Process modeling

Why, what, how?

Istvan David

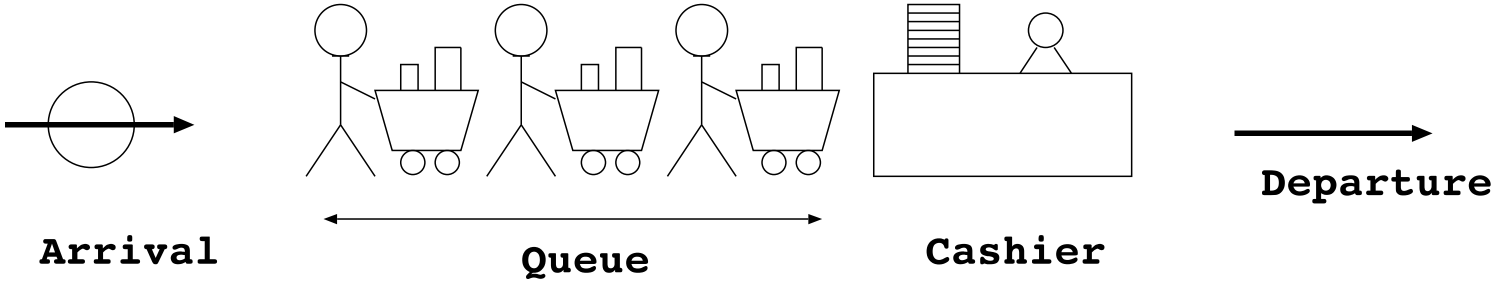
Borrowed material: Joachim Denil (DSM-TP 2015, Antwerp)

25.11.2016, Antwerp

Topics

- Processes in general
- Software processes and notable examples
- Process modeling
  - Why and how?
  - FTG+PM
- Process improvement

Example process



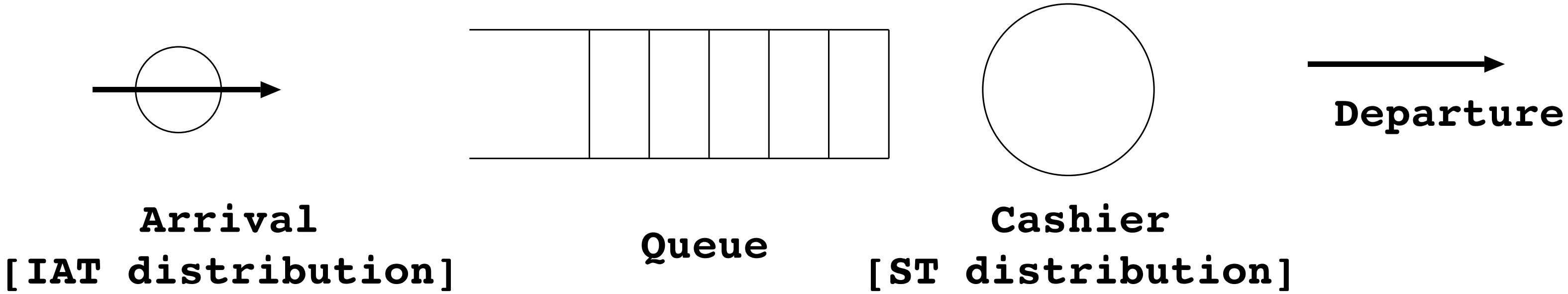
Arrival

Queue

Cashier

Departure

Physical View



Arrival
[IAT distribution]

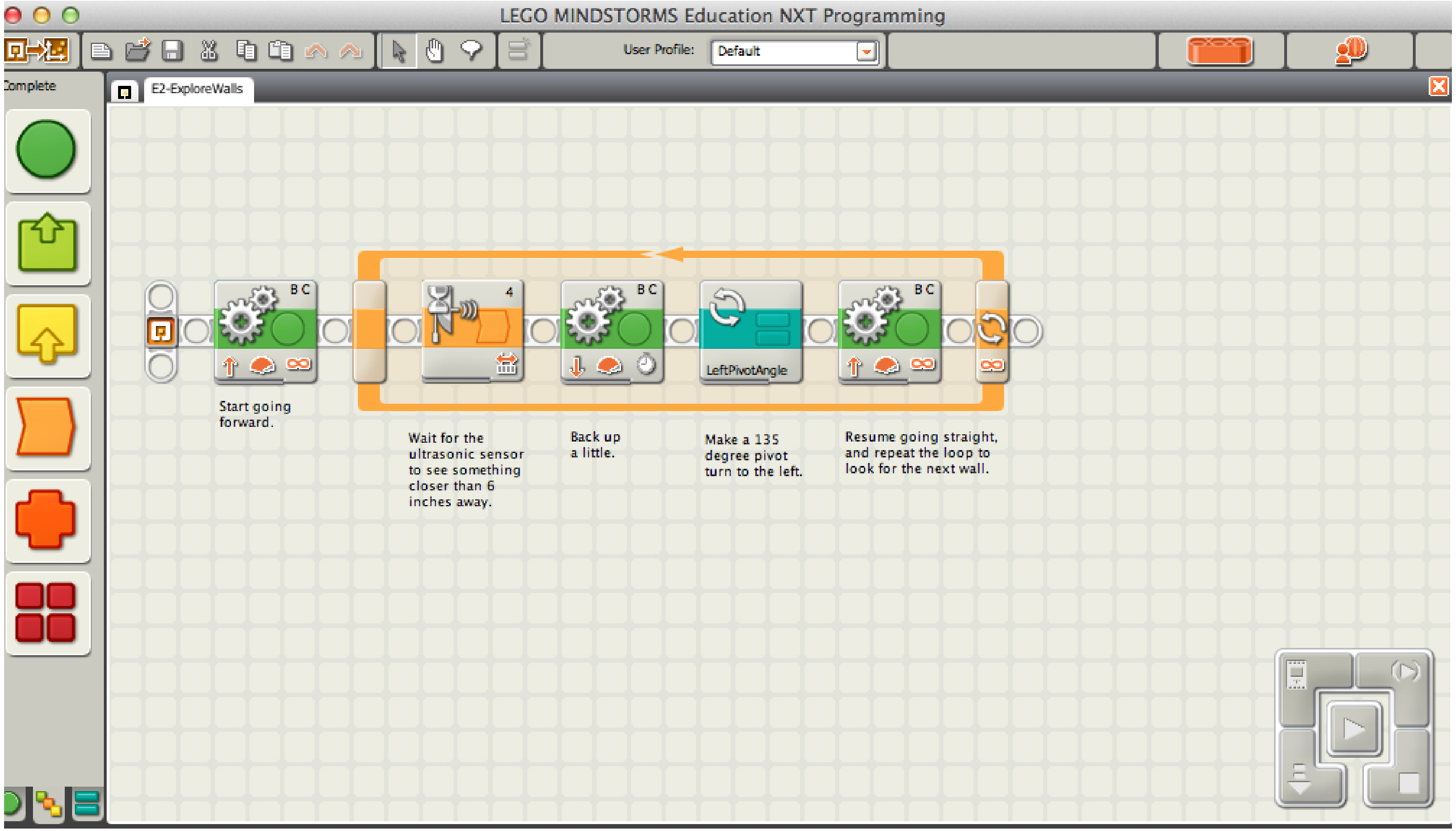
Queue

Cashier
[ST distribution]

Departure

Abstract View

...and another one



Start going forward.

Wait for the ultrasonic sensor to see something closer than 6 inches away.

Back up a little.

Make a 135 degree pivot turn to the left.

Resume going straight, and repeat the loop to look for the next wall.

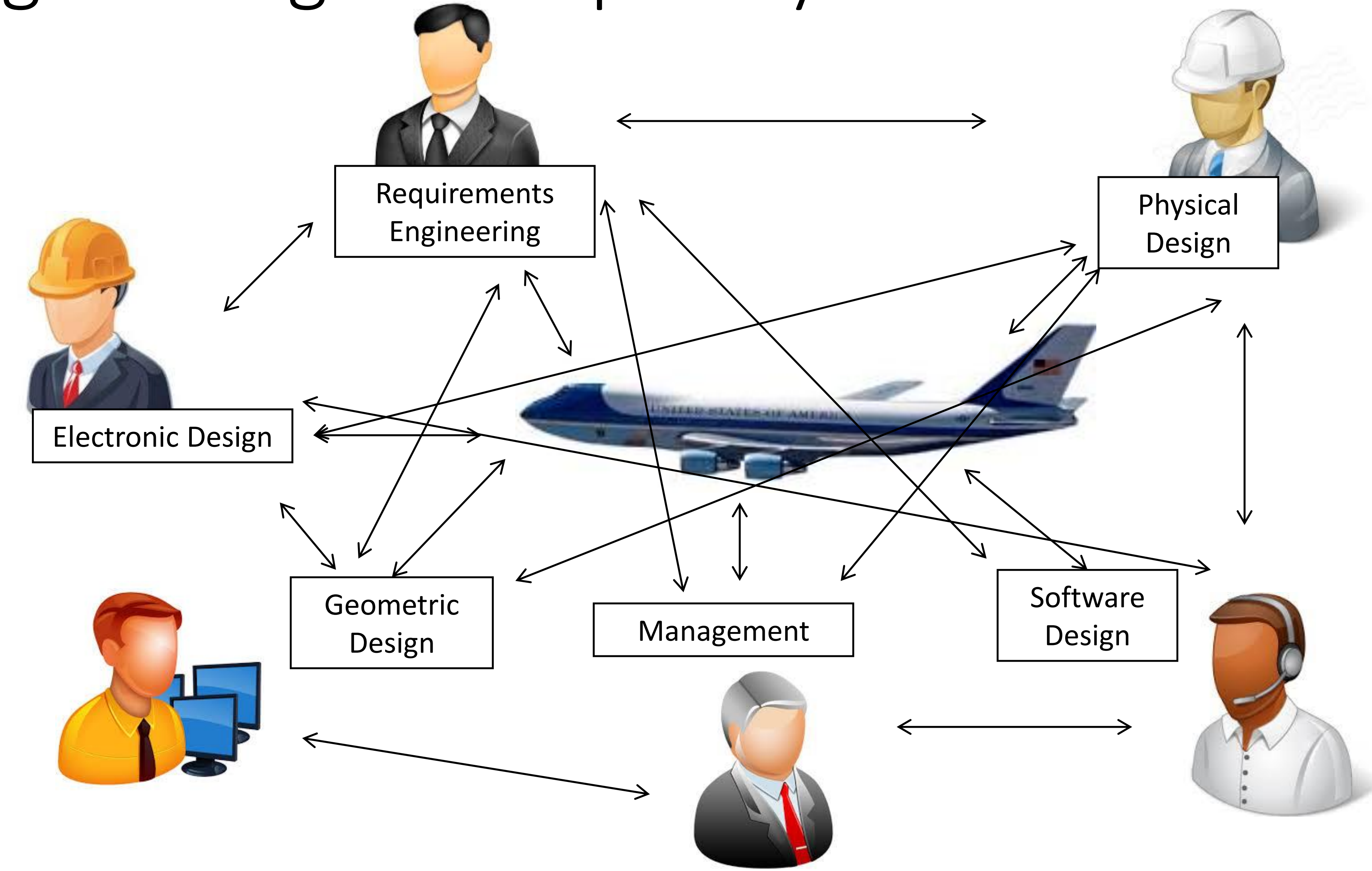
From the requirements to the product

Requirements

PROCESS

Product

Engineering of complex systems



Requirements Engineering

Physical Design

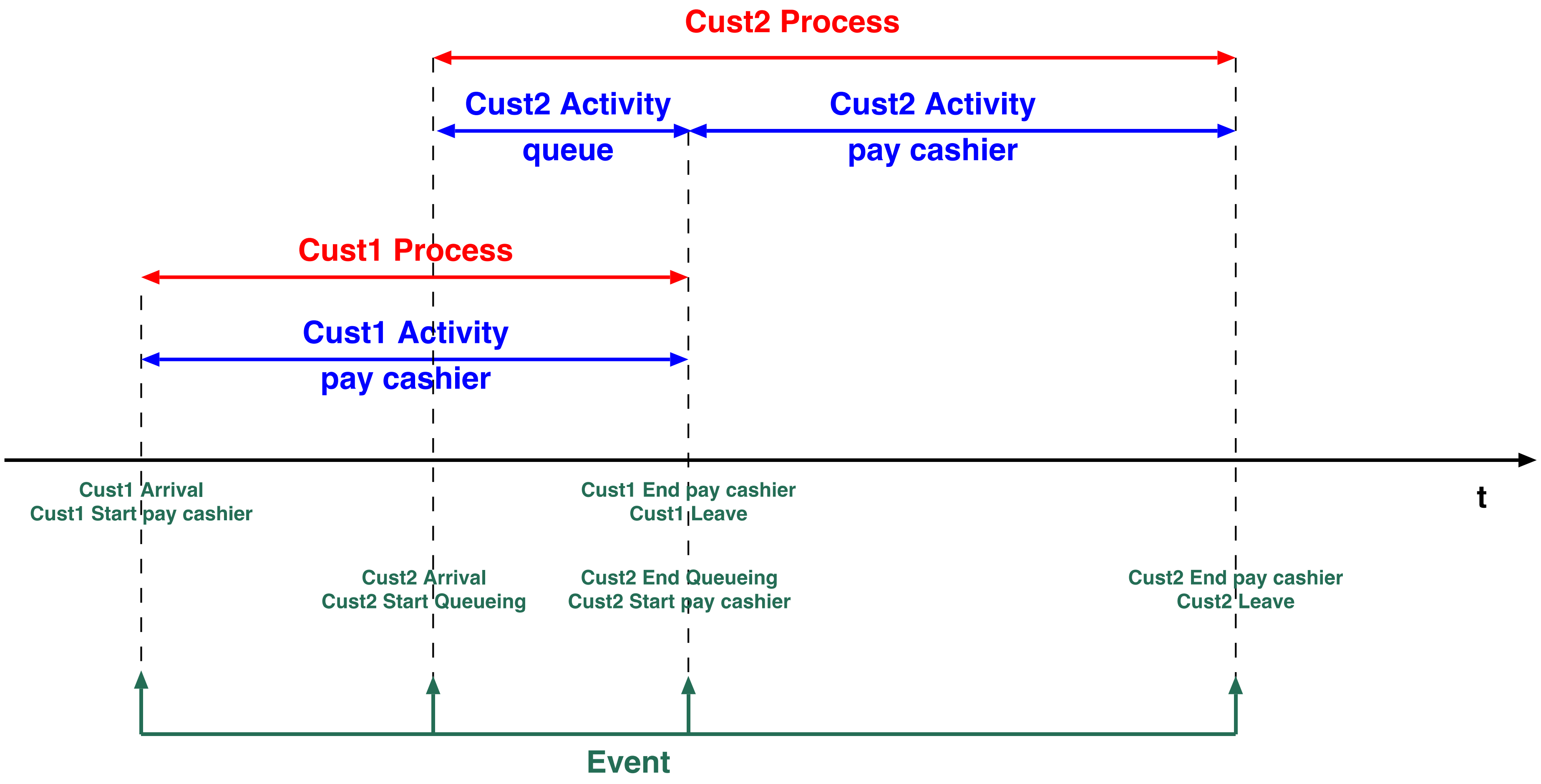
Electronic Design

Geometric Design

Management

Software Design

Event/Activity/Process



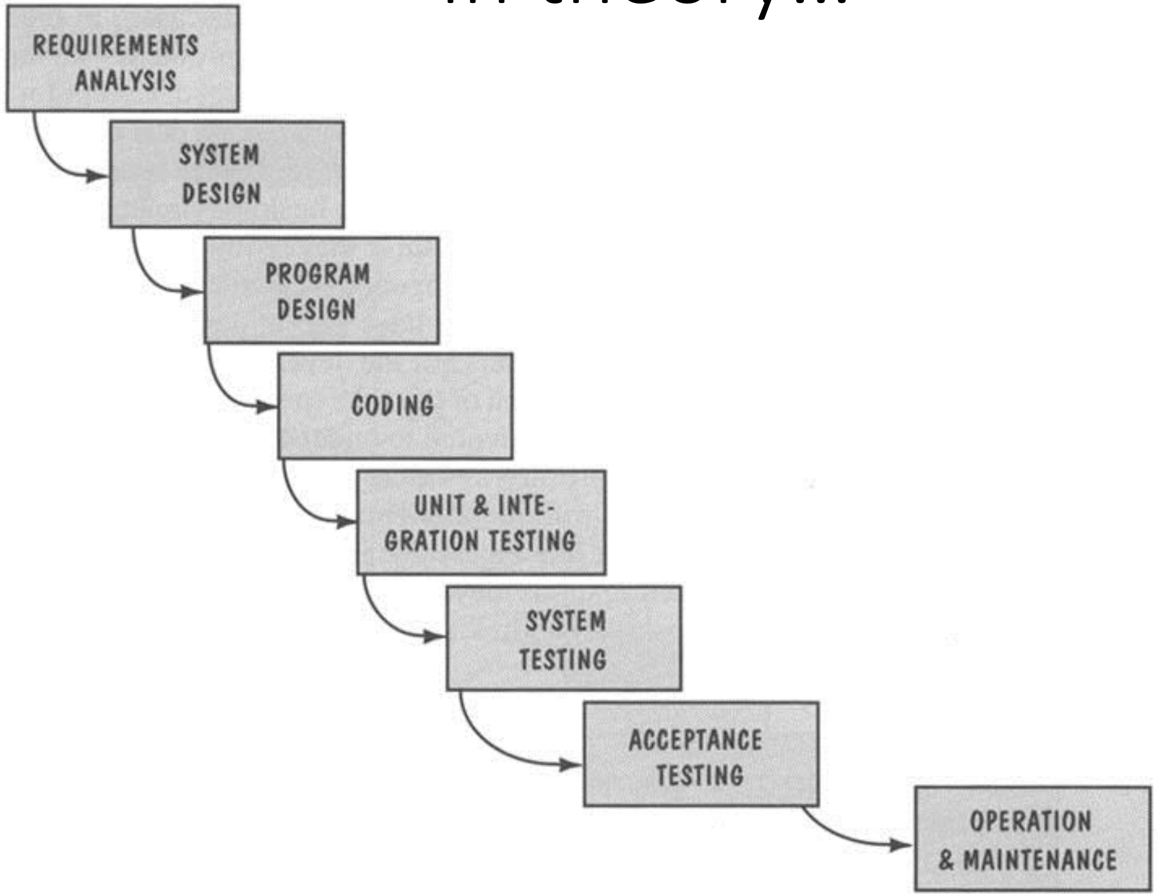
Software Processes

“The Software Engineering **process** is the total set of Software Engineering **activities** needed to transform requirements into software”.

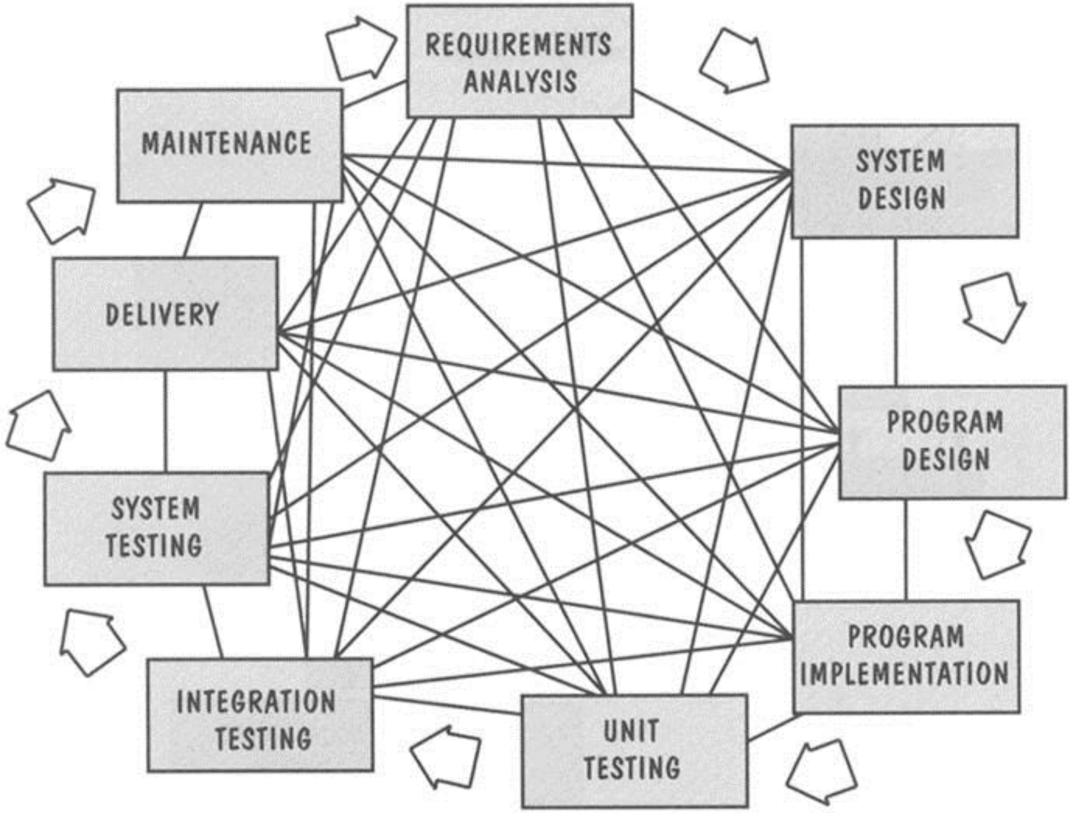
Watts S. Humphrey. Software Engineering Institute, CMU. (portal.acm.org/citation.cfm?id=75122)

Waterfall Process

In theory…

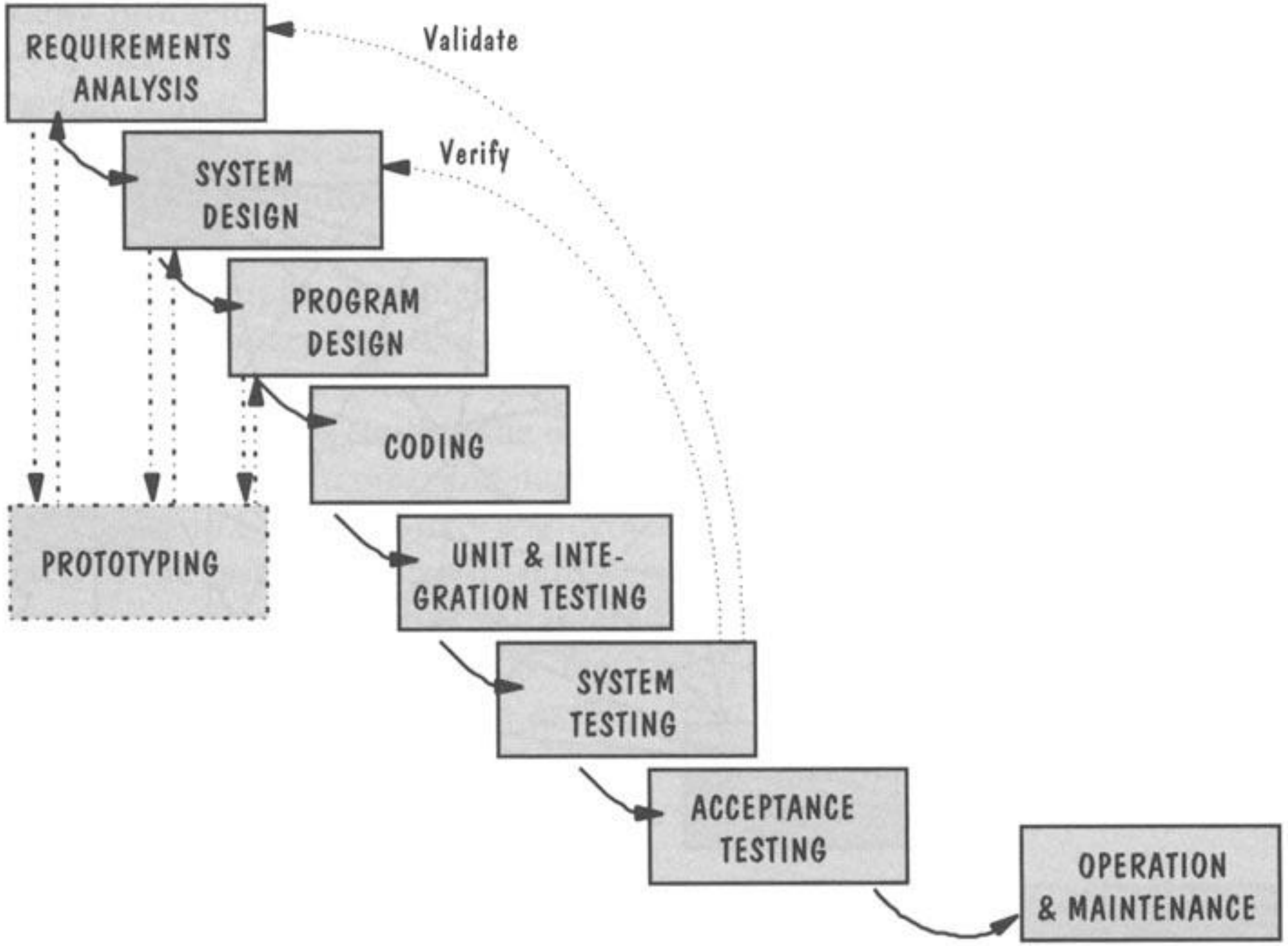


…and in the reality



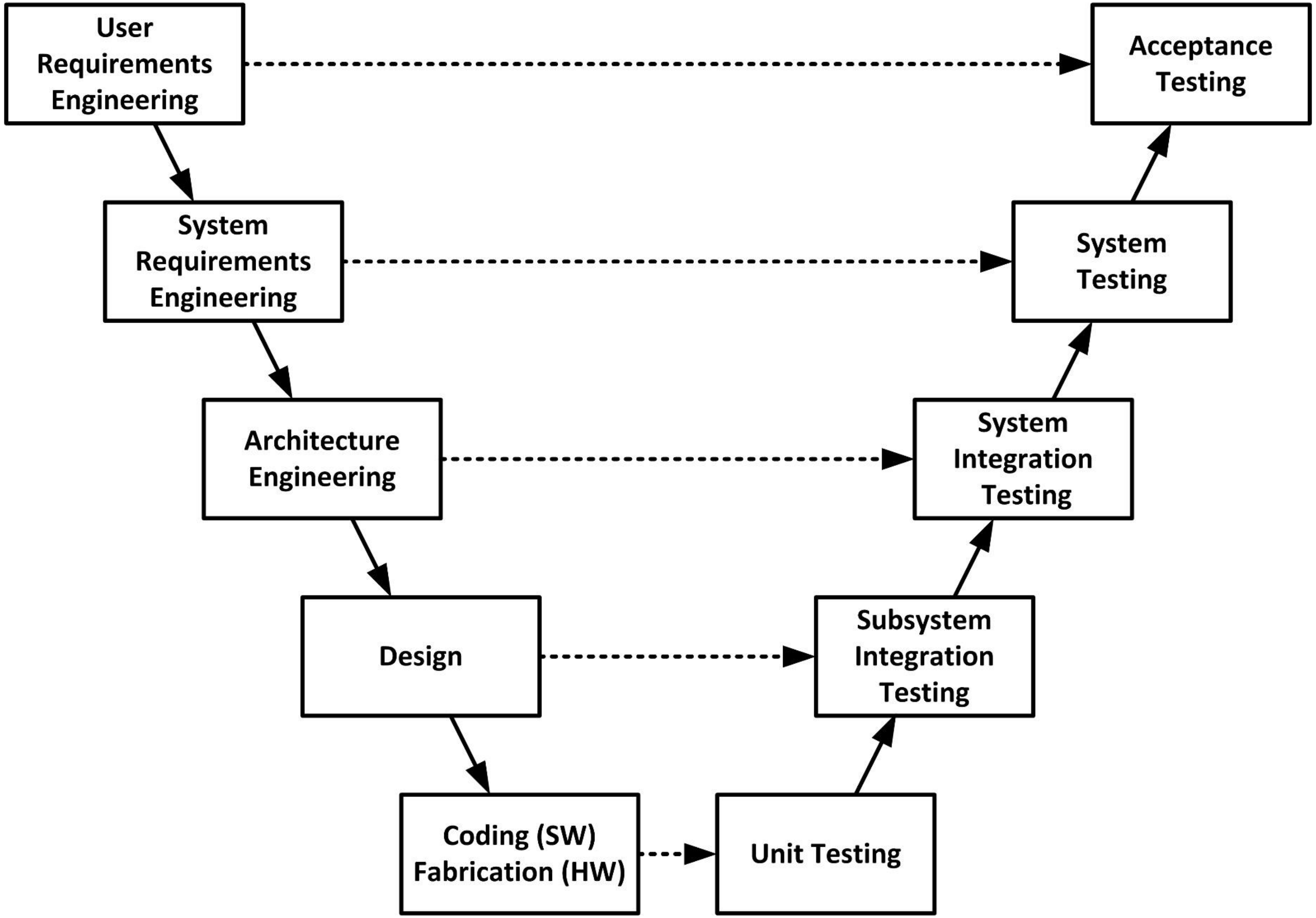
Shari Lawrence Pfleeger. Software Engineering: Theory and Practice (Second Edition). Prentice Hall. 2001.

Waterfall with prototyping



Shari Lawrence Pfleeger. Software Engineering: Theory and Practice (Second Edition). Prentice Hall. 2001.

The V Model



User Requirements Engineering

System Requirements Engineering

Architecture Engineering

Design

Coding (SW) Fabrication (HW)

Unit Testing

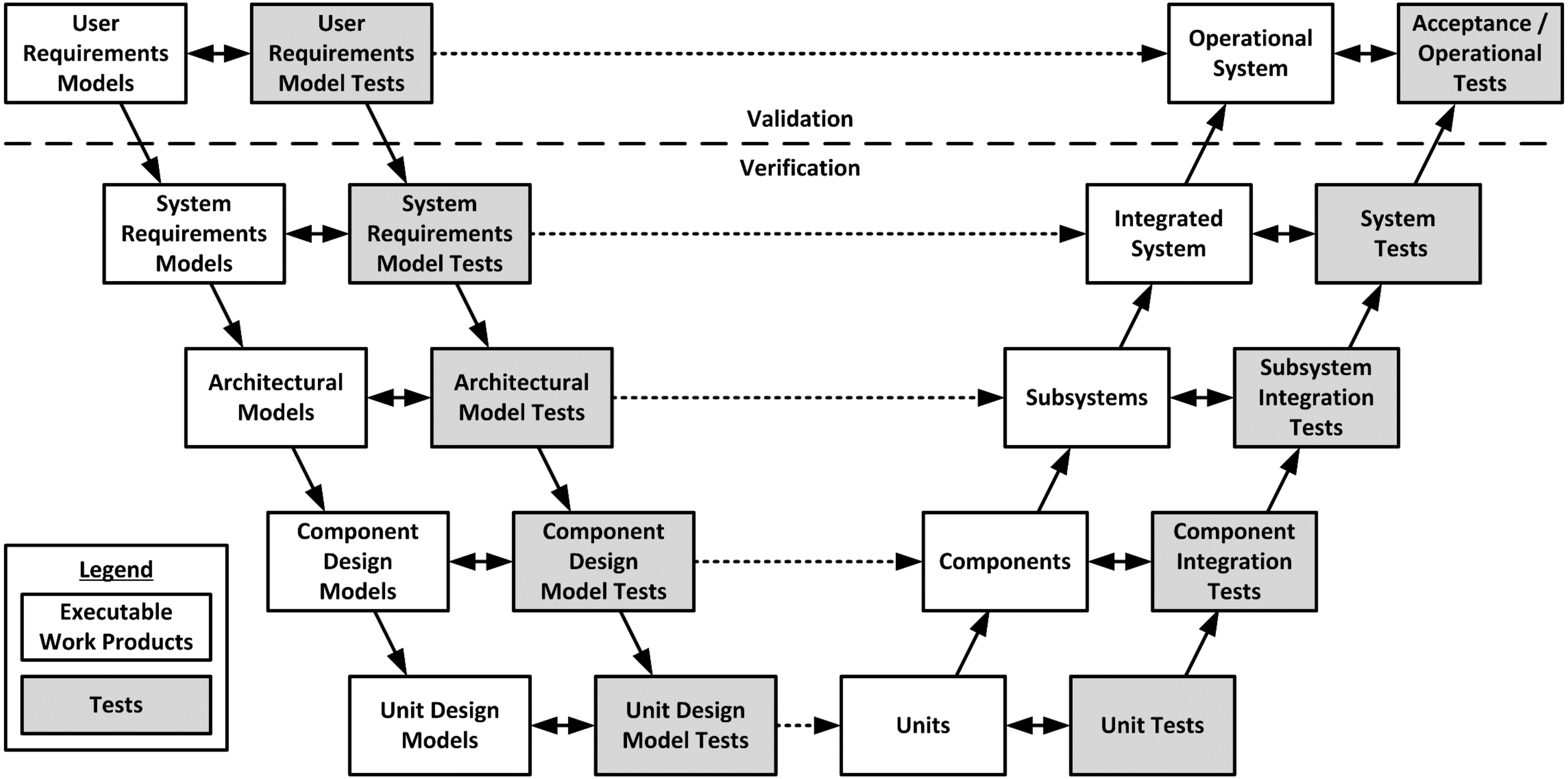
Subsystem Integration Testing

System Integration Testing

System Testing

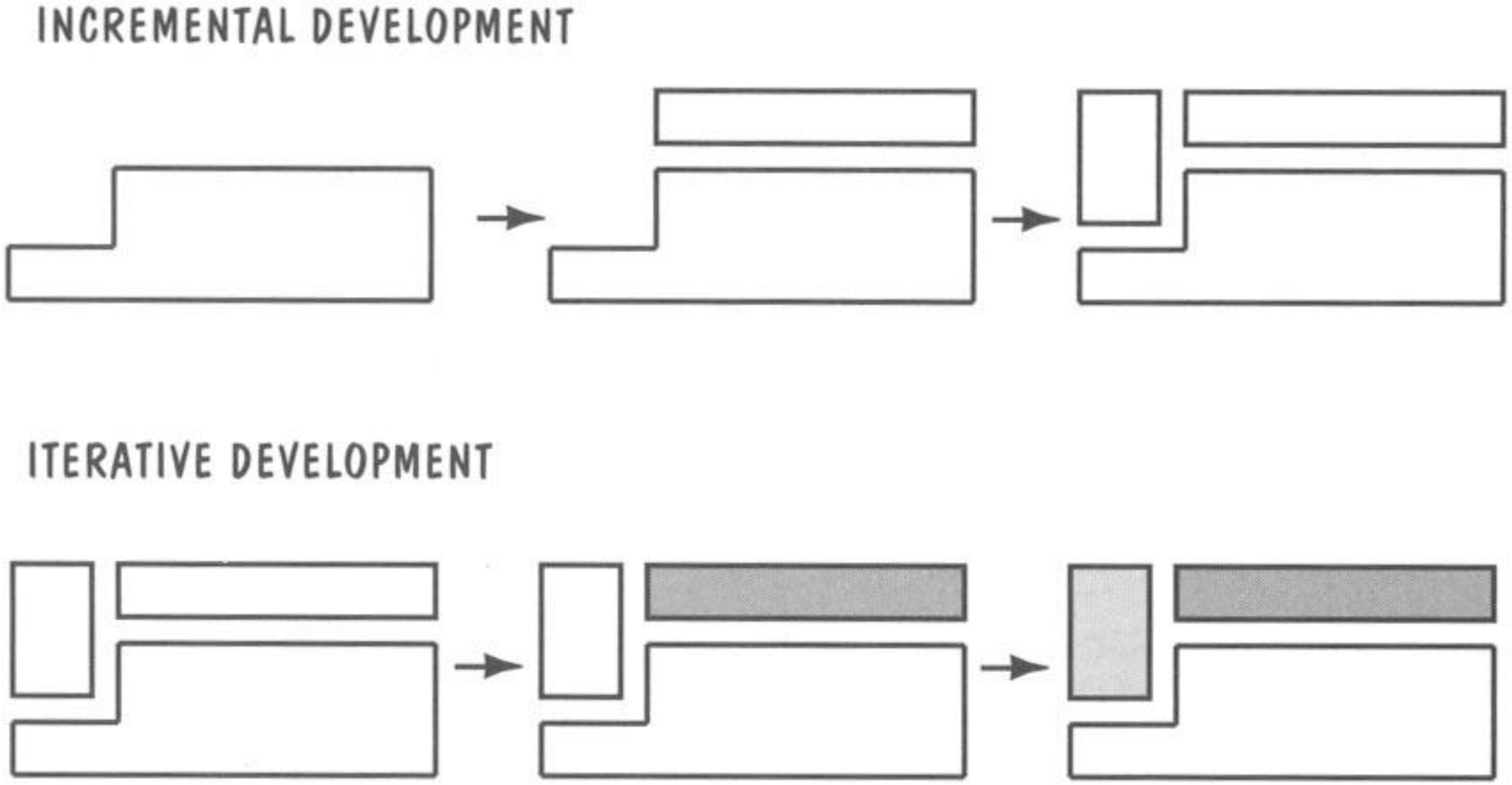
Acceptance Testing

The tester’s double V Model



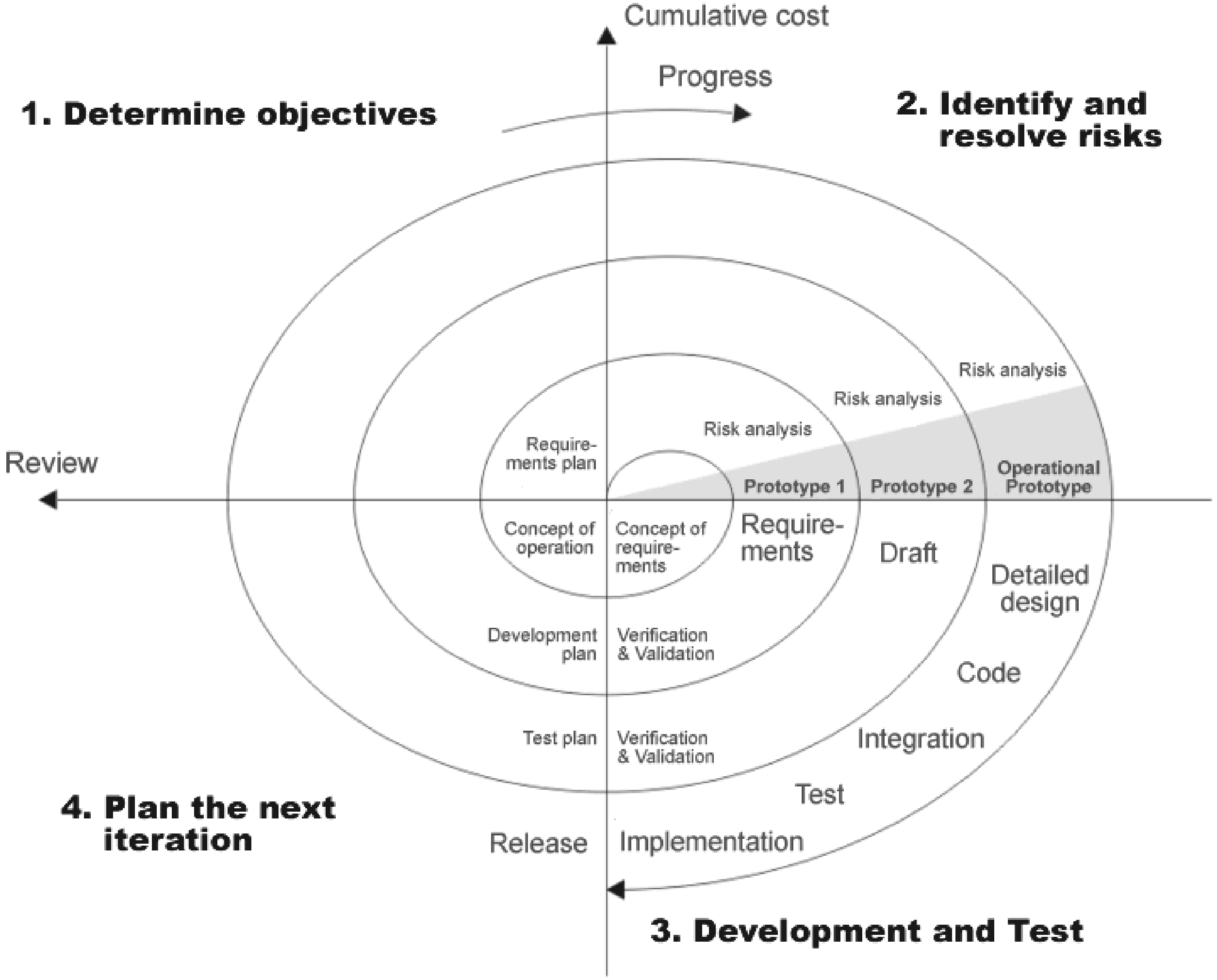
Kevin Forsberg and Harold Mooz, “The Relationship of System Engineering to the Project Cycle,” in Proceedings of the First Annual Symposium of National Council on System Engineering, October 1991: 57–65.

Iterative vs. Incremental



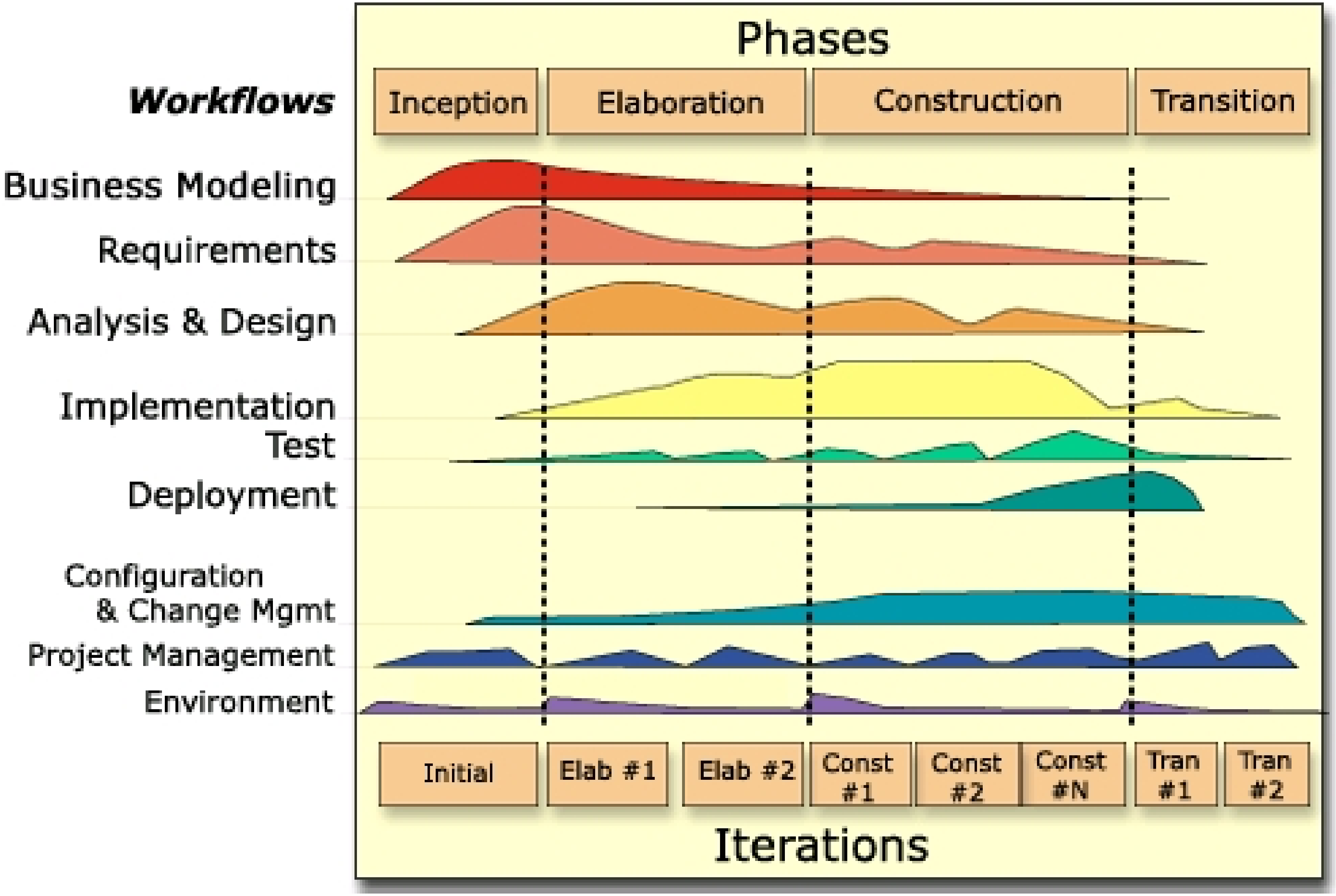
Shari Lawrence Pfleeger. Software Engineering: Theory and Practice (Second Edition). Prentice Hall. 2001.

Spiral Process



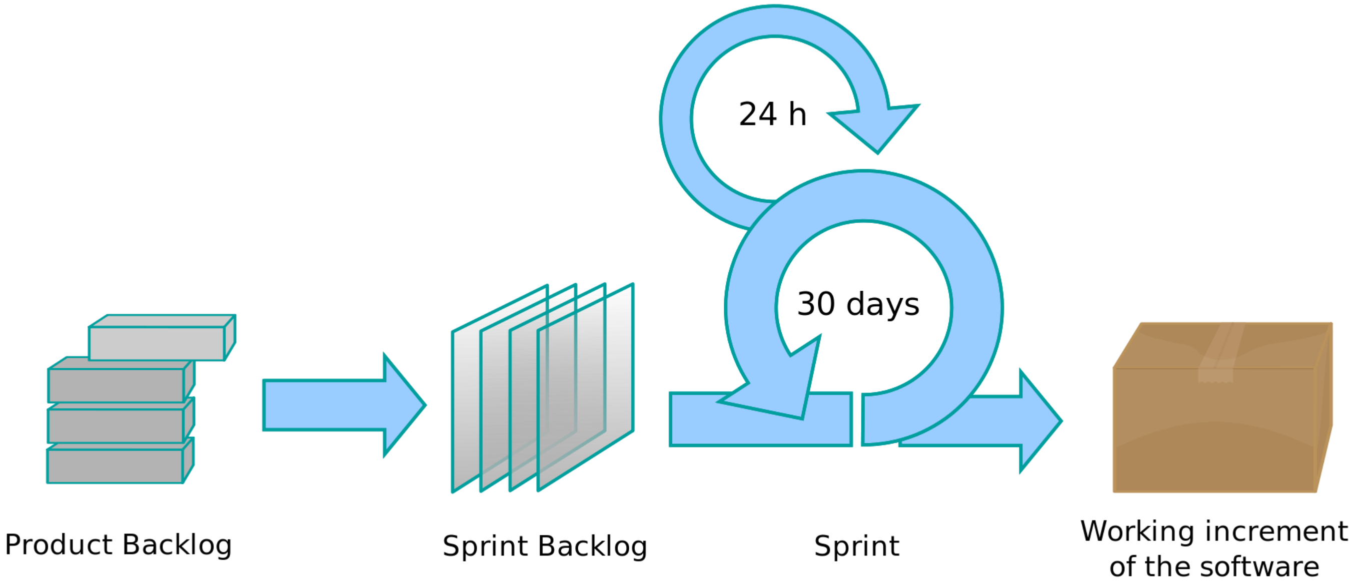
Boehm B, "A Spiral Model of Software Development and Enhancement", IEEE Computer, IEEE, 21(5):61-72, May 1988

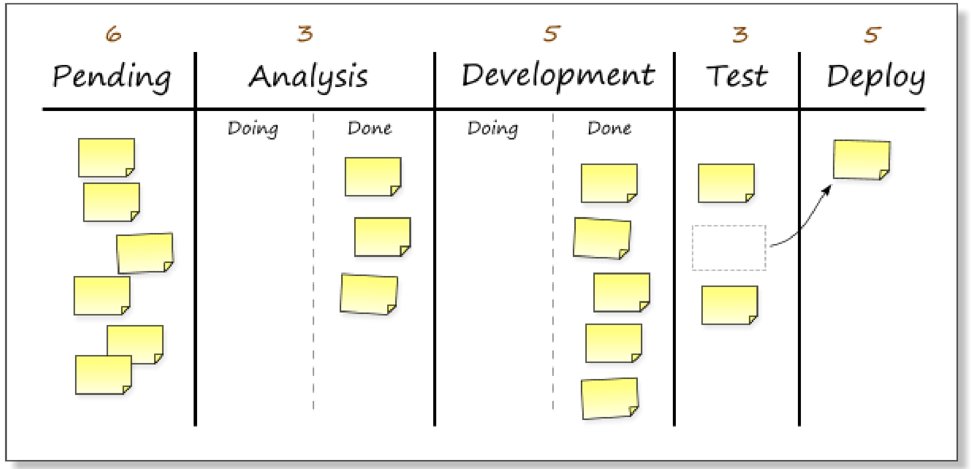
Rational Unified Process (RUP)



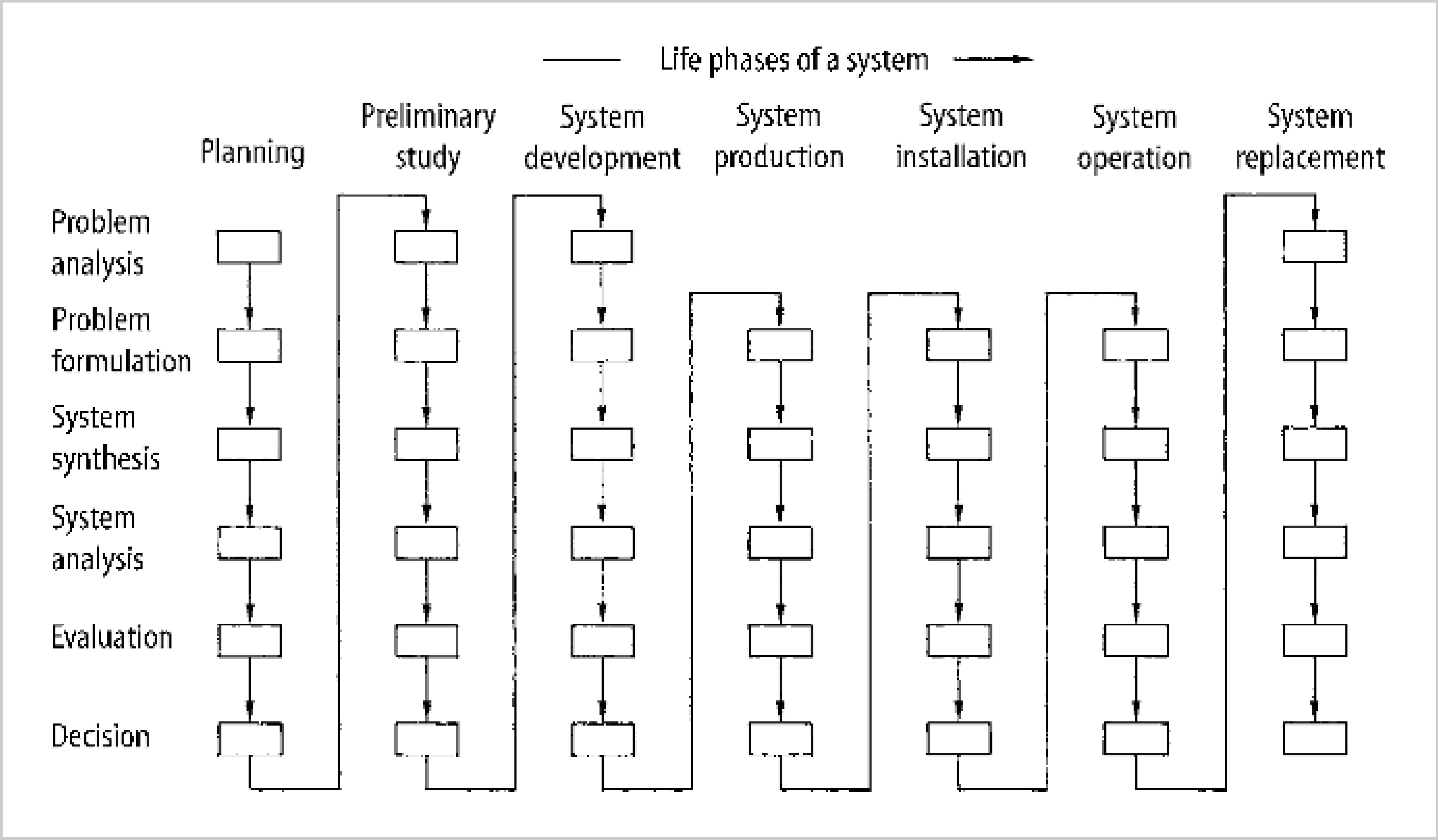
Agile methodologies

- Core principles
  - Adaptive planning
  - Evolutionary development
  - Early delivery
  - Continuous improvement
  - Rapid and flexible response to change
- Agile practices
  - TDD, CI, cross-functional team,
    pair programming, timeboxing, retrospective…
- Methodologies
  - Scrum, Kanban, Lean, AUP, Crystal Clear…



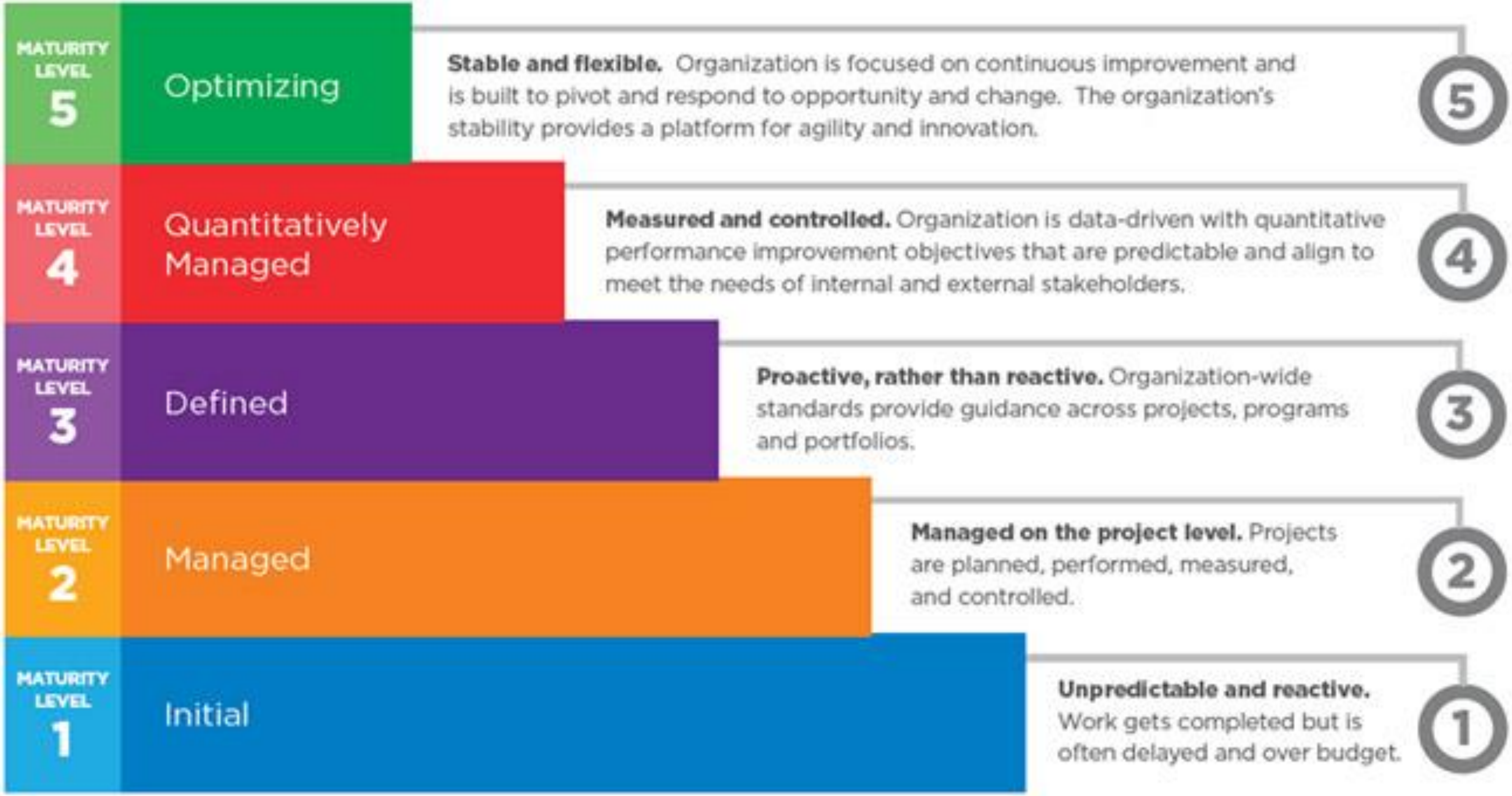


Not only Software!



From: G. Pahl and W. Beitz and J. Feldhusen and K.-H. Grote; Engineering Design – A Systematic Approach; Springer; 2007

Capability Maturity Model Integration (CMMI)



Stable and flexible. Organization is focused on continuous improvement and is built to pivot and respond to opportunity and change. The organization's stability provides a platform for agility and innovation.

Measured and controlled. Organization is data-driven with quantitative performance improvement objectives that are predictable and align to meet the needs of internal and external stakeholders.

Proactive, rather than reactive. Organization-wide standards provide guidance across projects, programs and portfolios.

Managed on the project level. Projects are planned, performed, measured, and controlled.

Unpredictable and reactive. Work gets completed but is often delayed and over budget.

From: http://www.isaca.org/

Process modeling

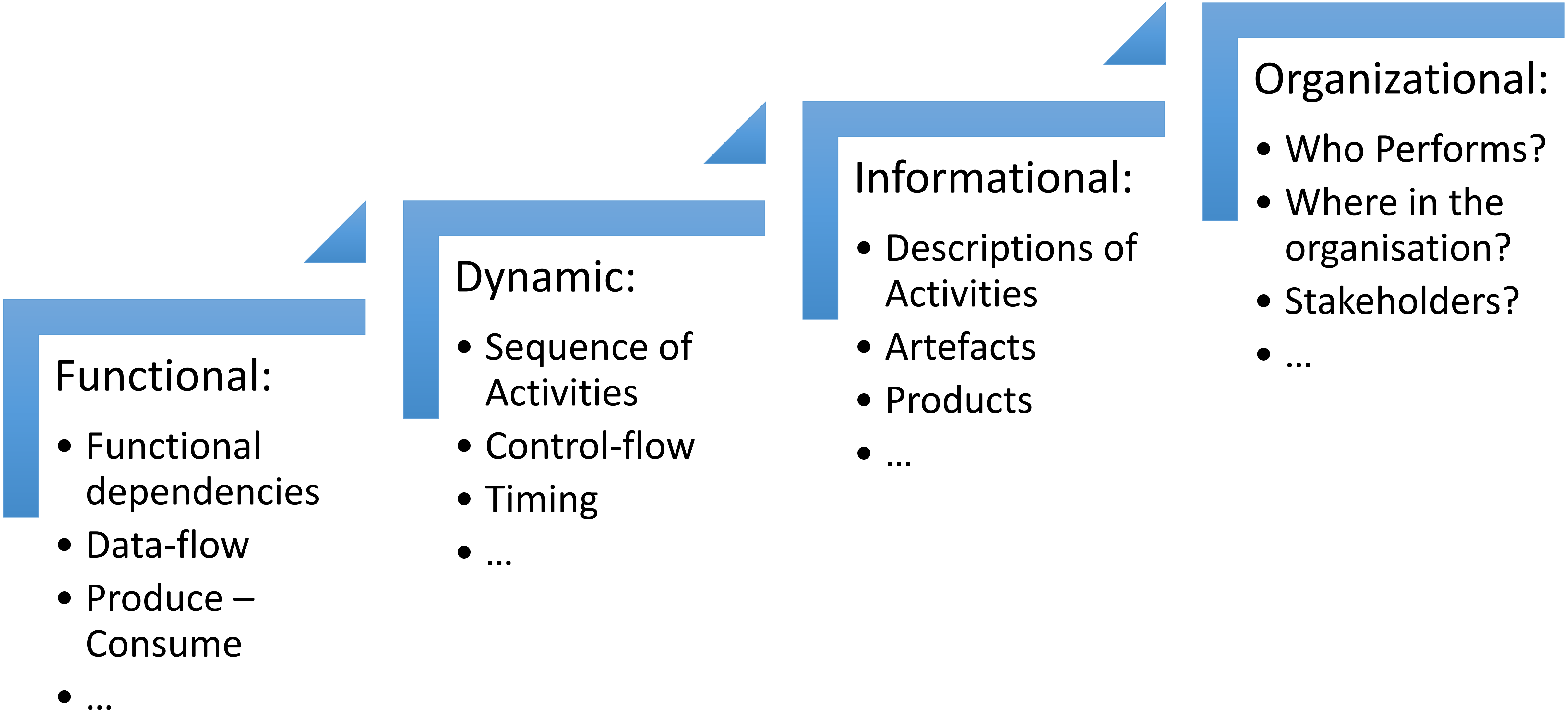
Why explicit modeling?

Descriptive

Prescriptive

Proscriptive

Describing Processes



Functional:

- Functional dependencies
- Data-flow
- Produce – Consume
- …

Dynamic:

- Sequence of Activities
- Control-flow
- Timing
- …

Informational:

- Descriptions of Activities
- Artefacts
- Products
- …

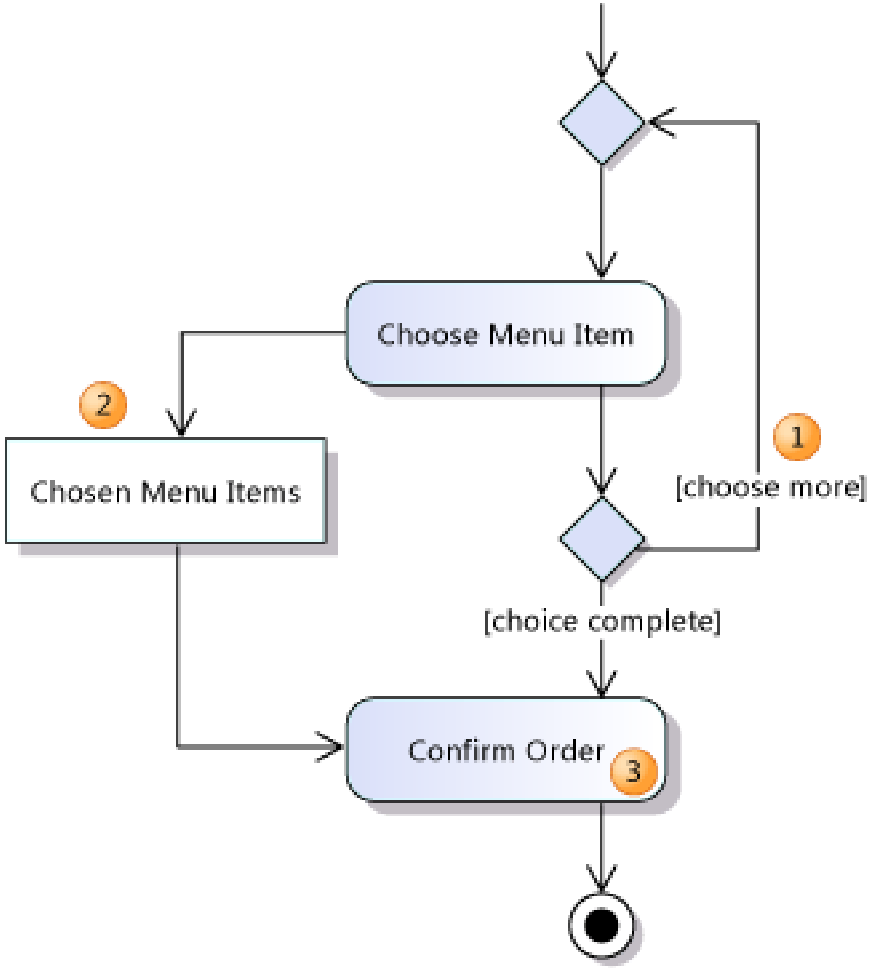
Organizational:

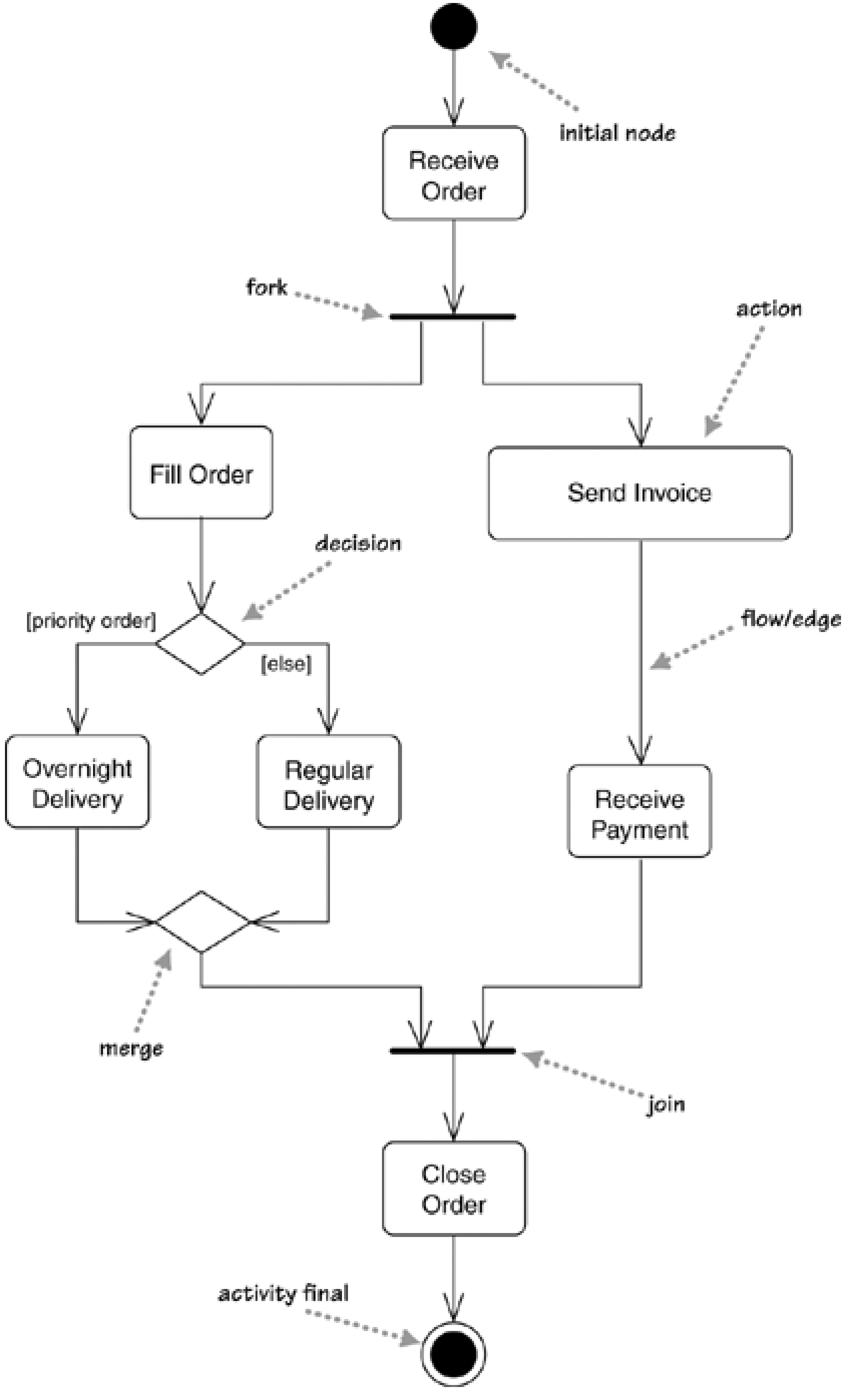
- Who Performs?
- Where in the organisation?
- Stakeholders?
- …

Languages

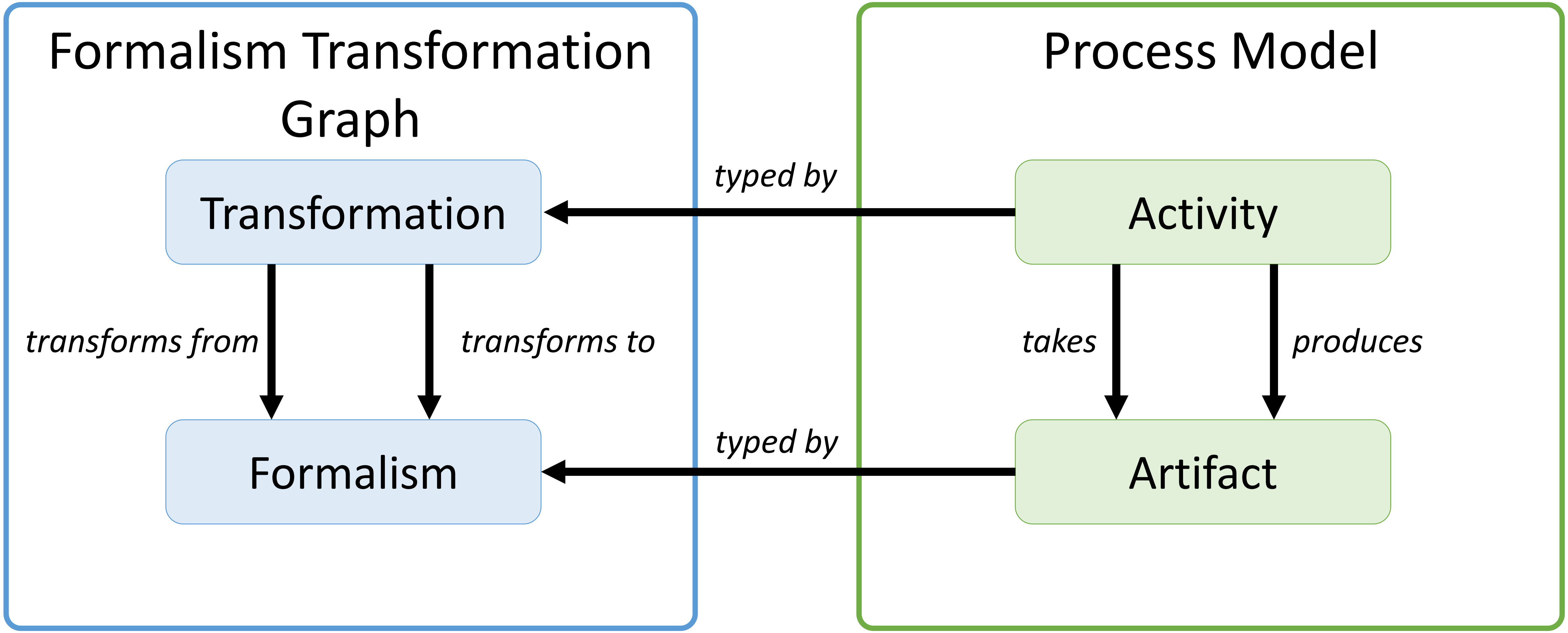
- **UML Activities**
- Business Process Modeling Notation (BPMN)
- jBPM
- Event Process Chains
- Petri-nets
- Role Activity Diagram
- **FTG+PM**
- Etc.

UML Activities

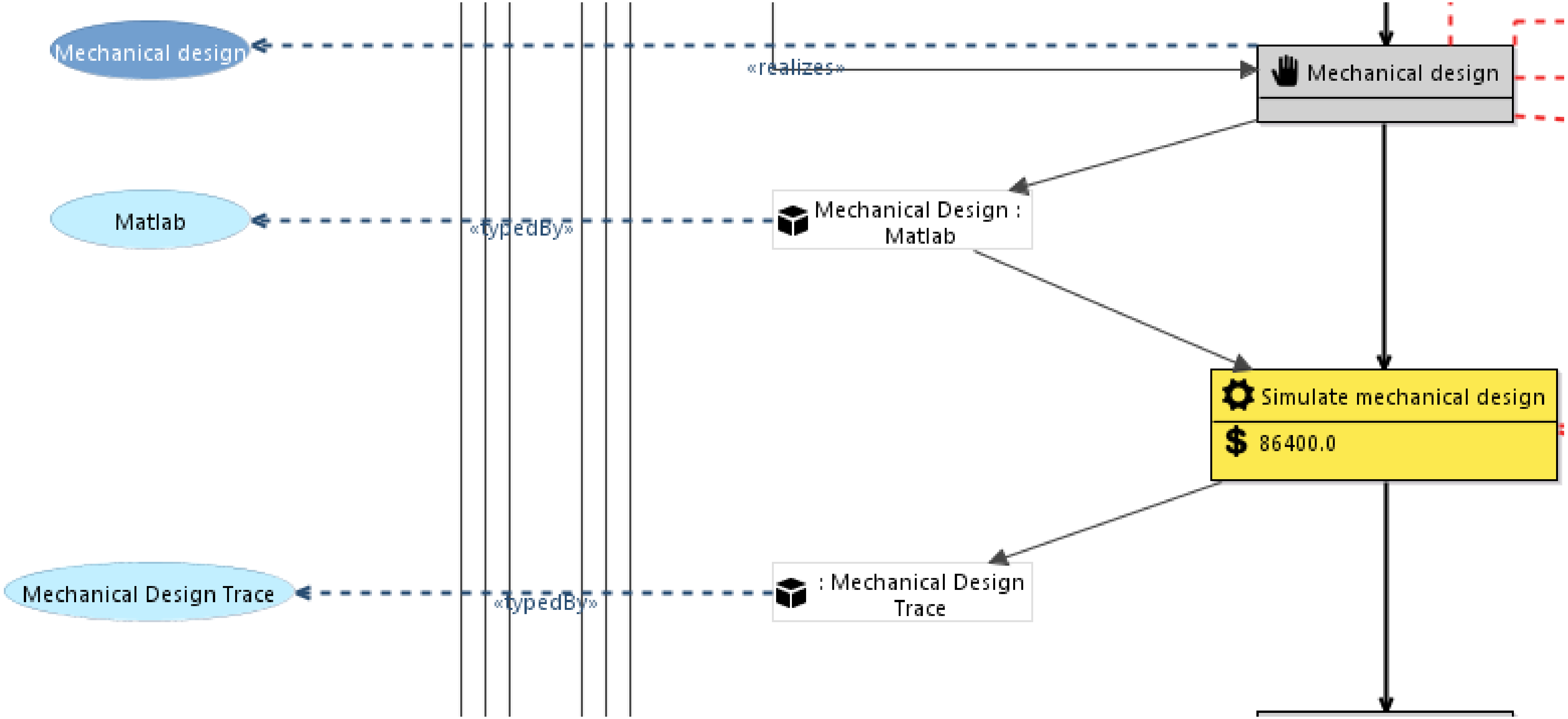




FTG+PM

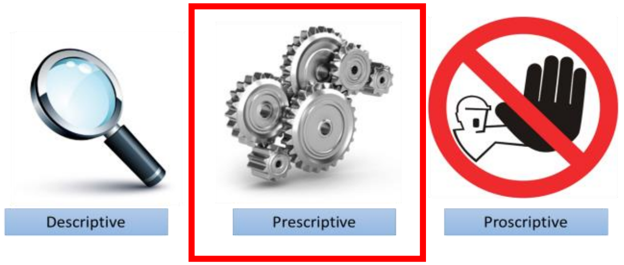

Formalism Transformation Graph
Process Model
Transformation
Formalism
transforms from
transforms to
typed by
typed by
Activity
Artifact
takes
produces

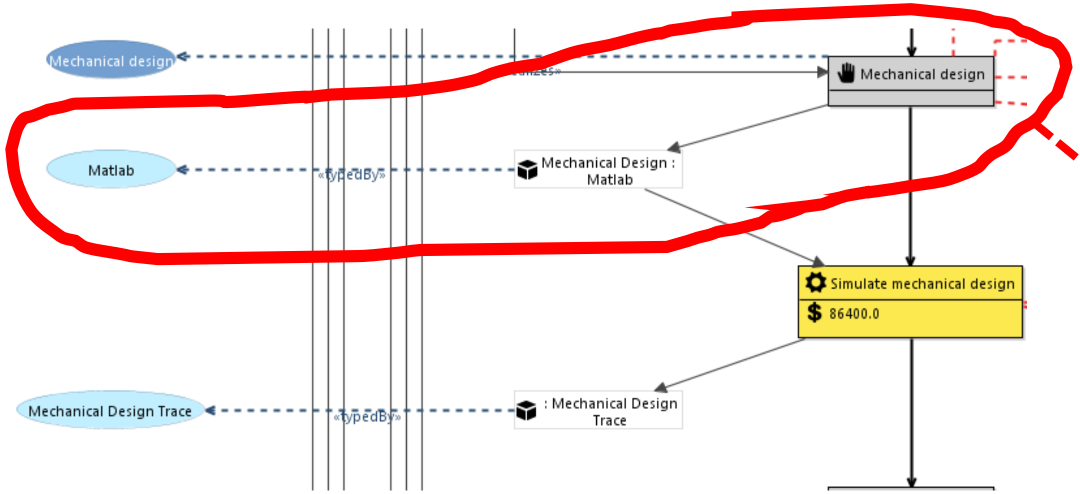
FTG+PM

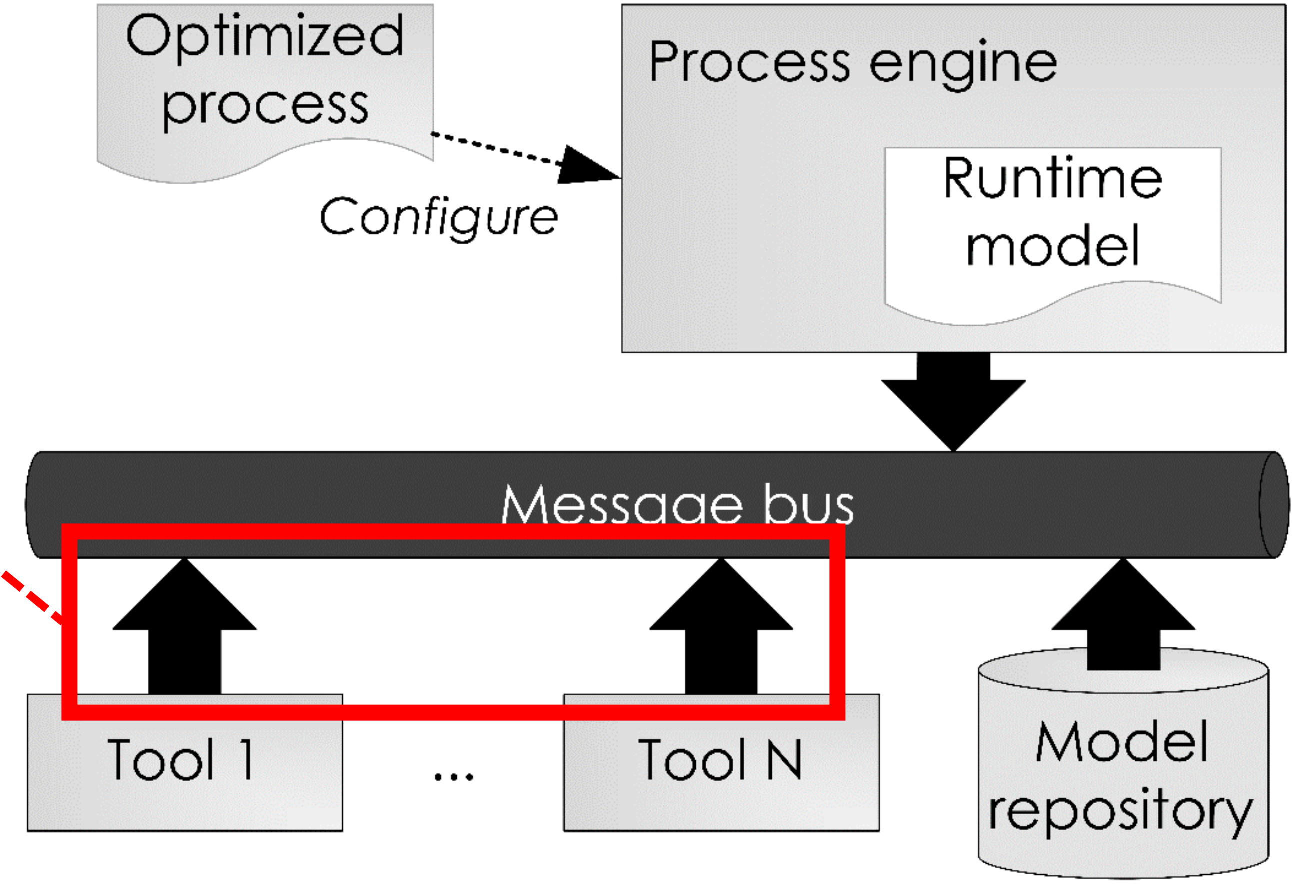

Mechanical design
«realizes»
Mechanical design
Matlab
«typedBy»
Mechanical Design : Matlab
Simulate mechanical design
$ 86400.0
Mechanical Design Trace
«typedBy»
: Mechanical Design Trace

Tool interoperability and process orchestration

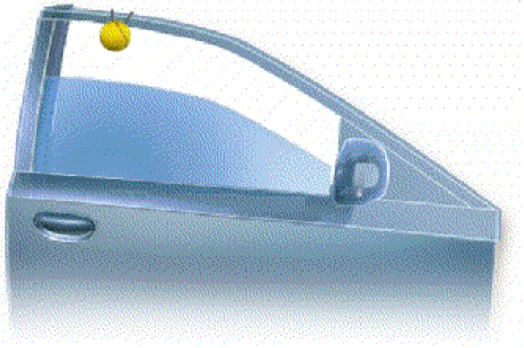
Why explicit modeling?

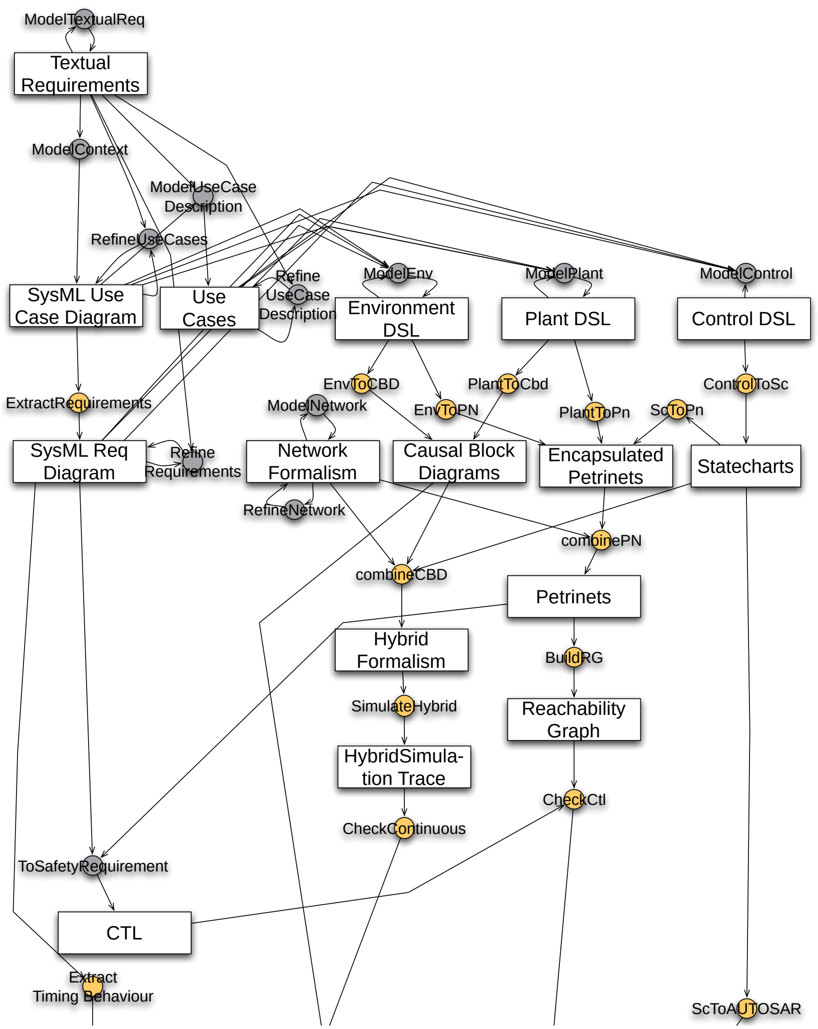


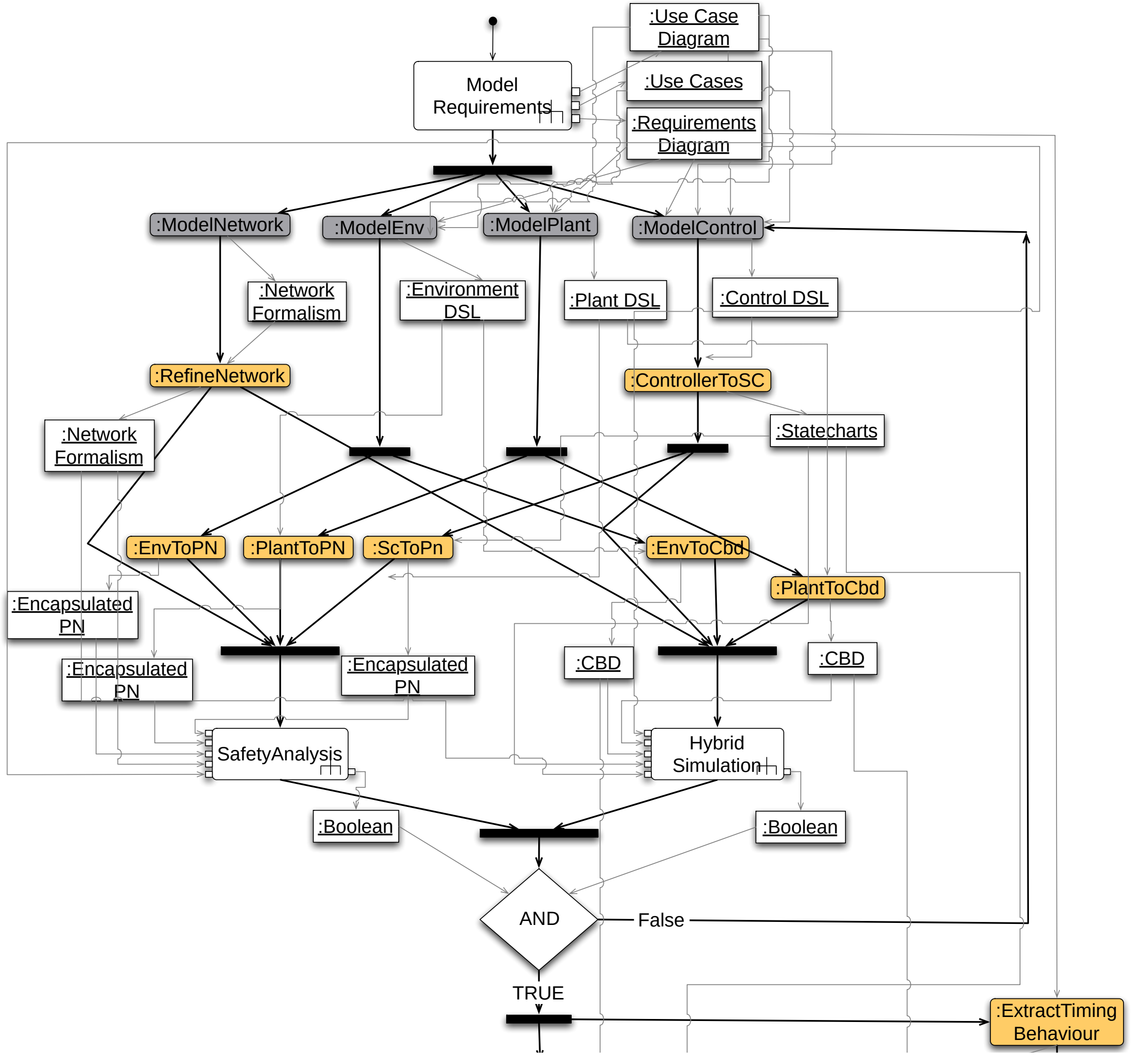




Power Window Example

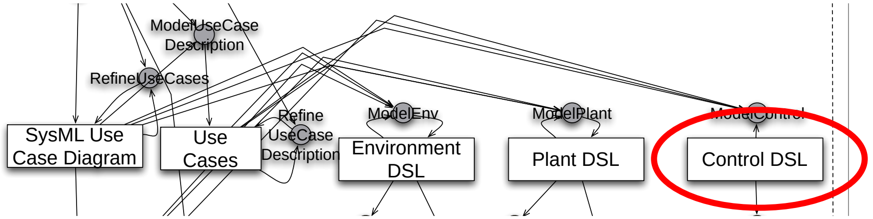


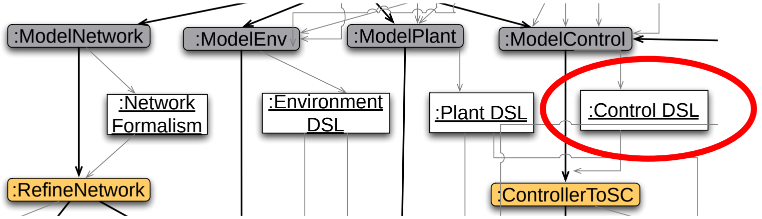


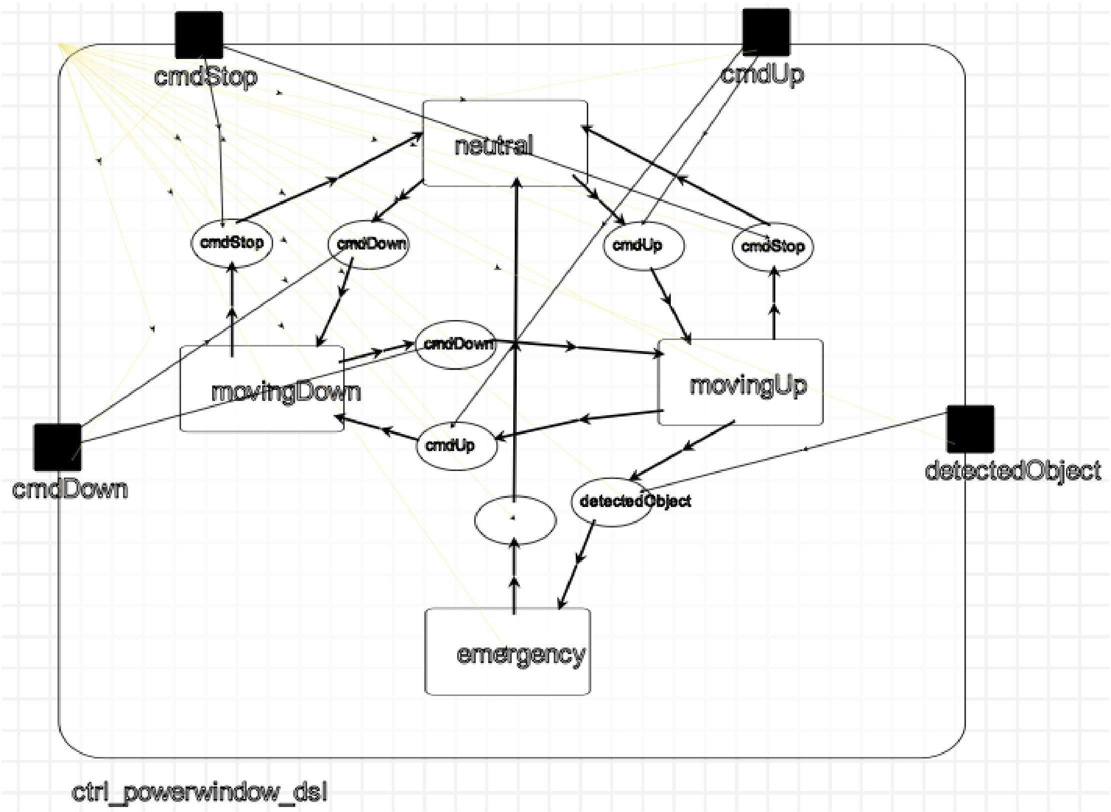


- *Levi Lucio, Sadaf Mustafiz, Joachim Denil, Hans Vangheluwe, Maris Jukss, FTG+PM: An Integrated Framework for Investigating Model Transformation Chains. SDL Forum 2013: 182-202*
- *Sadaf Mustafiz, Joachim Denil, Levi Lucio, and Hans Vangheluwe; "The FTG+PM Framework for Multi-Paradigm Modelling: An Automotive Case Study"; Accepted @ MPM2012 of Models2012, 2012*

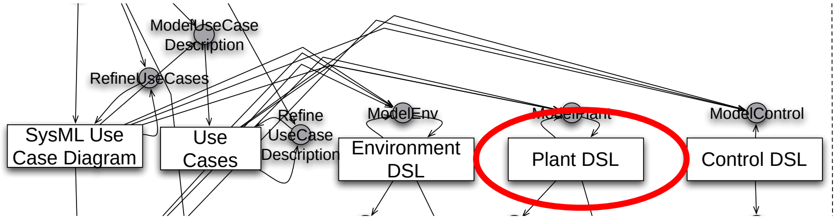
Controller

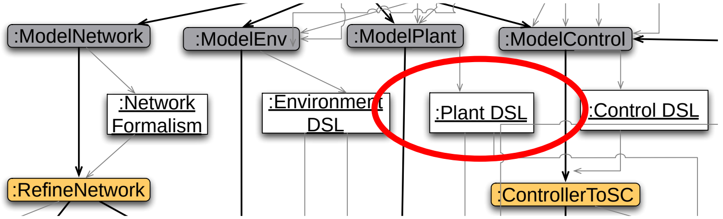

ModelUseCase Description
RefineUseCases
SysML Use Case Diagram
Use Cases
Refine UseCase Description
ModelEnv
Environment DSL
ModelPlant
Plant DSL
ModelControl
Control DSL

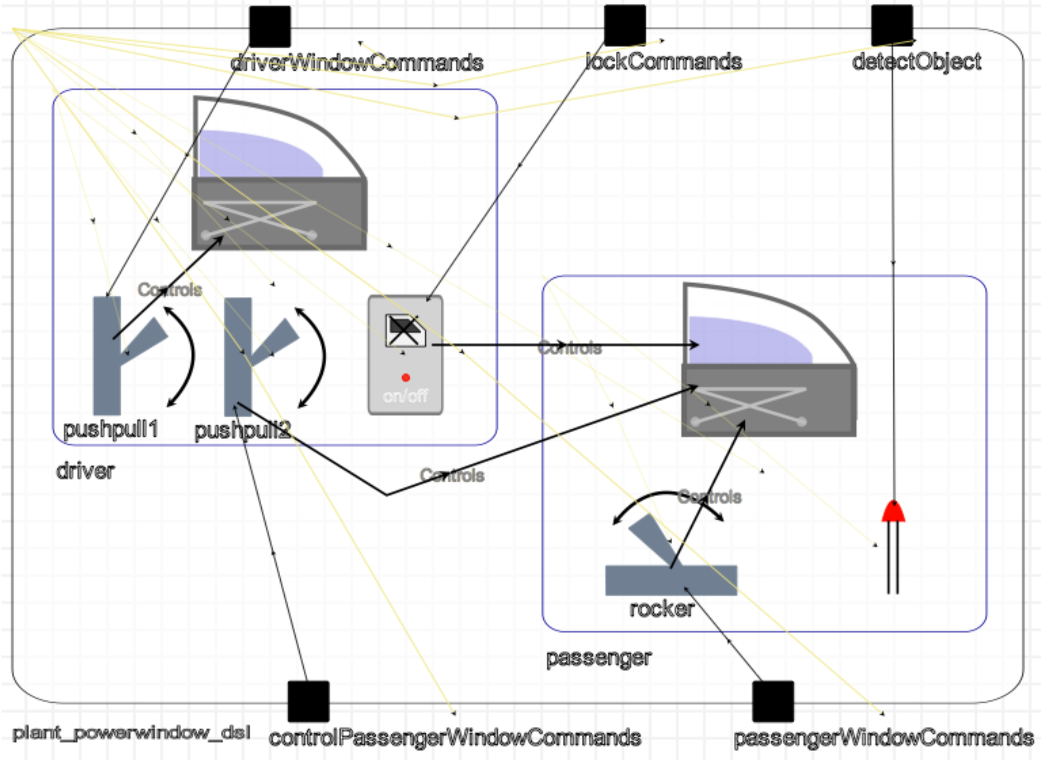

:ModelNetwork
:ModelEnv
:ModelPlant
:ModelControl
:Network Formalism
:Environment DSL
:Plant DSL
:Control DSL
:RefineNetwork
:ControllerToSC


cmdStop
cmdUp
neutral
cmdStop
cmdDown
cmdUp
cmdStop
cmdDown
movingDown
movingUp
cmdUp
cmdDown
detectedObject
detectedObject
emergency
ctrl_powerwindow_dsl

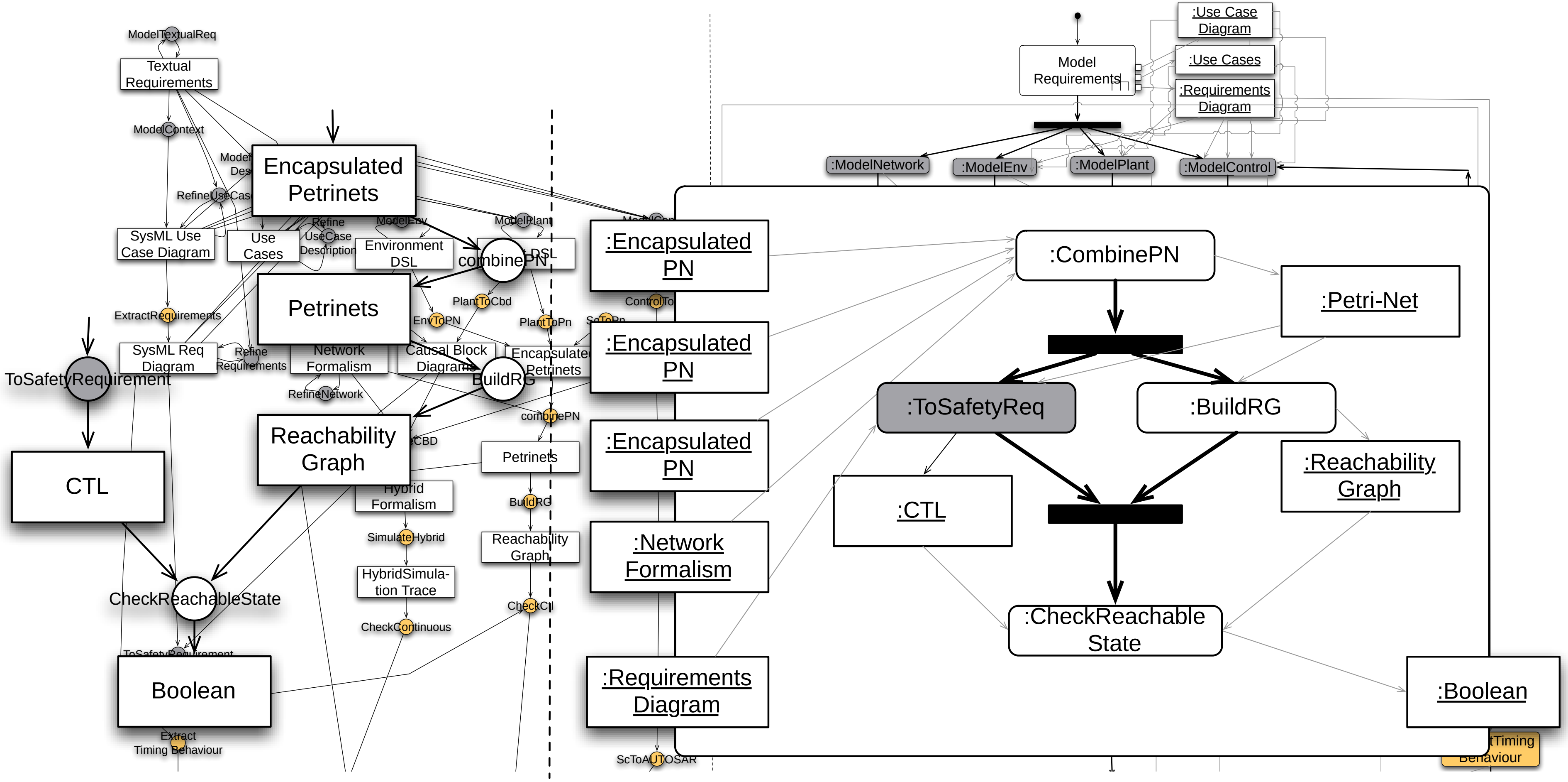
Plant Model


ModelUseCase Description
RefineUseCases
SysML Use Case Diagram
Use Cases
Refine UseCase Description
ModelEnv
Environment DSL
ModelPlant
Plant DSL
ModelControl
Control DSL


:ModelNetwork
:ModelEnv
:ModelPlant
:ModelControl
:Network Formalism
:Environment DSL
:Plant DSL
:Control DSL
:RefineNetwork
:ControllerToSC


driverWindowCommands
lockCommands
detectObject
Controls
pushpull1
pushpull2
on/off
driver
Controls
Controls
Controls
rocker
passenger
plant_powerwindow_dsl
controlPassengerWindowCommands
passengerWindowCommands

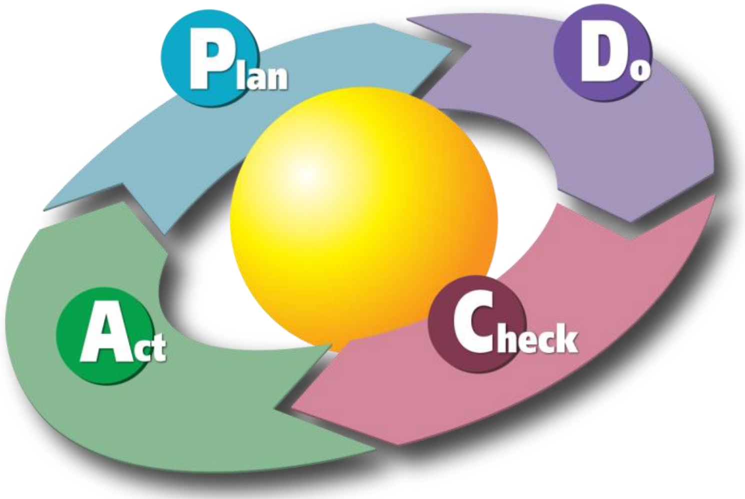
Safety Analysis



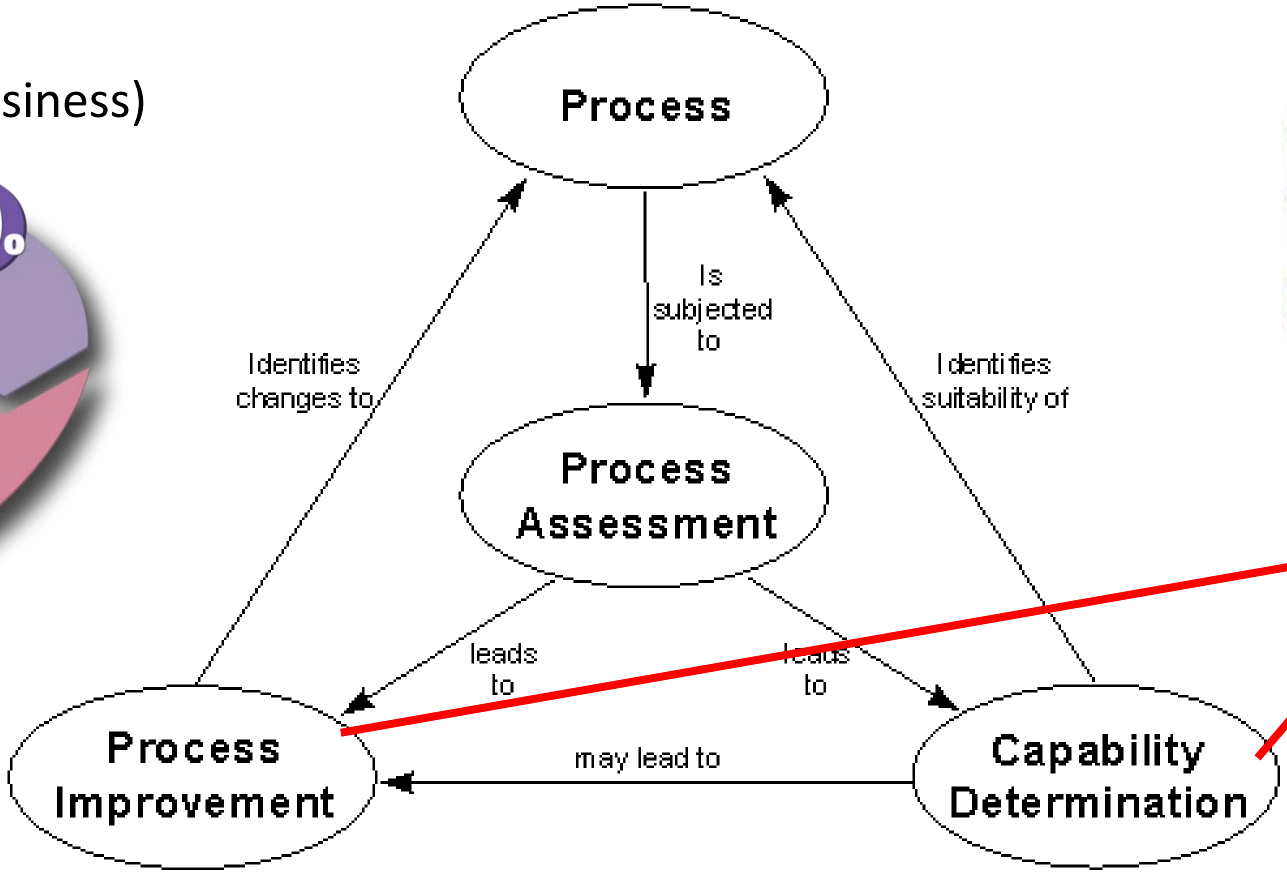
Process improvement
(and optimization)

Software process improvement (SPI)

Analogy:
PDCA cycle in controlling (business)



Elements of an SPI framework



Specific implementation

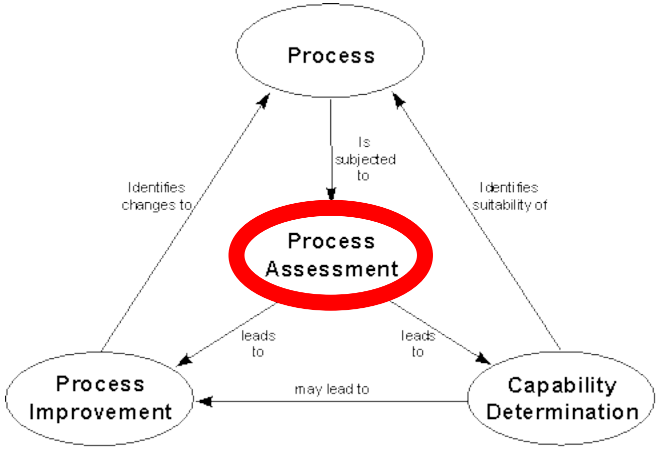
SPICE

ISO/IEC 15504

Software
Process
Improvement w/
Capability
d**E**termination

T. Rout: "SPICE and the CMM: is the CMM compatible with ISO/IEC 15504." AquIS'98 (1998).

Process assessment



- Required for improvement
- Identifies the changes to the process

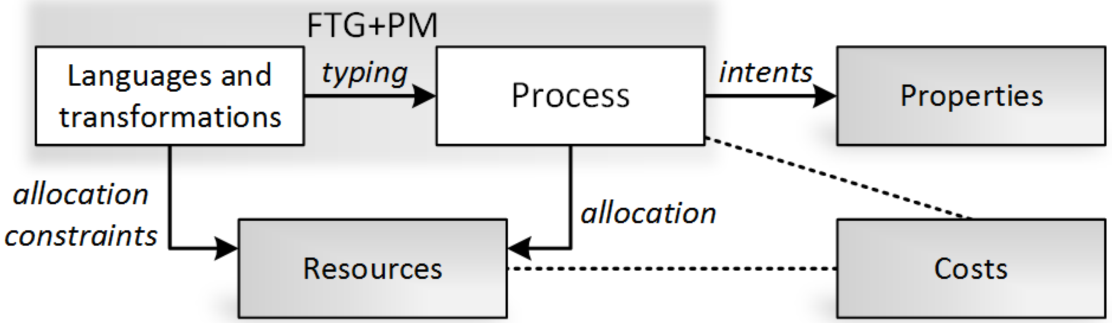
Performance metrics (business term: “KPI”)

- Typical performance metrics
  - Minimize cost ($$$)
  - Minimize queueing time
  - Maximize throughput
  - …

Material costs?
Resource costs?
License costs?
…

Multiple complex metrics!

PI in the FTG+PM



FTG+PM

Languages and transformations

typing

Process

intents

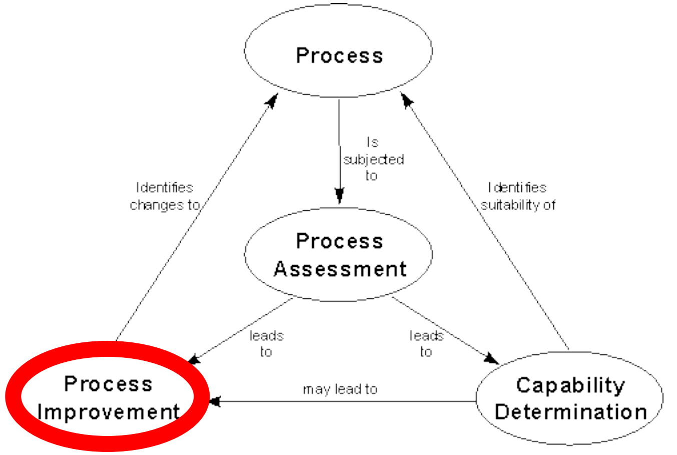
Properties

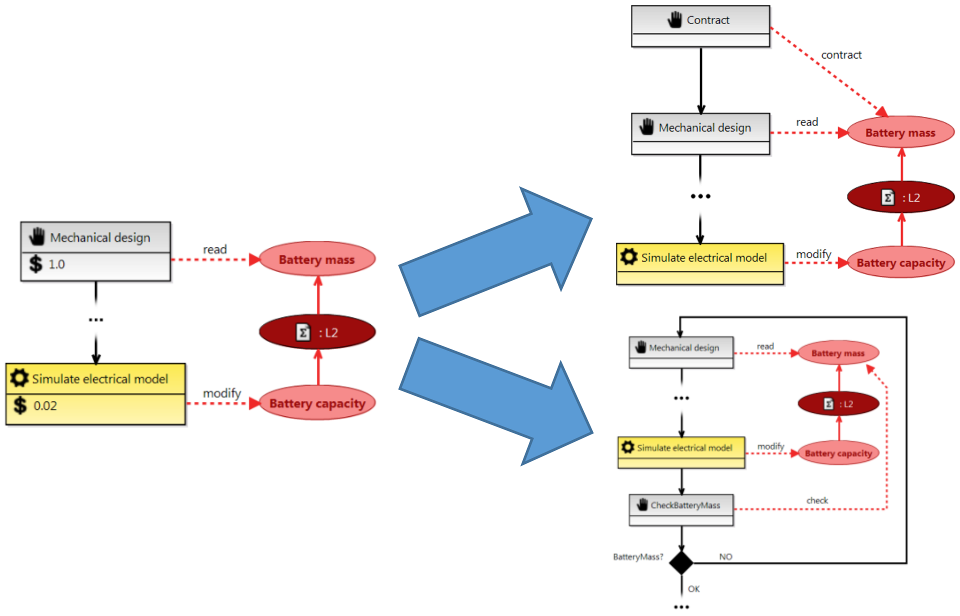
allocation constraints

allocation

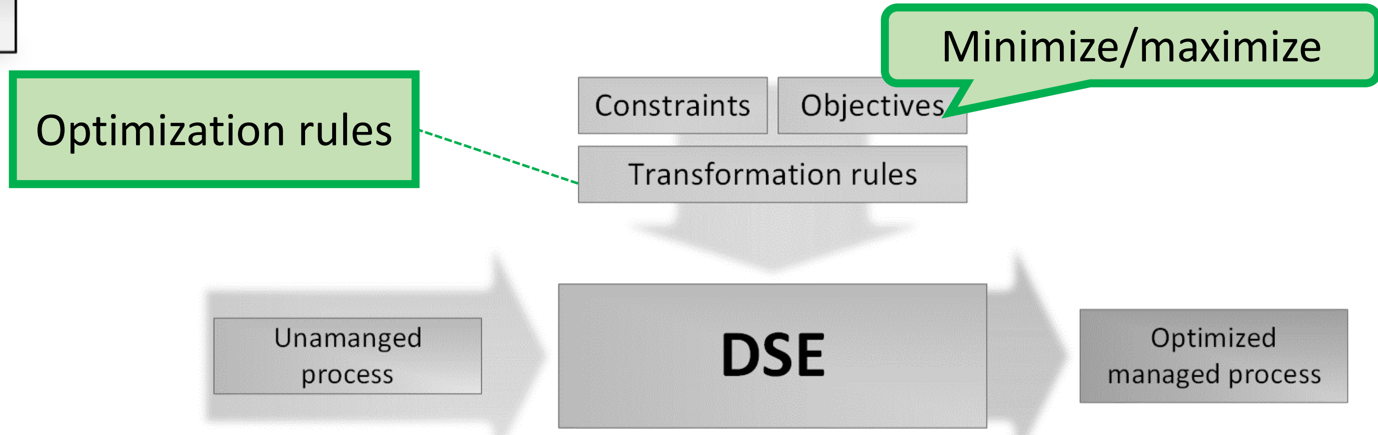
Resources

Costs





Rule-based multi-objective design space exploration (DSE)



Minimize/maximize

Optimization rules

Constraints

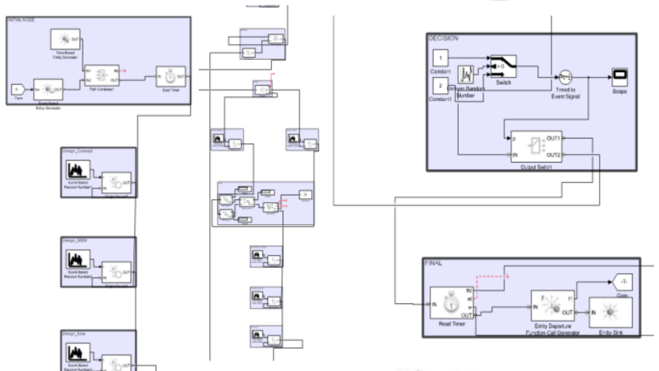
Objectives

Transformation rules

Unamanged process

DSE

Optimized managed process



SimEvents

Stochastic simulations: event queueing networks (EQN)

RCPSP: A special PI problem

- Resources of limited availability
- Activities of known durations
- Resource requests, linked by precedence relations
- Problem:
  - **find a schedule**
  - of **minimal duration**
  - by **assigning a start time** to each activity
  - such that the **precedence relations**
  - and the **resource availabilities** are respected
- NP-hard in the strong sense!

$$S_j - S_i \geq p_i \quad \forall (A_i, A_j) \in E$$

$$\sum_{A_i \in \mathcal{A}_t} b_{ik} \leq B_k \quad \forall R_k \in \mathcal{R},\ \forall t \geq 0$$

C. Artigues, S. Demassey, and E. Neron. Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications. ISTE, 2007.

Summary

- Process: level of abstraction matters
- Software processes as guidelines for SW Engineering
  - Waterfall, V Model, RUP, Agile
- Process modeling
  - Explicit modeling helps analysis, maintenance and improvement
- Lots of process modeling languages
  - Don’t reinvent the wheel!
- FTG+PM
- Software process improvement (SPI)