

PLM and Innovation
Excellence

Learning Campus

Your partner for
Business Learning

Siemens
Core
Learning
Program

Risk & Uncertainty

Author:
Rüdiger Kreuter, CT

Vision

For my part I know nothing with any certainty, but the sight of the stars makes me dream.

[Vincent van Gogh]



Source: http://commons.wikimedia.org/wiki/File:VanGogh-starry_night.jpg

Uncertainty Management

Learning objectives

- Understand Risk Management
- Be aware of kinds and sources of uncertainties
- Know how to manage uncertainties
- Be able to apply a set of appropriate measures

Risk & Uncertainty

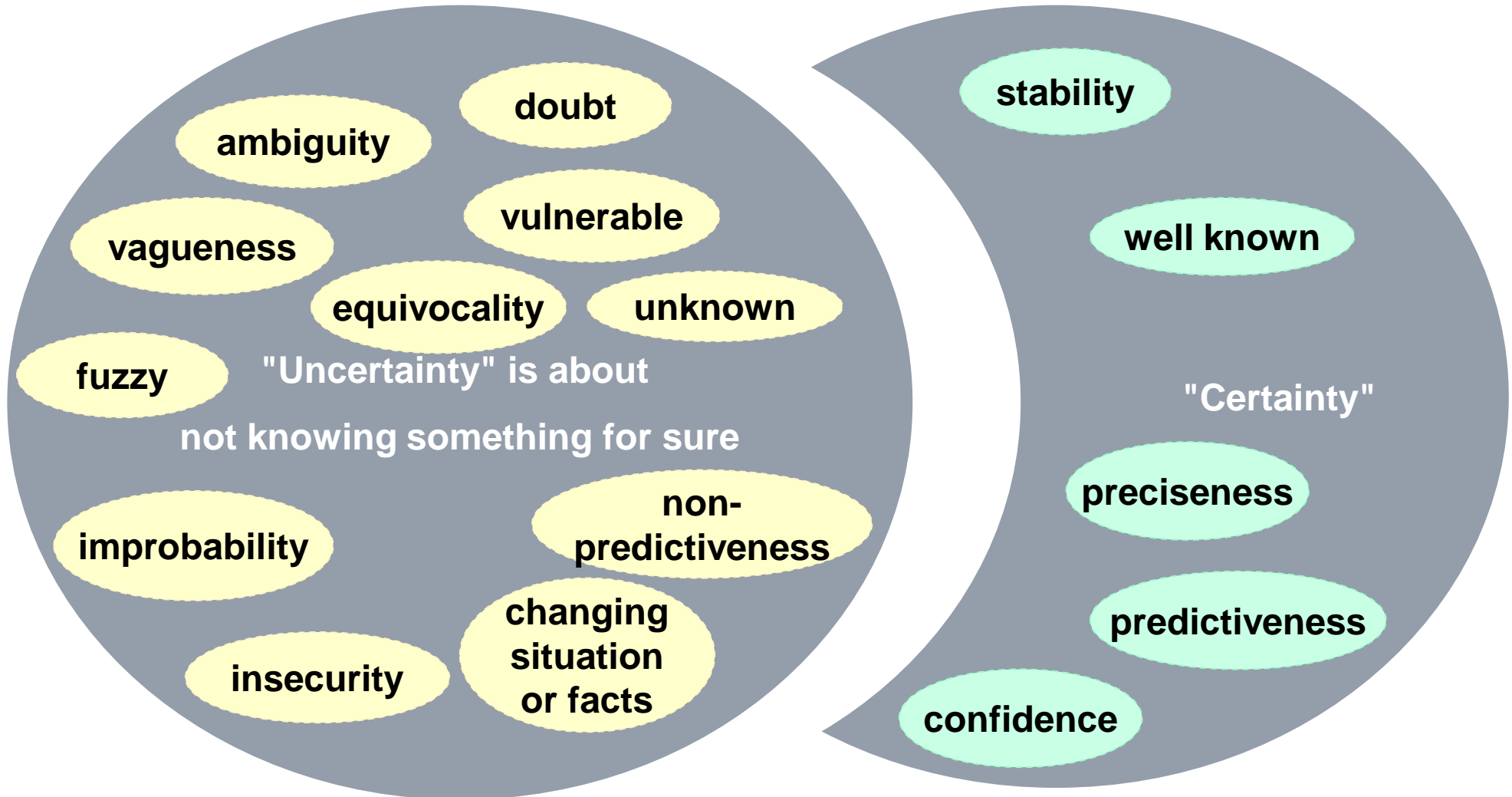
Agenda

**Dealing with missing, immature,
contradicting or changing information**

Risk Management

Summary

Uncertainty: Synonyms and Antonyms



Uncertainty Sources

resource
(capacity) shift

offshoring
issues

conflicting or
over-determined
information

patent
litigation

missing acceptance
(of innovative products
and solutions)

regulatory
changes

internal
resistance

distributed
development

disruptive / destructive
innovations

not invented here
syndrome

incomplete / old
documentation
(of used parts or
components)

supplies
in development

changing
market conditions

false or contradicting
requirements

unstable
interfaces

unclear
scope

outsourcing
issues

financial /
political crisis

changing
environment

errata in
supplier parts

communication
issues

changing / evolving
standards

missing / unknown
information

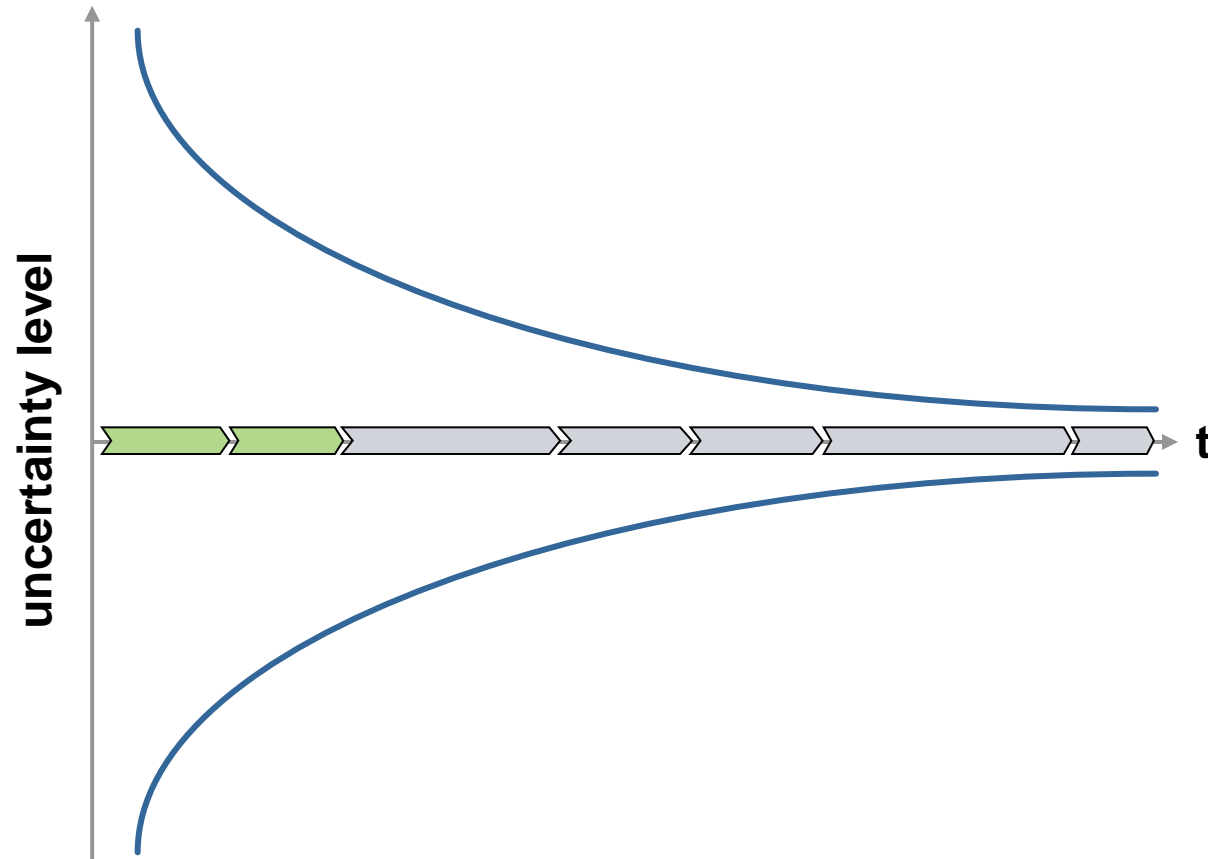
unknown or
not involved
Éminence grise

changing
requirements or
knowledge

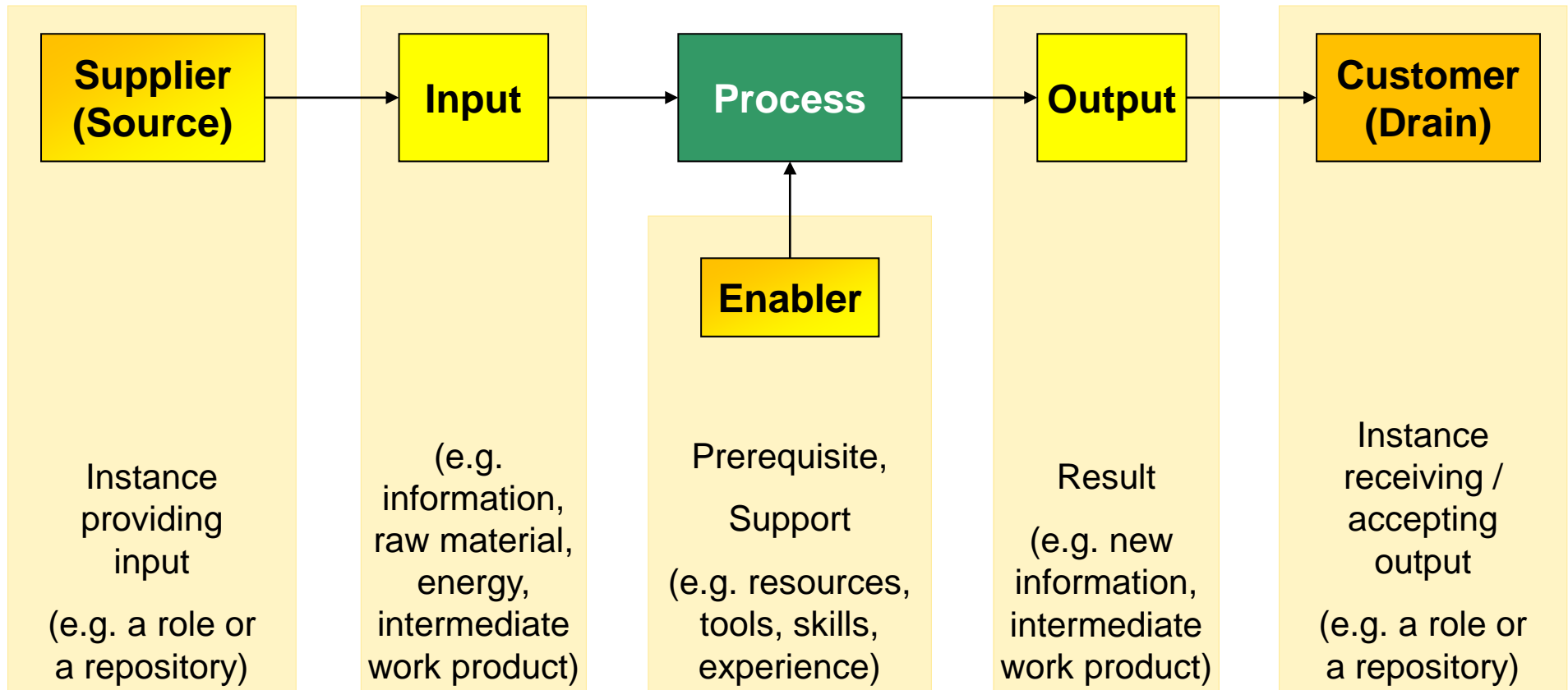
missing / non-identified
requirements

incorrect
assumptions

Cone of uncertainty

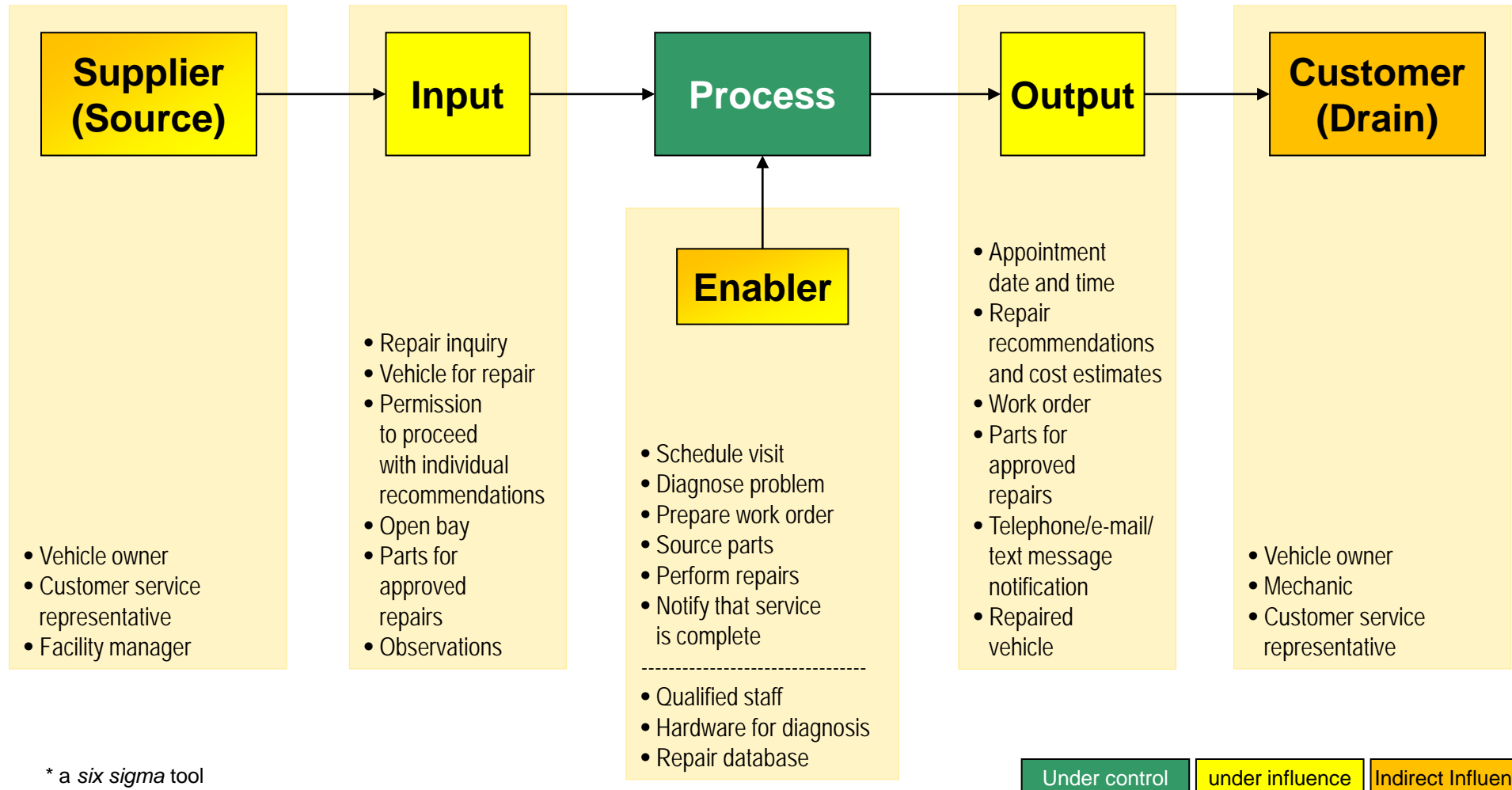


SIPOC* Model



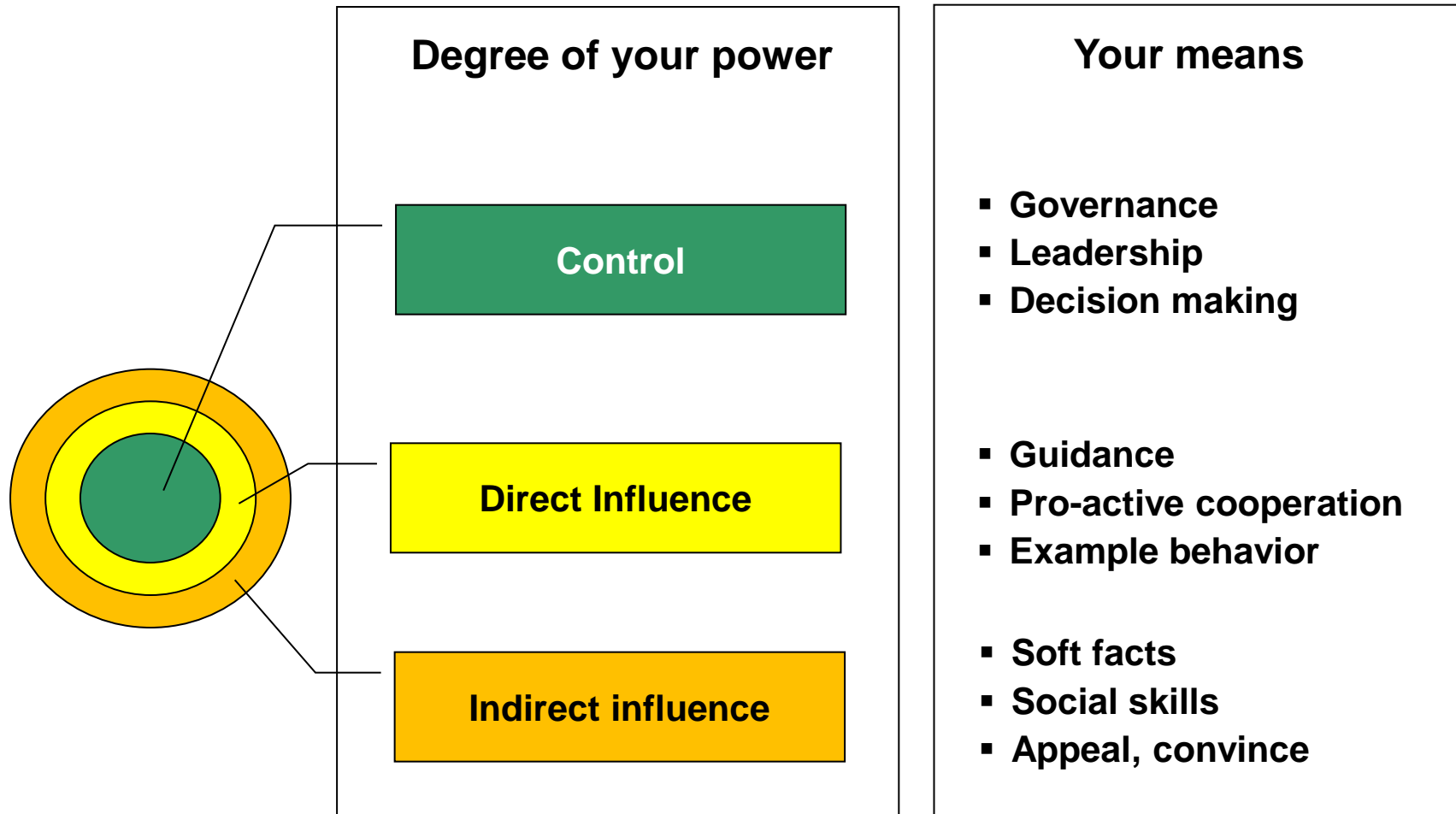
* a six sigma tool

SIPOC Model – the example from wikipedia



* a six sigma tool

Color coding & the circle of influence (more fine-grained version)



Certainty



Wanted → “Certainty”

Process steps transform a given input into intended work products, e.g. released documents or working (sub-)systems

Uncertainty



Reality → “Uncertainty”

Output may be impaired due to one or both of

- incomplete or wrong input
- incomplete or wrong processes

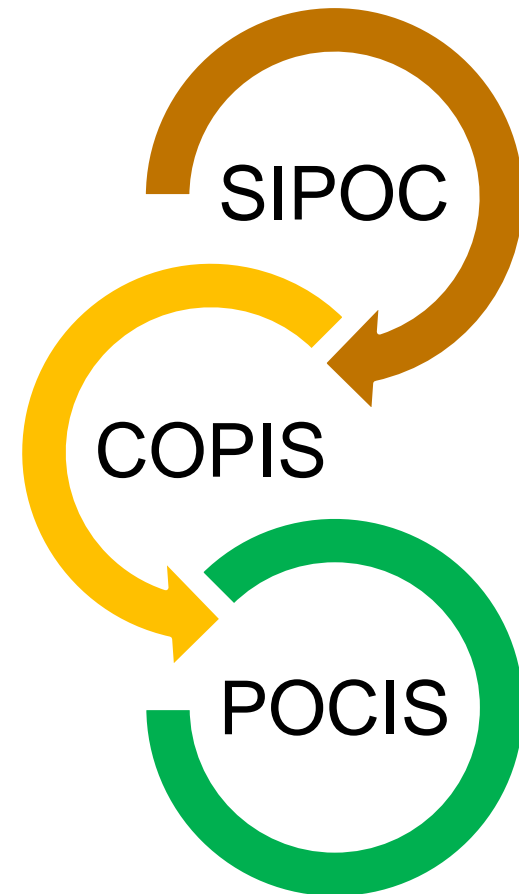
Unwanted deviations need to be limited or eliminated by **preventive** or **corrective** measures

Practical Usage (II)

SIPOC is an abbreviation to recall the elements of the model

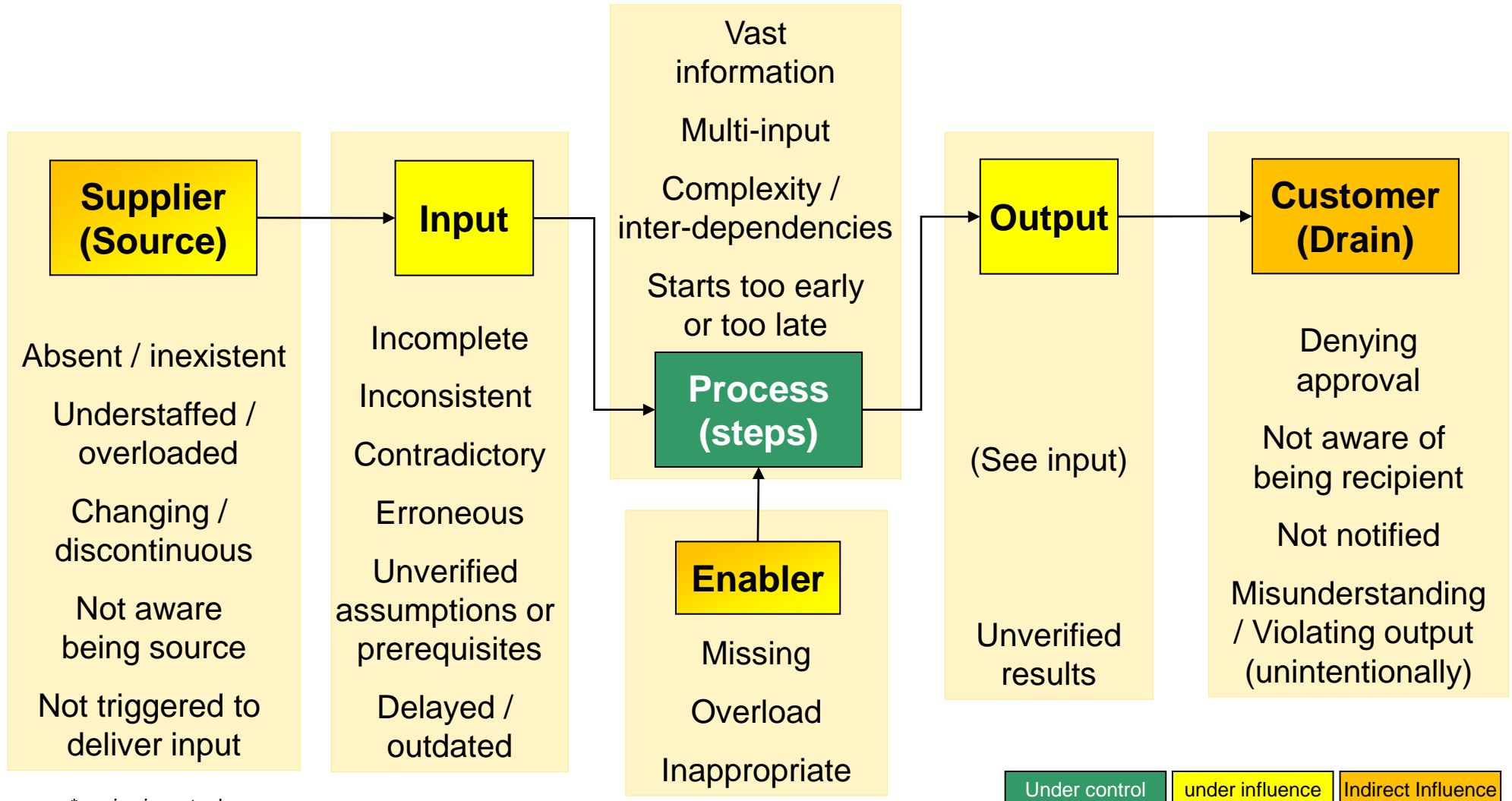
For practical usage it might be better to start with a customer oriented view and to reverse working on the process snapshot.

Even better is POCIS, which starts with the process itself, the outputs and then the customer, finalizing the process snapshot with Inputs and the Source



SIPOC* Model

Typical sources of trouble

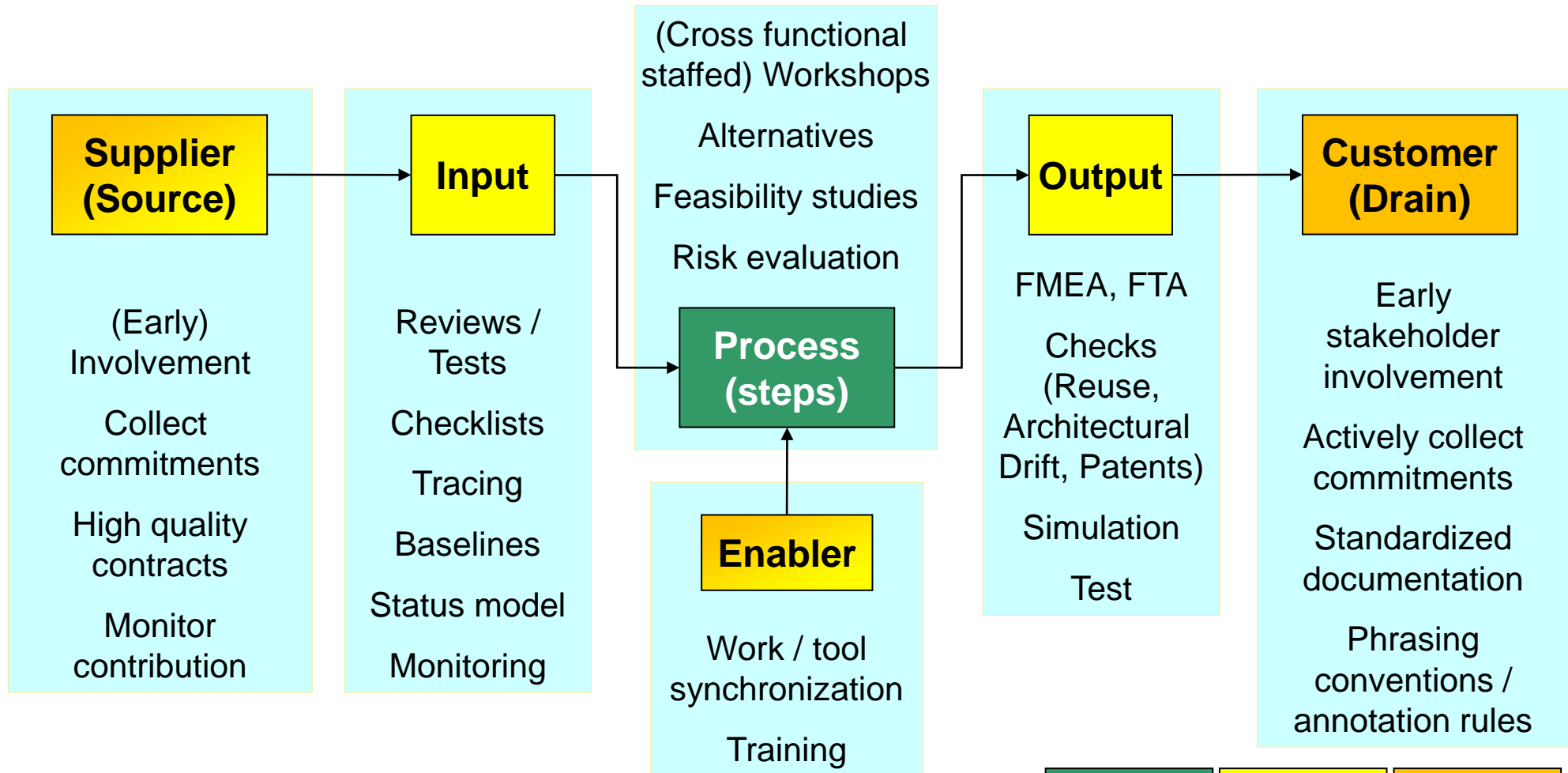


* a six sigma tool

Restricted © Siemens AG 2016-2017

SIPOC* Model

Typical approaches to deal with uncertainty sources



FMEA Failure Mode Effect Analysis
FTA Fault Tree Analysis

Under control

under influence

Out of control

* a six sigma tool

Risk & Uncertainty

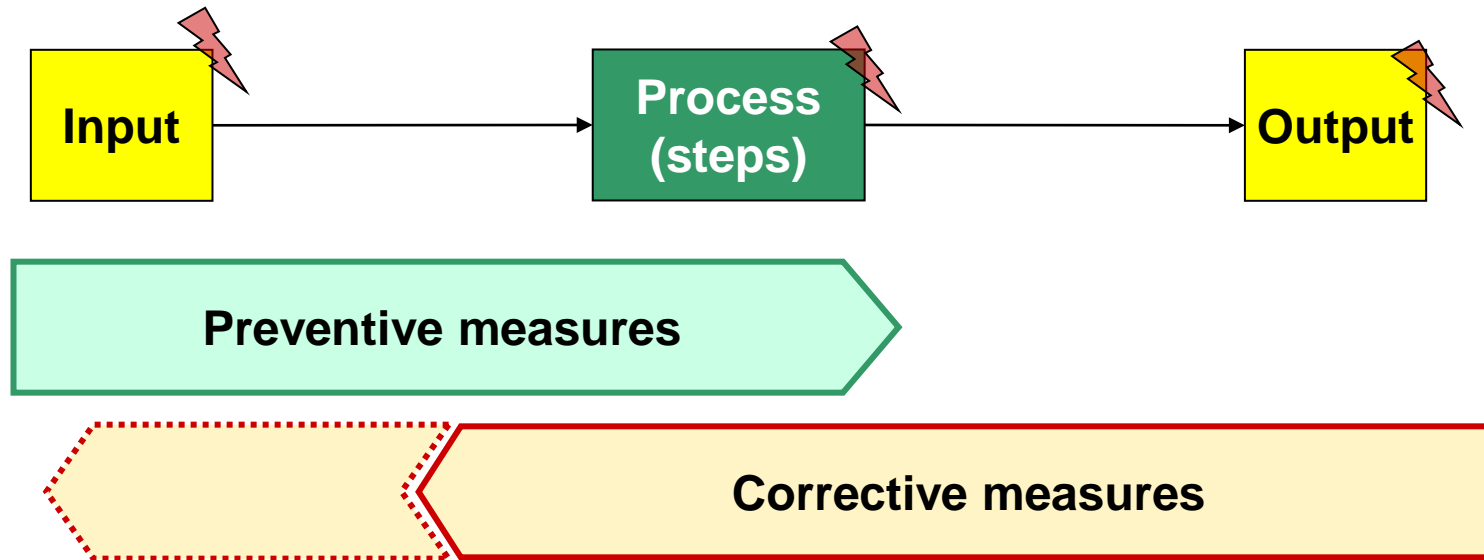
Agenda

Dealing with missing, immature,
contradicting or changing information

Risk Management

Summary

Preventive and Corrective measures



Preventive measures (predictive)

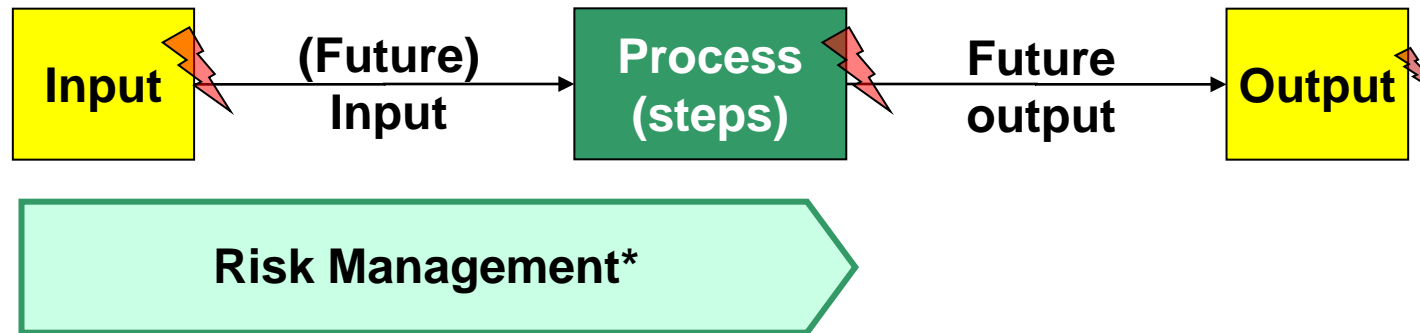
- Risk management
 - FMEA / FTA
 - Technical balancing
 - ...
- + mitigation actions

Corrective measures (reactive)

- (Peer) Reviews
 - Testing / Simulation
 - ...
- + corrective actions

FMEA = Failure Mode Effect Analysis
FTA = Fault Tree Analysis

Preventive Measures Risk Management



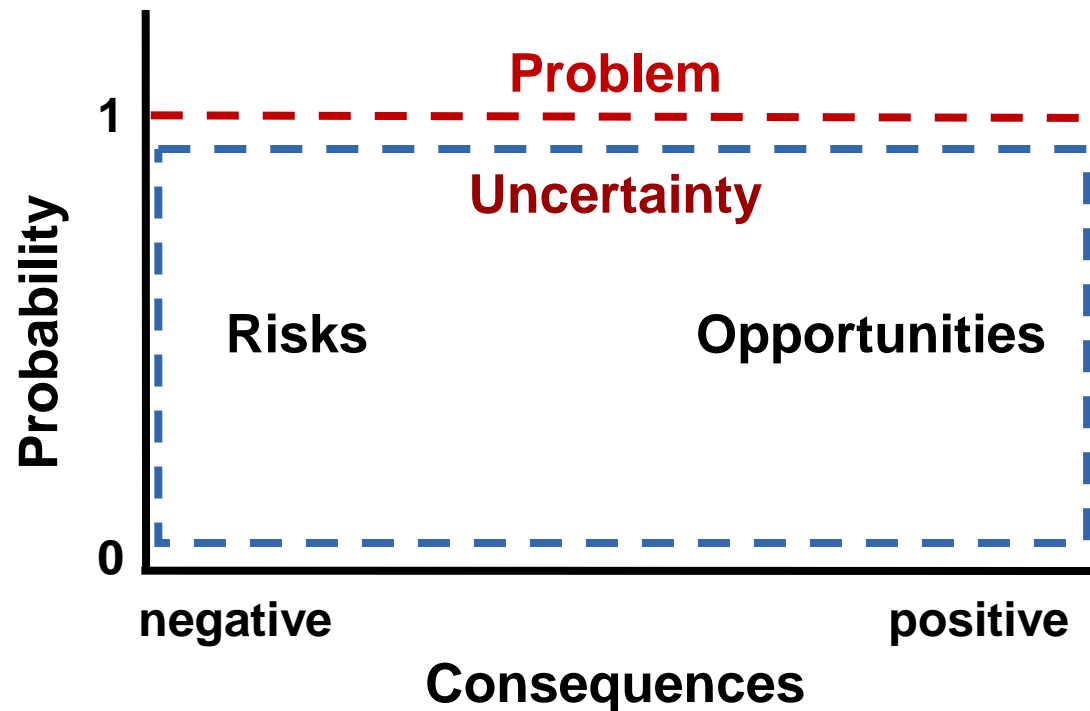
Risk Management**, within a development project, is a predictive and preventive measure, dealing with

- the early identification and understanding of possible problems, and
- the execution of mitigations (to prevent a risk or reduce the impact)
- or the preparation of reactive scenarios (in case the problem occurs)

* Risk Management is typically driven by the (Development) Project Manager; architects provide input, e.g. technical risks

** Risk Management has a partly different meaning in a solution project (PM@Siemens scope)

Uncertainty metrics

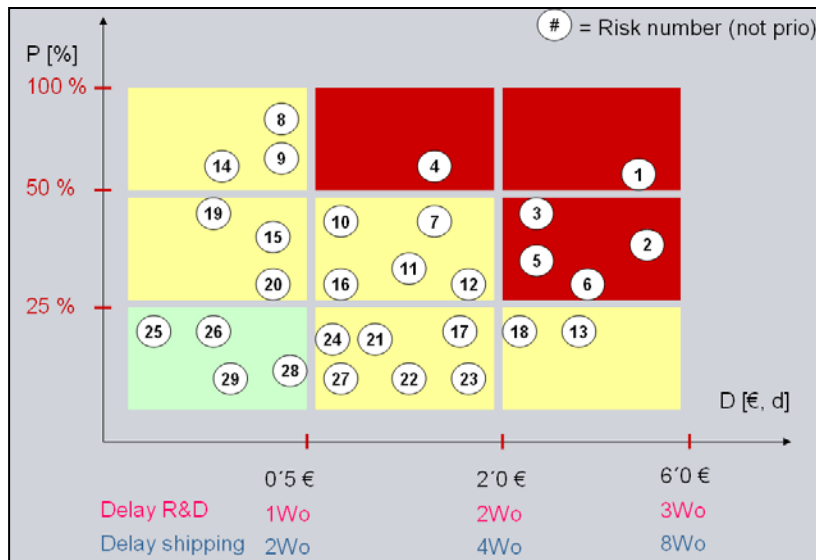


Not every risk
becomes a problem

Not every uncertainty
is a risk

Risk Management

A "Risk" is an incident with negative impact that,
based on a current uncertainty,
may or may not occur in the future;
in case it occurs, the risk becomes a "problem".



Risks identification includes an impact estimation:

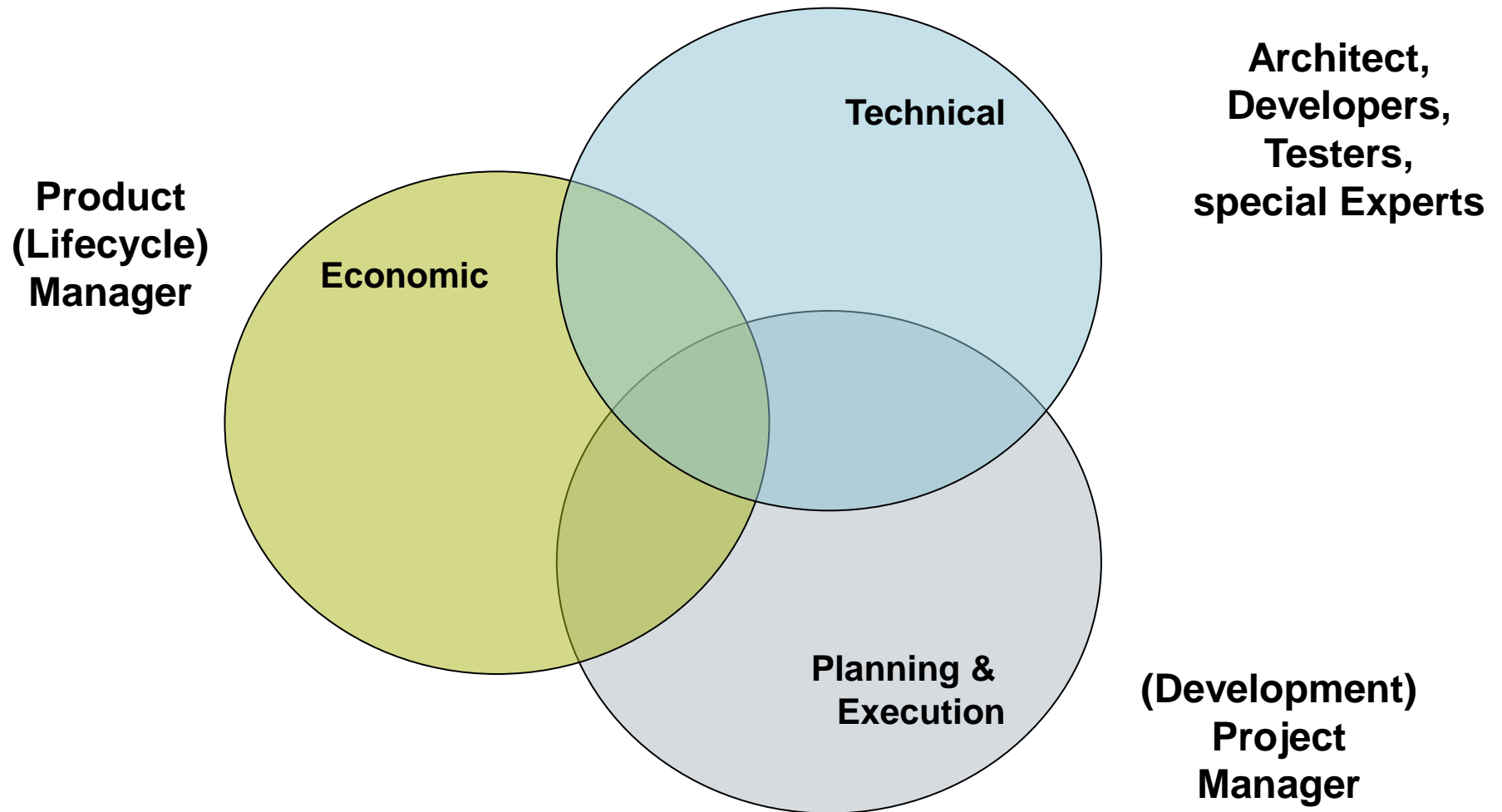
- a probability [P] / occurrence of an incident
- and a possible damage [D] / impact, usually in form of extra costs or time

Risks may change over time, i.e. they

- need monitoring, and in case
- adaptation to a changed situation

(Note: overdue clarifications are a problem – not a risk!)

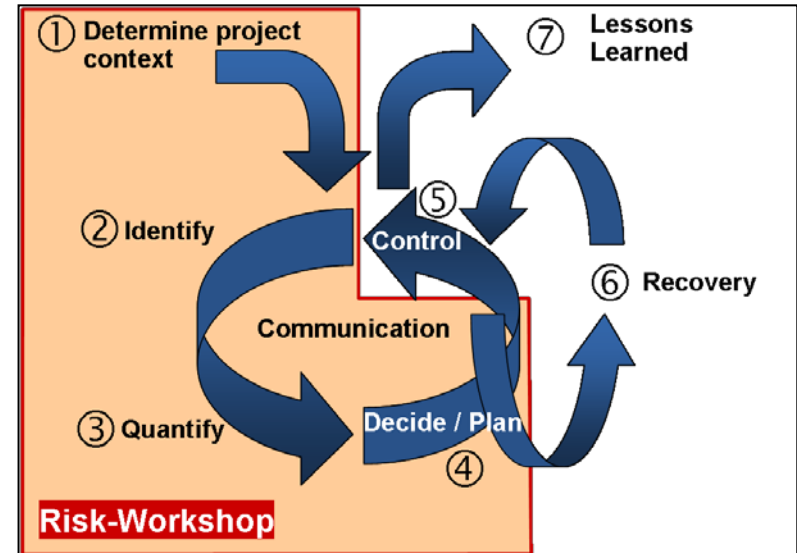
Typical risk (identification) responsibilities in a product development environment



Risk Management Process

Identify risks

- Determine scope & context and invite (appropriate) stakeholders
- Identify / elicit risks (e.g. using creativity methods)
- For each risk estimate probability and damage
- Elaborate and initiate measures
 - Mitigations*; i.e. immediate preventive action or
 - Reaction scenarios, if a risk becomes a problem
- Estimate remaining risk potentials after mitigations; repeat proceeding if necessary



Manage risks

- regularly monitor & update risk status
- if conditions changed >> update risk list and each risk's attributes
- if a risk became a problem >> react
- if a risk became obsolete >> delete it (or archive)

* check if mitigation costs exceed damage costs, and take this into account

General idea

Use risk management as a handshake between technology and business



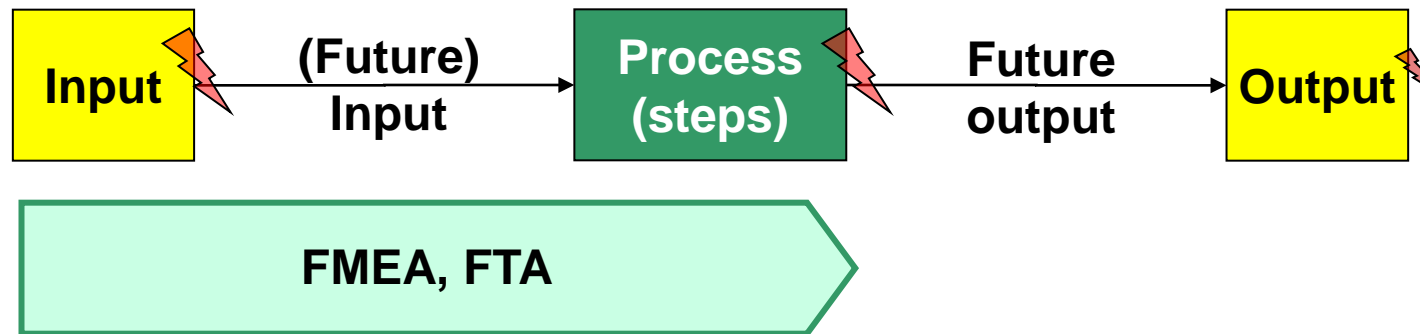
Risk Management Tip

Establish an organized approach to managing risks. One simple approach is to track risks the same way you track bugs. Any one can identify a risk and each risk is tracked until it is no longer a risk. Risks are prioritized and reviewed when their status changes or when there is new information. This helps remove emotion from the discussion and makes it easier to remember to re-evaluate them periodically.

From: 35/97 things every software architect should know



Preventive Measures Technical Risks



Risk analysis methods like FMEA can be used, to identify the ways in which a system can fail, including severity and consequences of each failure.

} Risk-based test strategy

An FTA or a Cause-Effect Analysis are used to find the root causes of a top-level failure.

} Root cause analysis (RCA)

FMEA	Failure Mode and Effect Analysis
FTA	Fault Tree Analysis

Risk analysis: FMEA

(Failure Mode and Effect Analysis)

FMEA is a formalized analytical method, used by an interdisciplinary team, to find

- potential failures
- their potential effects
- and their potential causes

in products and processes, for deriving the resulting risks and if meaningful mitigation measures.

FMEAs are used to

- enhance functional safety and reliability of products and processes
- reduce costs by failure prevention
- estimate cost risks during guaranty time frame

FMEA

Process and Template

SIEMENS										FMEA (Failure Mode and Effect Analysis)											
Object of analysis:					Department:					Date:											
					Core team:					(last editing of the contents)											
		potential			actual state						recommended state					new state					
No.	Characteristic (Part / Step)	Failure Mode	Effect of Failure	Cause of Failure	Prevention / Detection	O	S	D	RPN	Date	Action recommended	O	S	D	RPN	Responsibi- lity / Date	Action taken	O	S	D	RPN
1									0				0		0				0		0
2									0				0		0				0		0

Basic FMEA process – always performed by teams

- Define the scope of analysis and build up the (interdisciplinary) team
- Decompose the object of analysis into manageable elements
- Elicit potential failure modes for each element (use creativity methods)
- Elaborate and describe effects and causes for each failure mode
- Estimate and assign O, S, D rankings for each failure mode
- Calculate RPN (= O x S x D) for each failure mode
- Elaborate recommended (mitigation) actions for failure modes with unacceptable RPNs
- Take actions
- Estimate new O / S / D ratings for each failure mode
- Calculate new RPN for each failure mode

O	Occurrence rating
S	Severity rating
D	Detection rating
RPN	Risk Priority Number

FMEA

How to use the ratings

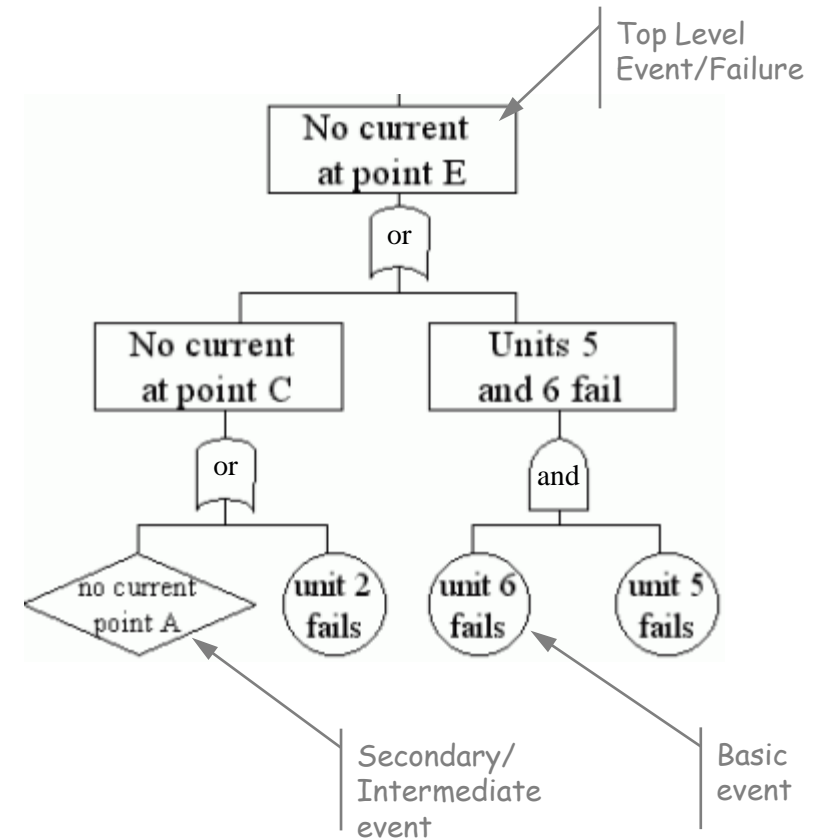
Occurrence		Severity		Detection	
Evaluation	Explanation	Evaluation	Explanation	Evaluation	Explanation
1	No Errors Almost no errors occurring	1	Minimum Error Effect The customer will probably not notice the error	1	High The errors will be inevitably discovered
2	Very Minor Rare; Design corresponds to proven designs. Process is statistically controlled	2	Minor Error Effect The errors are insignificant, and the customer will probably not be concerned	2	Moderate The error is apparent.
3		3		3	
4		4		4	
5	Minor Occasionally; from comparable solutions it is well-known that these errors will occur occasionally	5	Moderately Severe Error Effect The errors are causing the customer to be discontented. The customer feels troubled due to the error or is annoyed.	6	Low The error can be discovered by sample inspections or similar actions
6		6		7	
7		7		8	
8	Moderate Design and/or process is problematic; comparable solutions caused repeated errors	8	Severe Error Effect The errors are very annoying to the customer	9	Very Low The error is difficult to find
9		9		10	
10	High It is known that this error will occur frequently or in large quantities	10	Very Severe Error Effect These errors can cause serious operational failure (9) or customers could come to harm (10)		None The features can not be checked and the errors can not be detected (e.g. inaccessible, over life span)

FTA – Fault Tree Analysis

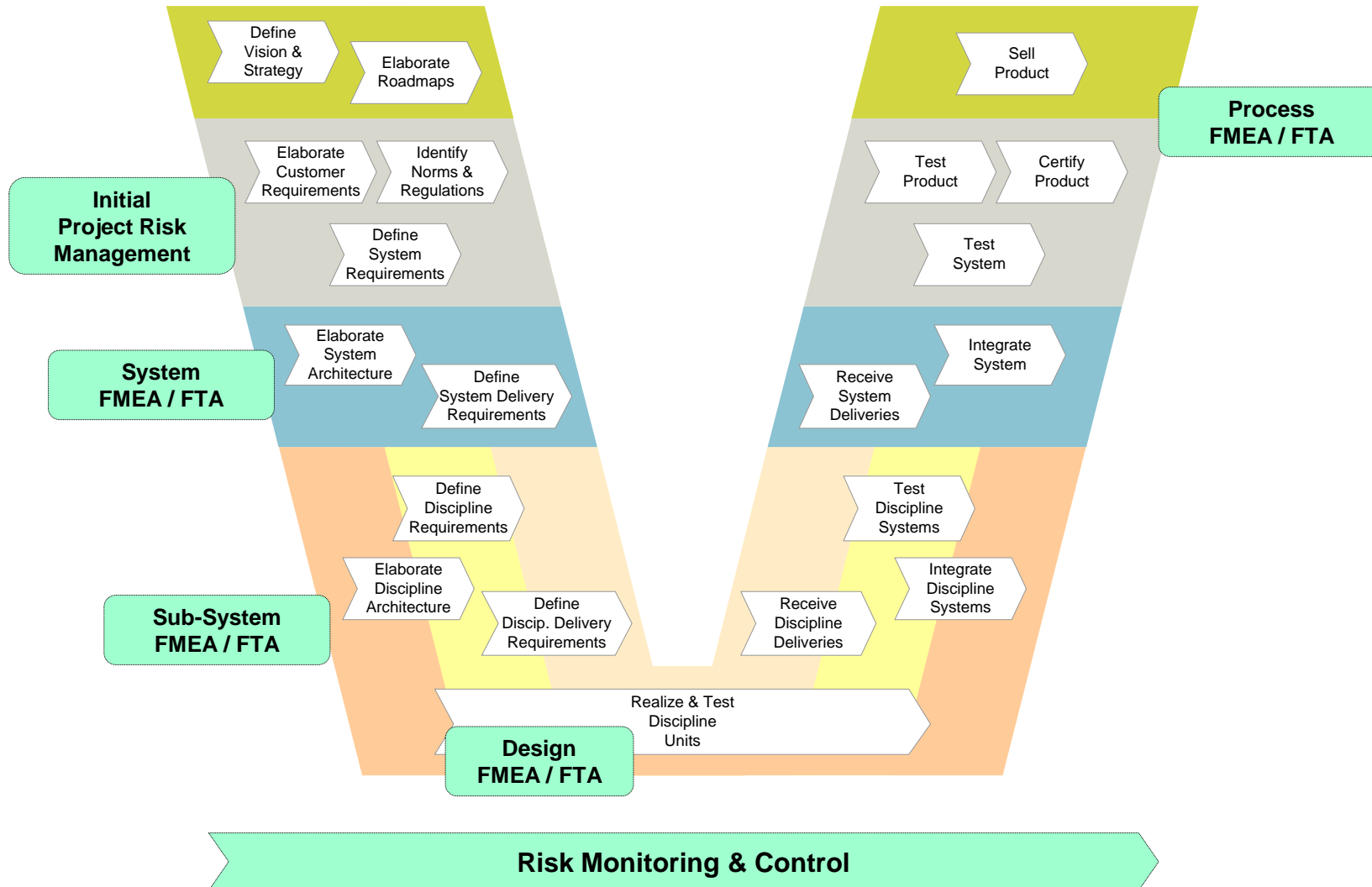
A Fault Tree Analysis starts with the top level failure and decomposes recursively the root causes for the fault.

There are different methods and tools to draw fault trees. Usually logic symbols, like "And", "OR", "XOR", ... are used to merge the root causes.

During the analysis ensure to dig down to the real root causes; e.g. by using techniques like a 5-Why Analysis.



When to do Risk Analysis (process view)



Risk & Uncertainty

Agenda

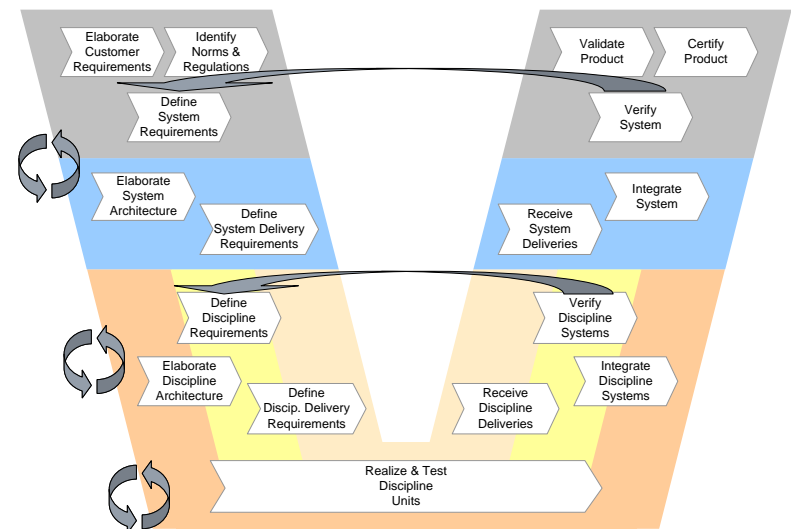
**Dealing with missing, immature,
contradicting or changing information**

Risk Management

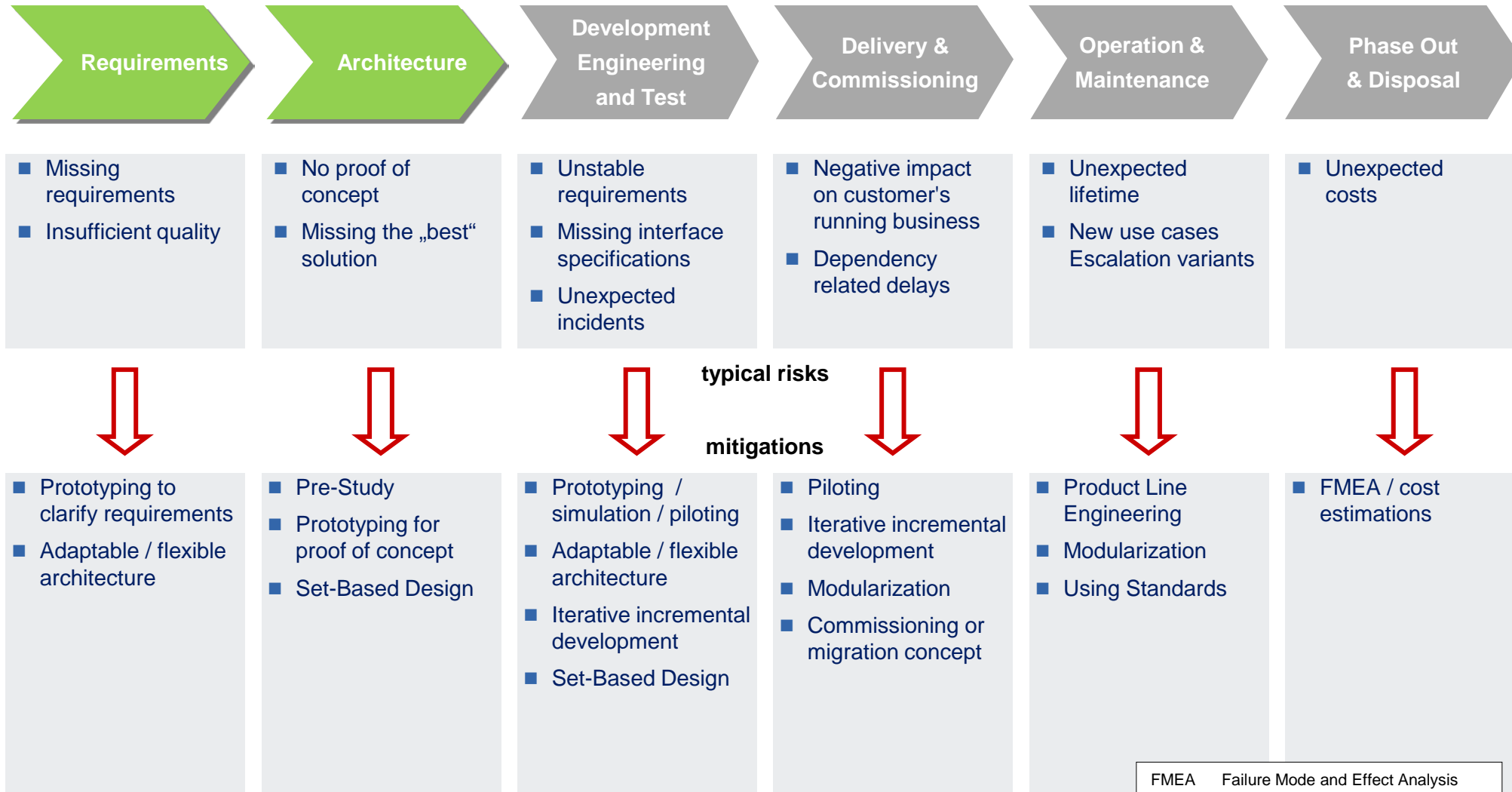
Summary

Some approaches to deal with uncertainty

- Clear goal before project start
- Standardized processes, methods, guidelines, glossary or ontology
- Stakeholder and early (preferred) supplier involvement
- (Professional) requirements engineering
- Situation analysis
- Domain models, system scope definition
- Use of proven solutions or knowledge
- Design to <flexibility, cost, manufacturing, ...>
- Set-Based Design, solution variants
- Modeling and simulation
- Peer and stakeholder reviews
- (Risk based) testing
- Iterative / incremental development (in short cycles)
- ...



Major architecting related risks across the lifecycle and appropriate mitigations



What we have learned

There are many uncertainty sources across the product or solution development lifecycle.

Risk management techniques, including FMEA / FTA are methods to deal with uncertainties.

But, not every uncertainty is a risk!

There are risk mitigation methods to prevent damage and pre-planned reaction scenarios to deal with problems, once they show up.

Not all uncertainties are under control or influence of Architects.



A departing thought

Iterative development projects are not easier to setup, to plan, or to control just because they are iterative.

The project manager will actually have a more challenging task, especially during his or her first iterative project, and most certainly during the early iterations of that project, when risks are high and early failure is possible.


[Philippe Kruchten, 2000]



Further readings

Use the SSA Wiki :
<https://wiki.ct.siemens.de/x/fReTBQ>

and check the “Reading recommendations”:
<https://wiki.ct.siemens.de/x/-pRgBg>

- 
- **Architect's Resources:**
 - Competence related content
 - Technology related content
 - Design Essays
 - Collection of How-To articles
 - Tools and Templates
 - Reading recommendations
 - Job Profiles for architects
 - External Trainings
 - ... more resources

Backup

Backup

Preventive and Corrective measures

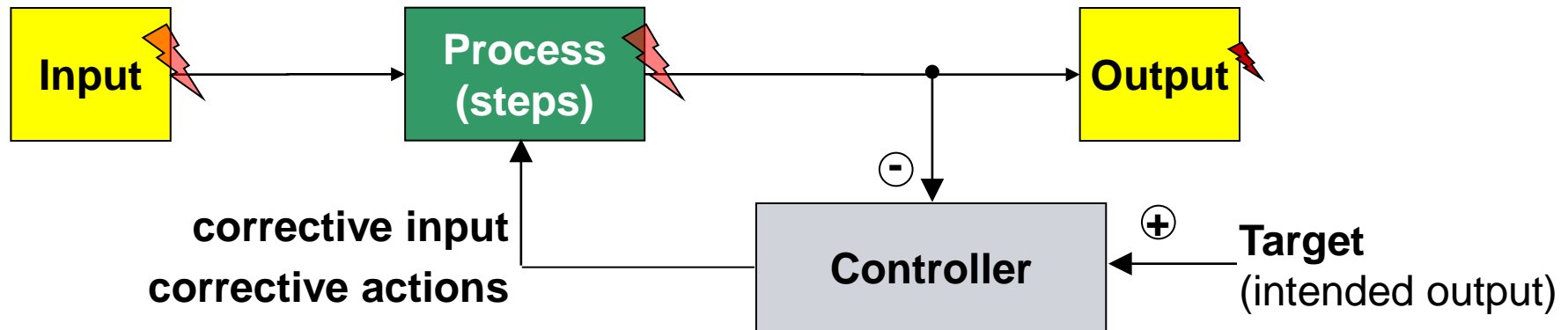


Corrective measures (reactive)

- (Peer) Reviews
 - Testing / Simulation
 - ...
- + corrective actions

FMEA = Failure Mode Effect Analysis
FTA = Fault Tree Analysis

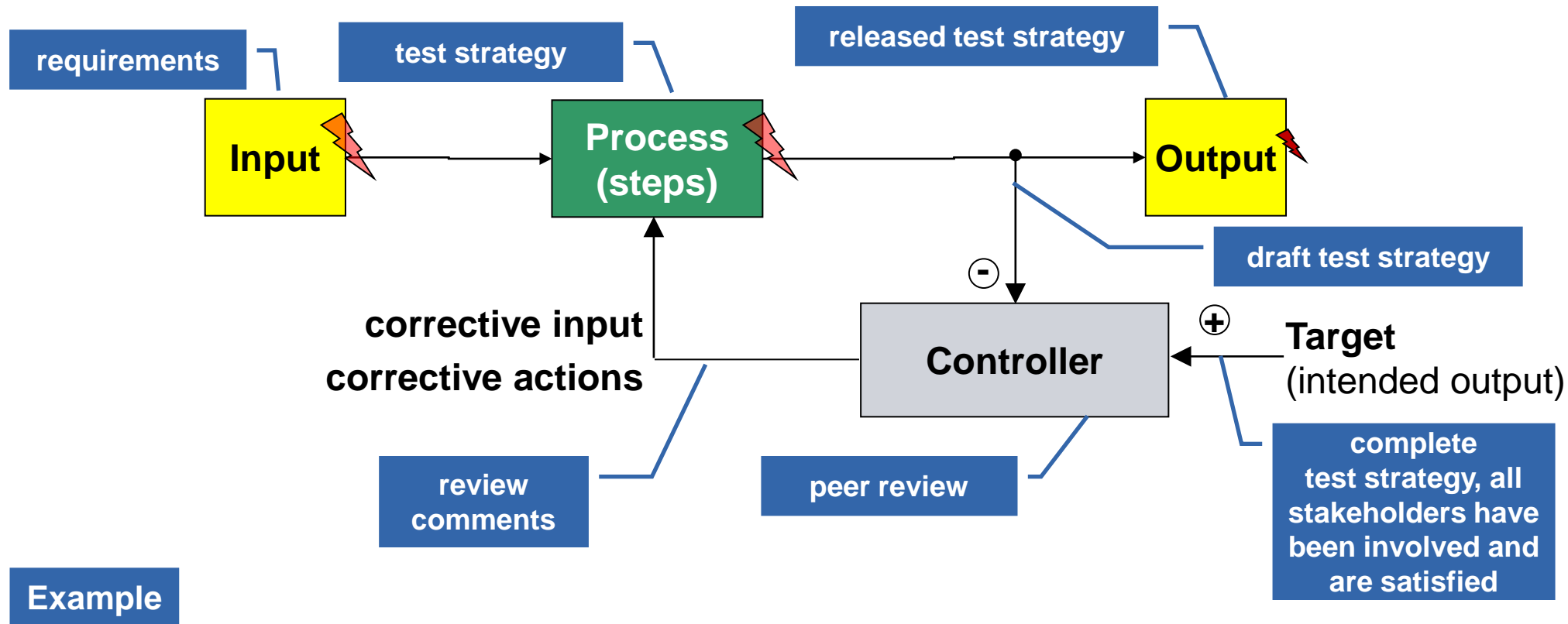
Corrective measures



Corrective measures → Control loop (real time action)

- Output may be impaired due to disturbance of input or activities.
- Controller verifies the current output against intended output; e.g. by using reviews, tests, or simulation
- Corrective input as counter measure in order to minimize the impact of the disturbance.

Example: Corrective measures



Example

Requirements and constraints are collected, but the safety expert was busy and not involved.

- The peer review reveals that important topics of the most up to date safety regulations had not been considered
- The test strategy is released only after all (high level) review comments have been incorporated.

Increasing reality using “in the loop simulation”

“in the loop” simulations are used in the context of control systems development, where a control SW (“controller”) interacts with a mechatronic system (“plant”).

Model in the Loop (MiL)

- using a model of the control to work with a model of the plant
- extremely fast development; possible to make small changes to the control model and immediately test the system.

Software in the Loop (SiL)

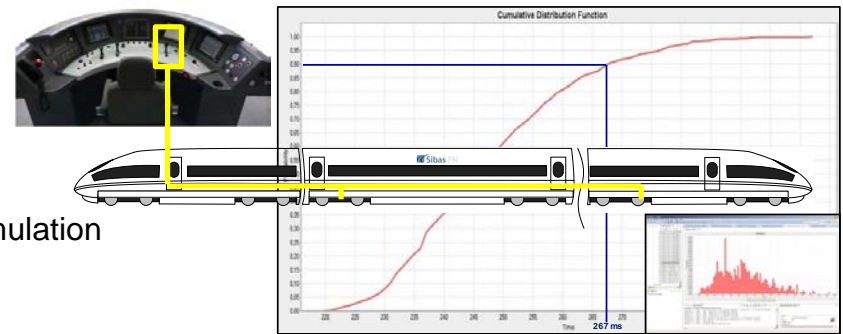
- coded control SW in simulation environment connected to plant simulation
- coding failures or issues start to become evident
- design iteration slows down slightly from MiL

Processor in the Loop (PiL)

- controller SW deployed to a representative microprocessor, coupled via a high speed bus with plant simulation
- execution issues on the embedded processor are easy to find and fix; e.g. sufficient performance of the embedded processor
- design iteration slows noticeably; changes must be coded and deployed to the control system;

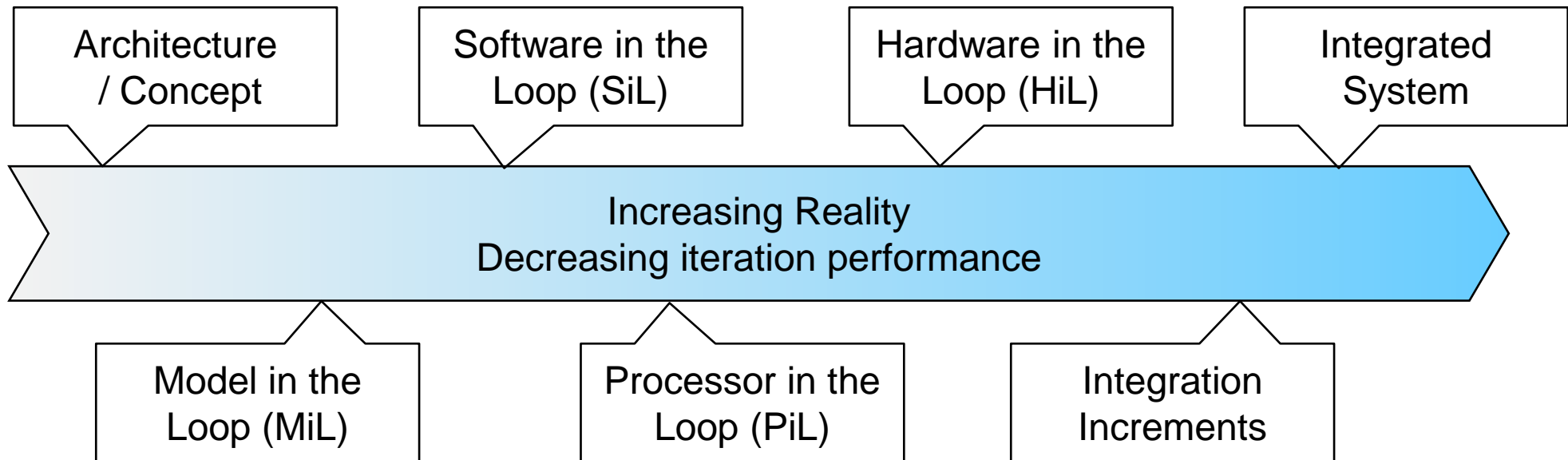
Hardware in the Loop (HiL)

- real control system connected with plant model, running on a real-time computer, through proper controller IO
- often used only for software validation rather than development as the design iteration is very slow at this point; however, this test is closest to the final application and therefore exposes most of the problems that will be seen.
- fidelity of the plant model and the test vectors determine the only difference between the final application and the HiL environment



Increasing Reality
Decreasing iteration performance

Increasing reality using “in the loop simulation”



Case study (Motivation) Interacting problems

What happens, if several problems occur simultaneously?

- Probability?
- Does the severity of consequences compound?
- does your mitigation measures still work in the combined case?

Your ideas?

Case study: Fukushima Disaster

Failure Scenario	Consequence	Mitigation Measure
Failure of cooling systems	Reactor meltdown	Redundant cooling systems
Blackout	No electricity for running the cooling systems	Use diesel generators as backup
Earthquake	Structural damage	Structural Reinforcement
Flood	Water damaging the support/safety systems	Perimeter walls
Tsunami	Damage of perimeter walls and flooding	Reinforce the perimeter walls

But...

what happens if more than one failure scenario happens at the same time?

Case study: Fukushima Disaster a multi failure scenario ...



Massive earthquake with a tsunami resulted in:

- Structural damage of the reactor, safety and backup systems:
 - Leaks in main reactor resulted in higher coolant needs
- Emergency Response affected
 - Physical obstructions
 - Evacuation of personal and surrounding population
- Massive blackout that lasted several days
 - Need to run the diesel generators for a longer period
- Massive disruptions in regional supply systems
 - Difficulty to resupply diesel

***This compounded failure resulted in a cooling system break down,
because of structural damages and lack of diesel ...
... and finally fires and reactor meltdown***