Mastering Systems Integration all slides

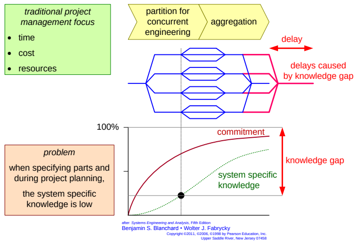
by *Gerrit Muller*

TNO-ESI, USN-SE

Abstract

The course Mastering Systems Integration discusses multiple perspectives on systems integration. The dominating principle underlying the course is that unknowns and major uncertainties need to be found as early as possible to mitigate system and project consequences. A good subtitle of the course is ”Architects meet Project Leaders meet Reality.”

March 3, 2024
status: concept
version: 1.0



after: *Systems Engineering and Analysis*, Fifth Edition
Benjamin S. Blanchard • Wolter J. Fabrycky
Copyright ©2011, ©2006, ©1998 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458

Mastering Systems Integration; Introduction

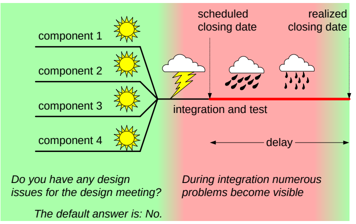
by *Gerrit Muller*   TNO-ESI, University College of South-Eastern Norway
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

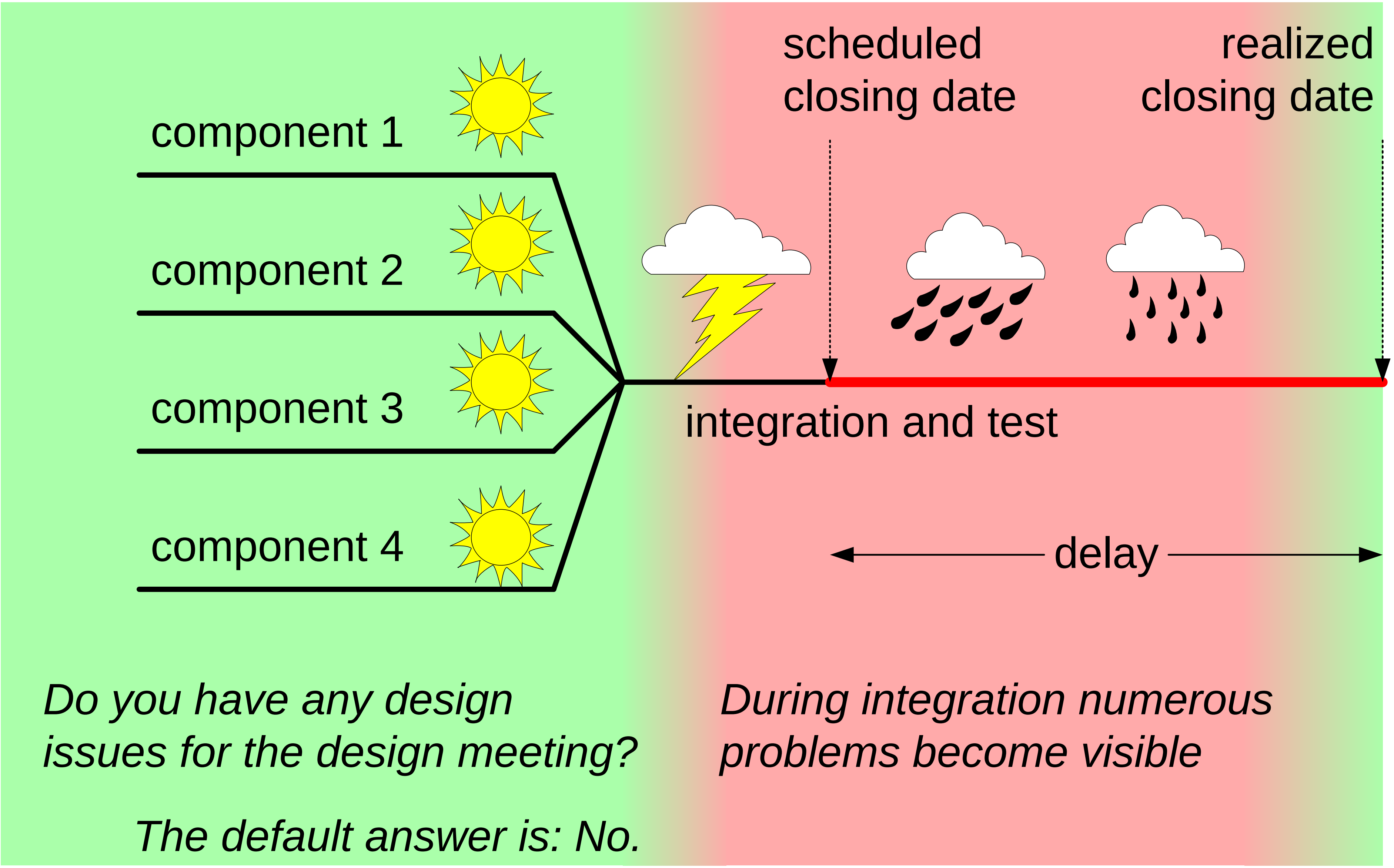
This presentation introduces the ideas behind the course Mastering Systems Integration. Systems integration requires cooperation from many project members, such as project leader, product manager, architect, lead designer, integrator, and tester. Integration is more than a simple aggregation as the reverse of the decomposition. The purpose of systems integration is to detect anything nasty that has not been foreseen as early as possible.

Distribution

March 3, 2024
status: preliminary draft
version: 0.5



Integration uncovers hidden problems



Do you have any design issues for the design meeting?

The default answer is: No.

During integration numerous problems become visible

Project Team; Contributions to Integration

Operational

Project Leader

- *planning*
- *organizing*
- *resources*
- *progress*

Technical

Architect
Lead Designer
Integrator

- *key functionality*
- *key performance parameters*
- *concept selection*
- *system design*
- *integration sequence*

Tester

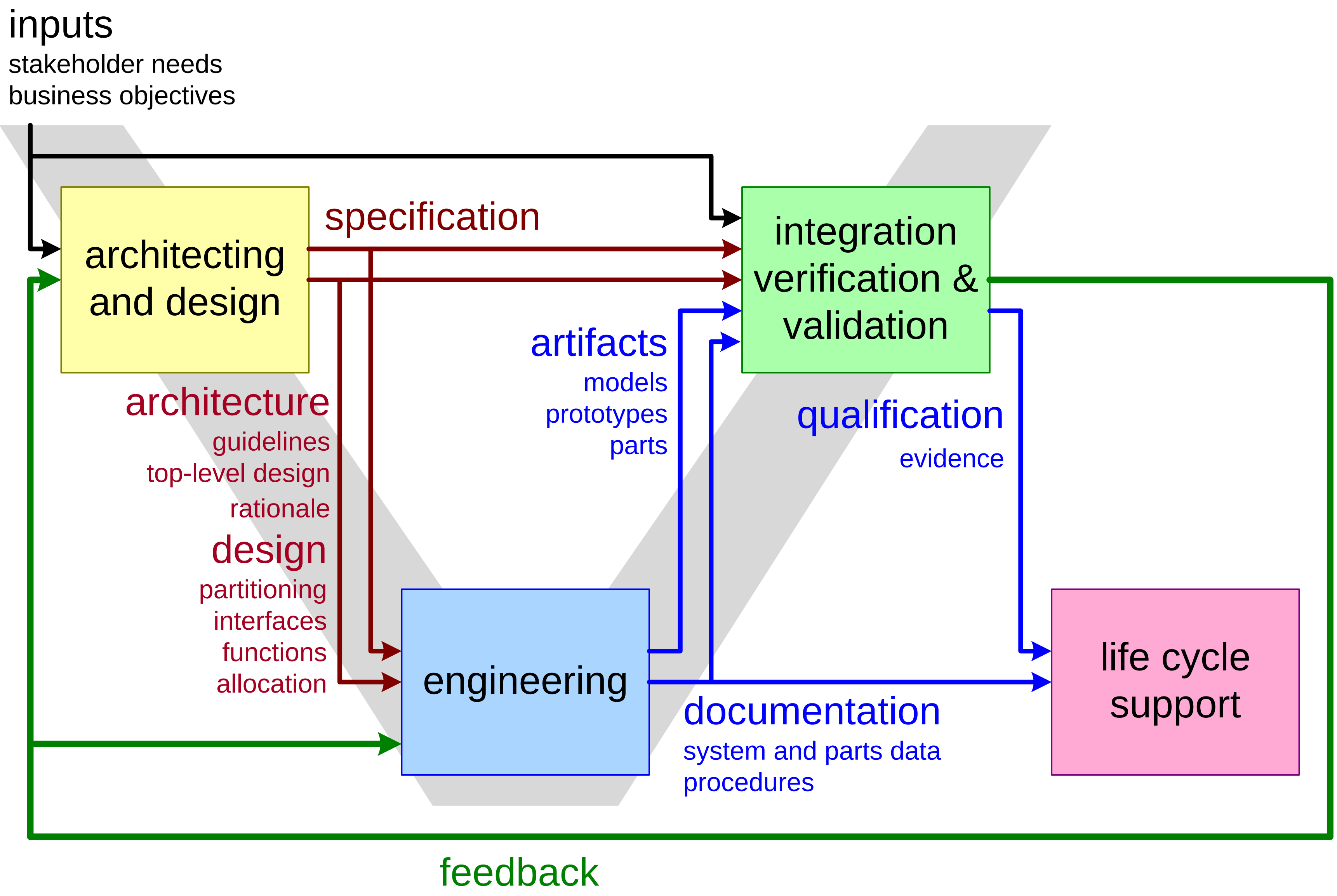
- *testing*
- *test configuration*
- *testware*
- *test specifications*
- *test reports*

Commercial

Product Manager

- *customer needs*
- *customer value*
- *system specification*

The Role of Integration in Development

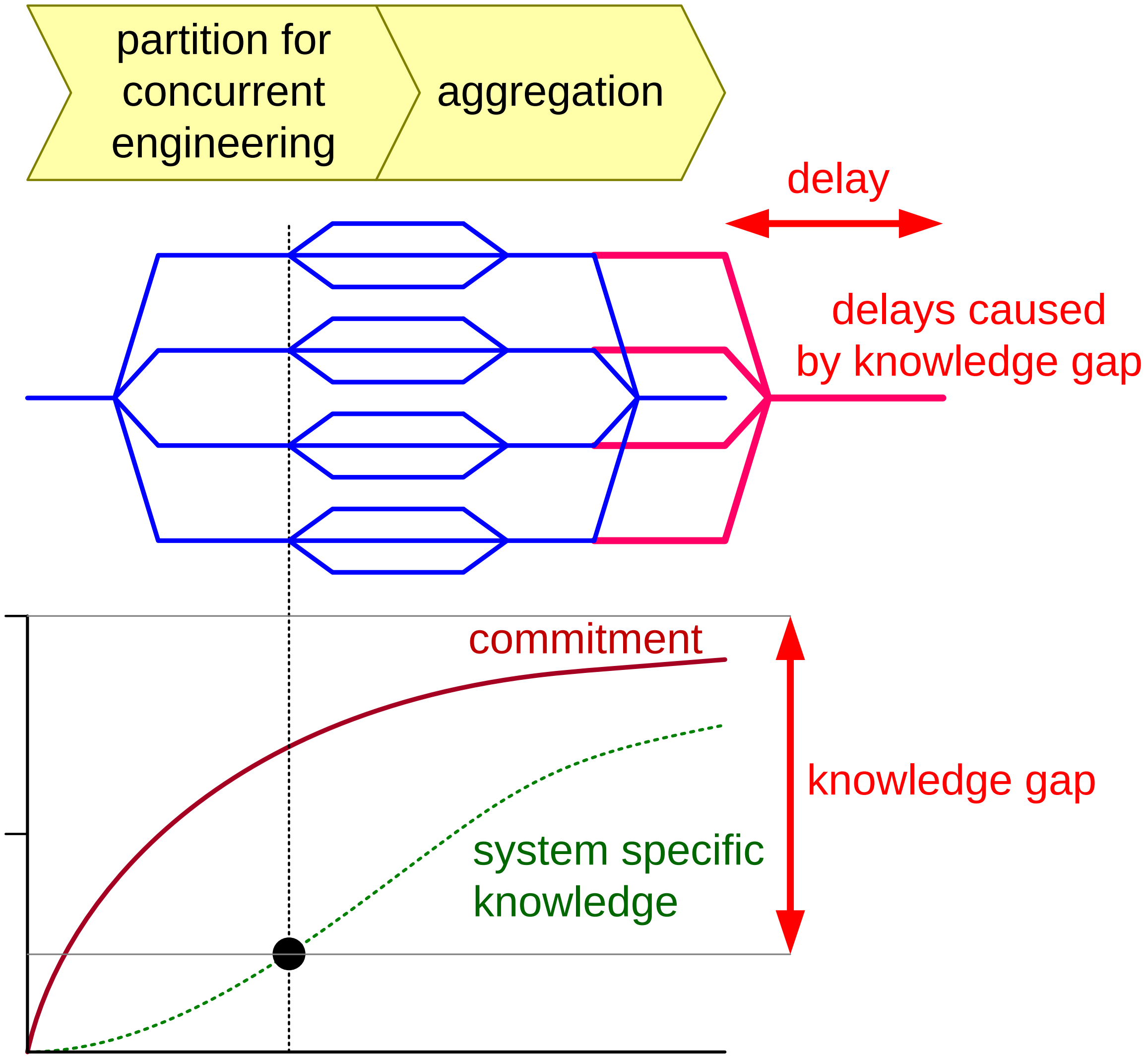


version: 0.5
March 3, 2024
MSIINdefinition

The Pain of Systems Integration

traditional project management focus

- time
- cost
- resources



problem

when specifying parts and during project planning,

the system specific knowledge is low

after: *Systems Engineering and Analysis*, Fifth Edition
Benjamin S. Blanchard • Wolter J. Fabrycky
Copyright ©2011, ©2006, ©1998 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458

Systems Integration Approach

Systems Integration starts when the project starts

The Integration perspective **drives** the project schedule by addressing the **major risks** from

volatility, **uncertainty**, **complexity** and **ambiguity**

Systems Integration strives to **Fail Early**; it is an early verification and validation

Systems Integration requires **multidisciplinary teamwork**, e.g.

Integrators, **Testers**, **Architects**, **Designers**, **Engineers**, **Project Leaders**, **Product Managers**, and others

Systems Integration complements Systems Architecting

Mastering Systems Integration; Course Overview

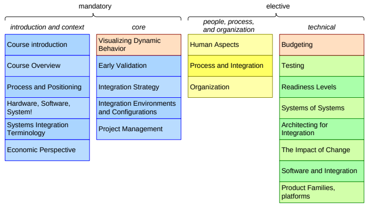
by *Gerrit Muller* TNO-ESI, University College of South-Eastern Norway
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

Course overview of the course Systems Integration.

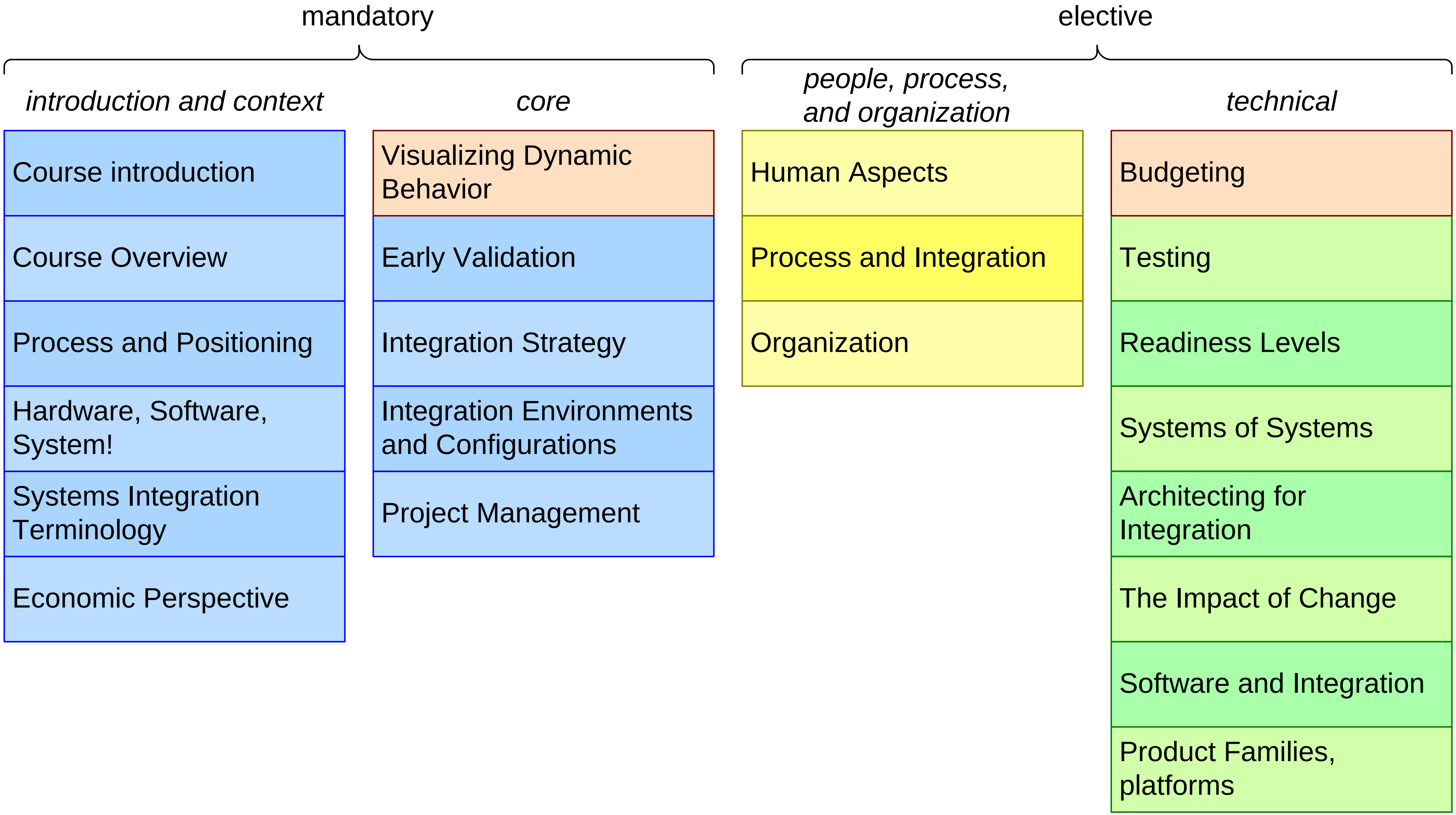
Distribution

March 3, 2024
status: draft
version: 0.5



mandatory		elective	
introduction and context	*core*	*people, process, and organization*	*technical*
Course introduction	Visualizing Dynamic Behavior	Human Aspects	Budgeting
Course Overview	Early Validation	Process and Integration	Testing
Process and Positioning	Integration Strategy	Organization	Readiness Levels
Hardware, Software, System!	Integration Environments and Configurations		Systems of Systems
Systems Integration Terminology	Project Management		Architecting for Integration
Economic Perspective			The Impact of Change
			Software and Integration
			Product Families, platforms

Nuggets Course Mastering Systems Integration



mandatory

elective

introduction and context

- Course introduction
- Course Overview
- Process and Positioning
- Hardware, Software, System!
- Systems Integration Terminology
- Economic Perspective

core

- Visualizing Dynamic Behavior
- Early Validation
- Integration Strategy
- Integration Environments and Configurations
- Project Management

people, process, and organization

- Human Aspects
- Process and Integration
- Organization

technical

- Budgeting
- Testing
- Readiness Levels
- Systems of Systems
- Architecting for Integration
- The Impact of Change
- Software and Integration
- Product Families, platforms

Content per Nugget

mandatory

introduction and context

Course introduction
Why is integration difficult and poorly understood

Course Overview
nuggets, face-to-face program

Process and Positioning
integration in phase gate process
fail early, qualification
top-down and bottom-up

Hardware, Software, System!
from components to system qualities
different failure modes and patterns
early HW SW integration

Systems Integration Terminology
validation, verification
qualification, certification objective evidence, regulatory agencies
site and factory acceptance

Economic Perspective
knowledge and life-cycle commitment
time-to-market
cost milestones

core

Visualizing Dynamic Behavior

Early Validation
V-model and late validation
from waterfall to iterative
continuous integration
from waterfall to agile processes

Integration Strategy
approach: fail early, reduce risk
perspectives
dynamic behavior
qualities
project, product, component
robustness of integration
cookbook

Integration Environments and Configurations
test configurations
modeling
configuration management
testware

Project Management
integration planning
test facilities, resource and supplier planning
Last Planner

elective

people, process, and organization

Human Aspects
bias <https://en.wikipedia.org/wiki/List_of_cognitive_biases>
team work
motivation
approbation mindset versus internal

Process and Integration
process capability (of organization)
support with procedures, templates, tools
Governance

Organization
roles

technical

Budgeting

Testing
modeling, simulation prototyping, testing, monitoring (development & field), measuring, reviewing, inspecting
ALT, HALT, HASS
automation, regression
trouble shooting
diagnosis
design for experiment
testware

Readiness Levels
technology readiness
integration readiness

Systems of Systems integration
interoperability
COTS & characterization
Outsourcing (component, system, service)
data
cloud
time dimension
compatibility

Architecting for Integration
integration technologies
integration patterns
integration pitfalls

The Impact of Change

Software and Integration

Product Families, platforms
projects, products

Assignments in Face-to-Face Module

System Specification

- determine ***KPPs*** and their quantified specification
- assess ***risk*** of KPPs caused by volatility, uncertainty, complexity and ambiguity

  pick one ***high-risk*** KPP to elaborate
- describe ***typical use*** (including circumstances in the ***context***) related to KPP

System design

- make system, SW, and HW ***block diagrams*** (parts, interfaces, connections)
- model ***dynamic behavior*** resulting in the KPP
- map ***dynamic behavior*** on ***block diagrams***

  and ***budget***: quantify contributions to KPP
- re-assess ***risks*** of KPP

Systems Integration Plan

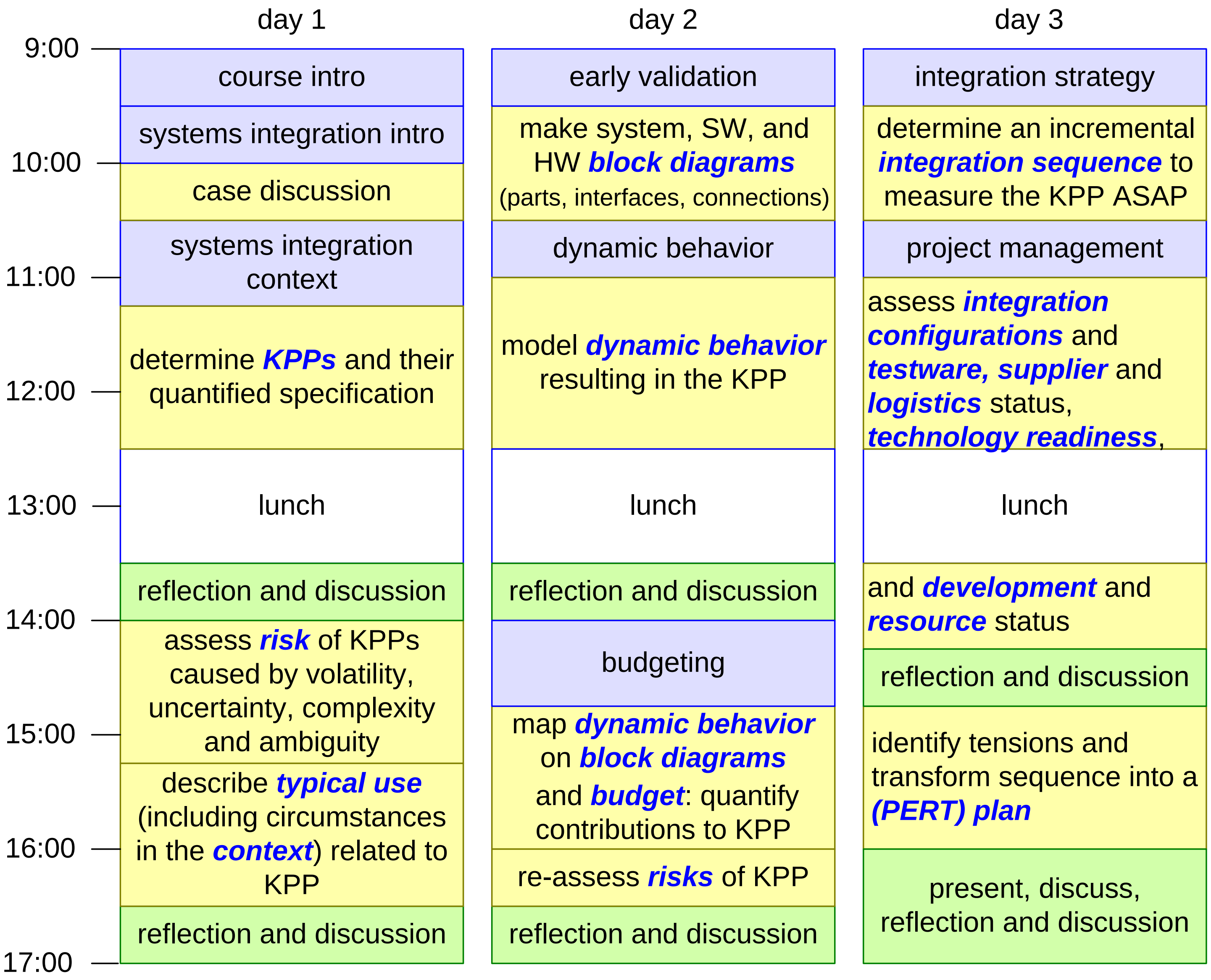
- determine an incremental ***integration sequence*** to measure the KPP as early as possible
- assess for the parts contributing to the KPP
  - ***fitness for purpose*** in customer context
  - ***integration configurations*** and ***testware***
  - ***supplier*** and ***logistics*** status
  - ***technology readiness***
  - ***development*** and ***resource*** status
- Identify ***tensions*** with development, logistics status, and availability of testware

  and transform the sequence in a ***(PERT) plan*** with required resources and ***integration configurations***
- assess ***robustness*** of the plan
- capture results in presentation

Reflection and Evaluation

- identify ***tensions*** or ***gaps*** in processes, organization, people, tools, instrumentation, context knowledge, etc. for executing the integration.

3-Day Face-to-Face Schedule



day 1

- course intro
- systems integration intro
- case discussion
- systems integration context
- determine ***KPPs*** and their quantified specification
- lunch
- reflection and discussion
- assess ***risk*** of KPPs caused by volatility, uncertainty, complexity and ambiguity
- describe ***typical use*** (including circumstances in the ***context***) related to KPP
- reflection and discussion

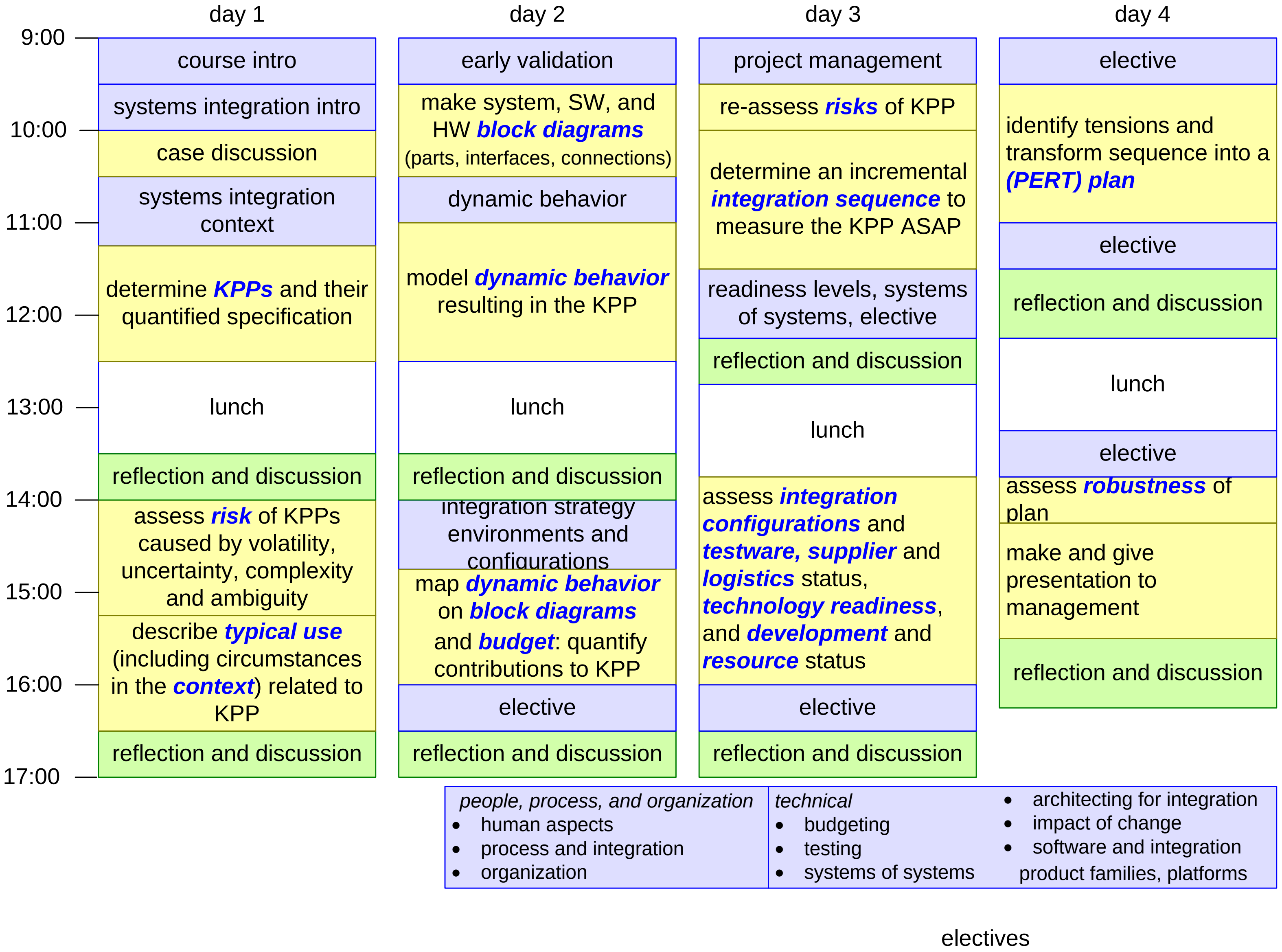
day 2

- early validation
- make system, SW, and HW ***block diagrams*** (parts, interfaces, connections)
- dynamic behavior
- model ***dynamic behavior*** resulting in the KPP
- lunch
- reflection and discussion
- budgeting
- map ***dynamic behavior*** on ***block diagrams*** and ***budget***: quantify contributions to KPP
- re-assess ***risks*** of KPP
- reflection and discussion

day 3

- integration strategy
- determine an incremental ***integration sequence*** to measure the KPP ASAP
- project management
- assess ***integration configurations*** and ***testware, supplier*** and ***logistics*** status, ***technology readiness***,
- lunch
- and ***development*** and ***resource*** status
- reflection and discussion
- identify tensions and transform sequence into a ***(PERT) plan***
- present, discuss, reflection and discussion

4-Day Face-to-Face Schedule



day 1

- course intro
- systems integration intro
- case discussion
- systems integration context
- determine ***KPPs*** and their quantified specification
- lunch
- reflection and discussion
- assess ***risk*** of KPPs caused by volatility, uncertainty, complexity and ambiguity
- describe ***typical use*** (including circumstances in the ***context***) related to KPP
- reflection and discussion

day 2

- early validation
- make system, SW, and HW ***block diagrams*** (parts, interfaces, connections)
- dynamic behavior
- model ***dynamic behavior*** resulting in the KPP
- lunch
- reflection and discussion
- integration strategy environments and configurations
- map ***dynamic behavior*** on ***block diagrams*** and ***budget***: quantify contributions to KPP
- elective
- reflection and discussion

day 3

- project management
- re-assess ***risks*** of KPP
- determine an incremental ***integration sequence*** to measure the KPP ASAP
- readiness levels, systems of systems, elective
- reflection and discussion
- lunch
- assess ***integration configurations*** and ***testware, supplier*** and ***logistics*** status, ***technology readiness***, and ***development*** and ***resource*** status
- elective
- reflection and discussion

day 4

- elective
- identify tensions and transform sequence into a ***(PERT) plan***
- elective
- reflection and discussion
- lunch
- elective
- assess ***robustness*** of plan
- make and give presentation to management
- reflection and discussion

people, process, and organization	*technical*	
- human aspects - process and integration - organization	- budgeting - testing - systems of systems	- architecting for integration - impact of change - software and integration product families, platforms

electives

Project Overview How To

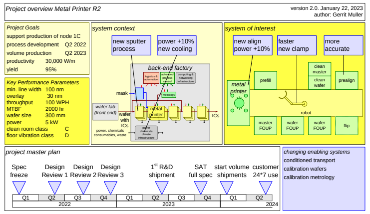
by *Gerrit Muller* USN-SE
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

A project overview shows the overview of a project on a single slide or sheet. The overview helps the team to share the same understanding of scope, objectives, and timeline.

Distribution

March 3, 2024
status: draft
version: 0.2



Project overview Metal Printer R2

version 2.0. January 22, 2023
author: Gerrit Muller

Project Goals

support production of node 1C	
process development	Q2 2022
volume production	Q2 2023
productivity	30,000 W/m
yield	95%

Key Performance Parameters

min. line width	100 nm
overlay	30 nm
throughput	100 WPH
MTBF	2000 hr
wafer size	300 mm
power	5 kW
clean room class	C
floor vibration class	D

changing enabling systems
conditioned transport
calibration wafers
calibration metrology

Project Overview Canvas

Project Title

meta information, e.g. version, date
author, owner

Project Goals

- specific and quantified

Key Performance Parameters

- specific and quantified

system context

- visualization (drawing, block diagram, 3D model, or photo) of the system context
- indication of changes in the context

system of interest

- visualization (drawing, block diagram, 3D model, or photo) of the system
- indication of changes in the system of interest

project master plan with timeline

- timeline with 5 to 10 milestones, especially deliverables
- specific and quantified

optional information, e.g.

- enabling systems
- stakeholders
- external or internal interfaces
- constraints, e.g. applicable legislation

Example Project Overview

Project overview Metal Printer R2

version 2.0. January 22, 2023
author: Gerrit Muller

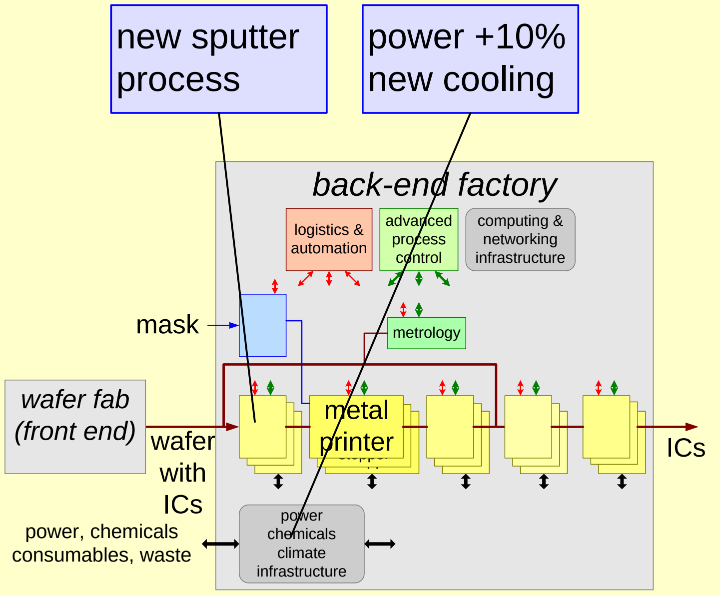
Project Goals

support production of node 1C	
process development	Q2 2022
volume production	Q2 2023
productivity	30,000 W/m
yield	95%

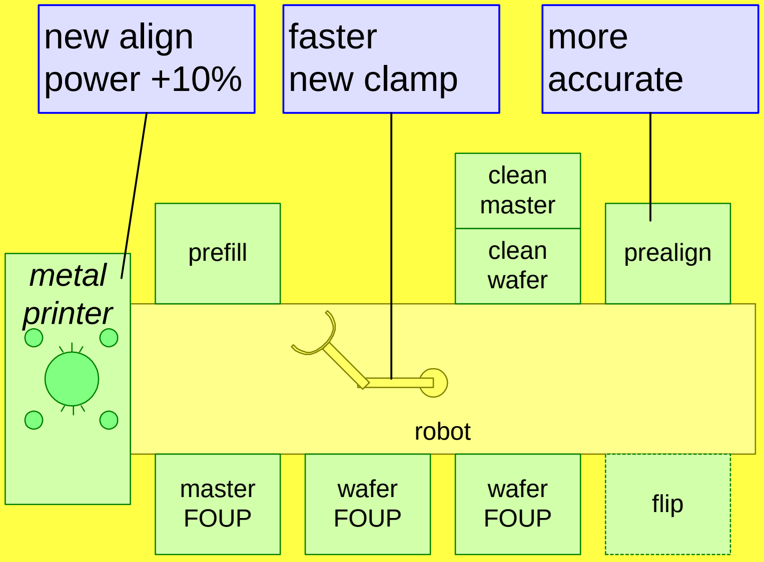
Key Performance Parameters

min. line width	100 nm	
overlay	30 nm	
throughput	100 WPH	
MTBF	2000 hr	
wafer size	300 mm	
power	5 kW	
clean room class		C
floor vibration class		D

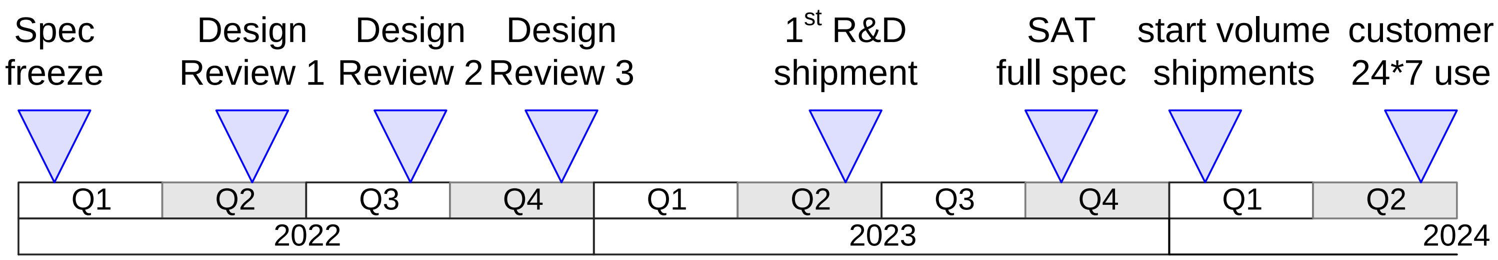
system context



system of interest



project master plan



changing enabling systems

conditioned transport

calibration wafers

calibration metrology

Project Overview Canvas

Project Title

meta information, e.g. version, date
author, owner

Work Breakdown Structure

- visualization
- *builds upon the Product Breakdown Structure*

Project Master Plan

- PERT plan with major milestones

project organization

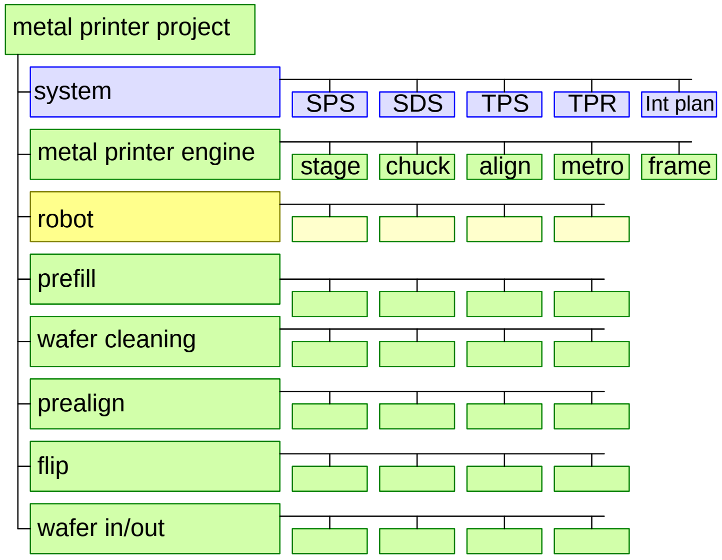
- allocation of roles
- specific additions or deviations

Example Project Overview

Metal Printer

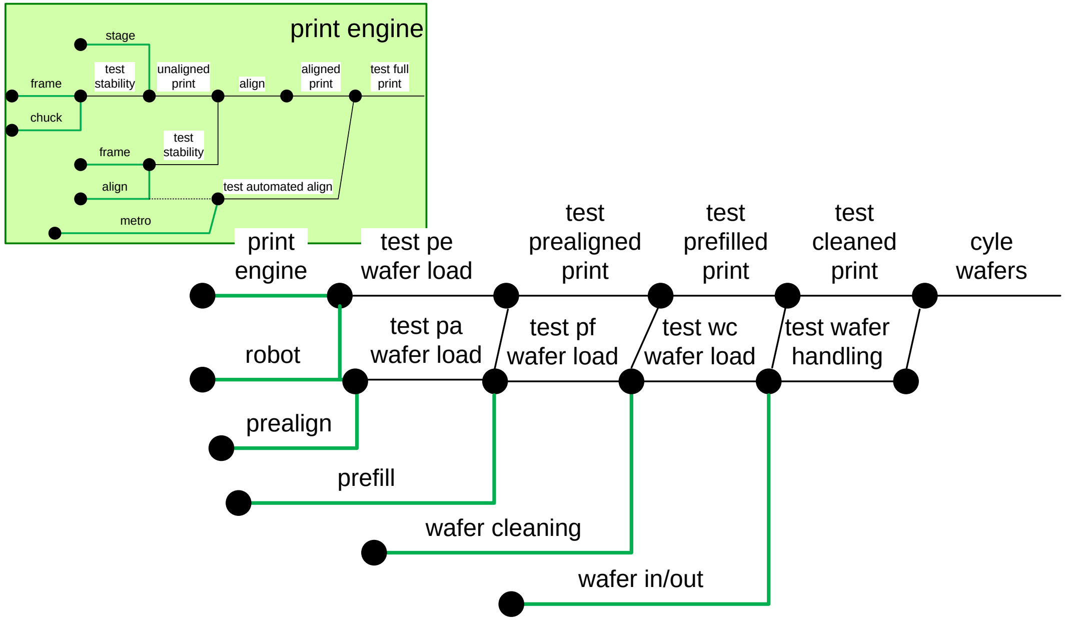
version 0.1, 2023-02-11
author: Gerrit Muller

Work Breakdown Structure



- metal printer project
  - system
    - SPS
    - SDS
    - TPS
    - TPR
    - Int plan
  - metal printer engine
    - stage
    - chuck
    - align
    - metro
    - frame
  - robot
  - prefill
  - wafer cleaning
  - prealign
  - flip
  - wafer in/out

Project Master Plan



project organization

Project Leader: P.L. Eader

Product Manager: P.M. Anager

Architect: Archie Tect

Mastering Systems Integration; Assignments

by *Gerrit Muller* TNO-ESI, USN-SE
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

All assignments of the course Mastering Systems Integration.

Distribution

March 3, 2024
status: draft
version: 0.5

logo
TBD

Pre-assignment

Make an initial project overview for your own project

Project Title

meta information, e.g. version, date, author, owner

Project Goals

- 3 to 5 specific and quantified objectives

Key Performance Parameters

- 5 to 10 specific and quantified requirements

system context

- sketch the next generation system context, e.g. a drawing. block diagram, 3D model, photo of the system context
- indicate changes in the context compared to the current generation system

system of interest

- sketch your next generation system, e.g. drawing. block diagram, 3D model, photo
- indicate changes compared to the current generation system

project master plan with timeline

- time line with 5 to 10 milestones, especially deliverables. specific and quantified
- first light, prototype shipment, 1st SAT @OEM, 1st SAT @OEM's customer, start volume production

other relevant project information

- enabling systems
- stakeholders
- ecternal or internal interfaces
- constraints, e.g. applicable legislation

Discuss the Case

Sketch the system-of-interest

What are the most relevant project goals?

Sketch the project master plan (the main milestones and their timing)

Determine KPPs

Determine 5 to 10 ***Key Performance Parameters*** (KPP) of the System

Quantify these KPPs

Define the KPPs roughly, using a ***Use Case***

VUCA Causes Risks

VUCA =

- Volatility
- Uncertainty
- Complexity
- Ambiguity

Assess Risks of KPPs

Assess the risk for each KPP

Explain why this KPP may suffer from this risk

Select one KPP to work on in the remainder of the Face-to-Face workshop

- this KPP should be “hot” (lot of organizational buzz)
- you may also select two “conflicting” KPPs

Describe Typical Use

Define the typical use (by customer stakeholders) of the system in relation to the selected KPP.

This use case helps to define the KPP further

This use case will guide the verification and validation

Make Block Diagrams

Make block diagrams of the system, the software, and the hardware.

Block diagrams show parts, and interfaces or connections.

These block diagrams need tens of blocks.

Model 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 Budget

Map the Dynamic Behavior on the block diagrams.

Transform this into a budget:

- Quantify contributions of parts and functions to the KPP.

Re-assess Risks

Re-assess the risks for the chosen KPP

using the insights gained so far

These risks are leading when defining the integration sequence

Determine an Incremental Integration Sequence

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.

Assess Other Planning Perspectives

assess the planning from the perspectives:

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

Transform into PERT plan

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.

Assess Robustness

Assess how well the PERT plan addresses the original risks in the KPPs.

Assess the robustness of the PERT-plan for changes.

All assumptions in the integration plan will probably change. A good integration PERT-plan shouldn't change much.

Prepare Final Presentation

Prepare a presentation for the management

- to communicate the systems integration approach
- its rationale
- and its impact on the project, the test configurations, the schedule, the organization, and the suppliers

Add a slide on the course learnings and reflections

Content for Final Presentation

- Goal of this presentation, main message, desired outcome
- Mission/goal of end customer
- Master plan, milestones and dates
- Key performance parameters of the system
- Block diagrams (20..30 blocks), dynamic behavior (20..30 blocks), some supporting visualizations, budget
- Integration strategy and sequence
- PERT plan (proposal)
- Summary and specific actions, and recommendations (e.g. allocate these resources, order …, etc.)

Add a slide on the course learnings and reflections

Homework Project

A4 Project/System Overview

A4 Master plan with time lines

Project description

- Architecture overview (enabling systems, context, SOI, partitioning, interfaces,dynamic behavior, KPP's, MOEs, critical Design and Technology choices) + Rationale
- Risk Analysis + Rationale
- Ideal integration sequence PERT plan + Rationale
- Project plan inputs (resources, suppliers, customers, technologies, test mgmt...) + Rationale
- Project plan + Rationale

Evaluation of the project description (how mature, how accurate, where are gaps?)

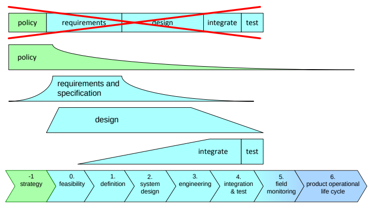
Reflection report

Mastering Systems Integration; Process and Positioning

by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

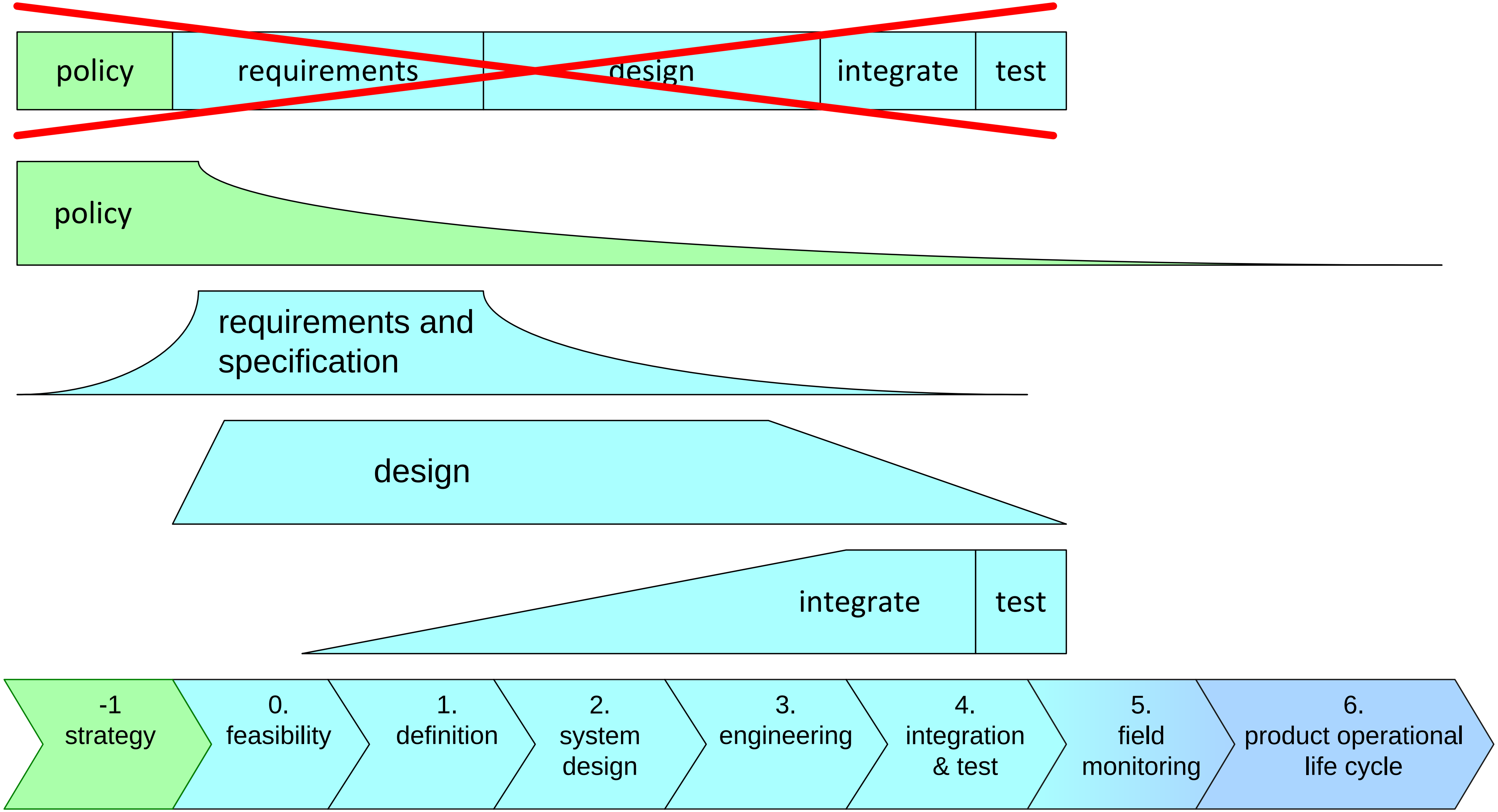
This lesson positions systems integration as process in the development processes.



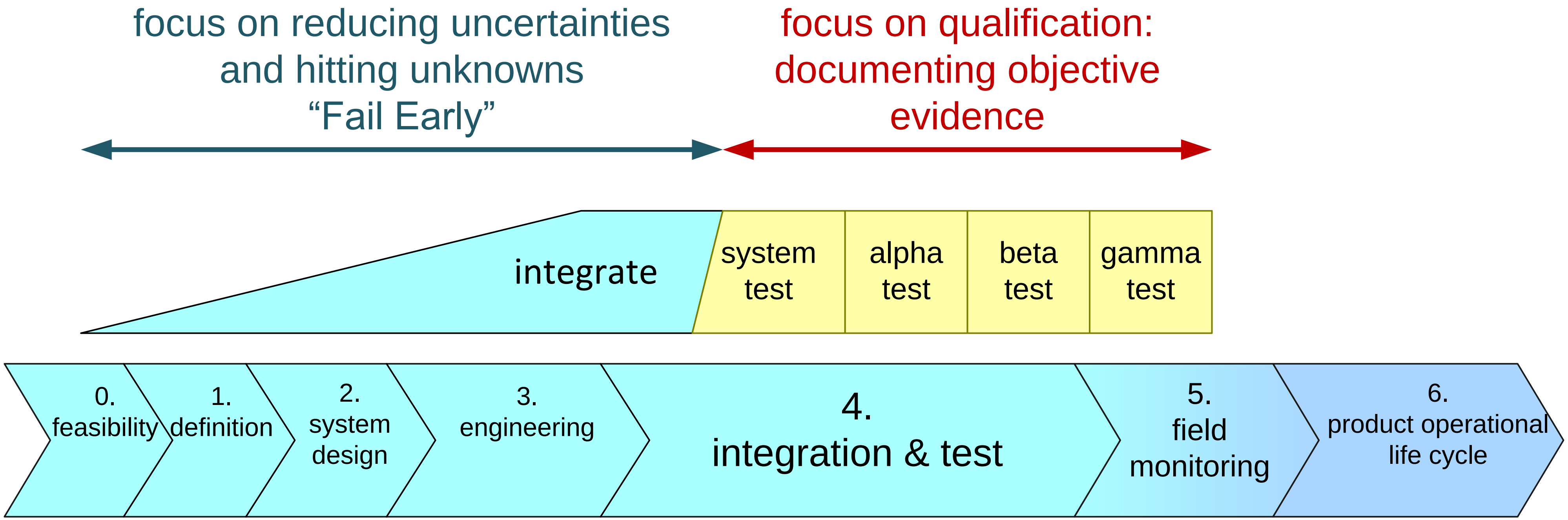
March 3, 2024
status: planned
version: 0.3

Distribution

Typical Concurrent Product Creation Process



Zooming in on Integration and Tests



focus on reducing uncertainties and hitting unknowns "Fail Early"

focus on qualification: documenting objective evidence

integrate

system test

alpha test

beta test

gamma test

0. feasibility

1. definition

2. system design

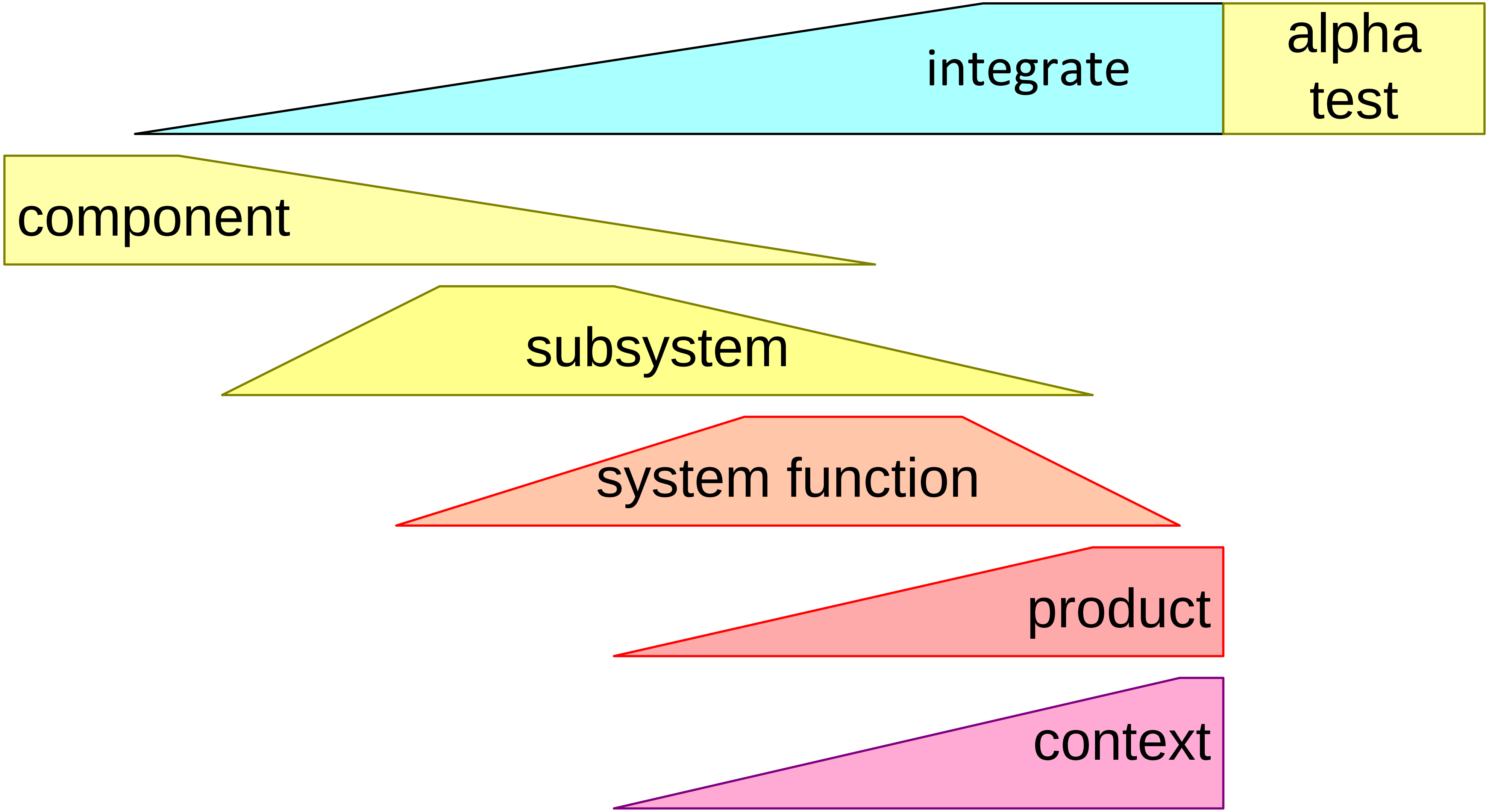
3. engineering

4. integration & test

5. field monitoring

6. product operational life cycle

Integration Takes Place in a Bottom-up Fashion



Mastering Systems Integration; Hardware, Software, Systems!

by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

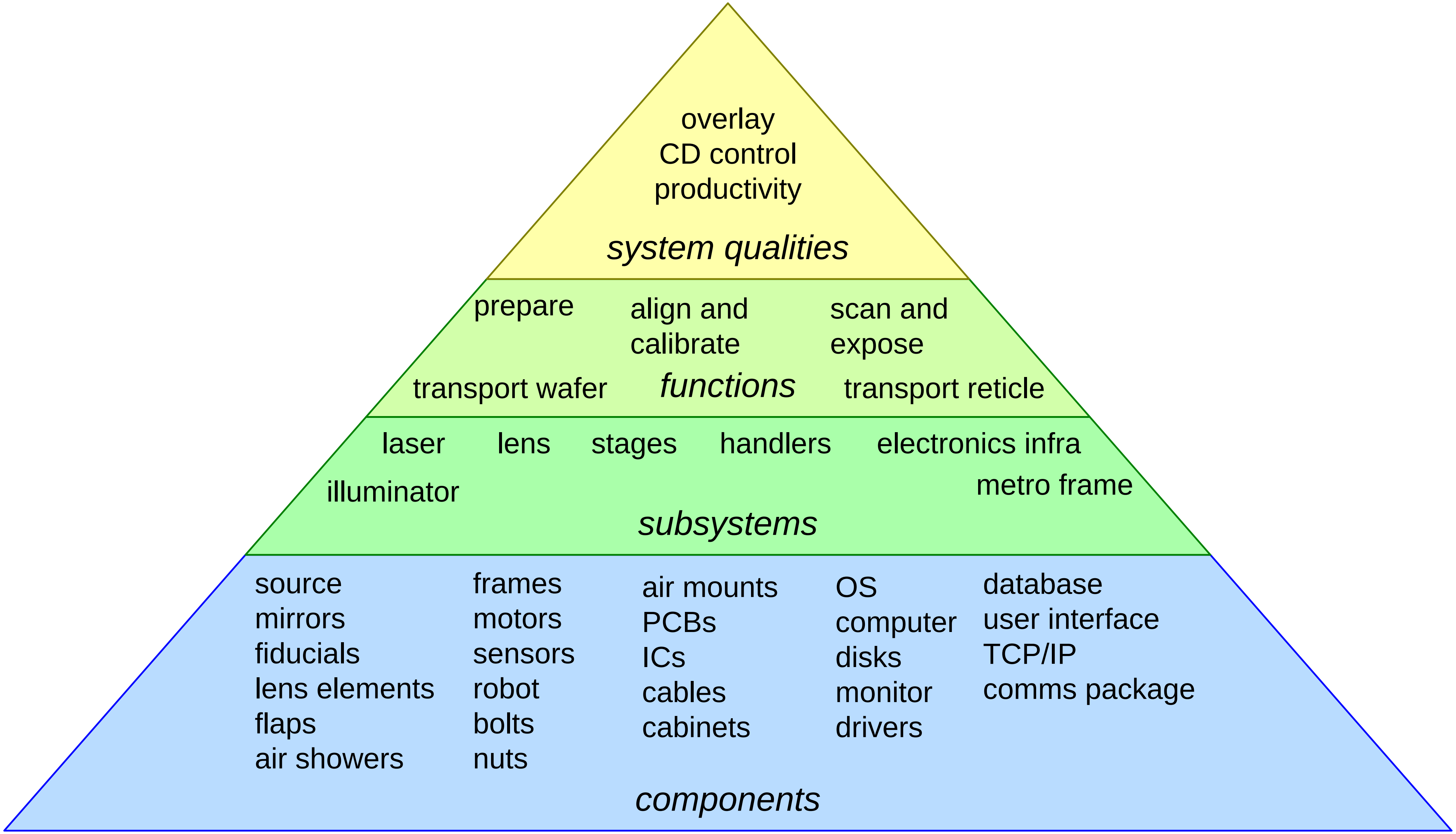
Hardware and software differ in their characteristics, which impacts their role in systems integration. The main message in this lesson is that the focus of integration is at the system, where both hardware and software contribute.

Distribution

March 3, 2024
status: planned
version: 0.1

logo
TBD

From Components to System Qualities



overlay
CD control
productivity

system qualities

prepare
align and calibrate
scan and expose
transport wafer
functions
transport reticle

laser
lens
stages
handlers
electronics infra
illuminator
metro frame
subsystems

source
mirrors
fiducials
lens elements
flaps
air showers

frames
motors
sensors
robot
bolts
nuts

air mounts
PCBs
ICs
cables
cabinets

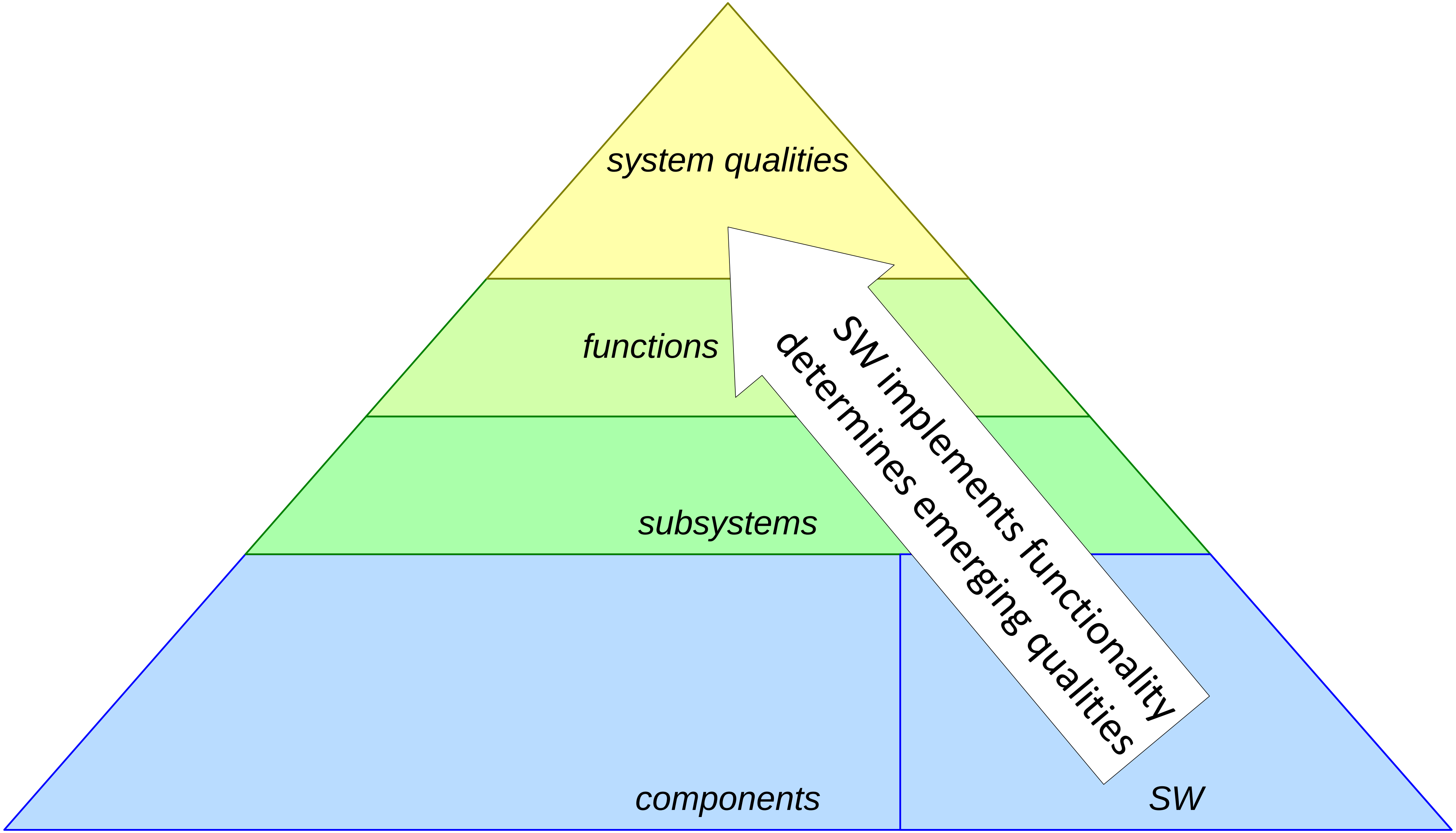
OS
computer
disks
monitor
drivers

database
user interface
TCP/IP
comms package

components

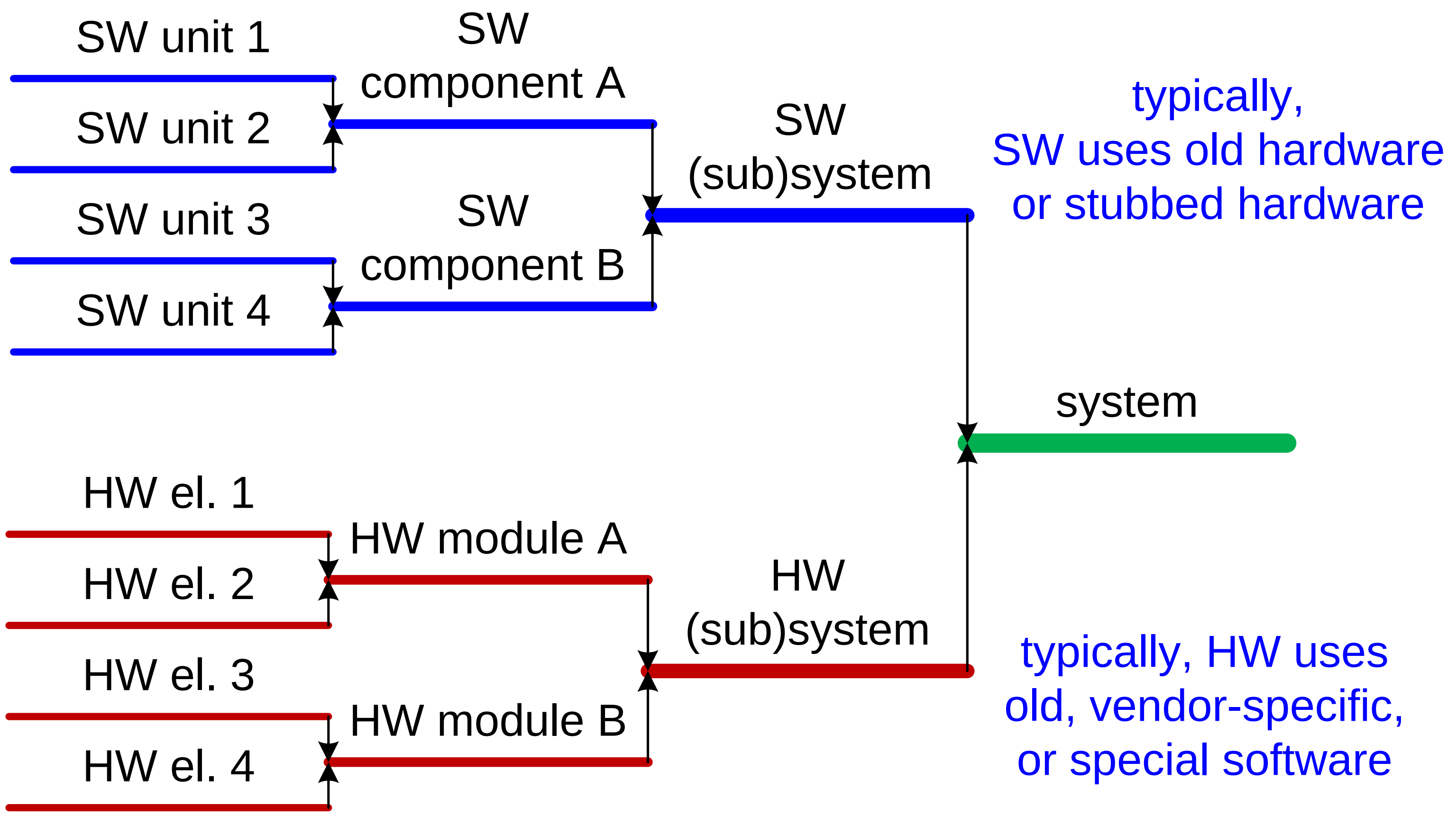
version: 0.1
March 3, 2024
ATlayers

Role of Software



version: 0.1
March 3, 2024
TSAITlayers

Pitfall: Late HW-SW Integration

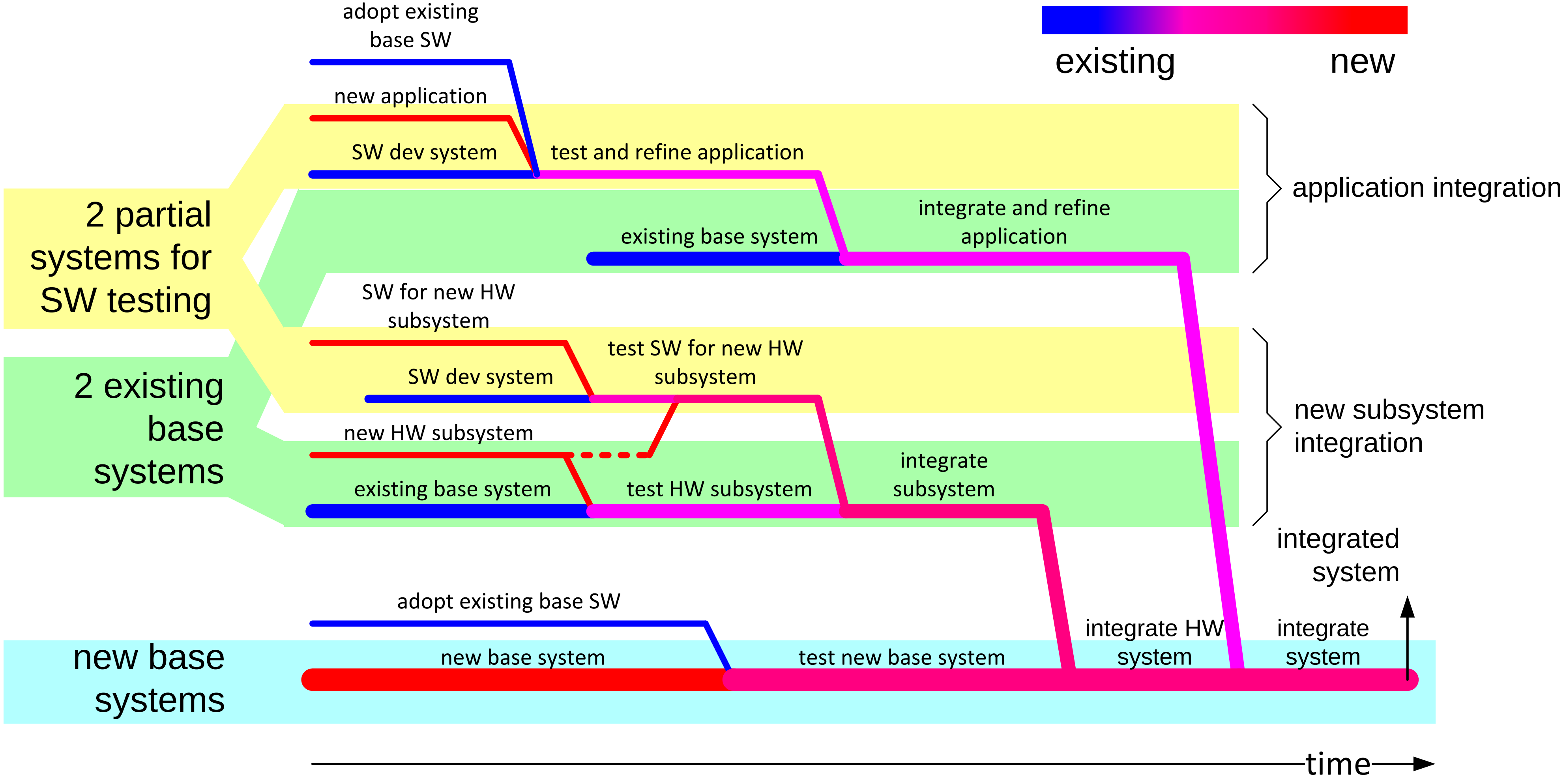


typically, SW uses old hardware or stubbed hardware

typically, HW uses old, vendor-specific, or special software

Segregation of hardware and software is a typical organizational problem. Such segregation ignores close coupling of hardware and software.	Erroneous assumptions about hardware are discovered late. Key performance parameters are visible late.

Transition from Previous System to New System



Mastering Systems Integration; Terminology

by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

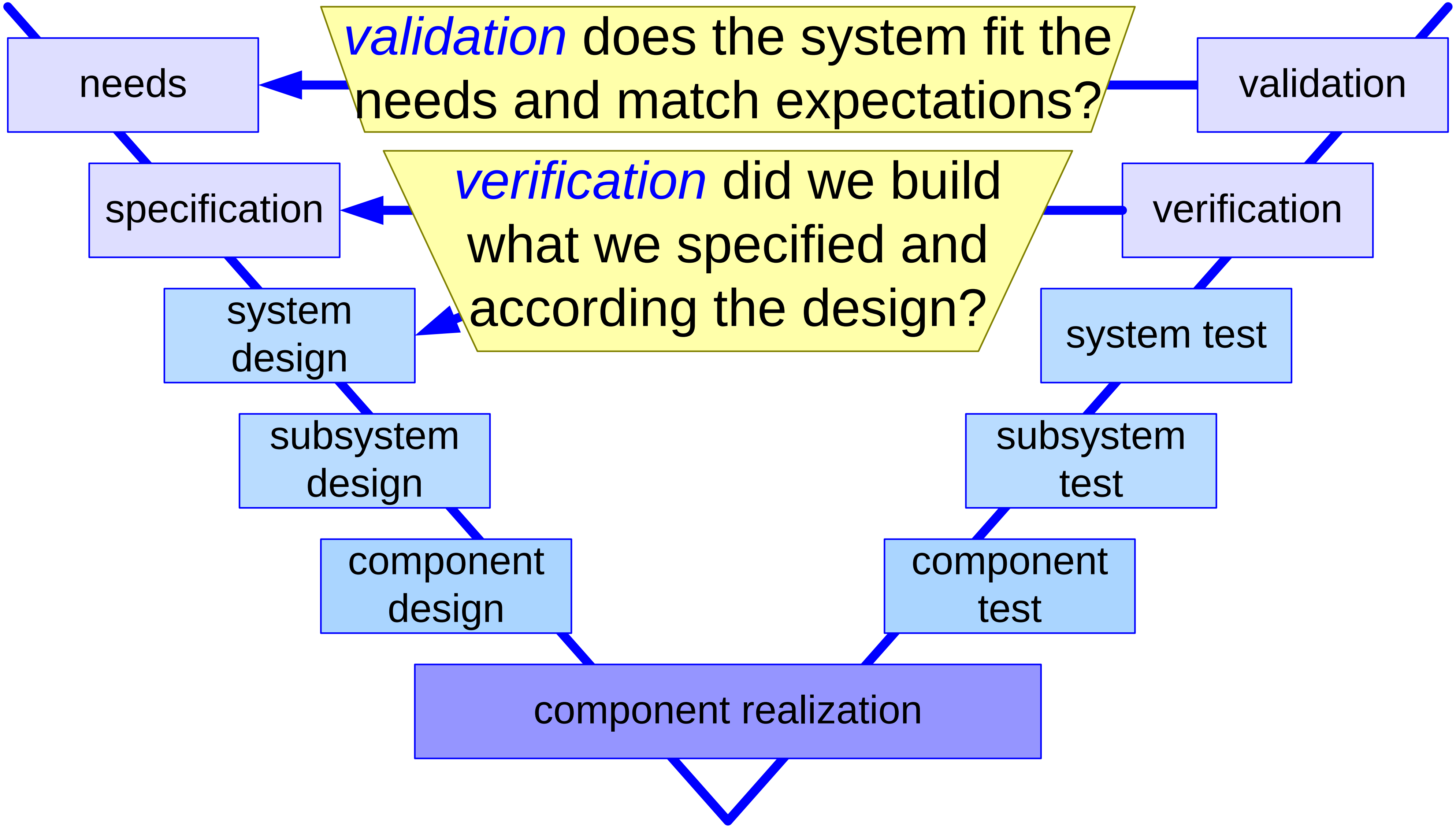
Abstract

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

March 3, 2024
status: planned
version: 0

logo
TBD

Validation and Verification in the V-model

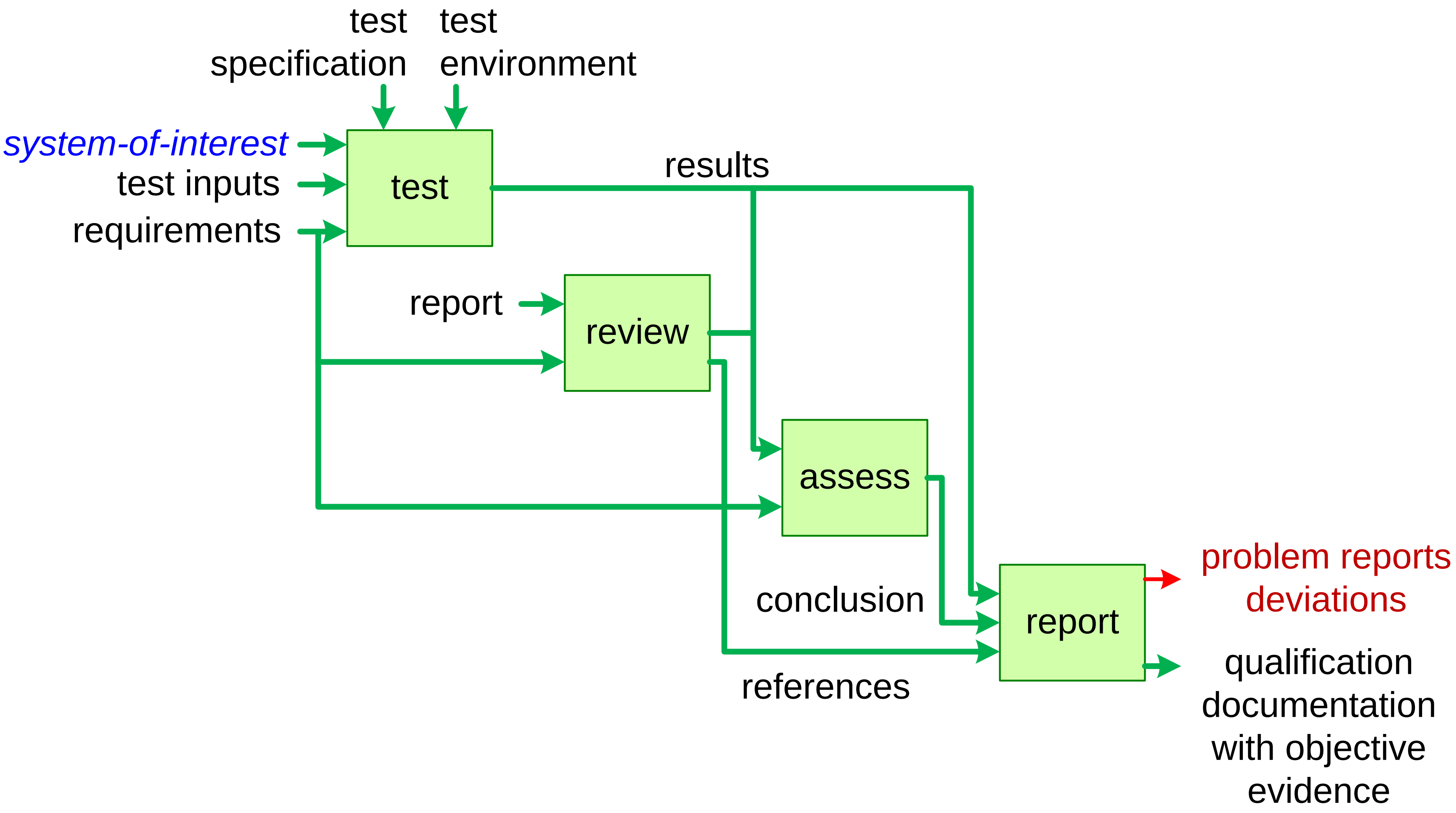


validation does the system fit the needs and match expectations?

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

version: 0
March 3, 2024
VVTvModel

Functional Model of Verification

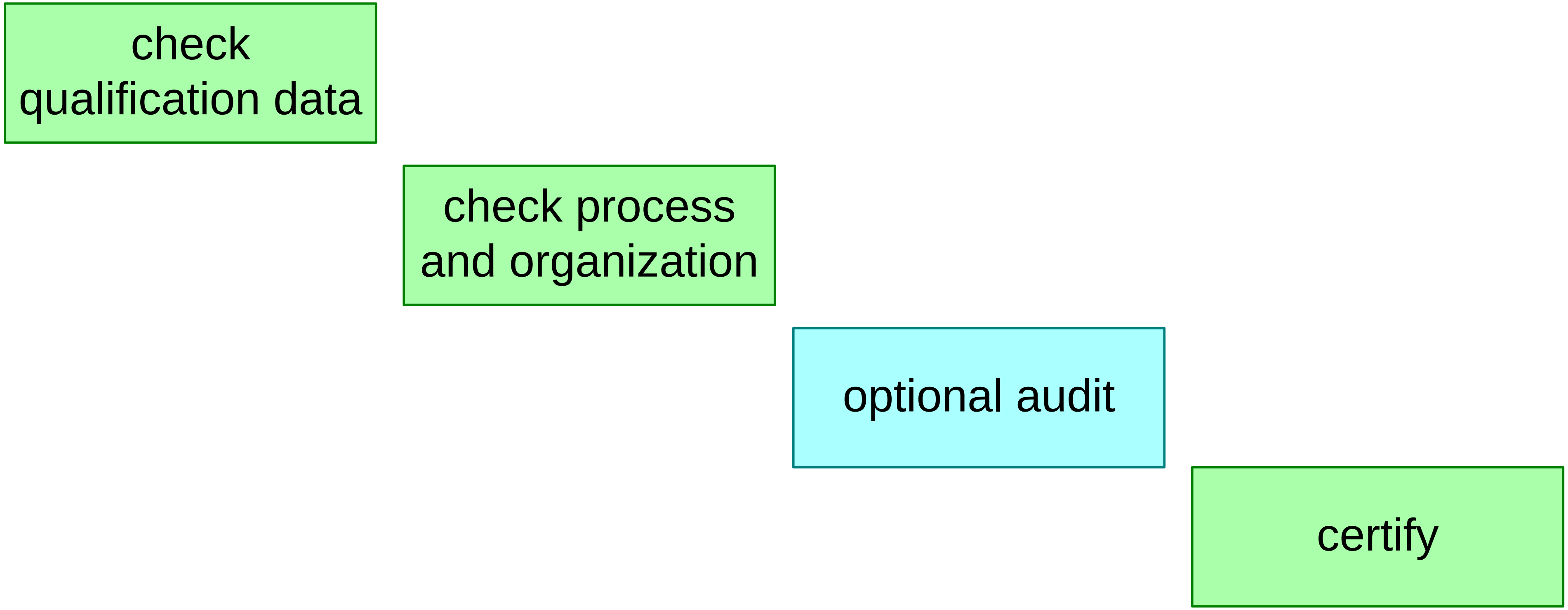


version: 0
March 3, 2024
VVTverification

Certification

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.



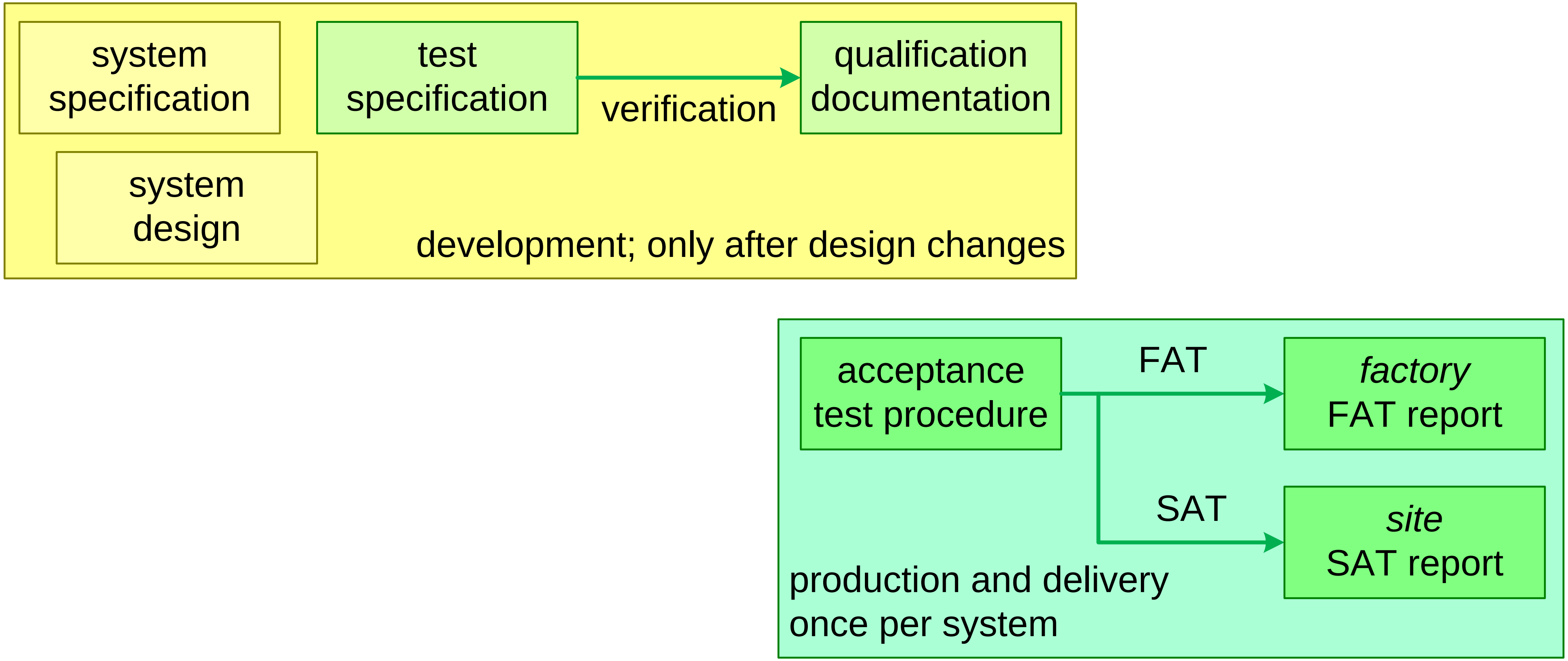
check qualification data

check process and organization

optional audit

certify

Development and (repeated) Production



system specification

test specification

verification

qualification documentation

system design

development; only after design changes

acceptance test procedure

FAT

factory FAT report

SAT

site SAT report

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 ItIs and What It Definitely Is Not),
by Denise Dion http://www.eduquest.net/Advisories/EduQuest%20Advisory_ObjectiveEvidence.pdf

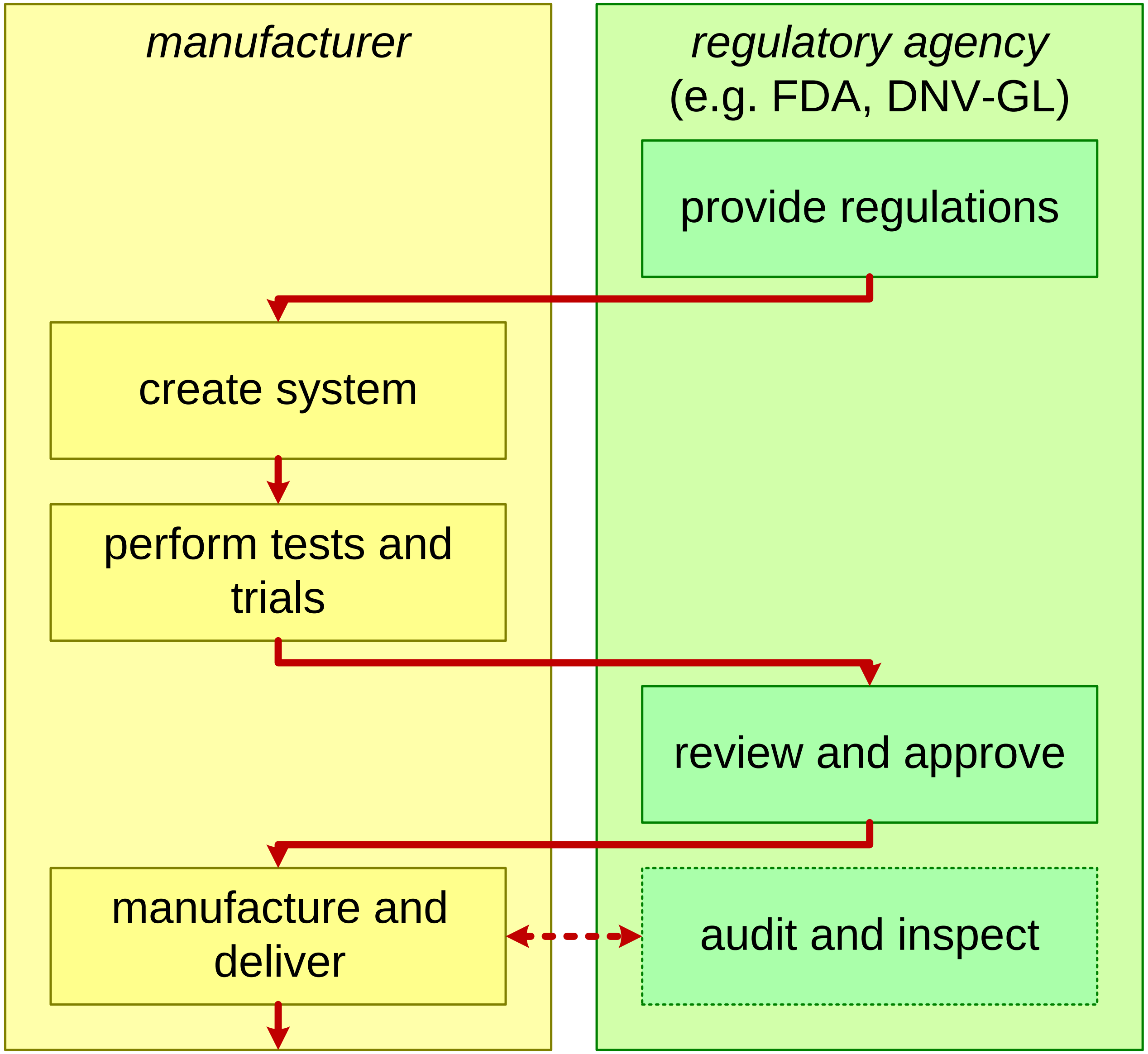
FDA Requirements for Objective Evidence

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 re-creation 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

regulatory agency
(e.g. FDA, DNV-GL)

provide regulations

create system

perform tests and trials

review and approve

manufacture and deliver

audit and inspect

version: 0
March 3, 2024
MSITMapprovalProcess

Mastering Systems Integration; Economic Perspective

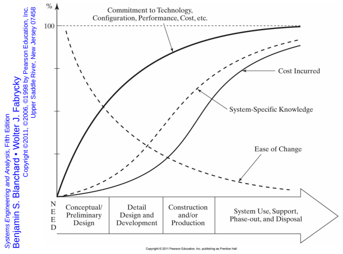
by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

This presentation discusses economic aspects related to integration.

Distribution

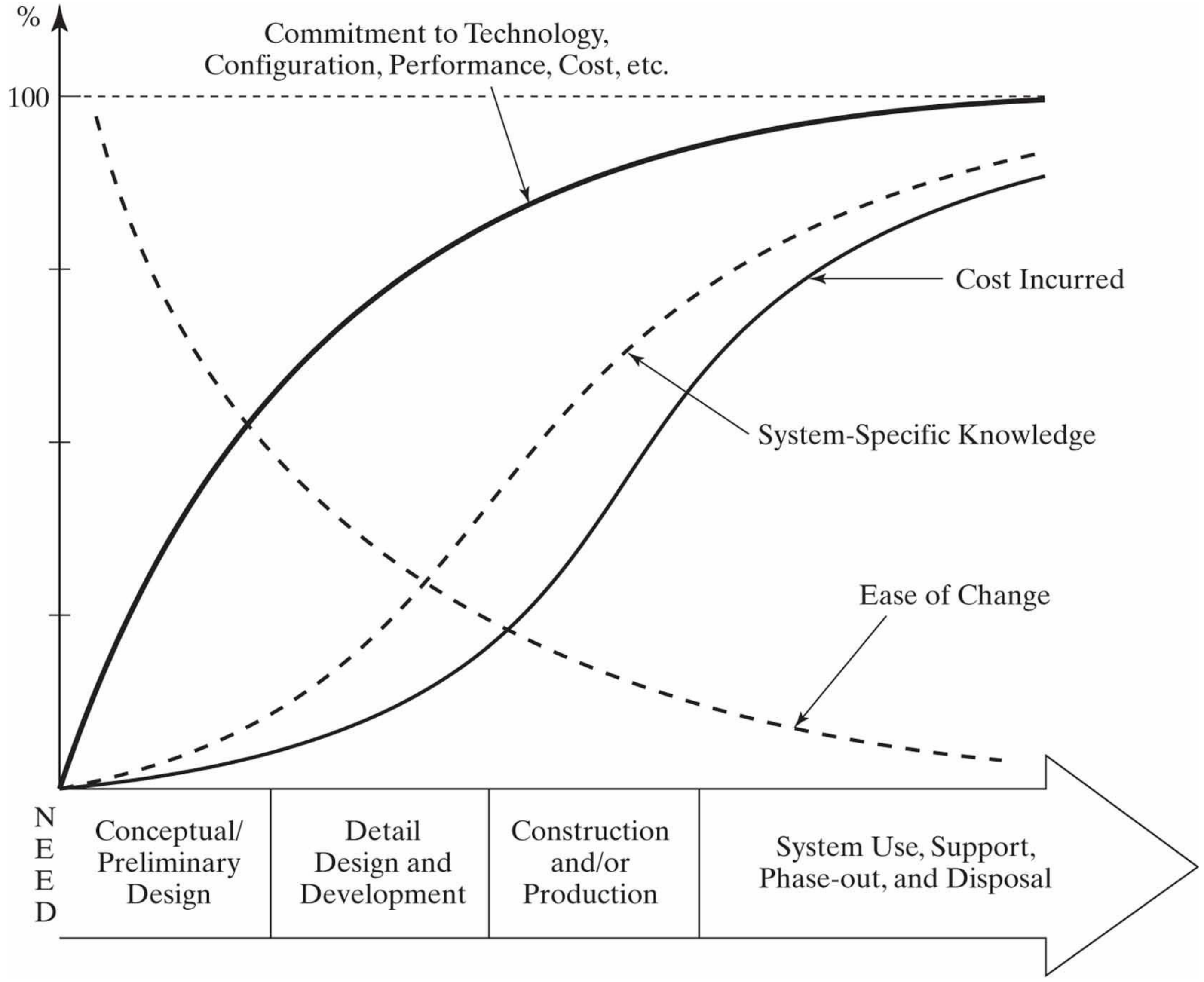
March 3, 2024
status: planned
version: 0.1



Systems Engineering and Analysis, Fifth Edition
Benjamin S. Blanchard • Wolter J. Fabrycky
Copyright ©2011, ©2006, ©1998 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458

Life-cycle commitment, knowledge, and incurred cost

Systems Engineering and Analysis, Fifth Edition
Benjamin S. Blanchard • Wolter J. Fabrycky
Copyright ©2011, ©2006, ©1998 by Pearson Education, Inc.
Upper Saddle River, New Jersey 07458



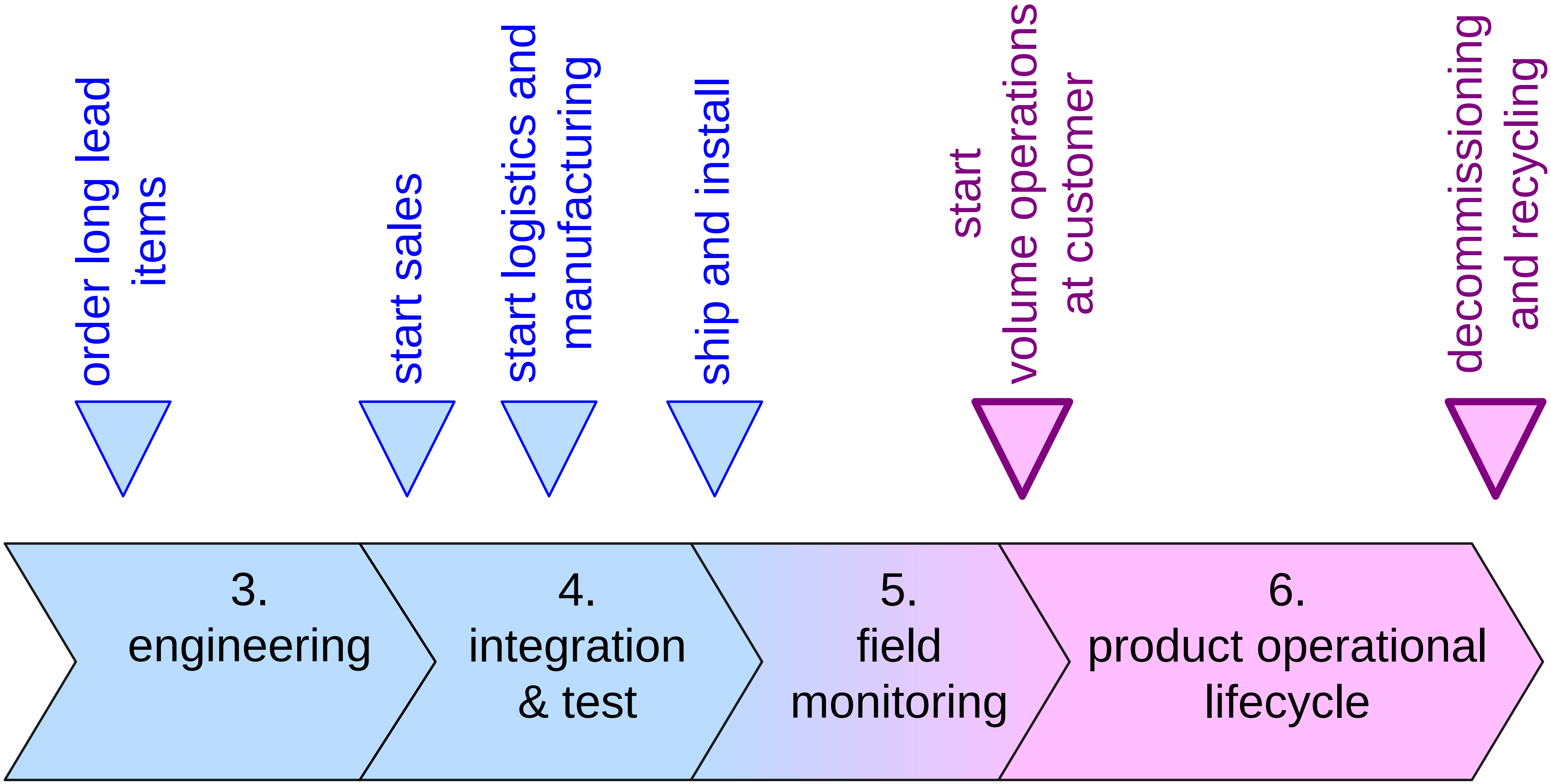
Delays are Expensive

Importance of Time to Market

- No loss of value by customer
- Return on investement & profit ASAP
- Competitive edge, market share
- Free up critical resources for next projects ASAP
- Obsolesence

Cost Related Milestones

What knowledge does the project need to make these critical steps?



Visualizing Dynamic Behavior

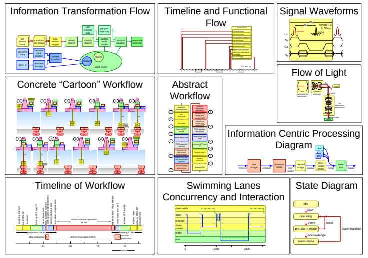
by *Gerrit Muller*   TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

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.

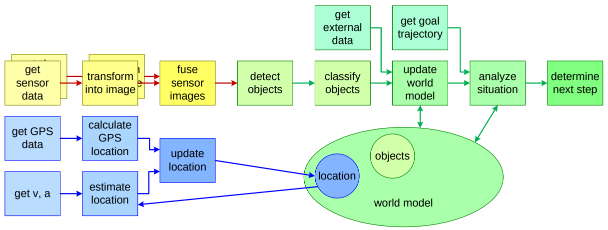
Distribution

March 3, 2024
status: preliminary draft
version: 0

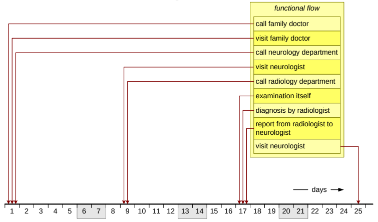


Overview of Visualizations of Dynamic Behavior

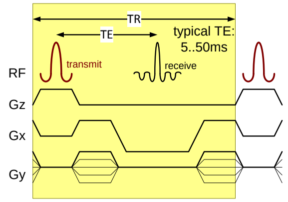
Information Transformation Flow



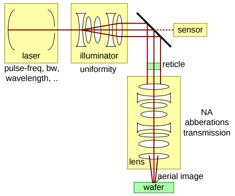
Timeline and Functional Flow



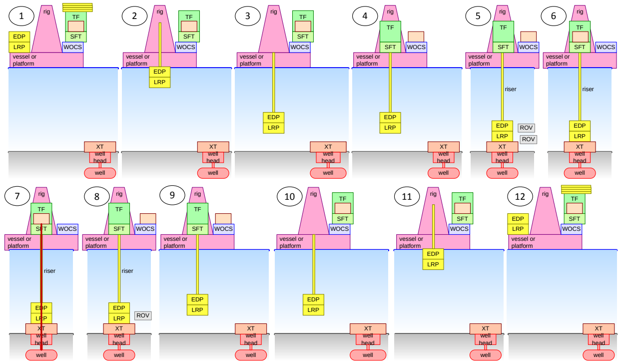
Signal Waveforms



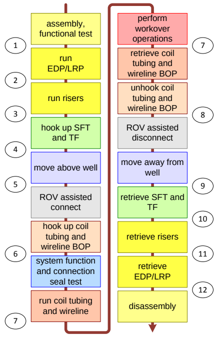
Flow of Light



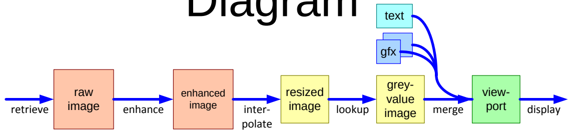
Concrete "Cartoon" Workflow



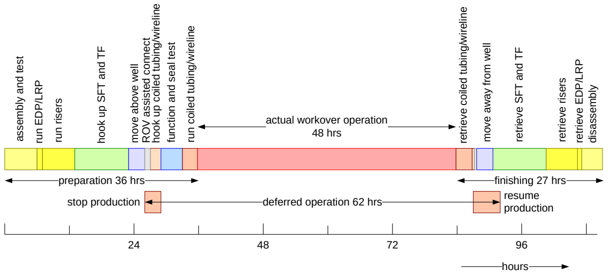
Abstract Workflow



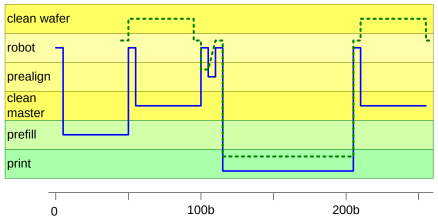
Information Centric Processing Diagram



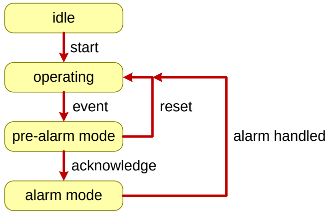
Timeline of Workflow



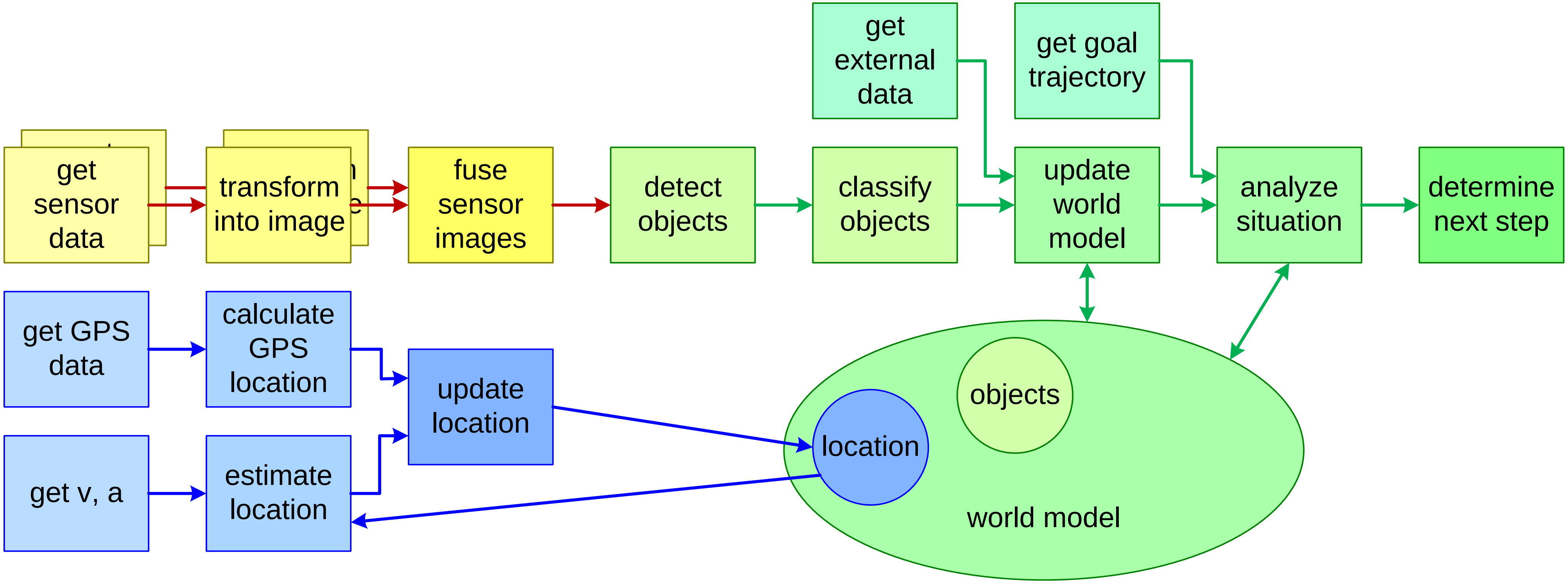
Swimming Lanes Concurrency and Interaction



State Diagram



Example Functional Model of Information Flow



get sensor data

transform into image

fuse sensor images

detect objects

classify objects

get external data

get goal trajectory

update world model

analyze situation

determine next step

get GPS data

calculate GPS location

update location

get v, a

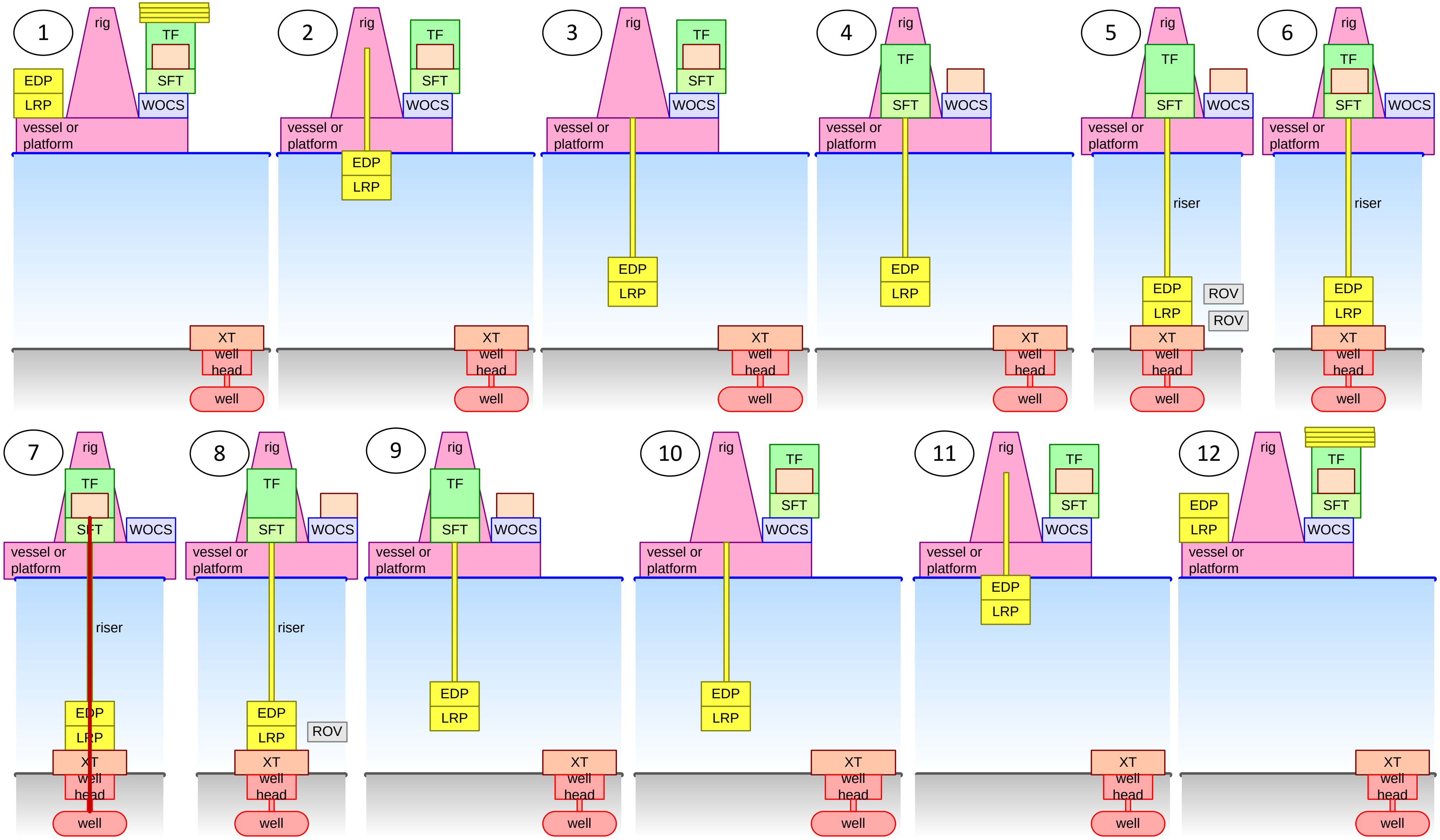
estimate location

objects

location

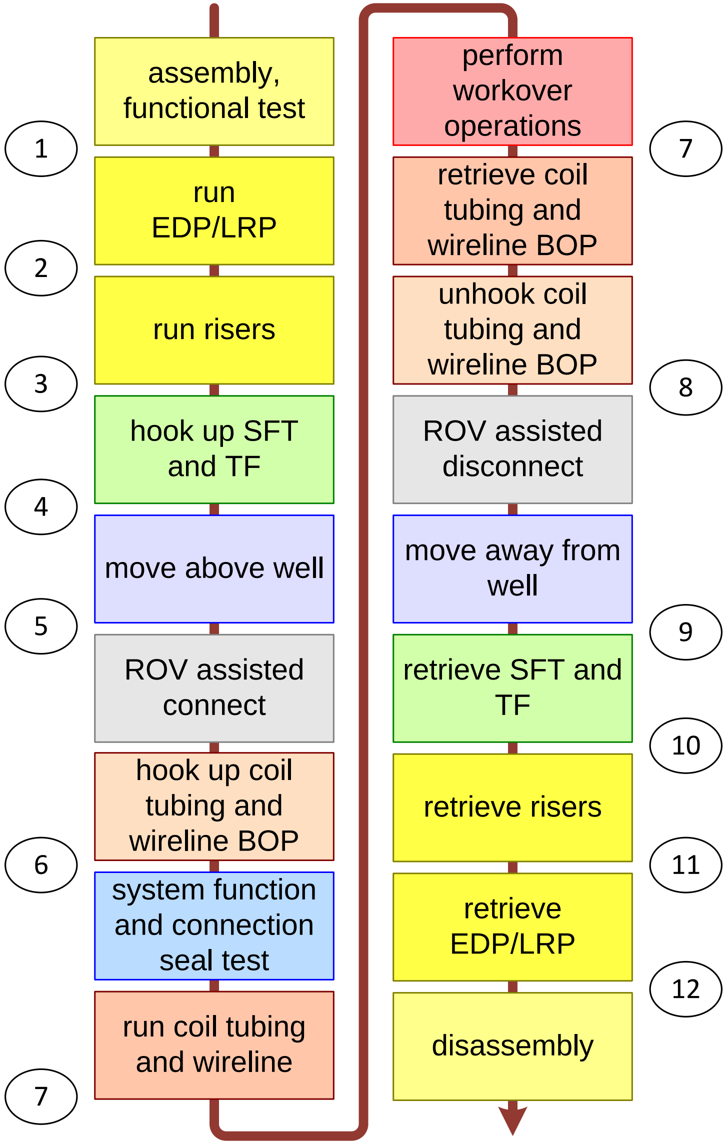
world model

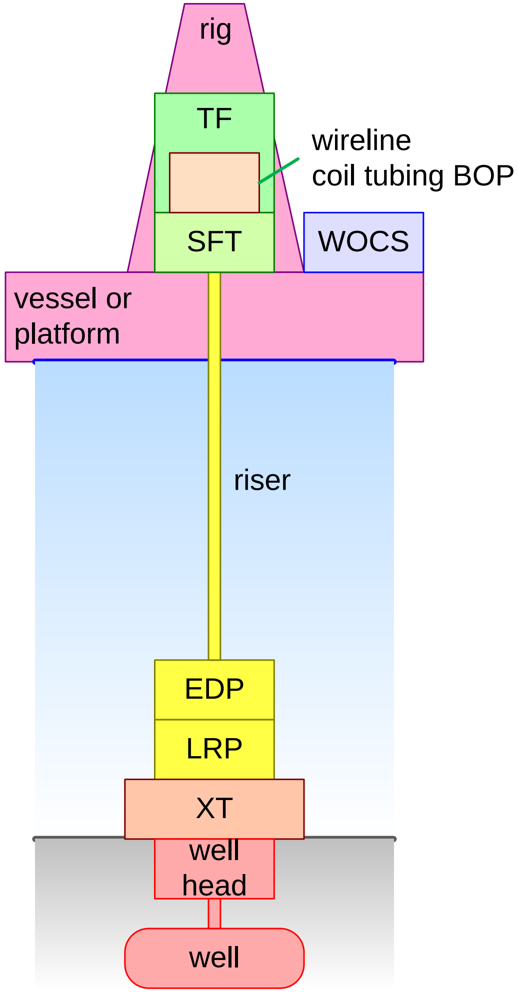
”Cartoon” Workflow



version: 0
March 3, 2024
SSMEtypicalWorkoverOperationCartoon

Workflow as Functional Model





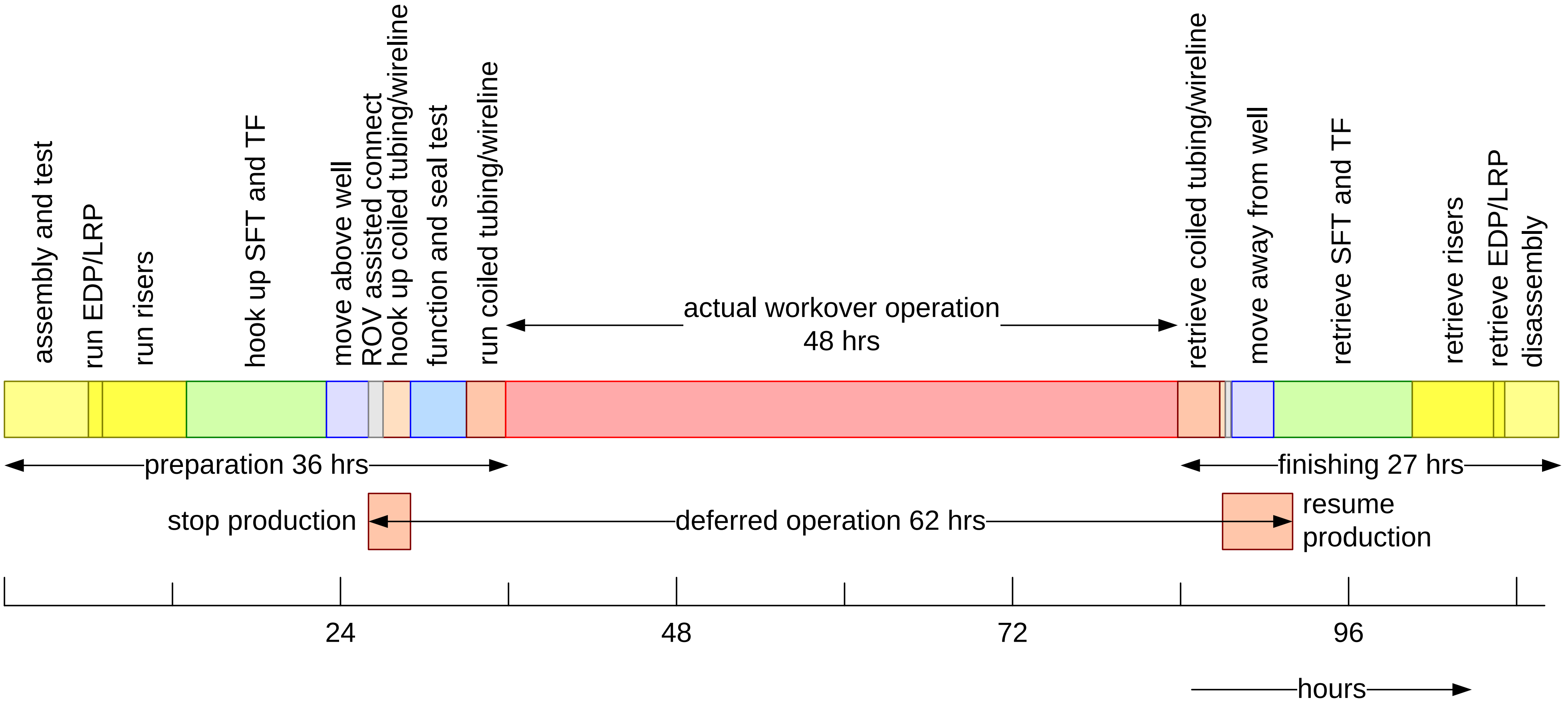
Workflow as Timeline

assumptions:

running and retrieving risers: 50m/hr

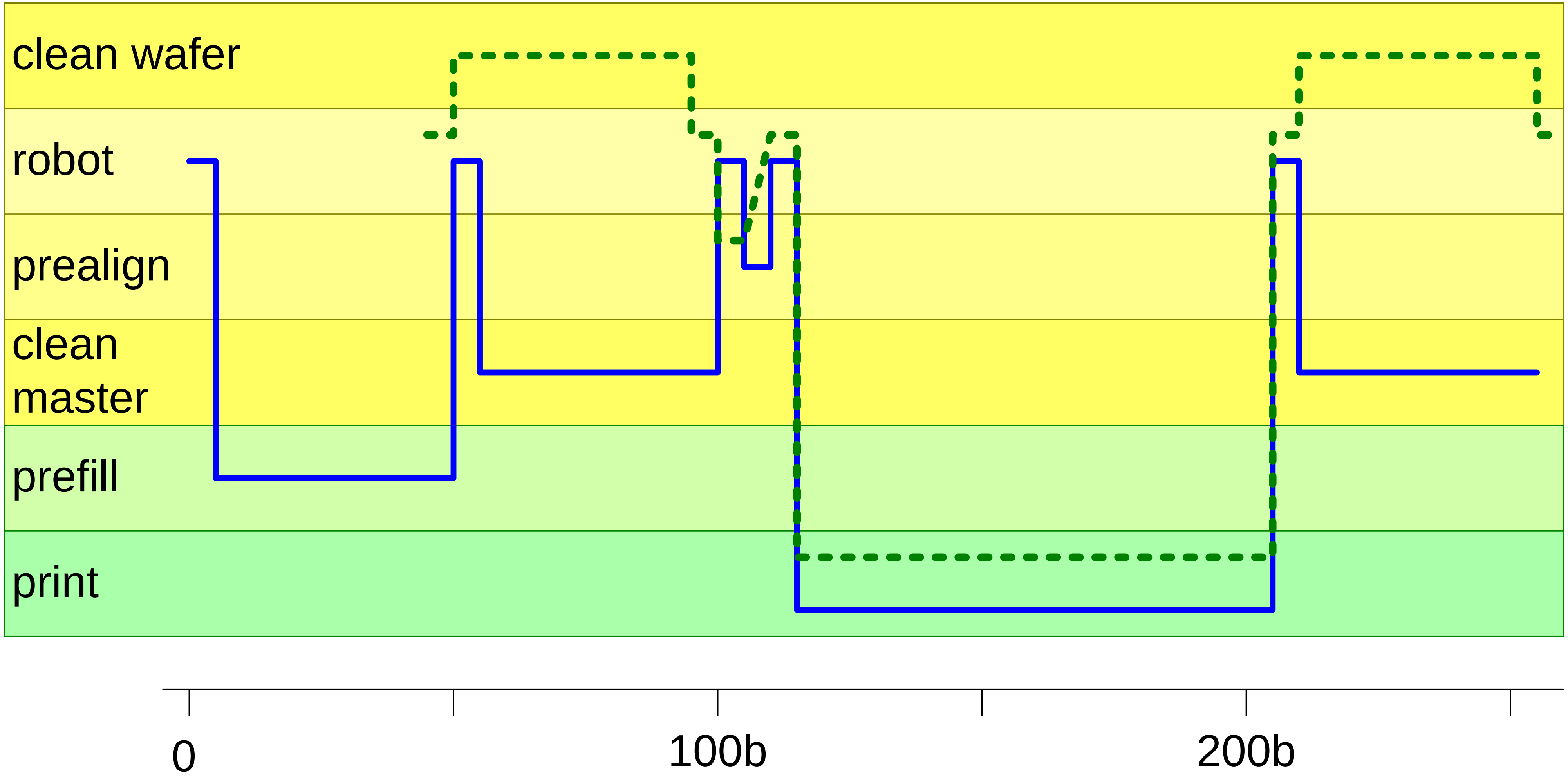
running and retrieving coiled tubing/wireline: 100m/hr

depth: 300m

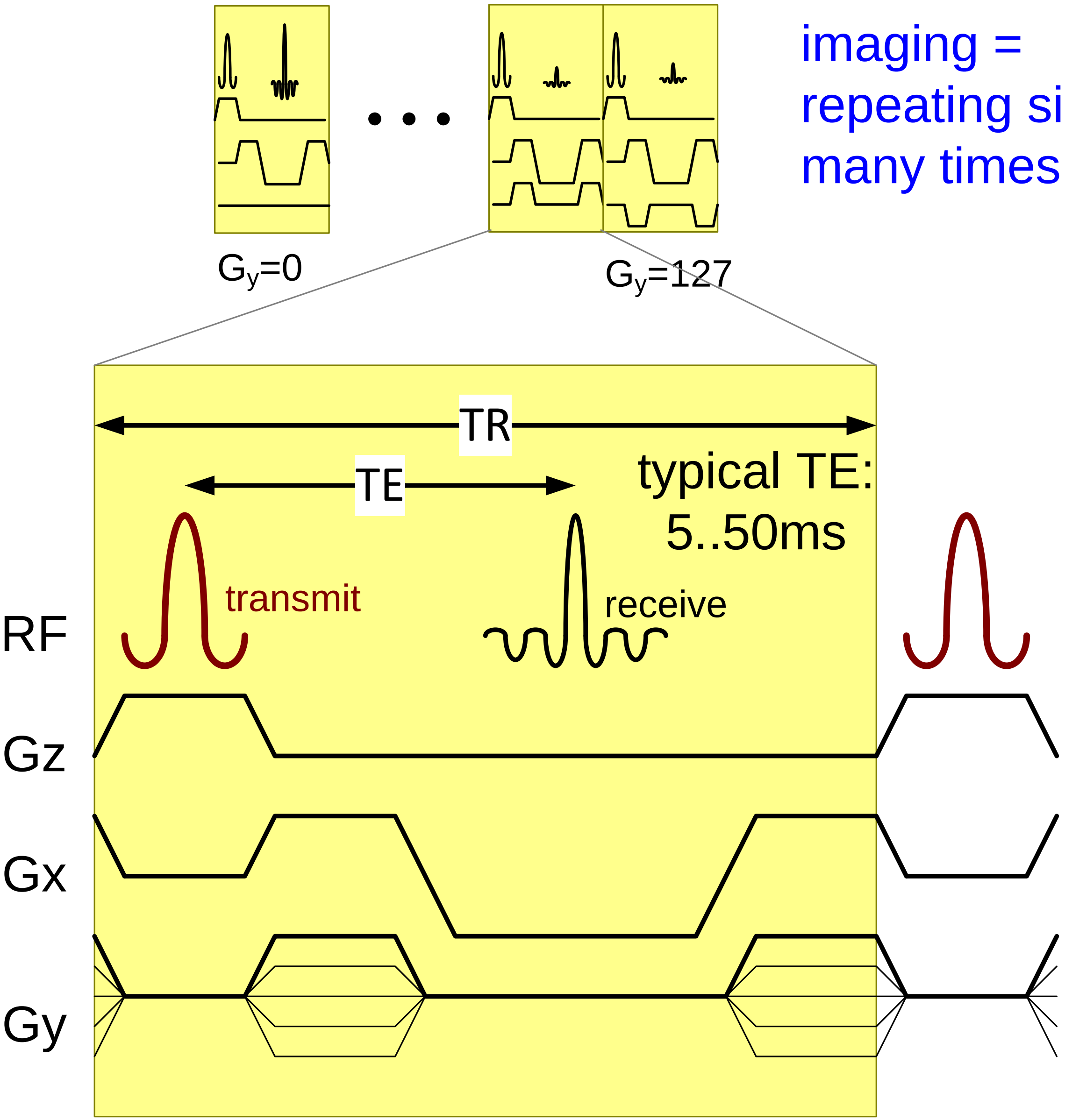


version: 0
March 3, 2024
SSMEtypicalWorkoverOperationTimeline

Swimming Lane Example

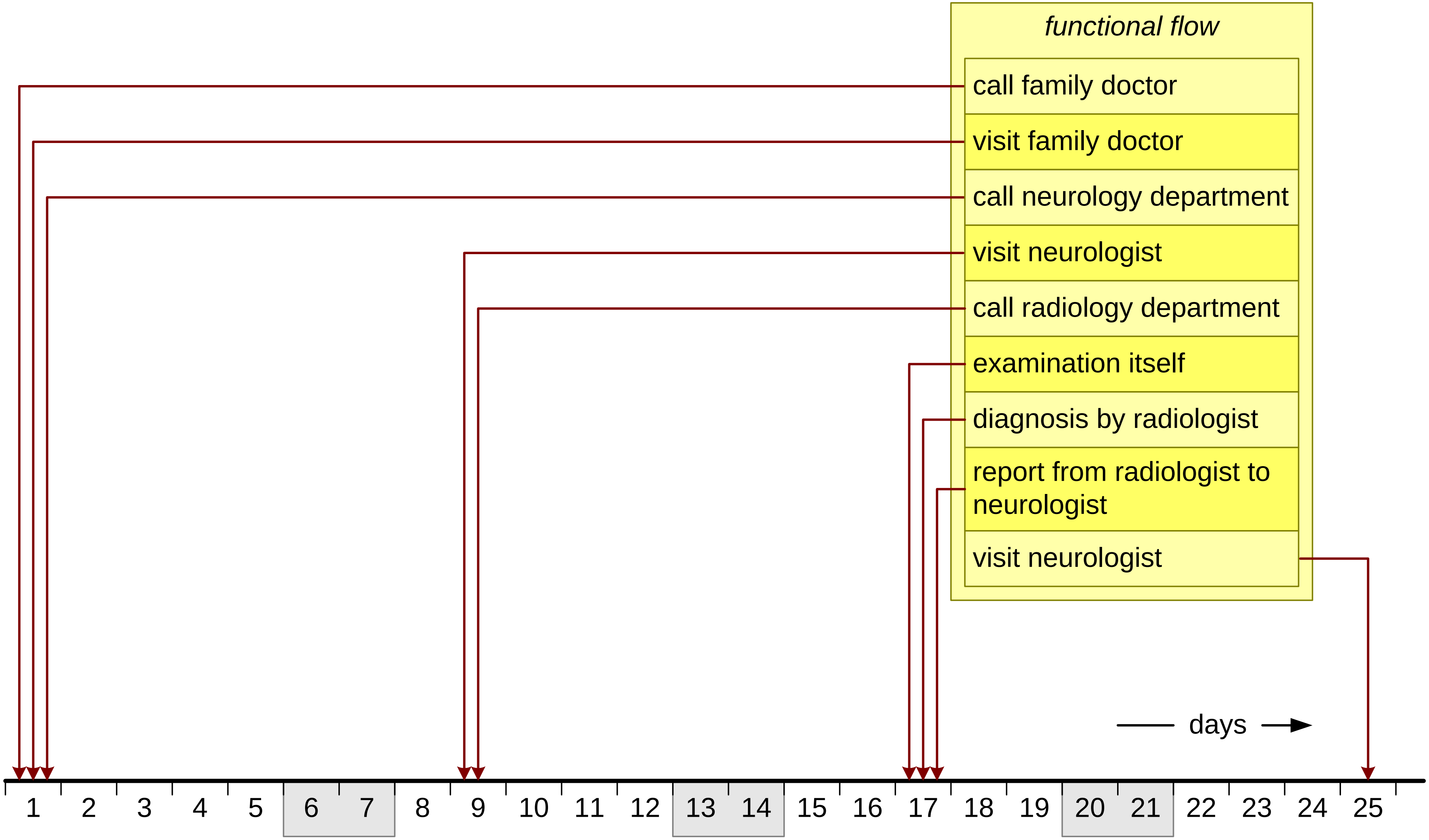


Example Signal Waveforms



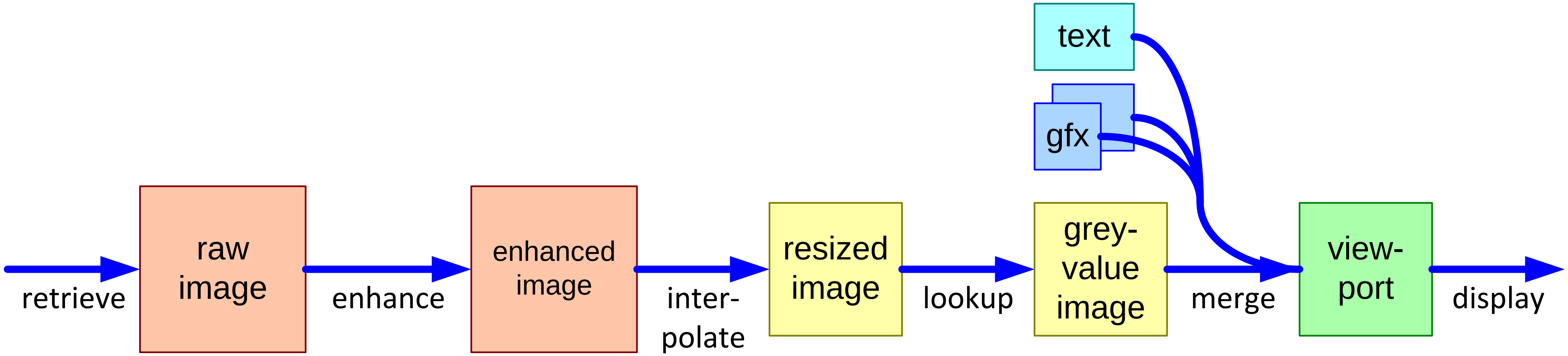
imaging =
repeating similar pattern
many times

Example Time Line with Functional Model

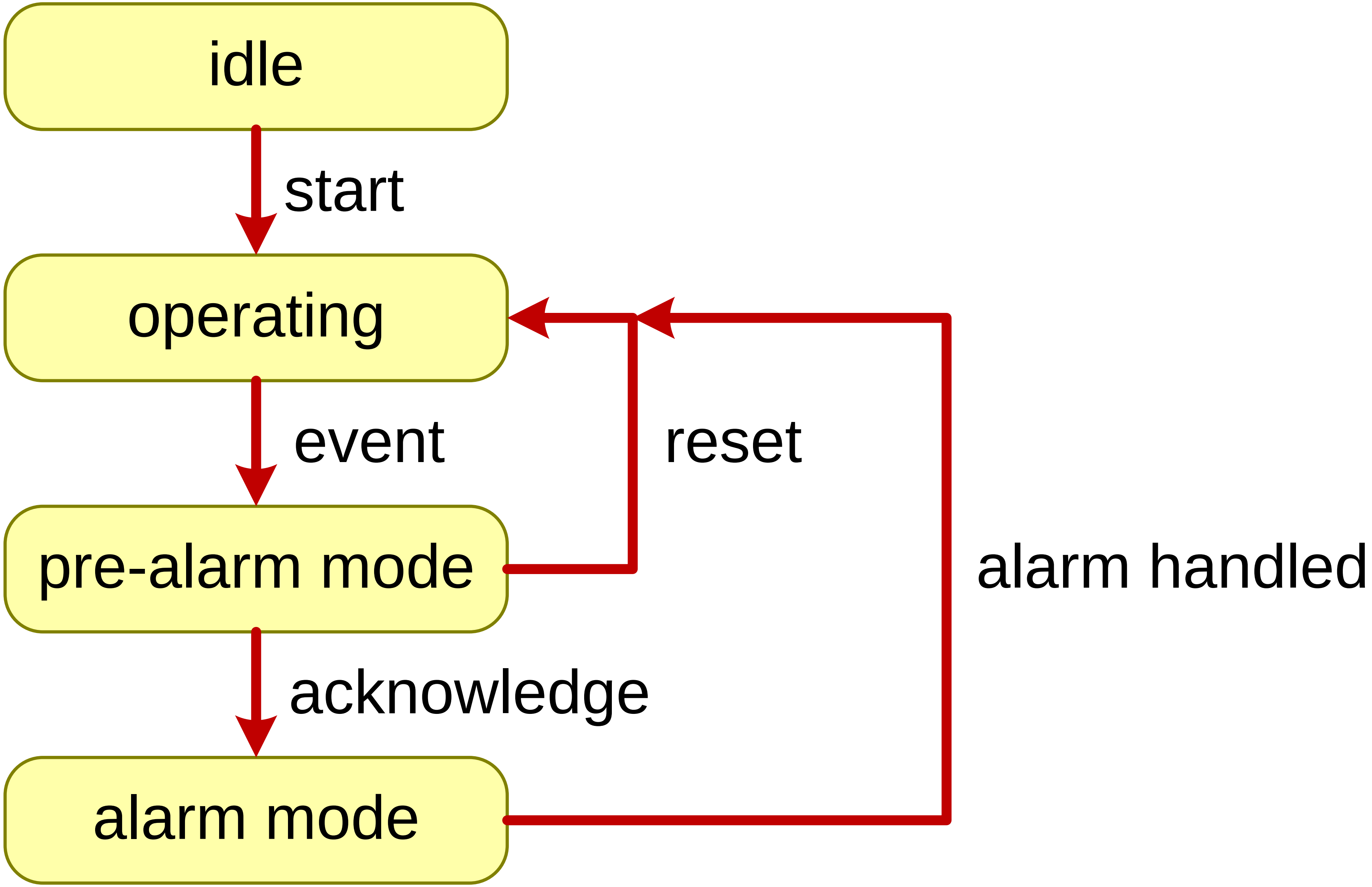


version: 0
March 3, 2024
MRendToEndTimeLine

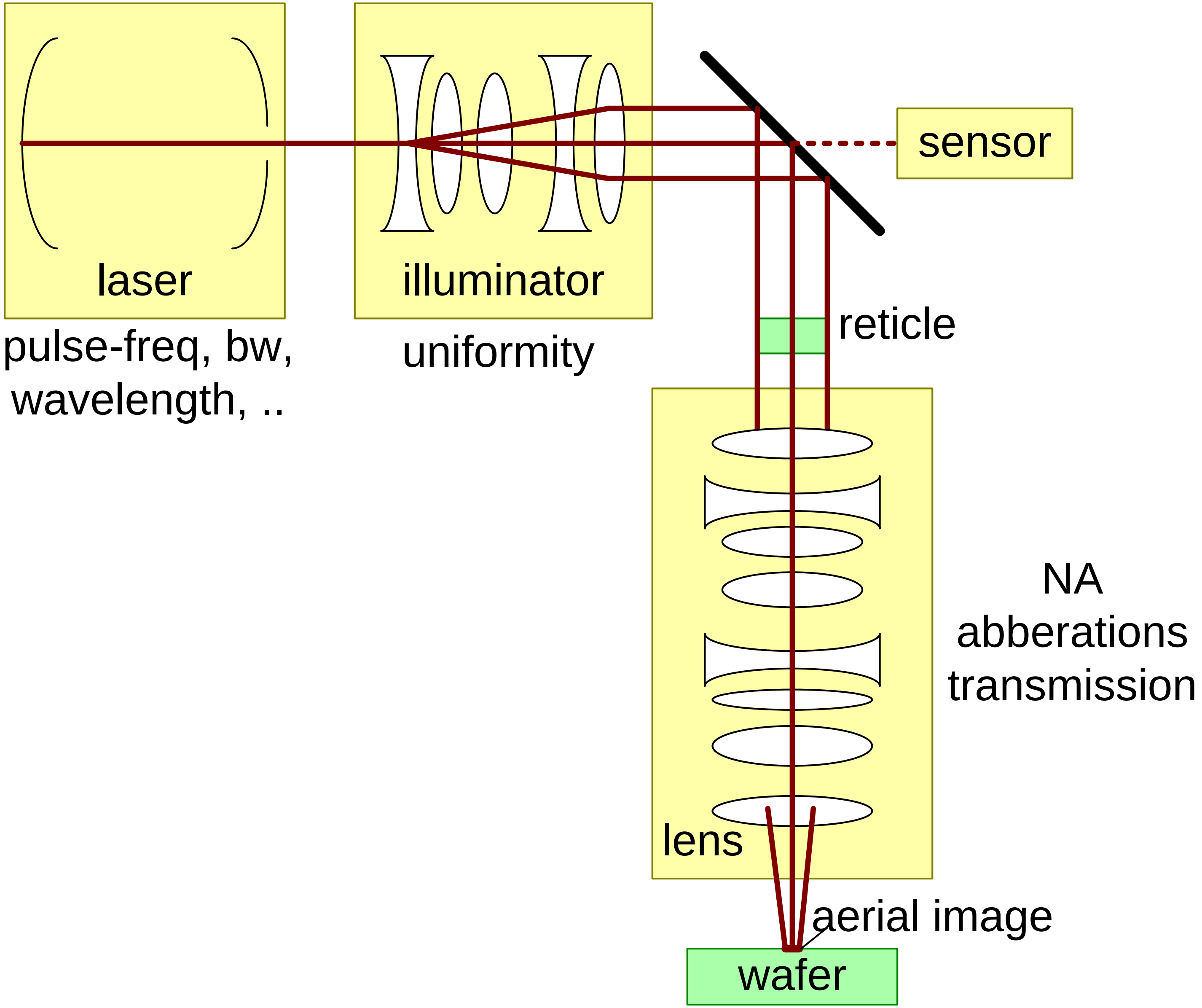
Information Centric Processing Diagram



Example State Diagram



Flow of Light (Physics)



pulse-freq, bw,
wavelength, ..

uniformity

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

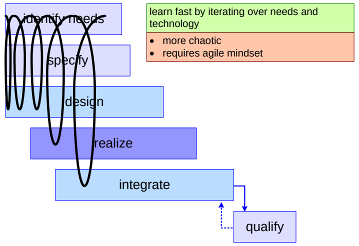
multi-dimensional information and dynamic behavior

Mastering Systems Integration; Early Validation

by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

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.

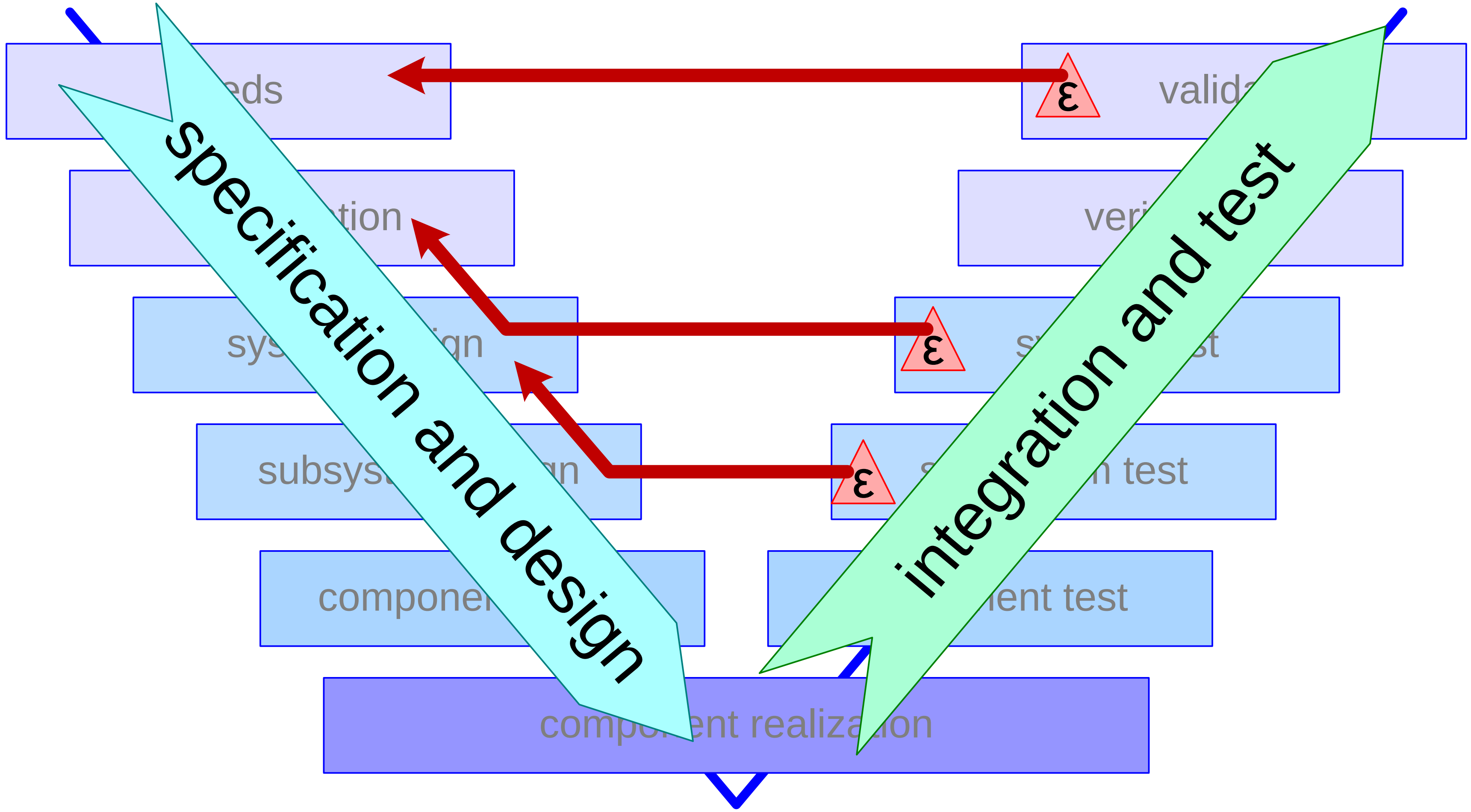


Distribution

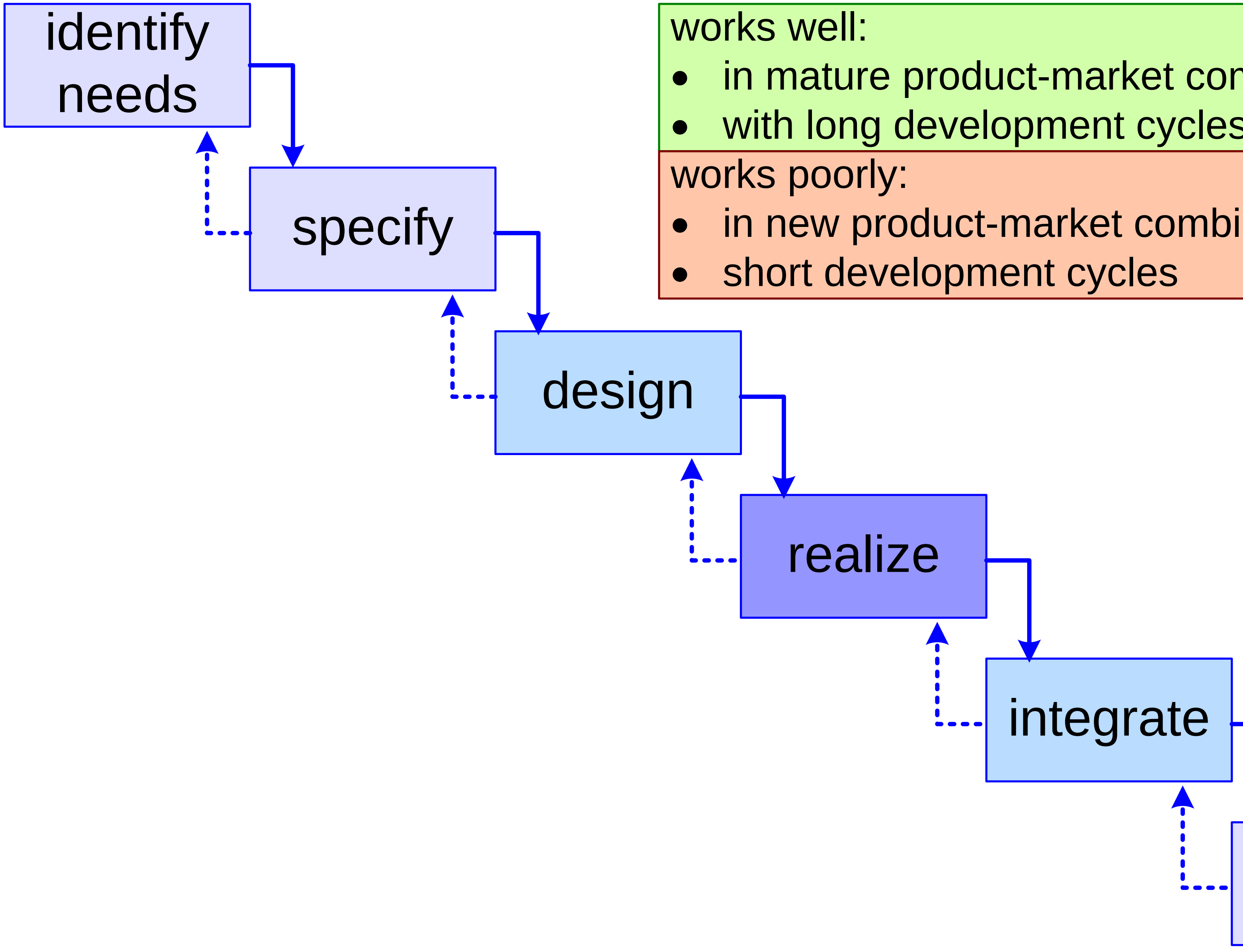
March 3, 2024
status: planned
version: 0.5

Most Problems are Found Late

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



Waterfall model



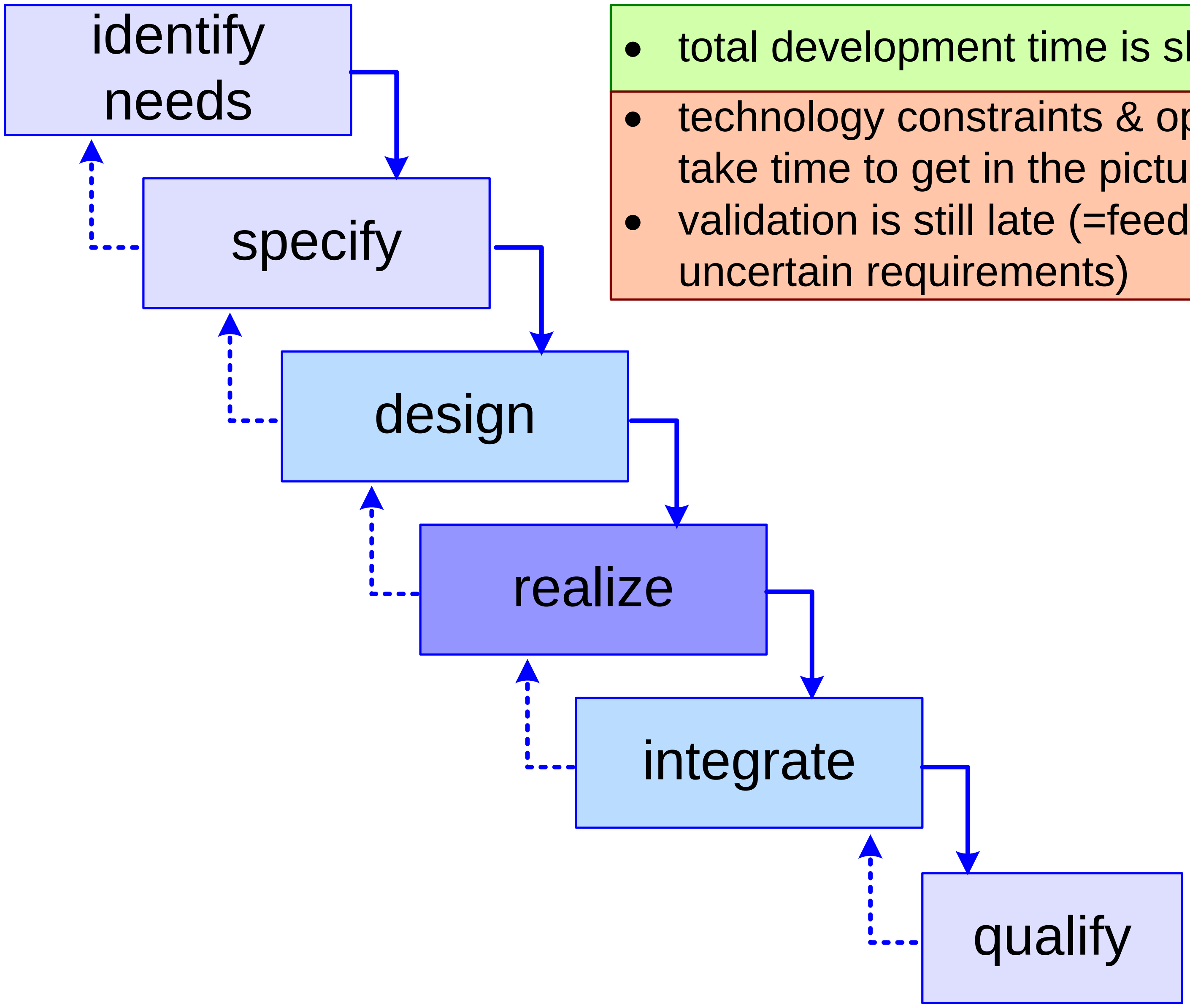
works well:

- in mature product-market combinations
- with long development cycles

works poorly:

- in new product-market combinations
- short development cycles

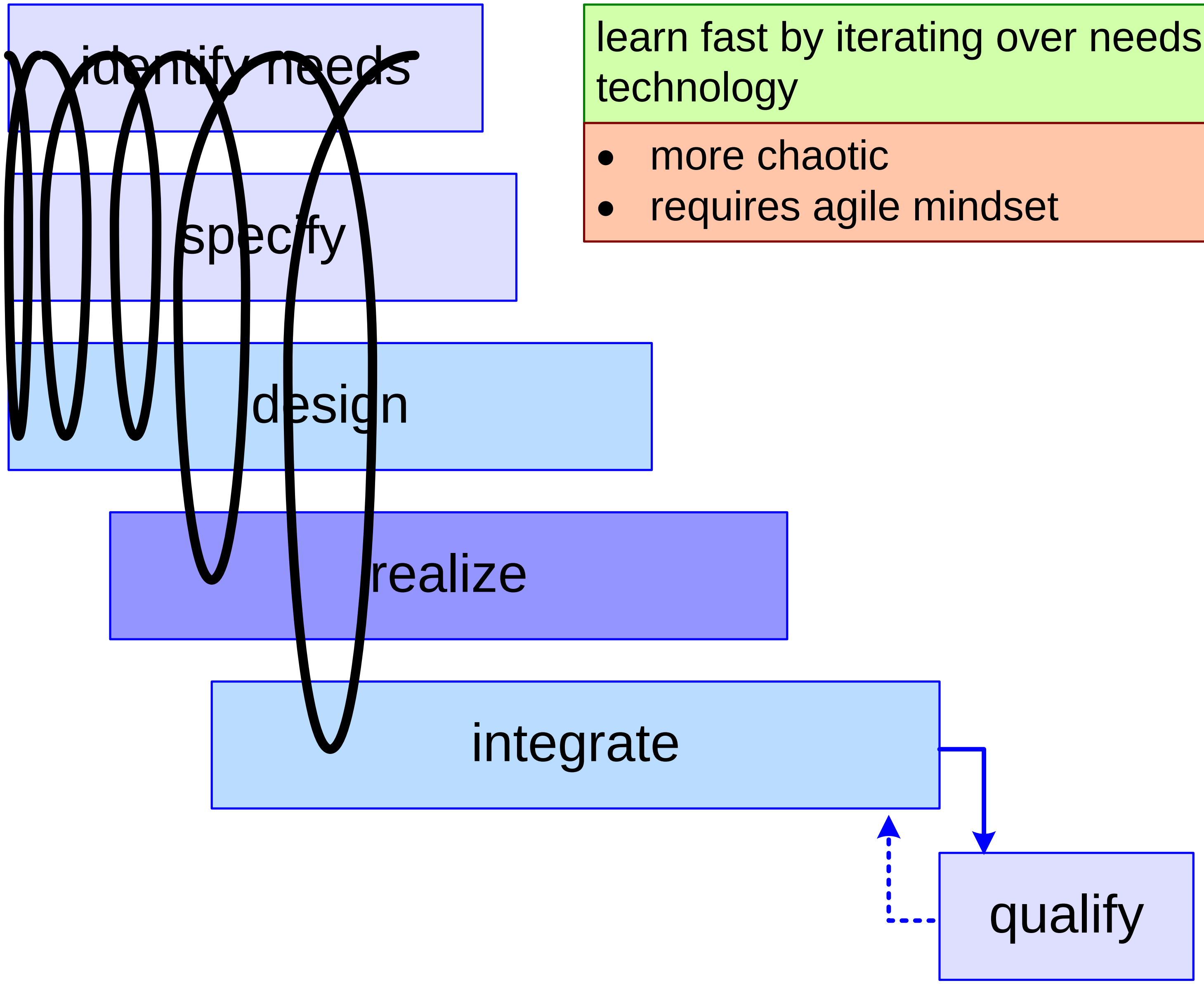
Concurrent Engineering



- total development time is shorter

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

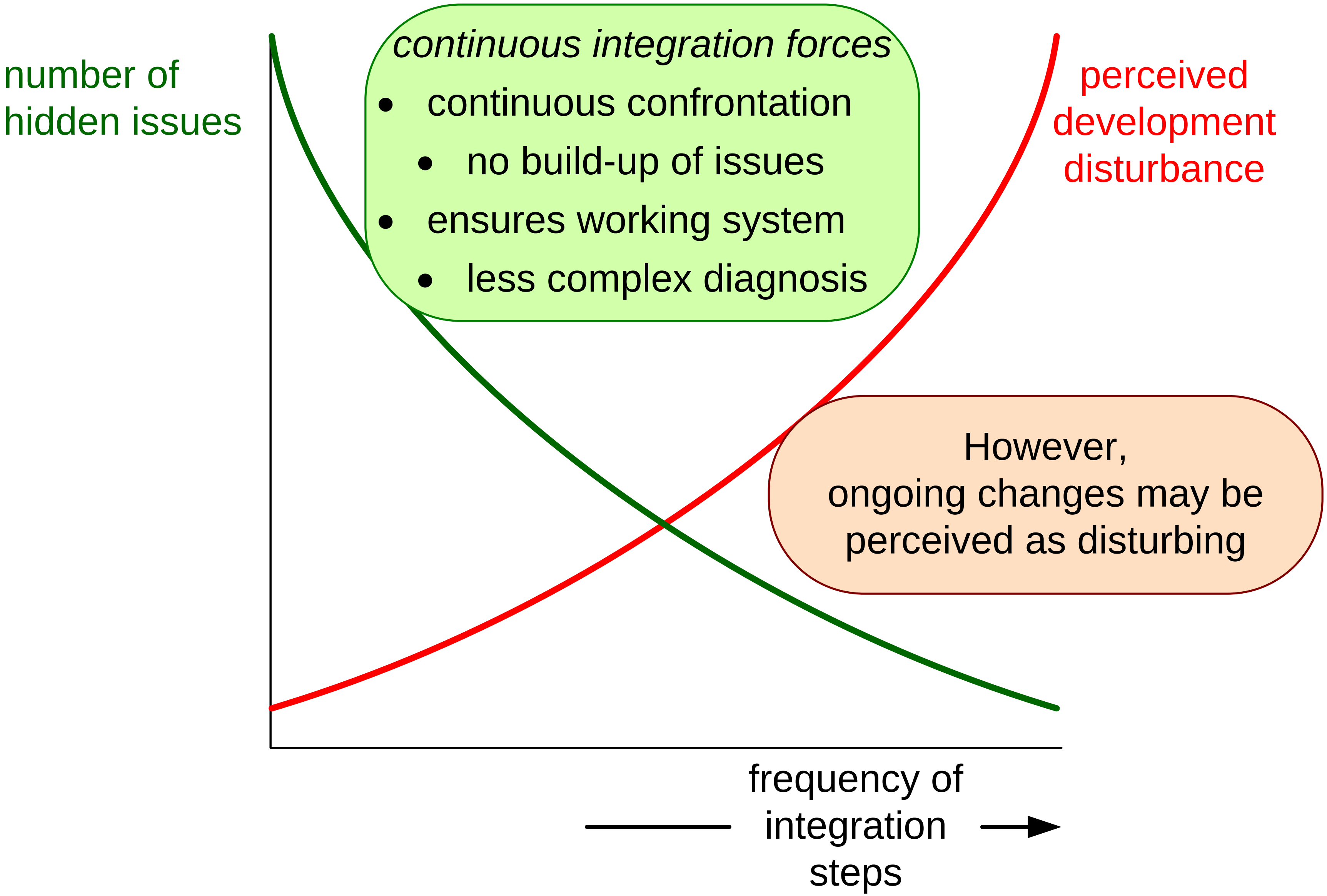
Iterative Approach



learn fast by iterating over needs and technology

- more chaotic
- requires agile mindset

Continuous Integration



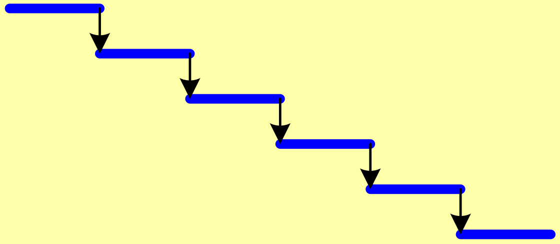
continuous integration forces

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

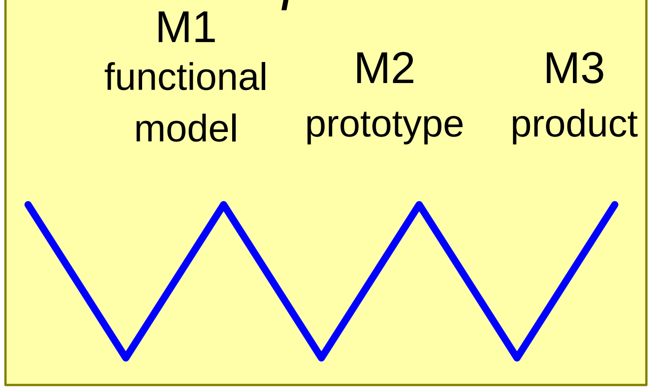
However,
ongoing changes may be
perceived as disturbing

Development Processes From Waterfall to Agile

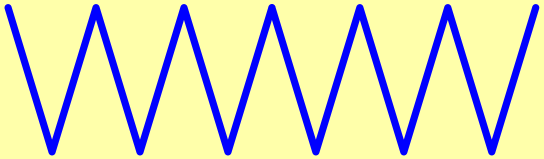
waterfall



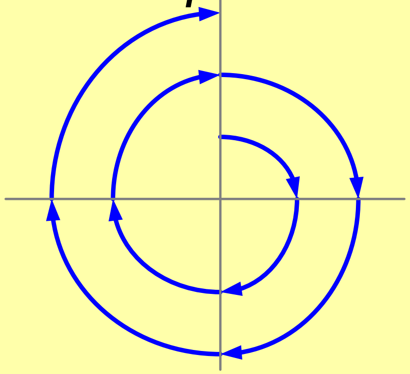
triple-V



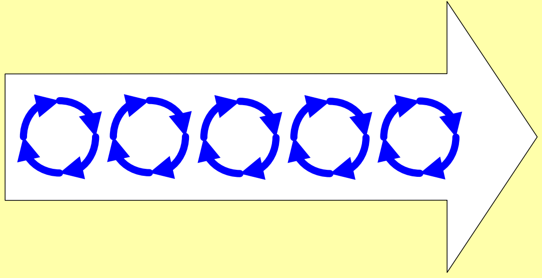
many Vs



spiral



*agile/incremental/
continuous*



*and all kinds of
hybrids*

Mastering Systems Integration; Integration Strategy

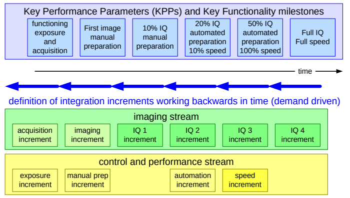
by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

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.

Distribution

March 3, 2024
status: preliminary draft
version: 0.5



Integration Strategy

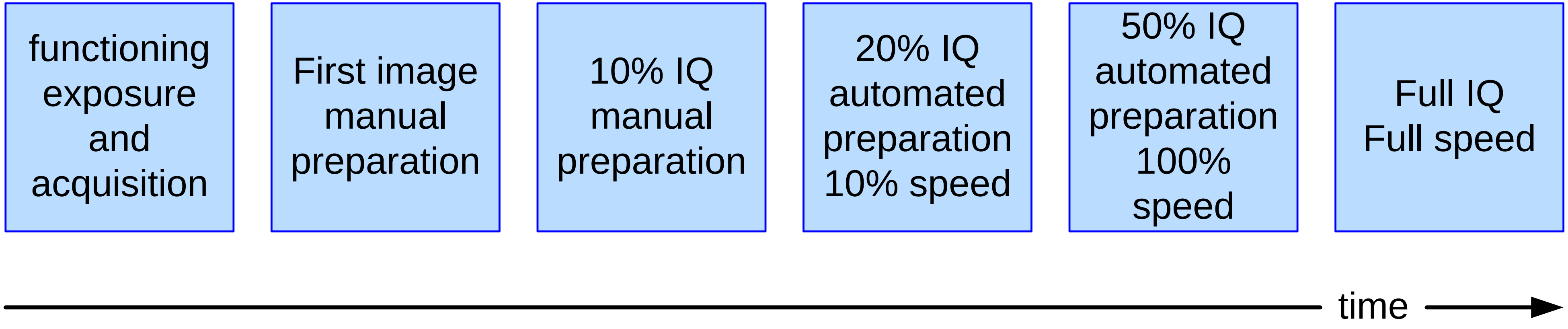
- Get Key Performance Parameters functioning ASAP

- 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

time

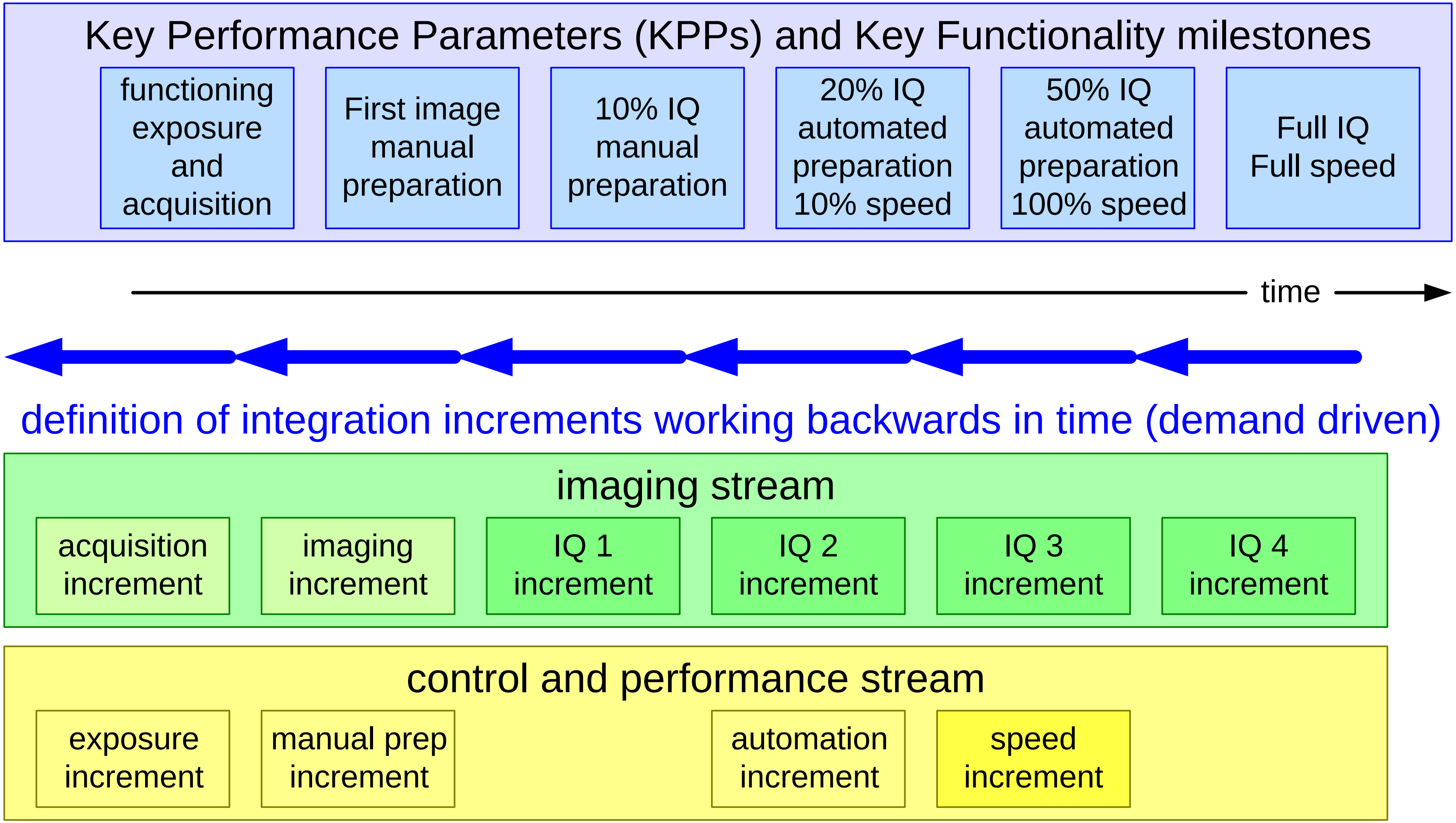
pacing:

maximum 6 month between milestones

depending on technology and domain

version: 0.5
March 3, 2024
MSIPMpacingMilestones

Defining an Integration Sequence in Increments



Key Performance Parameters (KPPs) and Key Functionality 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

time

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

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

Mastering Systems Integration; Integration Environments

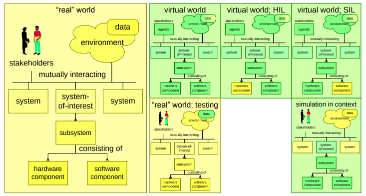
by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

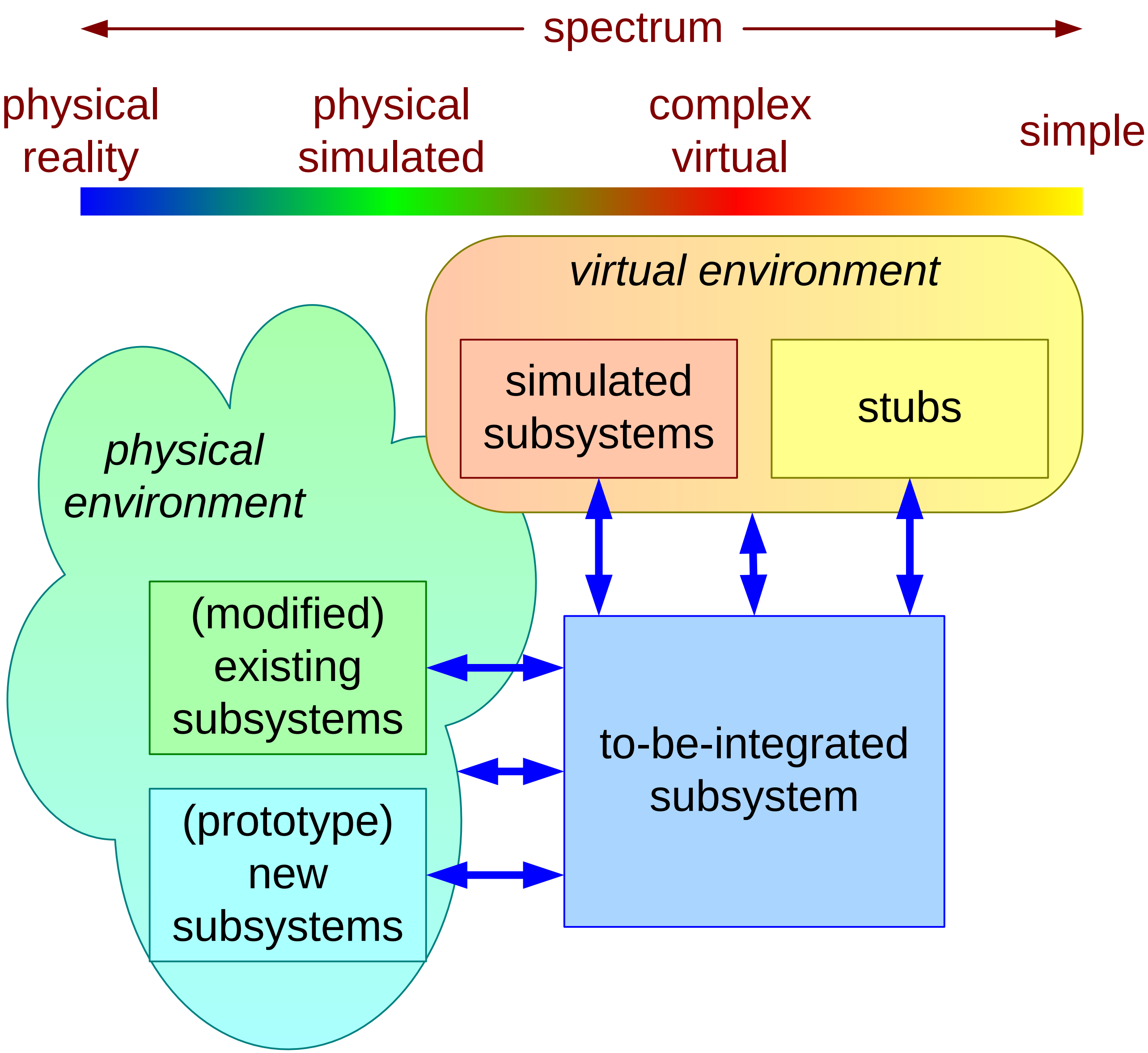
Integration requires an environment that serves as vehicle for the integration. Typically, a wide variation of enviroments supports the integration.

Distribution

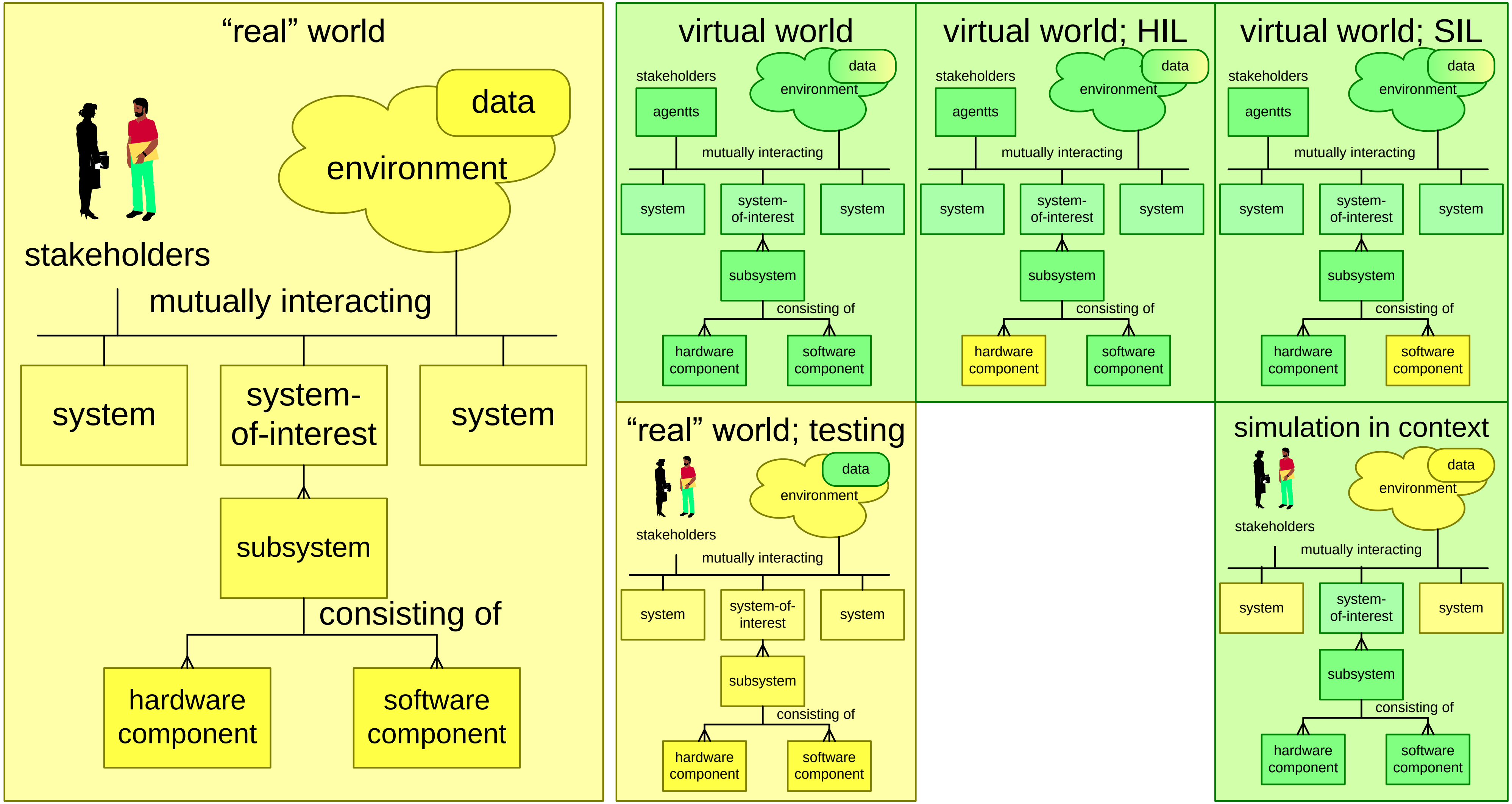
March 3, 2024
status: preliminary draft
version: 0.2



Spectrum of Environments to Support Integration



Spectrum from Real to Virtual Systems



Scope of Test Configuration Management

components $10^4..10^6$

functions emerging from components

specifications and designs

infoware

tools

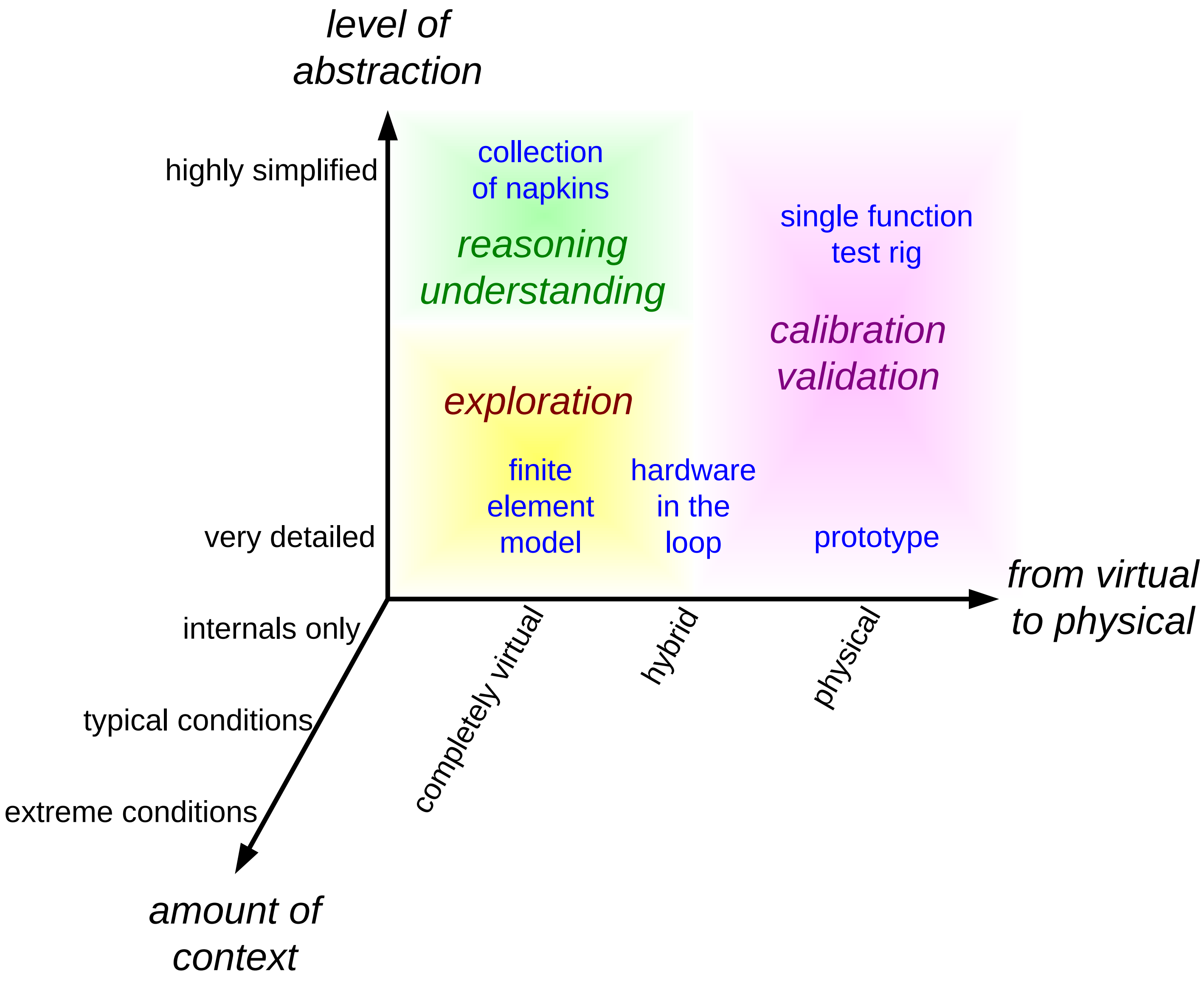
generating, building, manufacturing, configuring, transporting, installing, commissioning, diagnosing, analyzing, teaching, ...

test data and objects

environment physical conditions, physical and information infrastructure

people and their behavior

Modeling Space



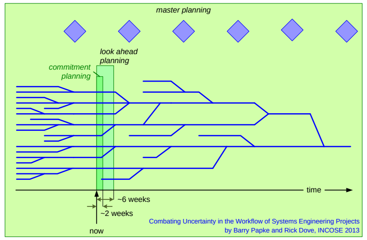
version: 0.2
March 3, 2024
SMIMmodelSpace

Mastering Systems Integration; Project Management

by *Gerrit Muller*   TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

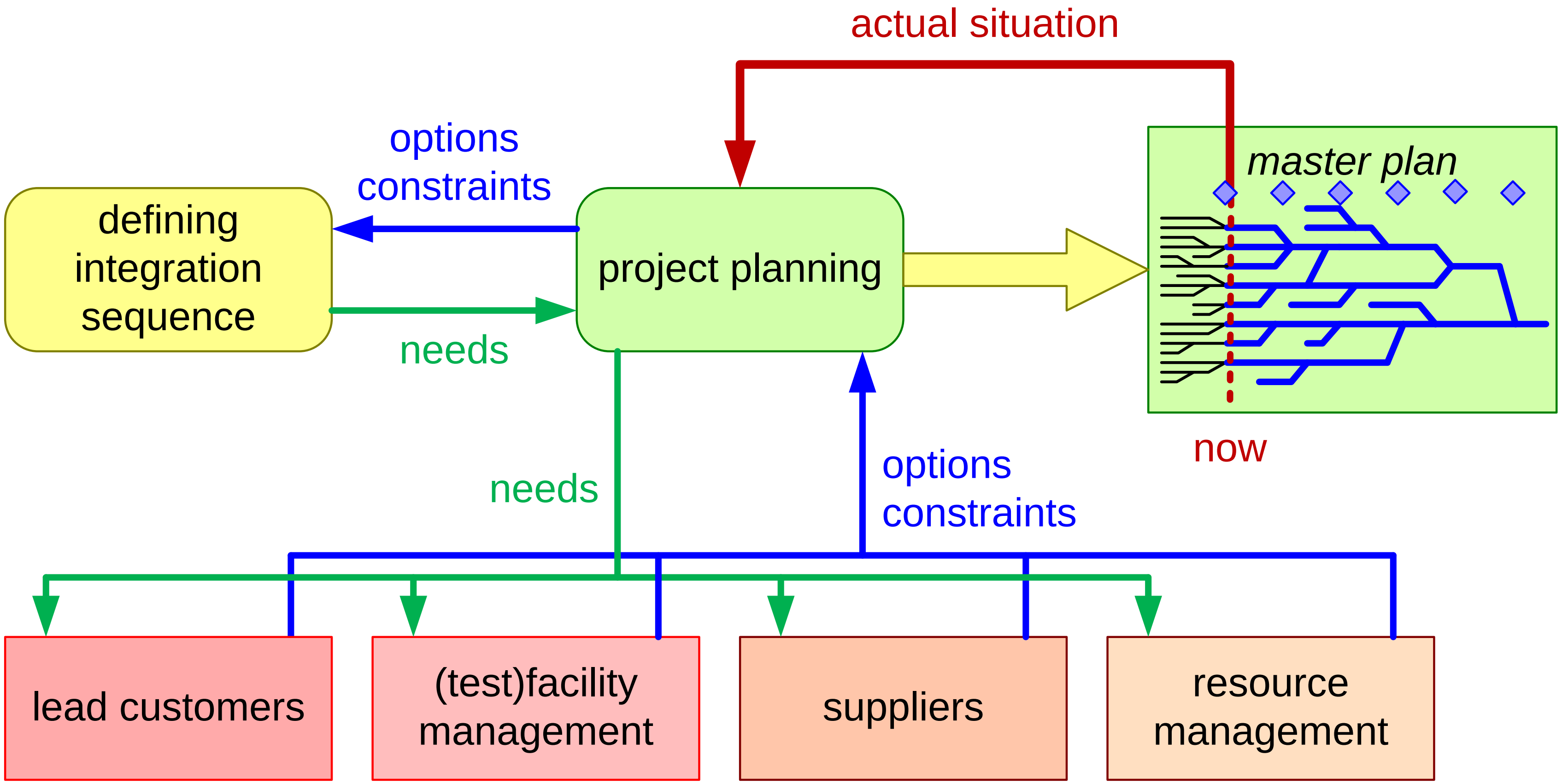
Systems Integration requires specific project management. The challenge for project managers is to plan ahead, knowing that the integration plan will need continuous adaptations.



Distribution

March 3, 2024
status: preliminary draft
version: 0.6

Integration Planning



actual situation

options
constraints

defining
integration
sequence

project planning

master plan

needs

now

needs

options
constraints

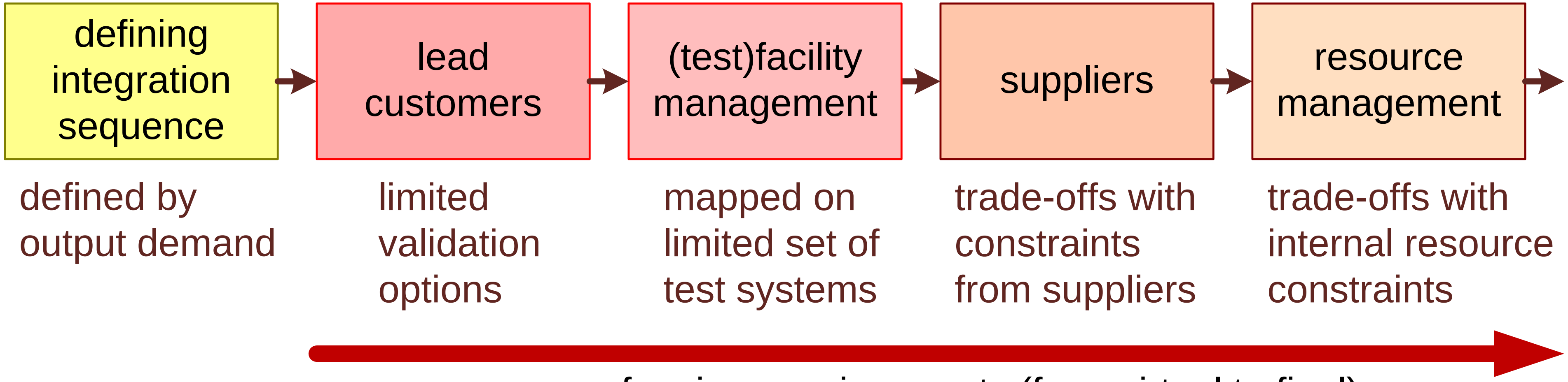
lead customers

(test)facility
management

suppliers

resource
management

Demand Driven, Fitting Constraints



defining integration sequence → lead customers → (test)facility management → suppliers → resource management →

defined by output demand

limited validation options

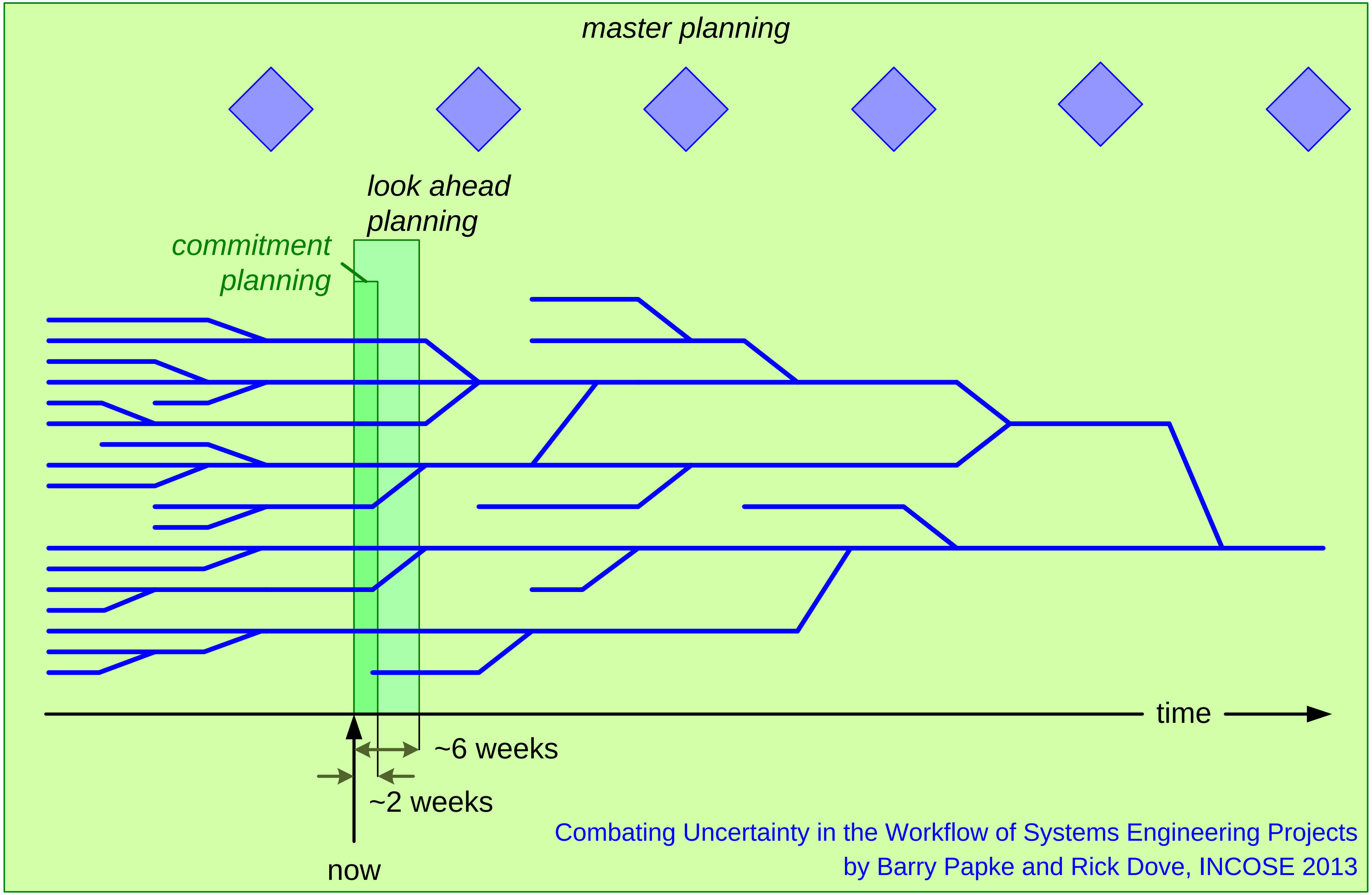
mapped on limited set of test systems

trade-offs with constraints from suppliers

trade-offs with internal resource constraints

use of various environments (from virtual to final)
reordering of integration sequence

Last Planner; Look Ahead!



Combating Uncertainty in the Workflow of Systems Engineering Projects
by Barry Papke and Rick Dove, INCOSE 2013

Mastering Systems Integration; Process and Integration

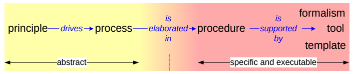
by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

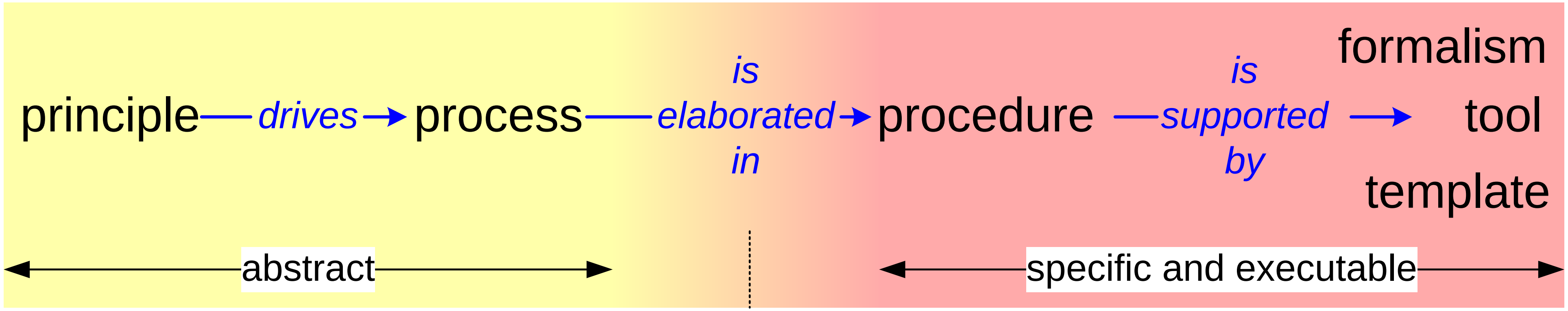
This lesson discusses process aspects of systems integration, such as the organizational capabilities.

Distribution

March 3, 2024
status: planned
version: 0



What is a Process?



principle —drives→ process —is elaborated in→ procedure —is supported by→ formalism / tool / template

abstract

specific and executable

Mastering Systems Integration; Organization

by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

This presentation discusses organizational aspects, such as roles of people, of Systems Integration.

Distribution

March 3, 2024
status: planned
version: 0

logo
TBD

Roles in Systems Integration

project leader

organization
resources
schedule
budget

systems architect/
engineer/integrator

system requirements
design inputs
test specification
schedule rationale
troubleshooting
participate in test

system tester

test
troubleshooting
report

logistics and
administrative support

configuration
orders
administration

engineers

design
component test
troubleshooting
participate in test

machine owner

maintain test model
support test

Modeling and Analysis: Budgeting

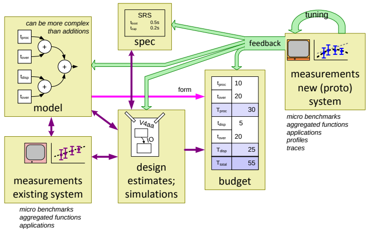
by *Gerrit Muller* TNO-ESI, HSN-NISE
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

This presentation addresses the fundamentals of budgeting: What is a budget, how to create and use a budget, what types of budgets are there. What is the relation with modeling and measuring.

Distribution

March 3, 2024
status: preliminary draft
version: 1.0



Budgeting

content of this presentation

What and why of a budget

How to create a budget (decomposition, granularity, inputs)

How to use a budget

What is a Budget?

A ***budget*** *is*

a ***quantified instantation*** *of a* ***conceptual model***

A ***budget*** *can*

prescribe *or* ***describe*** *the* ***contributions***

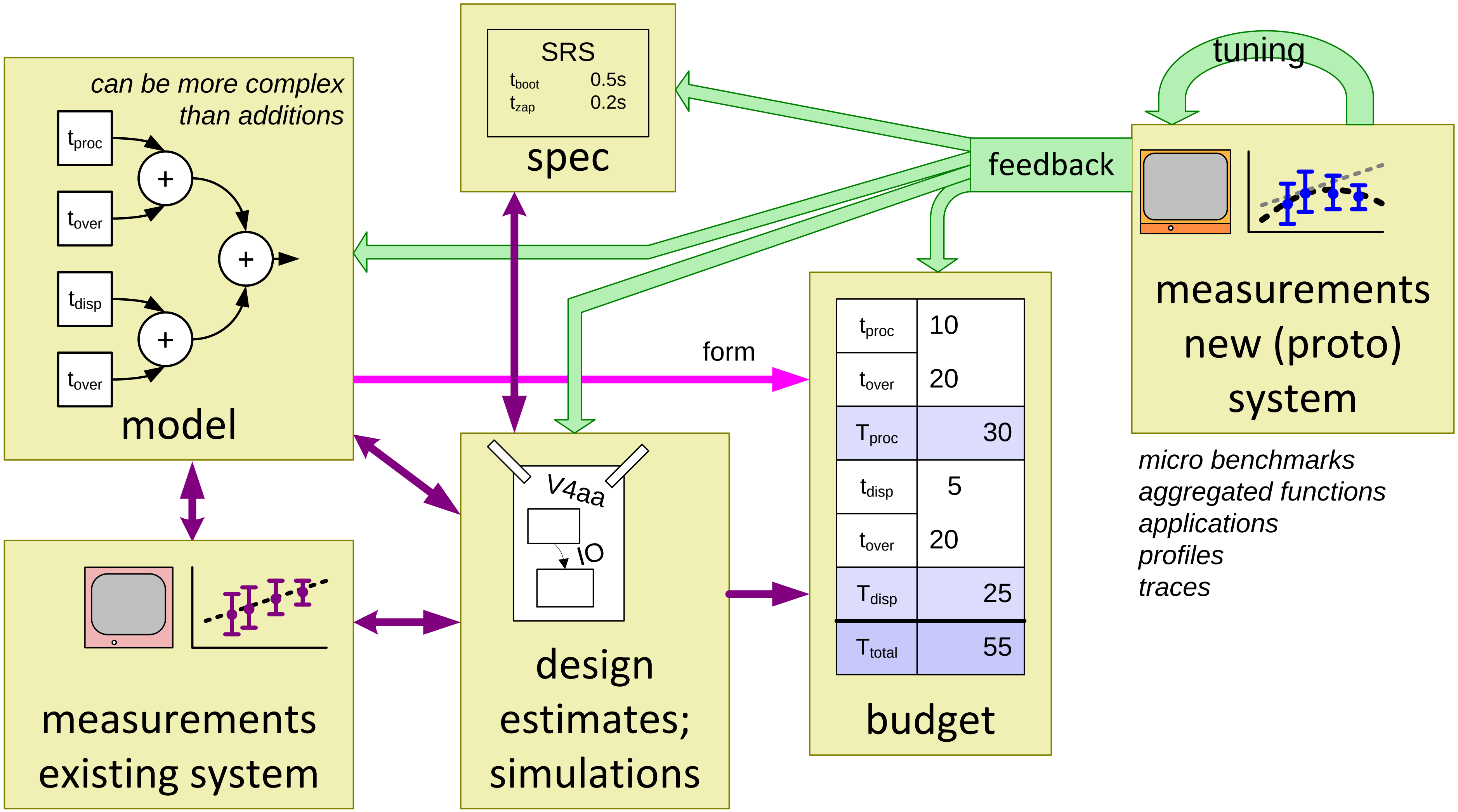
by ***parts*** *of the* ***solution***

to the ***system quality*** *under consideration*

Why Budgets?

- to make the design explicit
- to provide a baseline to take decisions
- to specify the requirements for the detailed designs
- to have guidance during integration
- to provide a baseline for verification
- to manage the design margins explicitly

Visualization of Budget Based Design Flow



t_{proc}	10
t_{over}	20
T_{proc}	30
t_{disp}	5
t_{over}	20
T_{disp}	25
T_{total}	55

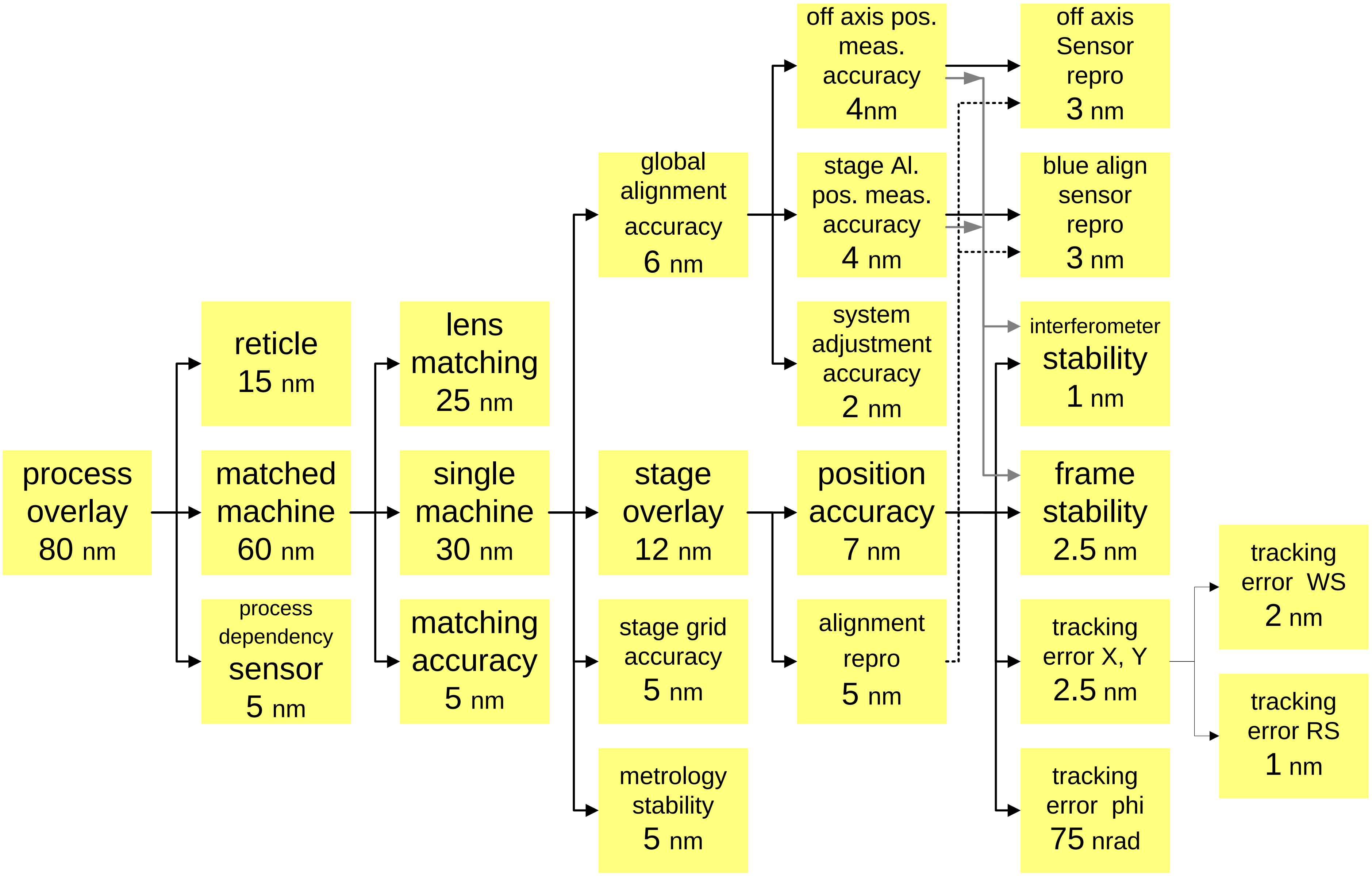
micro benchmarks
aggregated functions
applications
profiles
traces

micro benchmarks
aggregated functions
applications

Stepwise Budget Based Design Flow

step	example
1A measure old systems	micro-benchmarks, aggregated functions, applications
1B model the performance starting with old systems	flow model and analytical model
1C determine requirements for new system	response time or throughput
2 make a design for the new system	explore design space, estimate and simulate
3 make a budget for the new system:	models provide the structure measurements and estimates provide initial numbers specification provides bottom line
4 measure prototypes and new system	micro-benchmarks, aggregated functions, applications profiles, traces
5 Iterate steps 1B to 4	

Budgets Applied on Waferstepper Overlay



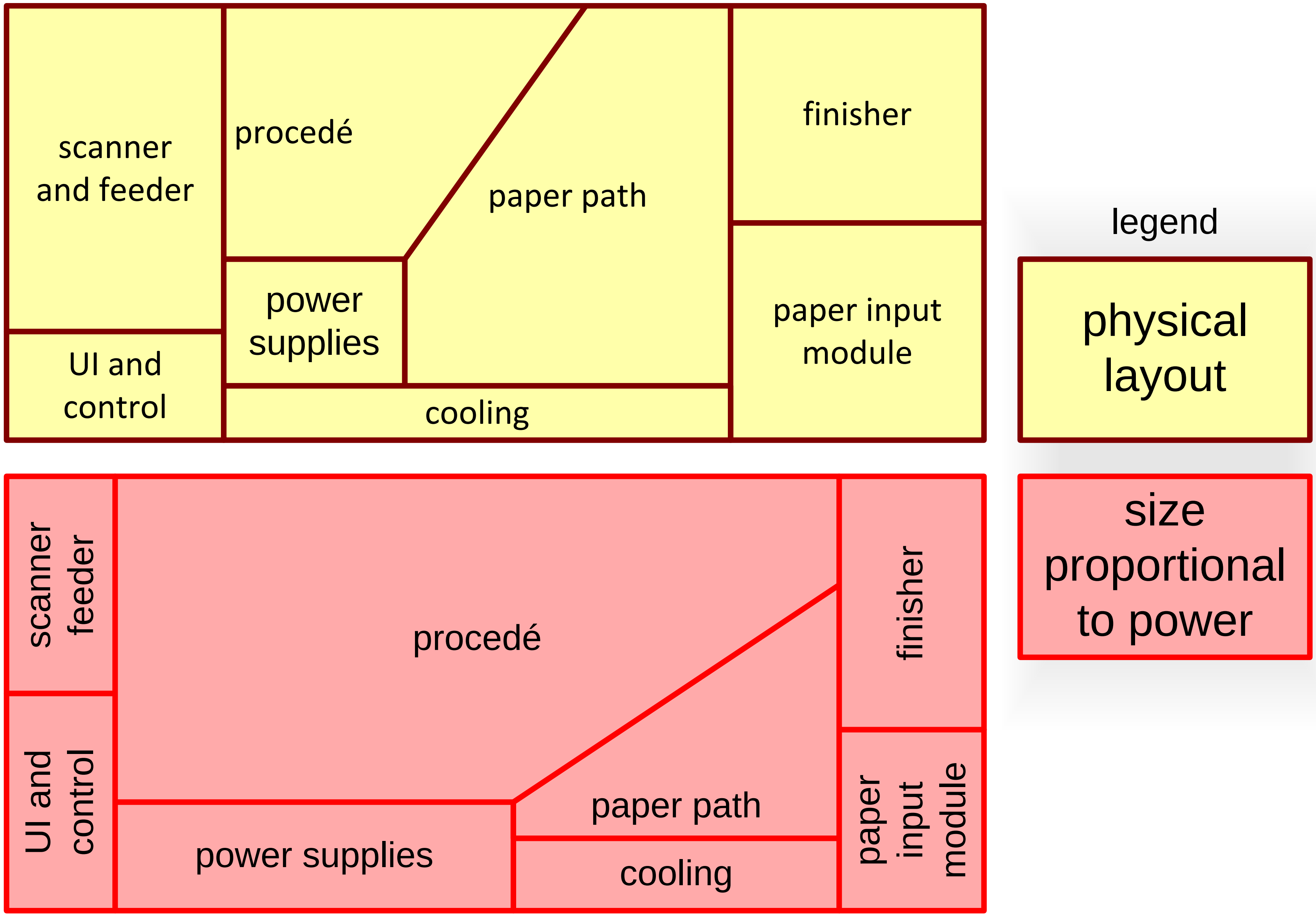
version: 1.0
March 3, 2024
ASMLoverlayBudget

Budgets Applied on Medical Workstation Memory Use

memory budget in Mbytes	code	obj data	bulk data	total
shared code	11.0			11.0
User Interface process	0.3	3.0	12.0	15.3
database server	0.3	3.2	3.0	6.5
print server	0.3	1.2	9.0	10.5
optical storage server	0.3	2.0	1.0	3.3
communication server	0.3	2.0	4.0	6.3
UNIX commands	0.3	0.2	0	0.5
compute server	0.3	0.5	6.0	6.8
system monitor	0.3	0.5	0	0.8
application SW total	13.4	12.6	35.0	61.0
UNIX Solaris 2.x				10.0
file cache				3.0
total				74.0

version: 1.0
March 3, 2024
RVmemoryBudgetTable

Power Budget Visualization for Document Handler



scanner and feeder

procedé

paper path

finisher

UI and control

power supplies

cooling

paper input module

legend

physical layout

scanner feeder

procedé

finisher

UI and control

paper path

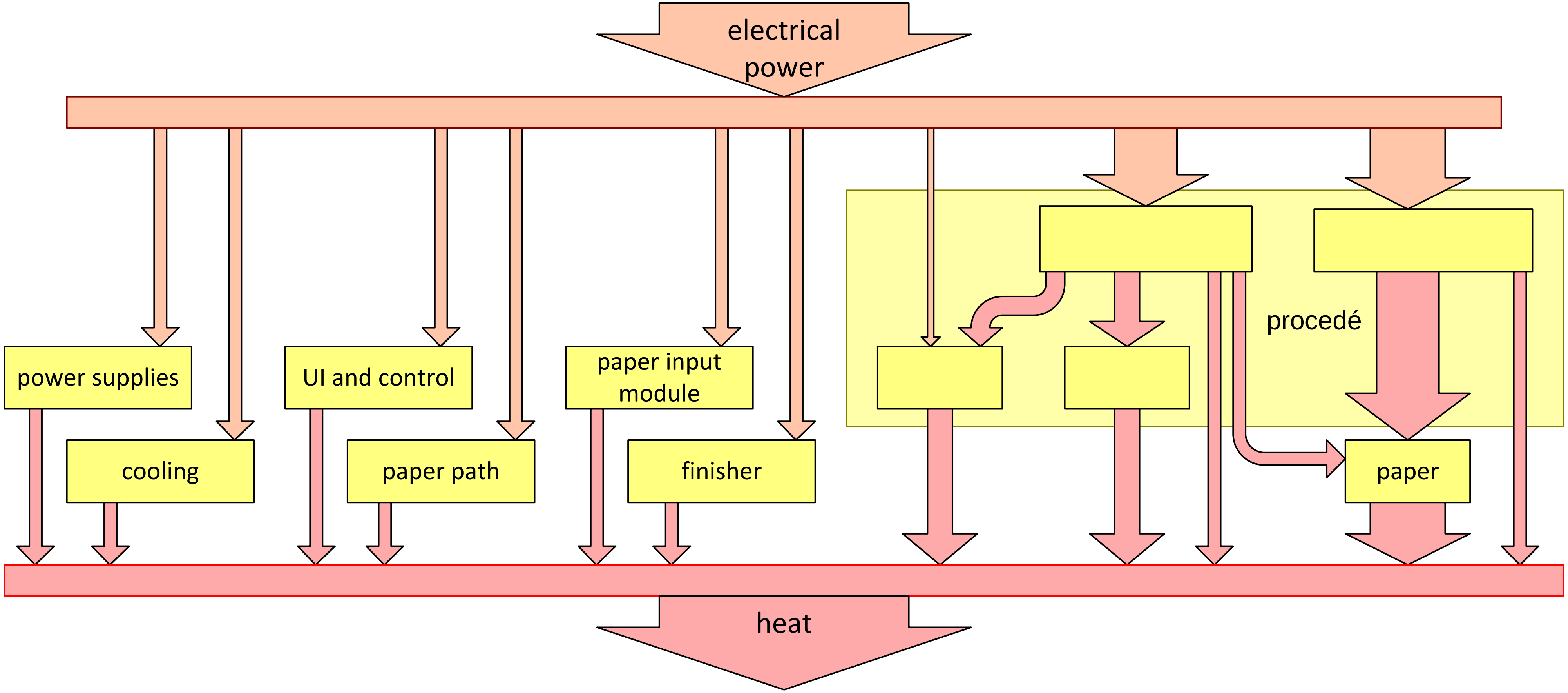
power supplies

cooling

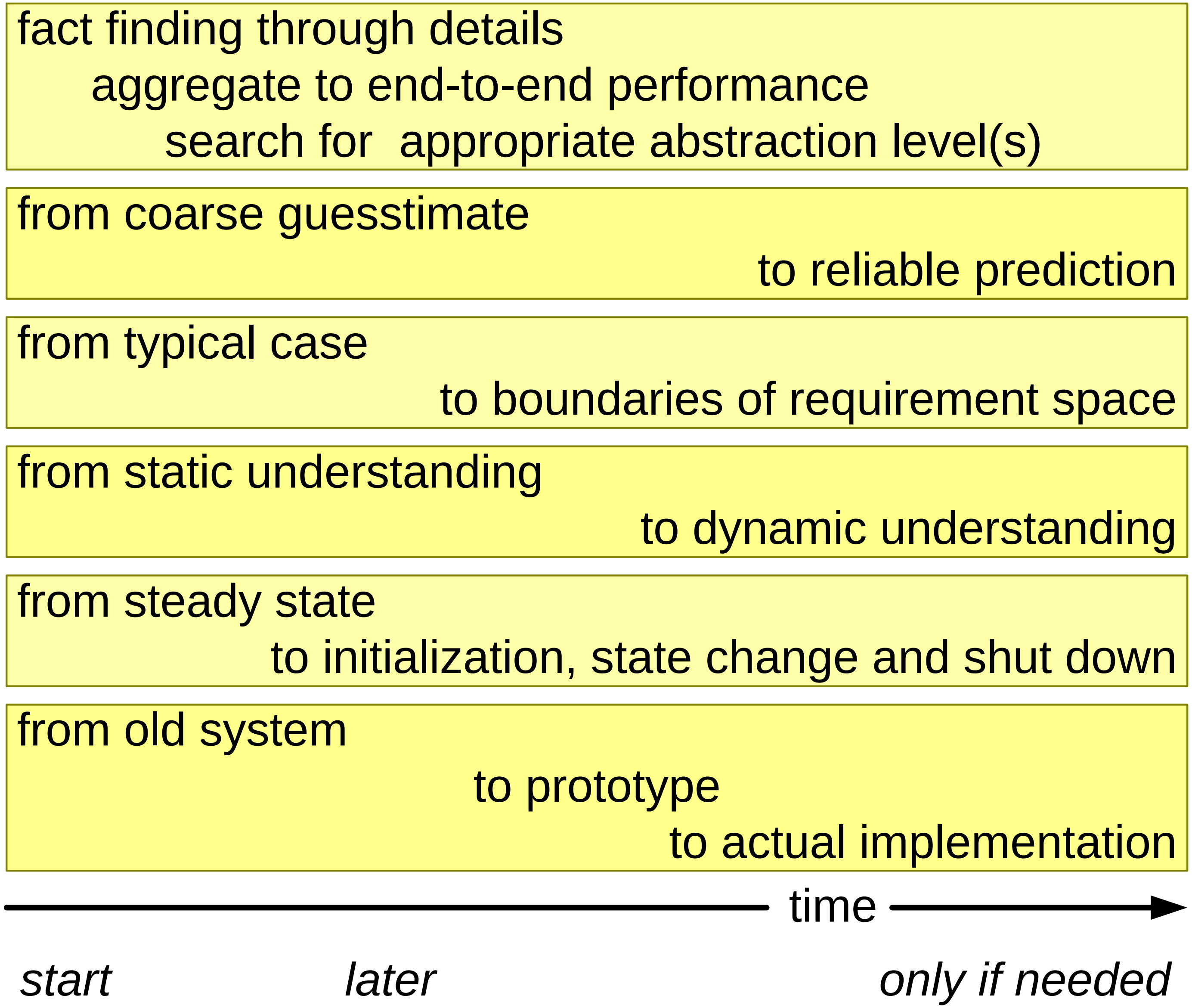
paper input module

size proportional to power

Alternative Power Visualization



Evolution of Budget over Time



fact finding through details
aggregate to end-to-end performance
search for appropriate abstraction level(s)

from coarse guesstimate
to reliable prediction

from typical case
to boundaries of requirement space

from static understanding
to dynamic understanding

from steady state
to initialization, state change and shut down

from old system
to prototype
to actual implementation

Potential Applications of Budget based design

- resource use (CPU, memory, disk, bus, network)
- timing (response, latency, start up, shutdown)
- productivity (throughput, reliability)
- Image Quality parameters (contrast, SNR, deformation, overlay, DOF)
- cost, space, time

What kind of budget is required?

static	dynamic
typical case	worst case
global	detailed
approximate	accurate

is the budget based on
wish, empirical data, extrapolation,
educated guess, or expectation?

Summary of Budgeting

A budget is a quantified instantiation of a model

A budget can prescribe or describe the contributions by parts of the solution to the system quality under consideration

A budget uses a decomposition in tens of elements

The numbers are based on historic data, user needs, first principles and measurements

Budgets are based on models and estimations

Budget visualization is critical for communication

Budgeting requires an incremental process

Many types of budgets can be made; start simple!

Colophon

The Boderc project contributed to Budget Based Design. Especially the work of

Hennie Freriks, Peter van den Bosch (Océ),

Heico Sandee and *Maurice Heemels* (TU/e, ESI)

has been valuable.

Mastering Systems Integration; Testing

by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

During integration, the integrators continuously test parts, functions, and systems. Testing requries the creation of an experimental set-up, where the test enviroment offers stimuli and measures responses. This lesson discusses some of the testing methods and considerations.

Distribution

March 3, 2024
status: planned
version: 0.1

logo
TBD

Why Testing?

Objectives of testing during integration:

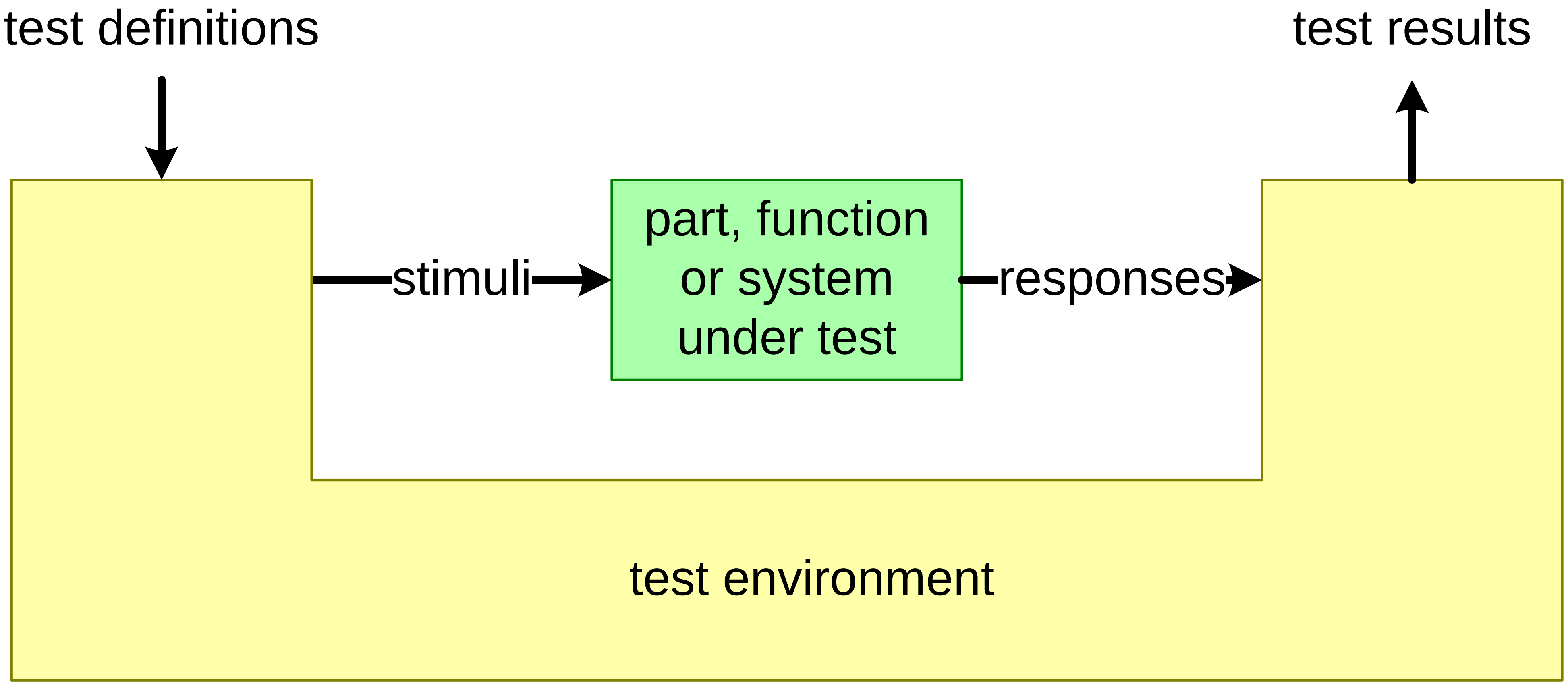
- to find potential quality attribute and behavior problems at specification and design level as early as possible.
- to learn as much as possible about the emerging quality attributes and behaviors.

Consequences for testing:

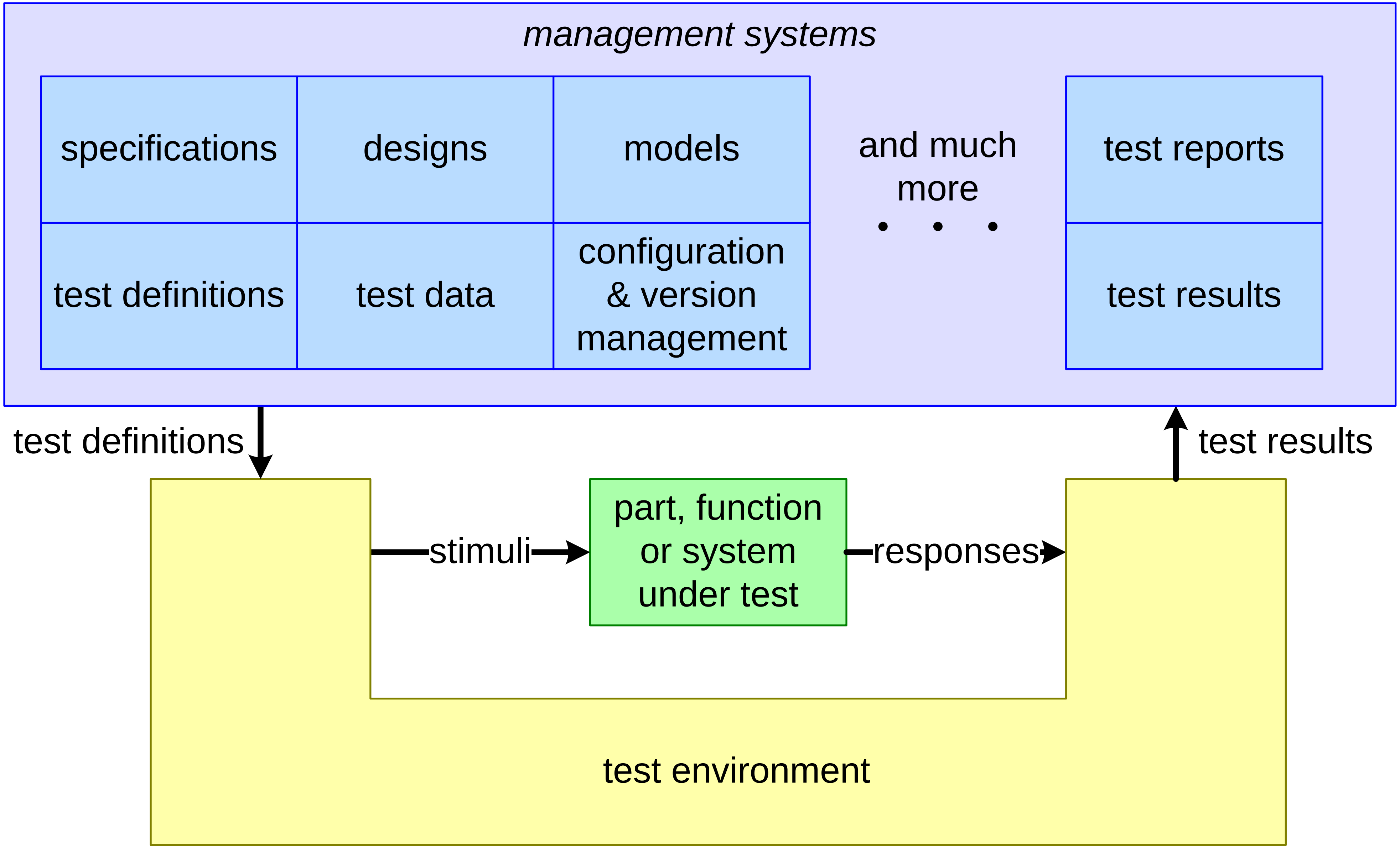
- stimulate the object under test externally and internally (insertion)
- observe the system externally (specification) and internally (design)

version: 0.1
March 3, 2024
MSITEobjectives

Testing Environment



Testing Environment Management Systems Context



management systems

specifications

designs

models

and much more . . .

test reports

test definitions

test data

configuration & version management

test results

test definitions

test results

stimuli

part, function or system under test

responses

test environment

Accelerated Testing

During normal use, stimuli are periodic, with frequencies f_0, f_1, f_2, etc.

During accelarated testing these frequencies are increased.

- **ALT** (Accelerated Life Testing) is **Test-to-Pass** (showing how long the system can operate)
- **HALT** (Highly Accelerated Life Testing) is **Test-to-Fail** (learning weaknesses and margins)

The concepts are applicable in hardware, software, and systems. However, engineers know the stimuli for hardware better (temperature, humidity, vibrations, etc.).

version: 0.1
March 3, 2024
MSITEacceleration

Mastering Systems Integration; Readiness Levels

by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

Readiness level models offer a yardstick to assess the status of specific project aspects. Examples are technology readiness and integration readiness.

March 3, 2024
status: planned
version: 0

logo
TBD

Technology Readiness Levels

TRL 9	actual system proven in operational environment
TRL 8	system complete and qualified
TRL 7	system prototype demonstration in operational environment
TRL 6	technology demonstrated in relevant environment
TRL 5	technology validated in relevant environment
TRL 4	technology validated in lab
TRL 3	experimental proof of concept
TRL 2	technology concept formulated
TRL 1	basic principles observed

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

Integration Readiness Levels

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

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

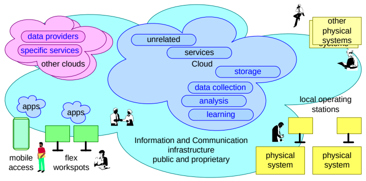
Mastering Systems Integration; Systems of Systems

by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

Most end-user functionality and services are realized by Systems of Systems. Many of these systems may include organizations and humans; the systems aren't technical artifacts anymore. These systems evolve over time individually and typically lack a centralized governance. The resulting end-to-end qualities depend on all consituent systems and their interoperability.

March 3, 2024
status: preliminary draft
version: 0.2



Types of Systems of Systems

Directed - The SoS is centrally managed

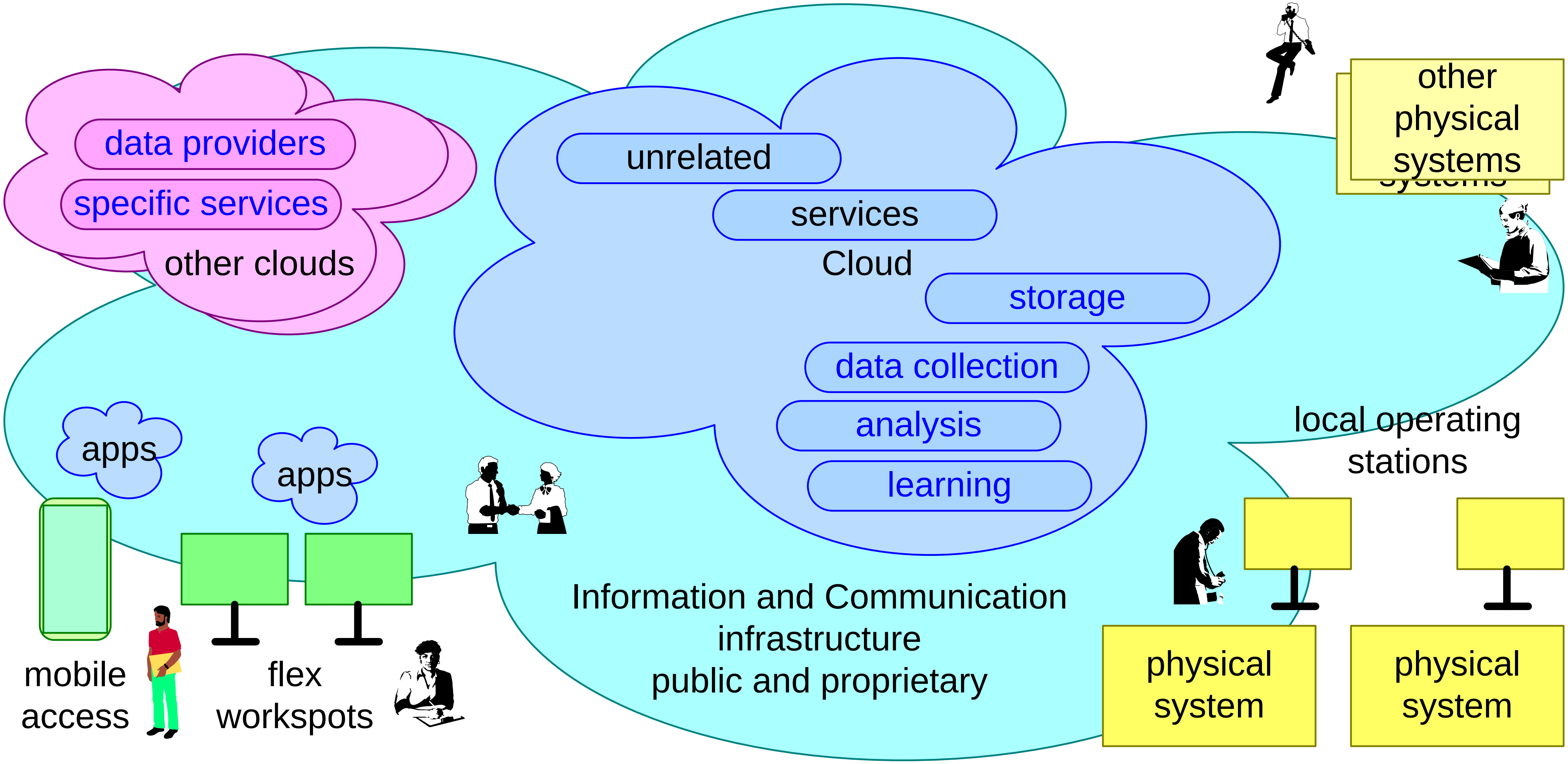
Acknowledged - The SoS has recognized objectives, and active cooperation between SoS and constituent systems

Collaborative - The constituent systems and stakeholders cooperate

Virtual - The SoS nature more or less emerge from the constituent systems

J. Dahmann and K. Baldwin. 2008. "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering." IEEE Systems Conference 2008 in Montreal, 2008

Where are the System Boundaries?



data providers

specific services

other clouds

unrelated

services

Cloud

storage

data collection

analysis

learning

other physical systems

local operating stations

apps

apps

mobile access

flex workspots

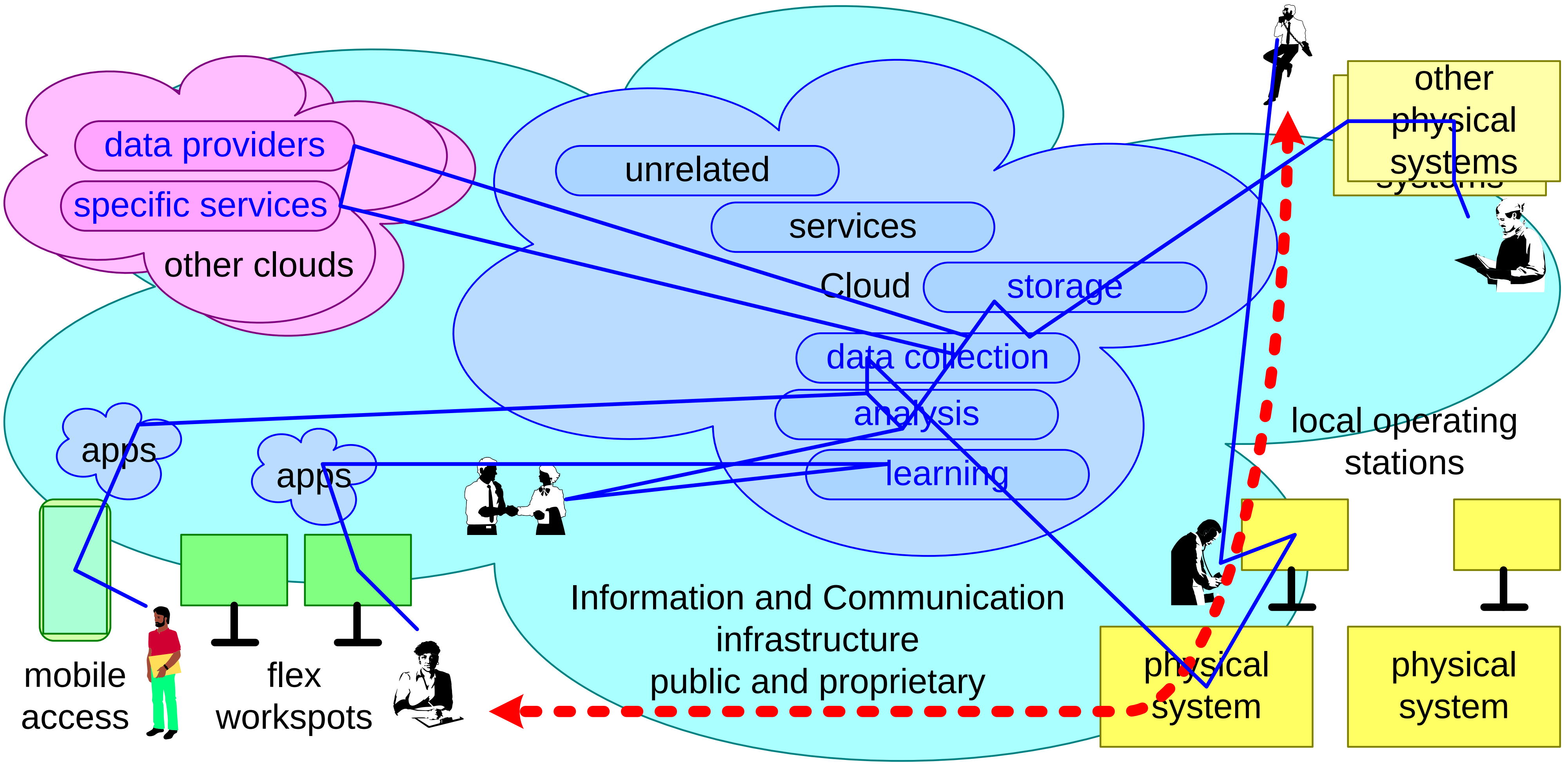
Information and Communication infrastructure public and proprietary

physical system

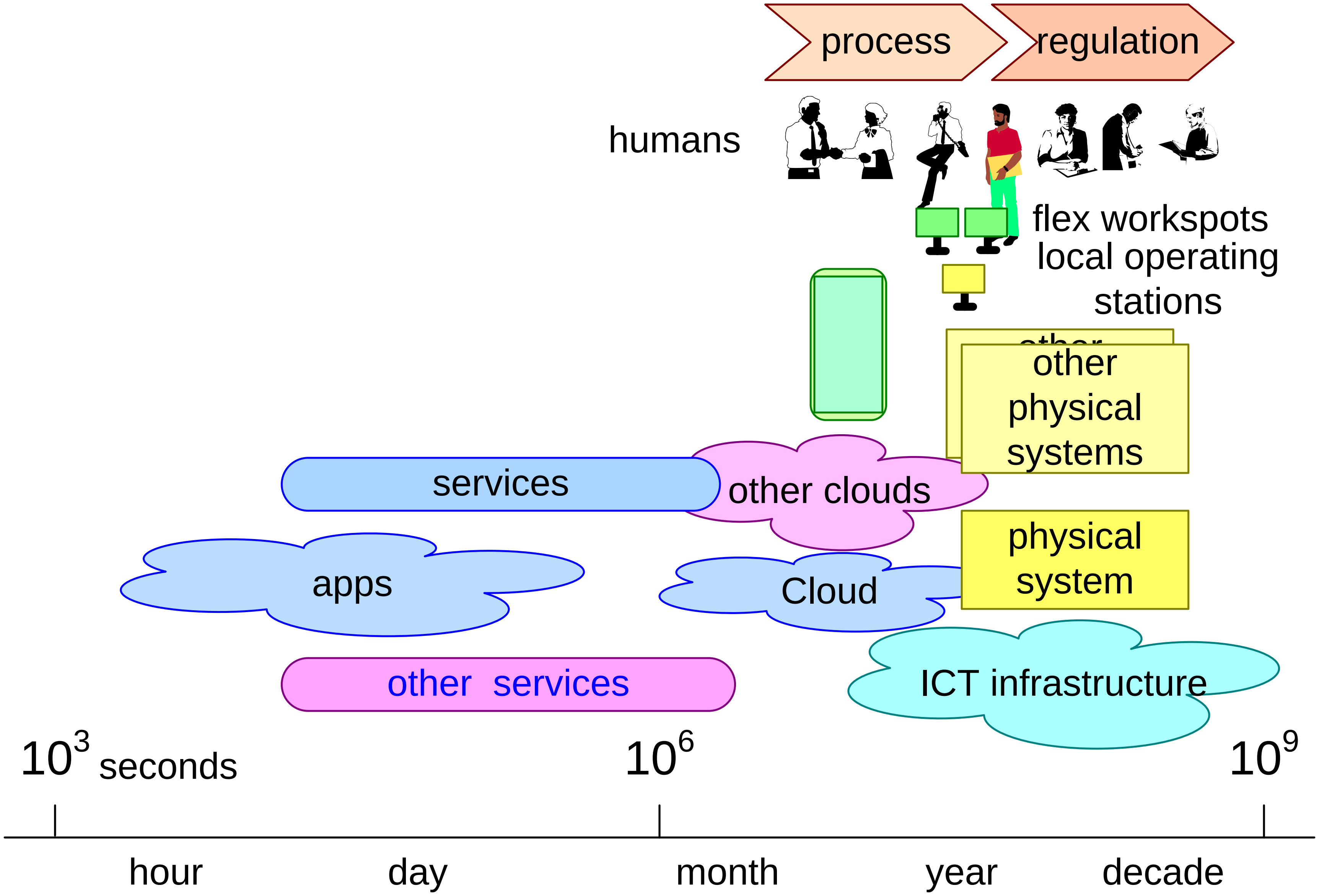
physical system

version: 0.2
March 3, 2024
MSISSboundaries

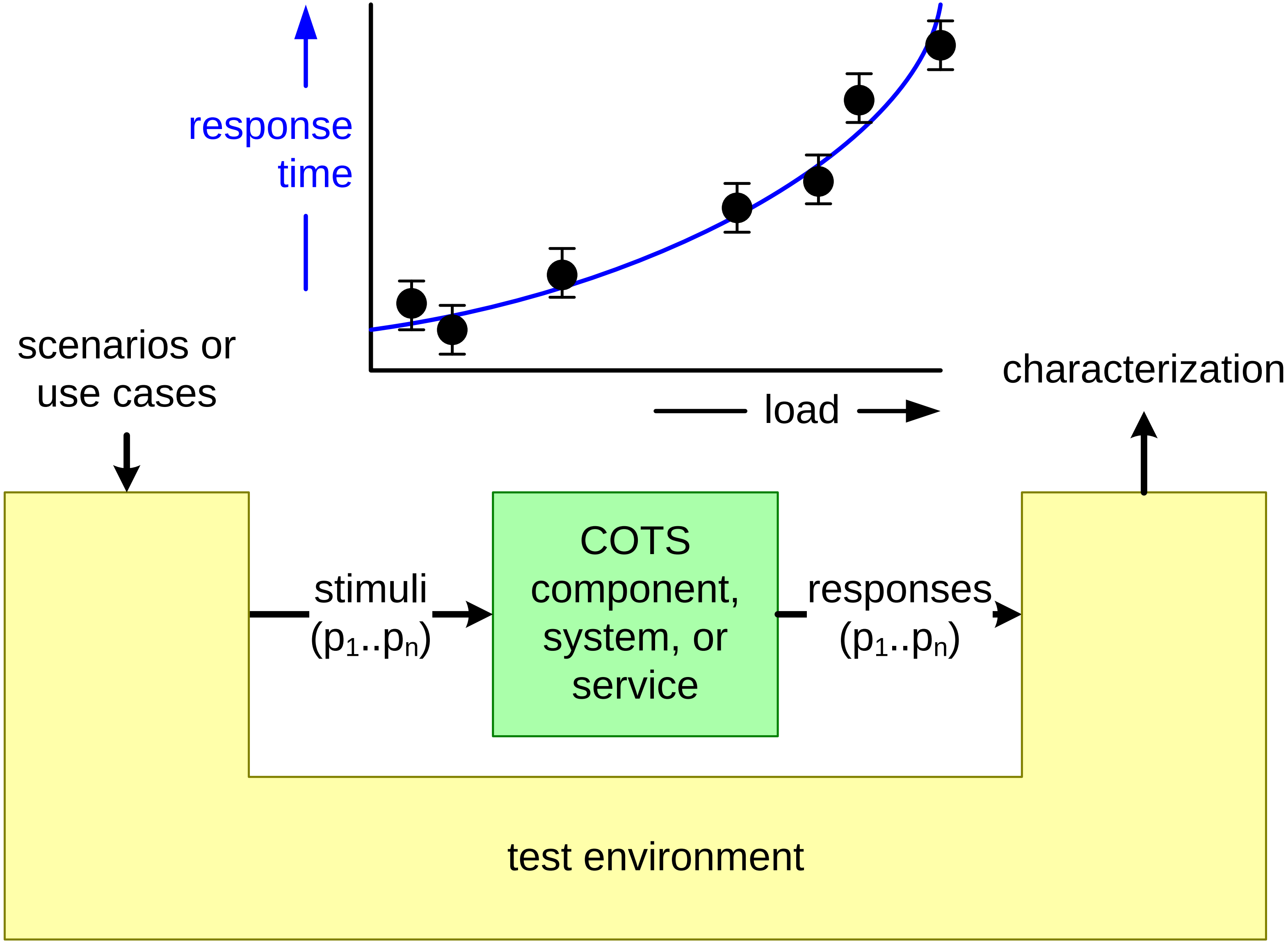
End-to-End Function



Varying Dynamics



Characterization of Black Box Parts



Summary

- Systems of Systems Integration **continues in the field** during operation
- **Ownership** and **responsibility** for end-to-end performance is **ill-defined**
- **Your system** may be **blamed** for problems with a **root cause elsewhere**
- End-to-end performance depends on a mix of
  - traditional **technical** systems
  - **modern technologies** like learning
  - **humans** in their organizational and societal context (psychological, social, political, economical, legal, etc.)
  - the **physical** context (location, climate, etc.) and laws of physics

Keywords from various SoS models in literature

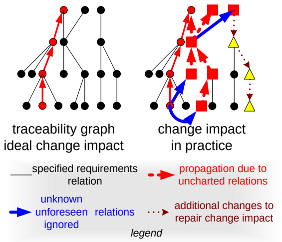
Boardman and Sauser	Maier	DeLaurentis	Dahmann and Baldwin
Autonomy	Operational independence	Type	Directed
Belonging	Managerial independence	Control (or autonomy)	Acknowledged
Connectivity	Geographic separation	Connectivity	Collaborative
Diversity	Emergent behavior		Virtual
Emergence	Evolutionary development		

Mastering Systems Integration; Impact of Change

by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

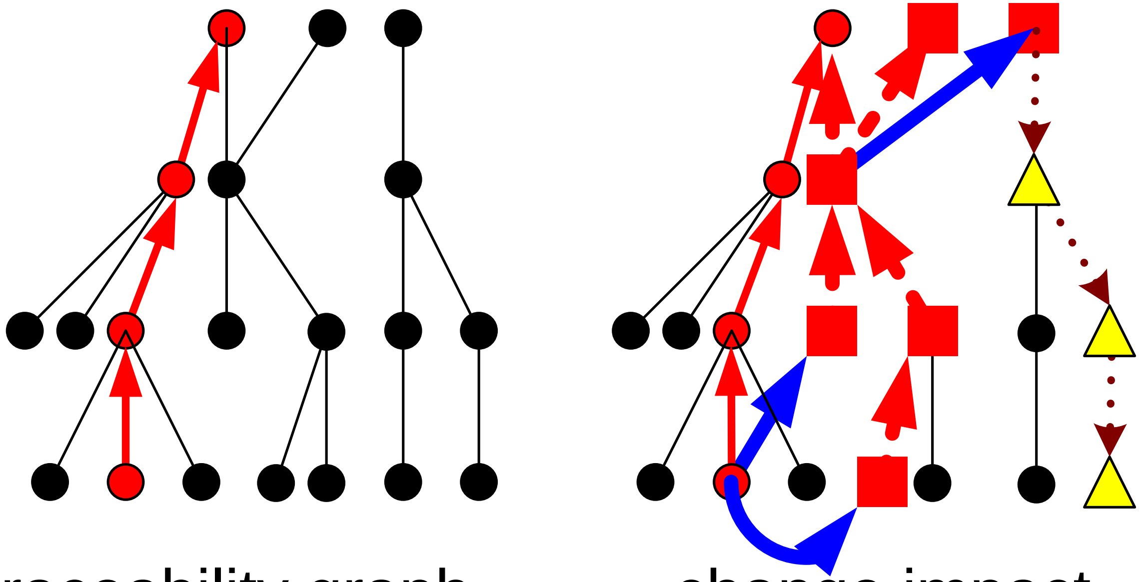
This presentation explains the imact of change. A frequent problem is that people do not foresee the imoact of changes they make to the system. A naieve assumption is that traceability or dependecy graphs will show them such impact. Realty then hits them during systems integration.



Distribution

March 3, 2024
status: planned
version: 0

Impact of Change due to Unforeseens



traceability graph
ideal change impact

change impact
in practice

____ specified requirements relation

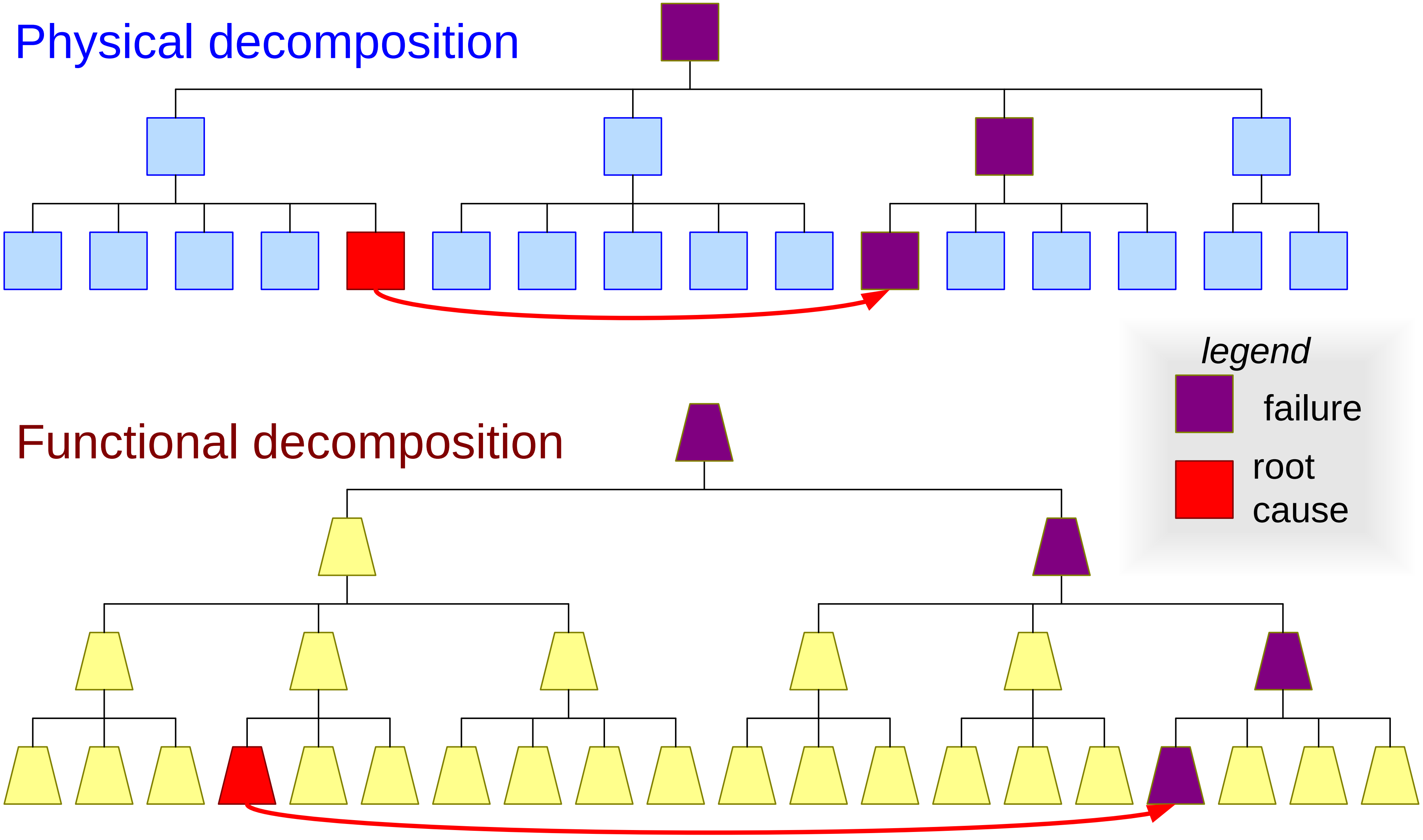
- - ➤ propagation due to uncharted relations

➤ unknown unforeseen ignored relations

· · · ➤ additional changes to repair change impact

legend

Root Cause is Often Elsewhere



Physical decomposition

Functional decomposition

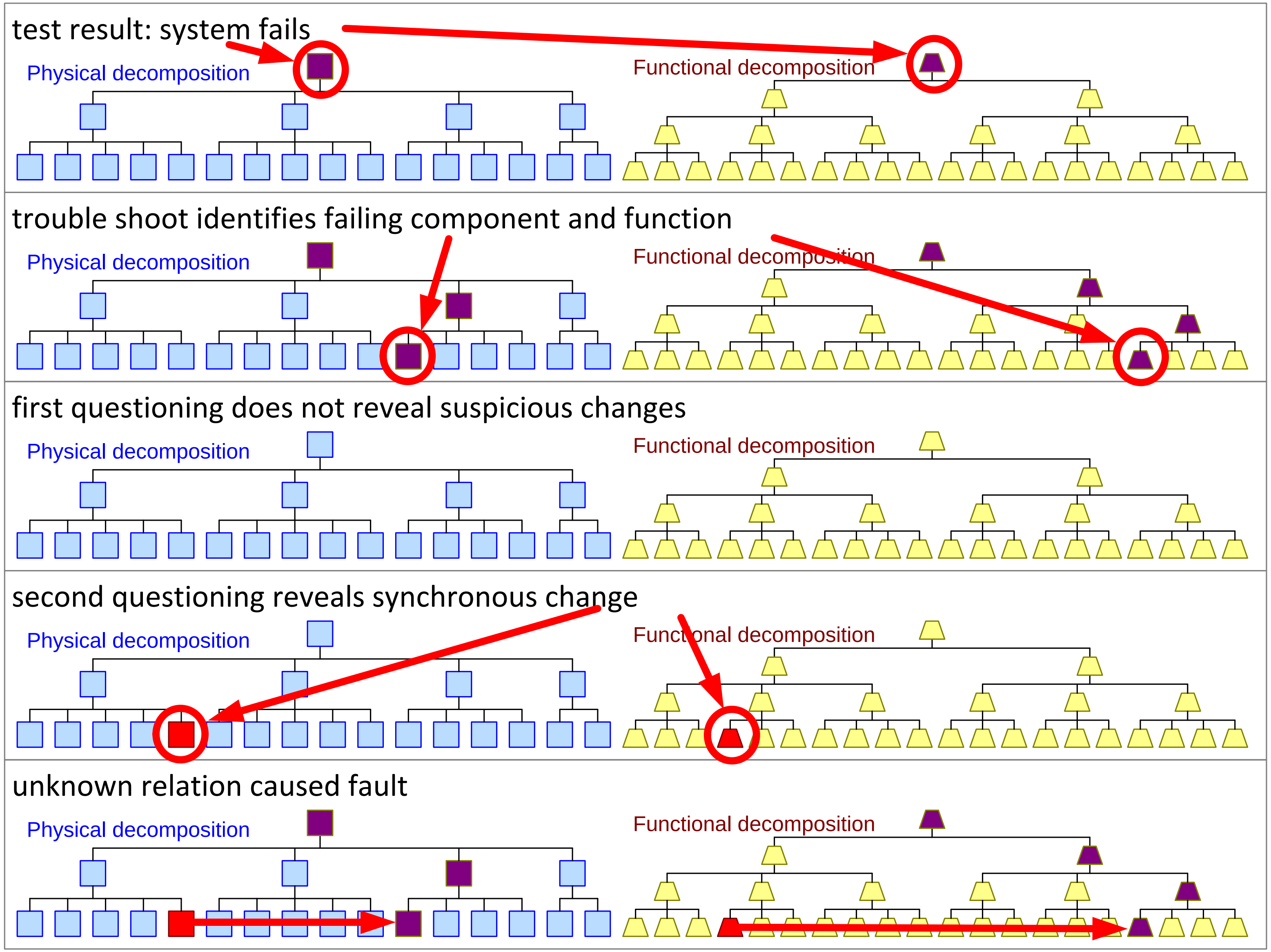
legend

🟪 failure

🟥 root cause

unknown or unexpected relation

"Nothing has been changed..."



Mastering Systems Integration; Software and Integration

by *Gerrit Muller*   TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

Software has a number of characteristics, which impact systems integration.

Distribution

March 3, 2024
status: planned
version: 0

logo
TBD

System or Software?

When SW engineers demand "requirements",

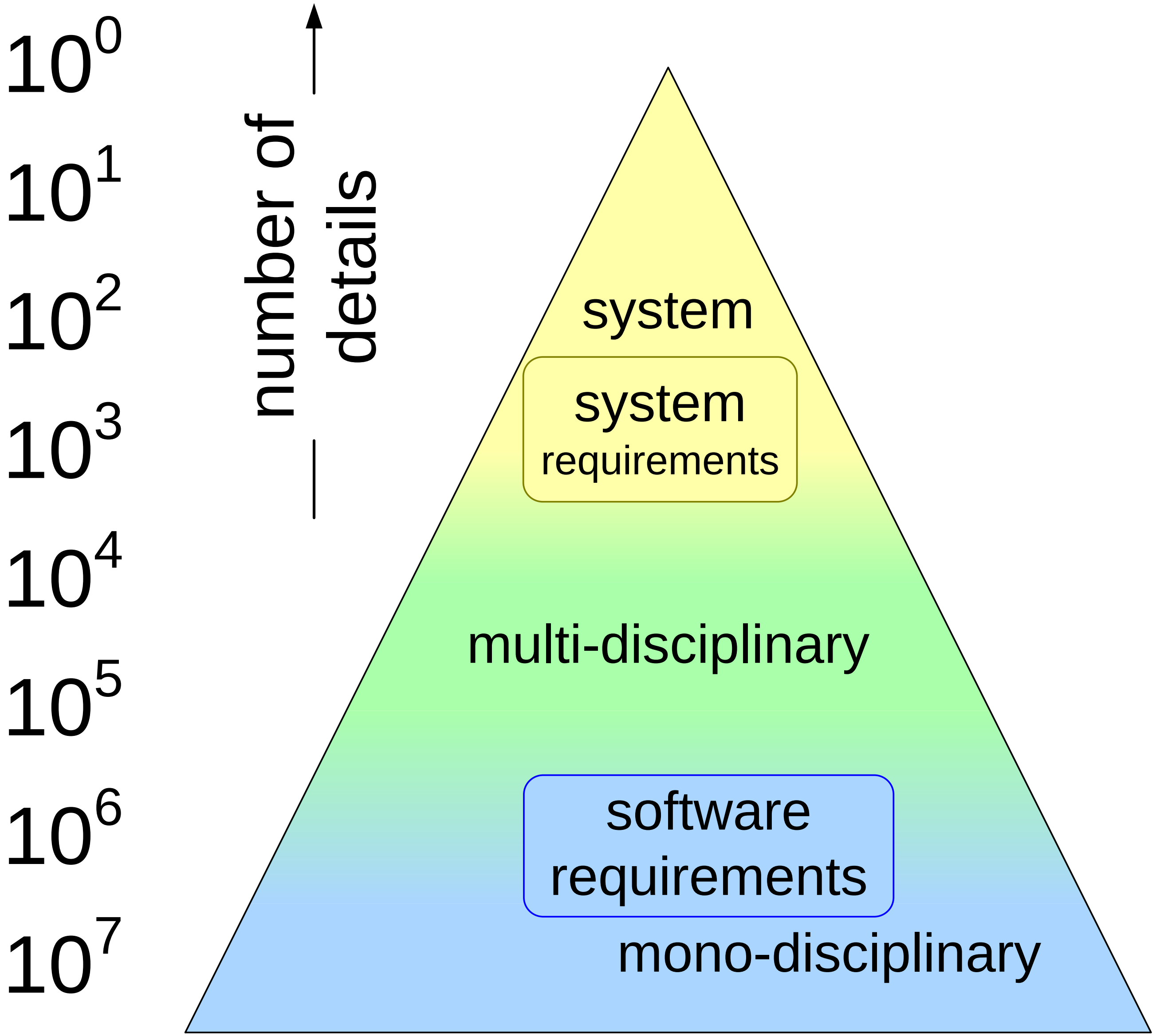
then they expect *frozen* inputs

to be used for

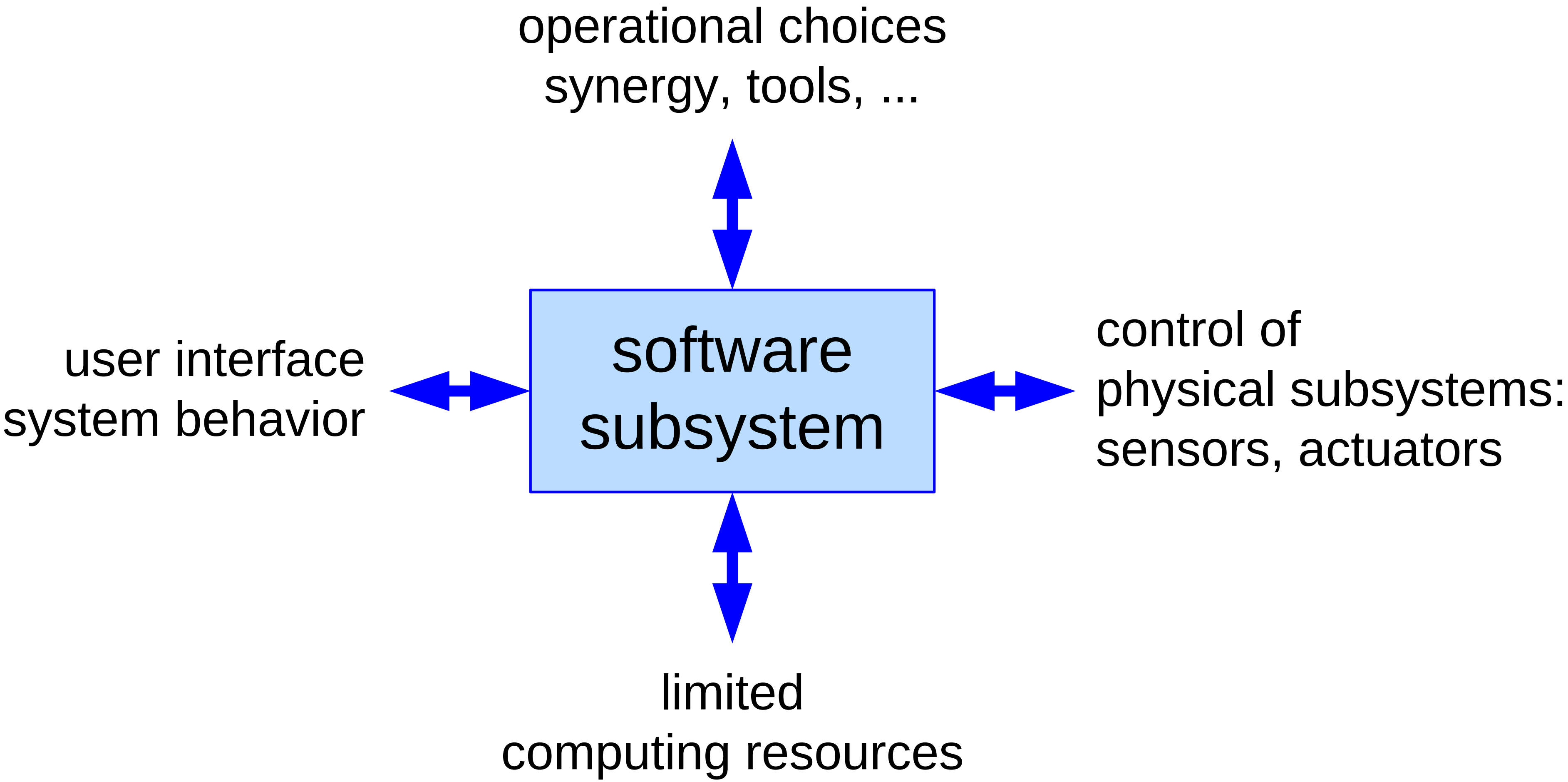
the design, implementation and validation

of the software

System vs Software Requirements



Why is the Software Requirement Specification so Large?



operational choices
synergy, tools, ...

user interface
system behavior

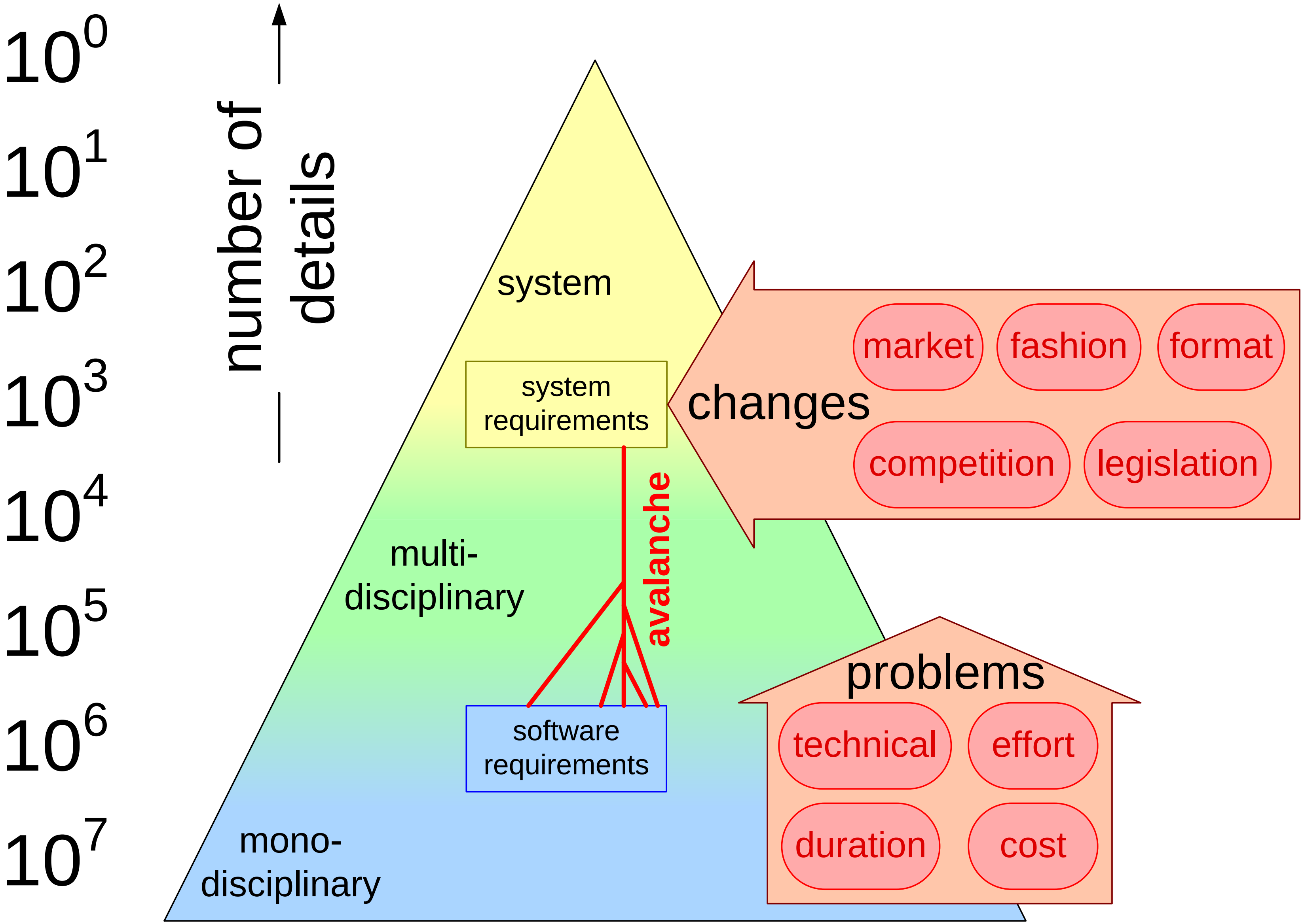
software
subsystem

control of
physical subsystems:
sensors, actuators

limited
computing resources

version: 0
March 3, 2024
VREQsoftwareSubsystem

And why is it never up-to-date?



version: 0
March 3, 2024
VREQdynamics

Different Focus of Software and System

System engineering focus

qualities
productivity
image quality
reliability

concepts
domain requirements
models

concerns
integral design (quality, balance)
system context
lifecycle
operational processes

education
principles
heuristics
analysis and synthesis
processes

SW engineering focus

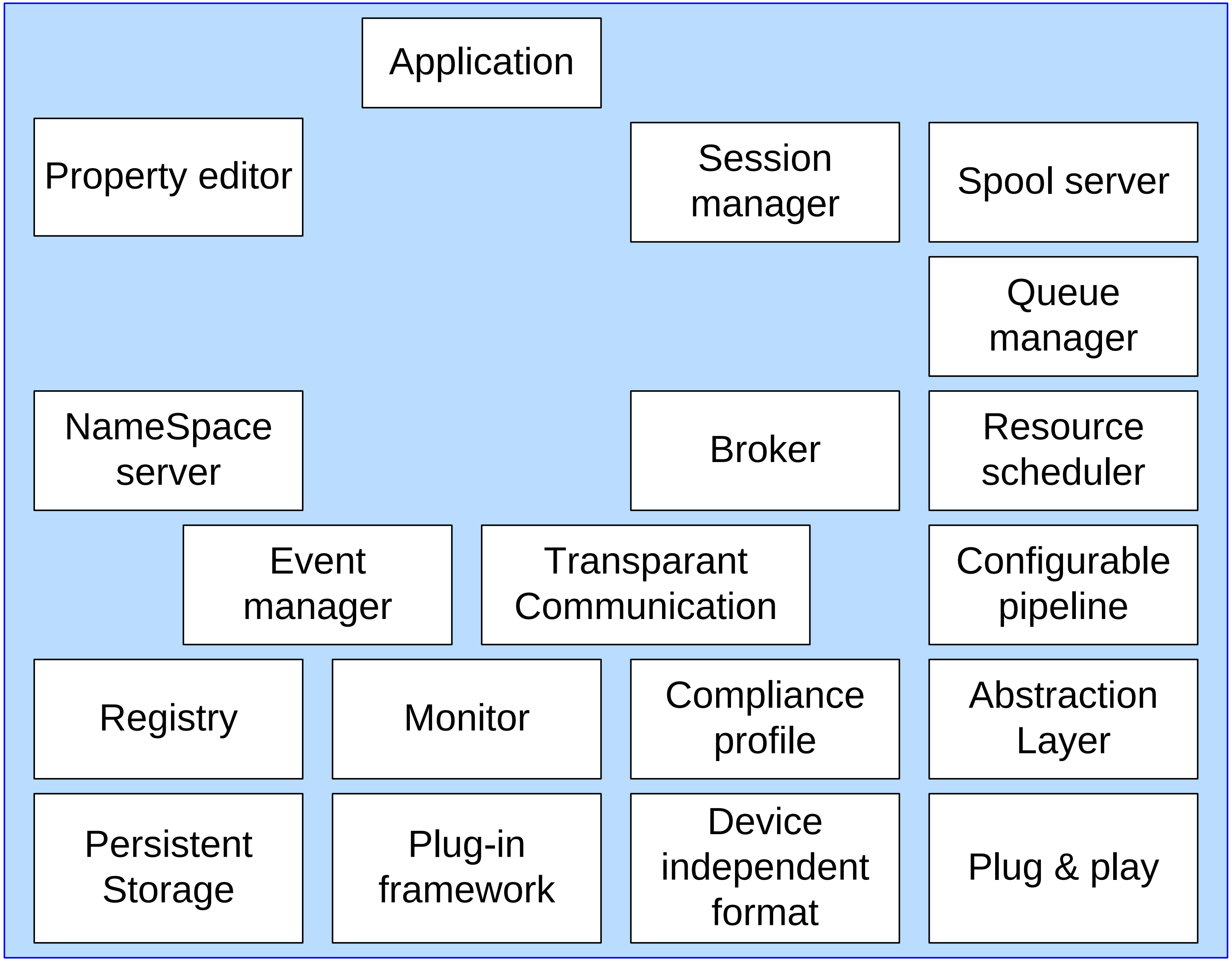
qualities
functionality
maintainability
variability

concepts
structure
(generic) mechanisms

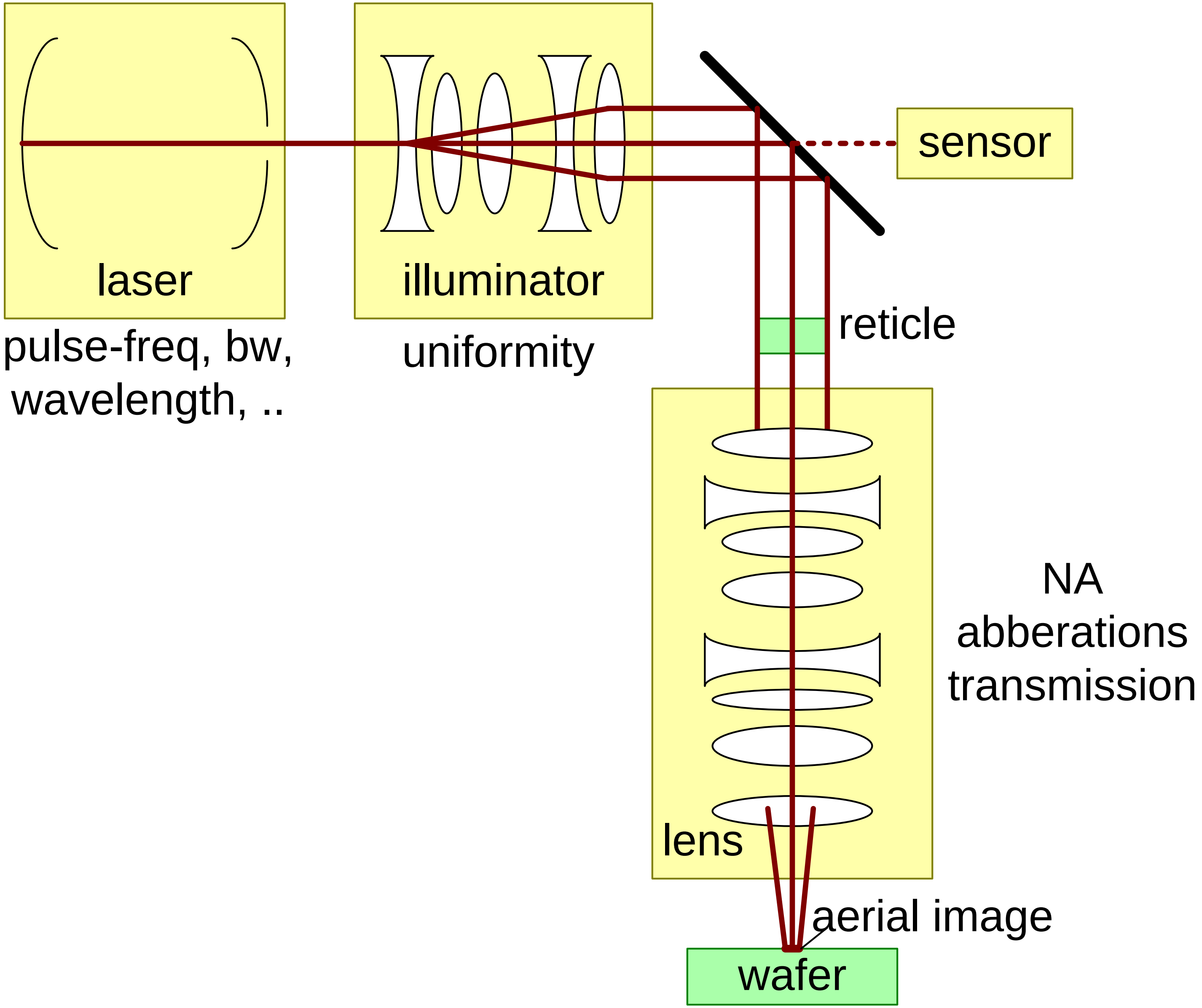
concerns
configuration management
release procedure
tools
SW processes
SW problems, change requests

education
languages
operating systems
algorithms
formal methods

Caricature of a SW Architecture

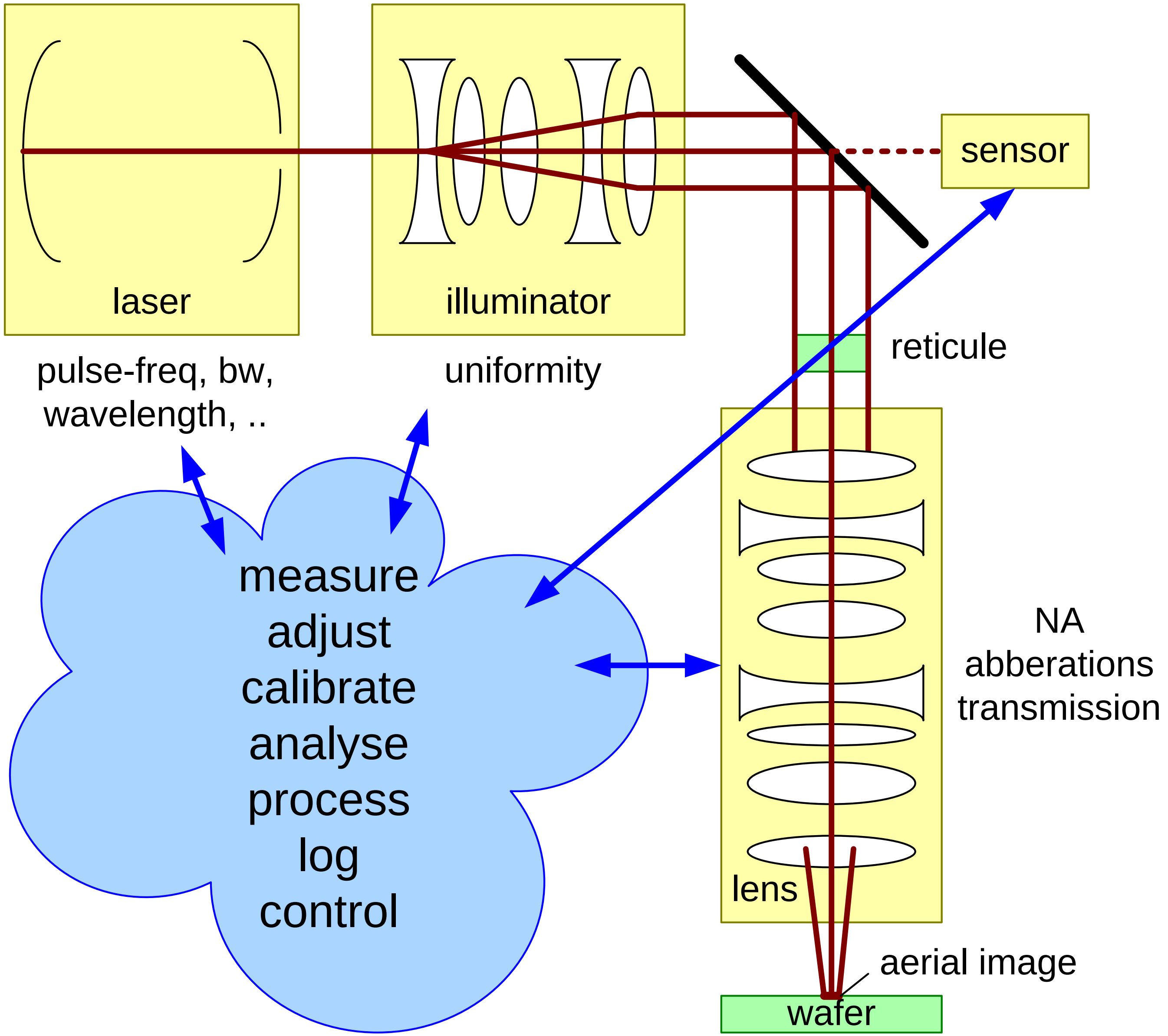


Caricature of Physics Systems View



version: 0
March 3, 2024
TSAITphysicsView

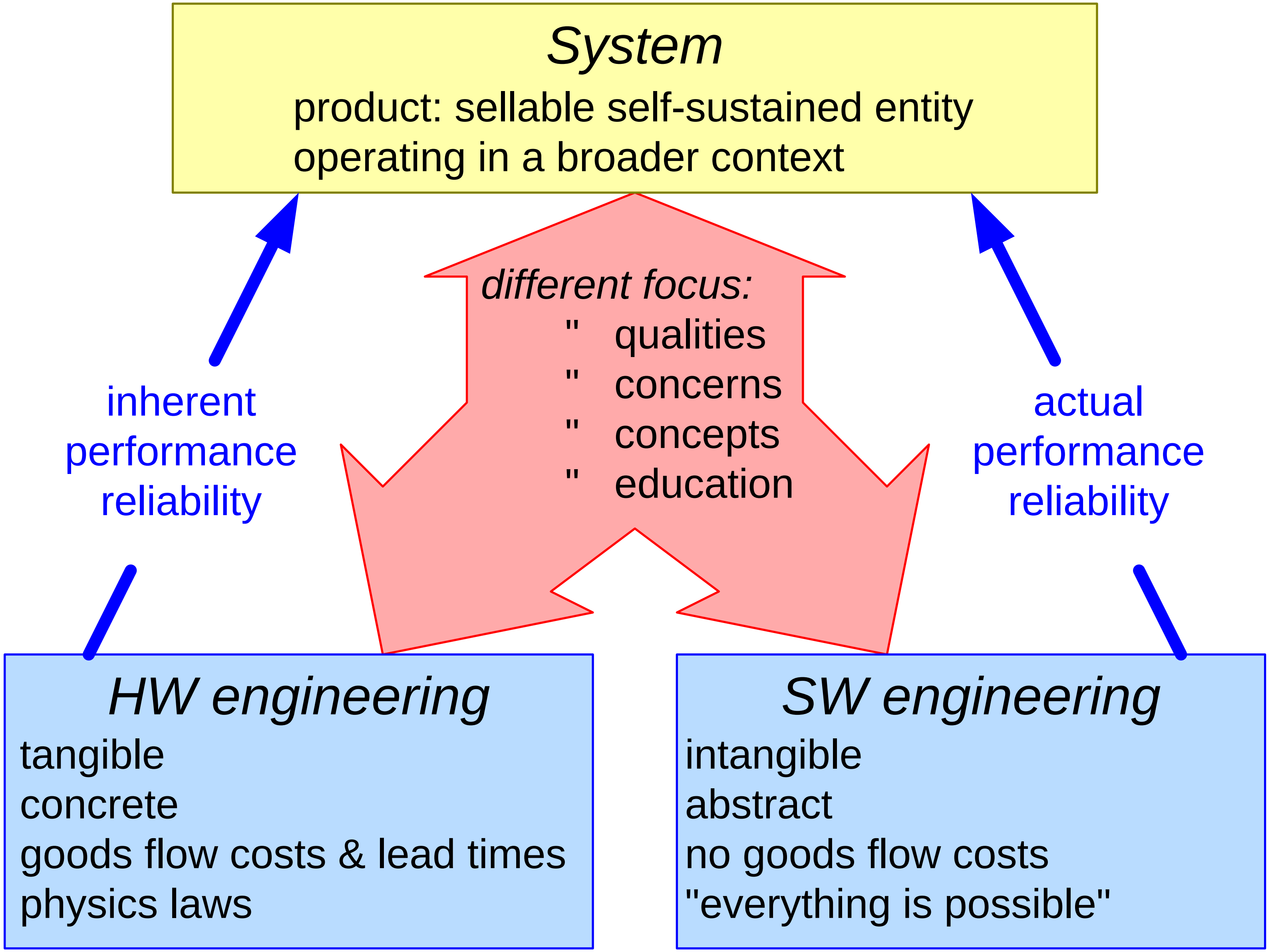
Relation SW and Physics



Symptoms of too isolated SW efforts

symptoms	*counter measures*
SW people are clustered together	colocation per function, subsystem or quality
SW is alpha tested before system integration	continuous system integration
SW team uses own specification and design process	higher level processes are shared
SW specification is in SW jargon or formalism	interaction between SW, HW and system engineers

Hardware Software System



System

product: sellable self-sustained entity
operating in a broader context

different focus:

- qualities
- concerns
- concepts
- education

inherent
performance
reliability

actual
performance
reliability

HW engineering

tangible
concrete
goods flow costs & lead times
physics laws

SW engineering

intangible
abstract
no goods flow costs
"everything is possible"

Mastering Systems Integration; Product Families and Platforms

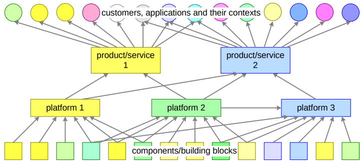
by *Gerrit Muller* TNO-ESI, University of South-Eastern Norway]
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

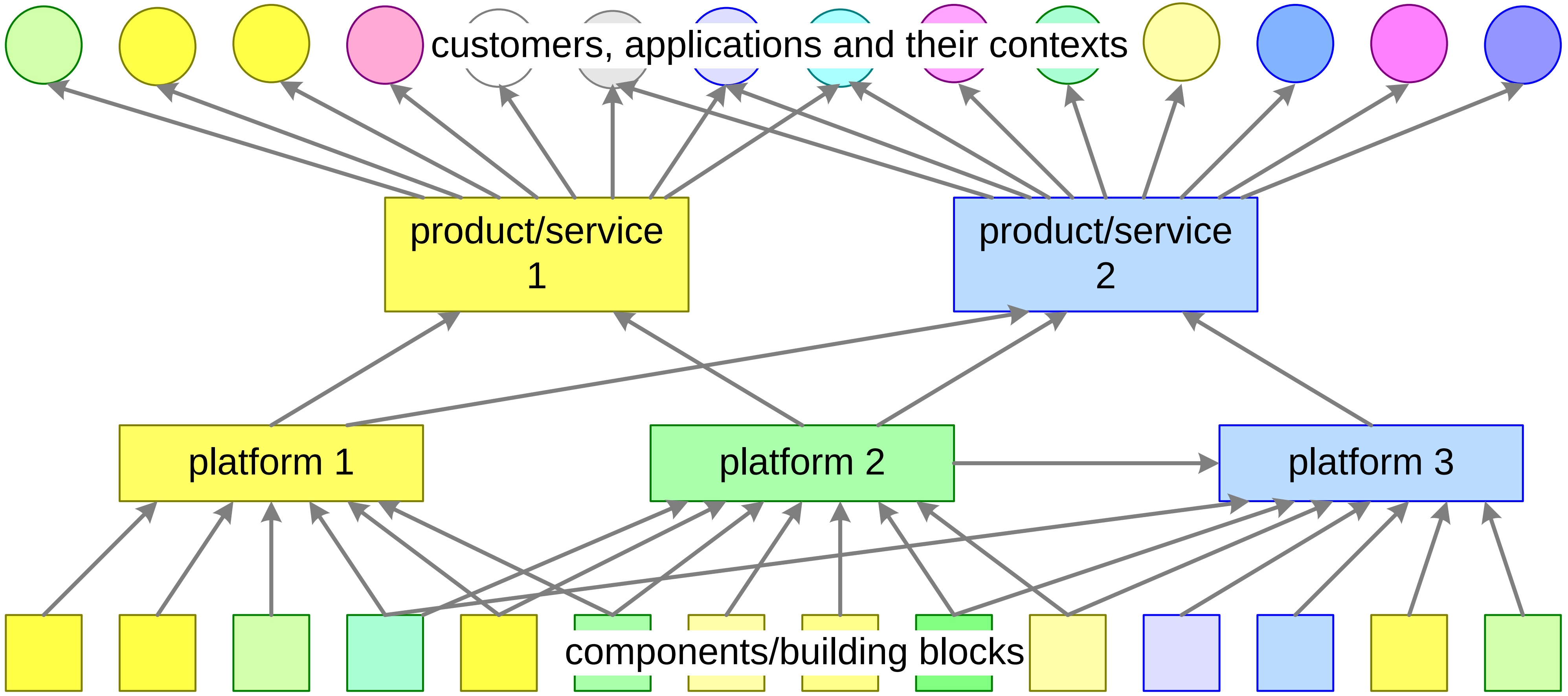
Many systems, products, and services depend on sharing realizations of functions and subsystems. The organization may organize the sharing in platforms, such that the product developers focus primarily on their added value to the applications. Sharing, however, is a complication for integration. Changes in a component may propagate to various platforms to multiple products and services in widely different circumstances.

Distribution

March 3, 2024
status: preliminary draft
version: 0.1

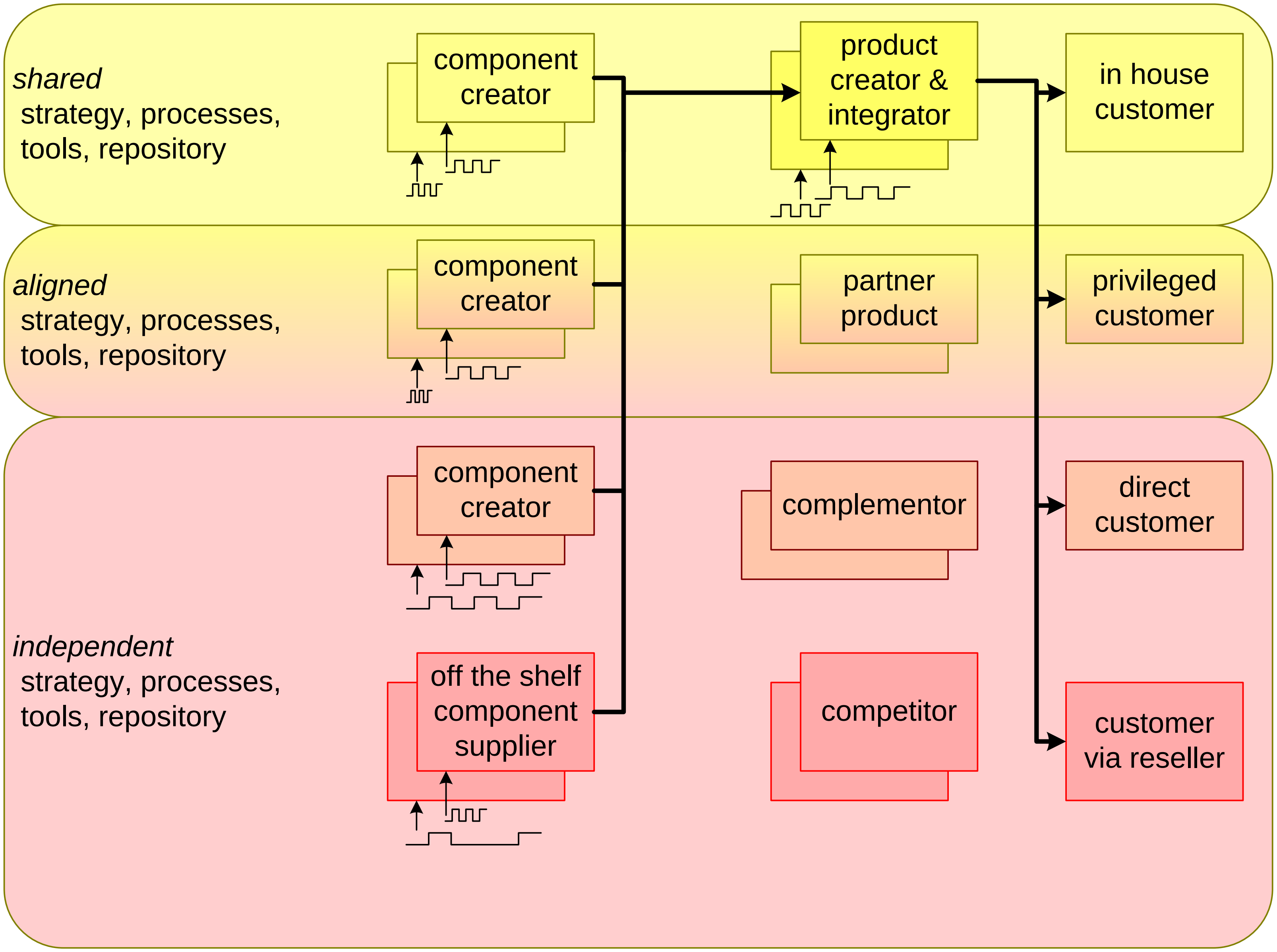


Dependency Network



version: 0.1
March 3, 2024
MSIPFdependencyNetwork

Varying Clock Cycles across Creation Chain

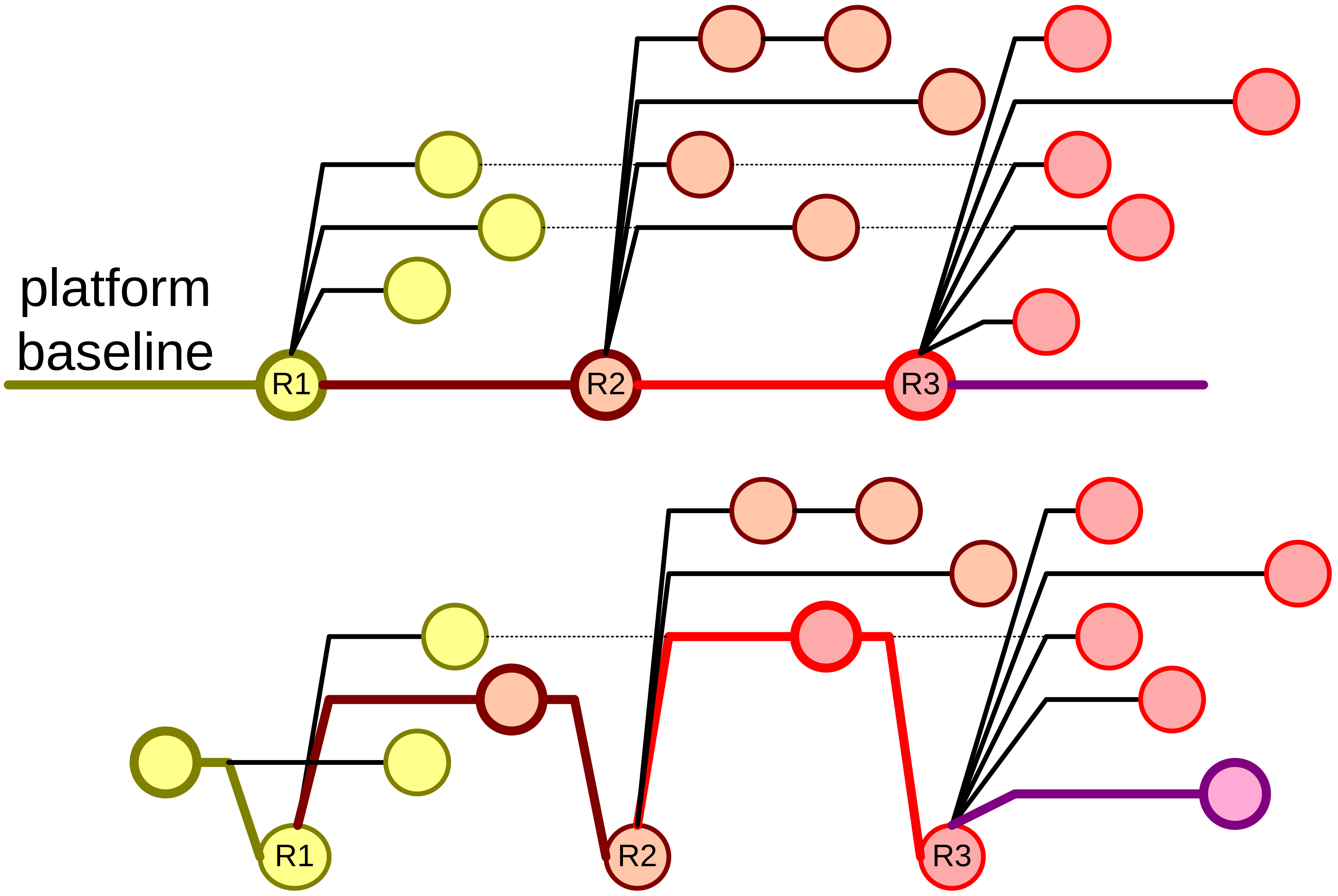


shared
strategy, processes,
tools, repository

aligned
strategy, processes,
tools, repository

independent
strategy, processes,
tools, repository

Release Models for Platforms and Projects/Products



platform
baseline

platform
as consolidation baseline

version: 0.1
March 3, 2024
HMPAreleaseModel

Mastering Systems Integration; Course Material

by *Gerrit Muller* TNO-ESI, University College of South-Eastern Norway
e-mail: `gaudisite@gmail.com`
`www.gaudisite.nl`

Abstract

Listing the course material for the course Systems Integration

Distribution

March 3, 2024
status: planned
version: 0.5

logo
TBD

Colofon

The Systems Integration course is partially derived from the Systems Integration and Test course developed at *TNO-ESI* by *Teade Punter*, *Frans Beenker*, and *many others*.

Introduction

core

Mastering Systems Integration; Introduction

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

optional

Course Overview

core

Mastering Systems Integration; Course Overview

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

optional

Process and Positioning

core

Mastering Systems Integration; Process and Positioning

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

optional

SESA /SARCH Module 01, System Architecture Context

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

Hardware, Software, Systems

core

Course Systems Integration; Hardware, Software, System

http://www.gaudisite.nl/info/MSIhardwareSoftwareSystem.info.html

optional

Tutorial Software as Integrating Technology in Complex Systems

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

Terminology

core

Course Systems Integration; Terminology

http://www.gaudisite.nl/info/MSIterminology.info.html

optional

Understanding Objective Evidence: (What It Is and What It Definitely Is Not),

by Denise Dion

http://www.eduquest.net/Advisories/EduQuest%20Advisory_ObjectiveEvidence.pdf

List of Cognitive Biases, Wikipedia:

https://en.wikipedia.org/wiki/List_of_cognitive_biases

Economic Perspective

core

Mastering Systems Integration; Economic Perspective

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

optional

Simplistic Financial Computations for System Architects.

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

Visualizing Dynamic Behavior

core

Visualizing Dynamic Behavior

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

optional

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

http://gaudisite.nl/INCOSE2015_MullerEtAl_SubseaOverviewA3.pdf

Early Validation

core

Course Systems Integration; Early Validation

http://www.gaudisite.nl/info/MSIearlyValidation.info.html

optional

System Integration How-To

http://www.gaudisite.nl/info/SystemIntegrationHowTo.info.html

Save Money by Investing In Models; Failing Early is More affordable Than Failing Late

http://gaudisite.nl/SaveMoneyInvestInModelsSlides.pdf

Light Weight Architectures; The way of the future?

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

Project Management

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

Testing

core

Course Systems Integration; Testing

http://www.gaudisite.nl/info/MSItesting.info.html

optional

What is wrong with Reliability Engineering, by R.W.A. Barnard, Proceedings of INCOSE 2008 in Utrecht.

Highly accelerated life test

https://en.wikipedia.org/wiki/Highly_accelerated_life_test

Readiness Levels

core

Course Systems Integration; Readiness Levels

http://www.gaudisite.nl/info/MSIreadinessLevels.info.html

optional

From TRL to SRL: The Concept of Systems Readiness Levels

CSER 2006, Brian Sauser et al.

Technology Readiness Levels

https://en.wikipedia.org/wiki/Technology_readiness_level

System of Systems

core

Mastering Systems Integration; System of Systems

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

optional

J. Dahmann and K. Baldwin. 2008. "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering." IEEE Systems Conference 2008 in Montreal, 2008.

Boardman, J. and B. Sauser, System of Systems - the meaning of of, in IEEE/SMC International Conference on Systems of Systems Engineering. 2006, IEEE: Los Angeles.

Gorod, A., White, B.E., Ireland, V., Gandhi, J.S., and Sauser, B., (editors) “Case studies in System of Systems, Enterprise systems, and Complex Systems Engineering”, CRC Press, 2014.

Software and Integration

core

Course Systems Integration; Software and Integration

http://www.gaudisite.nl/info/MSIsoftwareAndIntegrationinfo.html

optional

Tutorial Software as Integrating Technology in Complex Systems

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