Module Supporting Processes

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

This module addresses supporting processes, for instance documentation, templates, and reviewing.

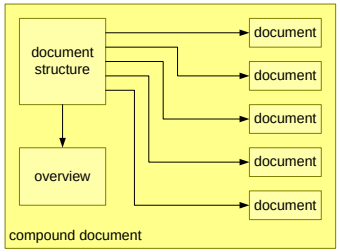
March 27, 2021
status: draft
version: 1.4

Granularity of Documentation

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

Abstract

The design of documentation is discussed, with emphasis on the requirements, the need for decomposition, the measures needed to maintain overview and criteria for granularity.



March 27, 2021
status: concept
version: 1.2

Requirements for the Entire Documentation Structure

Accessibility for the readers

Low threshold for the readers

Low threshold for the authors

Completeness

Consistency

Maintainability

Scalability

Evolvability

Process to ensure the quality of the information

Requirements from Reader Point of View

Convenient

easy

fast

viewing

printing

searching

Requirements per Document

High cohesion (within the unit)

Low coupling (outside of the unit)

Accessibility for the readers

Low threshold for the reader

Low threshold for the author

Manageable steps to create, review, and change

Clear responsibilities

Clear position and relation with the context

Well-defined status of the information

Timely availability

Accessibility Requirements

Ease of reading, “juiciness”

High signal-to-noise ratio: information should not be hidden in a sea of words.

Understandability

Reachability in different ways, e.g., by hierarchical or full search

Reachability in a limited number of steps

Responsibility Requirements

single author

limited amount of reviewers

Scalability Requirements

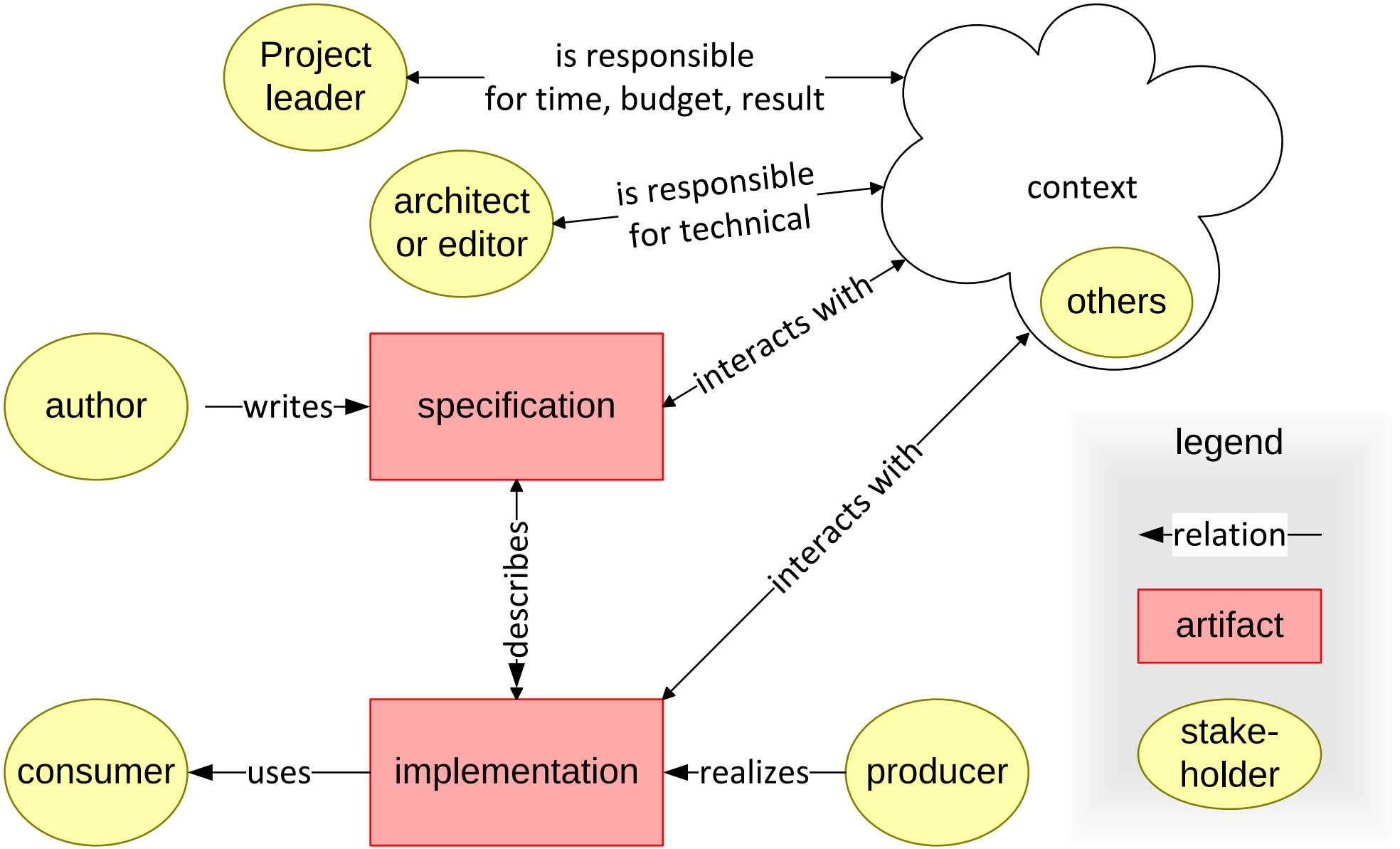
well defined documentation structure

overview specifications at higher aggregation levels

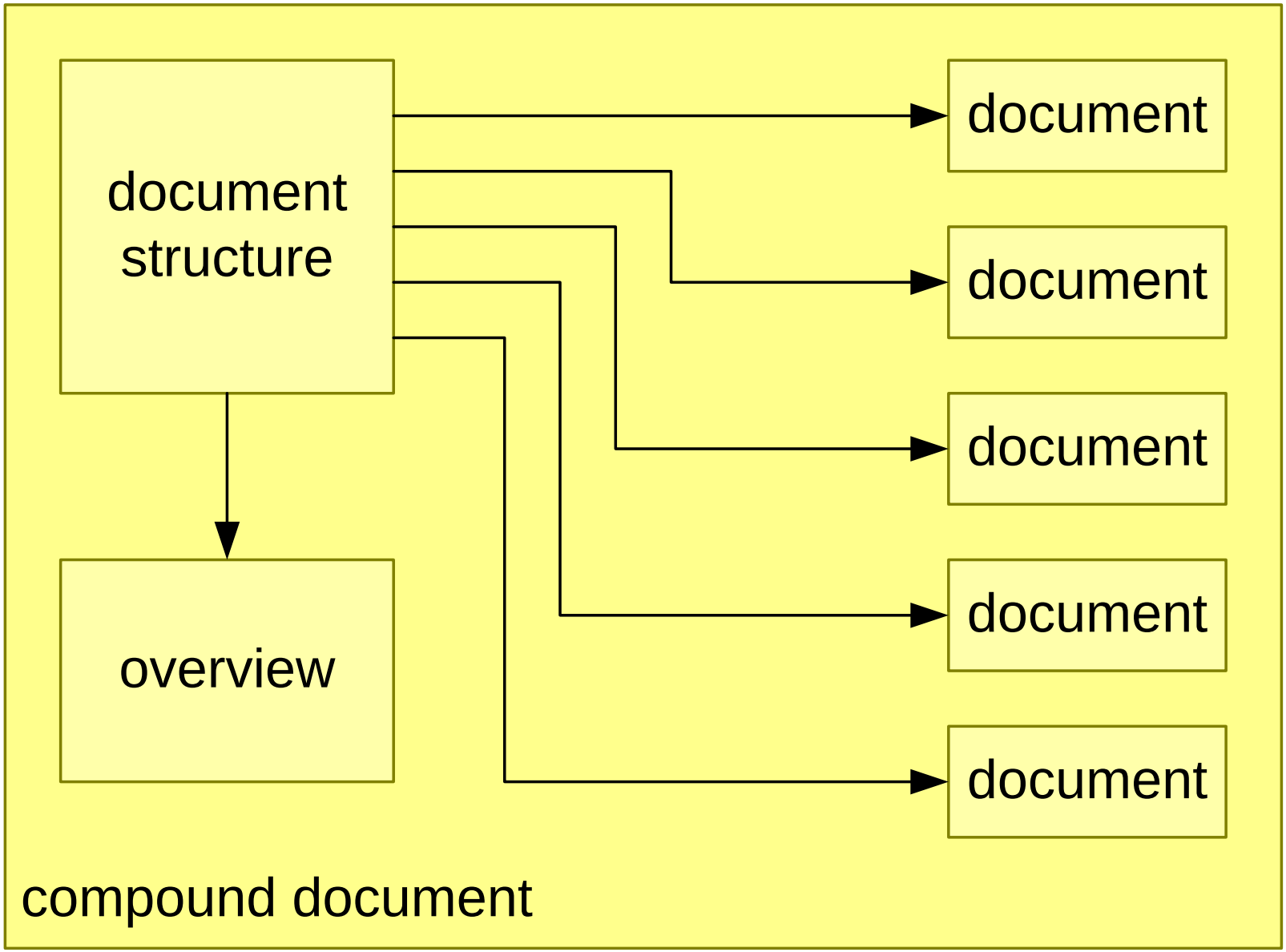
recursive application of structure and overview

delegation of review process

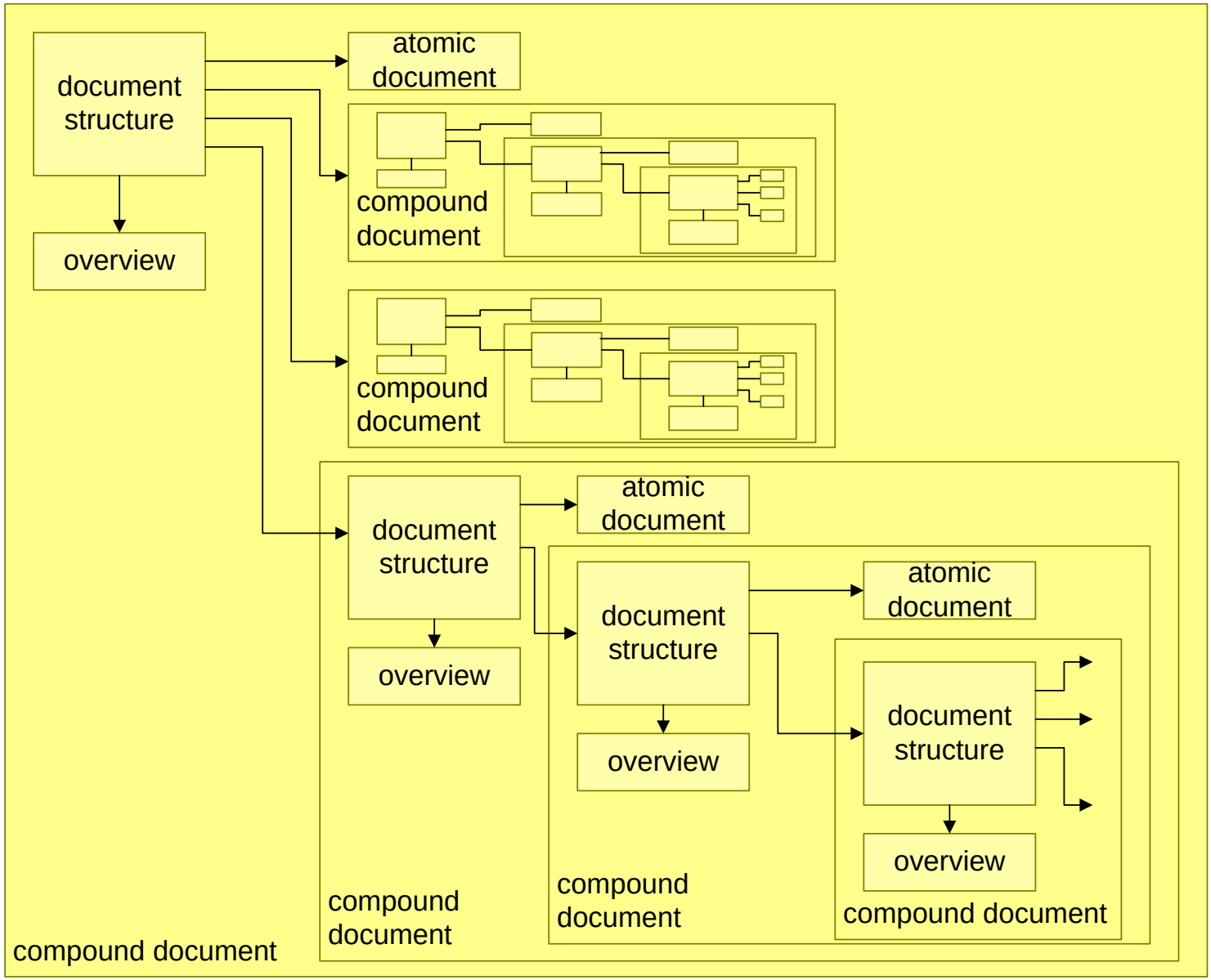
The Stakeholders of a Single Document



Decomposition of Large Documents

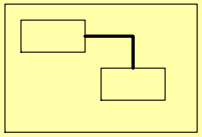
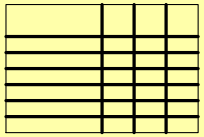


Documentation Tree by Recursive Decomposition



Payload: the Ratio between Content and Overhead

front page

title identification author distribution status review	history changes	 diagrams  tables	1. aap 2. noot 3. mies lists and ca 50% text

meta information
max 2 pages

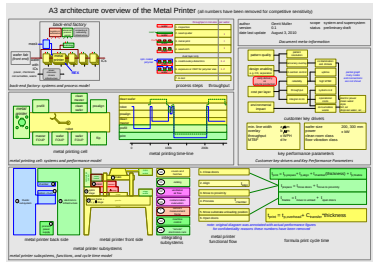
contents
2..18 pages

LEAN and A3 Approach to Supporting Processes

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

LEAN product development is in the process and means area pragmatic. Low tech tools, such as paper, pen and magnets, with very direct interaction are used. For communication the use of single A3-size documents is promoted, because this is a manageable amount of information.



March 27, 2021
status: draft
version: 0.1

Characteristics of LEAN

A holistic, systems approach to product development including people, processes, and technology.

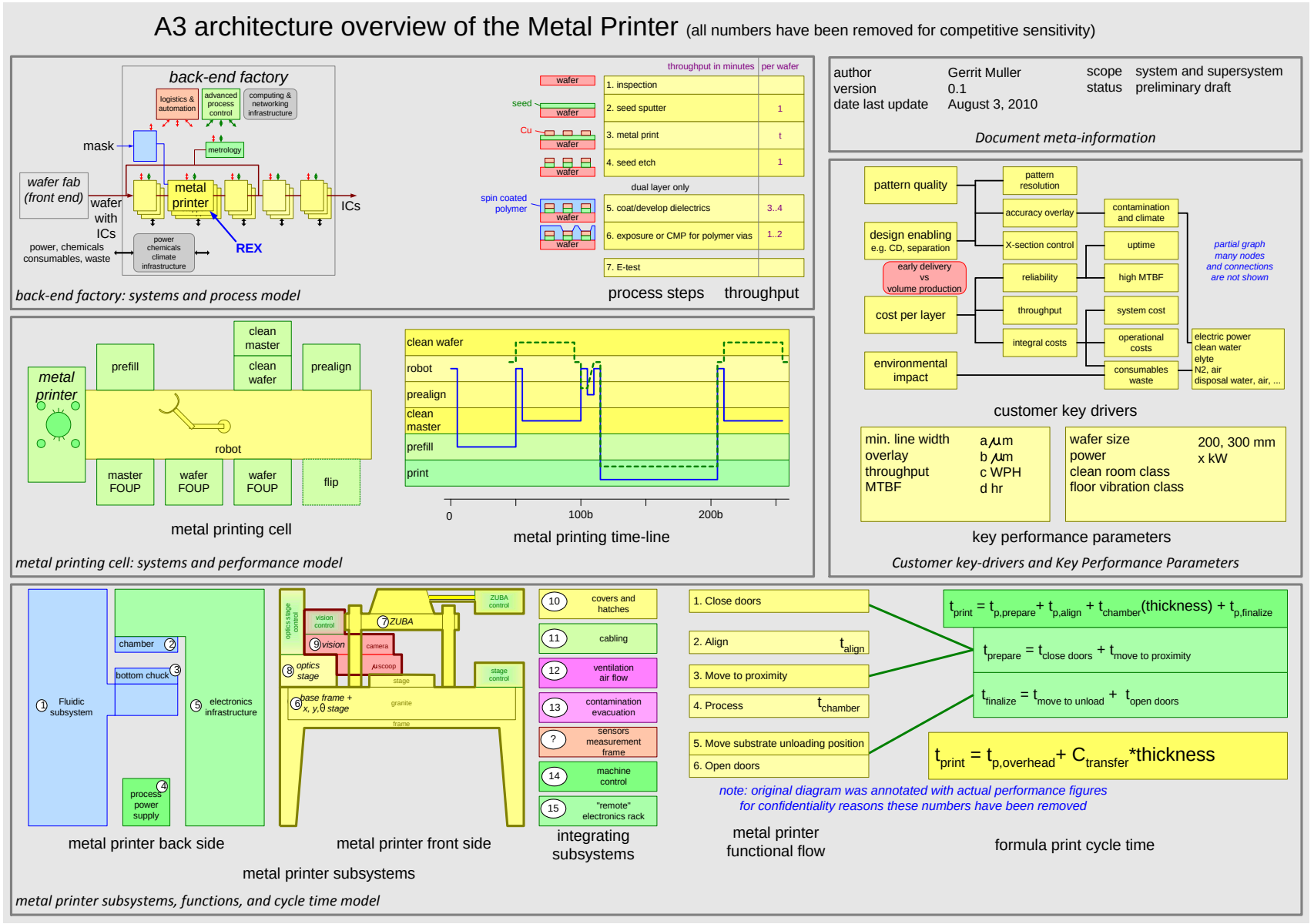
Multi-disciplinary from the early start, with a drive to be fact based.

Customer understanding as the the starting point.

Continuous improvement and learning as cultural value.

Small distance between engineers and real systems, including manufacturing, sales and service and the system of interest.

Example of A3 Architecture Overview

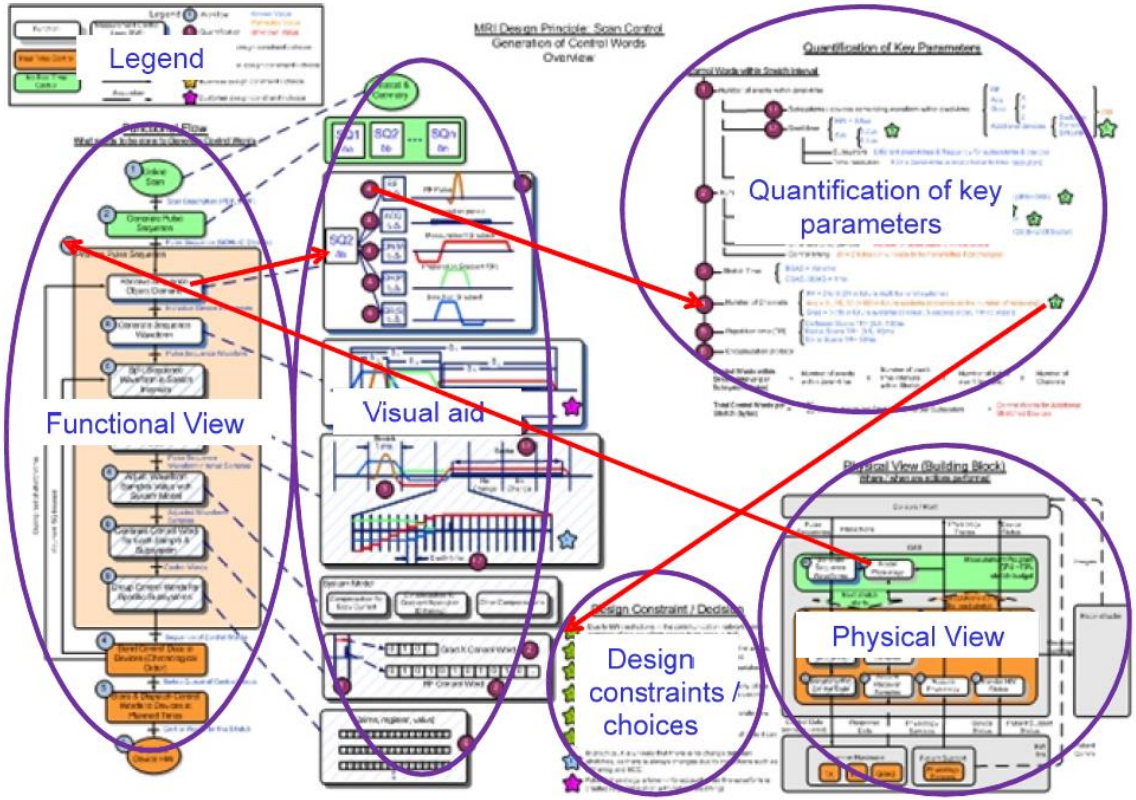


multiple related views

quantifications

one topic
per A3

capture
"hot" topics



source: PhD thesis Daniel Borches http://doc.utwente.nl/75284/

digestable
(size limitation)

practical
close to stakeholder experience

Light Weight Review Process

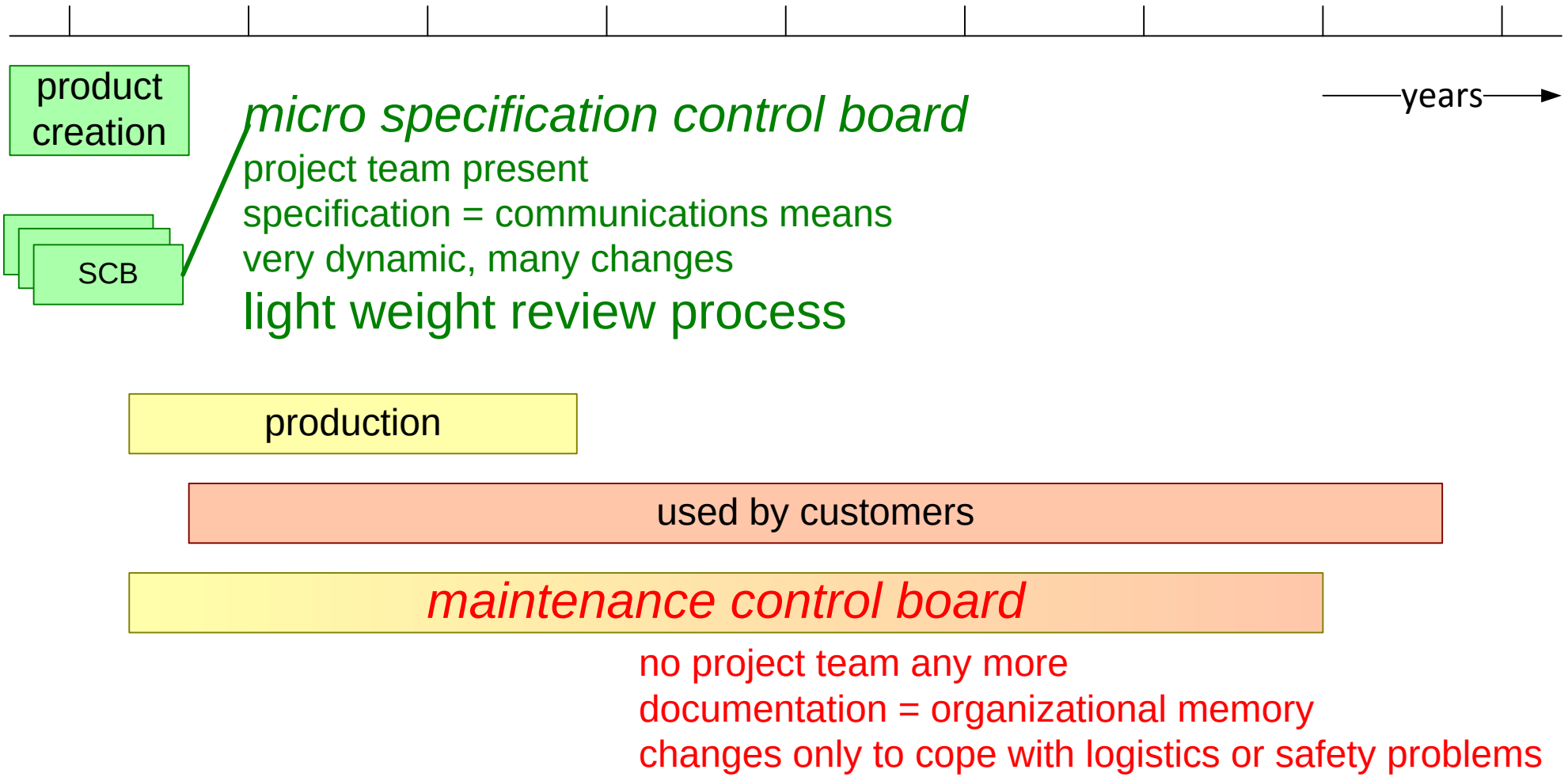
by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

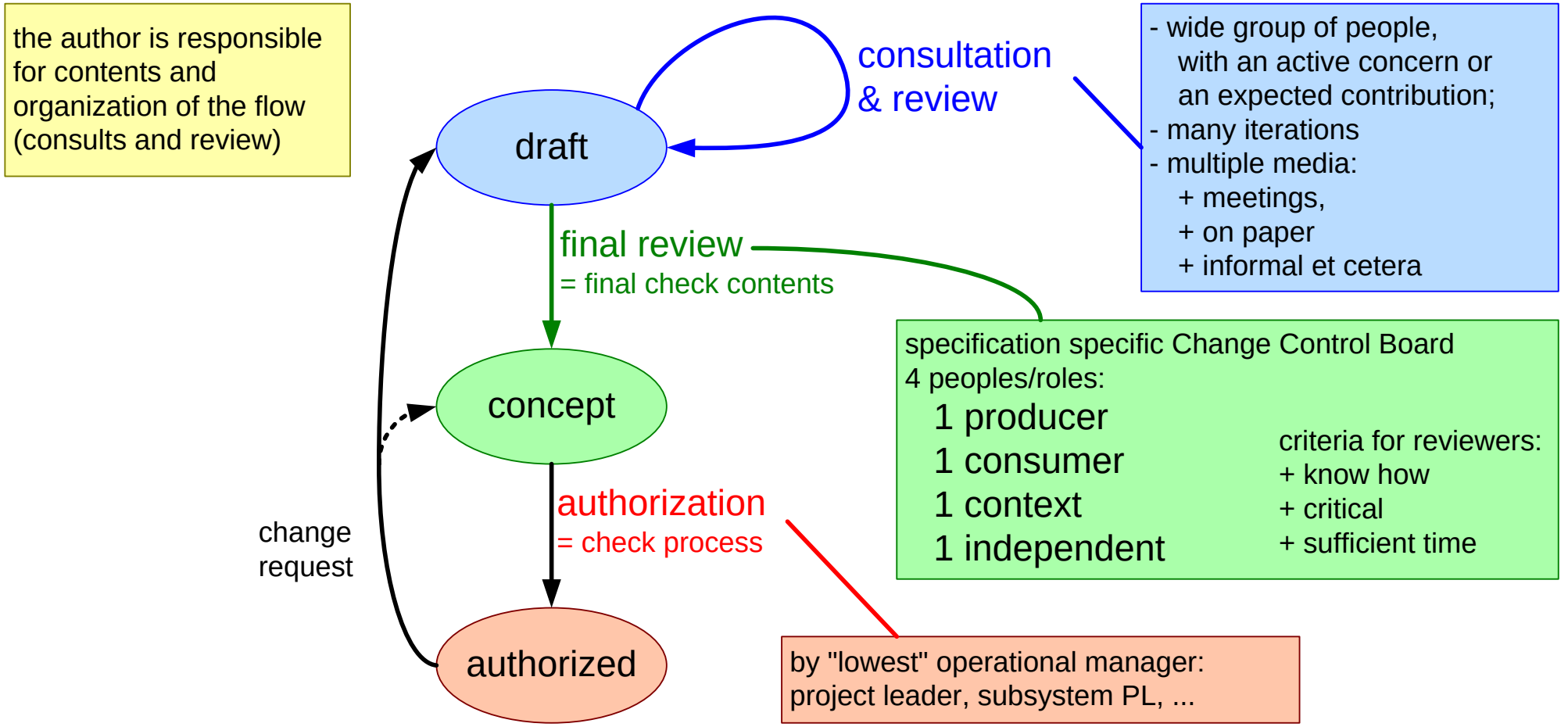
A light weight review process is described that can be used for documents made during product creation. This review process is focused on improving the contents of specifications as early as possible. The process is light weight to increase the likelihood that it is performed *de facto* instead of *pro forma*.

March 27, 2021
status: preliminary draft
version: 0

Product Life Cycle and Change Management



Light Weight Specification Review Process

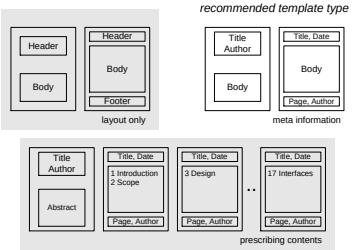


Template How To

by *Gerrit Muller* University of South-Eastern Norway-NISE
e-mail: gaudisite@gmail.com
www.gaudisite.nl

Abstract

The introduction of a new process (way of working) is quite often implemented by supplying ready-to-go tools and templates. This implementation mainly serves the purpose of a smooth introduction of the new process.
Unfortunately the benefits of templates are often cancelled by unforeseen side-effects, such as unintended application, inflexibility, and so on. This intermezzo gives hints to avoid the **Template Trap**, so that templates can be used more effectively to support introduction of new processes.



March 27, 2021
status: draft
version: 1.6

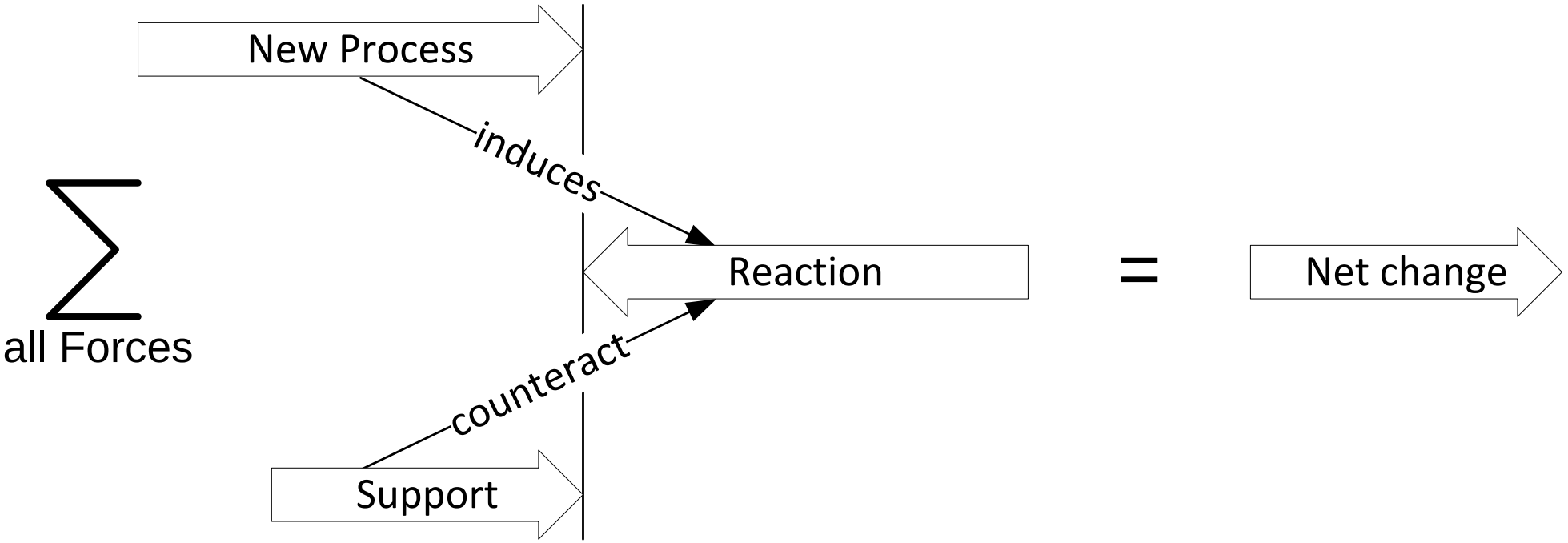
Rationale for Templates

- Low threshold to apply a (new) process (1)
- Low effort to apply a (new) process (2)
- No need to know low level implementation details (3)
- Means to consolidate and reuse experiences (4)

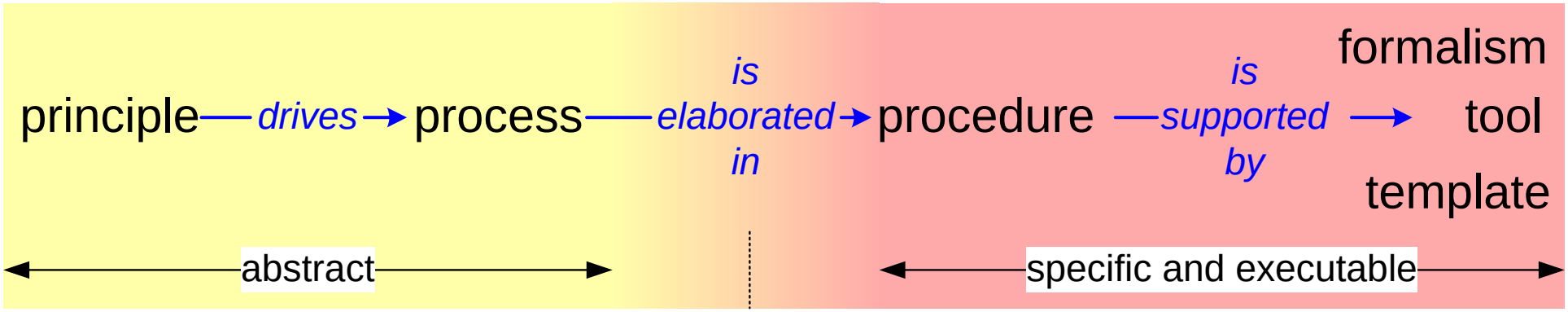
Bogus Arguments for Templates

- Obtain a uniform look (5)
- Force the application of a (new) process (6)
- Control the way a new process is applied (7)

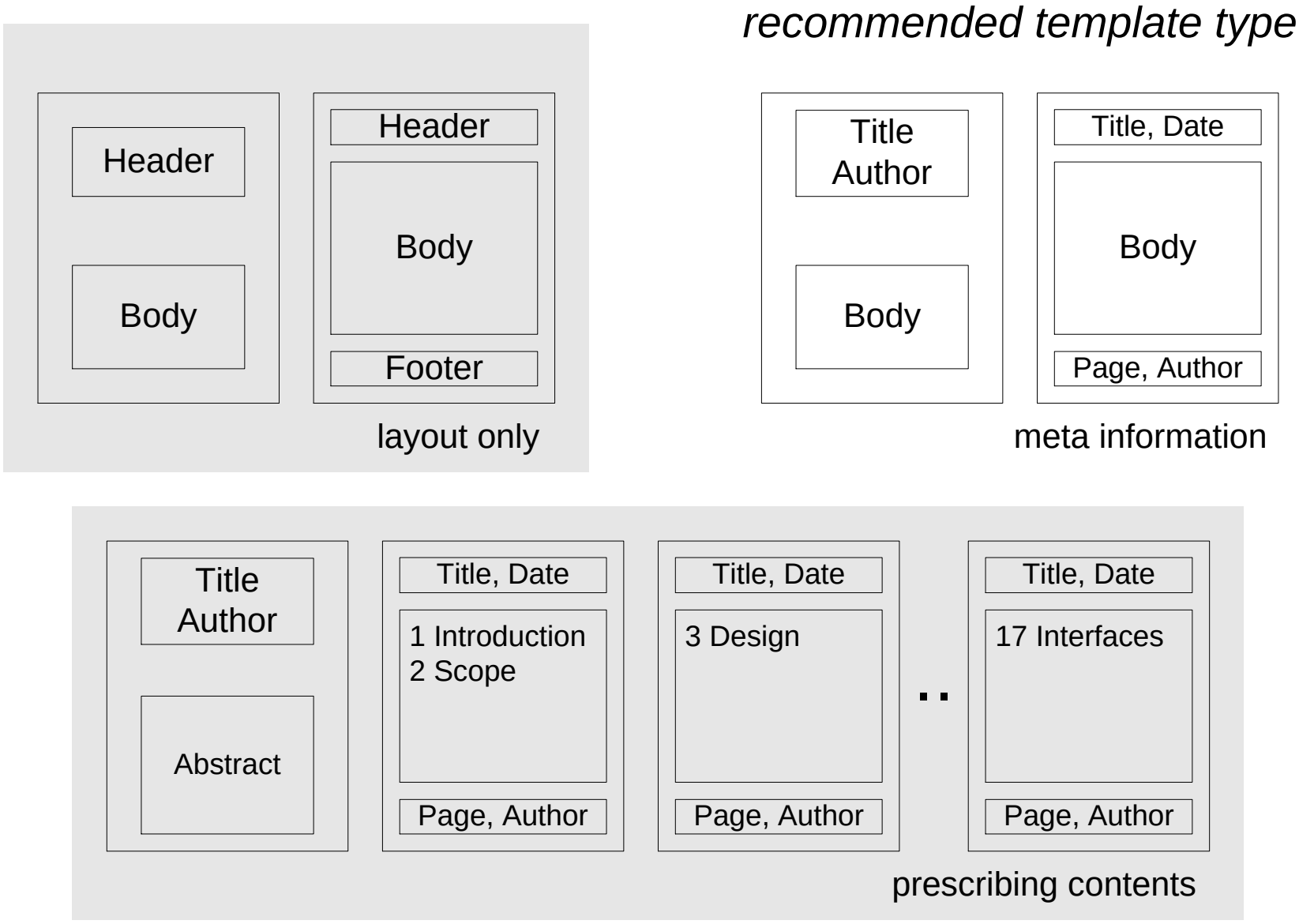
Forces of Change: Action = - Reaction



Template as Support for Process



Types of Templates



Recommendation

template type	context knowhow	value
layout only	no	low
meta information	process	high
prescribing content	process and domain	constraining

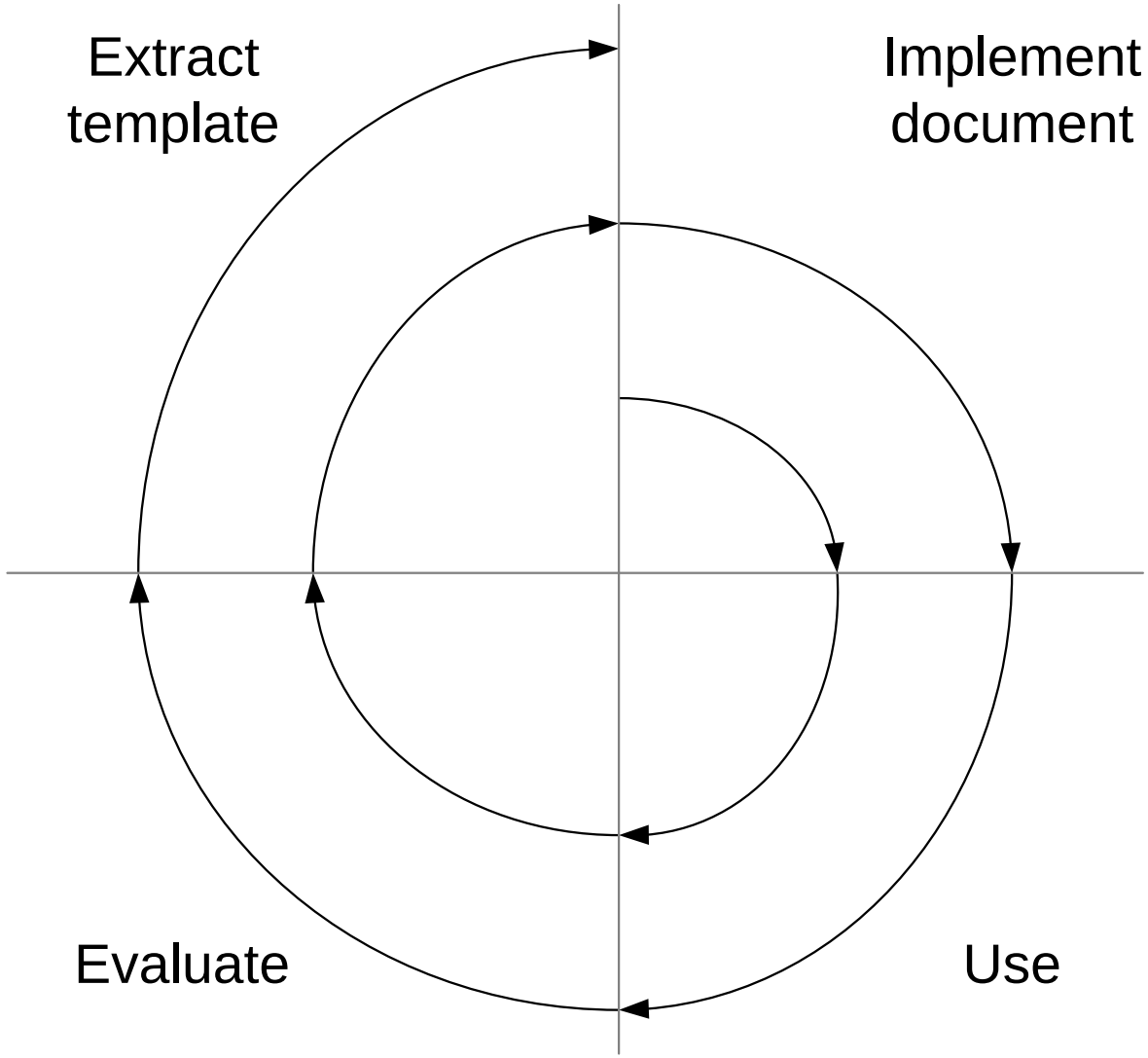
- Use templates for meta-information.
- Use checklists for structure and contents.

Template Development

Templates are an optimization of the Copy Paste Modify pattern:

- Look for a similar problem
- Copy its implementation
- Modify the copy to fulfil the new requirements

Spiral model: Use before Re-use



Example Guidelines Meta Information(1)

Mandatory per page:

- Author
- Title
- Status
- Version
- Date of last update
- Unique Identification
- Business Unit
- Page number

Example Guidelines Meta Information(2)

Mandatory per document:

- Distribution (Notification) list
- Reviewers and commentators
- Document scope (Product family, Product, Subsystem, Module as far as applicable)
- Change history

Example Guidelines Meta Information(3)

Recommended Practice:

- Short statement on frontpage stating what is expected from the addressed recipients, for example:
  - Please send comments before february 29, this document will be reviewed on that date
  - This document is authorized, changes are only applied via a change request
- See Granularity of Documentation [**?**] for guidelines for modularization and contents

Template Pitfalls

- Author follows template instead of considering the purpose of the document.
- Template is too complex.
- There is an unmanageable number of variants.
- Mandatory use of templates results in:
  - no innovation of templates (= no learning)
  - no common sense in deployment
  - strong dependency on templates

Recommendation:

- Enforce the procedure *(what)*
- Provide the template *(how)* as supporting means.

Summary

- Templates support (new) processes
- Use templates for layout and meta information support
- Do not use templates for documents structure or contents
- Stimulate evolution of templates, keep them alive
- Keep templates simple
- Standardize on **what** (process or procedure), not on **how** (tool and template)
- Provide (mandatory) guidelines and recommended practices
- Provide templates as a supportive choice, don't force people to use templates

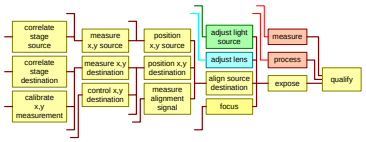
System Integration How-To

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

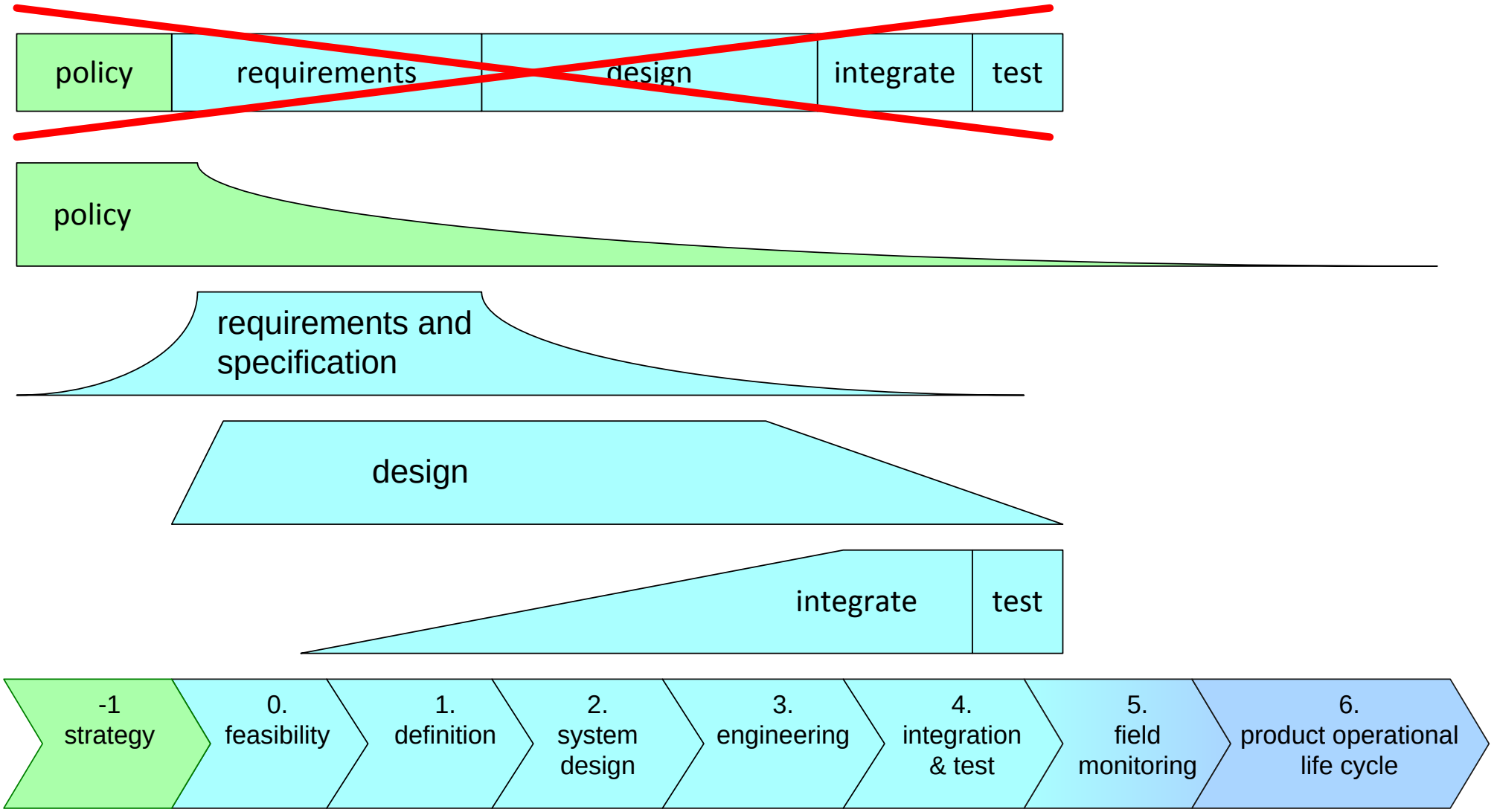
Abstract

In this document we will discuss the full integration flow. We will discuss the goal of integration, the relation between integration and testing, what is integration and how to integrate, an approach to integration, scheduling and dealing with disruptive events, roles and responsibilities, configuration management aspects, and typical order of integration problems occurring in real life.

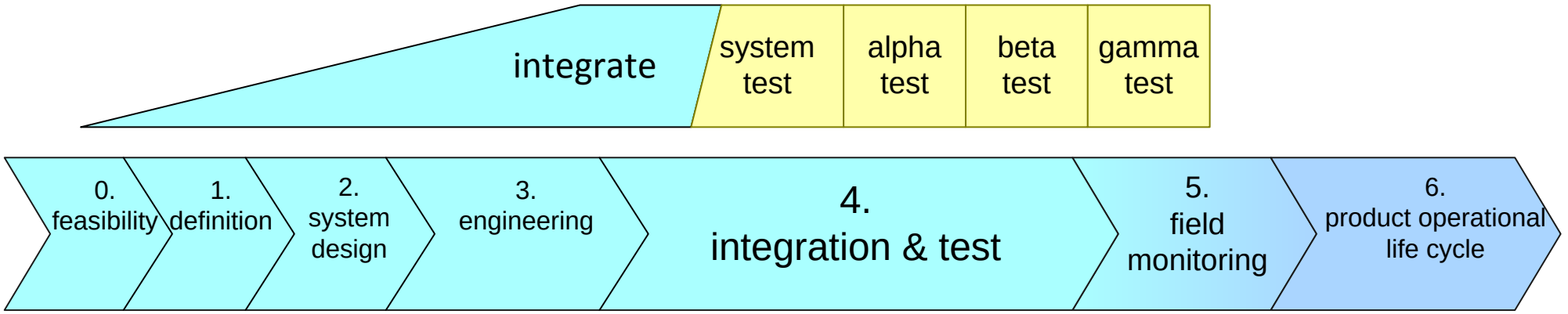
March 27, 2021
status: concept
version: 0.2



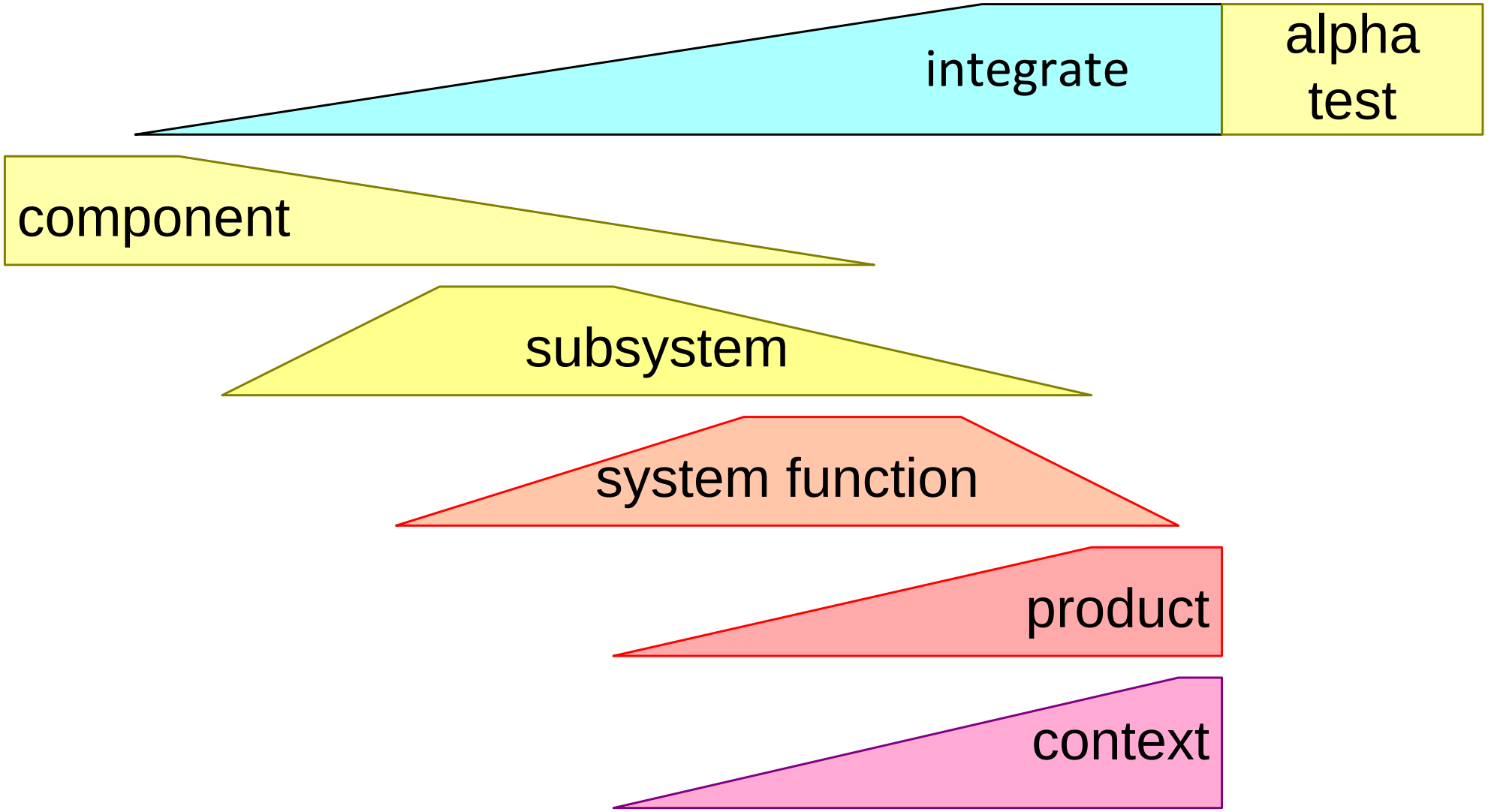
Typical Concurrent Product Creation Process



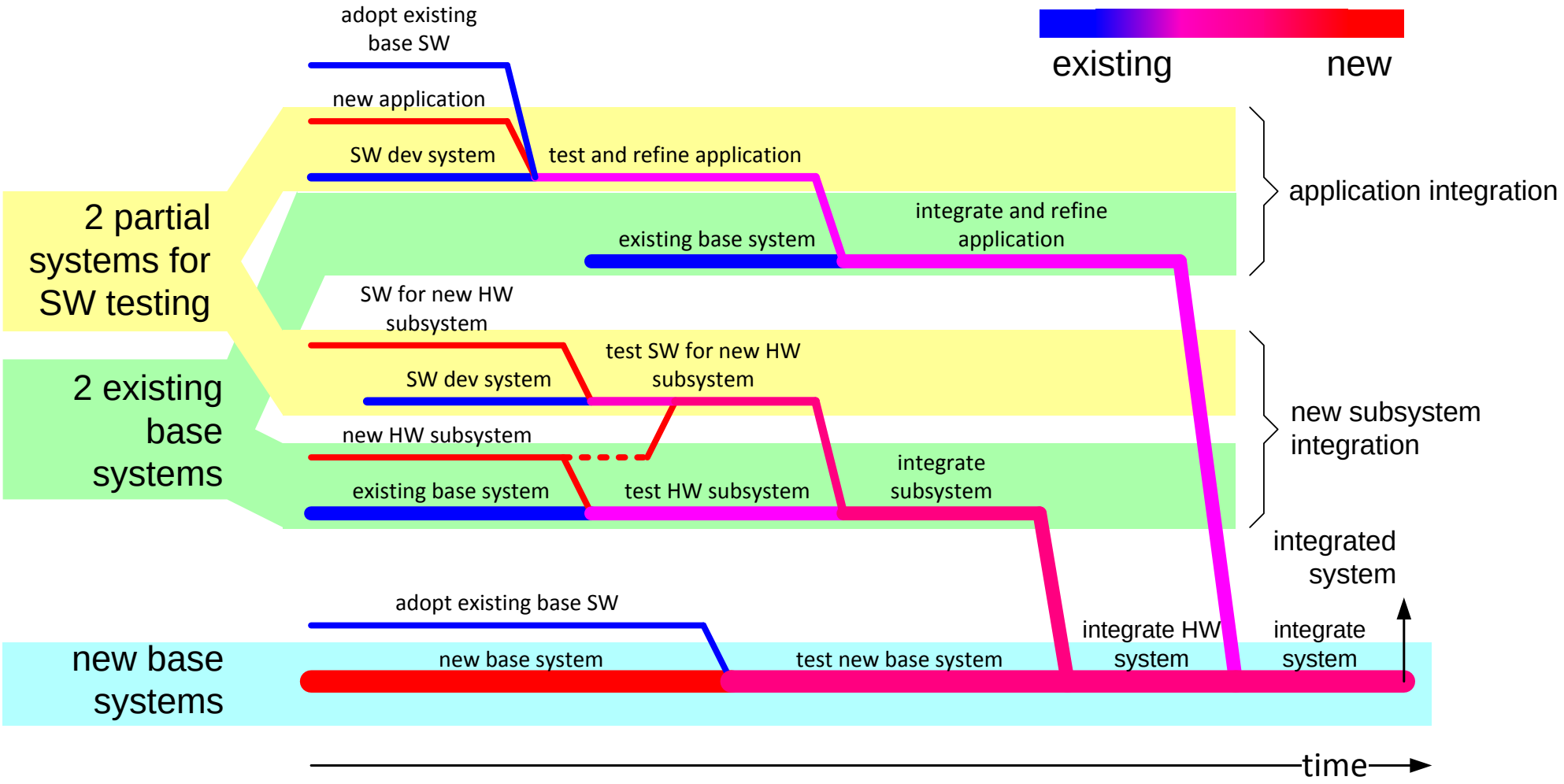
Zooming in on Integration and Tests



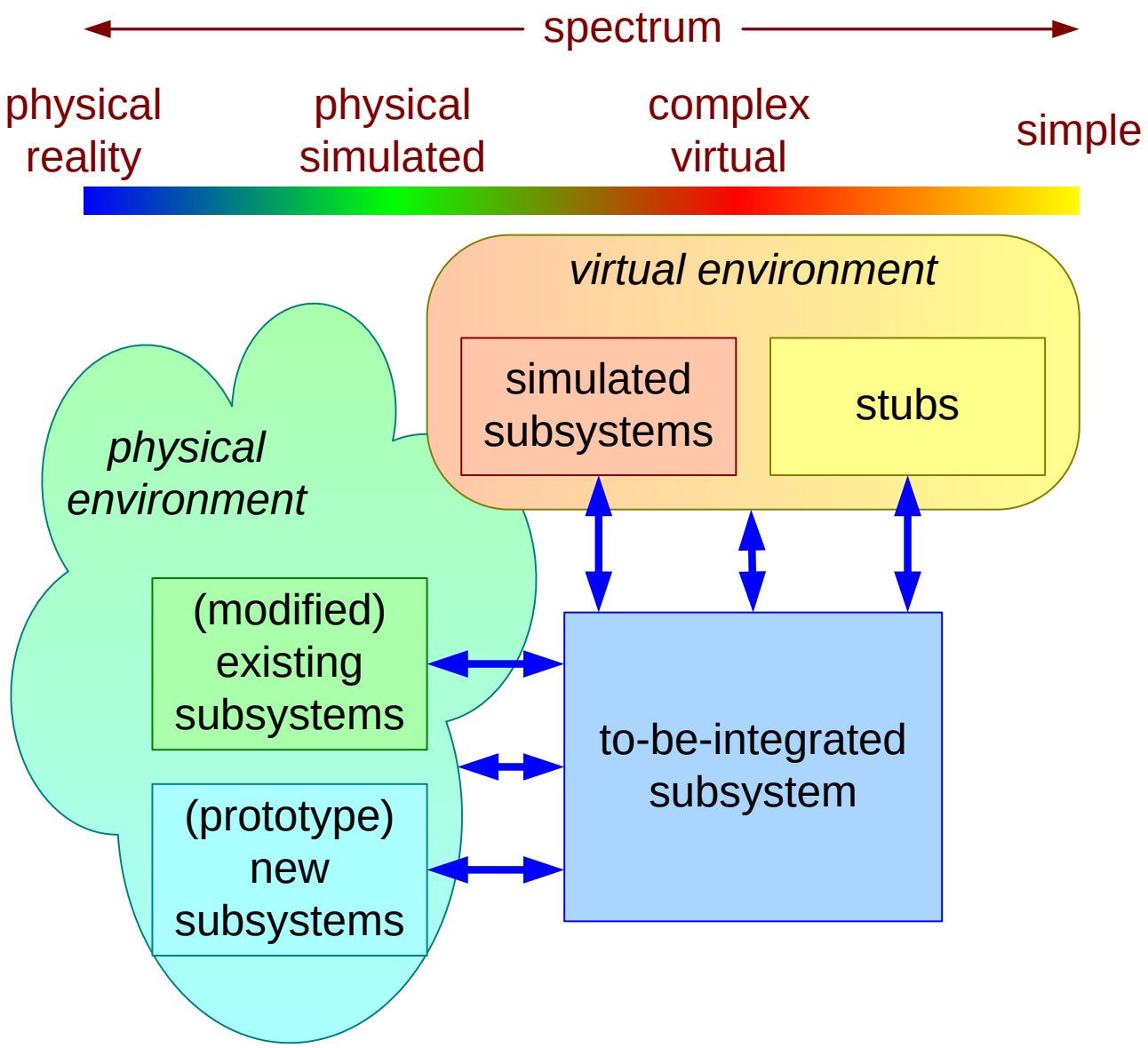
Integration Takes Place in a Bottom-up Fashion



Transition from Previous System to New System



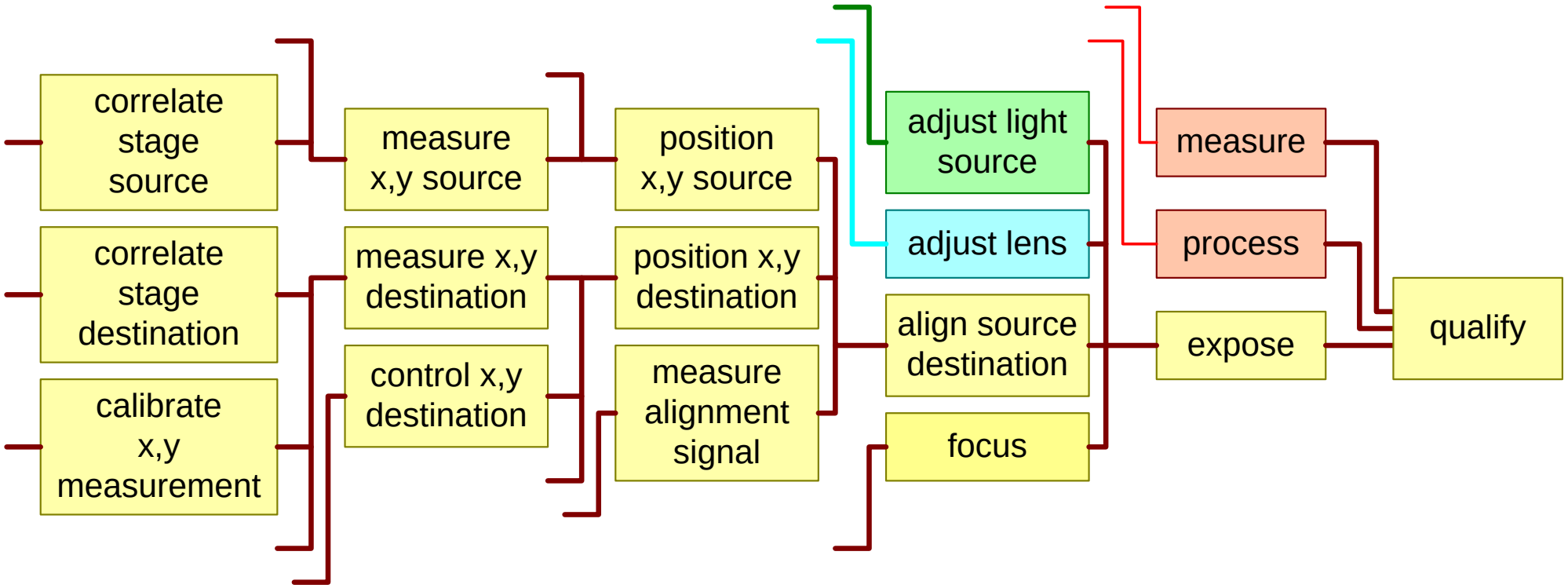
Alternatives to Integrate a Subsystem Early in the Project



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.

Order of Functions Required for the IQ of a Waferstepper



Roles and Responsibilities During the Integration Process

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

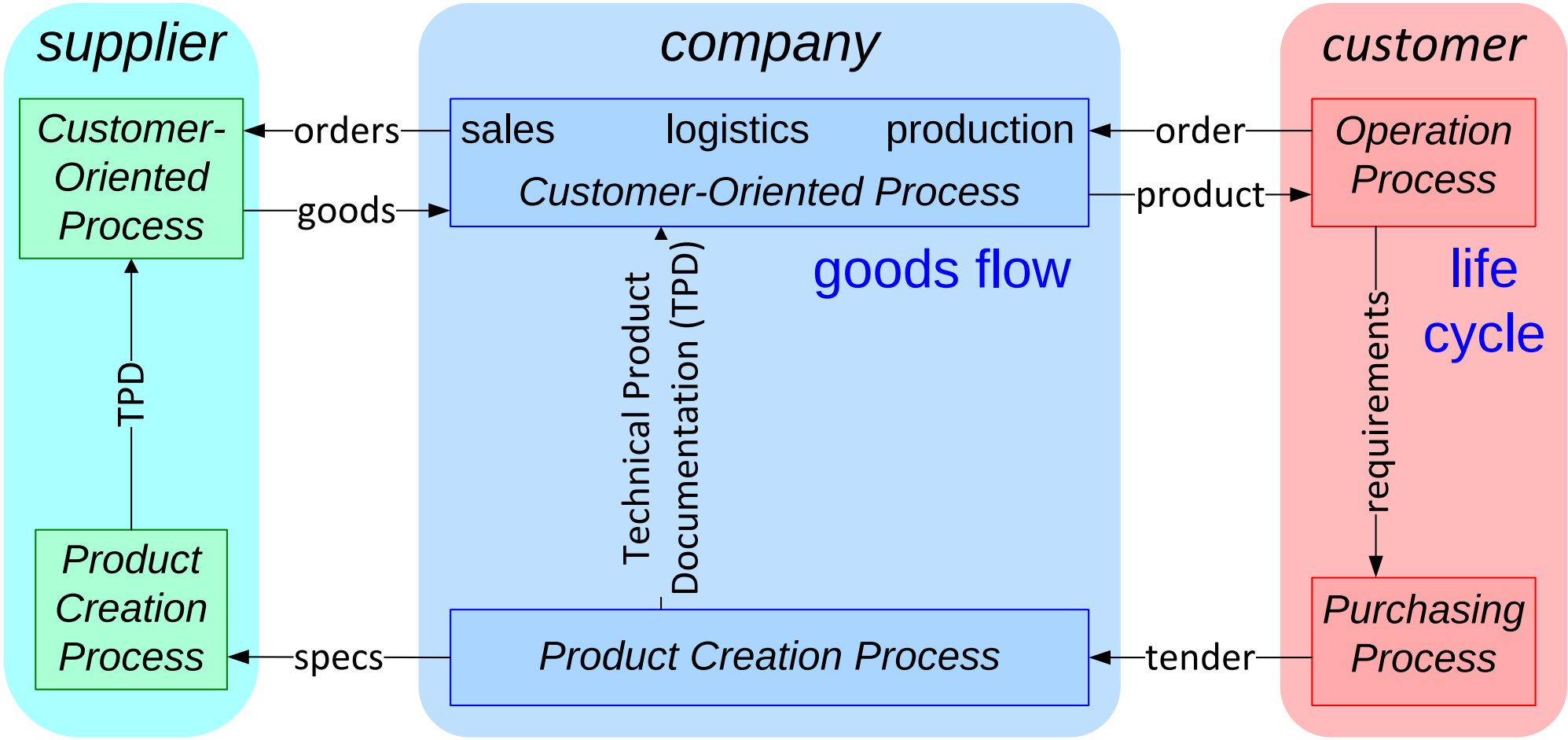
engineers

design
component test
troubleshooting
participate in test

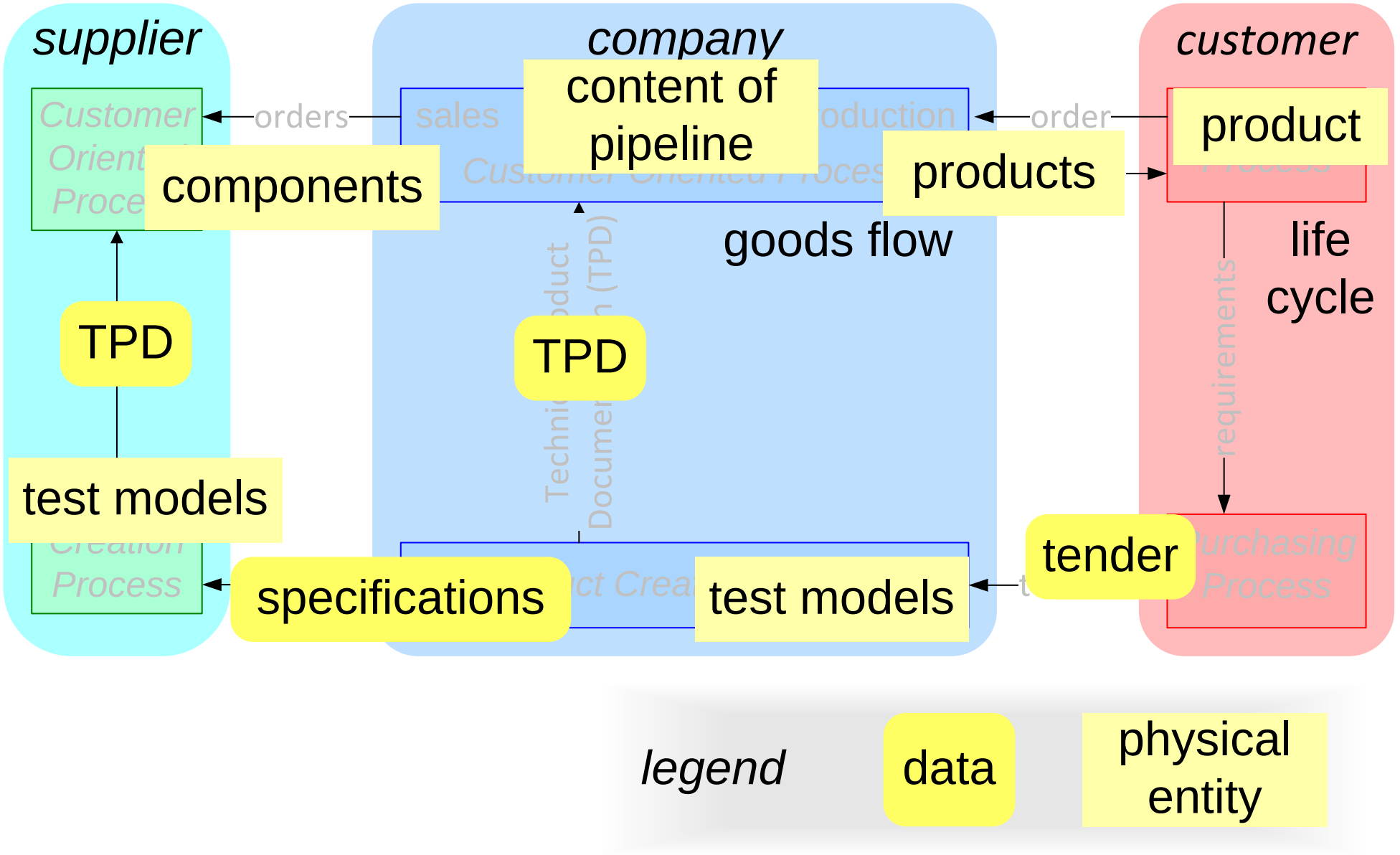
machine owner

maintain test model
support test

Simplified Process Diagram



Configuration Management Entities



Typical Order of Integration Problems

1. The (sub)system does not build.
2. The (sub)system does not function.
3. Interface errors.
4. The (sub)system is too slow.
5. Problems with the main performance parameter, such as image quality.
6. The (sub)system is not reliable.

Exercise Documentation

Make a design for the documentation structure of the case, take into account a.o.:

- target audience per documentation module
- lifecycle
- author
- size (budget)

Present (max 1 flip) the proposed documentation structure and the rationale.

Documentation

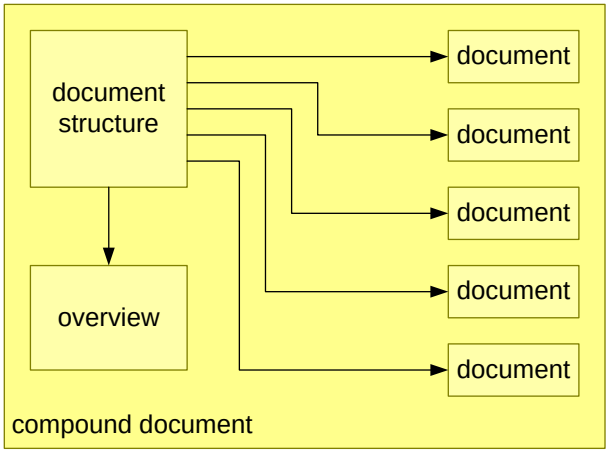
Requirements Entire Documentation

- Accessibility for the readers
- Low threshold for the readers
- Low threshold for the authors
- Completeness
- Consistency
- Maintainability
- Scalability
- Evolvability
- Process to ensure the quality of the information

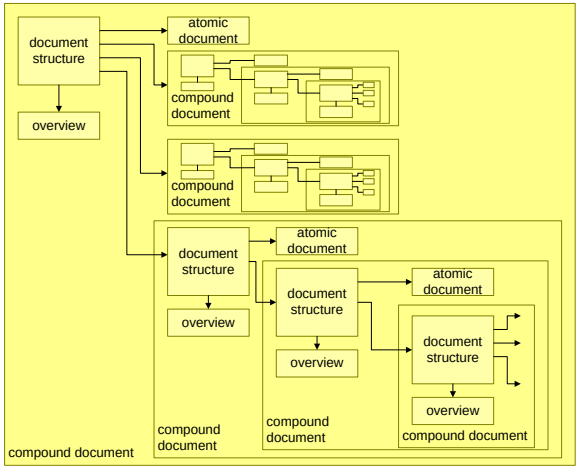
Requirements per Document

- High cohesion (within the unit)
- Low coupling (outside of the unit)
- Accessibility for the readers
- Low threshold for the reader
- Low threshold for the author
- Manageable steps to create, review, and change
- Clear responsibilities
- Clear position and relation with the context
- Well-defined status of the information
- Timely availability

Decompose Large Documents



Recursive Decomposition



Documentation

Maximize Payload

front page

title identification author distribution status review	history changes	diagrams tables	1. aap 2. noot 3. mies lists and ca 50% text

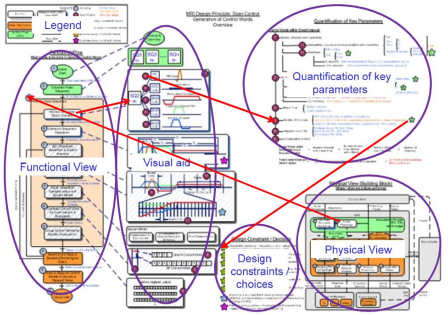
meta information max 2 pages ← → contents 2..18 pages

A3s

multiple related views — quantifications

one topic per A3

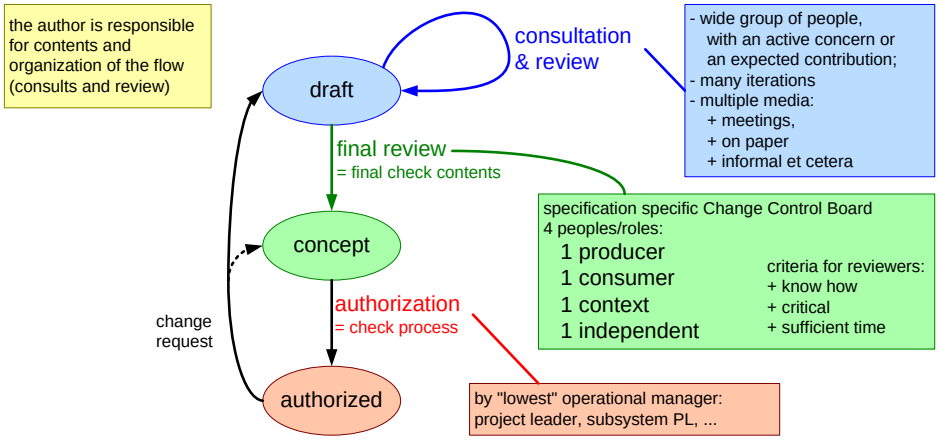
capture "hot" topics



source: PhD thesis Daniel Borches http://doc.utwente.nl/75284/

digestable (size limitation) — practical close to stakeholder experience

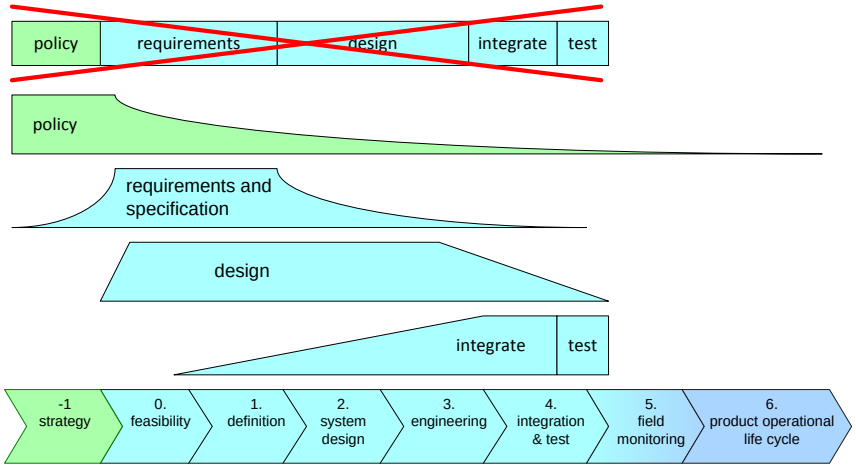
Light Weight Review



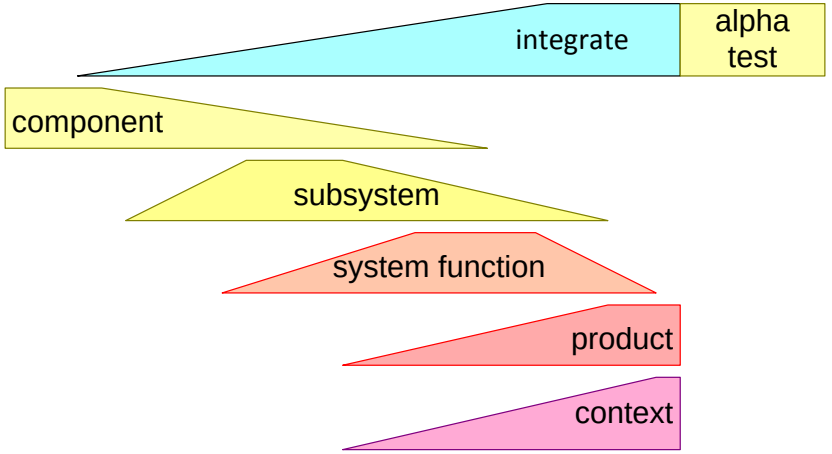
intentionally left blank

Systems Integration

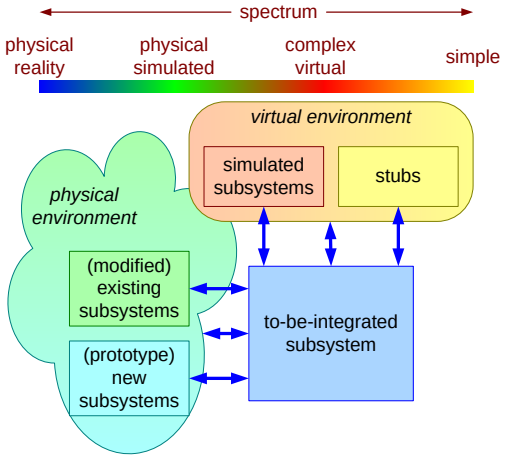
Integration Starts at Feasibility



Bottom-up



Alternatives for Early Integration



Propagation of Configuration Issues