better linkage of processes
- Better re-use of engineering
- **Important: Aligned with Test System Engineering process/operation**

Goal

1. Increase level of preparedness
2. Minimize changes, fixes
3. Increase productivity
4. Increase efficiency
5. Design for performance



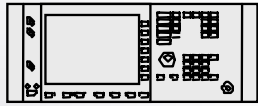
How?

- Integrate, align training
- Modular, common, pre-defined engineering process & structure
- Design and build for re-use
- Re-useable, practical templates
- Measure engineering performance, feed-back & adjust

Preparedness

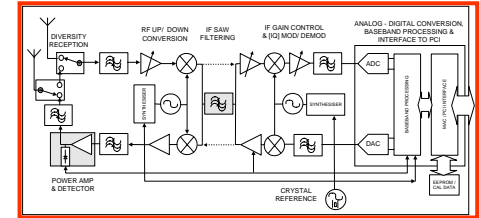
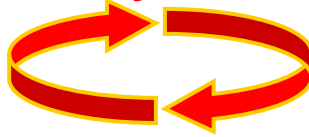
'Test' Technology

Device Technology



- Measurements
- Automation & Data
- Capacities

Align Training Cycles

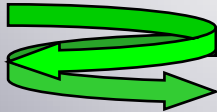


- Fundamentals
- DUT design
- Signals/Measurements
- Interfaces, Adjustments
- etc.

'Test Need'

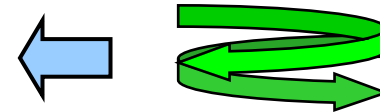
System Requirement Specifications

Solution "Fit"



Minimize Iterations
Minimize Delays
Improve performance

- System Configuration
- Test Capabilities
- Test Capacities



Minimize iterations
Minimize Delays
Reduce bugs

Benefits

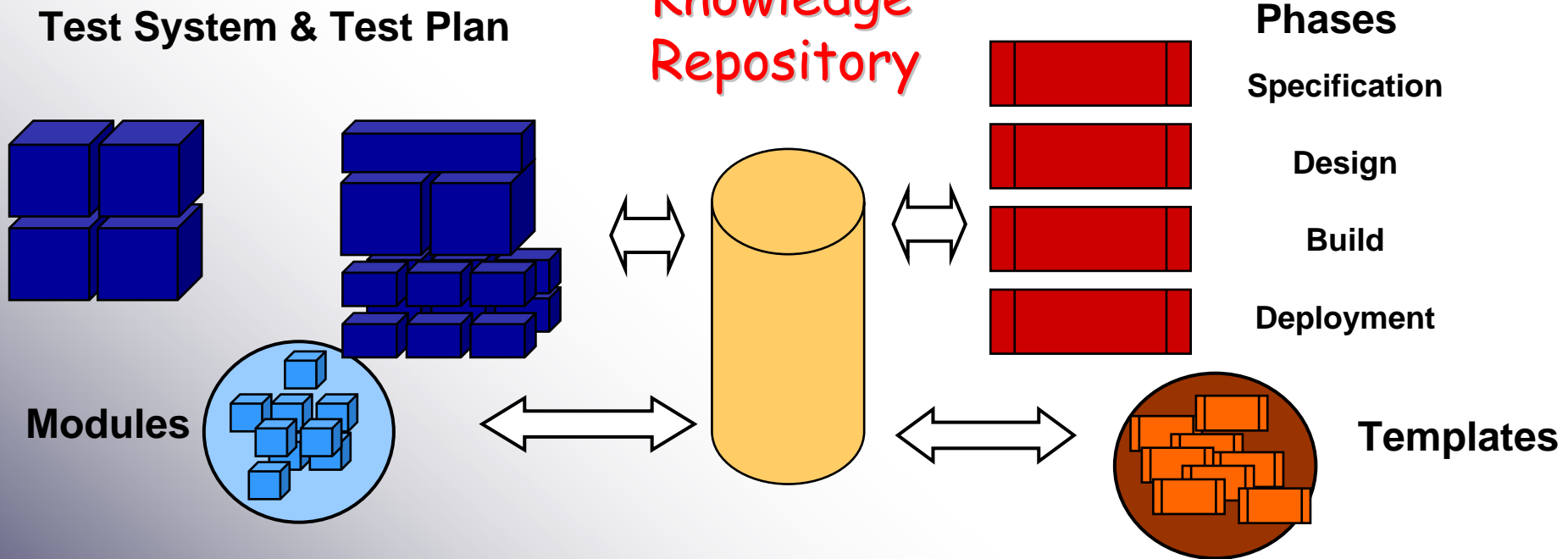
- Efficiency and success rate of requirement specification
- Accuracy of planning, costing
- Reduced lead times
- Reduce risk
- Minimize iterations, fixes, 'bugs'
- Early focus on testing and performance
- Quality of test design
- Improve ROI from training

Structured engineering

Solution Structure

+

Solution Process



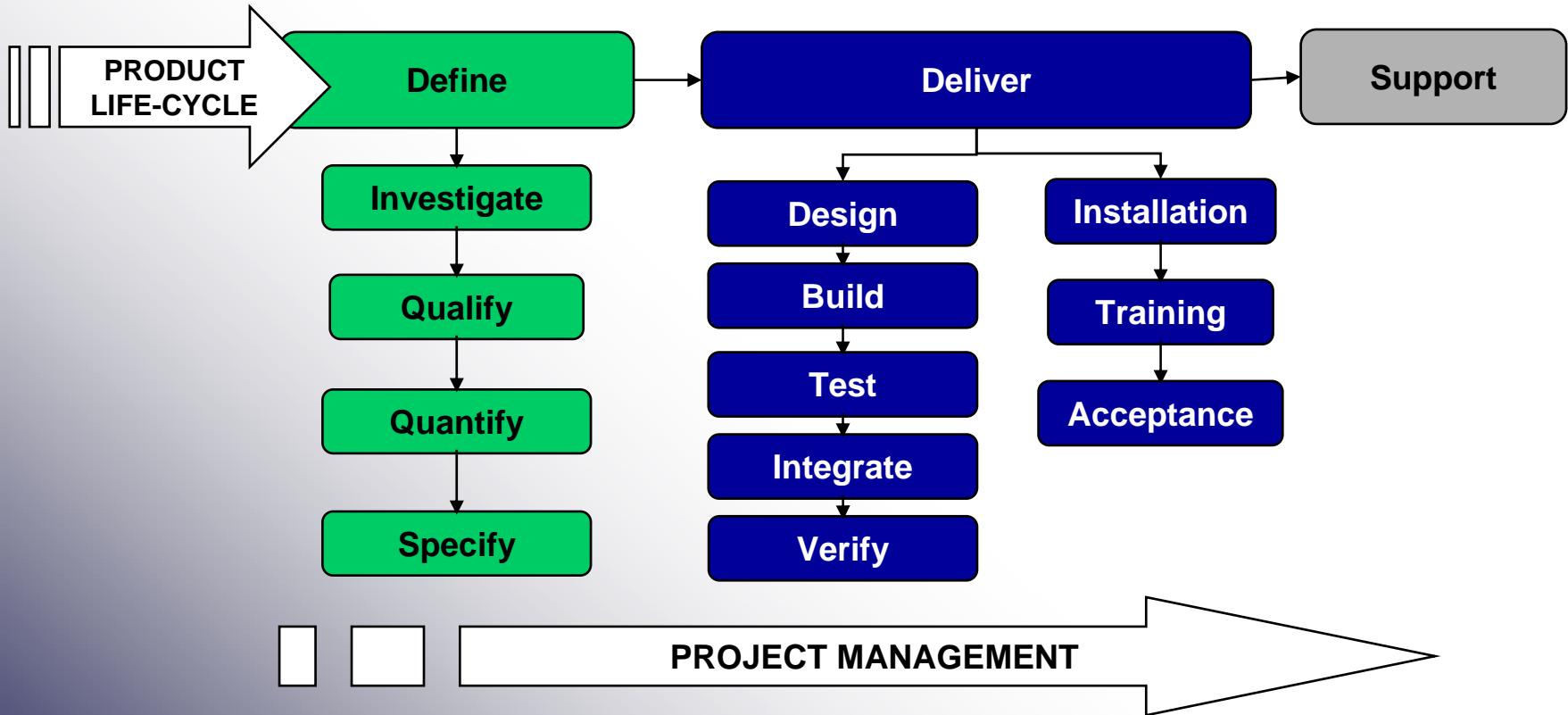
The integration of a common Engineering Process, modular Test System implementation and Knowledge Management embedded within the parent operation

Benefits Structured Engineering

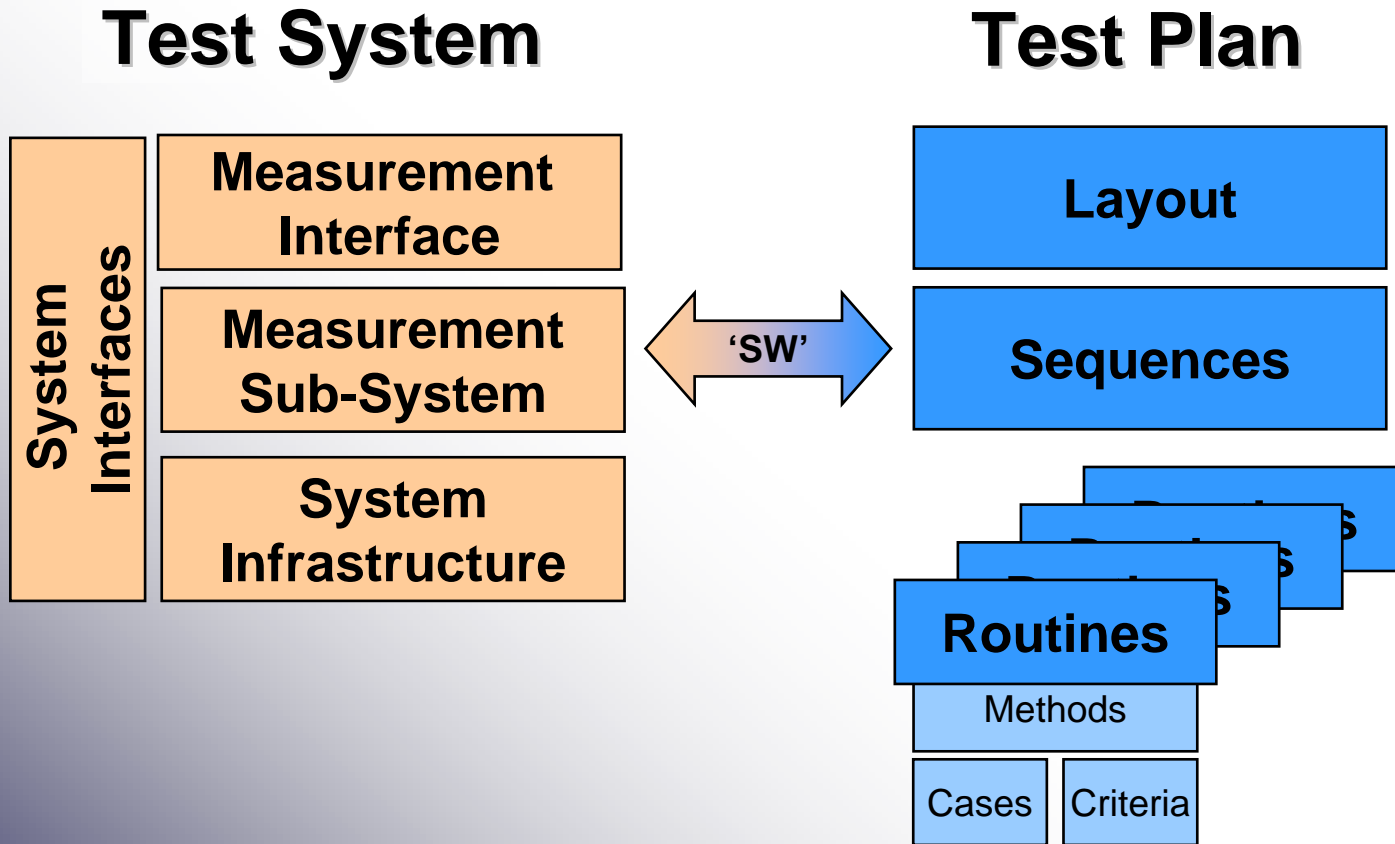
- Reduce specification, design time
- Minimize NRE, Maximize productivity
- Shorten product life-cycle, ramp-up
- Performance, Yield, Throughput
- Reliability
- Skill development
- Continuous Improvement

Structured (Test) Engineering

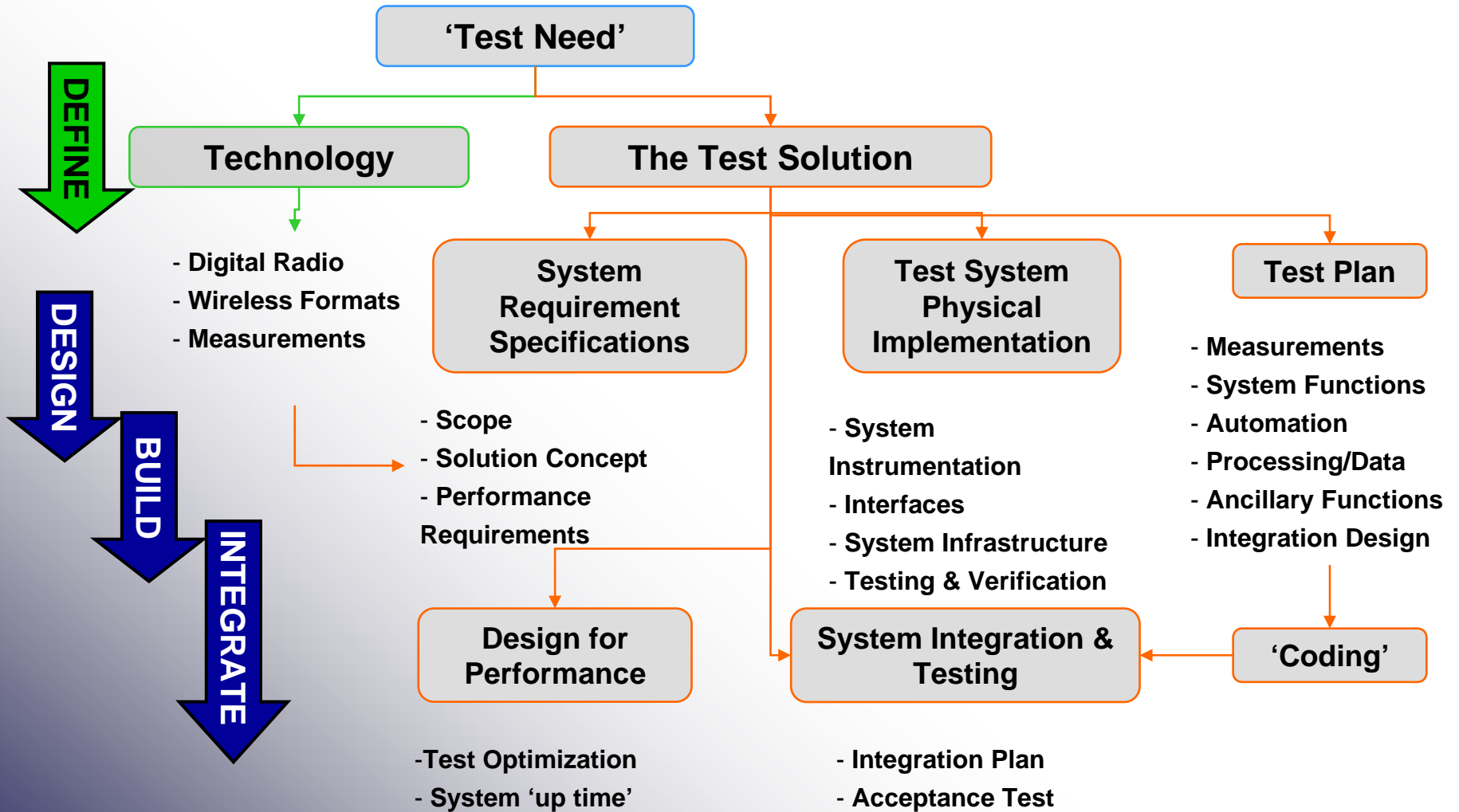
A basic Solution Process



A basic Solution structure

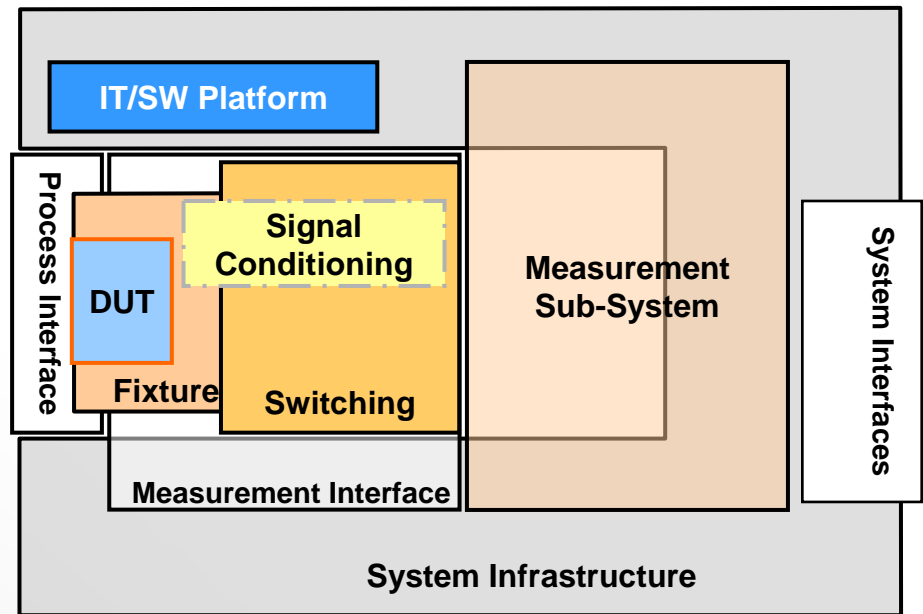


Align Process & Structure (Example: Wireless)

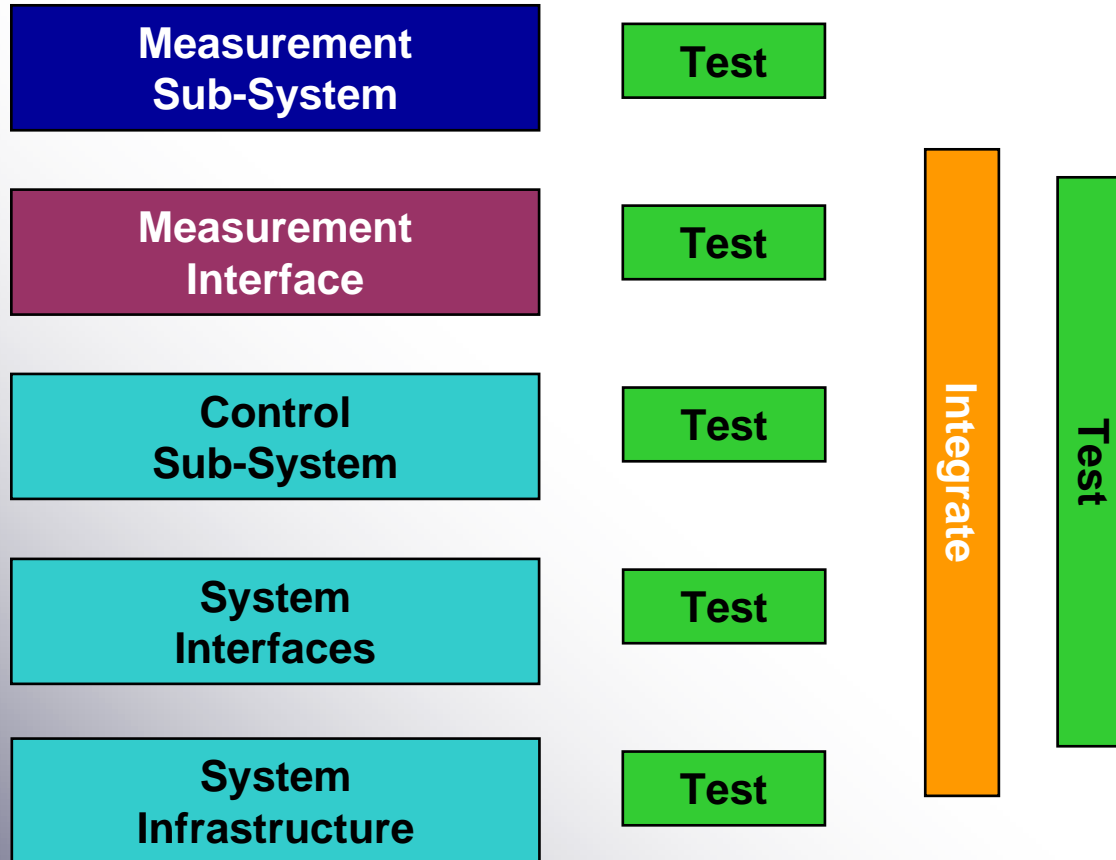


The 'physical' System

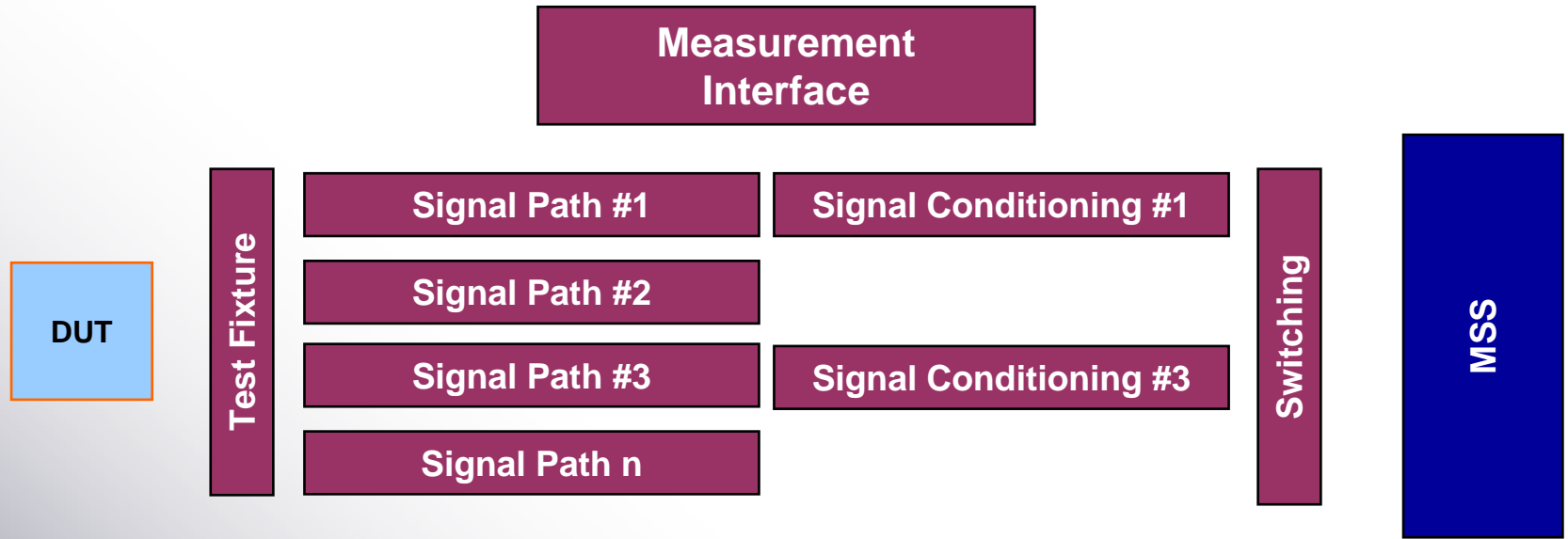
- Measurement Sub-System
 - Core Test Instruments
 - Ancillary Test Instruments
- Measurement Interface
 - Fixture
 - Switching
 - Signal Conditioning
- Interfaces
 - Process
 - System
- System Infrastructure
 - IT
 - Mechanical, Electrical, Environmental



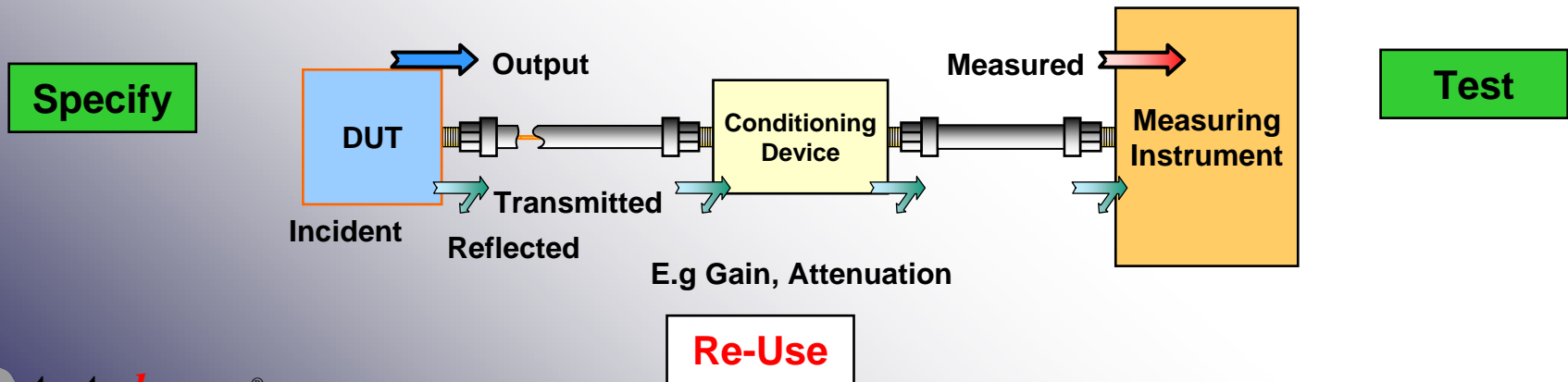
A Modular Test System (HW)



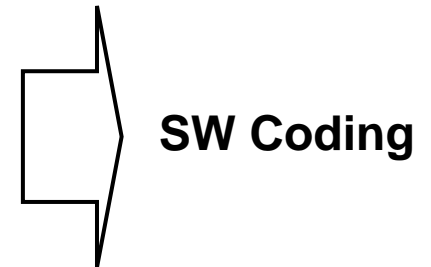
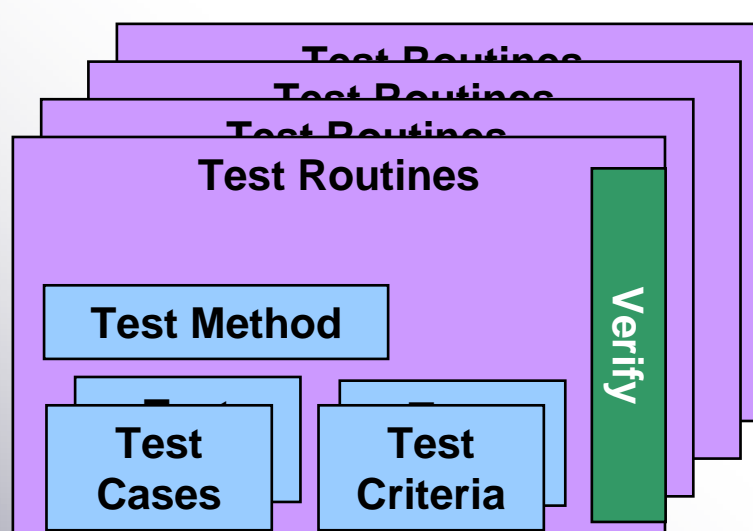
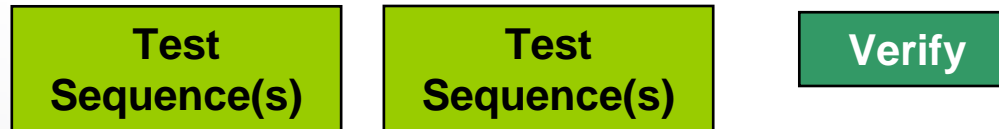
Example of System modularity



Allows (sensible) encapsulation of sub-assemblies

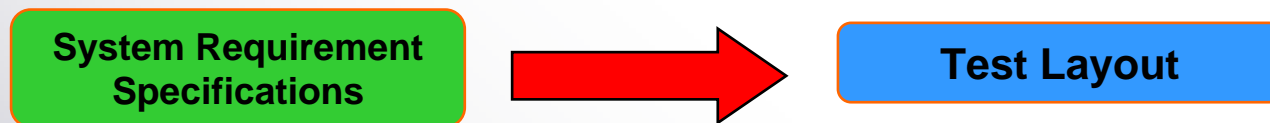


A Modular Test Plan



Test Layout

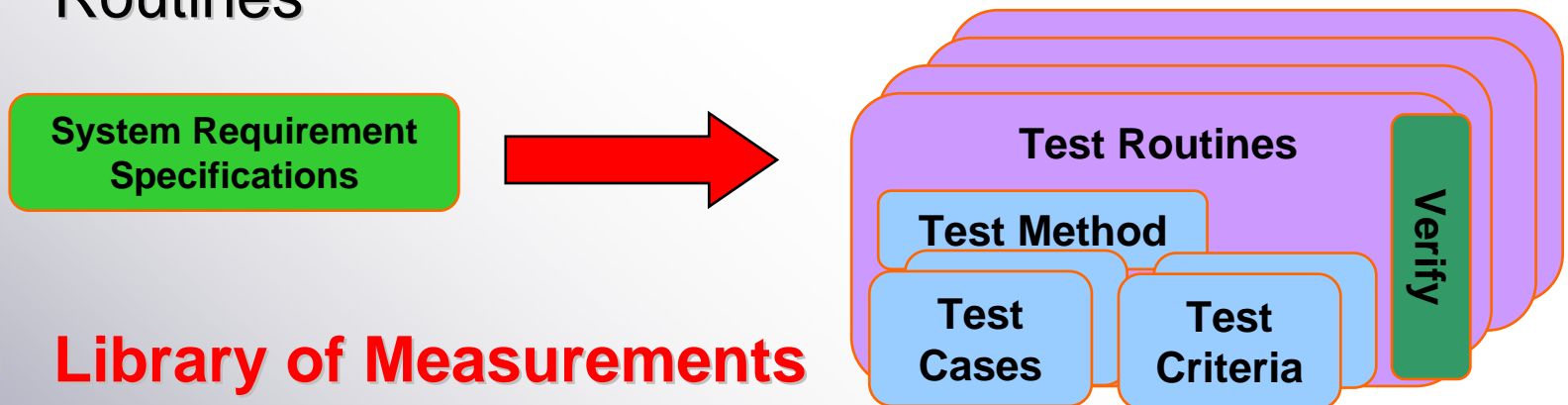
- Driven by Target/Test Process
 - Process stages
 - DUT build stages and prerequisites
- Capacity Requirements
 - Balance of Tests across stages



- **Outline of Test distribution**

Measurement Test Routines

- From the Test List of specified measurements
 - Test Methods (descriptive, e.g. Sensitivity meas.)
 - Test-Cases (Parameters, Instrument set-ups)
 - Test-Criteria (Limits, etc.)
- 'Manual' verification of Measurement Test Routines

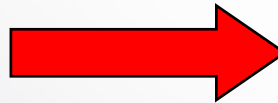


- **Library of Measurements**

System Routines

- DUT Set-up, System Set-up
- System Control Routines, e.g. Environmental
- Process Control Routines, e.g. Bar Code readers
- Interface Routines – Measurement, Control, System, Switching, Fixture Control, etc.
- Data management, SPC, Operations Systems
- Ancillary Functions, e.g. Reference signals, Safety, System and measurement checks
- Optimization Utilities, raw-data, 'stress test', etc.

**System Requirement
Specifications**



A template Test Plan

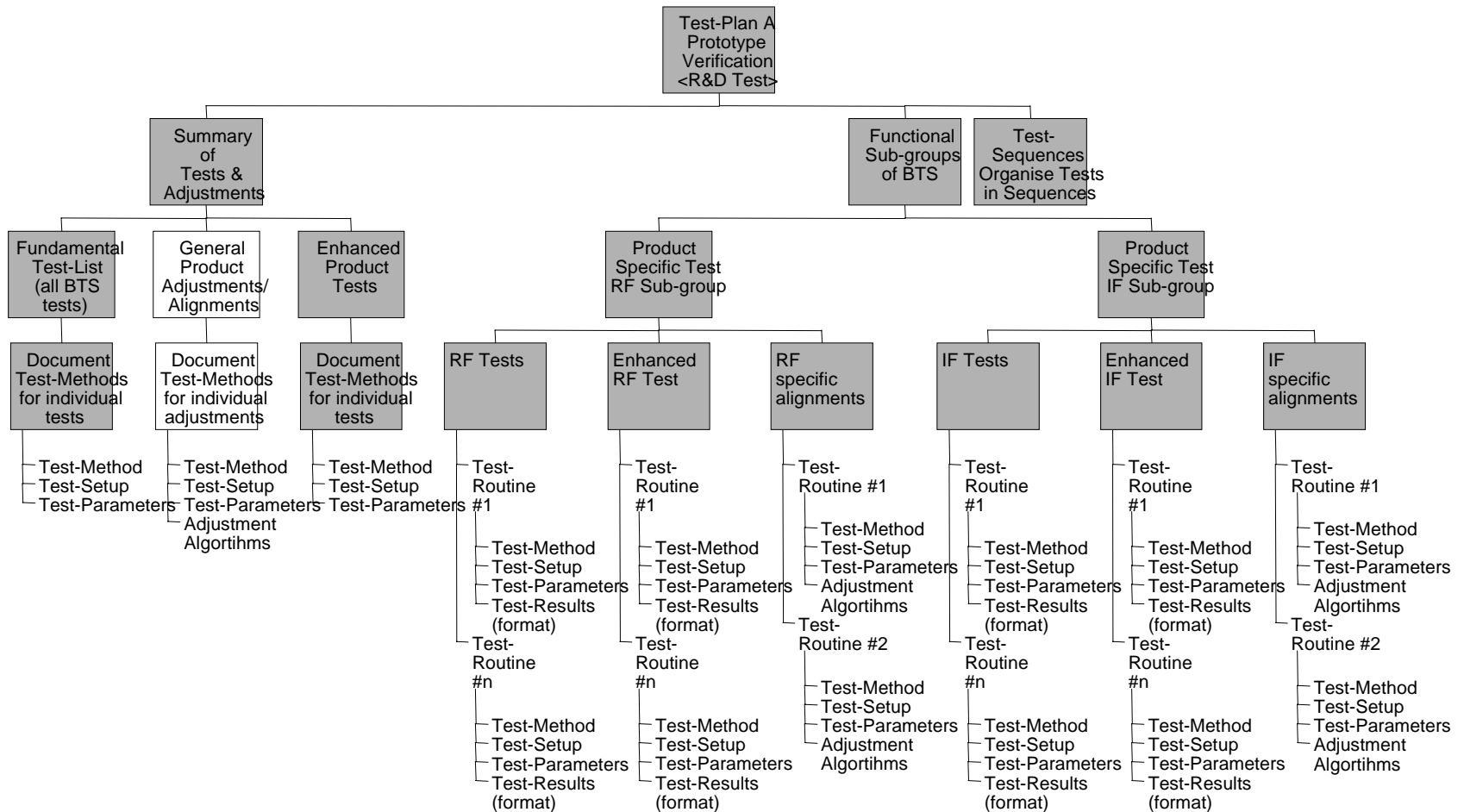
- **Test-Layout**
 - Test-Stages
 - **Test-Sequences**
 - **Test-Routines**
 - Test-Cases, Test Criteria
 - Adjustments Algorithms
 - Interface Control
 - *Test-Routine Results*
 - Manipulation, format
 - *Test Sequence Results*
 - Manipulation, format

Example Test Routine Template

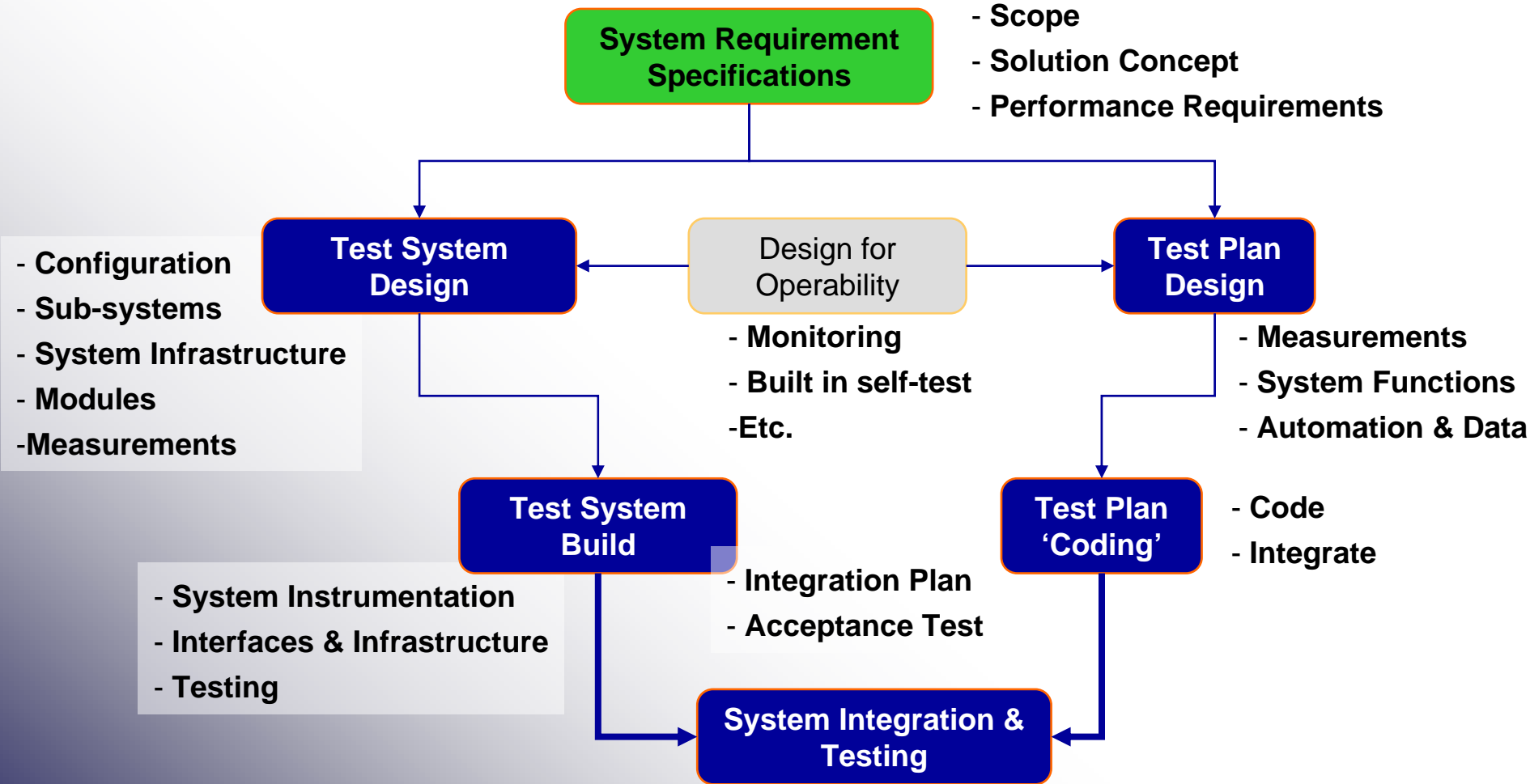
- Test Routine Description
- **Test-Setup**
 - Block-Diagram
- **Test-Method**
 - Test-Case(s)
 - Test-Settings
 - Test-Points
 - Test-Criteria
- **Dependencies**
- **Programming notes**
- **Adjustments, Algorithms**
- Reference Materials

Example Test Plan

Prototype Testing



Promotes Concurrent Engineering



Template of aligned Process & Structure

A template process, part 1

System Requirement Specification

Project Scope – Objectives, Needs and Constraints; Schedule and Budget

Technology and DUT

Target Process

Design Brief

DUT Product Test Specification

Operational and Environmental Requirements

Compiling the Functional Requirements

Functional Design Specification

System Requirements

Measurement Requirements

Measurement Instruments

Measurement Interface

System Infrastructure, Automation, Integration

The System Concept

Performance Requirements

System performance (e.g. SPC interface)

Measurement performance

Operational Performance (Capacity, e.g. throughput)

Compiling the System Requirement Specification (SRS)

Performance criteria for all HW/SW

Complete System Infrastructure

Environmental, Electrical, Mechanical Specifications

Deployment

Acceptance Testing Criteria

A template process, part 2

System Design

- Measurement Sub-System
- Measurement Interface
- DUT Interface
- Routing, Switching and Conditioning
- Control Sub-System
- Information Interface
- System Infrastructure
- System Interface (Environment and Process)
- Configuring the Measurement Sub-System

System Build

- Measurement Sub-System
- Testing the Measurement Sub-System
- Building the Measurement Interface
- Testing the Measurement Interface
- Control Sub-System
- Testing the Control Sub-System
- Testing Control Interface
- System Infrastructure
- Testing the System Infrastructure
- System Interfaces
- Testing the System Interfaces
- Integration
- System Testing and Verification

A template process, part 3

Test Plan Design & Build

- Extracting executable Test Requirements from the SRS

- Adapting to Measurement/HW Configuration

- Extracting Process, Information and Control Requirements

- Defining the Test Layout

- Defining the Test Sequence(s)

- Designing DUT Product Tests as Test Routines

 - Specifying Test Methods, Test Cases and Test Criteria

 - Creating Measurement Test-Routines

 - Defining and Creating Set-Up routines

 - DUT Set-up

 - System Set-up

 - Defining and Creating System Control Routines

 - Defining and Creating Process Control Routines

 - Defining and Creating Interface Routines – Measurement, Control, System

 - Defining and Creating Data management

 - Defining and Creating Ancillary Functions

 - Creating the Test Sequence(s)

Plan of testing / verification

- Implementation Design/SW Development

- Testing individual Routines

- Integration

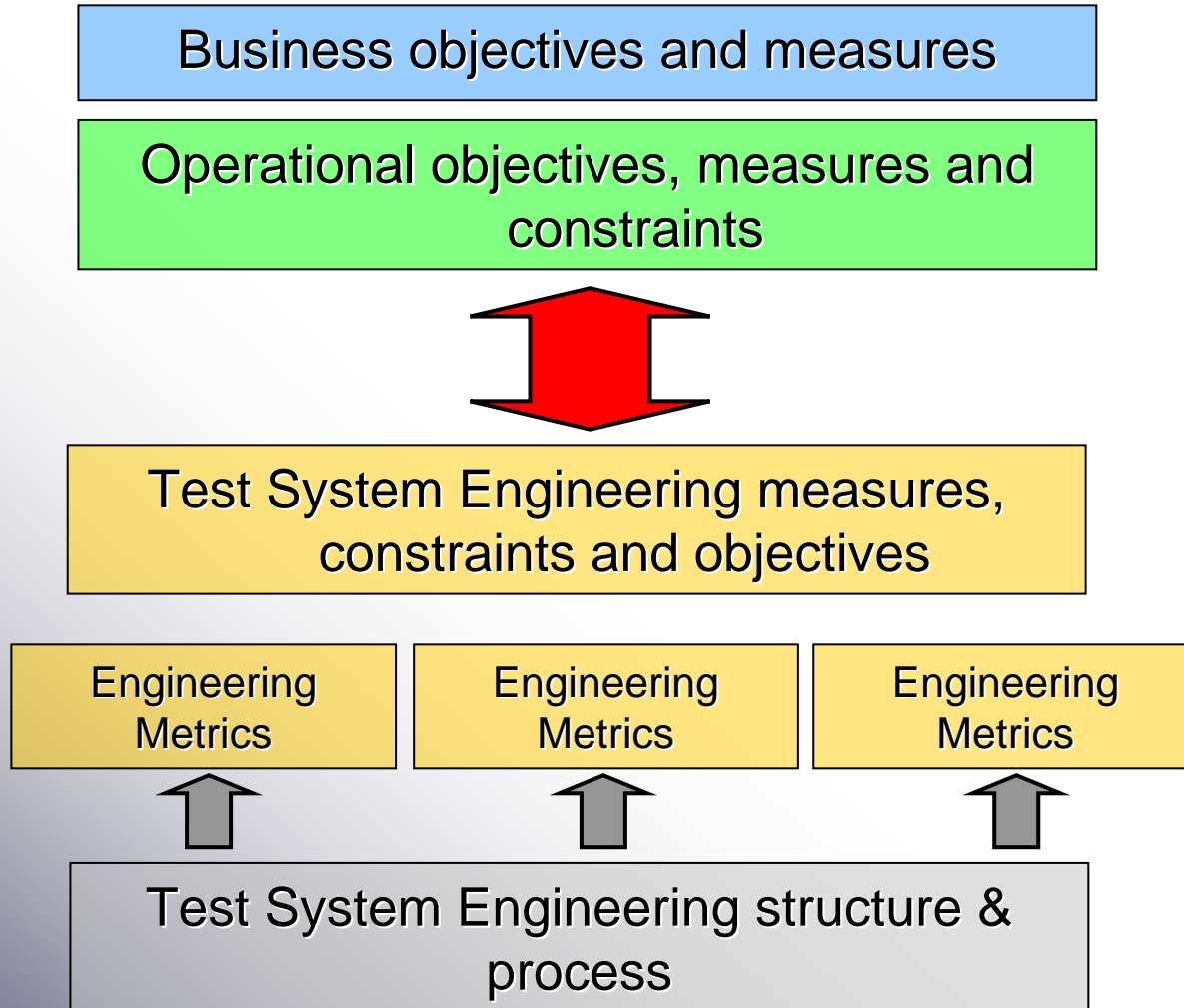
- Test and Verification

Operational Benefits

- Planning: Provides default WBS
- Improves costing, scheduling
- Reduces lead times
- Project Management: Default Project Plan
- Eases Change Management
- Provides template for project steps, e.g. SRS
- Provides default structure for Knowledge Management
- Provides structure for review
- Enhance team work
- *And many others...*

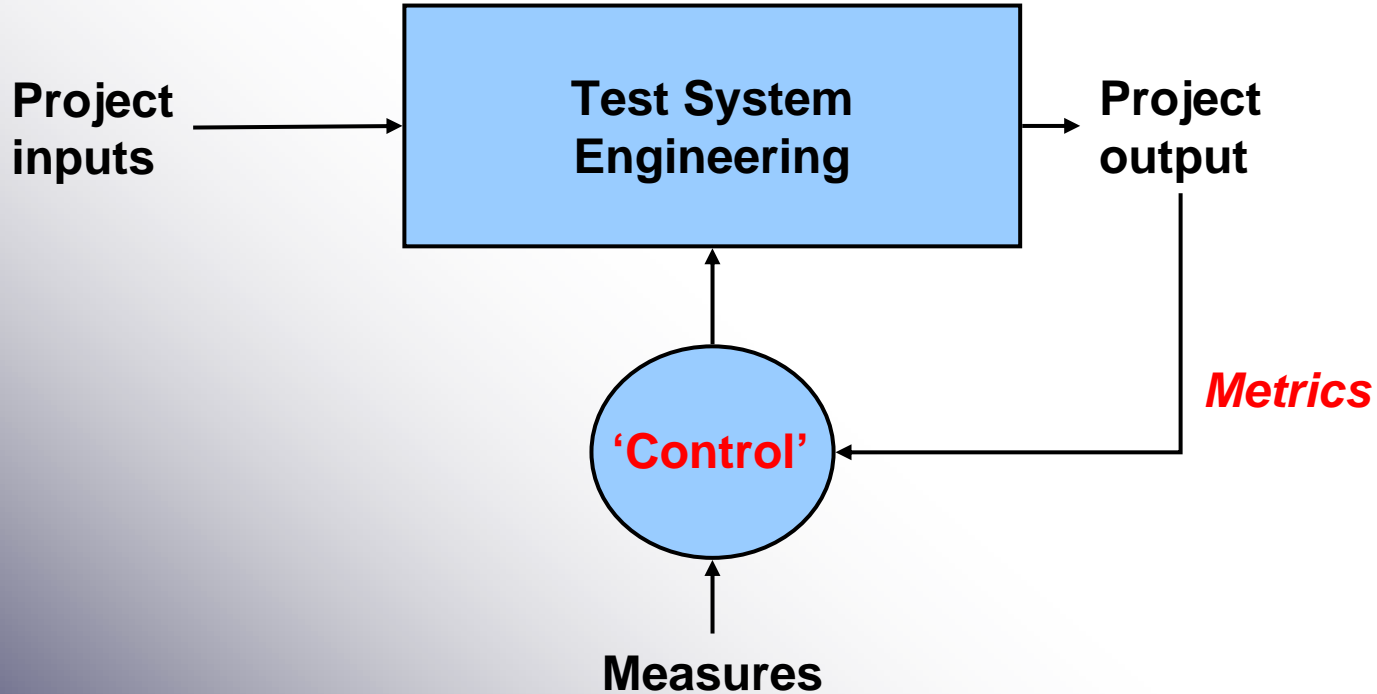
Measuring performance of Test Engineering

Principle Approach



Improving Engineering

- Controlling the process



Some Engineering Metrics

- Engineering Effort
- Efficiency: Productive Output/Effort
- Effectiveness: Re-useable Output/Effort
- Quality: Re-Work/Effort
- Non Recurrent Engineering (NRE):
Productive Output – Re-Useable output
- Workable metrics
- Use module costing as basis for metrics

Evolutionary Engineering

- Each project:
- Is reviewed
- Contributes a residual improvement to the test-department (process, structure)
- Provides a tangible value in new IP
- Uses less NRE than the previous project
- Drives training, preparation

Evolution vs. Optimization

- Not a single 'master plan' approach
- Successive, planned, targeted and measured improvements, based on areas requiring improvement
- Capture, retain, capitalize knowledge
- Focused preparation/training
- Common, use-able engineering structure
- Incremental change
- Sustainable, transferable improvement

**Thank you for your time
and interest!**

www.testadvance.com