System Engineering (at ESO)

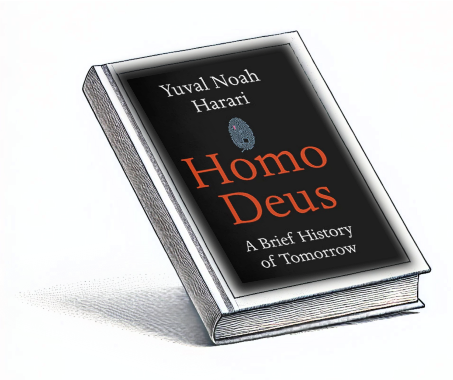
Christian Schmid[1]

[1] European Southern Observatory
Karl-Schwarzschild-Str. 2, 85748 Garching, Germany

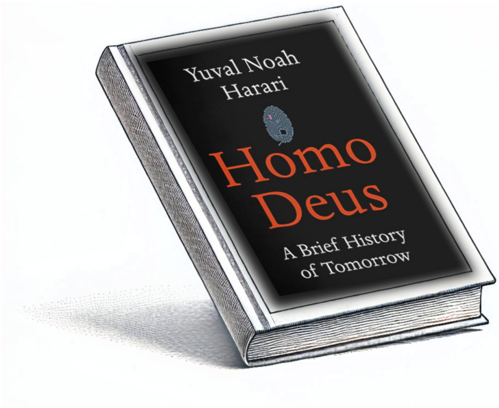
1. System Engineering in General
2. System Engineering at ESO

System Engineering in General

Overcoming the Limit



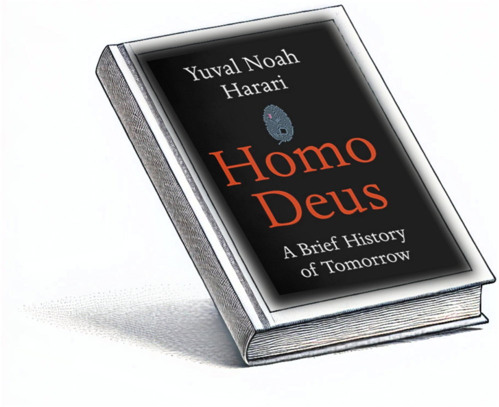
Overcoming the Limit



Distinguishing feature enabling Homo Sapiens to dominate all other animals?

⇒ Only species capable of co-operating flexibly in large numbers.

Overcoming the Limit

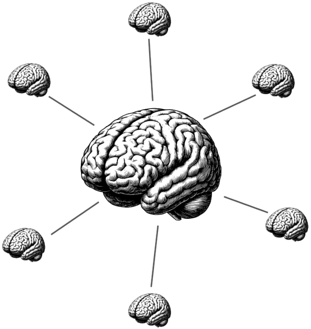


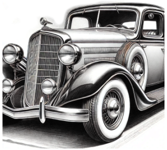
Distinguishing feature enabling Homo Sapiens to dominate all other animals?

⇒ Only species capable of co-operating flexibly in large numbers.

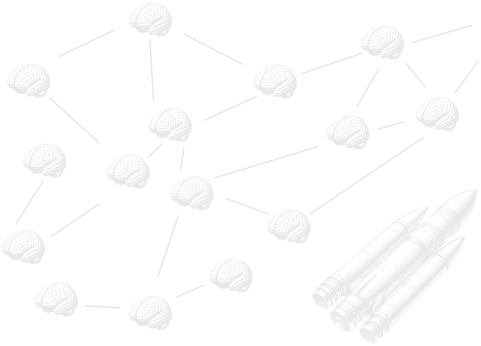
Overcoming Complexity

Chief Engineer / ”Craftsman” Approach



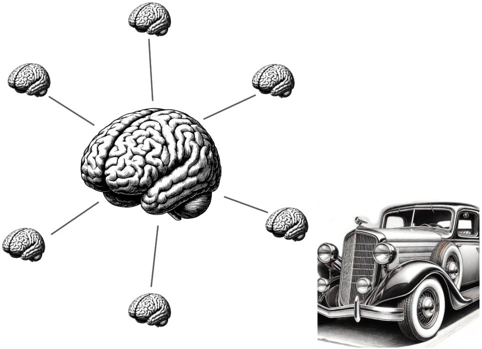


System Engineering Approach

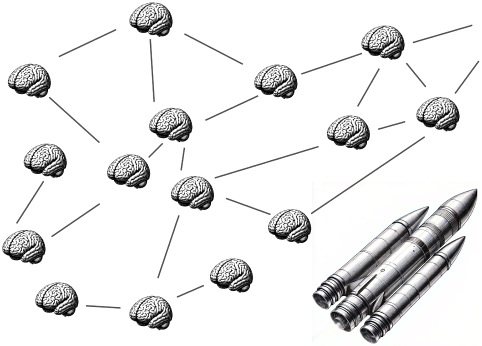


Overcoming Complexity

Chief Engineer / ”Craftsman” Approach



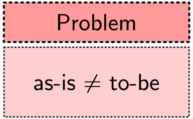
System Engineering Approach



Systems Engineering Concept Haberfellner/Daenzer ETH

Problem

Systems Engineering Concept Haberfellner/Daenzer ETH



Problem

as-is ≠ to-be

Systems Engineering Concept Haberfellner/Daenzer ETH

Problem

as-is $\neq$ to-be

Solution

Systems Engineering Concept Haberfellner/Daenzer ETH

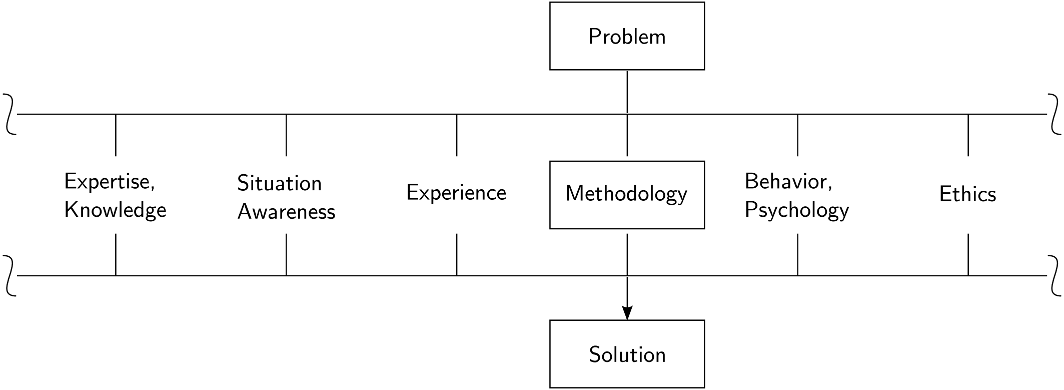
Problem

as-is $\neq$ to-be

Solution

as-is $=$ to-be

Systems Engineering Concept Haberfellner/Daenzer ETH



Problem

Expertise, Knowledge

Situation Awareness

Experience

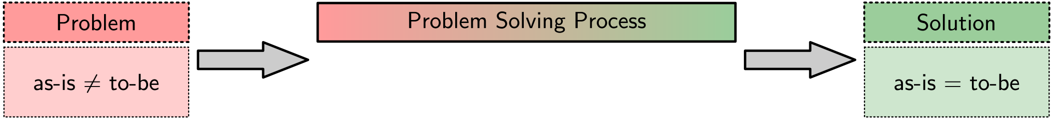
Methodology

Behavior, Psychology

Ethics

Solution

Systems Engineering Concept Haberfellner/Daenzer ETH



Problem

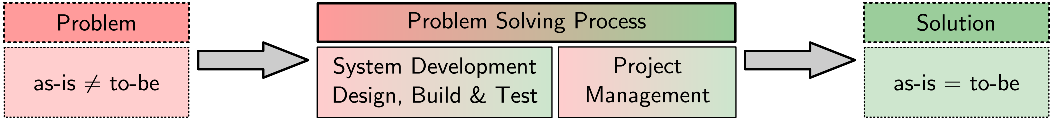
as-is ≠ to-be

Problem Solving Process

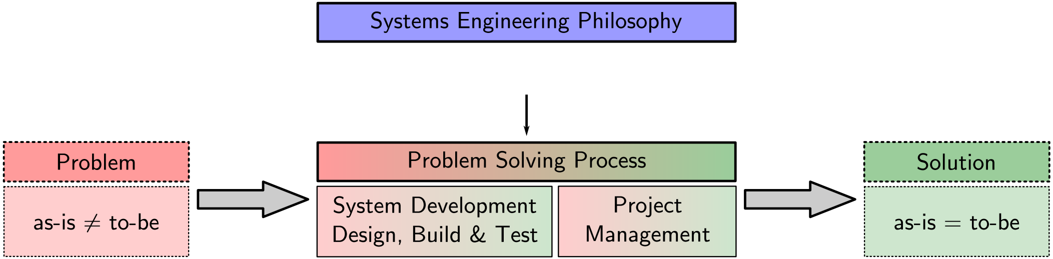
Solution

as-is = to-be

Systems Engineering Concept Haberfellner/Daenzer ETH



Systems Engineering Concept Haberfellner/Daenzer ETH



Systems Engineering Philosophy

Problem

as-is ≠ to-be

Problem Solving Process

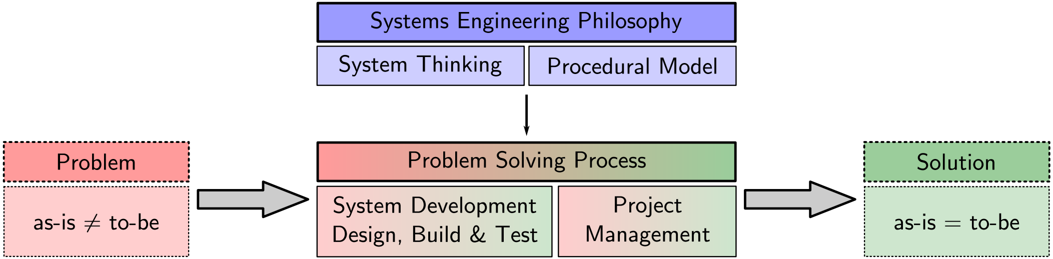
System Development Design, Build & Test

Project Management

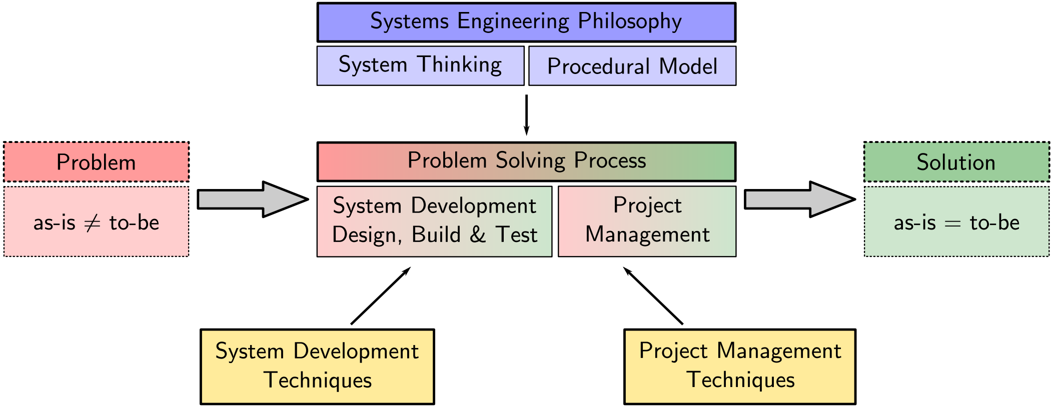
Solution

as-is = to-be

Systems Engineering Concept Haberfellner/Daenzer ETH



Systems Engineering Concept Haberfellner/Daenzer ETH



Systems Engineering Philosophy

System Thinking

Procedural Model

Problem

as-is $\neq$ to-be

Problem Solving Process

System Development Design, Build & Test

Project Management

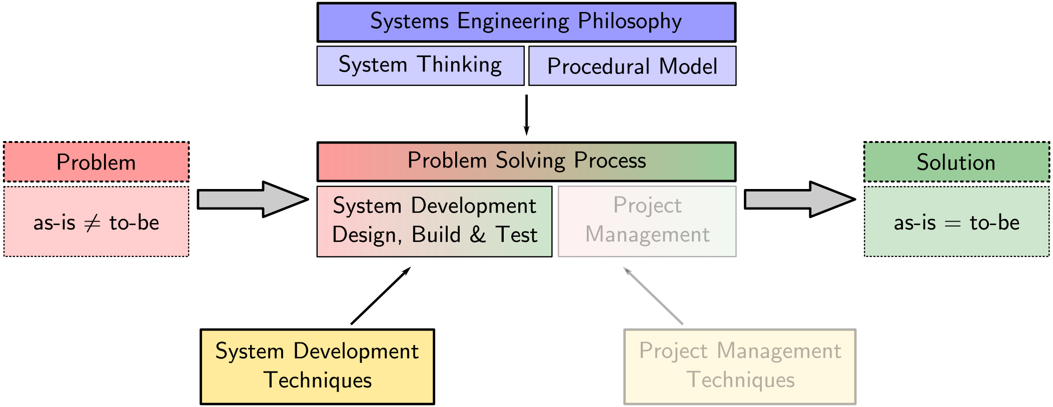
Solution

as-is $=$ to-be

System Development Techniques

Project Management Techniques

Systems Engineering Concept Haberfellner/Daenzer ETH


Systems Engineering Philosophy

System Thinking

Procedural Model

Problem

as-is ≠ to-be

Problem Solving Process

System Development Design, Build & Test

Project Management

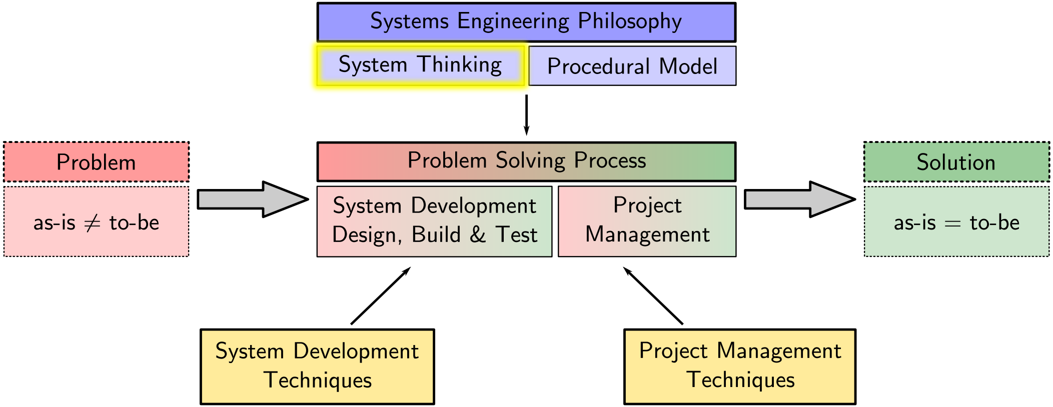
Solution

as-is = to-be

System Development Techniques

Project Management Techniques

Systems Engineering Philosophy Systems Thinking


Systems Engineering Philosophy
System Thinking
Procedural Model
Problem
as-is $\neq$ to-be
Problem Solving Process
System Development Design, Build & Test
Project Management
Solution
as-is $=$ to-be
System Development Techniques
Project Management Techniques

Systems Thinking

A holistic approach to decompose a given entity (system) into structural and functional parts according to their mutual interdependencies and environmental interactions.

Function

- What a system does
- Generates behavior
- Source of benefits
- Needs structure/ elements of form

Structure/Form

- What a system is
- Is dis-/assembled
- Source of costs
- Enables the function

$$\text{Value} = \frac{\text{Benefits}}{\text{Costs}}$$

⇒ related to Emergence

Systems Thinking

A holistic approach to decompose a given entity (system) into structural and functional parts according to their mutual interdependencies and environmental interactions.

Function

- What a system does
- Generates behavior
- Source of benefits
- Needs structure/ elements of form

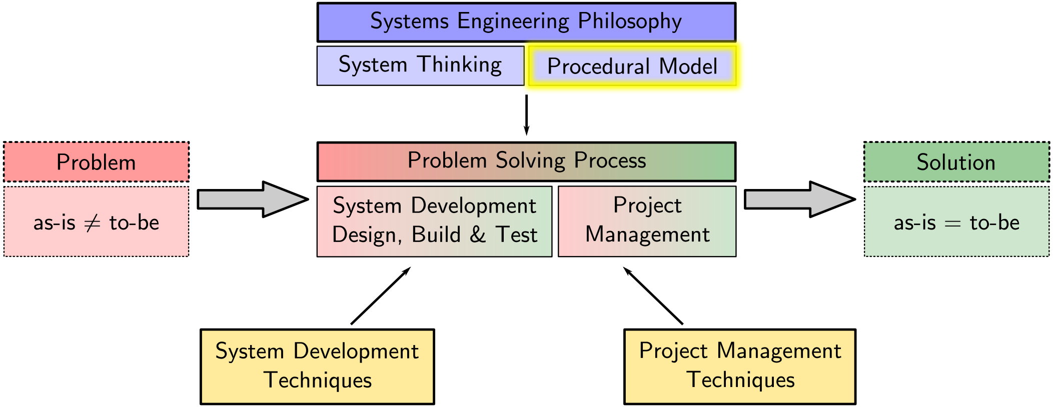
Structure/Form

- What a system is
- Is dis-/assembled
- Source of costs
- Enables the function

$$\text{Value} = \frac{\text{Benefits}}{\text{Costs}}$$

⇒ related to Emergence

Systems Engineering Philosophy Procedural Model



Systems Engineering Philosophy
System Thinking
Procedural Model
Problem
as-is $\neq$ to-be
Problem Solving Process
System Development Design, Build & Test
Project Management
Solution
as-is $=$ to-be
System Development Techniques
Project Management Techniques

Systems Engineering Philosophy   Procedural Model

⇒ Macro (high-level) logic on how to structure the problem solving process

- Plan-driven Models
  - Waterfall
  - Phase Models (A, B, C, D / CD, PD, FD, Build)
  - Spiral
  - etc.
- Agile Models (not well established in SE, rather SW projects)

  → **Note:** Agile $\langle$ Systems Engineering $\rangle \neq \langle$ Agile Systems $\rangle$ Engineering

⇒ Typically adapted and tailored to the specifics of the project.

Systems Engineering Philosophy   Procedural Model

⇒ Macro (high-level) logic on how to structure the problem solving process

- Plan-driven Models
  - Waterfall
  - Phase Models (A, B, C, D / CD, PD, FD, Build)
  - Spiral
  - etc.
- Agile Models (not well established in SE, rather SW projects)

  → **Note:** Agile $\langle$ Systems Engineering $\rangle \neq \langle$ Agile Systems $\rangle$ Engineering

⇒ Typically adapted and tailored to the specifics of the project.

Systems Engineering Philosophy Procedural Model

⇒ Macro (high-level) logic on how to structure the problem solving process

- Plan-driven Models
  - Waterfall
  - Phase Models (A, B, C, D / CD, PD, FD, Build)
  - Spiral
  - etc.
- Agile Models (not well established in SE, rather SW projects)

  → **Note:** Agile $\langle$ Systems Engineering $\rangle \neq \langle$ Agile Systems $\rangle$ Engineering

⇒ Typically adapted and tailored to the specifics of the project.

Systems Engineering Philosophy — Procedural Model

⇒ Macro (high-level) logic on how to structure the problem solving process

- Plan-driven Models
  - Waterfall
  - Phase Models (A, B, C, D / CD, PD, FD, Build)
  - Spiral
  - etc.
- Agile Models (not well established in SE, rather SW projects)

  → **Note:** Agile ⟨ Systems Engineering ⟩ ≠ ⟨ Agile Systems ⟩ Engineering

⇒ Typically adapted and tailored to the specifics of the project.

Systems Engineering Philosophy Procedural Model

⇒ Macro (high-level) logic on how to structure the problem solving process

- Plan-driven Models
  - Waterfall
  - Phase Models (A, B, C, D / CD, PD, FD, Build)
  - Spiral
  - etc.
- Agile Models (not well established in SE, rather SW projects)

  → **Note:** Agile $\langle$ Systems Engineering $\rangle \neq \langle$ Agile Systems $\rangle$ Engineering

⇒ Typically adapted and tailored to the specifics of the project.

Systems Engineering Philosophy   Procedural Model

⇒ Macro (high-level) logic on how to structure the problem solving process

- Plan-driven Models
  - Waterfall
  - Phase Models (A, B, C, D / CD, PD, FD, Build)
  - Spiral
  - etc.
- Agile Models (not well established in SE, rather SW projects)

  → **Note:** Agile ⟨ Systems Engineering ⟩ ≠ ⟨ Agile Systems ⟩ Engineering

⇒ Typically adapted and tailored to the specifics of the project.

Systems Engineering Philosophy Procedural Model

⇒ Macro (high-level) logic on how to structure the problem solving process

- Plan-driven Models
  - Waterfall
  - Phase Models (A, B, C, D / CD, PD, FD, Build)
  - Spiral
  - etc.
- Agile Models (not well established in SE, rather SW projects)

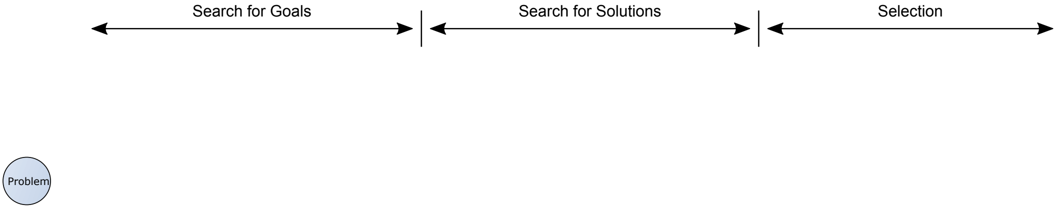
  → **Note:** Agile ⟨ Systems Engineering ⟩ ≠ ⟨ Agile Systems ⟩ Engineering

⇒ Typically adapted and tailored to the specifics of the project.

Fundamental Problem Solving Logic

⇒ Micro (low-level) logic repeatedly applied within stages of the Procedural Model

Fundamental Problem Solving Logic



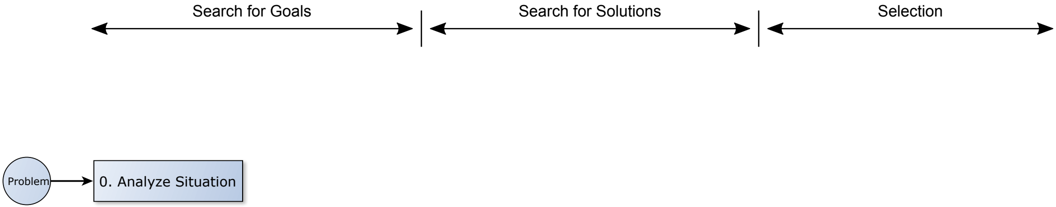
Search for Goals

Search for Solutions

Selection

Problem

Fundamental Problem Solving Logic



Search for Goals

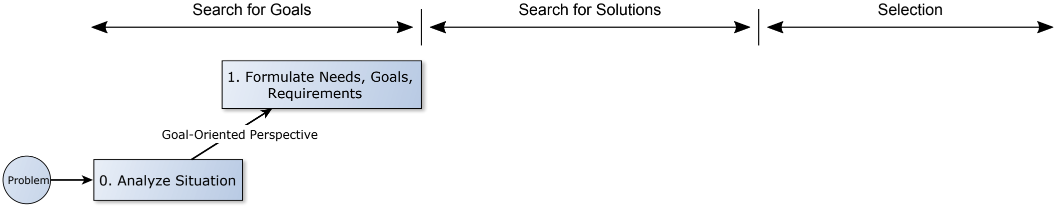
Search for Solutions

Selection

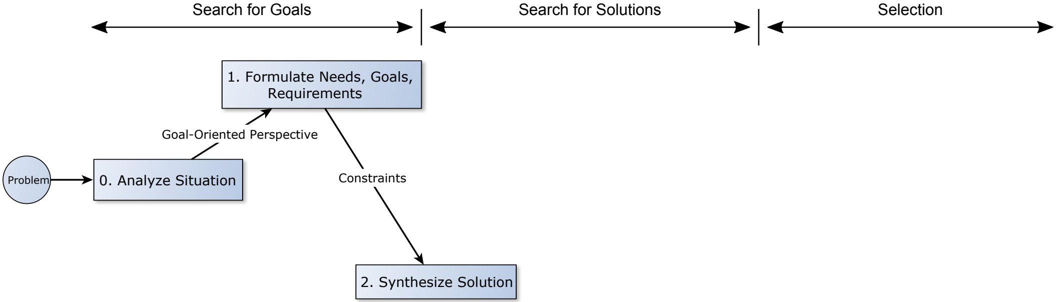
Problem

0. Analyze Situation

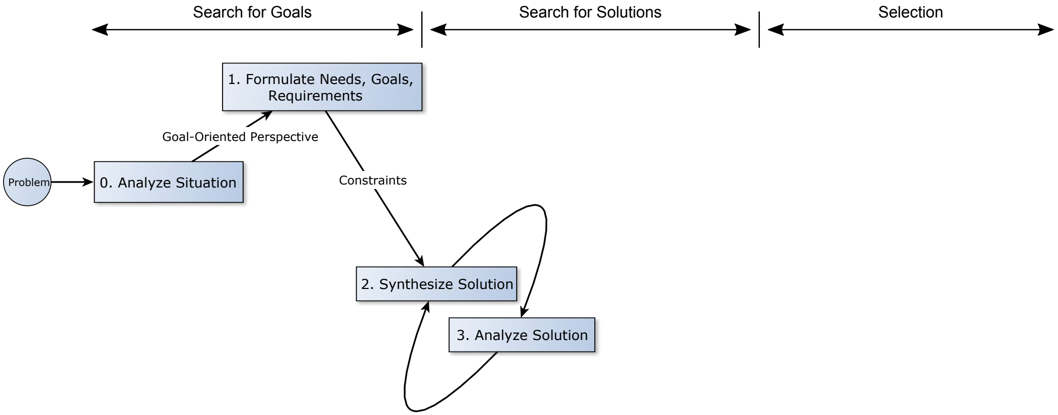
Fundamental Problem Solving Logic



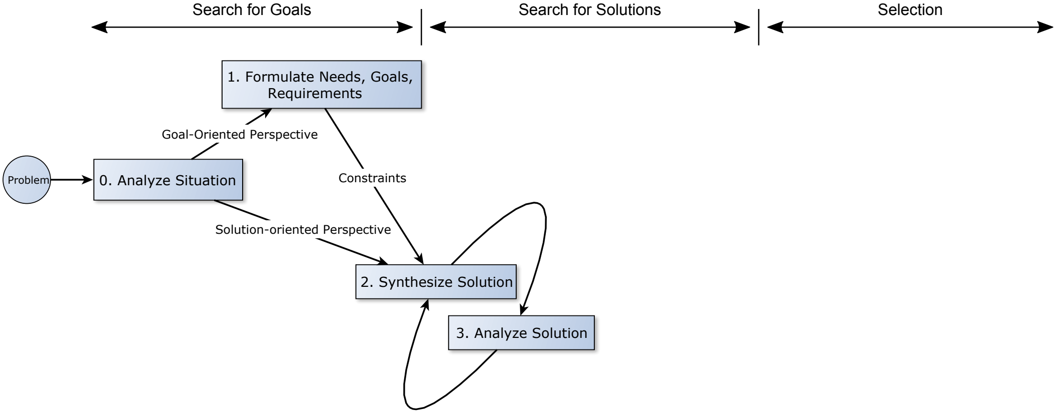
Fundamental Problem Solving Logic



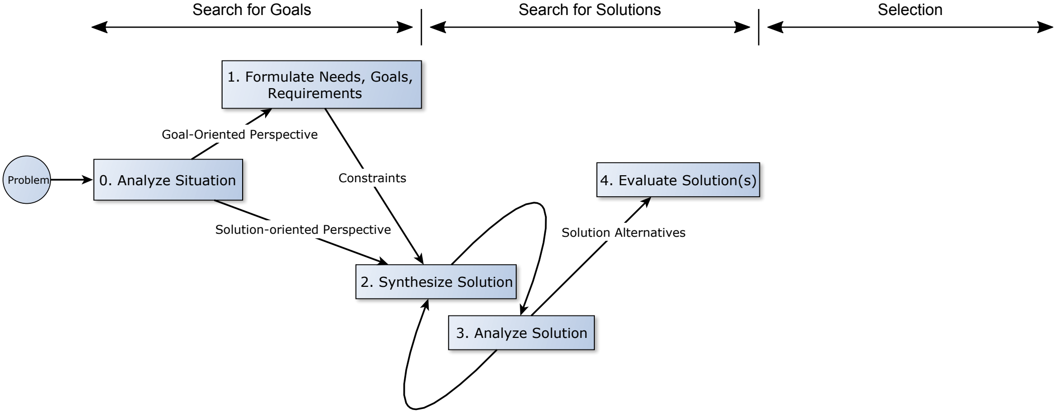
Fundamental Problem Solving Logic



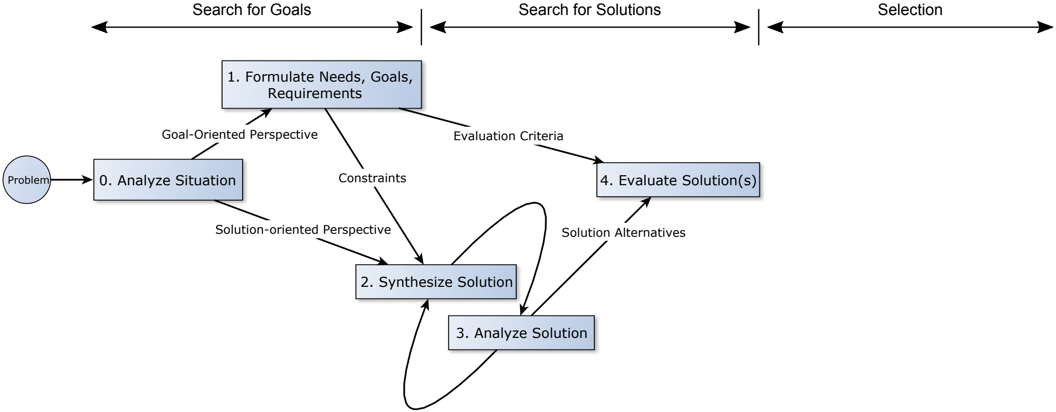
Fundamental Problem Solving Logic



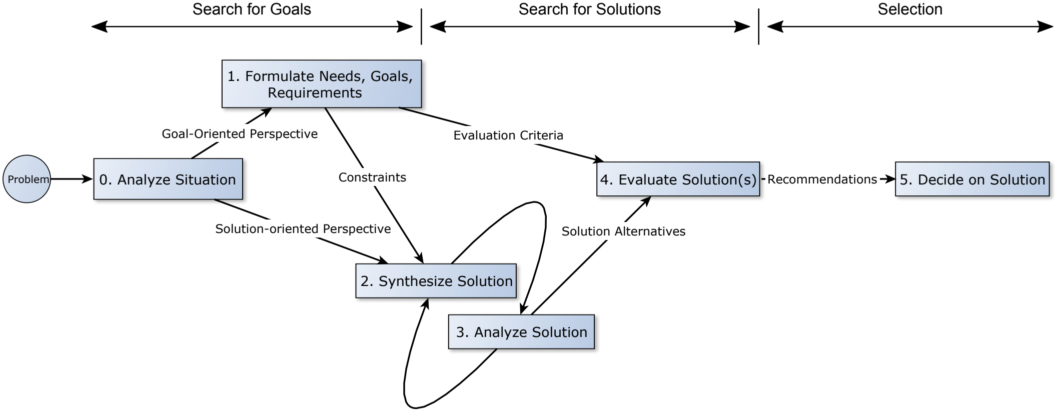
Fundamental Problem Solving Logic



Fundamental Problem Solving Logic

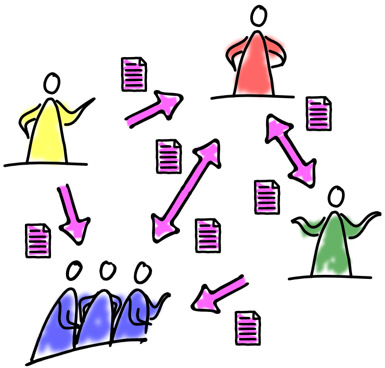


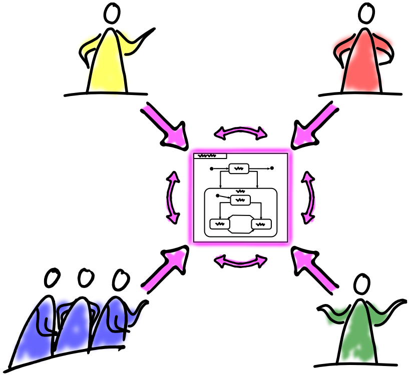
Fundamental Problem Solving Logic



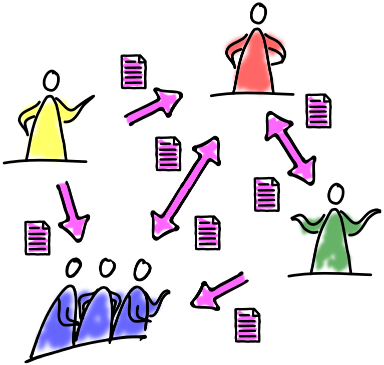
System Engineering at ESO

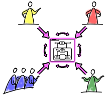
Document-Centric vs Model Based Systems Engineering





Document-Centric at ESO





Procedural Model

Project Phases at ESO → Europe

Pre-Phase A

- First studies of science cases

→ Phase A proposal

Phase A
Feasibility Study

- Consolidate science cases

→ TLR, TRS

→ System Architecture

Phase B
Concept Design

- Finalize Architecture

→ Subsystem TRS

→ Perf. Budgets

Phase C
Final Design

- Detailed Design
- Verification activities

→ Lower level qualification testing and manufacturing

Phase D
System Build

- System AIT

→ Preliminary Acceptance Europe

Procedural Model

Project Phases at ESO → Europe

Pre-Phase A

- First studies of science cases

→ Phase A proposal

Phase A
Feasibility Study

- Consolidate science cases

→ TLR, TRS

→ System Architecture

Phase B
Concept Design

- Finalize Architecture

→ Subsystem TRS

→ Perf. Budgets

Phase C
Final Design

- Detailed Design
- Verification activities

→ Lower level qualification testing and manufacturing

Phase D
System Build

- System AIT

→ Preliminary Acceptance Europe

Procedural Model

Project Phases at ESO → Europe

Pre-Phase A

- First studies of science cases

→ Phase A proposal

Phase A
Feasibility Study

- Consolidate science cases

→ TLR, TRS

→ System Architecture

Phase B
Concept Design

- Finalize Architecture

→ Subsystem TRS

→ Perf. Budgets

Phase C
Final Design

- Detailed Design
- Verification activities

→ Lower level qualification testing and manufacturing

Phase D
System Build

- System AIT

→ Preliminary Acceptance Europe

Procedural Model

Project Phases at ESO → Europe

Pre-Phase A

- First studies of science cases

→ Phase A proposal

Phase A
Feasibility Study

- Consolidate science cases

→ TLR, TRS

→ System Architecture

Phase B
Concept Design

- Finalize Architecture

→ Subsystem TRS

→ Perf. Budgets

Phase C
Final Design

- Detailed Design
- Verification activities

→ Lower level qualification testing and manufacturing

Phase D
System Build

- System AIT

→ Preliminary Acceptance Europe

Procedural Model Project Phases at ESO → Europe

Pre-Phase A

- First studies of science cases

→ Phase A proposal

Phase A
Feasibility Study

- Consolidate science cases

→ TLR, TRS

→ System Architecture

Phase B
Concept Design

- Finalize Architecture

→ Subsystem TRS

→ Perf. Budgets

Phase C
Final Design

- Detailed Design
- Verification activities

→ Lower level qualification testing and manufacturing

Phase D
System Build

- System AIT

→ Preliminary Acceptance Europe

Procedural Model

Project Phases at ESO → Europe

Pre-Phase A

- First studies of science cases

→ Phase A proposal

Phase A
Feasibility Study

- Consolidate science cases

→ TLR, TRS

→ System Architecture

Phase B
Concept Design

- Finalize Architecture

→ Subsystem TRS

→ Perf. Budgets

Phase C
Final Design

- Detailed Design
- Verification activities

→ Lower level qualification testing and manufacturing

Phase D
System Build

- System AIT

→ Preliminary Acceptance Europe

Procedural Model

Project Phases at ESO → Chile

Phase A
Feasibility Study

- Consolidate science cases

→ TLR, TRS

→ System Architecture

Phase B
Concept Design

- Finalize Architecture

→ Subsystem TRS

→ Perf. Budgets

Phase C
Final Design

- Detailed Design
- Verification activities

→ Qualification, testing

Phase D
System Build

- System AIT

→ Preliminary Acceptance Europe

Commissioning

- On-site Assembly, Integration
- On-sky verification

→ Science validation

Procedural Model

Project Phases at ESO → Chile

Phase A
Feasibility Study

- Consolidate science cases
- → TLR, TRS
- → System Architecture

Phase B
Concept Design

- Finalize Architecture
- → Subsystem TRS
- → Perf. Budgets

Phase C
Final Design

- Detailed Design
- Verification activities
- → Qualification, testing

Phase D
System Build

- System AIT
- → Preliminary Acceptance Europe

Commissioning

- On-site Assembly, Integration
- On-sky verification
- → Science validation

Procedural Model

Project Phases at ESO → Chile

Phase B Concept Design

- Finalize Architecture

→ Subsystem TRS

→ Perf. Budgets

Phase C Final Design

- Detailed Design
- Verification activities

→ Qualification, testing

Phase D System Build

- System AIT

→ Preliminary Acceptance Europe

Commissioning

- On-site Assembly, Integration
- On-sky verification

→ Science validation

Phase E Operation

→ Provisional acceptance Chile

Procedural Model

Project Phases at ESO → Chile

Phase B
Concept Design

- Finalize Architecture

→ Subsystem TRS

→ Perf. Budgets

Phase C
Final Design

- Detailed Design
- Verification activities

→ Qualification, testing

Phase D
System Build

- System AIT

→ Preliminary Acceptance Europe

Commissioning

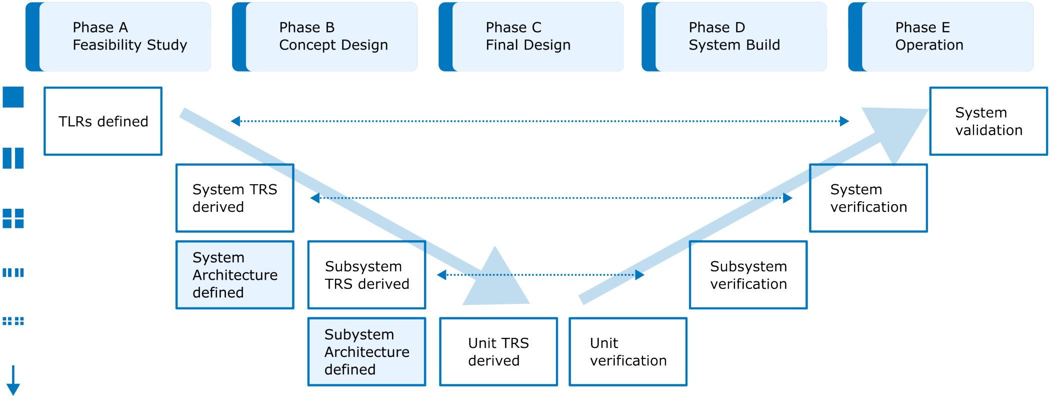
- On-site Assembly, Integration
- On-sky verification

→ Science validation

Phase E
Operation

→ Provisional acceptance Chile

V Model



Phase A
Feasibility Study

Phase B
Concept Design

Phase C
Final Design

Phase D
System Build

Phase E
Operation

TLRs defined

System TRS derived

System Architecture defined

Subsystem TRS derived

Subystem Architecture defined

Unit TRS derived

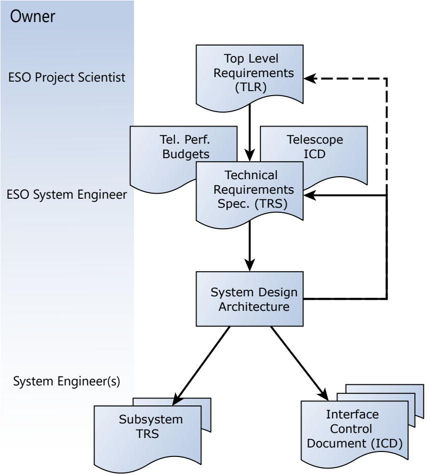
Unit verification

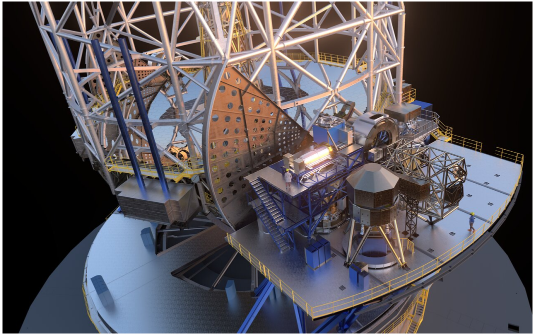
Subsystem verification

System verification

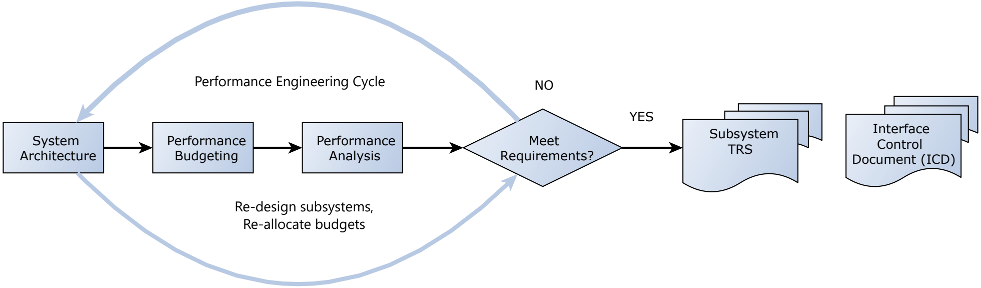
System validation

Involvement (→ Instruments)

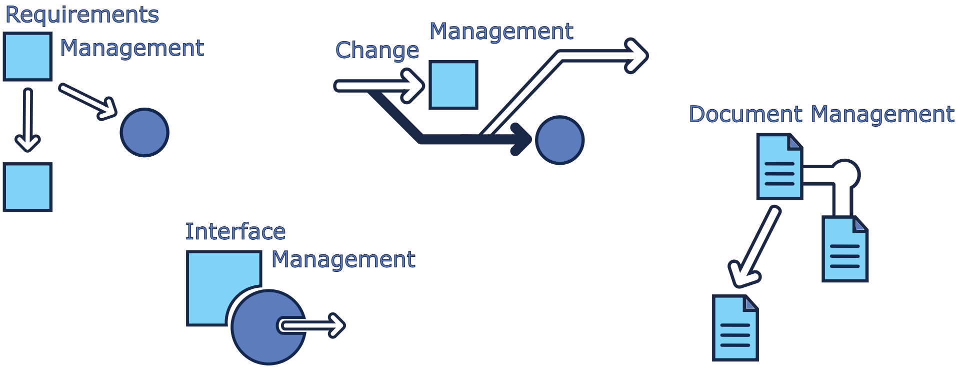

Owner
ESO Project Scientist
ESO System Engineer
System Engineer(s)
Top Level Requirements (TLR)
Tel. Perf. Budgets
Telescope ICD
Technical Requirements Spec. (TRS)
System Design Architecture
Subsystem TRS
Interface Control Document (ICD)



Involvement (→ Instruments)



Key Processes



Requirements Management

Change Management

Document Management

Interface Management