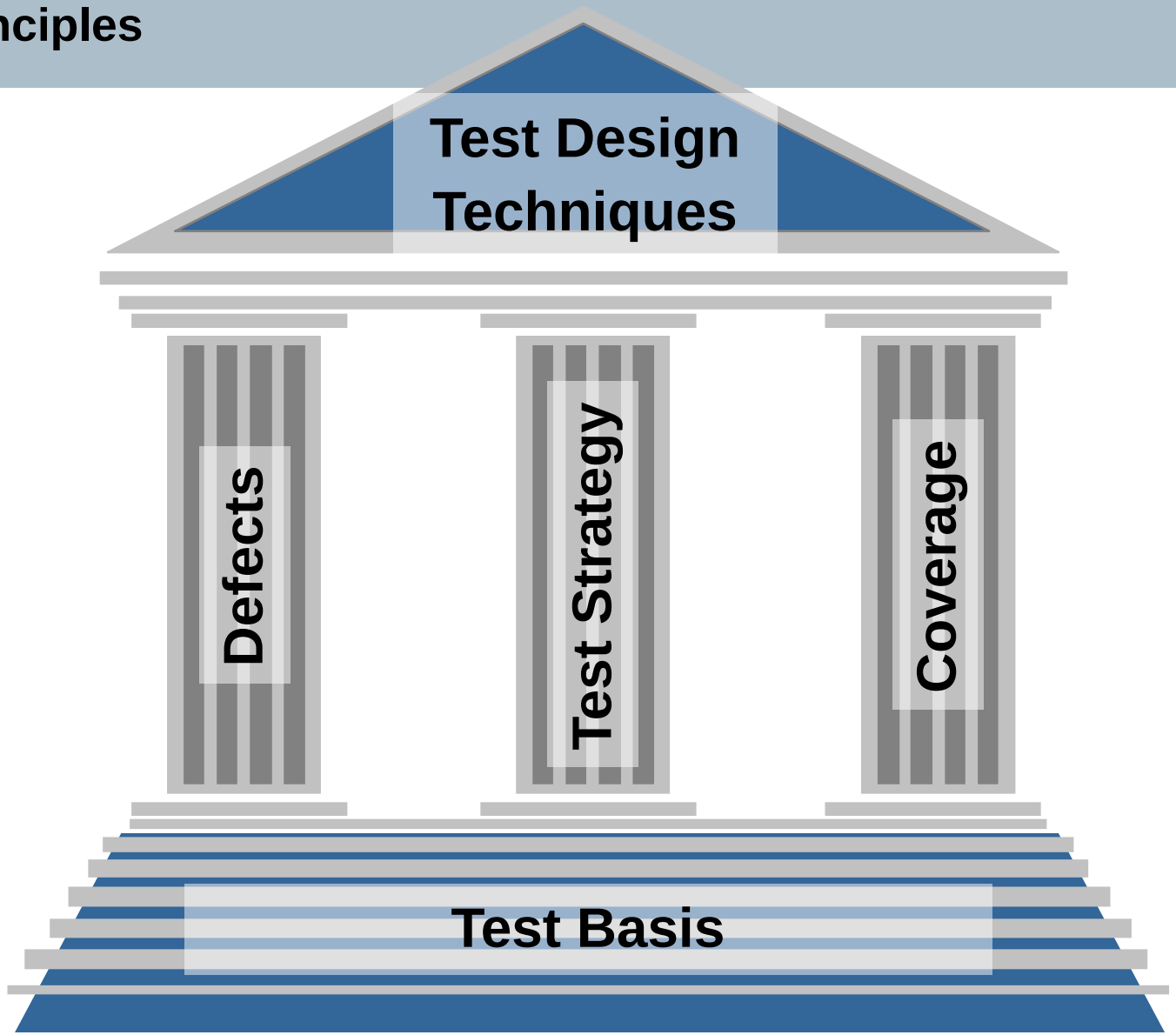
Test design techniques
Key principles

SIEMENS



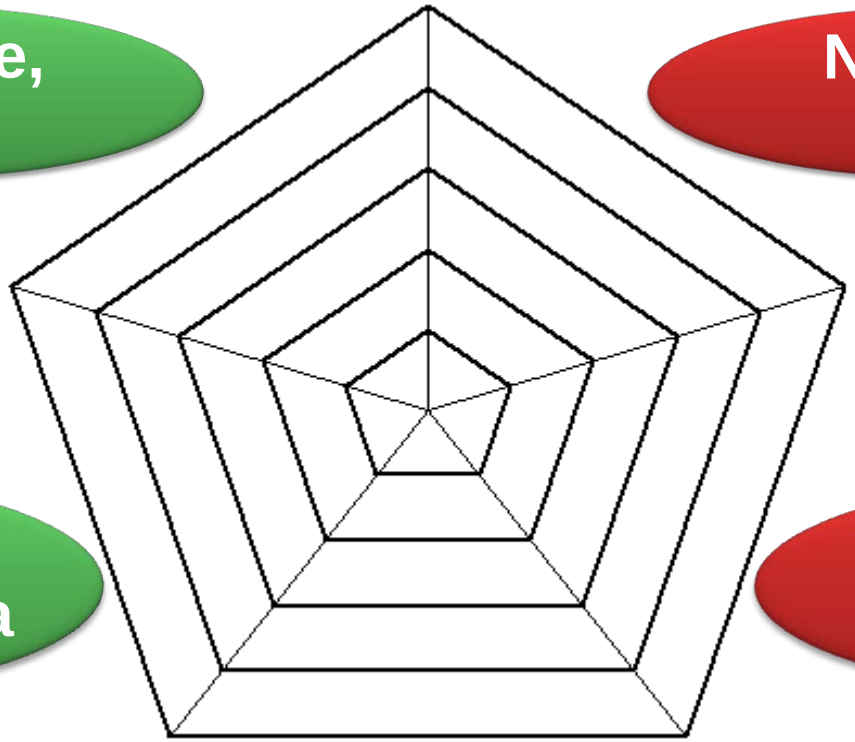
SIEMENS

Categories of test design techniques

Regression

Positive, valid

Negative, invalid



Grey-box

Fault-based

Models, interfaces, data

Internal structure, paths

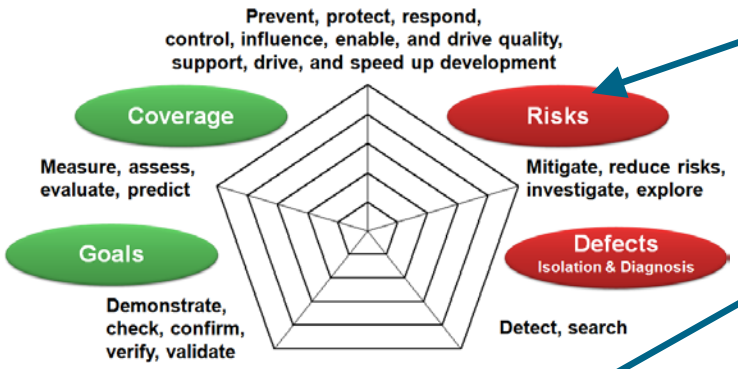
Black-box

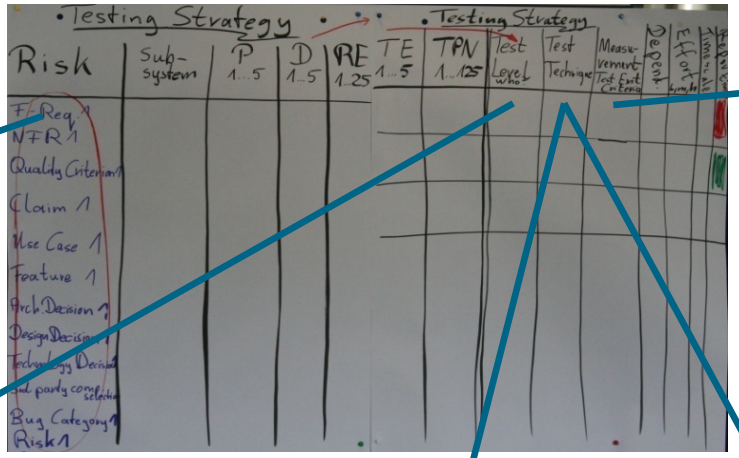
White-box:
control flow, data flow

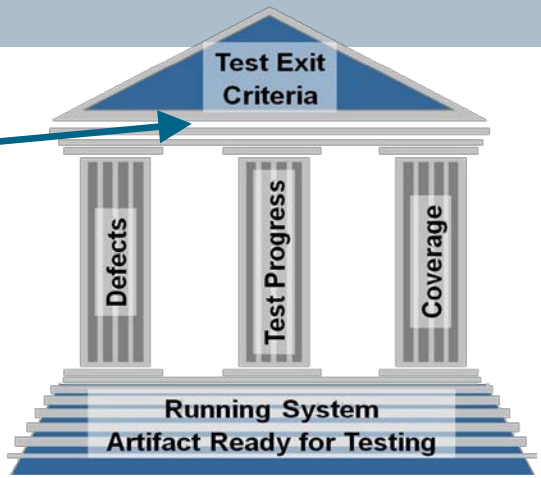
Summary: Building blocks of a *Risk-based Test Strategy (RBT)*

SIEMENS

Why do we test? Dimensions of testing

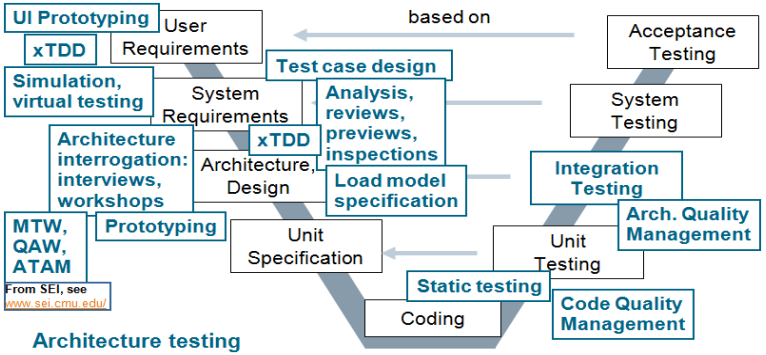






SIEMENS

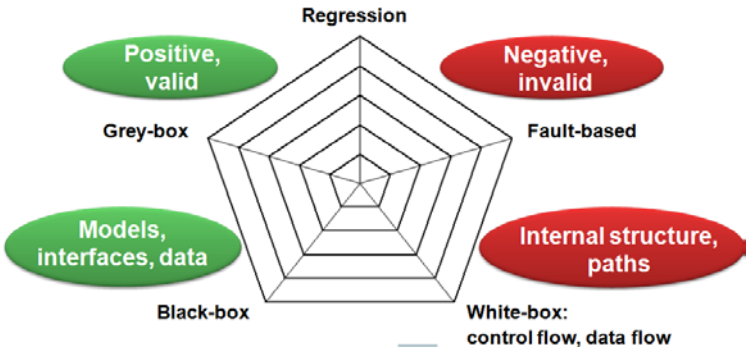
Test levels – Example V model with architecture testing

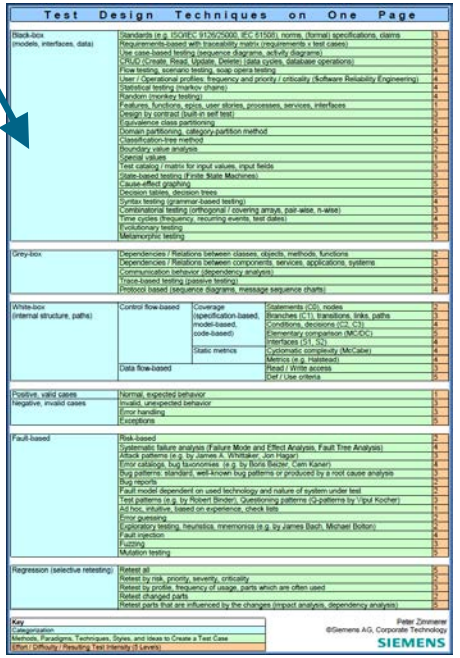


Architecture testing
is any testing of architecture and architectural artifacts
→ mapping of architectural risks to test levels

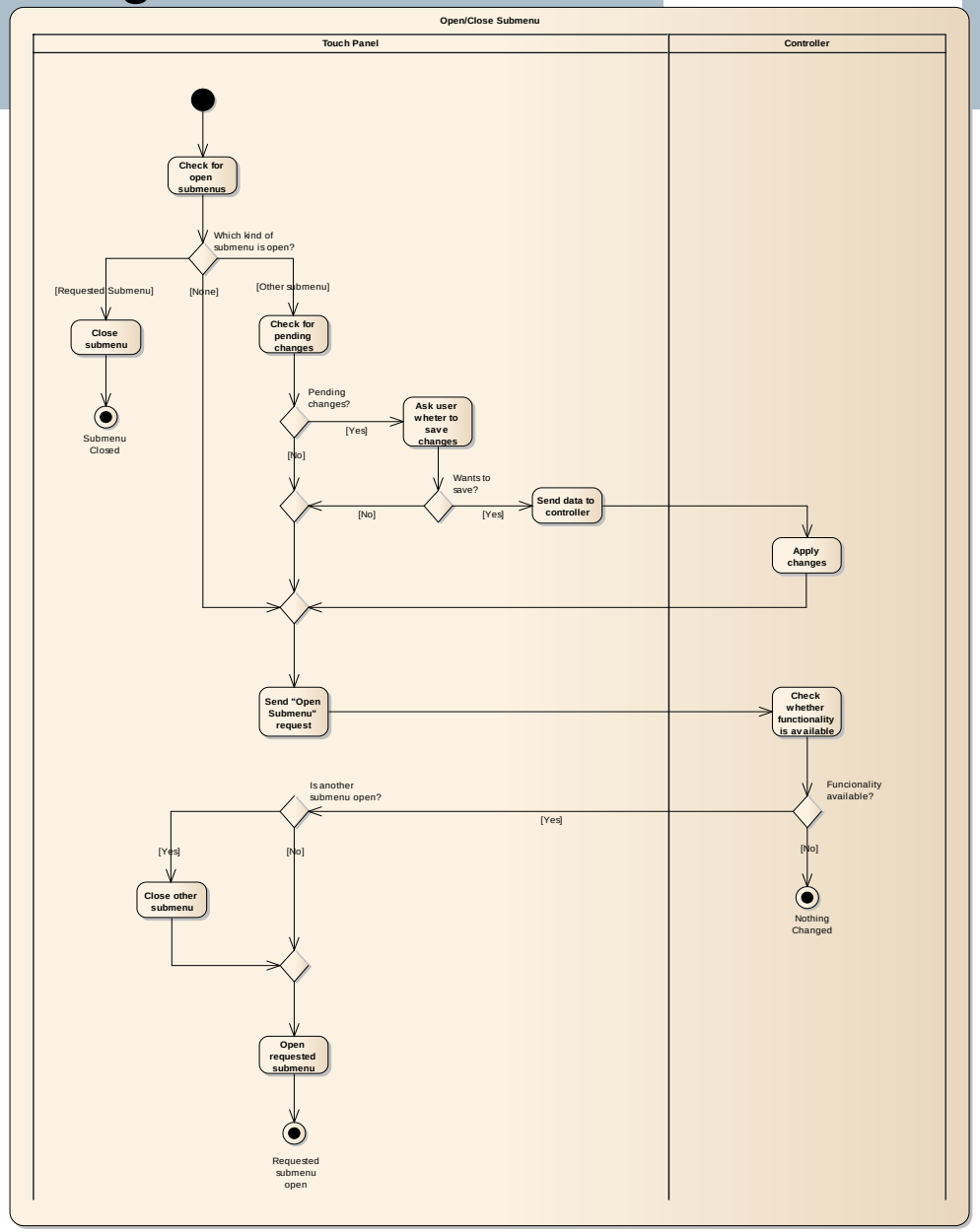
SIEMENS

Categories of test design techniques

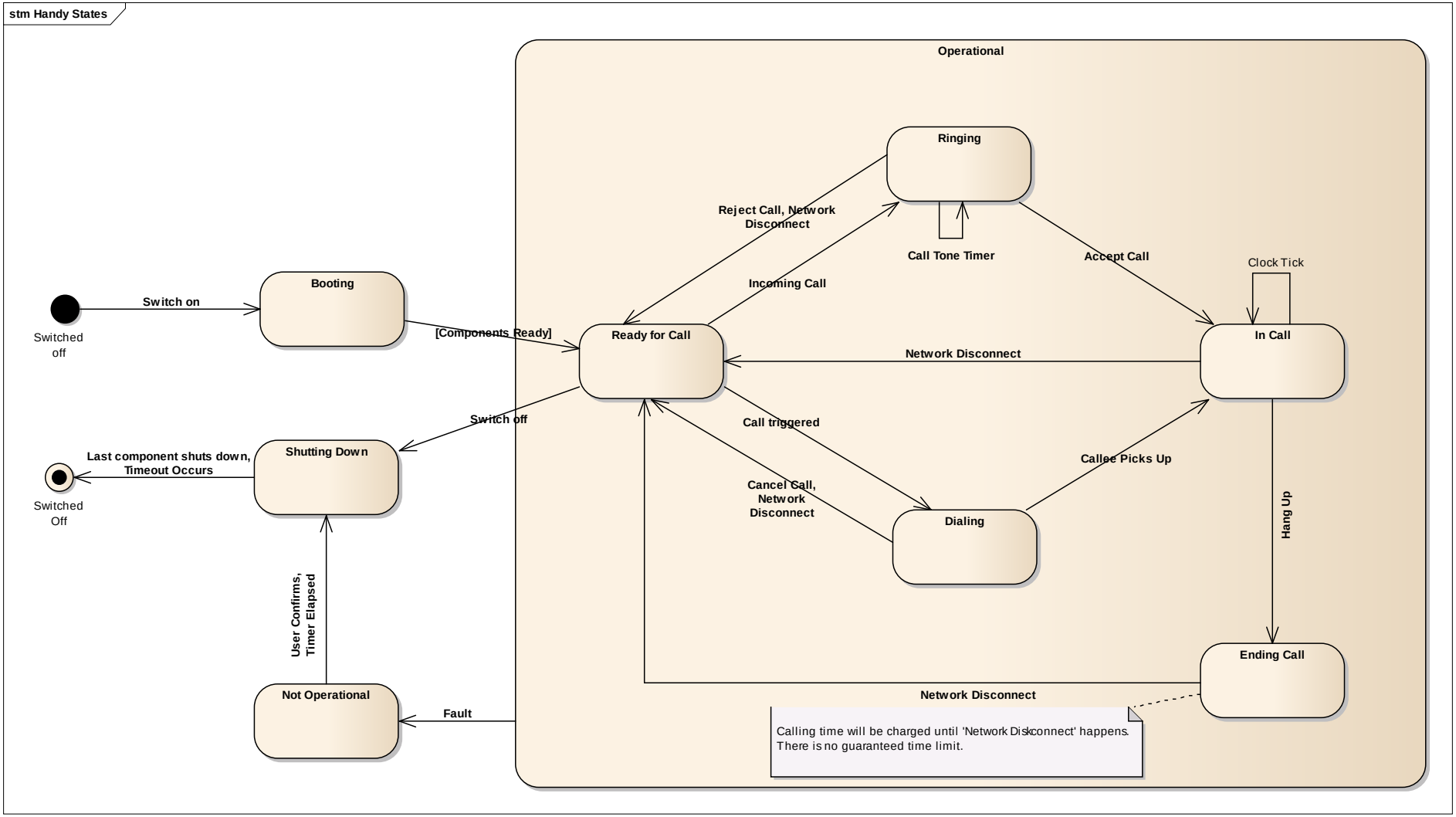




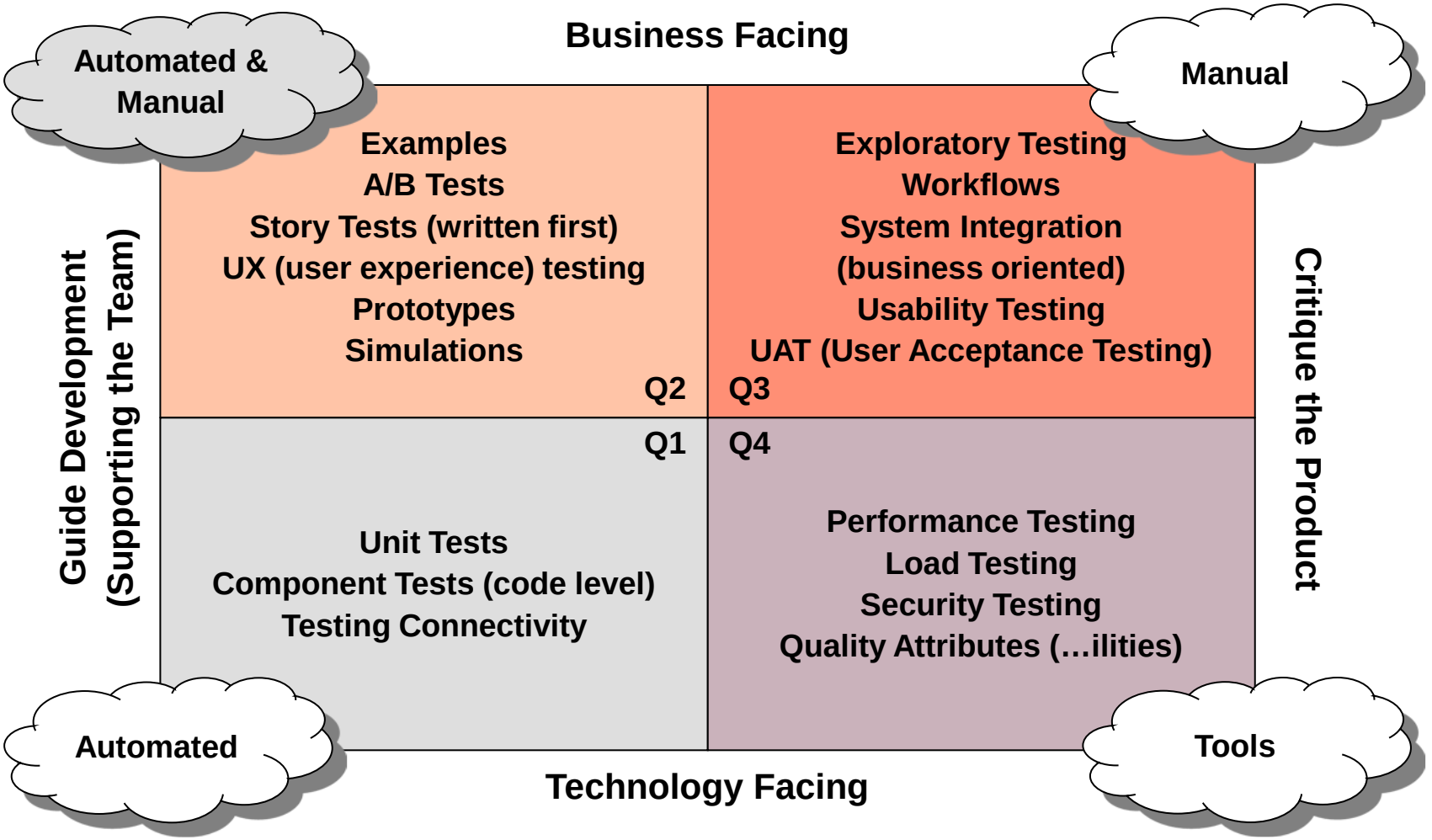
Use case-based testing – Scenario testing
Example: Activity diagram



Example: Phone system

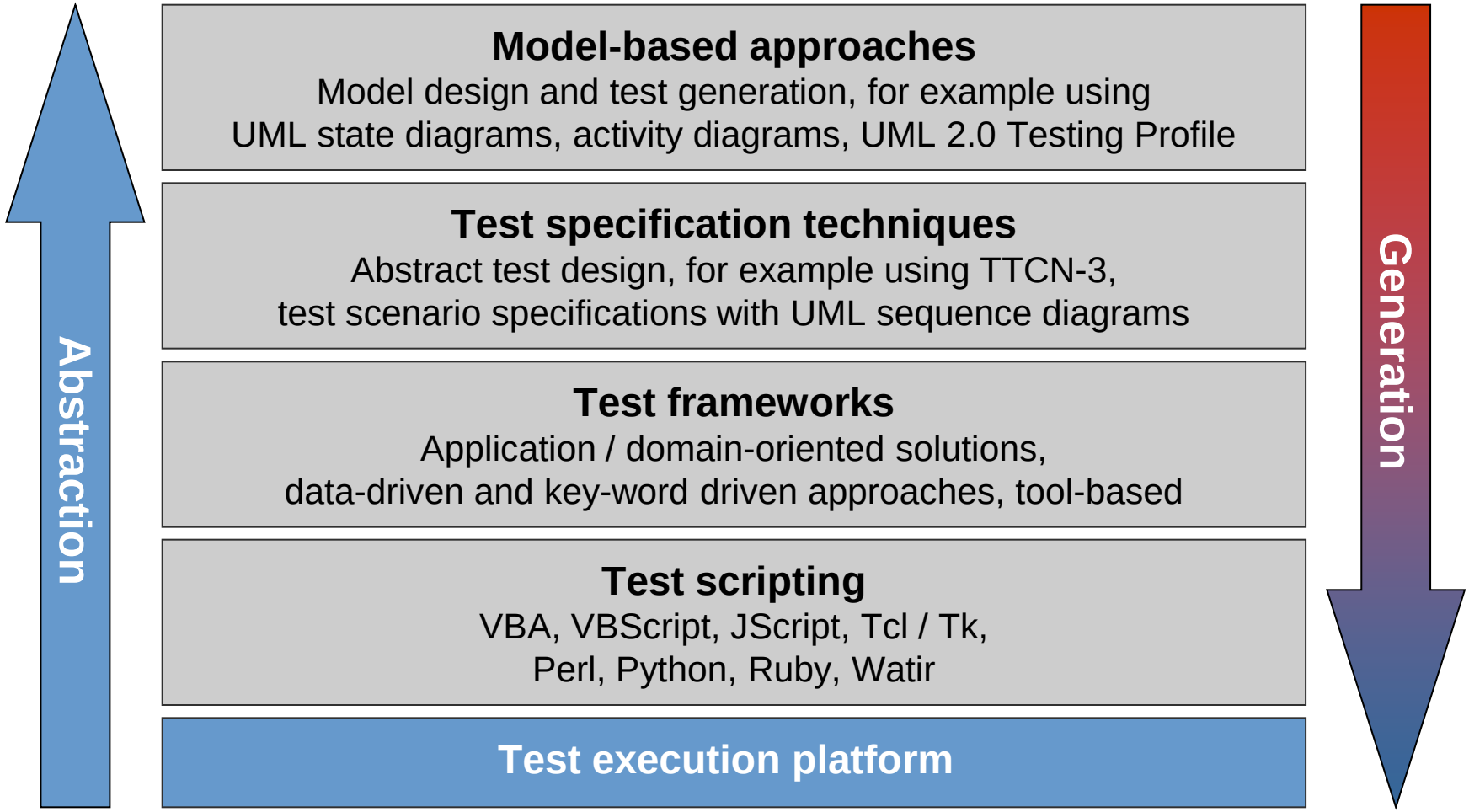


Agile testing quadrants

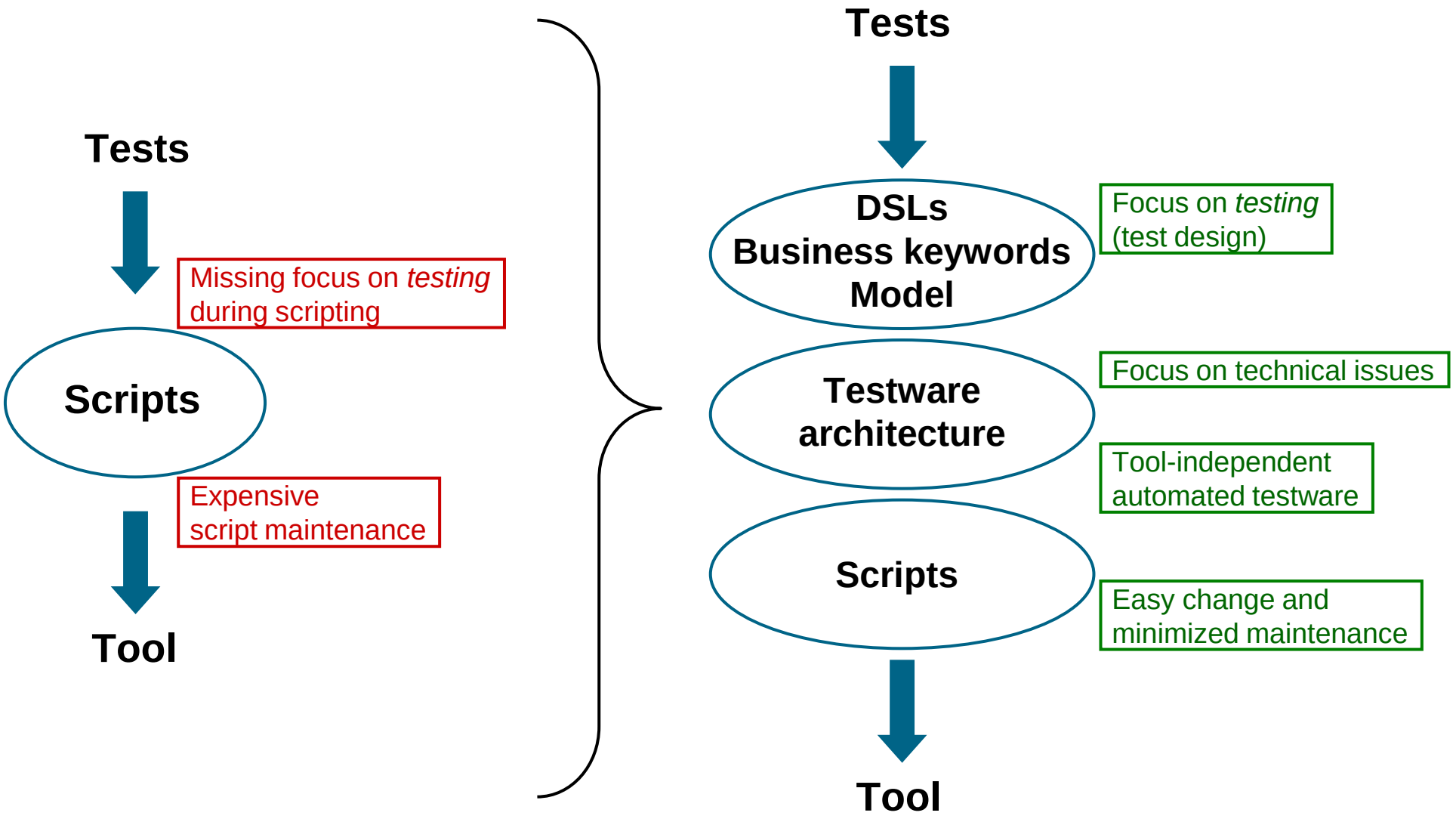


Reference: Brian Marick, Lisa Crispin, Janet Gregory

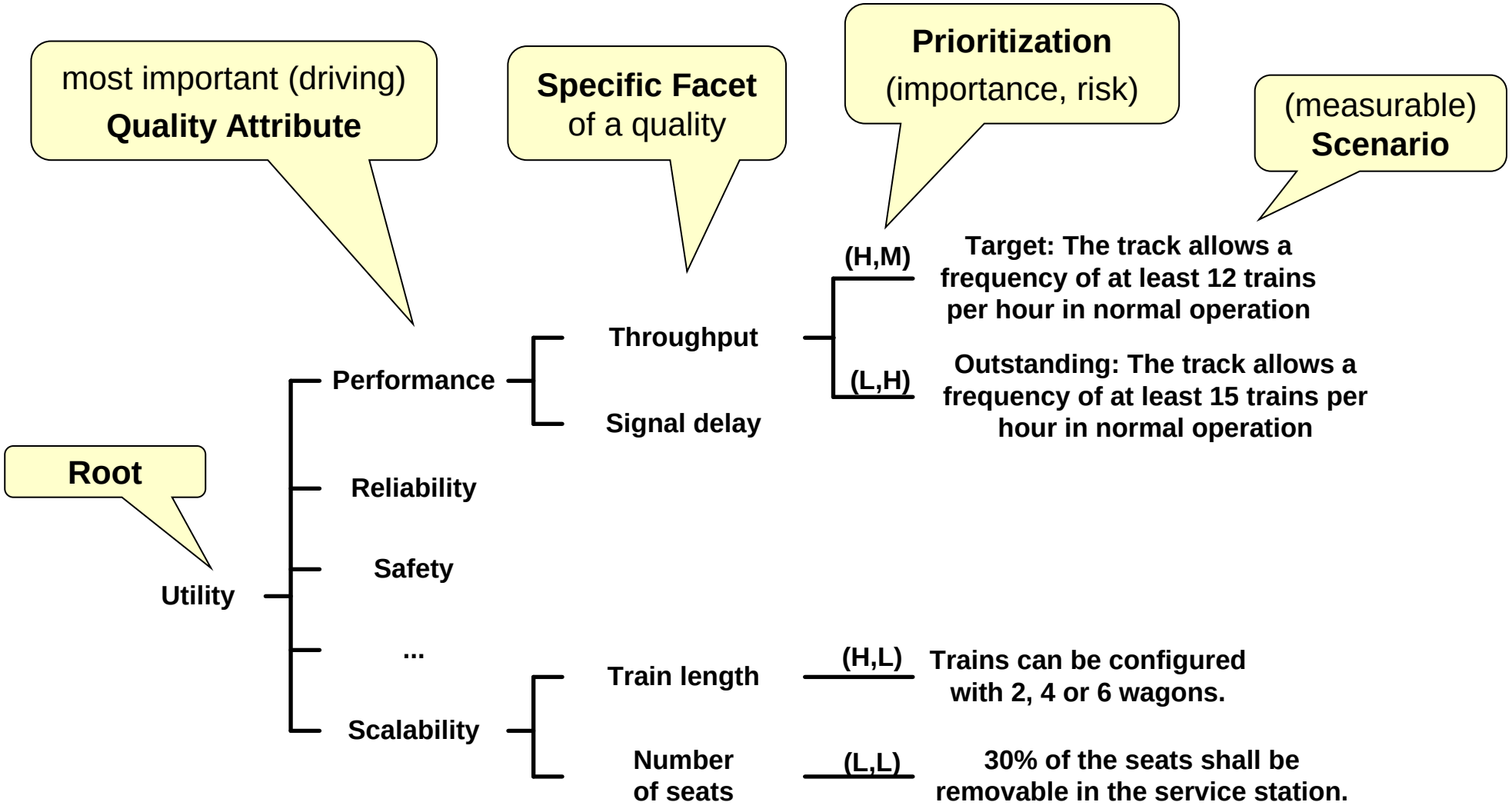
Test automation is software development



Levels of abstraction

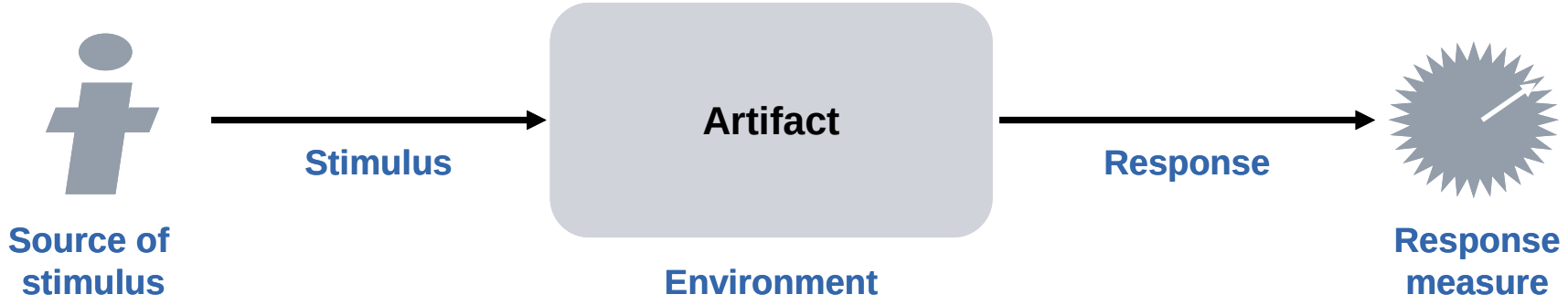


Attributes of the "Utility Tree"



Scenario description Template

SIEMENS



Source of stimulus	Who/what initiates the scenario.
Stimulus	Which periodic, stochastic or sporadic event initiates the scenario.
Artifact	What is the relevant unit; e.g. a (part of a) system or a feature.
Environment	What is the environmental condition for this scenario; e.g. normal, startup / shut down, maintenance, emergency, overload, etc.
Response	How does the artifact react to the event in the given environment. This may cause an environment change (e.g. from normal to shutdown mode).
Response measure	How can the response be measured, using indicators like: ▪ the time it takes to process the event (latency or deadline); or the variation in time (jitter) ▪ the amount of data, material or energy that can be processed in a particular time interval (throughput) ▪ or a characterization of the events that cannot be processed (e.g. miss rate, data / energy / material loss)

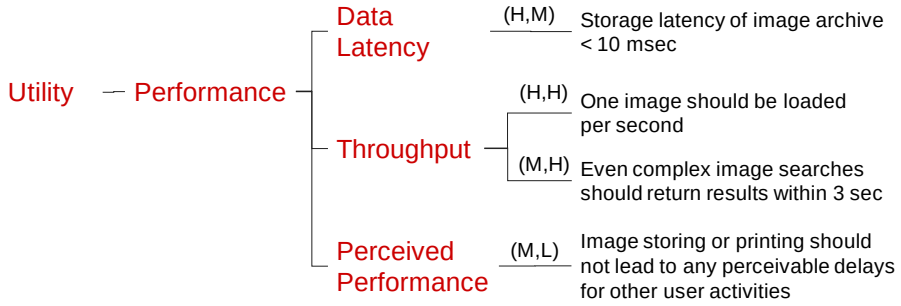
Exercise / Discussion
“Performance of Image Processing – Let’s Test”

SIEMENS

We built our quality attribute scenario…

Whenever a user searches for images in the User Interface (using keywords or advanced search criteria), finding and displaying the results should never take longer than 3 seconds.

… derived our utility tree …



Now let’s test this!

- What **test types** are needed?
- Which **test levels** should be used?
- What potential **obstacles** for writing test cases do you see?
- Which additional **input** is necessary? Which **stakeholders** can provide it?
- How can you – as Test Architects – **support** quality attribute testing activities?

… and selected our design strategies and tactics

Performance of Image Processing

Resource demand

- Use advanced image processing algorithms (if possible GPU based)
- Introduce caching strategies
- Either allow upper bound of clients or scale-out mechanisms depending on resources

Resource management

- Introduce a pool of worker threads for background loading, storing, printing
- Don't copy mass data but use meta files and refs for image processing/copying
- Use thumbnail images for browsing
- Only use full resolution when images are selected
- Apply eager loading

Resource arbitration

- Schedule resources preferably for processing visible images

There are various patterns for implementing some of the tactics such as
- Caching
- Lazy Evaluation
- Coordinator
- Eager Loading
- Evictor & Activator
- Half Sync / Half Async
- Command Processor

Testing quality attributes Approach

SIEMENS

Operational Level

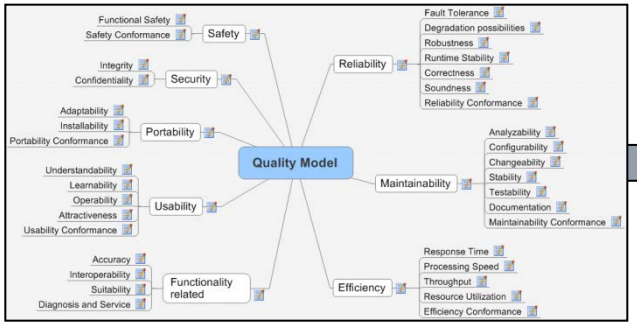
Operational Quality Requirements → Customer Acceptance Level / Criteria ← based on ← Acceptance, Field Test

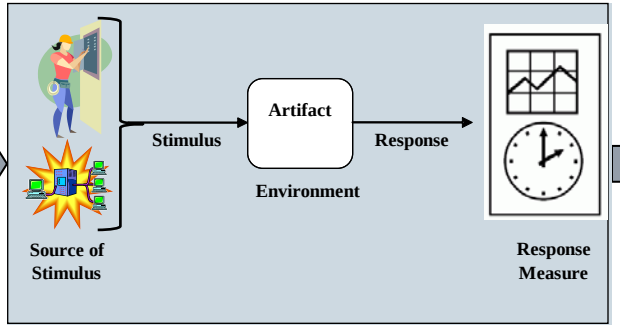
System Level

Utility Tree → Quality Facets Scenarios Risks ← based on ← Scenario Test (Black Box)

Architecture Level

Design Strategies & Tactics → Arch. Patterns Building Blocks Interactions Deployment ← based on ← Integration Test (Grey Box)





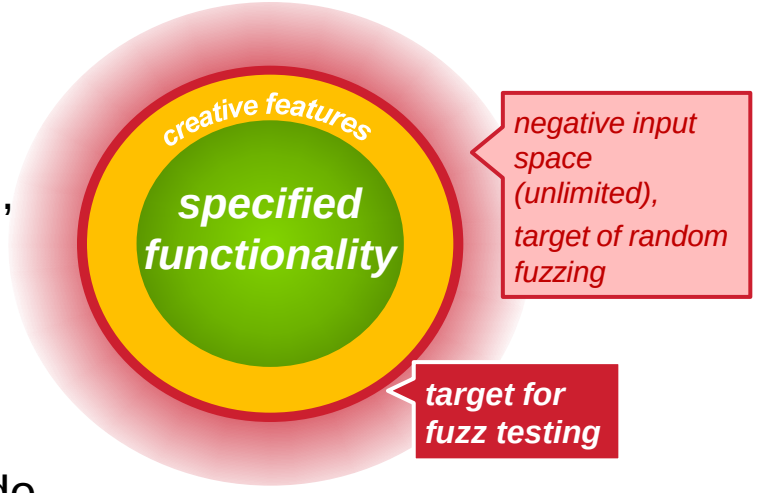
Security Testing
Fuzz Testing Approach

SIEMENS

- Fuzzing originally describe the generation of randomly generated test vectors (Miller et. al. in the early 1990s)
- **Random fuzzing:** has close to zero awareness of the tested interface.
- **Mutation-based fuzzing:** mutate existing data samples to create test data, breaks the syntax of the tested interface into blocks of data, which it semi-randomly mutates.

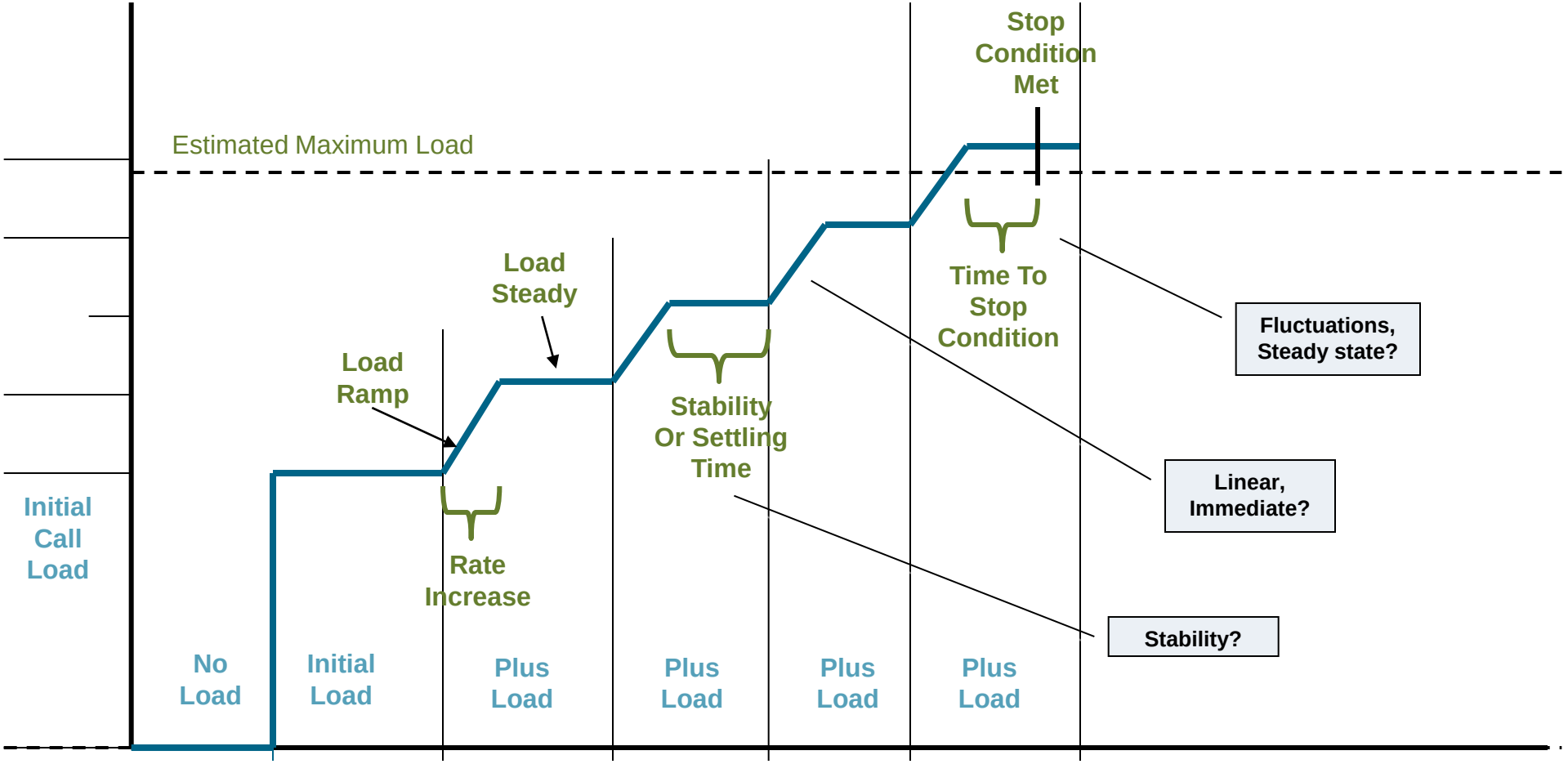
Model-based fuzzing

- uses models of the input domain (protocol models, e.g. context free grammars), for generating systematic non-random test cases
- in security testing purposes, the models are augmented with intelligent and optimized anomalies that will trigger the vulnerabilities in code
- finds defects which human testers would fail to find



See also: [TAK08]

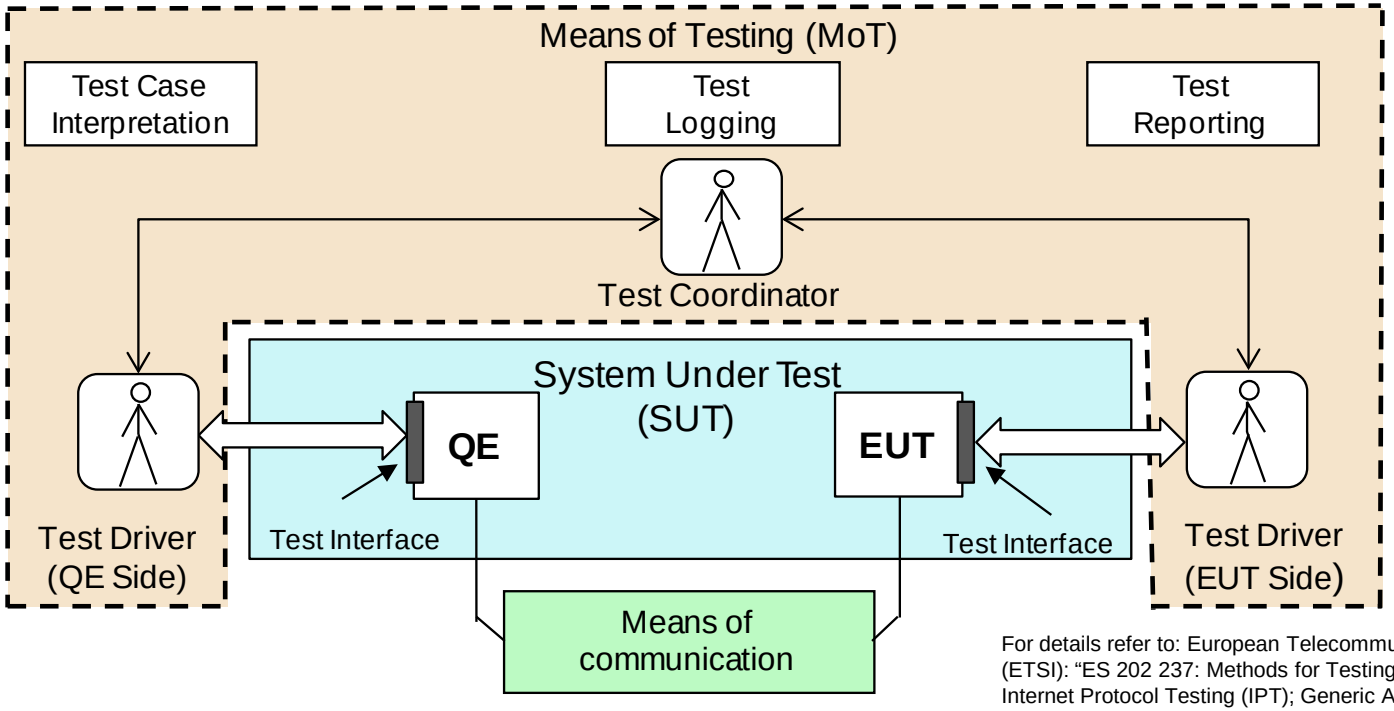
Performance Testing
Benchmarking Procedure



[DIN09]

Interoperability testing with GAIT

- Generic Approach to Interoperability Testing (GAIT) v1.2.1 (2010)
- Defined for software interoperability testing, without focusing on a specific software domain.
- QE – Qualified Equipment



For details refer to: European Telecommunications Standards Institute (ETSI): "ES 202 237: Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT); Generic Approach to Interoperability Testing", version 1.2.1, 2010.

Interoperability testing
Classification of IOP checking levels

SIEMENS

Interaction Scenario

- **what**: the required sequence of messages is validated
- **how**: timeout events indicate that some messages are not sent according to the specified sequence

Message Type

- **what**: check that IOP *message structure profile* constraints are fulfilled
- **how**: type checking upon receipt of a message from SUT according to IHE IOP integration profiles

Fields Conditionality

- **what**: the conditionally constraints across the fields within the same message is validated (e.g., if field1 is present, then field2 must be also present)
- **how**: special checking functions are used

Message Content

- **what**: the content of messages is inspected against expected values, code sets (tables), values imposed by standards, etc.
- **how**: message content checking functions

Semantic Correlations

- **what**: correlation of pieces of information *across different messages* within the flow has to be verified (e.g., for updating a patient, a Patient ID used in a previous step has to be used)
- **how**: using message tuning functions with semantic parameters or using semantic checking functions

SIEMENS

How can you lead without formal power?

An attempt at definition:

Leadership

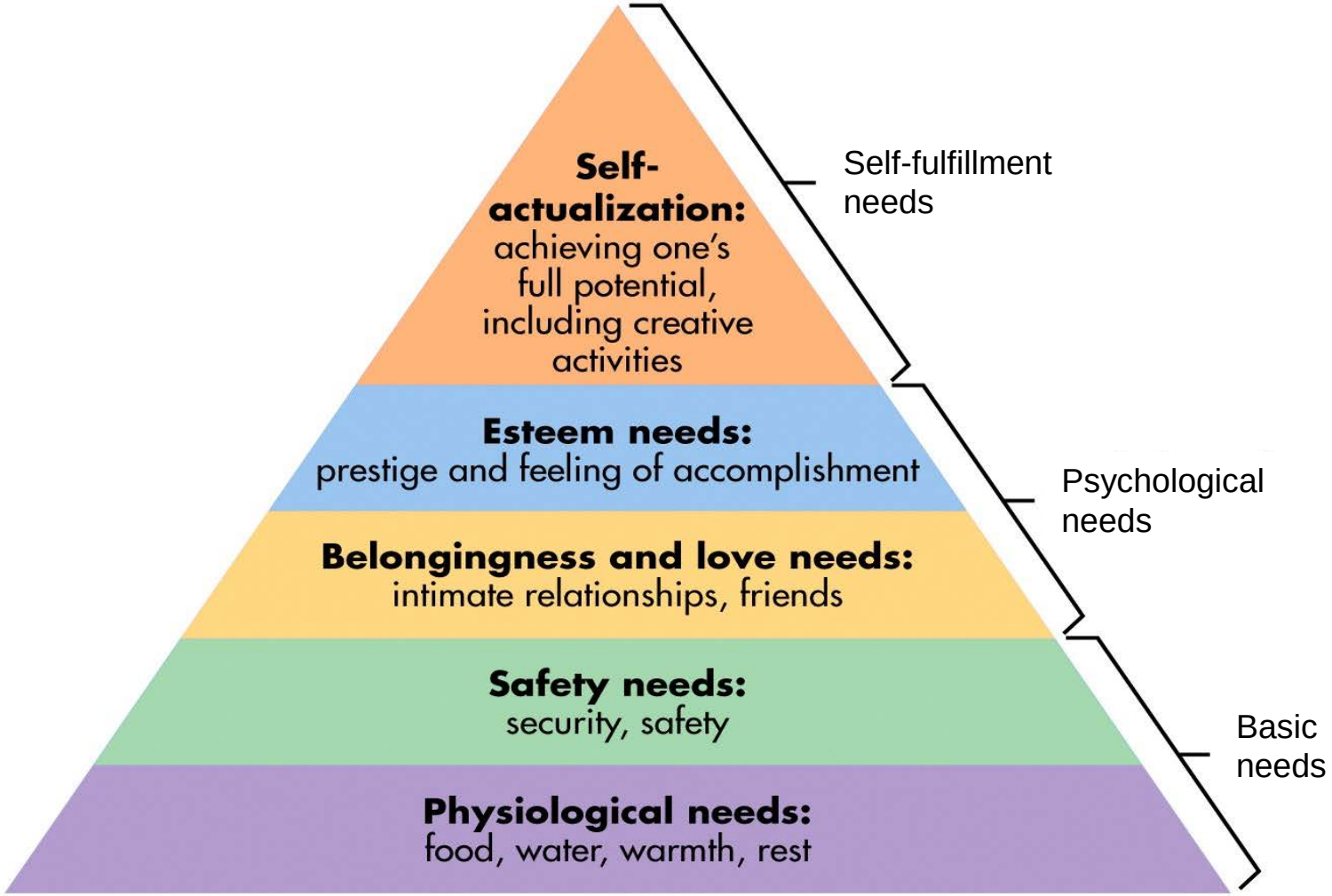
means **motivating** others

to **achieve** a common **goal**.

→ Set direction, align resources, inspire action, be responsible for results

SIEMENS

If you want to motivate – you should know the motives

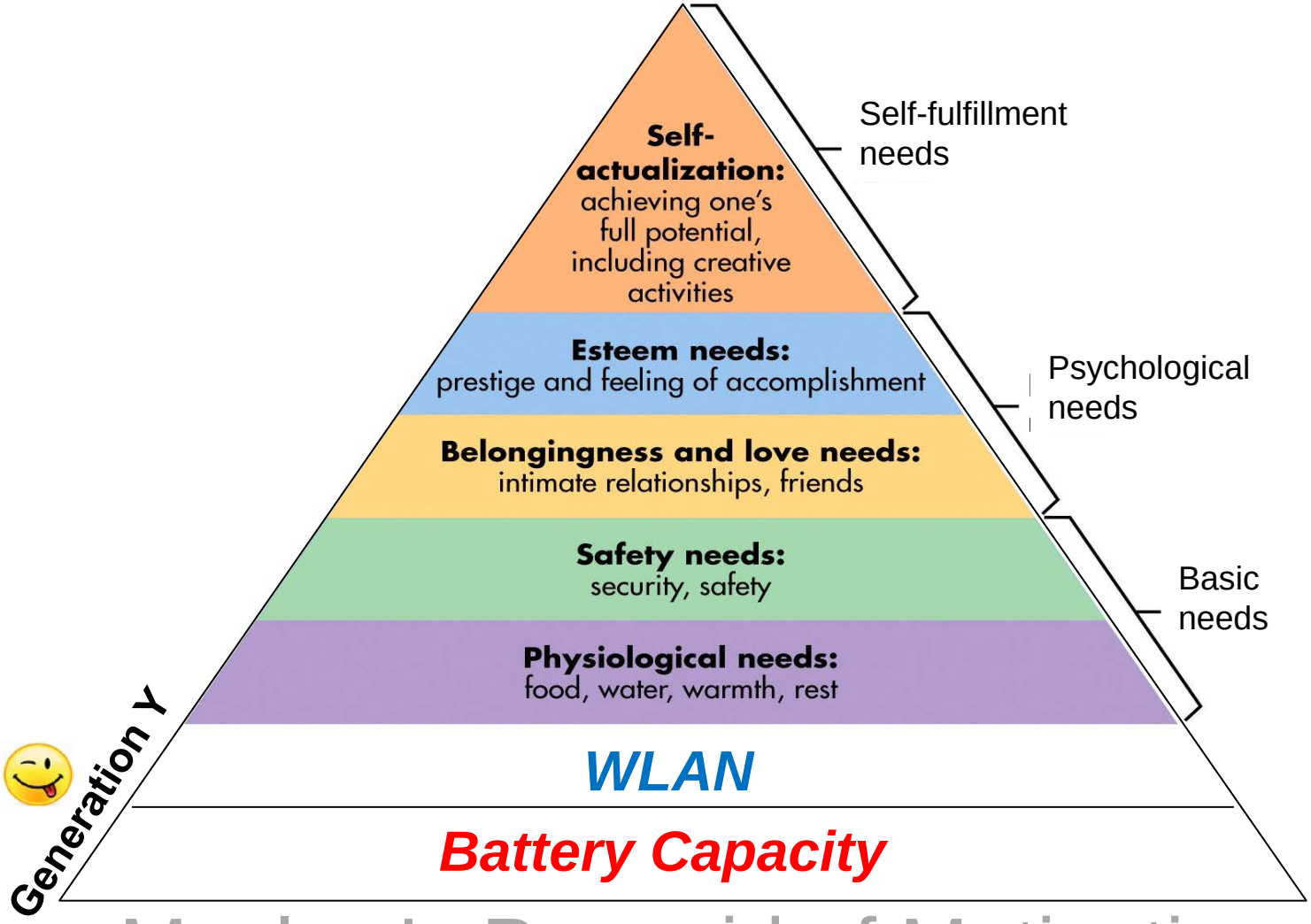


Abraham Maslow
(1908 –1970)

Maslow's Pyramid of Motivation

SIEMENS

If you want to motivate – you should know the motives



Abraham Maslow
(1908 –1970)

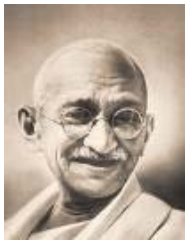
Maslow's Pyramid of Motivation

Natural and vested authority
two different sources of power

SIEMENS



Authority




Borrowed power
vested by someone else

A role
with specified powers,
duties and responsibilities

e.g. minister of transport,
project leader,
department chief.

→ What is given
can be taken away.

Helps to fill a role

Natural power
earned by yourself

A competence
developed through practice,
founded on personal attributes

e.g. ability to communicate,
solve conflicts,
convince others.

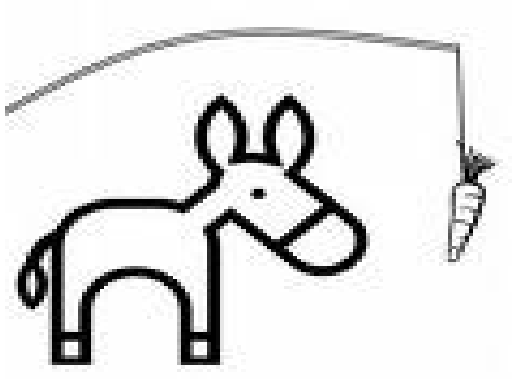
→ You take your skills with you
wherever you go.

SIEMENS

How *NOT* to motivate – Carrot and KITA

You need a maximum of controlling effort if you use force instead of motivation.
Manipulation instead of motivation will lead to frustration and a loss of trust.

How to motivate: The carrot only helps...



...if donkey is hungry...

...if donkey likes it,...

...if donkey can reach it and...

...if the task fits donkey's ability.

SIEMENS

To use force and manipulation fosters resistance

You need a maximum of controlling effort if you use force instead of motivation.

If you use manipulation it will eventually lead to frustration and a loss of trust.

What do you do if someone is not delivering as expected?

SIEMENS

Knowing

- Does he know what you expect him to do?
- Does he know the goal and why it is important?

Able

- Does he know how to do it?
- Does he have the necessary skills and experience?

Willing

- Is he motivated to do it?
- What would motivate him?

Allowed

- Do the circumstances allow him to fulfill the task?
- What barriers have to be removed?

Source: L. v. Rosenstiel, 1999

The Situational Leadership Model can provide orientation

SIEMENS

Supportive
leadership
style

Cooperative
leadership
style

Delegative
leadership
style

Directive
leadership
style

“Shadow Styles“
Better to avoid the dark side

SIEMENS

Charity
~~leadership~~ style

Chitchat
~~leadership~~ style

Laisser-faire
~~leadership~~ style

Authoritarian
~~leadership~~ style

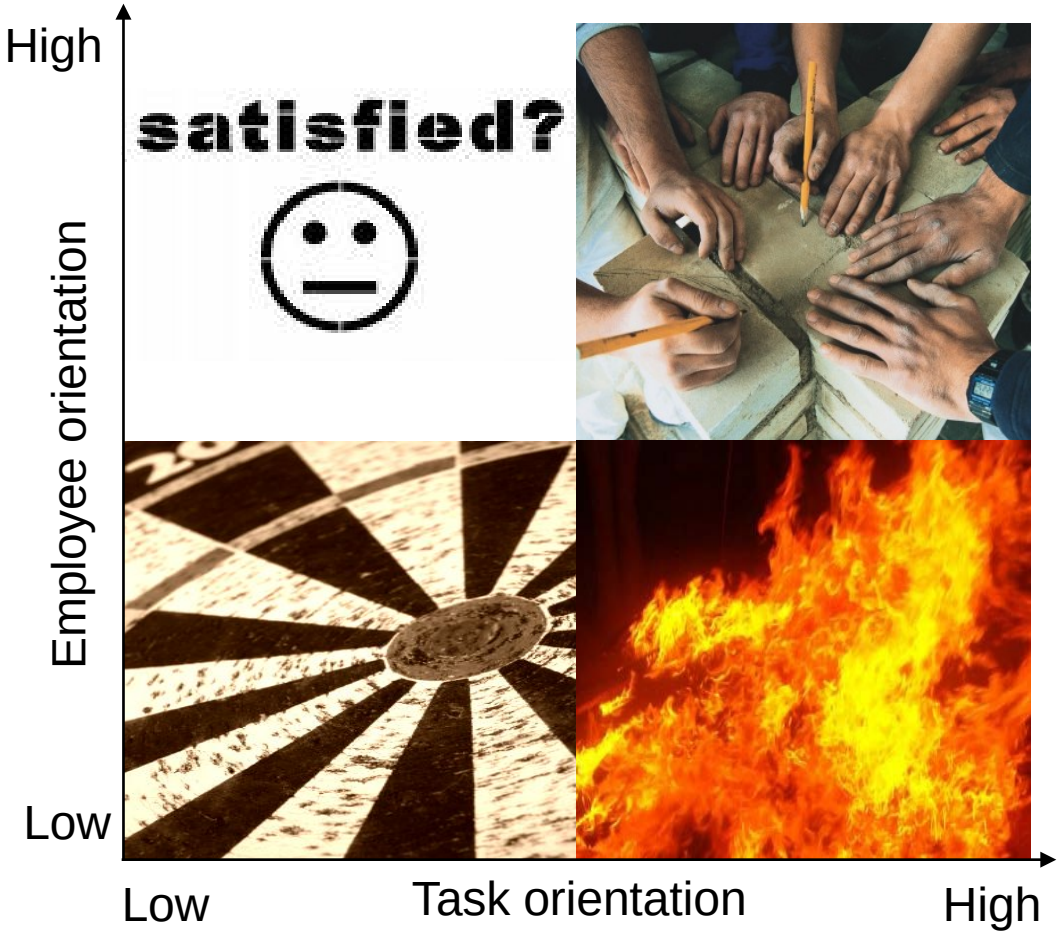
There are ways to become more effective

Active Listening
KAWA
Receive Feedback
Self Motivation
Natural Authority

SMART

Trust before
Trustworthiness !

Mastery
Purpose
Motivation



Provide Feedback
Facilitation
Conflict Management
Interdisciplinary Teams
Getting Past NO
Dilemma Reconciliation

Feedback
Escalation

Clarity in strategy
and instructions

Vested Authority