Testing Challenges

for Cyber Physical Systems

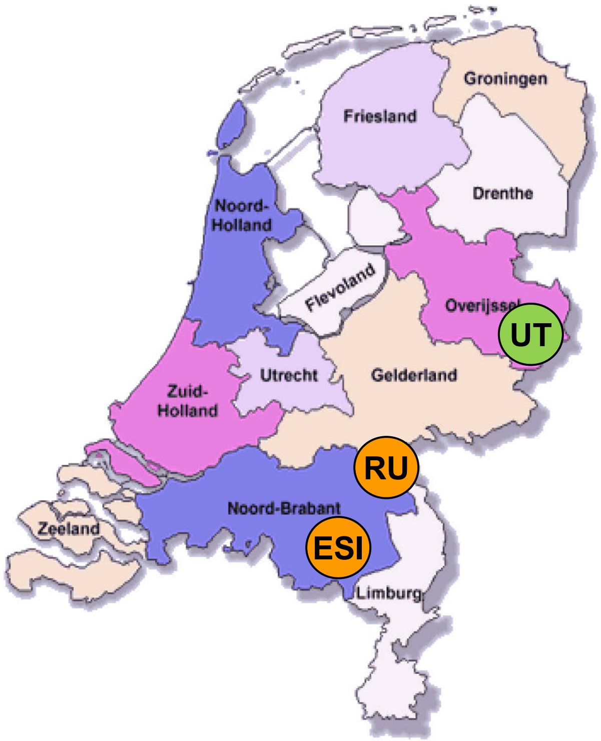
Jan Tretmans

TNO - ESI – Embedded Systems Innovation at TNO
Radboud University Nijmegen
Högskolan i Halmstad
jan.tretmans@tno.nl

Jan Tretmans

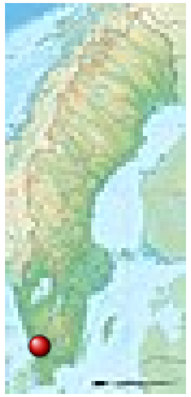
ESI

TNO - ESI
Embedded Systems Innovation
Eindhoven
The Netherlands



Radboud University
Nijmegen
The Netherlands

HÖGSKOLAN
I HALMSTAD



TNO innovation for life

Jan Tretmans

- Formal Methods
  - Maeslant Kering Rotterdam

- Software Testing

- Model-Based Testing MBT
  - **ioco**-theory for MBT
    with labelled transition systems
  - MBT tool **TorXakis**

TorXakis

TNO – ESI : Applied Research at a Glance

ESI

Synopsis

- ❑ Foundation ESI started in 2002
- ❑ ESI acquired by TNO per January 2013
- ❑ ~60 staff members, many with extensive industrial experience
- ❑ 7 Part-time Professors
- ❑ Working at industry locations

Focus

Managing complexity
of high-tech systems

through

- system architecting,
- system reasoning and
- model-driven engineering

delivering

- methodologies validated in cutting-edge industrial practice

Partners

ASML
ThermoFisher SCIENTIFIC
PHILIPS
itec
VANDERLANDE
Canon
THALES
TNO
UNIVERSITY OF AMSTERDAM
TUDelft Delft University of Technology
Radboud University Nijmegen
UNIVERSITY OF TWENTE.
TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY
Capgemini engineering

Cyber-Physical Systems

Semiconductor manufacturing equipmen

Traffic management

Combat management systems

Software is brain of system

- software controls, connects, monitors almost any aspect of ES system behaviour
- majority of innovation is in software

Software determines quality and reliability of Cyber-Physical System

- often > 50 % of system defects are software bugs

Industrial printers

Automotive

Agricultural robots

Robotized warehousing

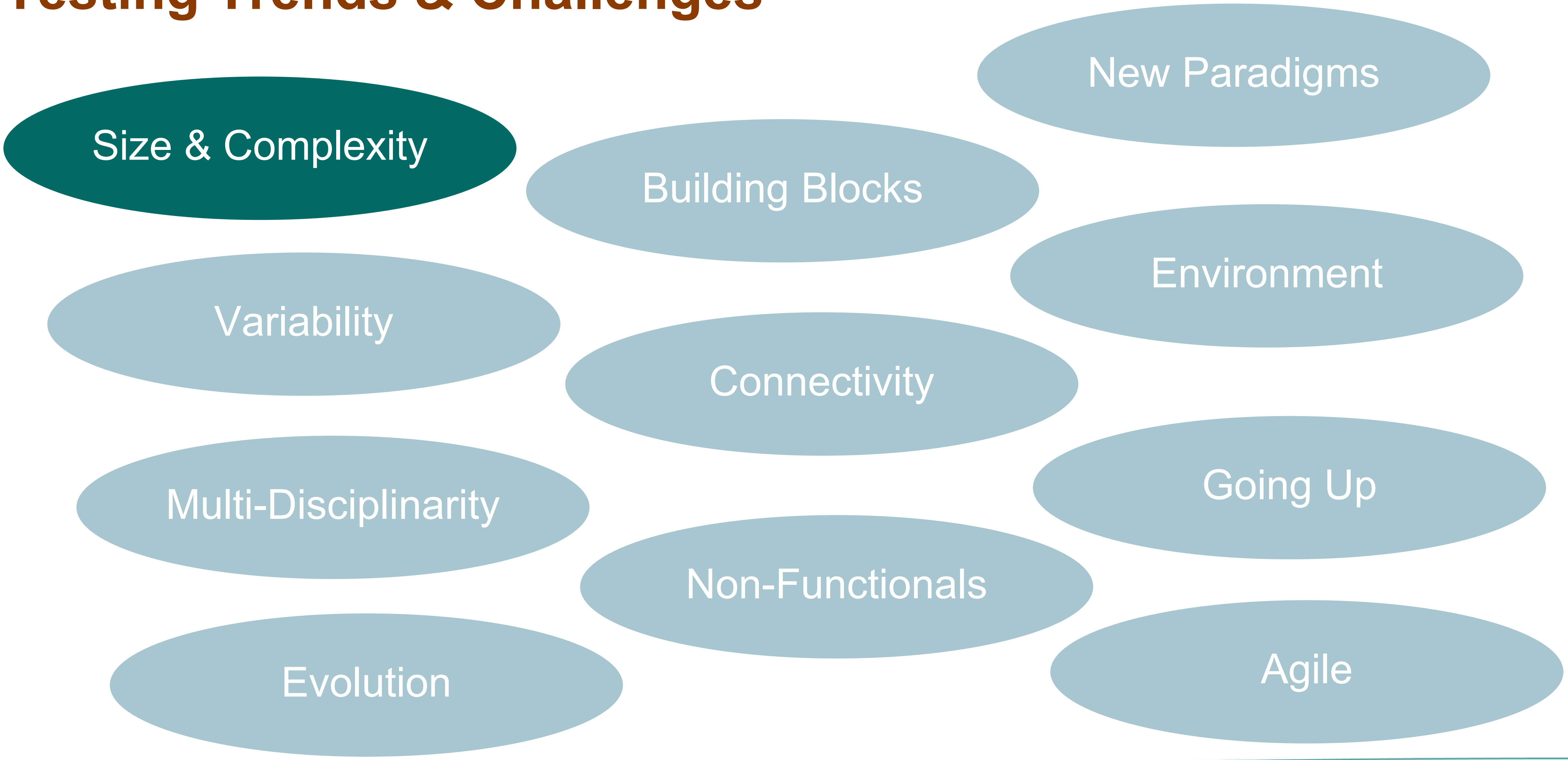
Dike

Testing Challenges for Cyber-Physical Systems

Testing Trends & Challenges

- Complexity
- Variability
- Multi-Disciplinarity
- Evolution
- Building Blocks
- Connectivity
- Non-Functionals
- New Paradigms
- Environment
- Going Up
- Agile

Testing Trends & Challenges



- New Paradigms
- Size & Complexity
- Building Blocks
- Environment
- Variability
- Connectivity
- Going Up
- Multi-Disciplinarity
- Non-Functionals
- Agile
- Evolution

Size & Complexity

Completely testing ‘ + ’ for 32-bit Int

- $2^{32} * 2^{32} = 10^{19}$ test cases
- 1 nsec / test = 585 years of testing

Car

- 100,000,000 LoC
- 40,000 parts
- 4,000 manufactured components

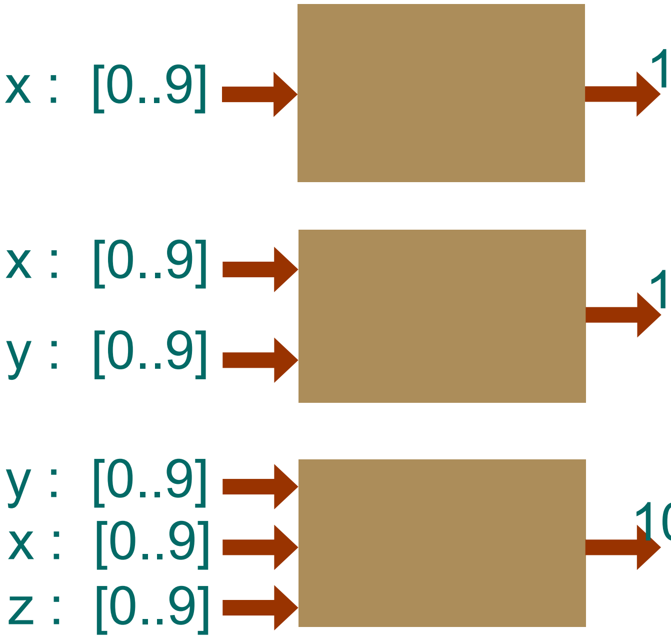
Machine with 300 parameters

- $2^{300} = 10^{90}$ different configurations
- #atoms on earth $= 10^{50}$, #atoms in known universe $= 10^{80}$

Size & Complexity

Testing effort grows exponentially with system size

Testing cannot keep pace with development



x : [0..9]

10 ways that it can go wrong

10 combinations of inputs to check

x : [0..9]

y : [0..9]

100 ways that it can go wrong

100 combinations of inputs to check

y : [0..9]

x : [0..9]

z : [0..9]

1000 ways that it can go wrong

1000 combinations of inputs to check

Size & Complexity

MURPHY'S LAW

IF IT CAN GO WRONG

IT WILL!!

Testing effort grows exponentially with system size

Testing cannot keep pace with development

→ combinatorial explosion of required testing effort

Testing Trends & Challenges

Complexity

Variability

Multi-Disciplinarity

Evolution

Building Blocks

Connectivity

Non-Functionals

New Paradigms

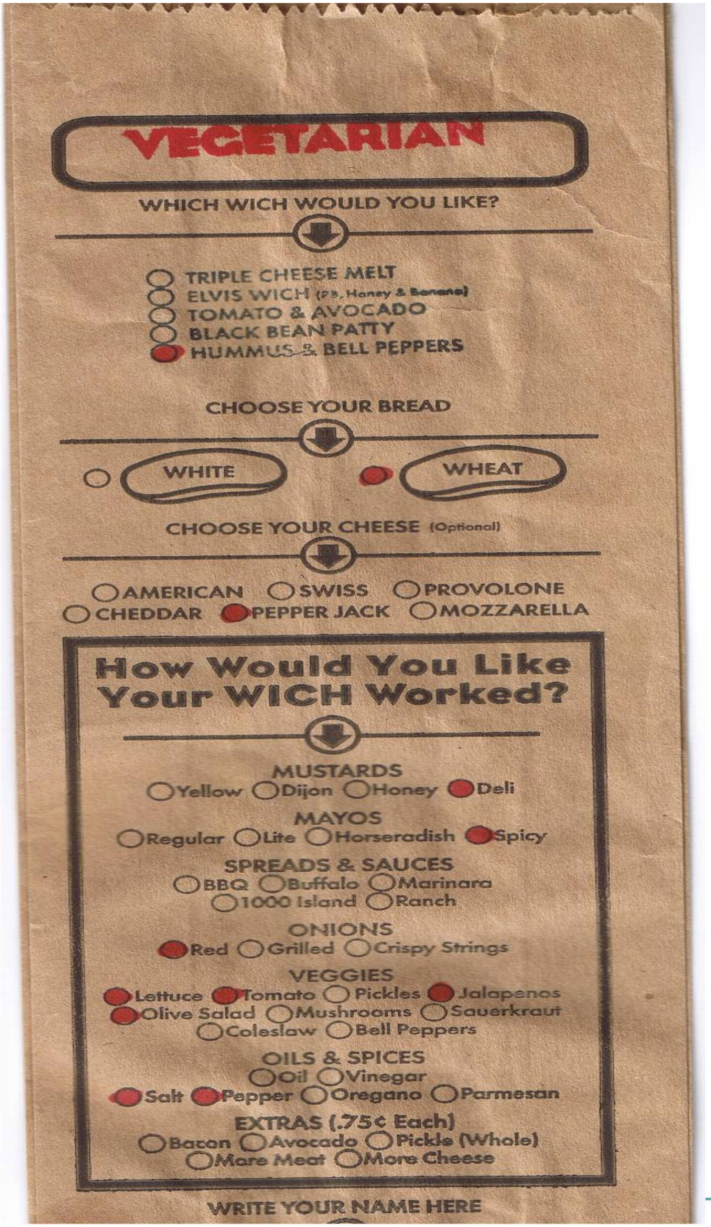
Environment

Going Up

Agile

Variability & Product Lines

or: How to Select your Sandwich



VEGETARIAN

WHICH WICH WOULD YOU LIKE?

○ TRIPLE CHEESE MELT
○ ELVIS WICH (PB, Honey & Banana)
○ TOMATO & AVOCADO
○ BLACK BEAN PATTY
● HUMMUS & BELL PEPPERS

CHOOSE YOUR BREAD

○ WHITE ● WHEAT

CHOOSE YOUR CHEESE (Optional)

○ AMERICAN ○ SWISS ○ PROVOLONE
○ CHEDDAR ● PEPPER JACK ○ MOZZARELLA

How Would You Like Your WICH Worked?

MUSTARDS
○ Yellow ○ Dijon ○ Honey ● Deli

MAYOS
○ Regular ○ Lite ○ Horseradish ● Spicy

SPREADS & SAUCES
○ BBQ ○ Buffalo ○ Marinara
○ 1000 Island ○ Ranch

ONIONS
● Red ○ Grilled ○ Crispy Strings

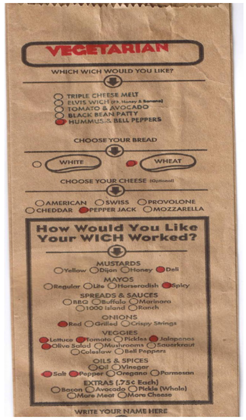
VEGGIES
● Lettuce ● Tomato ○ Pickles ● Jalapenos
● Olive Salad ○ Mushrooms ○ Sauerkraut
○ Coleslaw ○ Bell Peppers

OILS & SPICES
○ Oil ○ Vinegar
● Salt ● Pepper ○ Oregano ○ Parmesan

EXTRAS (.75¢ Each)
○ Bacon ○ Avocado ○ Pickle (Whole)
○ More Meat ○ More Cheese

WRITE YOUR NAME HERE

Variability



VEGETARIAN

WHICH WICH WOULD YOU LIKE?

○ TRIPLE CHEESE MELT
○ ELVIS WICH (PB, Honey & Banana)
○ TOMATO & AVOCADO
○ BLACK BEAN PATTY
● HUMMUS & BELL PEPPERS

CHOOSE YOUR BREAD

○ WHITE ● WHEAT

CHOOSE YOUR CHEESE (Optional)

○ AMERICAN ○ SWISS ○ PROVOLONE
○ CHEDDAR ● PEPPER JACK ○ MOZZARELLA

How Would You Like Your WICH Worked?

MUSTARDS
○ Yellow ○ Dijon ○ Honey ● Deli

MAYOS
○ Regular ○ Lite ○ Horseradish ● Spicy

SPREADS & SAUCES
○ BBQ ○ Buffalo ○ Marinara
○ 1000 Island ○ Ranch

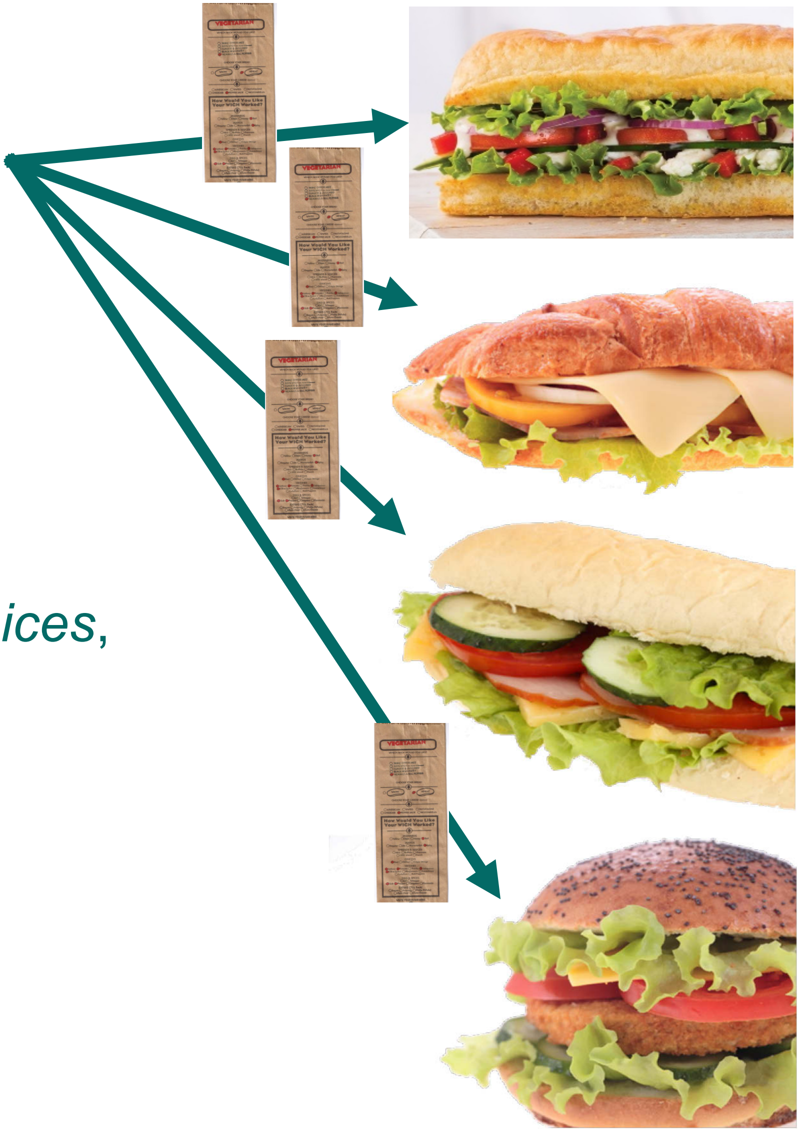
ONIONS
● Red ○ Grilled ○ Crispy Strings

VEGGIES
● Lettuce ● Tomato ○ Pickles ● Jalapenos
● Olive Salad ○ Mushrooms ○ Sauerkraut
○ Coleslaw ○ Bell Peppers

OILS & SPICES
○ Oil ○ Vinegar
● Salt ● Pepper ○ Oregano ○ Parmesan

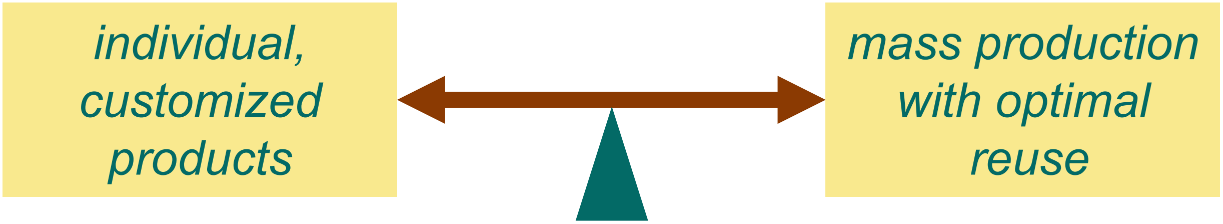
EXTRAS (.75¢ Each)
○ Bacon ○ Avocado ○ Pickle (Whole)
○ More Meat ○ More Cheese

WRITE YOUR NAME HERE



- Highly configurable sandwich*: exponential number of choices*, **a different sandwich for everybody on the planet!**
- Not all combinations make sense: *dependencies*
- **How to taste / test all of them?**
- Sandwich product line = family of sandwiches
- Also for high-tech systems
  Linux, cars, . . .

Variability Engineering



individual, customized products

mass production with optimal reuse

- Customization & reuse by developing families of 'similar' products

  → identify variation points

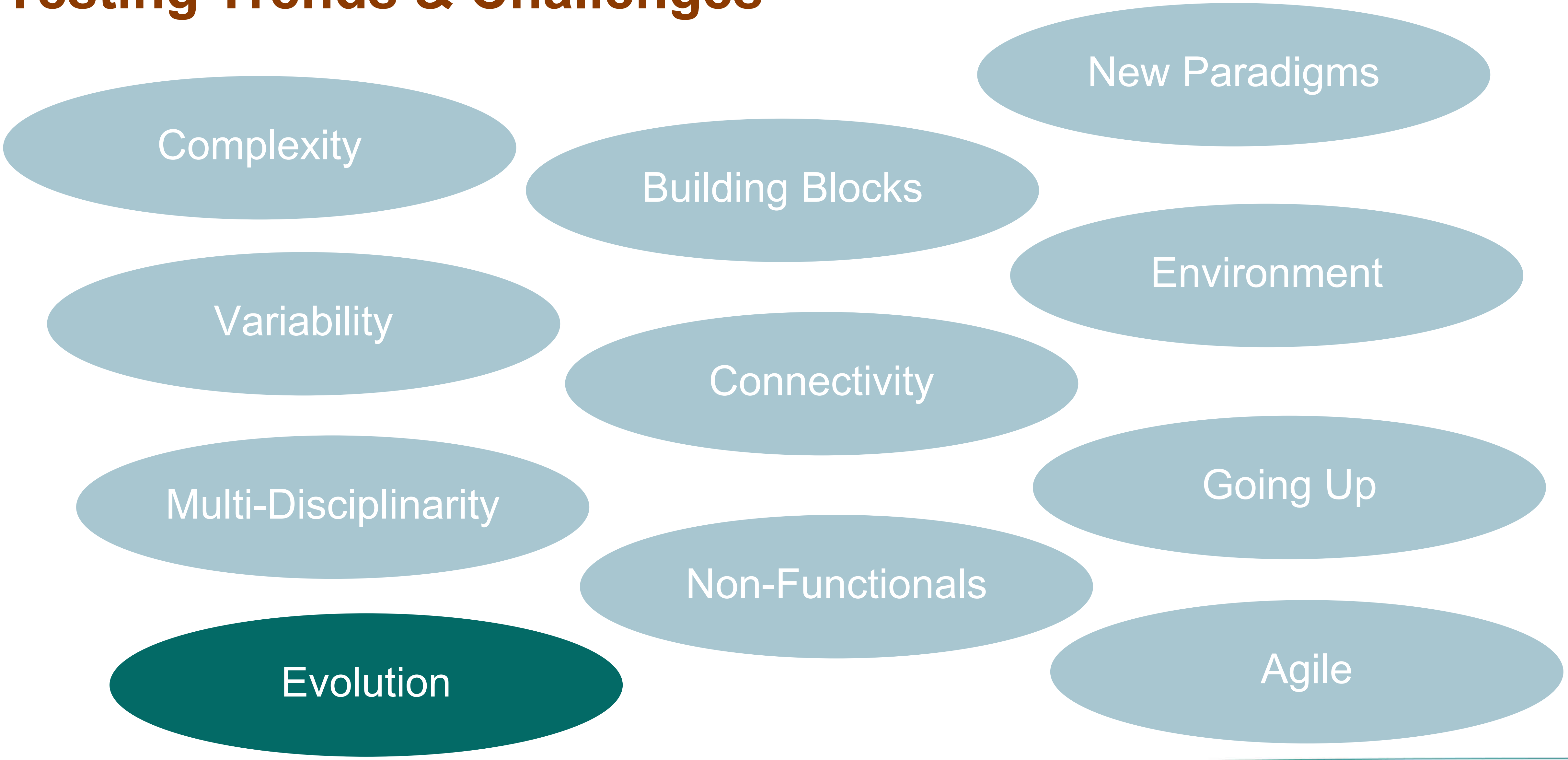
  → instantiate to different configurations = products

- *Aim*: instantiate as late as possible,
  to perform design, analysis, …, on the product family
  and not on each individual product

- *But*: testing is always on an individual product

  → **how to select configurations for testing ?**

Testing Trends & Challenges



Complexity

Variability

Multi-Disciplinarity

Evolution

Building Blocks

Connectivity

Non-Functionals

New Paradigms

Environment

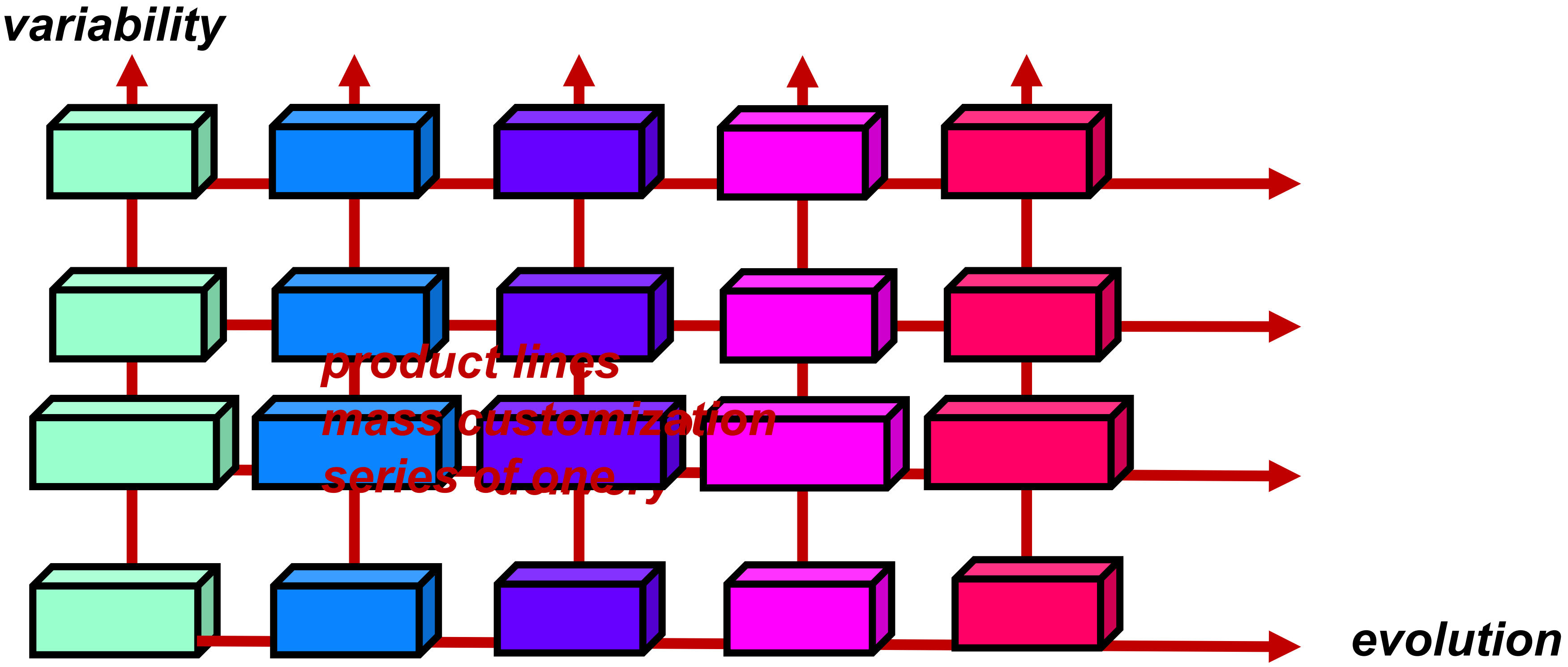
Going Up

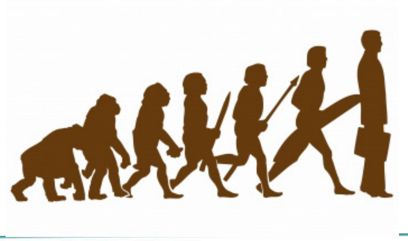
Agile

Evolution : Change over Time

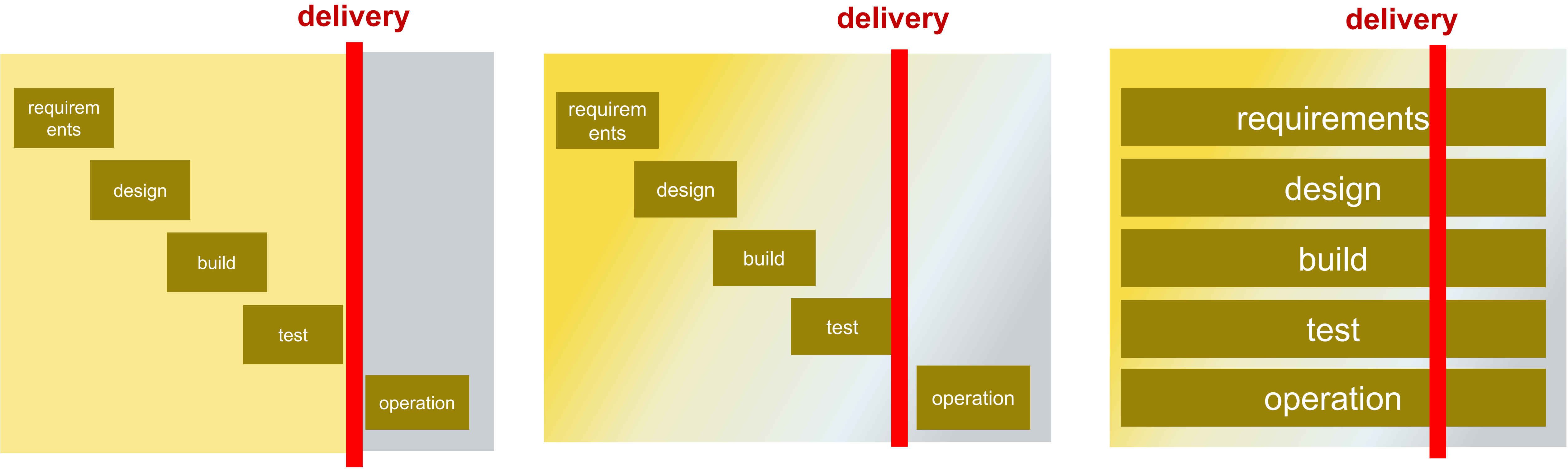
- system never comes alone: *variability*
- systems continuously change: *evolution*

yet another source of
Test Explosion





Evolution, Change : Fading Boundaries



delivery

requirem ents

design

build

test

operation

delivery

requirem ents

design

build

test

operation

delivery

requirements

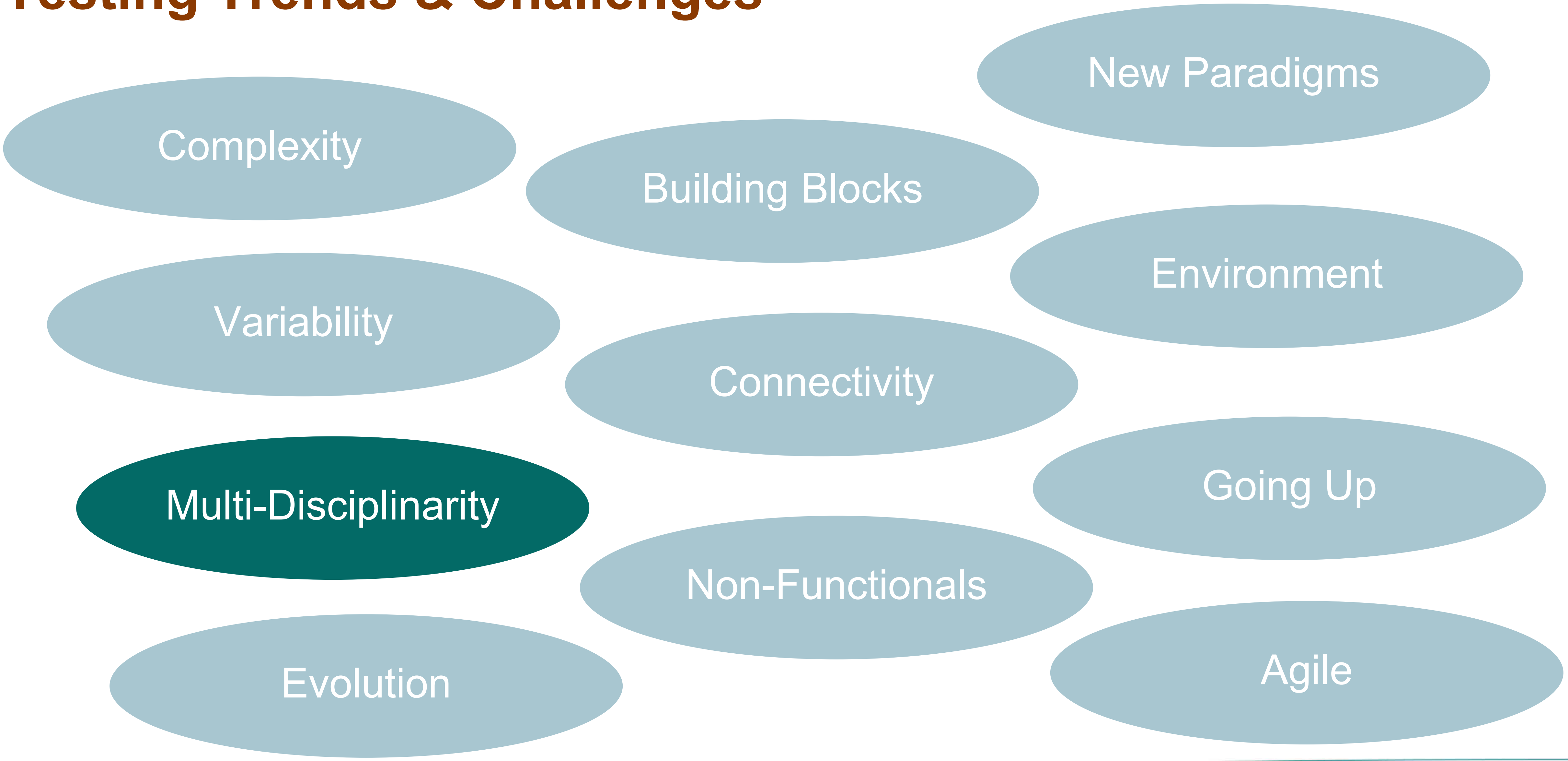
design

build

test

operation

Testing Trends & Challenges



Complexity

Variability

Multi-Disciplinarity

Evolution

Building Blocks

Connectivity

Non-Functionals

New Paradigms

Environment

Going Up

Agile

Cyber-Physical Systems

Semiconductor manufacturing equipment

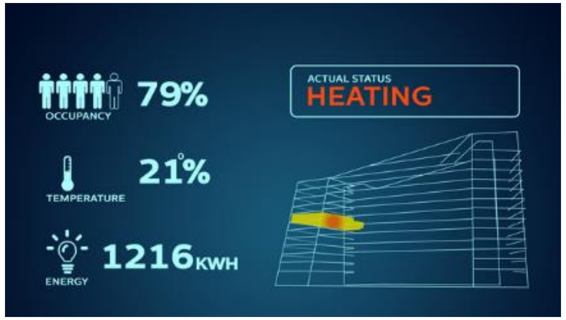
Medical systems

Food processing

Agricultural robots

Traffic management

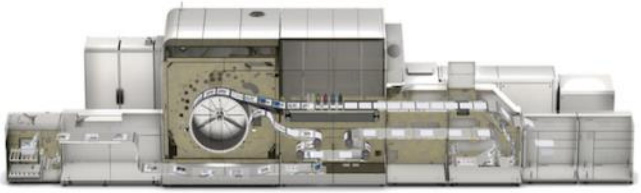
Electron microscopes



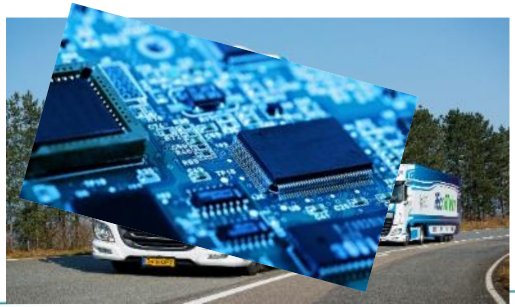
Building control

Robotized warehousing

Combat management systems



Industrial printers



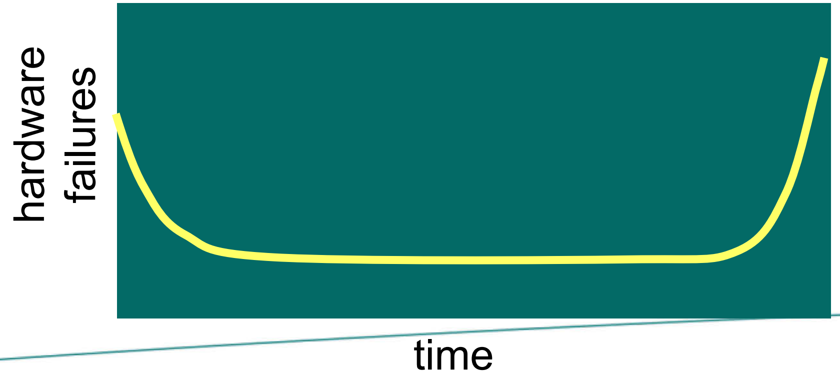
Automotive

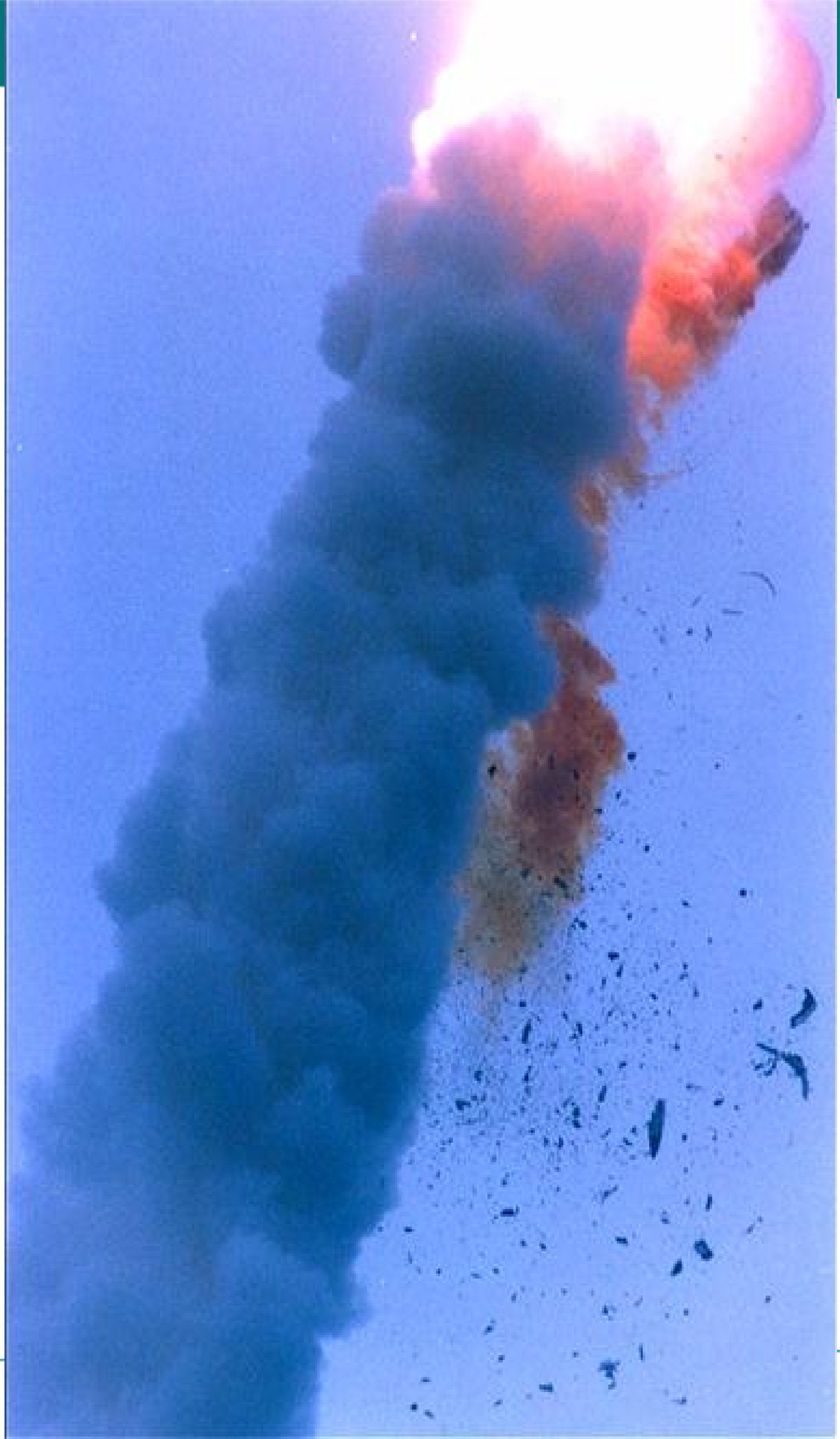
Dike

Software is Different

Software is different from hardware :

- non-continuous
- any bug is a design error
- adopting redundancy is useless
- no wear and tear
- no MTBF
- what is software reliability?





Multi-disciplinarity

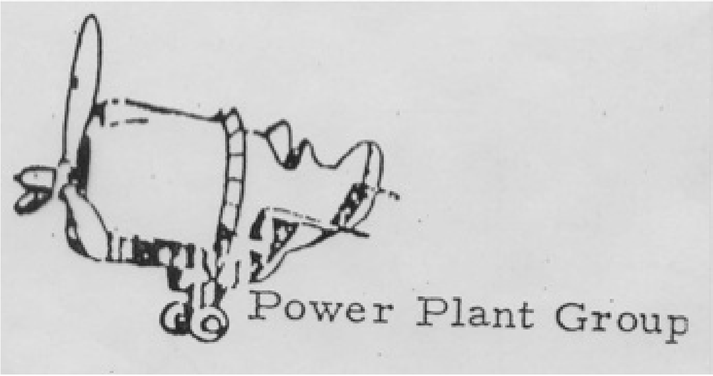
- Cyber-Physical Systems
- Combination of physics/mechanics/electronics … with computer/software
- Requires various expertises
- Testing such combinations requires

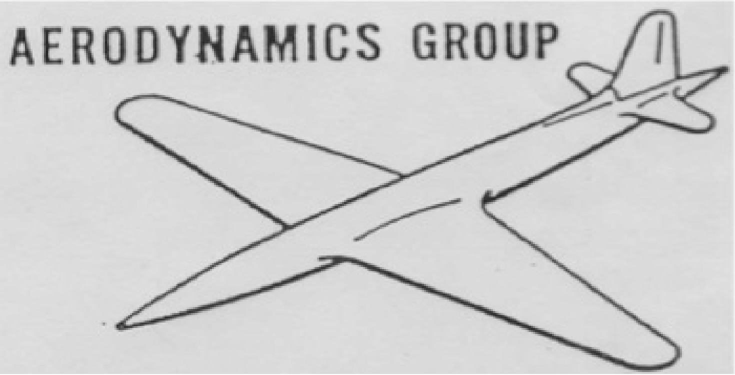
  stubs, simulations, virtualization, digital twin

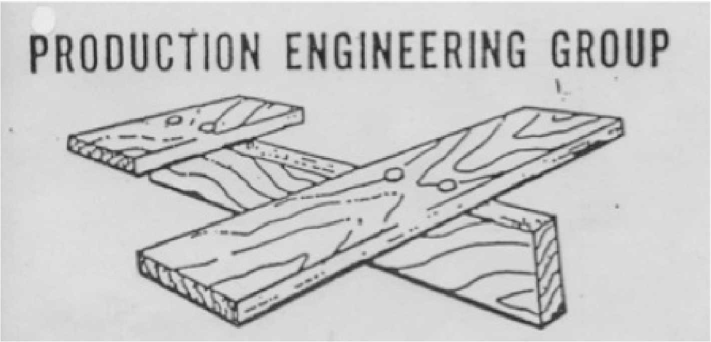
Multi-disciplinarity

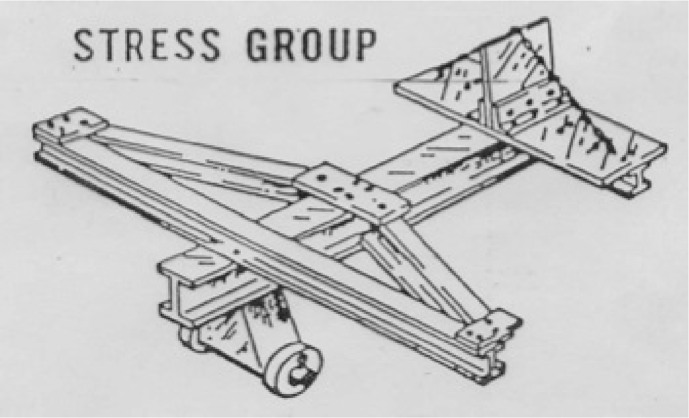
- **Virtualization**
  - models to simulate/emulate physical and environment in t
  - intelligent stub, in-the-loop testing
  - because real system is: expensive, infeasible, dangerous, too slow, too fast, cannot produce error scenarios, …
- **Modeling**
  - system ←→ physical part ←→ software ←→ environment
  - models for virtualization ←→ models for testing

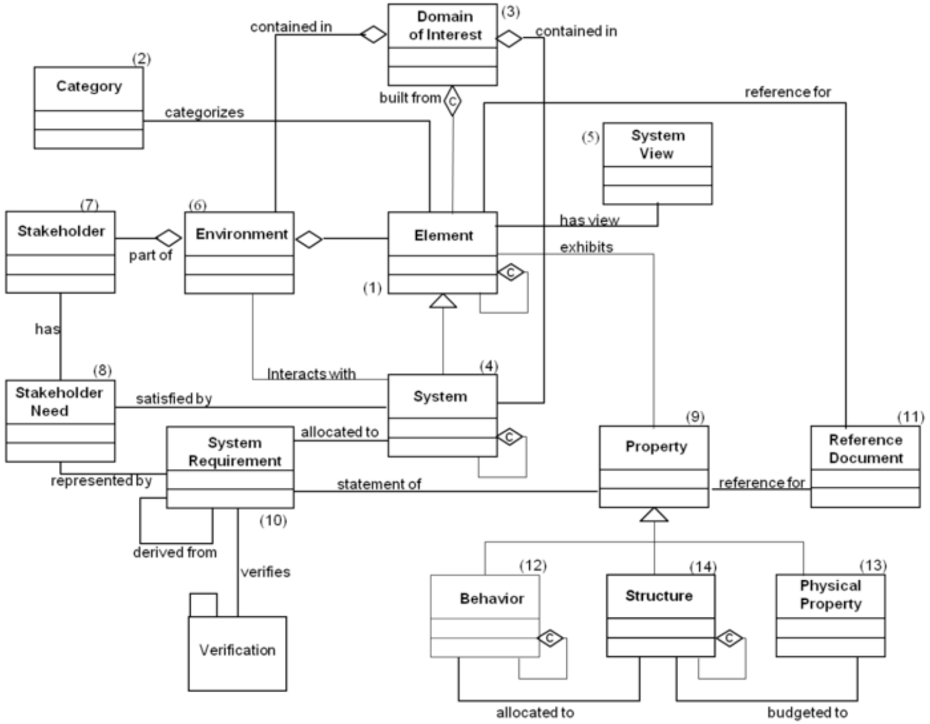
Multi-disciplinarity : Different Views on Systems





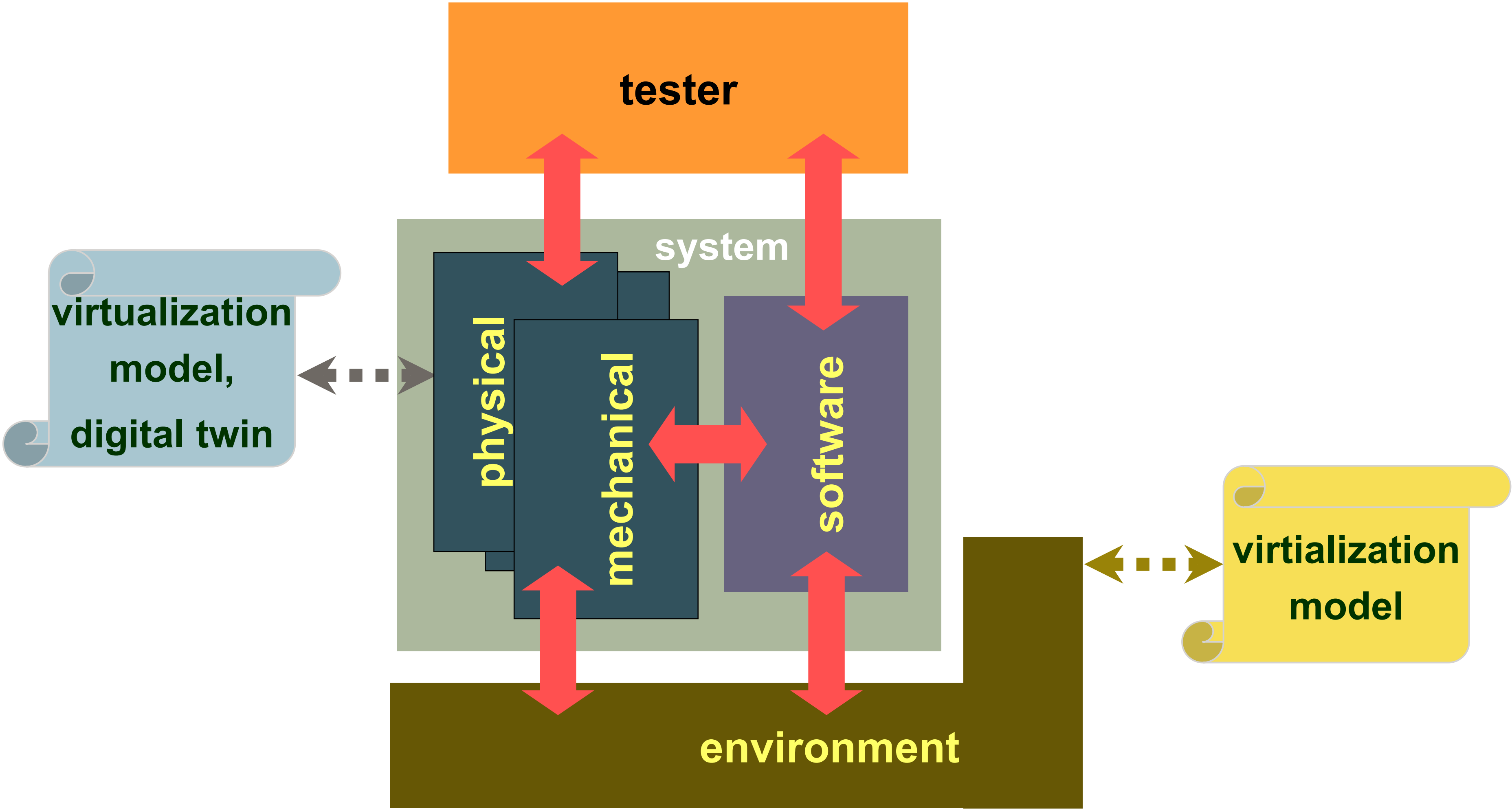




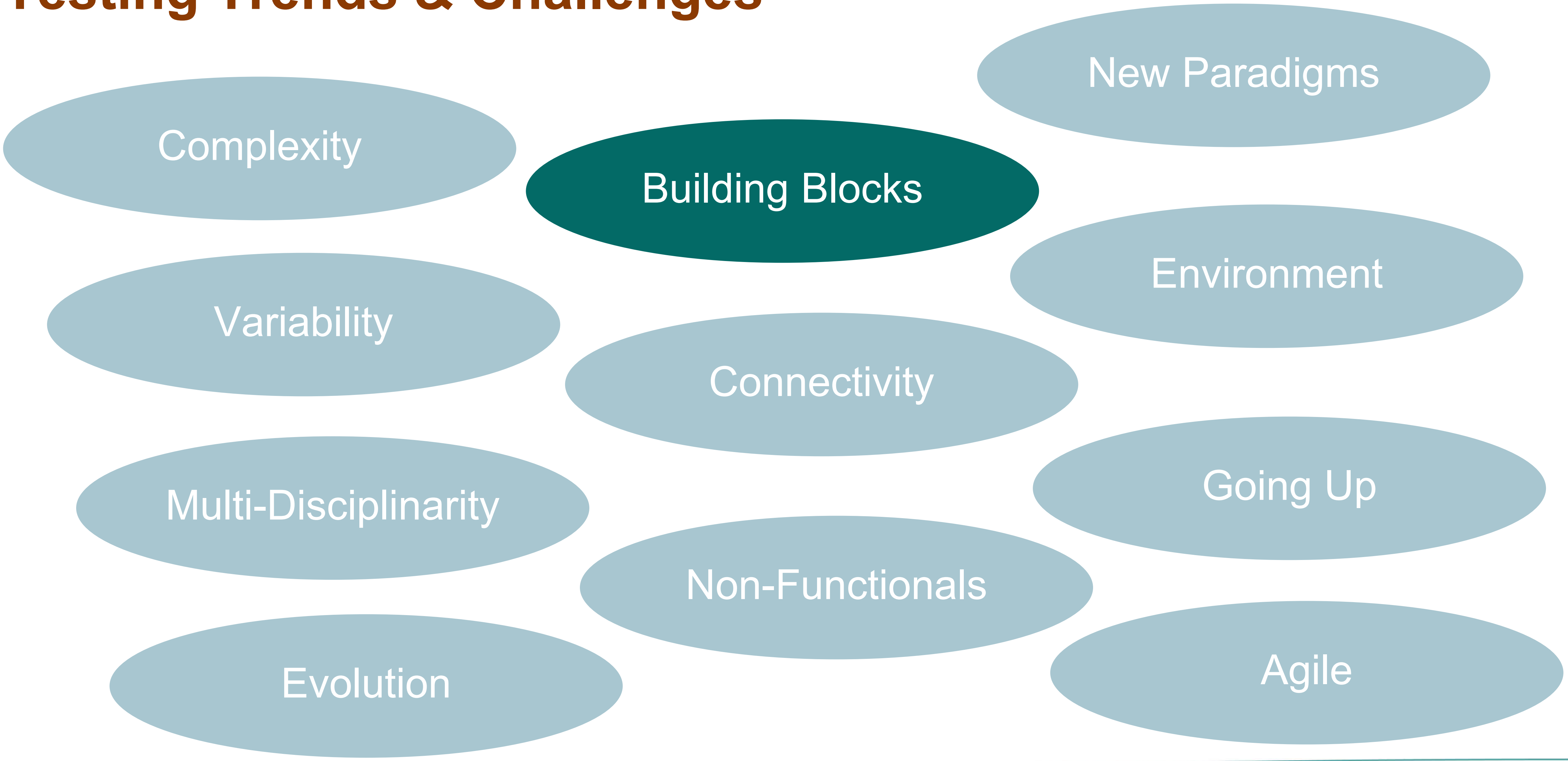


software group

Models for Multi-disciplinary Testing



Testing Trends & Challenges



Complexity

Variability

Multi-Disciplinarity

Evolution

Building Blocks

Connectivity

Non-Functionals

New Paradigms

Environment

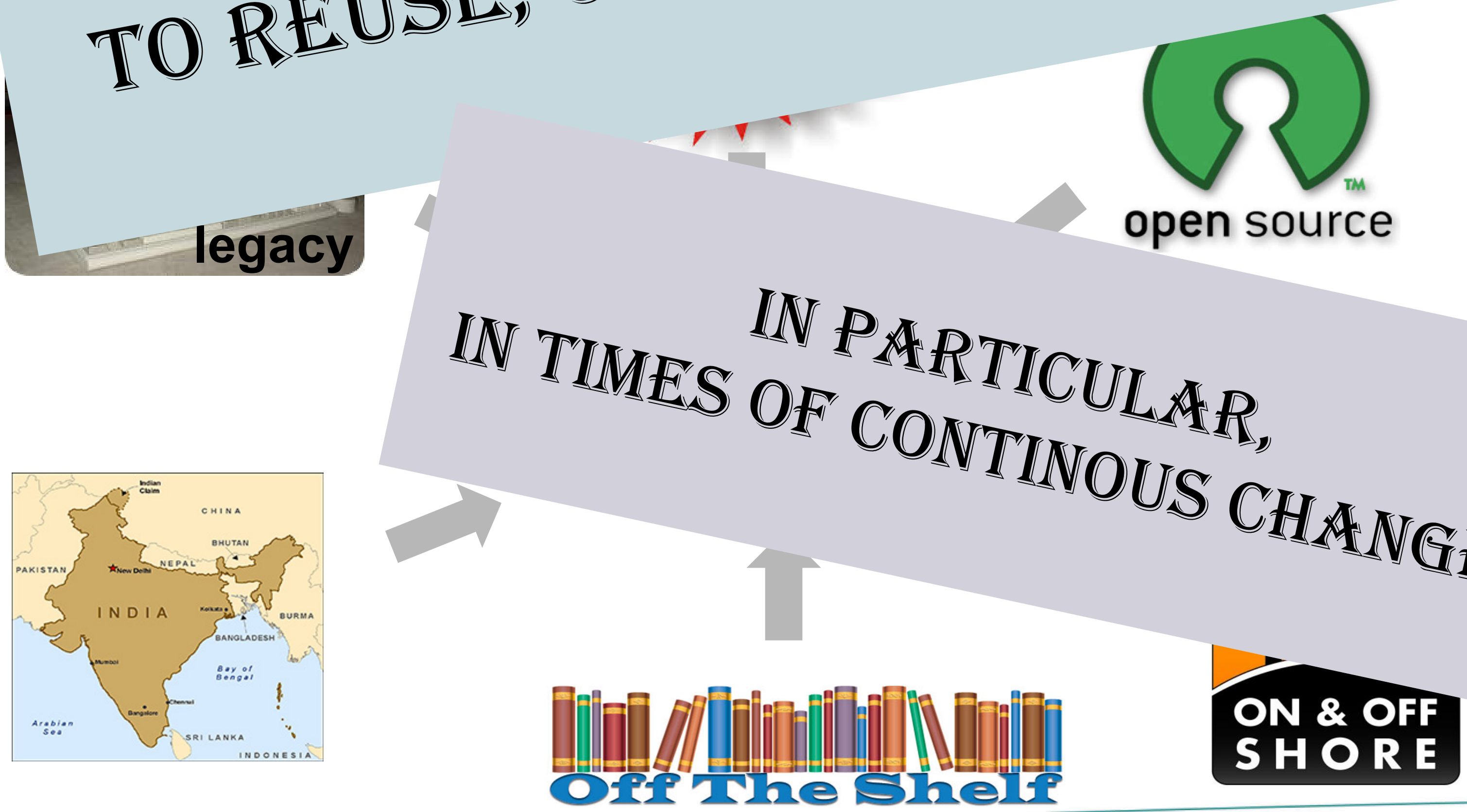
Going Up

Agile

Building Blocks : Components

Compon

TO REUSE, OR NOT TO REUSE



IN PARTICULAR,
IN TIMES OF CONTINOUS CHANGES

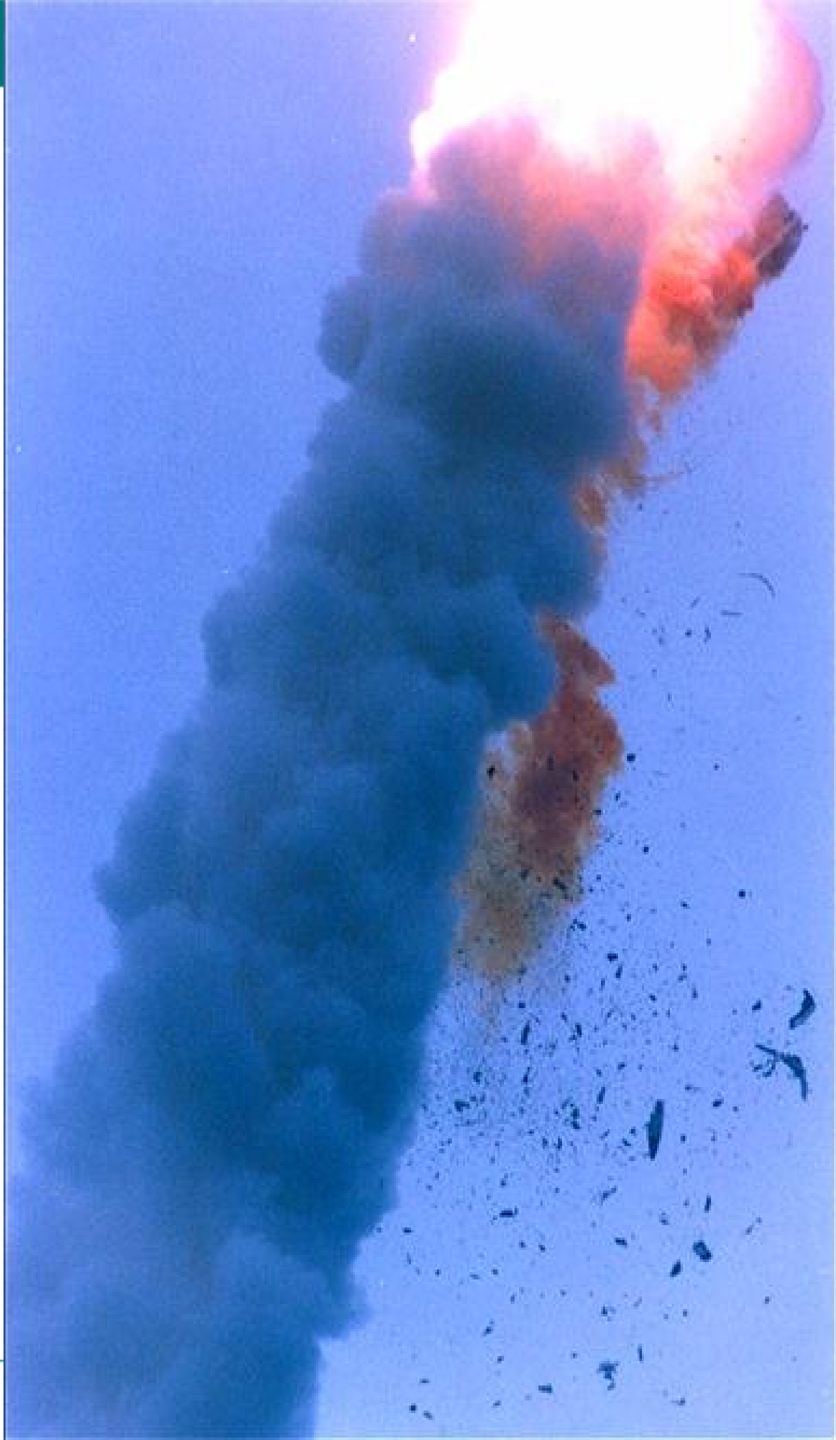
- reuse
- platform
- integration challenges
- dependencies
- when to test
- where to diagnose, repair

Components and Failures

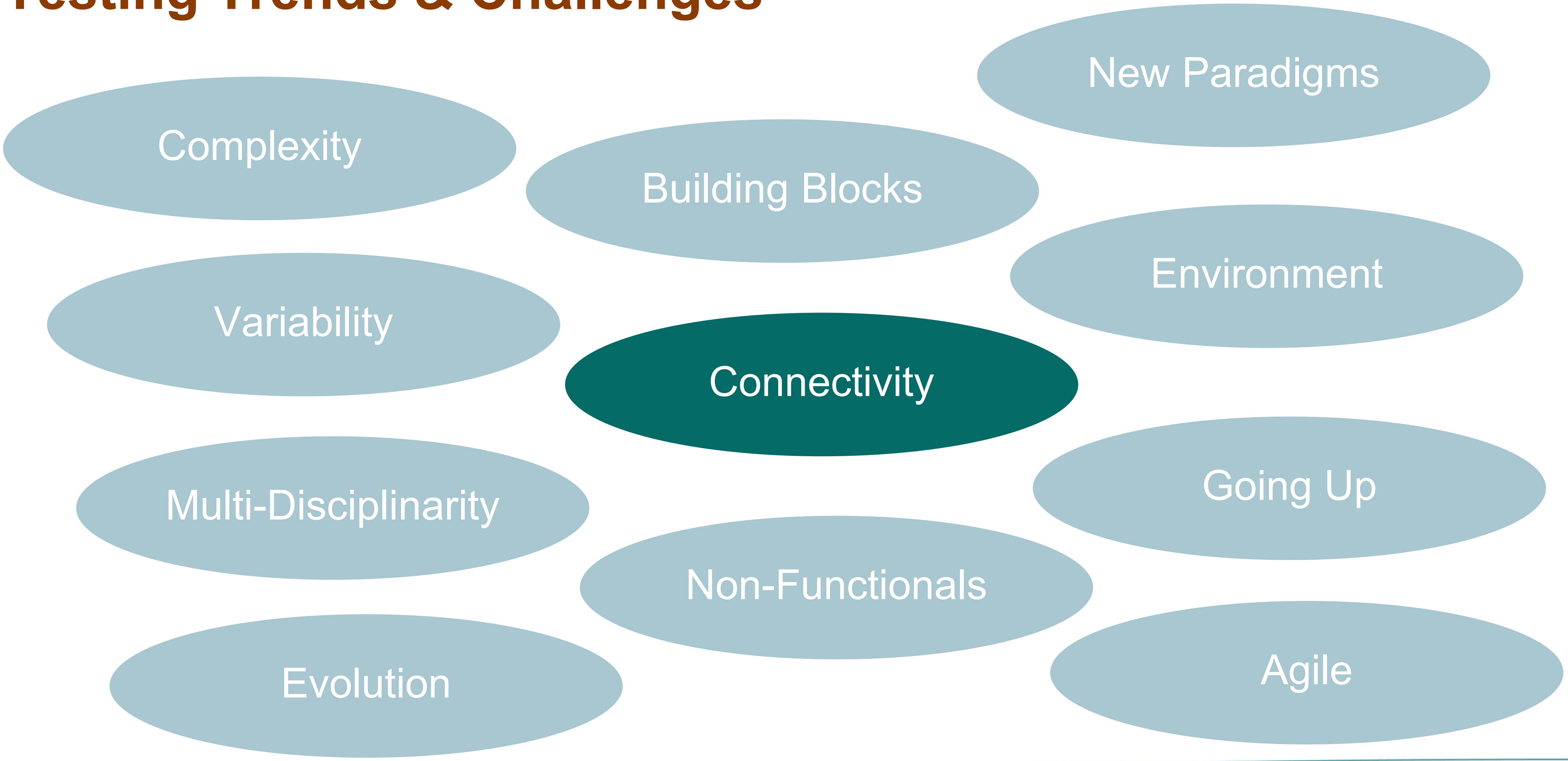
Ariane V rocket
- Design defects in control software

- Design
  - Exception handler assumed hardware errors only
  - Reuse of Ariane IV component in Ariane V without proper system testing

- Error
  - Software exception
- Failure
  - Mis-interpretation of diagnostic information



Testing Trends & Challenges



Complexity

Variability

Multi-Disciplinarity

Evolution

Building Blocks

Connectivity

Non-Functionals

New Paradigms

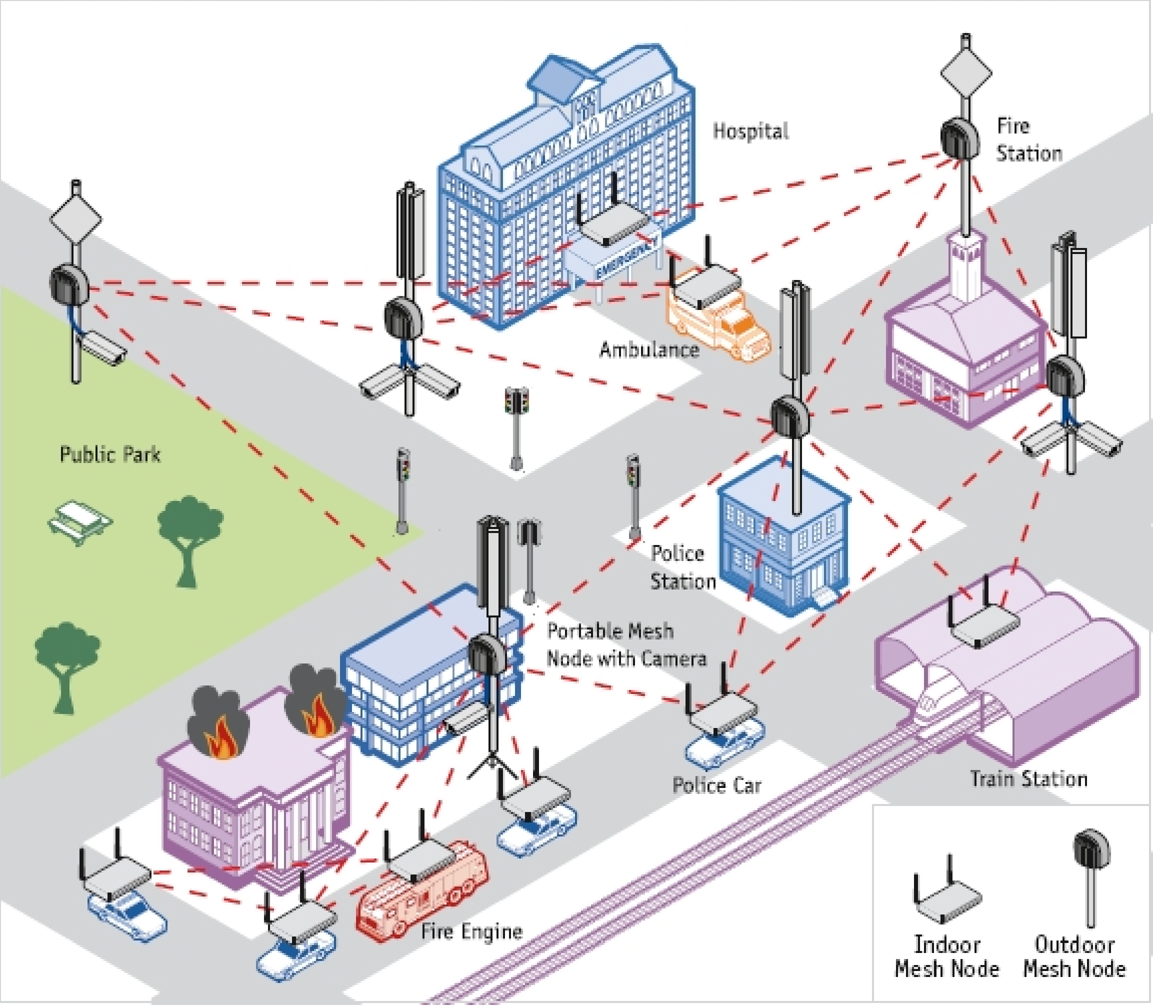
Environment

Going Up

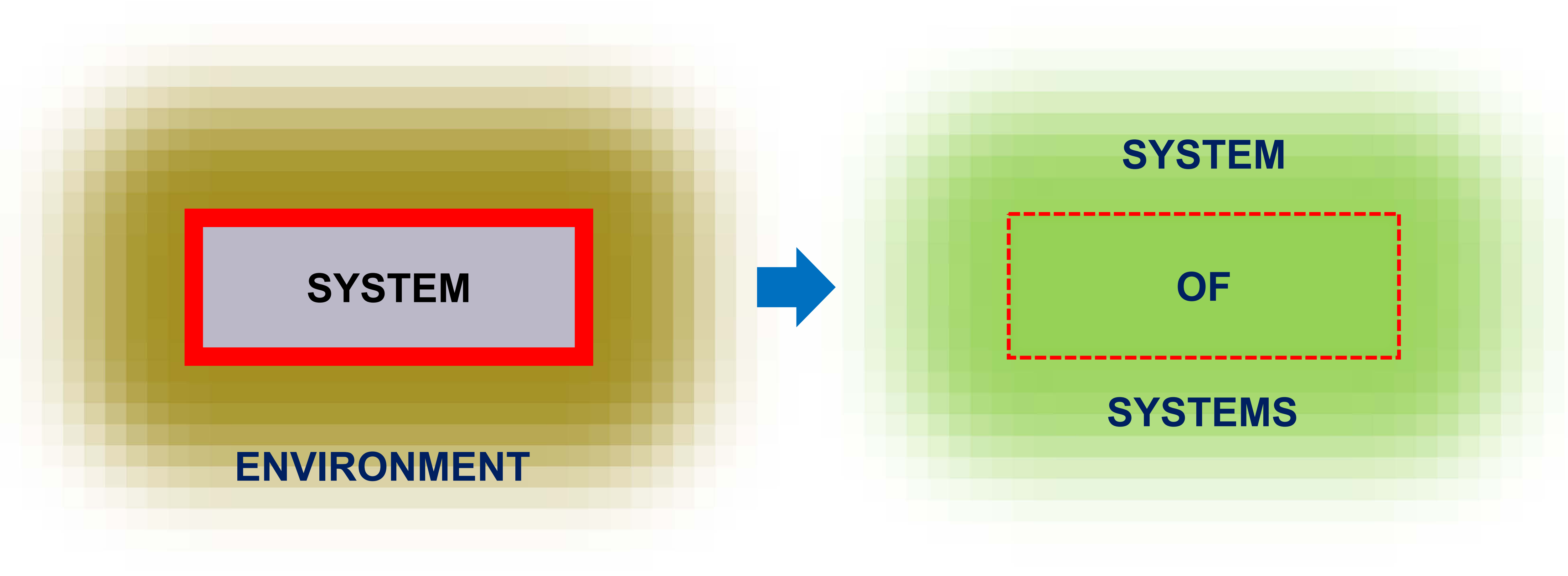
Agile

Connectivity

- Blurring boundaries of systems
  → everything connected
- Systems-Of-Systems
  - Dynamically connected systems
  - Not under own control
- Software is glue
  - with internal and external world
- Testing:
  - what is SUT ?
- Virtualization
  - which systems are available for testing ?
  - which systems must be virtualized?
- Dynamics
  - run-time testing and integration



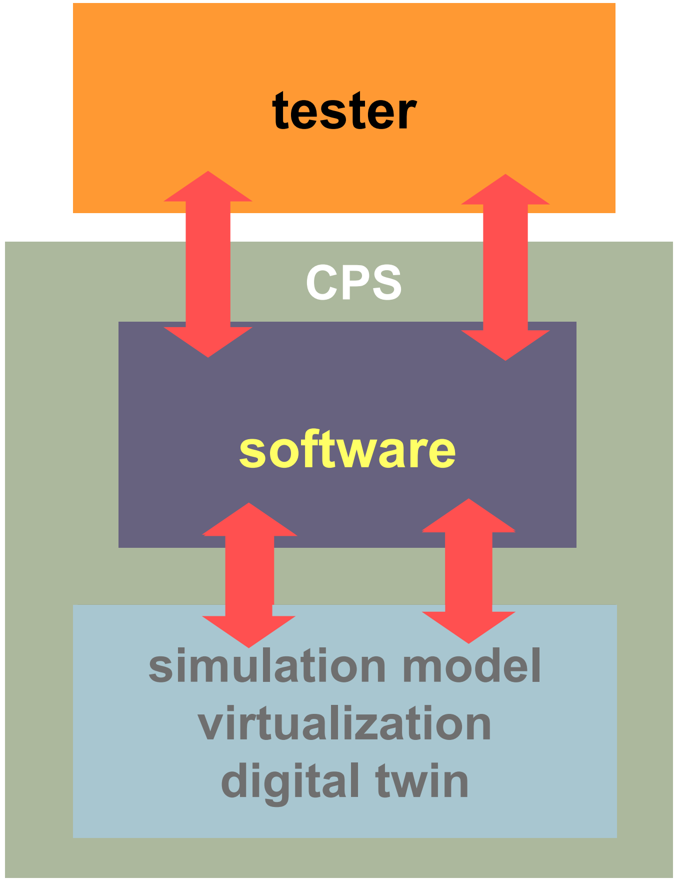
Fading Boundaries



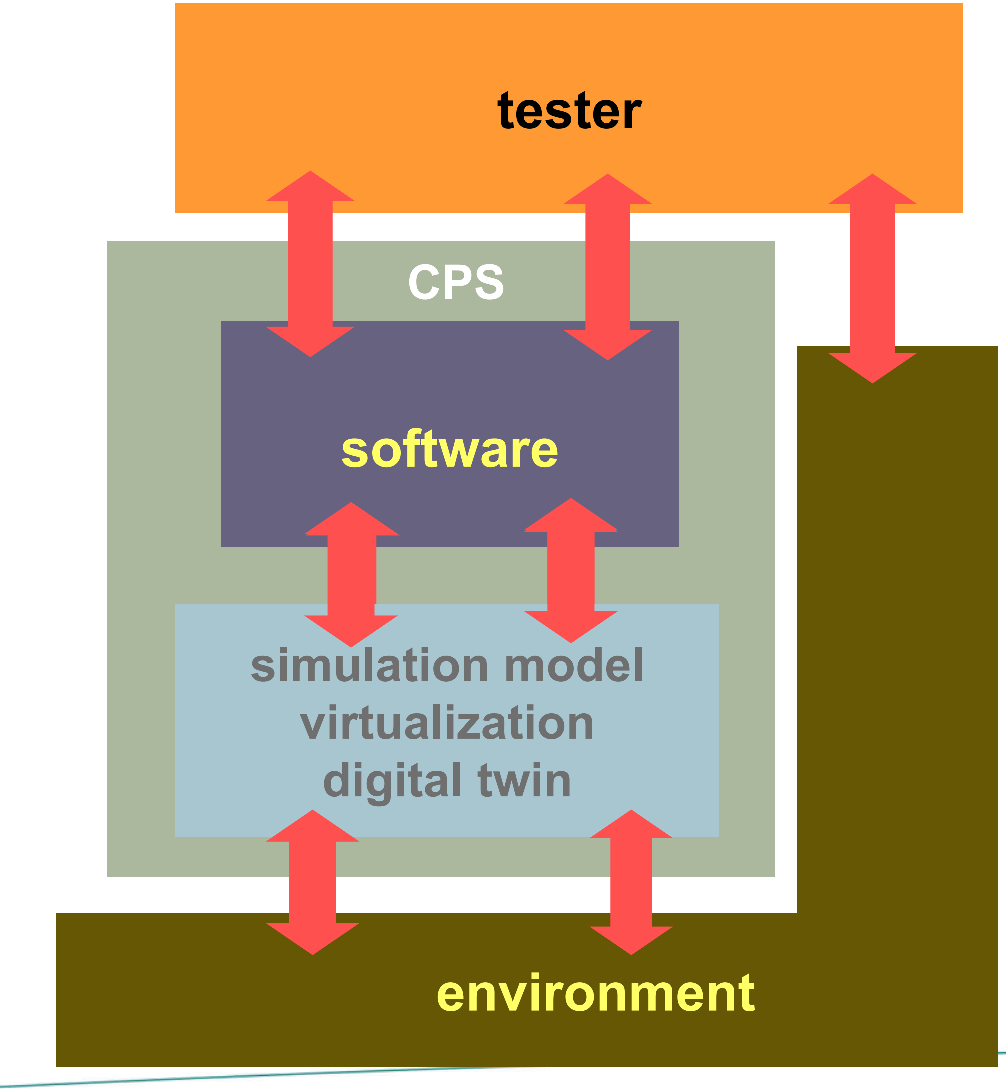
Testing Trends & Challenges

- Complexity
- Variability
- Multi-Disciplinarity
- Evolution
- Building Blocks
- Connectivity
- Non-Functionals
- New Paradigms
- **Environment**
- Going Up
- Agile

Environment

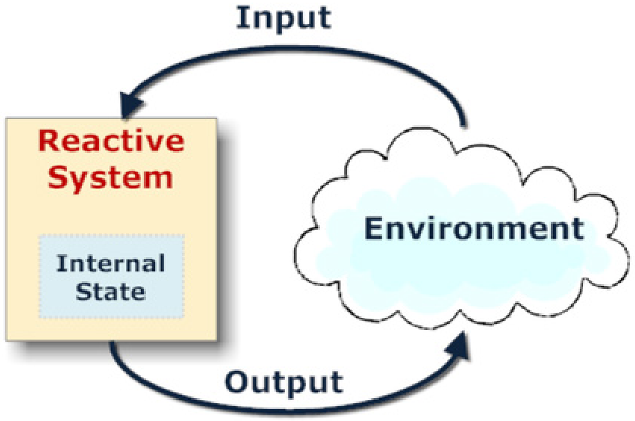


Environment



Environment

functional



state-based

autonomous

calculation : *I* → *O*

reactive: *I, S* → *O, S'*

tests over *I*

tests over *I, S*

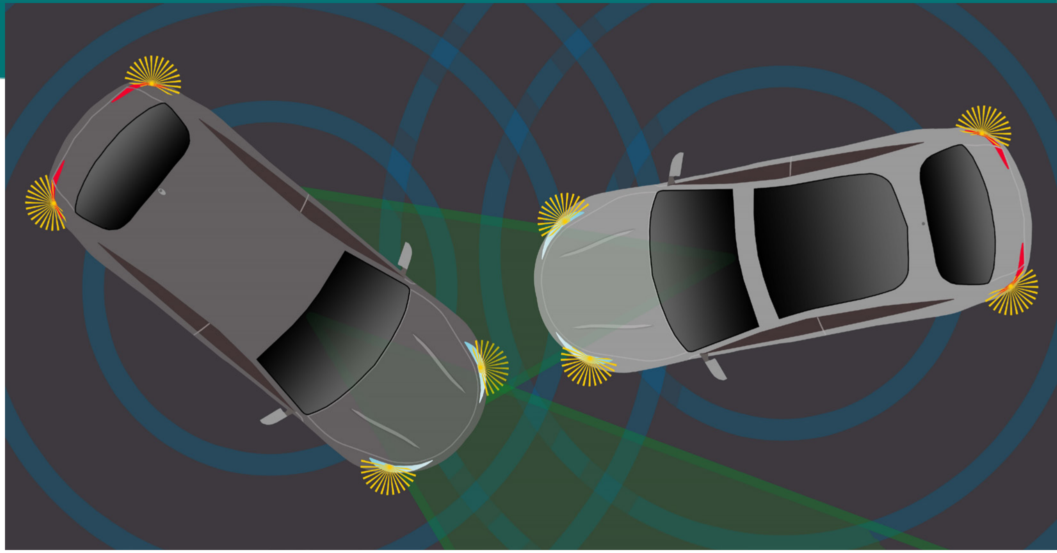
proactive : *I, S, E* → *O, S', E'*

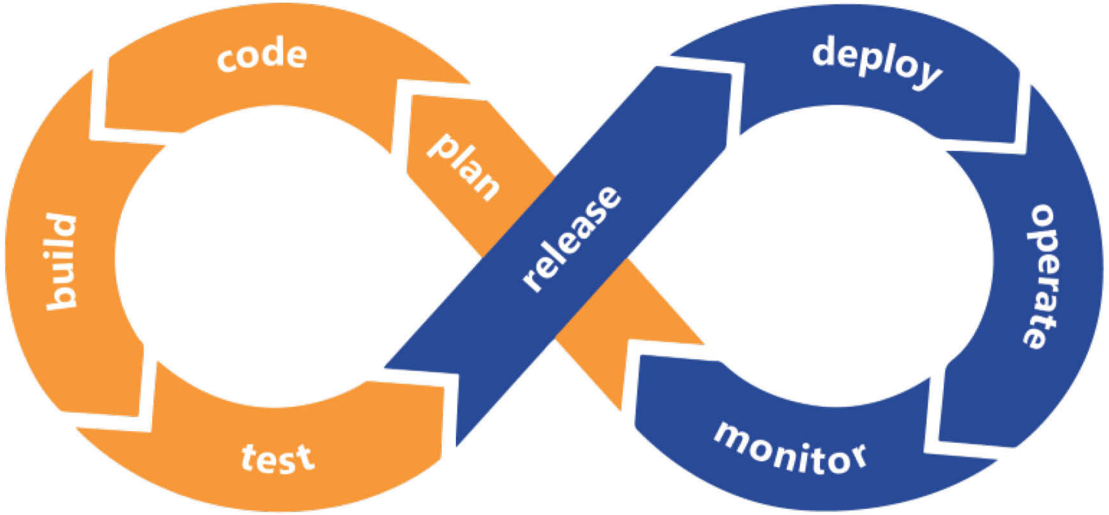
tests over *I, S, E*

for *safety*, *trustworthiness*, *dependability*,
the **environment** must be taken into account

Environment

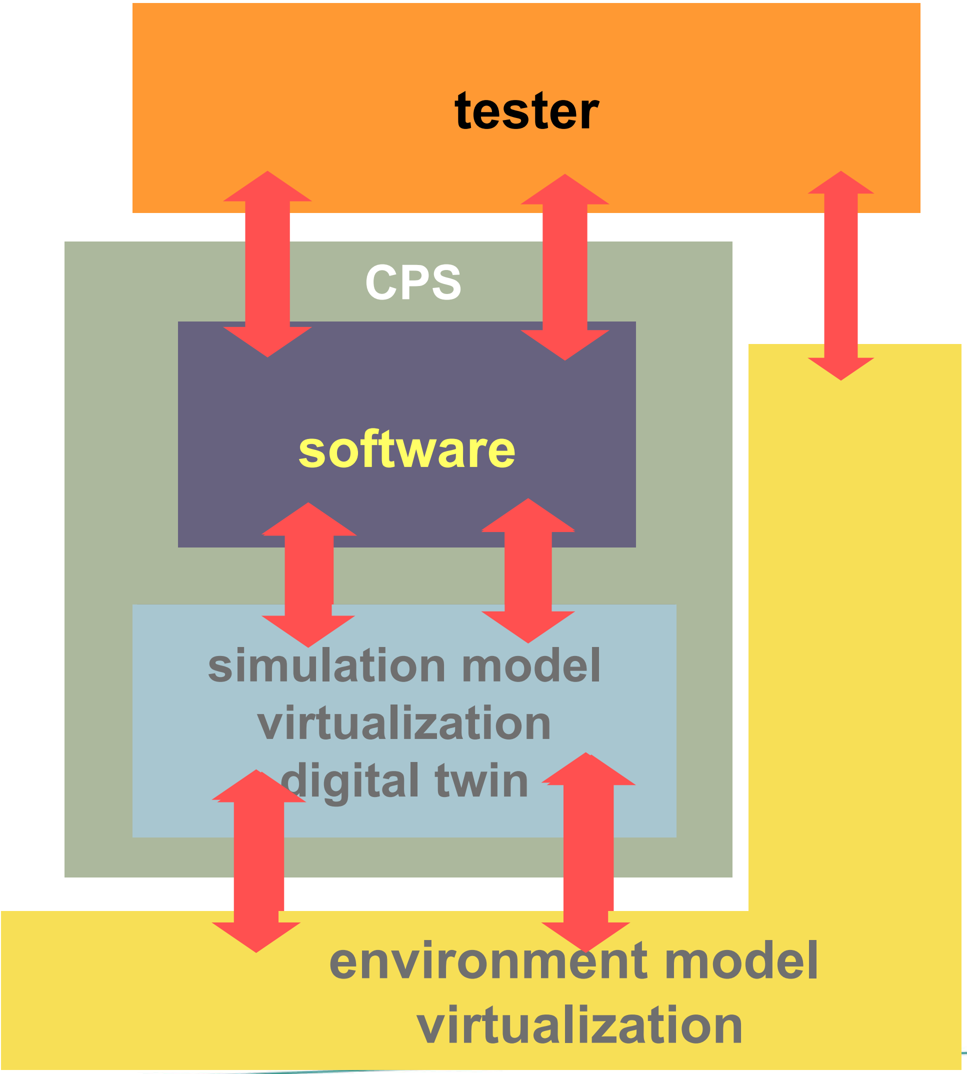
- Autonomous
  - → take part in environment
- Safety of autonomous cars
  - → test in all possible environments
- Environment
  - → not, or limited, under (test) control
- Environment
  - → can change
  - → new testing ?



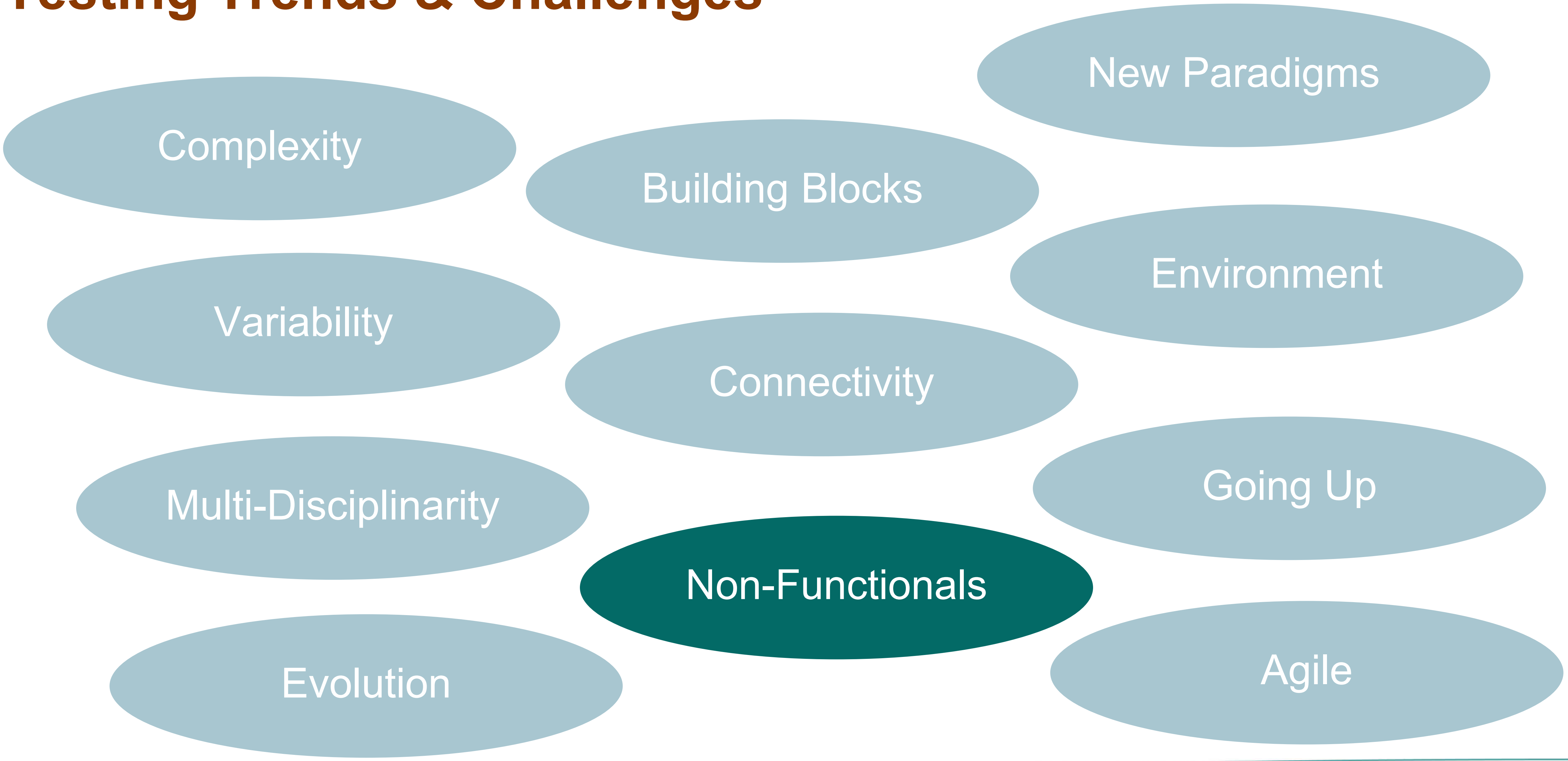


Testing everything before release is an illusion
→ continue quality control after release

Environment



Testing Trends & Challenges



Complexity

Variability

Multi-Disciplinarity

Evolution

Building Blocks

Connectivity

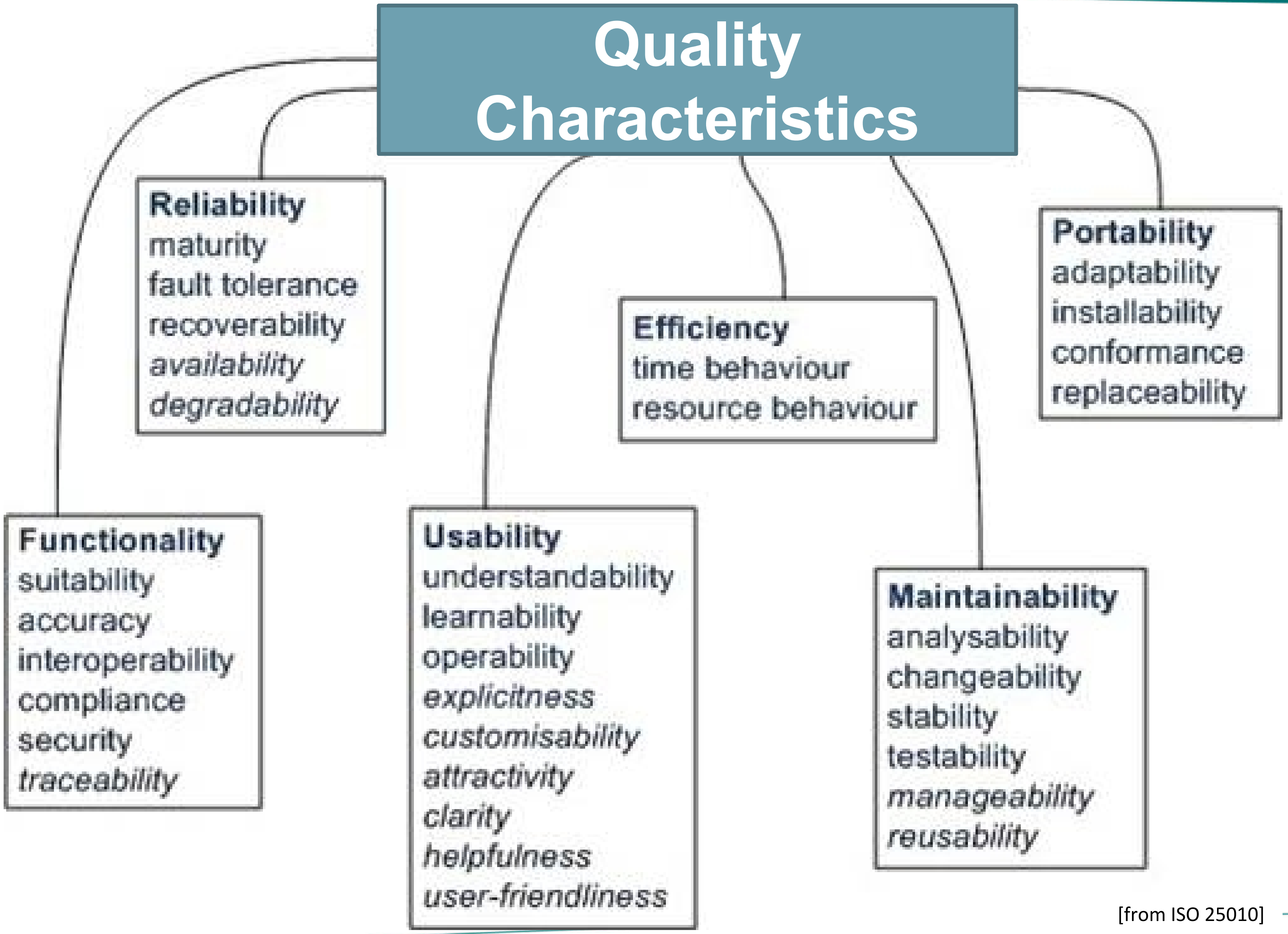
Non-Functionals

New Paradigms

Environment

Going Up

Agile



Quality Characteristics

Reliability
maturity
fault tolerance
recoverability
availability
degradability

Efficiency
time behaviour
resource behaviour

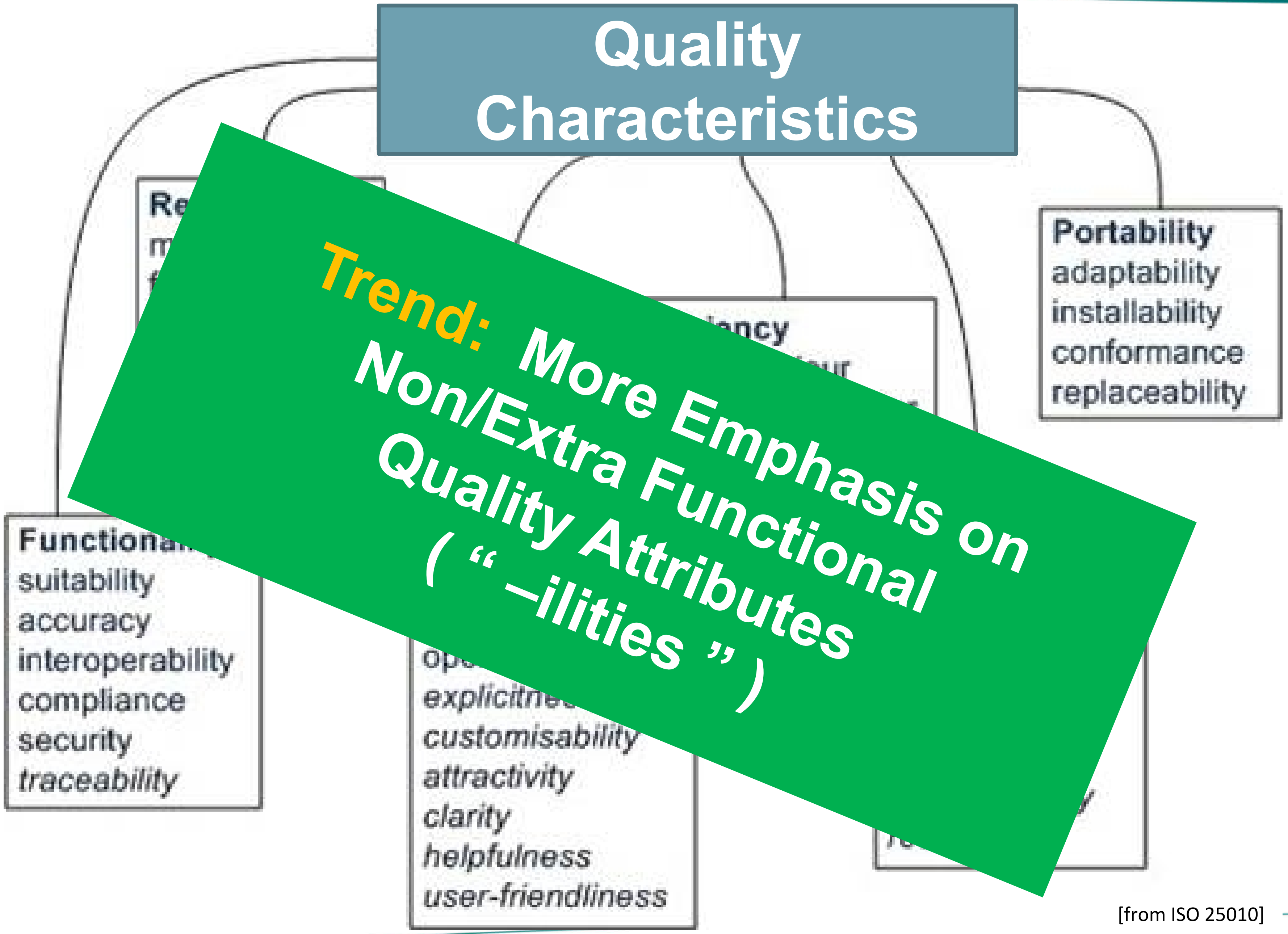
Portability
adaptability
installability
conformance
replaceability

Functionality
suitability
accuracy
interoperability
compliance
security
traceability

Usability
understandability
learnability
operability
explicitness
customisability
attractivity
clarity
helpfulness
user-friendliness

Maintainability
analysability
changeability
stability
testability
manageability
reusability

[from ISO 25010]



Quality Characteristics

Re
m
f

Portability
adaptability
installability
conformance
replaceability

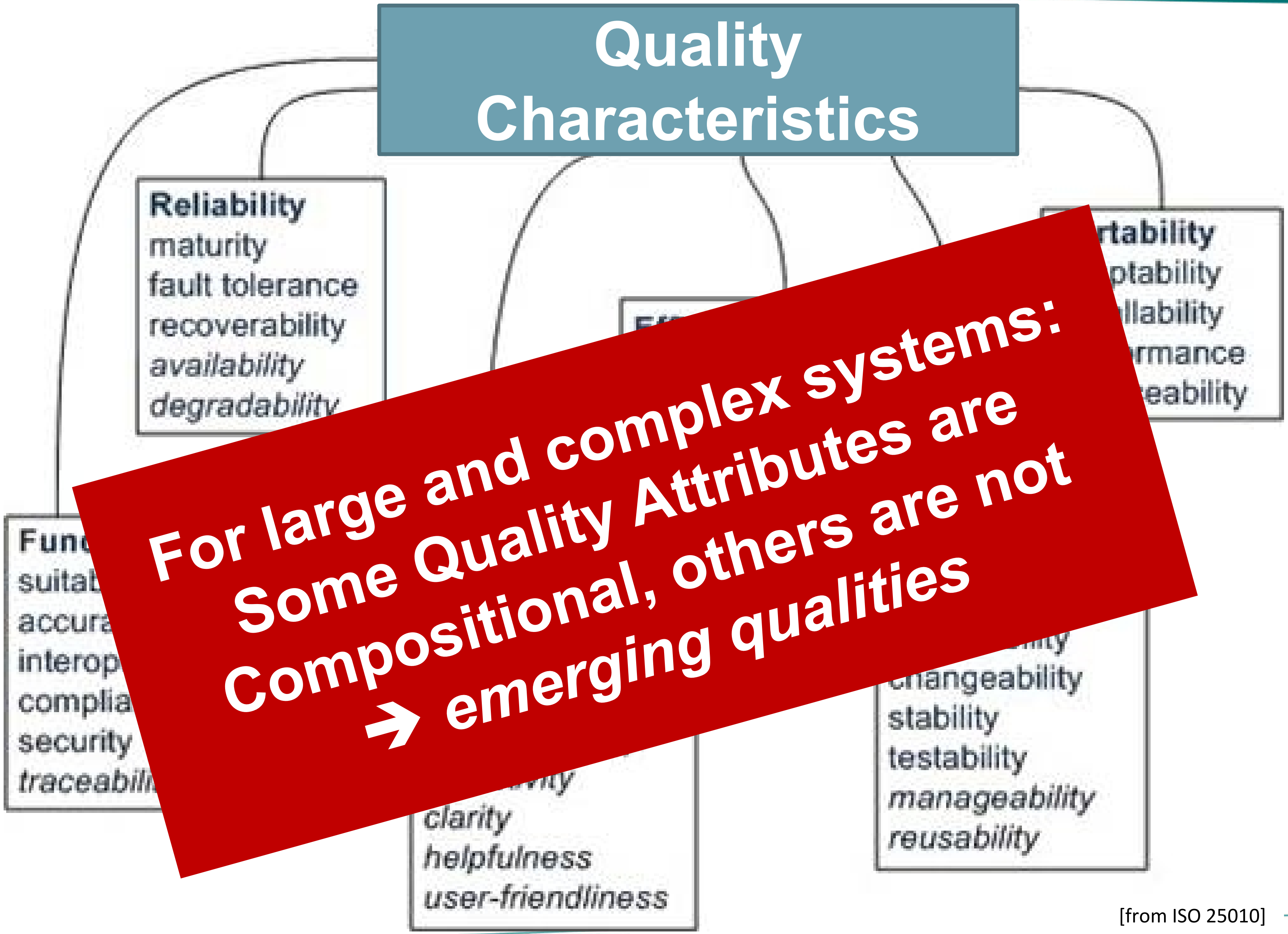
iency
our

Functiona
suitability
accuracy
interoperability
compliance
security
traceability

op
explicitne
customisability
attractivity
clarity
helpfulness
user-friendliness

Trend: More Emphasis on Non/Extra Functional Quality Attributes (“ –ilities ”)

[from ISO 25010]



Quality Characteristics

Reliability
maturity
fault tolerance
recoverability
availability
degradability

rtability
ptability
llability
rmance
eability

Eff

Func
suitab
accura
interop
complia
security
traceabili

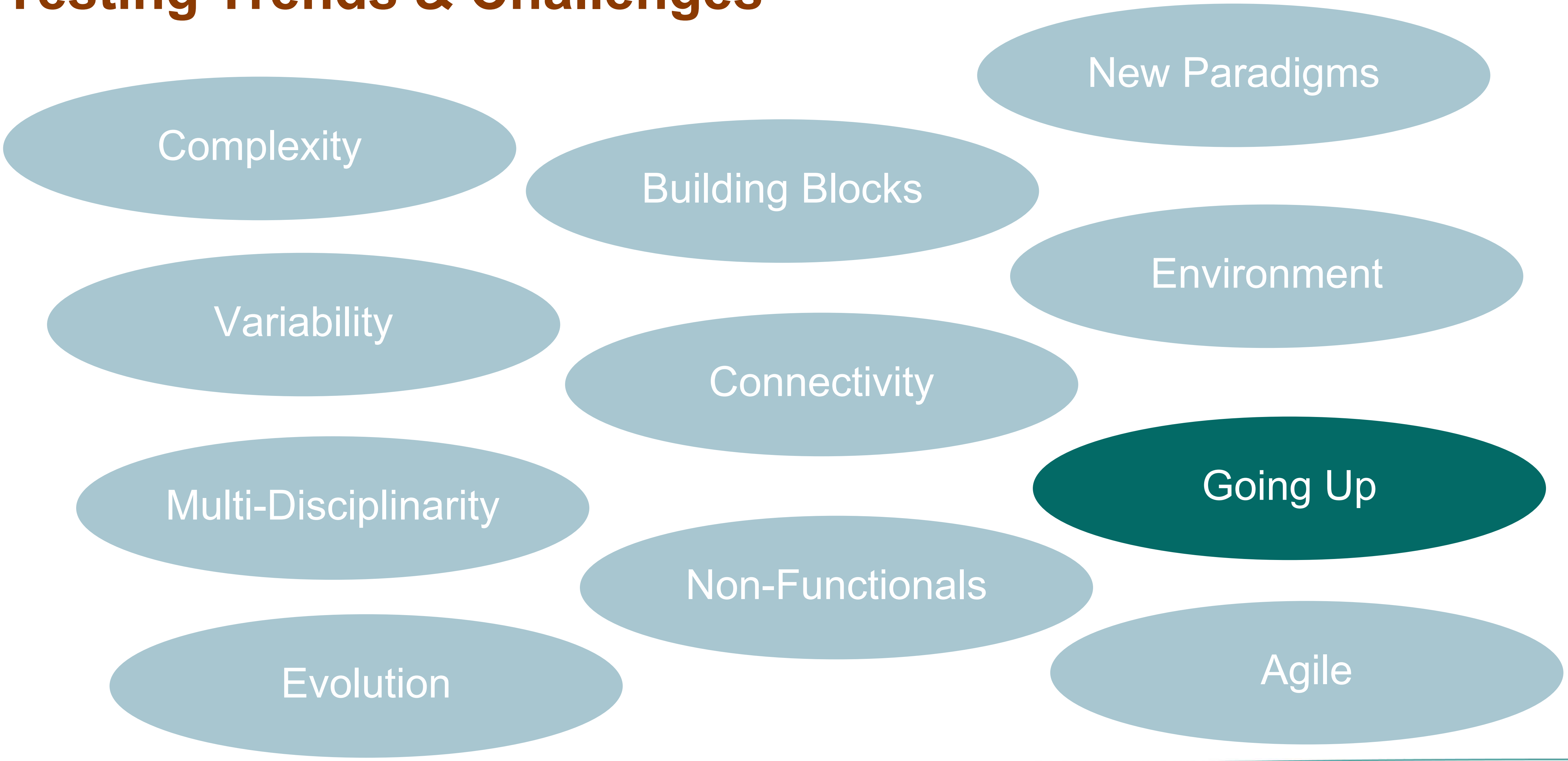
vity
clarity
helpfulness
user-friendliness

ity
changeability
stability
testability
manageability
reusability

For large and complex systems: Some Quality Attributes are Compositional, others are not ➔ emerging qualities

[from ISO 25010]

Testing Trends & Challenges



Complexity

Variability

Multi-Disciplinarity

Evolution

Building Blocks

Connectivity

Non-Functionals

New Paradigms

Environment

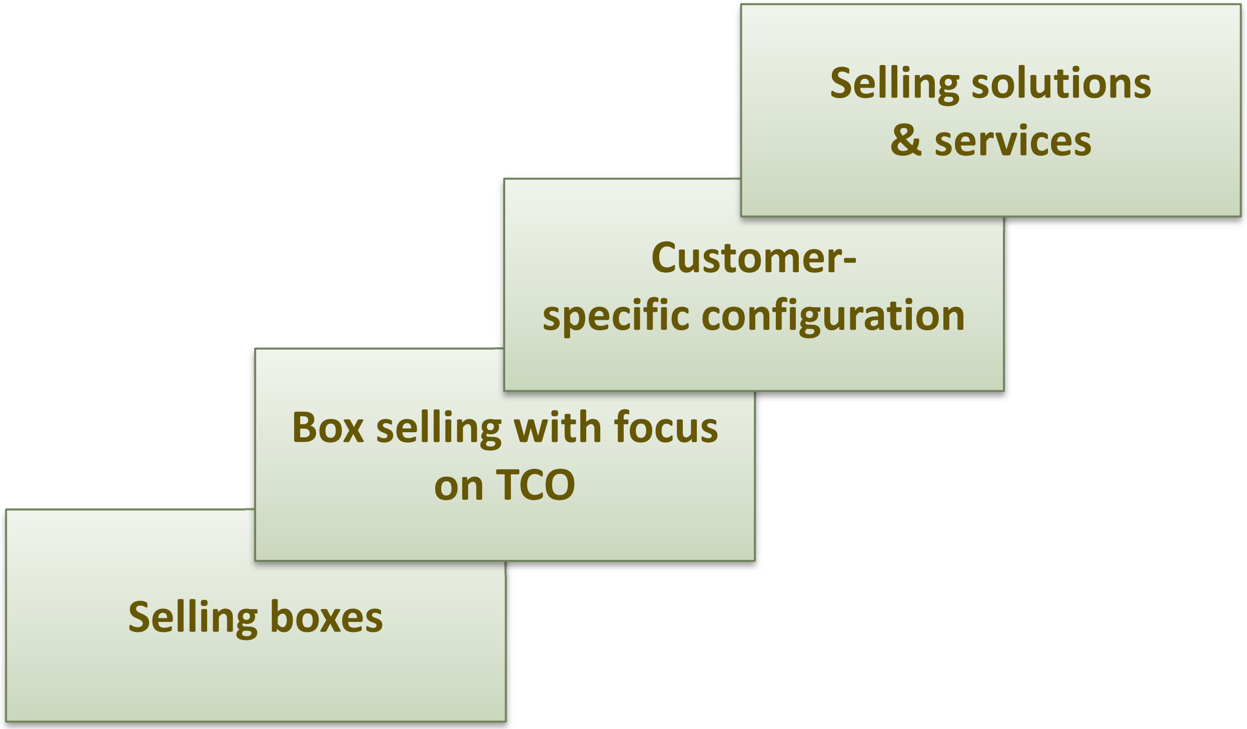
Going Up

Agile

Going Up

- In the Value Chain
  - new business models
  - testing quality-of-service

- In the Test Pyramid
  - everybody does unit tests
  - bugs are on the higher levels

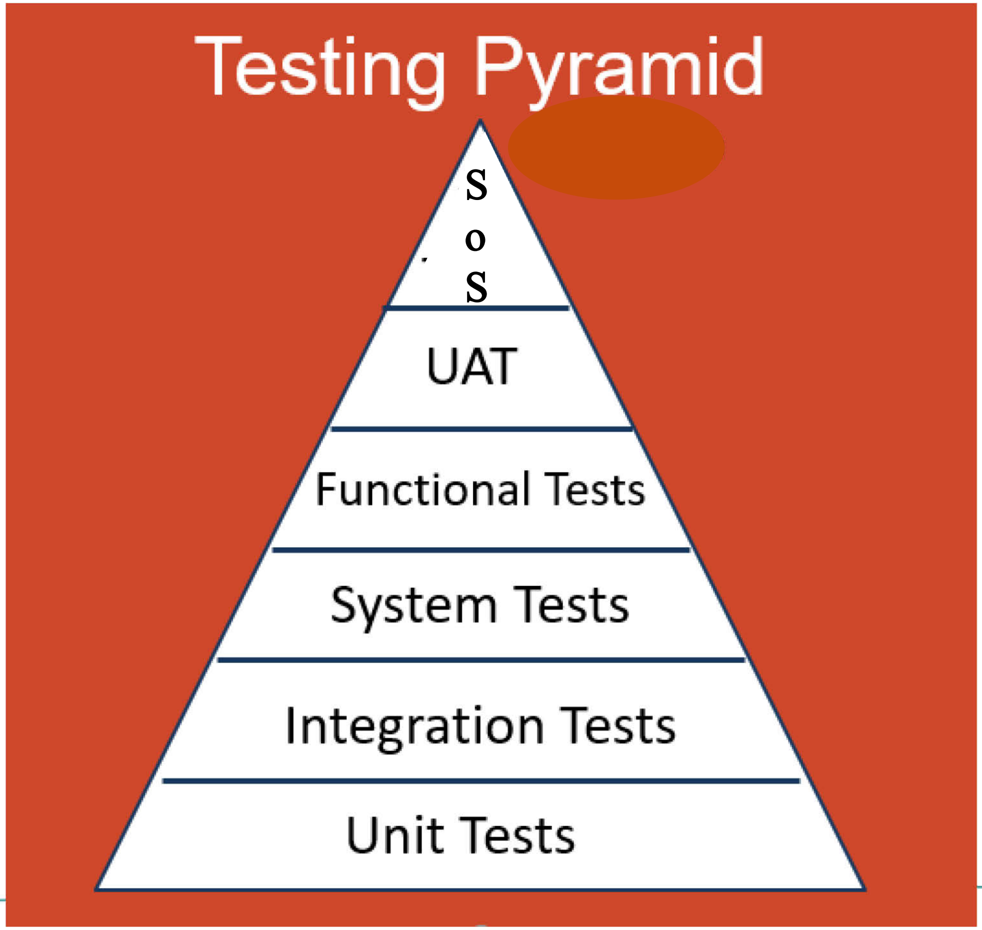


Selling solutions & services

Customer-specific configuration

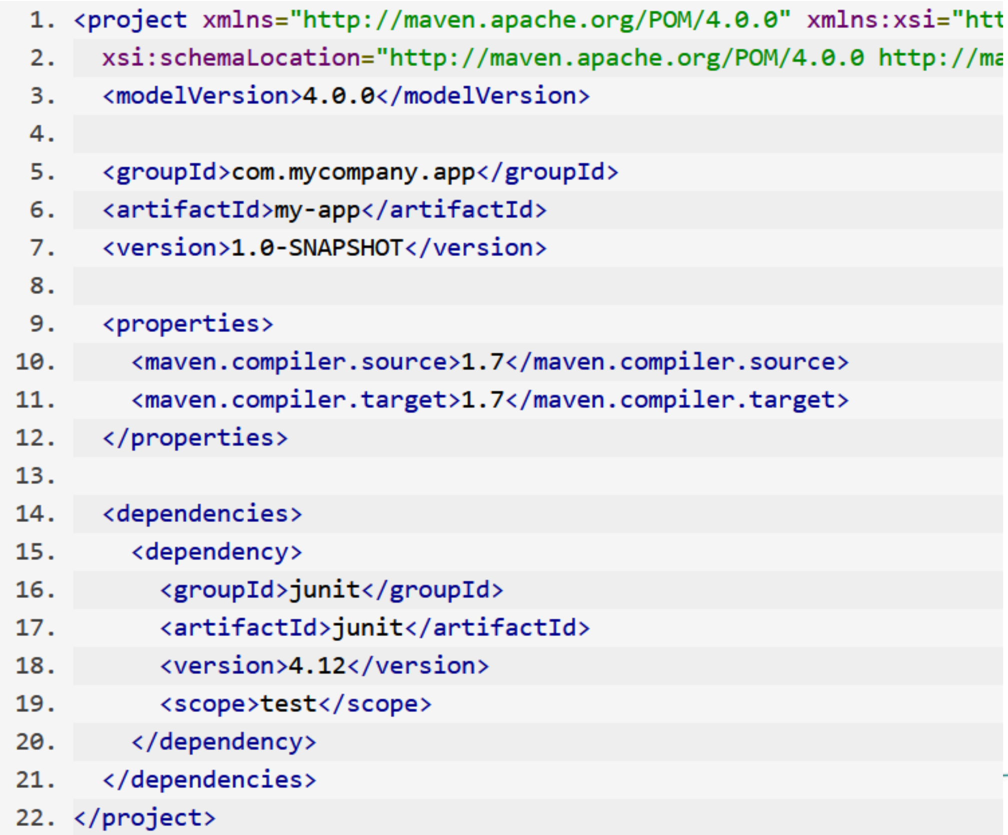
Box selling with focus on TCO

Selling boxes



Going Up

- In Coding
  - from software to meta-software: build tools, build scripts, configuration setting, . . .



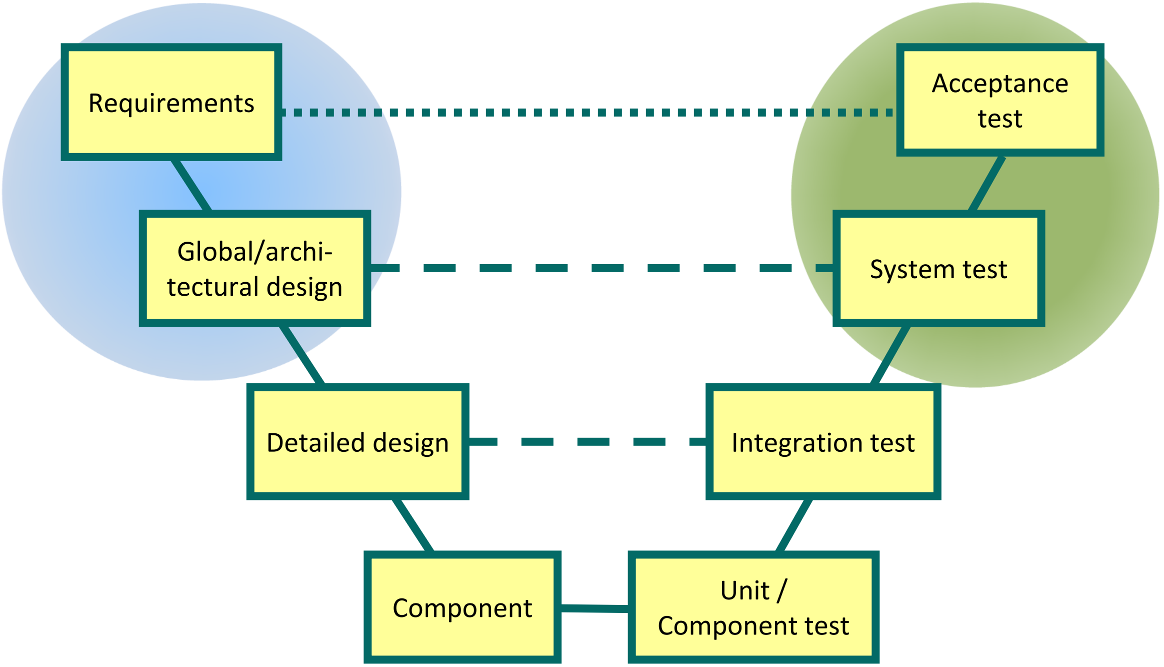
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="htt
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://ma
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

- In the V-Model
  - requirements, design, system test
  - detailed design, coding, unit tests outsourced

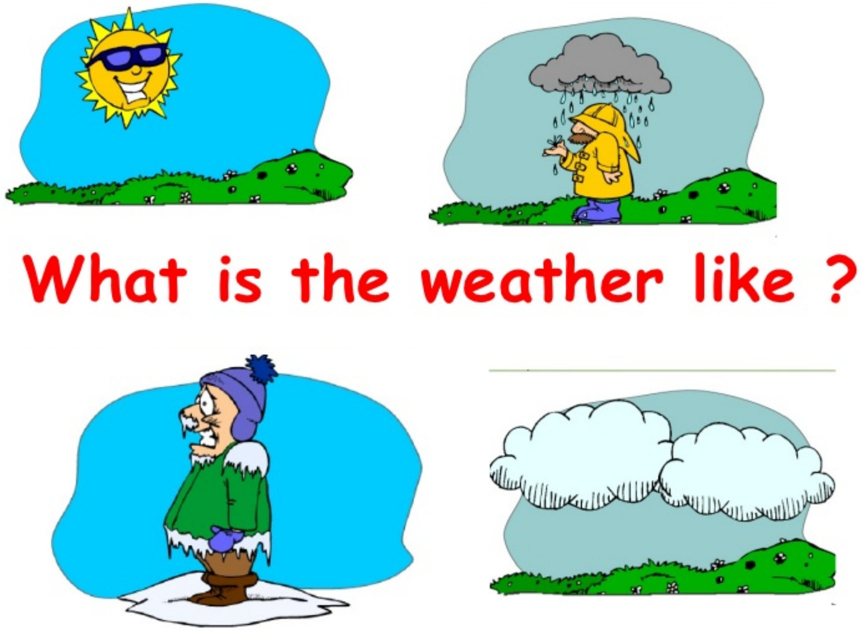


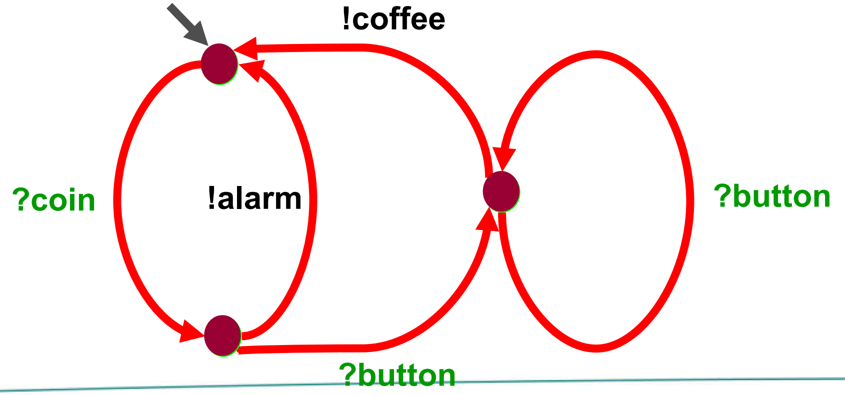
Going Up Consequence : Uncertainty & Non-Determinism

- Sometimes you don’t know …..
  - testing a search engine, weather forecast, …
  - systems-of-systems, big data, ...
- Sometimes you don’t want to know …..
  - no details
  - abstraction
  - particular view

Uncertainty of test outcomes & oracles

- **non-determinism**
- **probabilities**

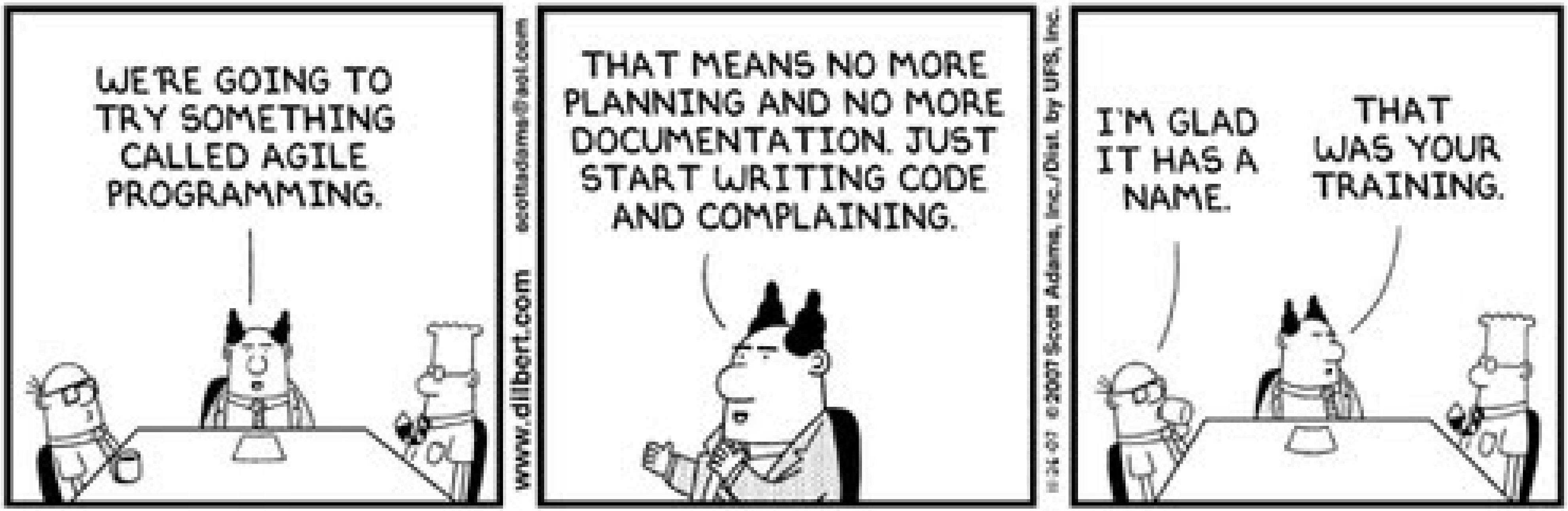




Testing Trends & Challenges

- Complexity
- Variability
- Multi-Disciplinarity
- Evolution
- Building Blocks
- Connectivity
- Non-Functionals
- New Paradigms
- Environment
- Going Up
- Agile

Agile ?


WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
scottadams@aol.com
www.dilbert.com
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
8-26-07 ©2007 Scott Adams, Inc./Dist. by UFS, Inc.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.

Agile

- Agile → test automation
  - test execution automation
  - test fast and often

→ Large repositories of scripted tests

- the night is too short
- traceability to requirements ?
- maintainability ?
- pesticide paradox : *how to increase variation in tests* ?

Agile - fallacy of complete specification:

We finally have the guts to admit that we don't know precisely what the system should do when we start coding.

Testing Trends & Challenges

- Complexity
- Variability
- Multi-Disciplinarity
- Evolution
- Building Blocks
- Connectivity
- Non-Functionals
- New Paradigms
- Environment
- Going Up
- Agile

New Paradigms and Technologies

- AI, Machine Learning
- Self-adaptive systems
- Quantum Computing
- Cloud
- Ethics, sustainability, …
- . . . . . .

Testing Trends & Challenges

- New Paradigms
- Complexity
- Building Blocks
- Environment
- Variability
- Connectivity
- Multi-Disciplinarity
- Non-Functionals
- Evolution