Abstract

This course touches all fundamentals of Systems Engineering. Topics are programs, projects, strategy and operation, stakeholders and concerns, needs elicitation and requirements management, concept selection, architecting and design, supply chain, risk management, systems integration, verification and validation, deployment, and life cycle.

Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.
Abstract

Course overview of the course Systems Engineering Fundamentals.
### Systems Engineering Fundamentals Course Overview

**Course Overview**

**Day 1**
- Course intro
- Systems engineering intro
- Case discussion

**Day 2**
- System life cycle supporting systems
- Sketch system life cycle
- Needs and requirements
- Identify needs and capabilities

**Day 3**
- Concept selection
- Perform concept selection
- Architecture and design
- Show dynamic behavior

**Day 4**
- Supply chain and logistics
- Sketch goods flow
- Risk management
- Assess risks

**Day 5**
- Verification and validation
- Project management
- Transform sequence into a PERT plan
- Deployment
- Sketch installation and commissioning

**Schedule**

<table>
<thead>
<tr>
<th>Time</th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00</td>
<td>Course intro</td>
<td>System life</td>
<td>Concept</td>
<td>Supply chain</td>
<td>Verification</td>
</tr>
<tr>
<td></td>
<td>Systems</td>
<td>cycle</td>
<td>selection</td>
<td>and logistics</td>
<td>and validation</td>
</tr>
<tr>
<td></td>
<td>Engineering</td>
<td>supporting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Intro</td>
<td>systems</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:00</td>
<td>Case discussion</td>
<td>Sketch system</td>
<td>Perform</td>
<td>Sketch goods</td>
<td>Transform</td>
</tr>
<tr>
<td></td>
<td></td>
<td>life cycle</td>
<td>concept</td>
<td>flow</td>
<td>sequence into</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>selection</td>
<td></td>
<td>a PERT plan</td>
</tr>
<tr>
<td>11:00</td>
<td>Company context</td>
<td>Needs and</td>
<td>Architecture</td>
<td>Risk</td>
<td>Deployment</td>
</tr>
<tr>
<td></td>
<td>Programs,</td>
<td>requirements</td>
<td>and design</td>
<td>Management</td>
<td>Sketching</td>
</tr>
<tr>
<td></td>
<td>Projects</td>
<td></td>
<td></td>
<td></td>
<td>installation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Dynamic behavior, functionality</td>
<td>Assess risks</td>
<td>and commissioning</td>
</tr>
<tr>
<td>12:00</td>
<td>Lunch</td>
<td>Lunch</td>
<td>Lunch</td>
<td>Lunch</td>
<td>Lunch</td>
</tr>
<tr>
<td>13:00</td>
<td>System</td>
<td>Reflection and</td>
<td>Reflection and</td>
<td>Reflection and</td>
<td>Wrap-up</td>
</tr>
<tr>
<td></td>
<td>development</td>
<td>discussion</td>
<td>discussion</td>
<td>discussion</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Process</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14:00</td>
<td>Sketch a typical</td>
<td>Requirements</td>
<td>Reflection and</td>
<td>Systems</td>
<td>Reflection and</td>
</tr>
<tr>
<td></td>
<td>mission and a</td>
<td>management</td>
<td>discussion</td>
<td>integration</td>
<td>discussion</td>
</tr>
<tr>
<td></td>
<td>scenario</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Partitioning</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>and interfaces</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:00</td>
<td>Identify</td>
<td>Determine 10</td>
<td>Make system</td>
<td>Determine an</td>
<td></td>
</tr>
<tr>
<td></td>
<td>stakeholders</td>
<td>SMART KPPs and</td>
<td>and work</td>
<td>incremental</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and concerns</td>
<td>use case</td>
<td>breakdown</td>
<td>integration</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>sequence</td>
<td></td>
</tr>
<tr>
<td>16:00</td>
<td>Reflection and</td>
<td>Reflection and</td>
<td>Reflection and</td>
<td>Reflection and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>discussion</td>
<td>discussion</td>
<td>discussion</td>
<td>discussion</td>
<td></td>
</tr>
</tbody>
</table>
Abstract

Listing the course material for the course Systems Engineering Fundamentals.
Introduction

**core**

Systems Engineering Fundamentals Introduction

http://gaudisite.nl/info/SEFintroduction.info.html

**optional**
### Course Overview

**core**

Systems Engineering Fundamentals Course Overview

http://gaudisite.nl/info/SEFoverview.info.html

**optional**
Assignments

**core**

Systems Engineering Fundamentals Assignments

http://gaudisite.nl/info/SEFassignments.info.html

**optional**
<table>
<thead>
<tr>
<th>Programs, Projects, Process, Organization</th>
</tr>
</thead>
</table>

**core**

Project Systems Engineering Introduction; Phasing, Process, Organization  
http://gaudisite.nl/info/ProjectSEintroPPO.info.html  
Module System Architecture Context  
Products, Projects, and Services; similarities and differences in architecting  
https://gaudisite.nl/ProductsProjectsServicesPaper.pdf

**optional**

System Engineering Management Plan (SEMP) DOES ONE SIZE FIT ALL?  
Zonnenshain, A., Malotaux, N., Honour, E., Kasser, J., Urio, U., Shabtay, M., INCOSE 2009  
Systems Engineering Management Plan (SEMP) Technical Content  
**Life Cycle**

- **core**
  - Systems Engineering Fundamentals Life Cycle
    - [http://gaudisite.nl/info/SEFlifeCycle.info.html](http://gaudisite.nl/info/SEFlifeCycle.info.html)
  - Modeling and Analysis: Life Cycle Models

- **optional**
  - SEBoK Life Cycle models
    - [https://www.sebokwiki.org/wiki/Life_Cycle_Models](https://www.sebokwiki.org/wiki/Life_Cycle_Models)
Needs and Requirements

**core**

Systems Engineering Fundamentals Needs Elicitation
http://gaudisite.nl/info/SEFneeds.info.html

**optional**

SEBoK Stakeholder Needs and Requirements
https://www.sebokwiki.org/wiki/Stakeholder_Needs_and_Requirements
<table>
<thead>
<tr>
<th>Requirements Management</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>core</strong></td>
</tr>
<tr>
<td>Systems Engineering Fundamentals Requirements Management</td>
</tr>
<tr>
<td><a href="http://gaudisite.nl/info/SEFrequirements.info.html">http://gaudisite.nl/info/SEFrequirements.info.html</a></td>
</tr>
<tr>
<td>Fundamentals of Requirements Engineering</td>
</tr>
<tr>
<td><strong>optional</strong></td>
</tr>
<tr>
<td>core</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>Concept Selection, Set Based Design and Late Decision Making</td>
</tr>
<tr>
<td><a href="https://gaudisite.nl/SEFconceptSelectionSlides.pdf">https://gaudisite.nl/SEFconceptSelectionSlides.pdf</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept Selection - Applying Pugh Matrices in the Subsea Processing Domain by Linda Lønmo and Gerrit Muller; INCOSE 2014 in Las Vegas</td>
</tr>
<tr>
<td>Researching the application of Pugh Matrix in the sub-sea equipment industry by Gerrit Muller, Dag Jostein Klever, Halvard H. Bjørnsen, and Michael Pennotti; CSER 2011 in Los Angeles</td>
</tr>
<tr>
<td><a href="https://gaudisite.nl/CSER2011_MullerEtAl_ResearchingPughMatrix.pdf">https://gaudisite.nl/CSER2011_MullerEtAl_ResearchingPughMatrix.pdf</a></td>
</tr>
<tr>
<td>core</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>Visualizing Dynamic Behavior</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creating an A3 Architecture Overview; a Case Study in SubSea Systems by Gerrit Muller, Damien Wee, and Martin Moberg; INCOSE 2015 in Seattle, WA, USA</td>
</tr>
</tbody>
</table>
### Systems Engineering Fundamentals; Course Material

#### Supply Chain and Logistics

<table>
<thead>
<tr>
<th><strong>core</strong></th>
<th>Systems Engineering Fundamentals Supply Chain and Logistics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><a href="https://gaudisite.nl/SEFsupplyChainSlides.pdf">https://gaudisite.nl/SEFsupplyChainSlides.pdf</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>optional</strong></th>
<th>Build to order <a href="https://en.wikipedia.org/wiki/Build_to_order">https://en.wikipedia.org/wiki/Build_to_order</a></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P-D Ratios <a href="https://oldleandude.com/2015/05/27/p-d-ratios/">https://oldleandude.com/2015/05/27/p-d-ratios/</a></td>
</tr>
</tbody>
</table>
core

Systems Engineering Fundamentals Risk Management

optional

Failure Mode and Effects Analysis
https://en.wikipedia.org/wiki/Failure_mode_and_effects_analysis
## Readiness Levels

<table>
<thead>
<tr>
<th>Core</th>
<th>Course Systems Integration; Readiness Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><a href="http://www.gaudisite.nl/info/MSIreadinessLevels.info.html">http://www.gaudisite.nl/info/MSIreadinessLevels.info.html</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optional</th>
<th>From TRL to SRL: The Concept of Systems Readiness Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CSER 2006, Brian Sauser et al.</td>
</tr>
<tr>
<td></td>
<td>Technology Readiness Levels</td>
</tr>
<tr>
<td></td>
<td><a href="https://en.wikipedia.org/wiki/Technology_readiness_level">https://en.wikipedia.org/wiki/Technology_readiness_level</a></td>
</tr>
<tr>
<td><strong>core</strong></td>
<td>Mastering Systems Integration; Process and Positioning</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td><a href="http://gaudisite.nl/info/MSIprocessAndPositioning.info.html">http://gaudisite.nl/info/MSIprocessAndPositioning.info.html</a></td>
</tr>
<tr>
<td><strong>optional</strong></td>
<td>SESA /SARCH Module 01, System Architecture Context</td>
</tr>
</tbody>
</table>
### Core

Course Systems Integration; Project Management

http://gaudisite.nl/info/MSIprojectManagement.info.html

### Optional

Combating Uncertainty in the Workflow of Systems Engineering Projects

INCOSE 2013, Barry Papke and Rick Dove
### Verification and Validation Terminology

<table>
<thead>
<tr>
<th>Core</th>
<th>Course Systems Integration; Terminology</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><a href="http://www.gaudisite.nl/info/MSIterminology.info.html">http://www.gaudisite.nl/info/MSIterminology.info.html</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optional</th>
<th>Understanding Objective Evidence: (What It Is and What It Definitely Is Not), by Denise Dion</th>
</tr>
</thead>
</table>
Abstract

All assignments of the course Systems Engineering Fundamentals.
Propose a Non-Lethal Urban Crowd Controller
Discuss the Case

Sketch the **system-of-interest**

Sketch some of the **environment** the system will be operating in

Sketch some of the **system internals**

Draw the **system boundary**
Map the Operational Organization

Make a map with names of individuals in the **operational organization** of one project and its context.

Identify the **relationships** of the **project core team**:

- geographical
- organizational
- psychological
Sketch a typical mission and a specific scenario.

The scenario needs to be highly specific:

- numbers (how much, how far, how accurate)
- names (where, who)
- circumstances (when, where)
- actions (what, how)
Identify Stakeholders and Concerns

Brainstorm **stakeholders**

Brainstorm for each stakeholder the **concerns**

Elaborate concerns in 5 to 10 words, make them more specific

Use the **mission** and **scenario** for inspiration
Sketch the system *life cycle*

from idea until decommissioning and recycling.

Identify **stakeholders** per phase or activity
Identify **stakeholder needs**

in terms of **capabilities**.

Capabilities typically are **functions**

with **quantifiable characteristics**

Use the mission, scenario, and stakeholder analysis for inspiration
Determine 5 to 10 *Key Performance Parameters* (KPP) of the System

Quantify these KPPs

Define the KPPs roughly, using a *Use Case*
Perform a Concept Selection

Make a **decision matrix** for one of the **concept selections**.

- define at least 3 concepts
- define 7 to 10 criteria for selection
- score the concepts against the criteria, for example using a scale from 1 to 5: 1 = very poor, 5 = very good
- recommend a concept with a rationale

<table>
<thead>
<tr>
<th></th>
<th>concept 1</th>
<th>concept 2</th>
<th>concept 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>criterion 1</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>criterion n</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

**best, because ...**
Show Dynamic Behavior

Model the Dynamic Behavior of the System.

Focus on the Dynamic Behavior that relates to the KPP.

Visualize the Dynamic Behavior with various sketches, diagrams, or graphs (see Visualizing Dynamic Behavior for inspiration).
Make a *system breakdown*

in subsystems and subsubsystems

and a *work breakdown structure*

to assist in organizing the project
Sketch the Goods Flow

sketch the *goods flow*

from (sub) *suppliers*

via *assembly* and *test*

to *customer site*,

*deployment*,

and *maintenance*
Assess risks

- *feasibility* of achieving *KPPs*
- *fitness for purpose* in customer context
- *integration configurations* and *testware*
- *supplier* and *logistics* status
- *technology readiness*
- *development* and *resource* status

Determine *probability* and *severity* per risk
Determine an incremental integration sequence to build confidence in the KPP ASAP.

Strive for about 6 main increments.

Reason starting at the end result and then backward in time.

For each increment determine its prerequisites in terms of parts, interfaces, functions, and performance levels.
Transform the integration sequence and the planning from the other perspectives into a **PERT-plan**.

A PERT-plan focuses on **activities** and their mutual **relations**; the logic of the plan. Time and resources are secondary information.
Sketch an installation and commissioning
Abstract

This presentation introduces the ideas behind the course Systems Engineering Fundamentals.
Architecture Top View

- customer value proposition
- business proposition
- system requirements
- system design & technology

Drives and enables relationships between the components.
Architecting Playing Field

organizational context
- customer organization
- business organization
- developing organization
- supplying organizations

operational and lifecycle context
- customer value proposition
- business proposition
- drives
- enables

system requirements
- drives
- enables

system design
technology
Simplified Systems Engineering Model

inputs
- stakeholder needs
- business objectives

architecting and design

specification

integration verification & validation

artifacts
- models
- prototypes
- parts

qualification
- evidence

documentation
- system and parts data
- procedures

life cycle support

feedback

architectures
- guidelines
- top-level design
- rationale

design
- partitioning
- interfaces
- functions
- allocation

evidence
artifacts
models
prototypes
parts

evidence
artifacts
models
prototypes
parts

evidence
artifacts
models
prototypes
parts

evidence
artifacts
models
prototypes
parts

version: 0
April 6, 2019
MSIINdefinition

Systems Engineering Fundamentals Introduction
Gerrit Muller
Level of Abstraction Single System

- Static system definition
- Monodisciplinary
- Number of details
- System requirements
- Multidisciplinary design
- Static system definition
- Monodisciplinary
From system to Product Family or Portfolio

The diagram illustrates the increase in complexity from a single system to a portfolio of systems. The number of details increases significantly as we move from a monodisciplinary system to a multidisciplinary system and then to a portfolio of systems. This is represented by the logarithmic scale on the y-axis, indicating the exponential increase in complexity.
Product Family in Context

- Parts, connections, lines of code
- Multidisciplinary design
- Systems
- Stakeholders
- Enterprise
- Enterprise context

Number of details:

- $10^9$
- $10^6$
- $10^3$
- $10^0$
- $10^9$
Capturing all information that is required for: logistics, manufacturing, legislation, maintenance, life cycle support.
from needs and requirements to design: decomposition, interface definition, allocation, concept selection, technology choices

anticipating engineering needs and constraints
Architecting: realization and design choices in context

- Some context details are essential
- Some technical details are essential

Number of details:
- $10^9$
- $10^6$
- $10^3$
- $10^0$
- $10^3$
- $10^6$
- $10^9$

Architecting

- Enterprise context
- Enterprise
- Stakeholders
- Systems
- Multidisciplinary design
- Parts, connections, lines of code
Project Systems Engineering Introduction; Phasing, Process, Organization

by Gerrit Muller University of South-Eastern Norway-NISE

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

The fundamental concepts and approach to project oriented Systems Engineering are explained. We look at project phasing, phase transition, processes, and organization.
Phased Project Approach

Legend:
- core information in draft
- 50% information
- most information available in concept
- information is stable enough to use heavier change control

Presentation of content and legend:
- Needs
- Specification
- Design
- Verification
- Engineering

Legend:
- full under development
- preparing or updating work
V-Model

needs

specification

system design

subsystem design

component design

component realization

validation

verification

system test

subsystem test

component test
All Business Functions Participate

0. feasibility
1. definition
2. system design
3. engineering
4. integration & test
5. field monitoring

sales
logistics
production
service
development & engineering: marketing, project management, design
Evolutionary PCP model

- Test and evaluate
- Requirements specification
- Design
- Build

2% of budget (EVO)
2 weeks (XP)
Up to 2 months per cycle
Simplified Process View

strategy
process

customer oriented (sales, service, production) process

value

product creation
process

people, process and technology
management process

supplying business

customer
Simplified Process; Money and Feedback

- Strategy
- Process
- Supplying business
- Value
- Customer oriented
- Feedback
- Long term; know how (soft) assets
- Short term; cashflow!
- Mid term; cashflow next year!
- Long term; know how (soft) assets
- People, process and technology
- Product creation
Simplified process diagram for project business

- policy and planning
- tender
- project execution
- deployment

- systems architecting
- product creation
- people, process, and technology management

- systems
- products or components
Decomposition of the Product Creation Process

Operational Management:
- specification
- budget
- time

Design Control:
- technical
  - needs: what is needed
  - specification: what will be realized
  - design: how to realize
  - verification: meeting specs following design engineering:
    - how to produce and to maintain

Marketing:
- profitability
- saleability
  - customer input
  - customer expectations
  - commercial structure
  - product pricing
  - market introduction
  - introduction at customer
  - feedback

Project Systems Engineering Introduction; Phasing, Process, Organization
version: 0.2
April 6, 2019
PCPdecomposition
Operational Organization of the PCP

- **Operational**
  - Entire portfolio
    - Portfolio operational manager

- **Technical**
  - Product family
    - Family operational manager
  - Subsystem
    - Subsystem project leader
  - Module
    - Developers

- **Commercial**
  - Entire portfolio
    - Portfolio marketing manager
  - Product family
    - Family marketing manager
  - Single product
    - Product manager
  - Subsystem
    - Subsystem architect
  - Module
    - Developers
Prime Responsibilities of the Operational Leader

Specification

Quality

Resources  Time
The Rules of the Operational Game

- define project
- update project

- assess risks
- determine feasibility
- accept or reject

- specification, resources, time
- accept

- execute project within normal quality rules

business management

project leader
Operational Teams

- Operational Leader (project leader)
- Operational Support (project manager)
- Marketing or Product Manager
- Application Manager
- Requirements Analyst
- Test Engineer
- Architect
- Technology-Specific Architects
- Development support
- Subsystem Operational Leaders
- Subsystem Architects
- Manufacturing
- Logistics
- Sales Manager
- Service
- Quality Assurance
- Operational Teams

PCPconcentric Teams
What Service Level to Deliver?

- use results
  - functional capability
  - technical capability
  - expert support
  - initial production

- capability management
- facility management
- customer support
- factory

- performance-based or service-level agreement
- conventional maintenance contract
- product acceptance and warranty
How the project will perform the systems engineering process:

- main events and activities
- roles and responsibilities
- work products
- procedures and standards

Bridge between project management and engineering (NASA 2016)
Abstract

The core principle of systems integration is early validation; are the assumptions of the needs, specifications and design decisions valid? It is better to fail early, then to hit faulty assumptions, unknowns, or uncertainties late in development.
Most Problems are Found Late

failures found during integration and test can be traced back to unknowns, unforeseens, and wrong assumptions
Waterfall model

- identify needs
- specify
- design
- realize
- integrate
- verify & validate

works well:
- in mature product-market combinations
- with long development cycles

works poorly:
- in new product-market combinations
- short development cycles
Concurrent Engineering

- identify needs
- specify
- design
- realize
- integrate
- qualify

- total development time is shorter
- technology constraints & opportunities take time to get in the picture
- validation is still late (=feedback on uncertain requirements)
Iterative Approach

- identify needs
- specify
- design
- realize
- integrate
- qualify

learn fast by iterating over needs and technology
- more chaotic
- requires agile mindset
Continuous Integration

- continuous confrontation
- no build-up of issues
- ensures working system
- less complex diagnosis

However, ongoing changes may be perceived as disturbing
Development Processes From Waterfall to Agile

- waterfall
- triple-V
  - M1: functional model
  - M2: prototype
  - M3: product
- many Vs
  - spiral
  - agile/incremental/continuous
  - and all kinds of hybrids
Abstract

Products and enterprises evolve over time. This presentation explores the impact of these changes on the system and on the business by making (small and simple) models of life cycle aspects.
Product Related Life Cycles

- Individual systems
- Service
  - System production
  - System sales
  - System creation
  - Upgrades and options production
  - Upgrades and options sales
  - Upgrades and options creation
- Disposal
System Life Cycle

- System order
- Ordering components
- Manufacturing
- Shipping
- Installation
- Using
- Add option
- Maintenance
- Upgrade
- Using
- Sales
- Shipping
- Refurbishing
- Shipping
- Installation
- Secondary use
- Maintenance
- Dispose

Local changes, e.g., accounts procedures
## Approach to Life Cycle Modeling

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify potential life cycle changes and sources</td>
<td></td>
</tr>
<tr>
<td>Characterize time aspect of changes</td>
<td>how often how fast</td>
</tr>
<tr>
<td>Determine required effort</td>
<td>amount type</td>
</tr>
<tr>
<td>Determine impact of change on system and context</td>
<td>performance reliability</td>
</tr>
<tr>
<td>Analyse risks</td>
<td>business</td>
</tr>
</tbody>
</table>

**See reasoning**
Abstract

This presentation uses the TRIZ 9 Windows diagram as framework to think about needs elicitation.
Our Primary Interest

developing organization

architect

system of interest
Context, Zoom-out and Zoom-in

customer organization

developing organization

architect

supplier organization

super system

system of interest

subsystems
Adding the Time Dimension

past  current  future

customer organization  past super system  super system  future super system

developing organization  past system of interest  system of interest  future system of interest

architect  knowledge  innovation

supplier organization  past subsystems  subsystems  future subsystems

based on TRIZ
Sources of Needs and Constraints

past current future

customer organization

devolving organization

architect

supplier organization

past super system

super system

future super system

past system of interest

system of interest

future system of interest

past subsystems

subsystems

future subsystems

needs

constraints

based on TRIZ
Abstract

Requirements engineering is one of the systems engineering pillars. In this document we discuss the fundamentals of systems engineering, such as the transformation of needs into specification. Needs and requirements prescribe what rather than how.
Definition of “Requirement”

Requirements describing the needs of the customer:  
**Customer Needs**

Requirements describing the characteristics of the final resulting system (product):  
**System (Product) Specification**

The *requirements management process* recursively applies this definition for every level of decomposition.

Requirements describing the needs of the company itself over the life cycle:  
**Life Cycle Needs**
Flow of Requirements

**What**
- choices
- trade-offs
- negotiations

**What**
- customer needs:
  - What is needed by the customer?
- product specification:
  - What are we going to realize?
- system design:
  - How are we going to realize the product?

**What**
- What are the subsystems we will realize?

**How**
- How will the subsystems be realized?

up to "atomic" components
System as a Black Box

- **system seen as black box**
- **inputs**
- **functions**
- **quantified characteristics**
- **outputs**
- **interfaces**
- **restrictions, prerequisites**
- **boundaries, exceptions**
- **standards, regulations**
Good Requirements are “SMART”

- **S**pecific
- **M**easurable
- **A**chievable (Attainable, Action oriented, Acceptable, Agreed-upon, Accountable)
- **R**ealistic (Relevant, Result-Oriented)
- **T**ime-bounded (Timely, Tangible, Traceable)
### Typical Use Case
- What is the user typically doing with the system in the system context?
- Quantify the operation and context in this typical case.

### Other Use Cases
- Operational variants
- Boundary behavior
- Exceptional cases

> SysML use cases
Concept Selection, Set Based Design and Late Decision Making

by Gerrit Muller  University of South-Eastern Norway-NISE

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

We discuss a systems design approach where several design options are maintained concurrently. In LEAN Product Development this is called set-based design. Conventional systems engineering also promotes the concurrent evaluation of multiple concepts, the so-called concept selection. Finally, LEAN product development advocates to keep options open as long as feasible; the so-called late decision making.

Distribution

This article or presentation is written as part of the Gaudi project. The Gaudi project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

April 6, 2019
status: planned
version: 0
Problem Solving Approach

1. Problem understanding by
   exploration and simple models

2. Analysis by
   + exploring multiple propositions (specification + design proposals)
   + exploring decision criteria (by evaluation of proposition feedback)
   + assessment of propositions against criteria

3. Decision by
   + review and agree on analysis
   + communicate and document

4. Monitor, verify, validate by
   + measurements and testing
   + assessment of other decisions

vague problem statement
conflicting other decision
insufficient data
no satisfying solution
invalidated solution

Concept Selection, Set Based Design and Late Decision Making
Gerrit Muller
version: 0
April 6, 2019
TORdecisionFlow
Examples of Pugh Matrix Application

### Swivel concept selection

<table>
<thead>
<tr>
<th>Evaluation criteria</th>
<th>Weight</th>
<th>CBV</th>
<th>Clamp</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maturity</td>
<td>10</td>
<td>5</td>
<td>50</td>
<td>2</td>
</tr>
<tr>
<td>Development level</td>
<td></td>
<td>20</td>
<td>40</td>
<td>2</td>
</tr>
<tr>
<td>Cost</td>
<td>20</td>
<td>4</td>
<td>80</td>
<td>2</td>
</tr>
<tr>
<td>Hardware cost</td>
<td></td>
<td>5</td>
<td>100</td>
<td>2</td>
</tr>
<tr>
<td>Development cost</td>
<td></td>
<td>5</td>
<td>100</td>
<td>2</td>
</tr>
<tr>
<td>Design robustness</td>
<td>25</td>
<td>5</td>
<td>125</td>
<td>3</td>
</tr>
<tr>
<td>Design life</td>
<td></td>
<td>5</td>
<td>125</td>
<td>3</td>
</tr>
<tr>
<td>Swivel cycles</td>
<td></td>
<td>5</td>
<td>125</td>
<td>3</td>
</tr>
<tr>
<td>Pressure cycles</td>
<td></td>
<td>5</td>
<td>125</td>
<td>3</td>
</tr>
<tr>
<td>Pressure range, internal</td>
<td>4</td>
<td>100</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Pressure range, external</td>
<td>5</td>
<td>125</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Temperature range</td>
<td>4</td>
<td>100</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Installation</td>
<td>10</td>
<td>2</td>
<td>40</td>
<td>3</td>
</tr>
<tr>
<td>Initial install/retrieval</td>
<td></td>
<td>4</td>
<td>80</td>
<td>3</td>
</tr>
<tr>
<td>Connection/disconnection</td>
<td></td>
<td>2</td>
<td>40</td>
<td>4</td>
</tr>
<tr>
<td>Operation</td>
<td>25</td>
<td>1</td>
<td>25</td>
<td>4</td>
</tr>
<tr>
<td>Swivel resistance</td>
<td></td>
<td>1</td>
<td>25</td>
<td>4</td>
</tr>
<tr>
<td>Spool Length Short</td>
<td></td>
<td>3</td>
<td>75</td>
<td>5</td>
</tr>
<tr>
<td>Spool Length Long</td>
<td></td>
<td>3</td>
<td>75</td>
<td>5</td>
</tr>
<tr>
<td>Hub loads</td>
<td></td>
<td>2</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td>Hub loads</td>
<td></td>
<td>2</td>
<td>50</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td><strong>Sum points</strong></td>
<td>985</td>
<td>1165</td>
<td>1290</td>
</tr>
</tbody>
</table>

### EDP-LRP connection

<table>
<thead>
<tr>
<th>Evaluation Criteria</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to connect</td>
<td>-</td>
</tr>
<tr>
<td>Need for ROV</td>
<td>+</td>
</tr>
<tr>
<td>Design</td>
<td>+</td>
</tr>
<tr>
<td>Robustness</td>
<td>+</td>
</tr>
<tr>
<td>Connector design</td>
<td>-</td>
</tr>
<tr>
<td>Number of parts</td>
<td>-</td>
</tr>
<tr>
<td>Handle roll-off</td>
<td>-</td>
</tr>
<tr>
<td>Influence other</td>
<td>+</td>
</tr>
<tr>
<td>Redundancy</td>
<td>+</td>
</tr>
<tr>
<td>Design</td>
<td>+</td>
</tr>
<tr>
<td>Interchangeability</td>
<td>+</td>
</tr>
<tr>
<td>Cost</td>
<td>-</td>
</tr>
<tr>
<td>HW cost</td>
<td>-</td>
</tr>
<tr>
<td>Manufacturing cost</td>
<td>S</td>
</tr>
<tr>
<td>Engineering cost</td>
<td>+</td>
</tr>
<tr>
<td>Service cost</td>
<td>+</td>
</tr>
<tr>
<td>Maturity</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDP-LRP connection</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDP-LRP connection</td>
</tr>
</tbody>
</table>

from master paper Halvard Bjørnsen, 2009

from master paper Dag Jostein Klever, 2009
Evolution of Design Options

Concept Selection, Set Based Design and Late Decision Making

version: 0
April 6, 2019
CSSBsetEvolution
Conclusions

Evolving multiple concepts increases insight and understanding (LEAN product development: set-based design, SE: Pugh matrix)

Articulation of criteria sharpens evaluation

The discussion about the Pugh matrix is more valuable than final bottomline summation

Delaying decisions may help to keep options (Lean Product Development: late decision making, finance: real options)
Abstract

This presentation explains the fundamentals behind Architecture and Design, such as conceptual and functional design, partitioning, interfaces, and allocation.
Visualizing Dynamic Behavior

by Gerrit Muller        TNO-ESI, University College of South East Norway

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

Dynamic behavior manifests itself in many ways. Architects need multiple complementary visualizations to capture dynamic behavior effectively. Examples are capturing information, material, or energy flow, state, time, interaction, or communication.
Overview of Visualizations of Dynamic Behavior

Information Transformation Flow

Concrete “Cartoon” Workflow

Timeline and Functional Flow

Signal Waveforms

Timeline of Workflow

Abstract Workflow

Flow of Light

Swimming Lanes

Information Centric Processing Diagram

Concurrency and Interaction

State Diagram
Example Functional Model of Information Flow

1. **Get Sensor Data**
   - Transform data into image

2. **Fuse Sensor Images**
   - Detect objects

3. **Classify Objects**
   - Update world model

4. **Get External Data**
   - Analyze situation

5. **Get Goal Trajectory**
   - Determine next step

6. **GPS Data**
   - Calculate GPS location

7. **Get V, a**
   - Estimate location

- **Location**
  - Objects

- **World Model**
Visualizing Dynamic Behavior

Gerrit Muller

version: 0
April 6, 2019
SSMтипicalWorkoverOperationCartoon

vessel or platform
rig
EDP
LRP
riser
XT
well
head
WOCS
TF
SFT

"Cartoon" Workflow
Workflow as Functional Model

1. assembly, functional test
2. run EDP/LRP
3. run risers
4. hook up SFT and TF
5. move above well
6. ROV assisted connect
7. hook up coil tubing and wireline BOP
8. system function and connection seal test
9. run coil tubing and wireline
10. perform workover operations
11. retrieve coil tubing and wireline BOP
12. unhook coil tubing and wireline BOP
13. ROV assisted disconnect
14. move away from well
15. retrieve SFT and TF
16. retrieve risers
17. retrieve EDP/LRP
18. disassembly
19. move above well
20. move away from well

Visualizing Dynamic Behavior
97 Gerrit Muller

version: 0
April 6, 2019
SSMTypicalWorkoverOperation
Workflow as Timeline

assumptions:
running and retrieving risers: 50m/hr
running and retrieving coiled tubing/wireline: 100m/hr
depth: 300m
Swimming Lane Example

- clean wafer
- robot
- prealign
- clean master
- prefill
- print

0 100b 200b
Example Signal Waveforms

imaging = repeating similar pattern many times

g_y=0

Gy=127

typical TE: 5..50ms
Example Time Line with Functional Model

- Call family doctor
- Visit family doctor
- Call neurology department
- Visit neurologist
- Call radiology department
- Examination itself
- Diagnosis by radiologist
- Report from radiologist to neurologist
- Visit neurologist

Days 1-25
Flow of Light (Physics)

- Laser
- Pulse-freq, bw, wavelength, ..
- Uniformity
- Illuminator
- Sensor
- Reticle
- Lens
- Aerial Image
- Wafer
- NA
- Abberations
- Transmission
Dynamic Behavior is Multi-Dimensional

How does the system work and operate?
Functions describe what rather than how.
Functions are verbs.
Input-Process-Output paradigm.
Multiple kinds of flows:
  physical (e.g. hydrocarbons, goods, energy)
  information (e.g. measurements, signals)
  control
Time, events, cause and effect
Concurrency, synchronization, communication

multi-dimensional information and dynamic behavior
Abstract

The presentation explains fundamental concepts of and approach to system partitioning.
Partitioning is Applied Recursively

```
system
  subsystem 1
    subsub system A
    atomic part
    subsub system B
    atomic part
  subsub system N
    atomic part

subsystem 2
  subsub system A
  subsub system B
  subsub system N

... ...

subsystem n
  subsub system A
  subsub system B
  subsub system N
```
Software plus Hardware Decomposition

Applications
- view
- PIP
- adjust
- view
- TXT

Services
- viewport
- menu

Toolboxes
- audio
- video
- TXT
- etc.

Drivers
- scheduler

Signal Processing Subsystem
- drivers
- frame-buffer
- MPEG
- DSP

Control Subsystem
- signal processing subsystem
- domain specific
- generic

OS
- networking
- file-system

CPU
- RAM
- etc

Control Subsystem
- etc
the part is cohesive

functionality and technology belongs together

the coupling with other parts is minimal

minimize interfaces

the part is self-sustained for production and qualification

can be in conflict with cost or space requirements

clear ownership of part

e.g. one department or supplier
How much self-sustained?

How self sustained should a part be?

trade-off:

cost/speed/space optimization

logistics/lifecycle/production flexibility clarity
Decoupling via Interfaces

- part e.g. pipe
- control interface e.g. CAN
- power interface
- part e.g. pressure and flow regulator
- hydrocarbon interface
- mechanical mounting interface
- other part with same interfaces can replace original
- part e.g. pipe
The Ideal Modularity

System is composed

by using standard interfaces

limited catalogue of variants (e.g. cost performance points)
Example Physical Decomposition and Visualization

Fluidic subsystem
- chamber
- bottom chuck
- process power supply

electronics infrastructure

base frame + x, y, z stage
- optics stage control
- vision
- optics stage
- scoop
- camera
- stage
- frame
- granite

ZUBA
- control
- stage

covers and hatches
- controls
- cabling
- ventilation air flow
- contamination evacuation
- sensors measurement frame
- machine control
- "remote" electronics rack

back side view
front side view
integrating
Example Work Breakdown Structure

**project organization**

- TIP:NBE R1
  - scanning
  - viewing
  - database
  - computing
  - system

**work packages**

- preparation
- conversion
- run time acq
- algorithms

- xDAS
- xFEC
- reconstruction hardware

- VDU
- console
- algorithms
- gfx
- UI

- database engine
- clinical
- bulk data
- import export
- archive

- host
- OS
- foundation classes
- start up shutdown
- exception handling

- SPS
- SDS
- TPS
- integration
- alfa test
- beta test
- conf man

**legend**

- project
- segment
- make SW
- make HW
- buy SW
- buy HW
- system
Example SW plus HW Decomposition

- Applications:
  - view
  - PIP
  - adjust
  - TXT

- Services:
  - viewport
  - menu

- Toolboxes:
  - audio
  - video
  - TXT
  - etc.

- Driver:
  - drivers
  - scheduler

- Hardware:
  - tuner
  - frame-buffer
  - MPEG
  - DSP

- Domain Specific Subsystem:
  - signal processing subsystem

- Generic Subsystem:
  - control subsystem

- Drivers:
  - scheduler

- Services:
  - browse
    - networking
    - file-system

- Hardware Components:
  - CPU
  - RAM
  - etc.
Abstract

The supply chain dominates the economic viability of systems. Developing a system and its business requires the design of the supply chain.
System Order Lead Time

for all systems, subsystems, modules, components, ...

Order

Build

Ship

Store
Considerations for Designing the Supply Chain

- **Flow** of goods; stock = cost
- Produce Delivery Ratio < 1
  - Facilitate *demand-driven* goods flow.
  - Forecasting causes stocks and risks of obsolescence or underrun)
- **Risk** management
  - supplier dependency (2\textsuperscript{nd} supplier policy)
- Production and service **life time**
- **Traceability** of configurations and versions
Abstract

Systems Engineering offers many methods and techniques to detect and mitigate risks. This presentation touches upon methods like Failure Mode Effect Analysis, Hazard Analysis, etc. addressing related concerns such as reliability, safety, and security.
Life-cycle Commitment, Knowledge, and Incurred Cost

Commitment to Technology, Configuration, Performance, Cost, etc.

Cost Incurred

System-Specific Knowledge

Ease of Change

Conceptual/ Preliminary Design
Detail Design and Development
Construction and/or Production
System Use, Support, Phase-out, and Disposal

Copyright © 2011 Pearson Education, Inc. publishing as Prentice Hall
# FMEA-like Analysis Techniques

<table>
<thead>
<tr>
<th>Safety</th>
<th>Potential Hazards</th>
<th>Damage</th>
<th>Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hazard Analysis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td>Failure Modes</td>
<td>Effects</td>
<td>Measures</td>
</tr>
<tr>
<td>FMEA</td>
<td>Exceptional Cases</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td>Vulnerability Risks</td>
<td></td>
<td>Measures</td>
</tr>
<tr>
<td>Maintainability</td>
<td>Change Cases</td>
<td>Impact, Effort, Time</td>
<td>Decisions</td>
</tr>
<tr>
<td>Performance</td>
<td>Worst Cases</td>
<td>System Behavior</td>
<td>Decisions</td>
</tr>
</tbody>
</table>

- (Systematic) brainstorm
- Analysis and assessment
  - Probability
  - Severity
  - Propagation

- Improve
  - Spec, design, process, procedure, ...

<table>
<thead>
<tr>
<th>Systems Engineering Fundamentals Risk Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>version: 0</td>
</tr>
<tr>
<td>April 6, 2019</td>
</tr>
<tr>
<td>MAANfmeaLikeAnalysis</td>
</tr>
</tbody>
</table>
Severity-Probability Classification Matrix

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>V</th>
<th>VI</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>moderate</td>
<td>high</td>
</tr>
<tr>
<td>B</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>low</td>
<td>moderate</td>
<td>high</td>
</tr>
<tr>
<td>C</td>
<td>low</td>
<td>low</td>
<td>moderate</td>
<td>moderate</td>
<td>high</td>
<td>unacceptable</td>
</tr>
<tr>
<td>D</td>
<td>low</td>
<td>moderate</td>
<td>moderate</td>
<td>high</td>
<td>unacceptable</td>
<td>unacceptable</td>
</tr>
<tr>
<td>E</td>
<td>moderate</td>
<td>moderate</td>
<td>high</td>
<td>unacceptable</td>
<td>unacceptable</td>
<td>unacceptable</td>
</tr>
</tbody>
</table>

Based on [https://en.wikipedia.org/wiki/Failure_mode_and_effects_analysis](https://en.wikipedia.org/wiki/Failure_mode_and_effects_analysis)
Abstract

Readiness level models offer a yardstick to assess the status of specific project aspects. Examples are technology readiness and integration readiness.
<table>
<thead>
<tr>
<th>TRL 9</th>
<th>actual system proven in operational environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRL 8</td>
<td>system complete and qualified</td>
</tr>
<tr>
<td>TRL 7</td>
<td>system prototype demonstration in operational environment</td>
</tr>
<tr>
<td>TRL 6</td>
<td>technology demonstrated in relevant environment</td>
</tr>
<tr>
<td>TRL 5</td>
<td>technology validated in relevant environment</td>
</tr>
<tr>
<td>TRL 4</td>
<td>technology validated in lab</td>
</tr>
<tr>
<td>TRL 3</td>
<td>experimental proof of concept</td>
</tr>
<tr>
<td>TRL 2</td>
<td>technology concept formulated</td>
</tr>
<tr>
<td>TRL 1</td>
<td>basic principles observed</td>
</tr>
</tbody>
</table>

after: https://serkanbolat.com/2014/11/03/technology-readiness-level-trl-math-for-innovative-smes/
### Integration Readiness Levels

<table>
<thead>
<tr>
<th>TRL 7</th>
<th>The integration of technologies has been <strong>verified and validated</strong> with sufficient detail to be actionable.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRL 6</td>
<td>The integrating technologies can <strong>accept, translate, and structure information</strong> for its intended application.</td>
</tr>
<tr>
<td>TRL 5</td>
<td>There is sufficient <strong>control</strong> between technologies necessary to establish, manage, and terminate the integration.</td>
</tr>
<tr>
<td>TRL 4</td>
<td>There is sufficient detail in the <strong>quality and assurance</strong> of the integration between technologies.</td>
</tr>
<tr>
<td>TRL 3</td>
<td>There is <strong>compatibility</strong> (i.e. common language) between technologies to orderly and efficiently integrate and interact.</td>
</tr>
<tr>
<td>TRL 2</td>
<td>There is some level of specificity to characterize the <strong>interaction</strong> (i.e. ability to influence) between technologies through their interface.</td>
</tr>
<tr>
<td>TRL 1</td>
<td>An <strong>interface</strong> (i.e. physical connection) between technologies has been identified with sufficient detail to allow characterization of the relationship.</td>
</tr>
</tbody>
</table>

(from: From TRL to SRL: The Concept of Systems Readiness Levels, CSER2006, by Sauser et al.)
Abstract

This lesson positions systems integration as process in the development processes.
Typical Concurrent Product Creation Process

-1 strategy
0. feasibility
1. definition
2. system design
3. engineering
4. integration & test
5. field monitoring
6. product operational life cycle
Zooming in on Integration and Tests

- **0. feasibility**
- **1. definition**
- **2. system design**
- **3. engineering**
- **4. integration & test**
- **5. field monitoring**
- **6. product operational life cycle**

**Integrate**

- **system test**
- **alpha test**
- **beta test**
- **gamma test**

Focus on reducing uncertainties and hitting unknowns

“Fail Early”

Focus on qualification:
Documenting objective evidence

Mastering Systems Integration; Process and Positioning

128  Gerrit Muller

version: 0.3
April 6, 2019
MSIPPlocus
Integration Takes Place in a Bottom-up Fashion

- Component
- Subsystem
- System function
- Product
- Context
- Integrate
- Alpha test
Abstract

This presentations discusses the strategy for integration. The strategy is transformed into an approach to determine an integration sequence based on Key Performance Parameters and potential risks to achieve them.
Integration Strategy

- Get Key Performance Parameters functioning ASAP
- Work on highest risks ASAP
- Use a pacing process (regular visible results)
  - with regular milestones
  - and increments in functionality and performance
- Merge constraints from test configurations, suppliers, resources, etc.
Pacing Milestones

functioning exposure and acquisition

First image manual preparation

10% IQ manual preparation

20% IQ automated preparation 10% speed

50% IQ automated preparation 100% speed

Full IQ Full speed

pacing:
maximum 6 month between milestones depending on technology and domain
Defining an Integration Sequence in Increments

<table>
<thead>
<tr>
<th>Key Performance Parameters (KPPs)</th>
<th>Key Functionality milestones</th>
</tr>
</thead>
<tbody>
<tr>
<td>functioning exposure and acquisition</td>
<td>Full IQ Full speed</td>
</tr>
<tr>
<td>First image manual preparation</td>
<td>50% IQ automated preparation 100% speed</td>
</tr>
<tr>
<td>10% IQ manual preparation</td>
<td>20% IQ automated preparation 10% speed</td>
</tr>
</tbody>
</table>

- **definition of integration increments working backwards in time (demand driven)**

**imaging stream**
- acquisition increment
- imaging increment
- IQ 1 increment
- IQ 2 increment
- IQ 3 increment
- IQ 4 increment

**control and performance stream**
- exposure increment
- manual prep increment
- automation increment
- speed increment
Stepwise Integration Approach

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Determine most critical system performance parameters.</td>
</tr>
<tr>
<td>2</td>
<td>Identify subsystems and functions involved in these parameters.</td>
</tr>
<tr>
<td>3</td>
<td>Work towards integration configurations along these chains of subsystems and functions.</td>
</tr>
<tr>
<td>4</td>
<td>Show system performance parameter as early as possible; start with showing &quot;typical&quot; system performance.</td>
</tr>
<tr>
<td>5</td>
<td>Show &quot;worst-case&quot; and &quot;boundary&quot; system performance.</td>
</tr>
<tr>
<td>6</td>
<td>Rework manual integration tests in steps into automated regression tests.</td>
</tr>
<tr>
<td>7</td>
<td>Monitor regression results with human-driven analysis.</td>
</tr>
<tr>
<td>8</td>
<td>Integrate the chains: show system performance of different parameters simultaneously on the same system.</td>
</tr>
</tbody>
</table>
Abstract

Systems Integration requires specific project management. The challenge for project managers is to plan ahead, knowing that the integration plan will need continuous adaptations.
Integration Planning

defining integration sequence

project planning

needs

options constraints

master plan

now

actual situation

options constraints

needs

lead customers

(test)facility management

suppliers

resource management
Demand Driven, Fitting Constraints

- **defining integration sequence** defined by output demand
- lead customers limited validation options
- (test)facility management mapped on limited set of test systems
- suppliers trade-offs with constraints from suppliers
- resource management trade-offs with internal resource constraints

use of various environments (from virtual to final) reordering of integration sequence
Combating Uncertainty in the Workflow of Systems Engineering Projects
by Barry Papke and Rick Dove, INCOSE 2013

~6 weeks
~2 weeks
now
Abstract

This presentation defines terms, which are used in relation to systems integration, such as validation, verification, qualification, evidence, approval process, certification, and acceptance.
Validation and Verification in the V-model

**validation** does the system fit the needs and match expectations?

**verification** did we build what we specified and according the design?

- needs
- specification
- system design
- subsystem design
- component design
- component realization
- component test
- subsystem test
- system test
- validation
Functional Model of Verification

system-of-interest → test specification → test environment

→ test

→ report

→ review

→ assess

→ conclusion

→ references

→ report

→ problem reports deviations

→ qualification documentation with objective evidence

→ test inputs

→ requirements
Certification: an independent agency (e.g. DNV-GL) certifies the quality of the system-of-interest, technology, or process.

Self-certification: the company has been accredited by the agency to do the certification themselves.

- check qualification data
- check process and organization
- optional audit
- certify
Development and (repeated) Production

- System specification
- Test specification
- Verification
- Qualification documentation
- Development; only after design changes

- Acceptance test procedure
- FAT
- FAT report
- Factory

- SAT
- SAT report
- Site

Production and delivery once per system
 Objective Evidence

From a business perspective: Objective evidence is “information based on facts that can be proved through analysis, measurement, observation, and other such means of research.”

From a legal perspective: Objective evidence is “real evidence, also known as demonstrative or objective evidence; this is naturally the most direct evidence.”

From a scientific perspective: “To be termed scientific, a method of inquiry must be based on gathering observable, empirical, and measurable evidence subject to specific principles of reasoning. A scientific method consists of the collection of data through observation and experimentation, and the formulation and testing of hypotheses.”

From a list of Plain English definitions related to the ISO 9000, 9001 and 9004: Objective evidence is “data that show or prove that something exists or is true. Objective evidence can be collected by performing observations, measurements, tests, or by using any other suitable method.”

from: Understanding Objective Evidence: (What It Is and What It Definitely Is Not), by Denise Dion http://www.eduquest.net/Advisories/EduQuest%20Advisory_ObjectiveEvidence.pdf
FDA is a science-based law enforcement agency and, therefore, requires answers that are scientifically and legally supported. FDA expects your objective data to answer the following questions:

- **Scientific** – Can the data be *evaluated by independent observers* to reach the same conclusions?

- **Scientific** – Are the data documented in a manner that *allows recreation of the data* or the events described?

- **Scientific** – Does the documented evidence provide *sufficient data* to prove what happened, when, by whom, how, and why?

- **Legal** – Was the documentation *completed concurrently* with the tasks?

- **Legal** – Is the documentation *attributable* (directly traceable to a person)?

- **Legal** – Have the data and associated documentation been maintained in a manner that *provides traceable evidence* of changes, deletions, additions, substitutions, or alterations?

- **Legal** – Are the data and associated documentation maintained in a manner that *protects and secures* them from changes, deletions, additions, substitutions, or alterations?

from: Understanding Objective Evidence: (What It Is and What It Definitely Is Not), by Denise Dion http://www.eduquest.net/Advisories/EduQuest%20Advisory_ObjectiveEvidence.pdf
Regulatory Approval Process

- **manufacturer**
  - create system
  - perform tests and trials
  - manufacture and deliver

- **regulatory agency** (e.g. FDA, DNV-GL)
  - provide regulations
  - review and approve
  - audit and inspect
Abstract

Deployment covers all steps from receiving shipped components to getting the system live and operating the system in its operational context.
Deployment Workflow

- **Installing**
  - Physical
  - Configuration data
e.g. I/O lists
  - Configuration doc.
- **Configuring**
  - Logical
  - Configured system
  - Customization data
e.g. doctrines, procedures
  - Configuration doc.
- **Commissioning**
  - Application
  - Operational system
  - Operational data, e.g. mission, maps, weather
  - Consumables, spare parts
  - Operations and maintenance doc.
- **Operating & Maintaining**
  - Pollution
  - Noise
  - Performance data
  - Recycling material

- Shipped parts
- Installation doc.