

# Accelerating Change

Frances Paulisch | Corporate Technology

# Digitalization impacts the entire value chain and every sector – Also Siemens

**SIEMENS**



**Digital  
Factory**

**Wind Power  
and  
Renewables**



**Power  
and Gas,  
Power  
Generation  
Services**



**Process  
Industries  
and Drives**

**Building  
Techno-  
logies**



**Energy  
Manage-  
ment**



**Healthiners**



**Mobility**

Products in all sectors are becoming increasingly individual –  
With an impact on all sectors and markets

SIEMENS

Flexibility

Flexibility in design and  
facility utilization

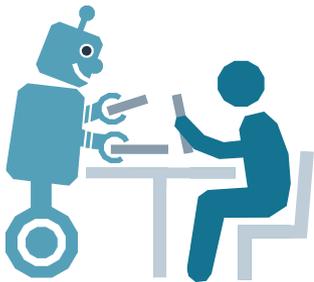
Time-to-market

Customer

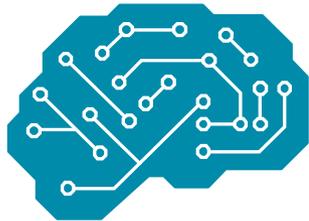
More  
product variants

Individualized  
products

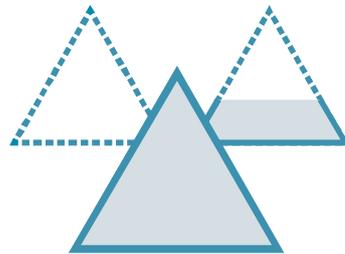
## Changing the way products are realized



**Advanced robotics**

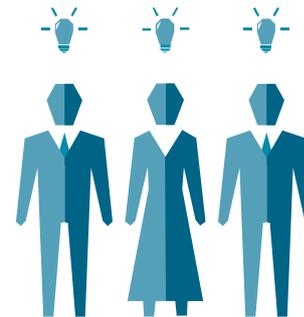


**Intelligent automation**

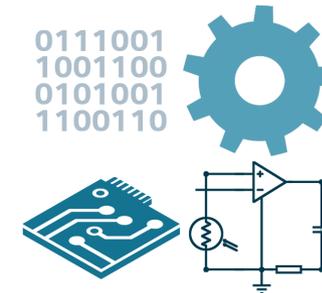


**Additive manufacturing**

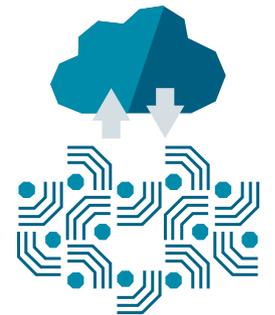
## Changing the way ideas come to life



**Crowdsourcing design**



**Systems thinking**



**Lifecycle analytics**

# Uncertainty is significant and growing in business, process, and technology

- Accelerating innovation cycles
- Fast feedback during and beyond development is essential (especially so for long-living systems)
- Increasing complexity
- Fast pace of change of software technologies
- Systems becoming much more interconnected (also across disciplines e.g. software/hardware)
- Humans “in the loop”

Software and digitalization are major drivers of such changes.

More features of mission-critical systems are realized in software.

Many past approaches are better suited for predictable environments.



“Walking on water and developing software from a specification are easy ... if both are frozen.”

Edward V. Berard

# Dealing with Complexity – A range of approaches depending on situation

## Drivers for Complexity

- Increasing functionality
- Quality attributes (non-functional requirements)
- Especially increasing flexibility
- History influences current behavior
- Feedback loops or non-linear behavior
- Poor decomposition, structure, interfaces
- Function-element allocation
- Embedded critical external
- Legacy or reuse elements
- Open interfaces/unclear scope
- Increasing technical debt
- Critical partner or supplier relationships
- (Geographically) distributed development

## Cynefin Framework<sup>1)</sup>

### Complex

#### Domain of emergent solutions

Cause and effect may be determined in retrospect

Probe, Sense, Respond

**Emergent**

### Complicated

#### Domain of good practices

Cause and effect can be analyzed with sufficient data

Sense, Analyze, Respond

**Good practice**

### Disorder

### Chaotic

#### Domain of novel solutions

No (discernible) relation between cause and effect

Act, Sense, Respond

**Novel**

### Simple

#### Domain of best practices

Cause and effect are understood and predictable

Sense, Categorize, Respond

**Best practice**

Source: Siemens System Architect Core Learning Program

1) A Leader's Framework for Decision Making, David Snowden and Mary Boone

## From “Embrace Change” to “Accelerate Change”

### Change

- Don't apply an approach more suited to the “simple” or “complicated” case when you are in a “complex” or even “chaotic” situation
- Learn to “surf” and embrace change – as well as possible within other constraints
- Apply set of approaches that help you to master and accelerate change
- Change as an Opportunity



“Instead of learning to surf, conventional organizations try to control the waves! This almost never works.”<sup>1)</sup>

Allen Ward

1) Cited by Mary Poppendieck in “Learning to Surf” keynote at ICSE 2013

## Depending on the context, choose appropriate approach

### Balance

- Discipline vs. agility
- Up-front planning vs. more continuous approaches
- Centralized vs. decentralized decision making
- Existing situation and future vision
- Single monolithic vs. set of microservices



“Becoming agile means accepting uncertainty about the future as a way of dealing with the future. Any project development effort should be a balance between anticipation (planning based on what we know now) and adaptation (responding to what we learn over time).”

James Highsmith

# Avoid

## Considerations for Avoiding Effort

- Systematic reuse
- Open Source
- Beware of “over-engineering”
- Only develop/maintain features that are really used
- Avoid waste from “silos” / handoffs
- Risk-based approaches

“There is nothing so useless as doing efficiently that which should not be done at all.”

Peter Drucker



# Anticipate

## Considerations for Anticipate

- Set up so that one expects and can deal with the unexpected
- Shift from mainly “prevent” to mix of prevent and react
- Resilience
- Avoid surprises, right balance not to lock down
- Allow freedom but be watchful e.g. early warning system
- Set up (perhaps automated) alarms

“By failing to prepare, you are preparing to fail.”

Benjamin Franklin



Creative Commons Crystal Ball by Christian Schnettelker licensed under CC BY 2.0

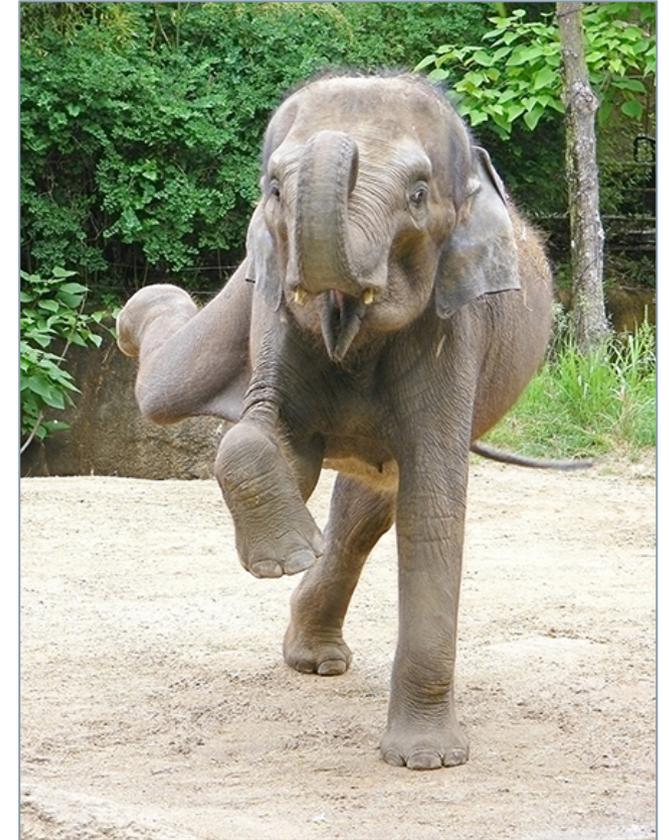
# Agile

## Considerations for Agile and Lean

- Optimize the entire value stream
- Build quality in from the beginning
- Deliver fast and create knowledge
- Establish pull and flow
- Do not plan whole project in detail, but the sprint

“Good people with good process will outperform good people with no process every time.”

Grady Booch



# Architecture

## Considerations for Architecture

- Decomposition, structure, interfaces
- Avoid dependencies
- Strive to build on trunk
- Quality attributes
- Adaptive partitioning



“Software and cathedrals are much the same — first we build them, then we pray.”

Sam Redwine

# Autonomous

## Considerations for Autonomous

- Avoid dependencies
- Encapsulate
- Different paces of change (e.g. operating system, apps)
- Fail-safe, fail-passive, fail-operational
- Self-adaptive, resilient



Creative Commons iphone 4 apps by Sean MacEntee is licensed under CC BY 2.0

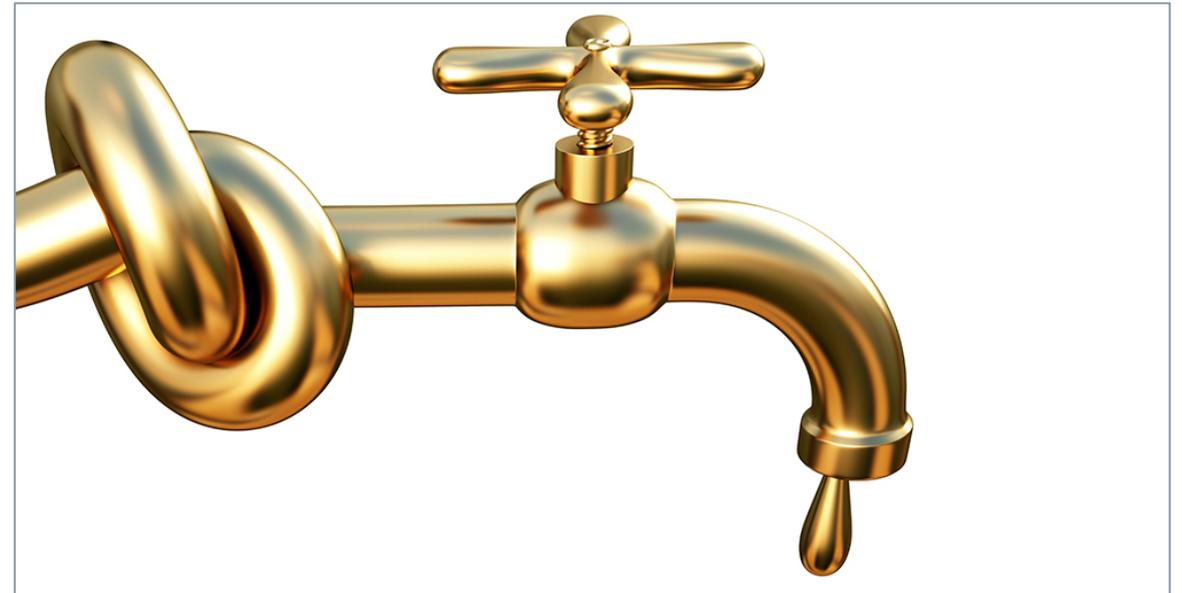
“Factor your systems so that artifacts that change at similar rates are together.”

Brian Foote and Joe Yoder

# Automation

## Considerations for Automation

- Significant investment also in infrastructure and automation
- Delivery pipeline
- Infrastructure as Code
- Test methodology AND Test Infrastructure (Test Architect)
- Concern the tests themselves and how to (automatically) evaluate the test results
- Testing of non-functional requirements



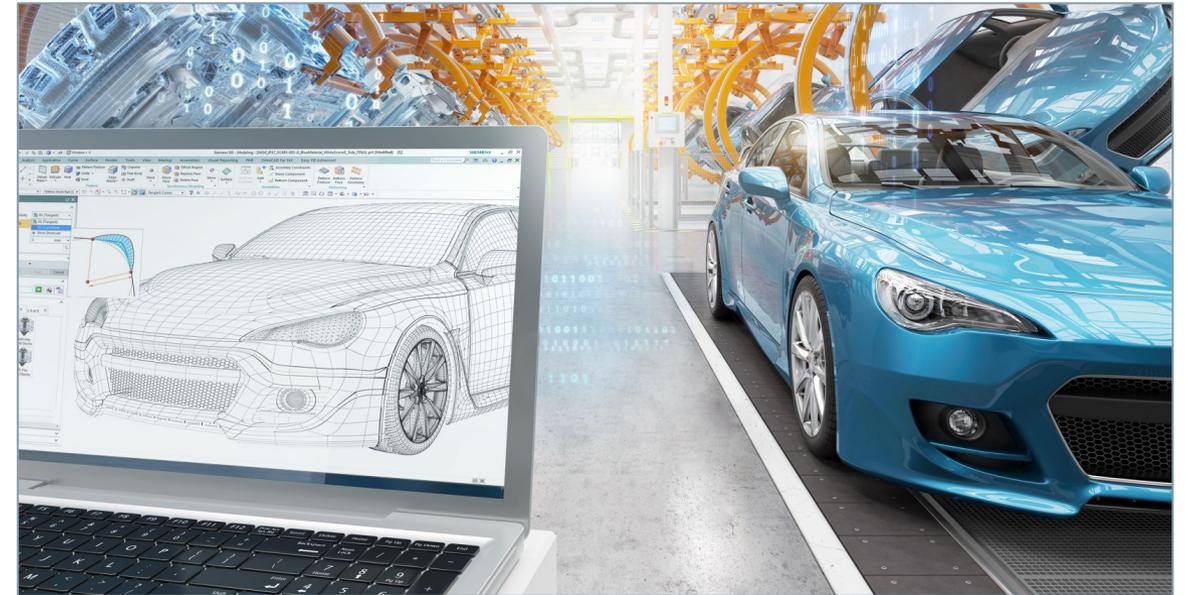
“Give me six hours to chop down a tree and  
I will spend the first four sharpening the axe.”

Abraham Lincoln

# Abstraction

## Considerations for Abstraction

- Model-based ...
- Digital Twin
- Simulation
- Early testing of non-functional requirements
- Early feedback



“The entire history of software engineering is that of the rise in levels of abstraction.”

Grady Booch

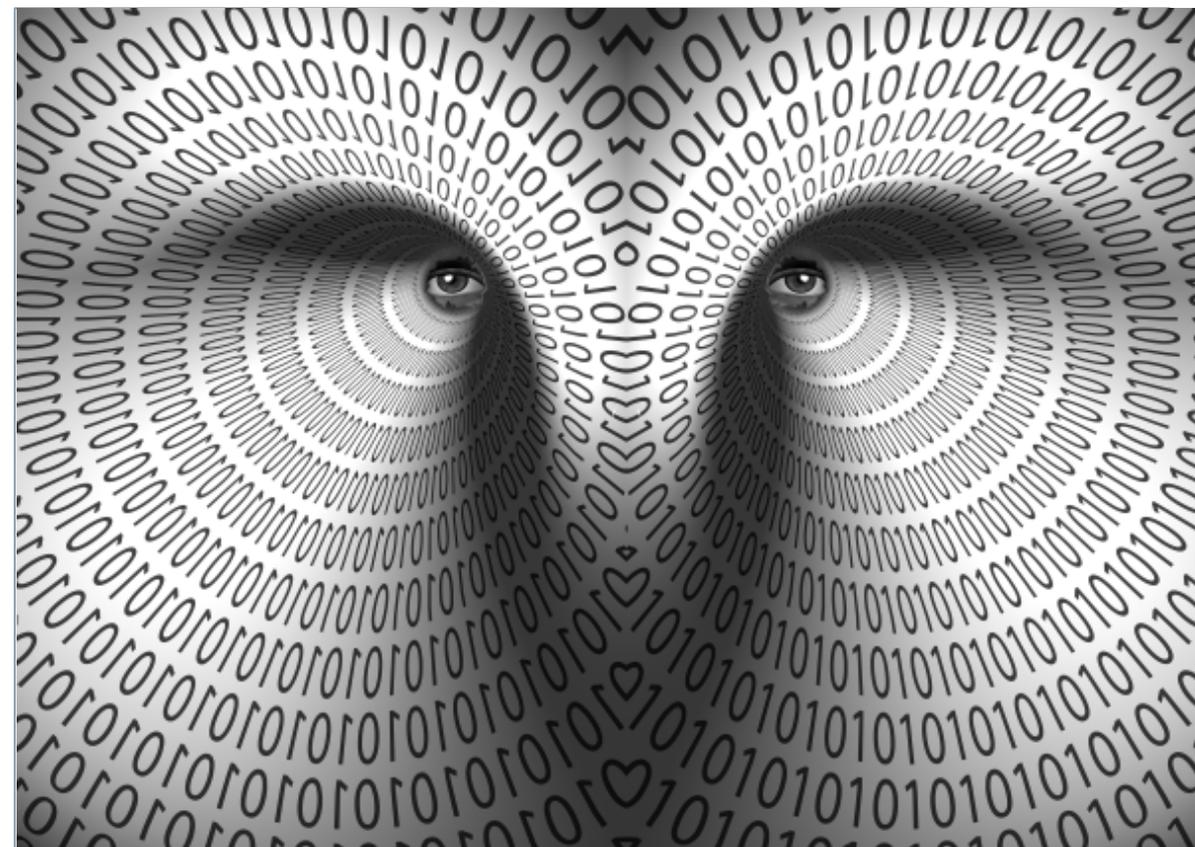
# Analytics

## Considerations for Analytics

- Take advantage of data in development and operations
- Build in run-time testing, online monitoring
- Operational Intelligence

“About 90 percent of the downtime comes from, at most, 10 percent of the defects.”

Barry Boehm



# Ability

## Considerations for Ability

- Roles
- Formats (in-person, online, coaching, networking,...)
- Consistent across roles (different perspectives)
- Collaboration

“Better train people and risk they leave — than do nothing and risk they stay”

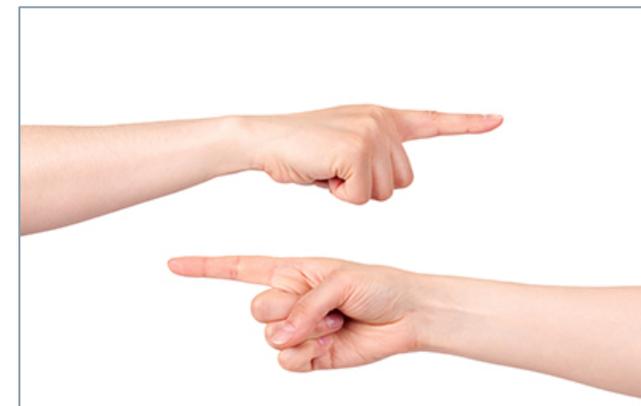
Anonymous



# Accept/Culture

## Considerations for Culture

- This is often the biggest hurdle
- Empowerment
- Motivation
- Learn from Feedback
- Common core principles
- Failure Culture (First Attempt in Learning)



“People are not afraid of failure, they’re afraid of blame.”

Seth Godin

## Accelerating Change through ...

Avoid

---

Anticipate

---

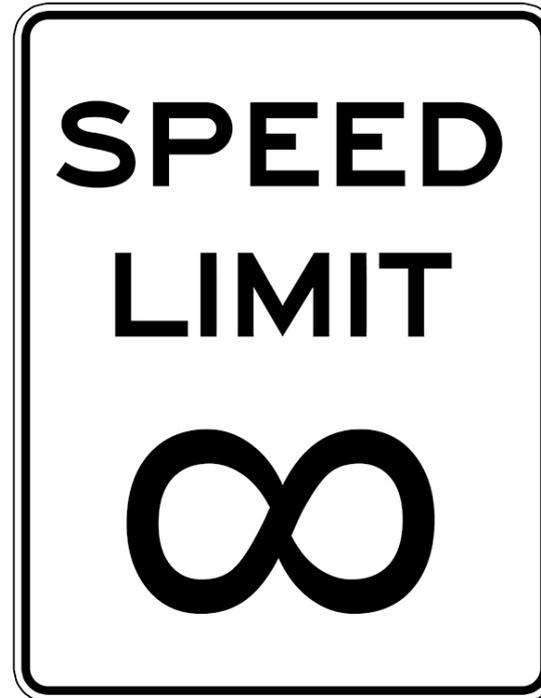
Agile

---

Architecture

---

Autonomous



Automation

---

Abstraction

---

Analytics

---

Ability

---

Accept

From “Embrace Change” (Agile)..... to “Accelerate Change” (DevOps)

# DevOps makes the vision of “Accelerate Change” practical by embedding it concretely in both Development and Operations

## A long-term bond

Common understanding

Aligned incentives (you build it, you run it)

Automate everything

Infrastructure as first class citizen

Full transparency

Quality (in development, operations, NFRs)

“...tear down this wall”

Ronald Reagan

## Dev and Ops



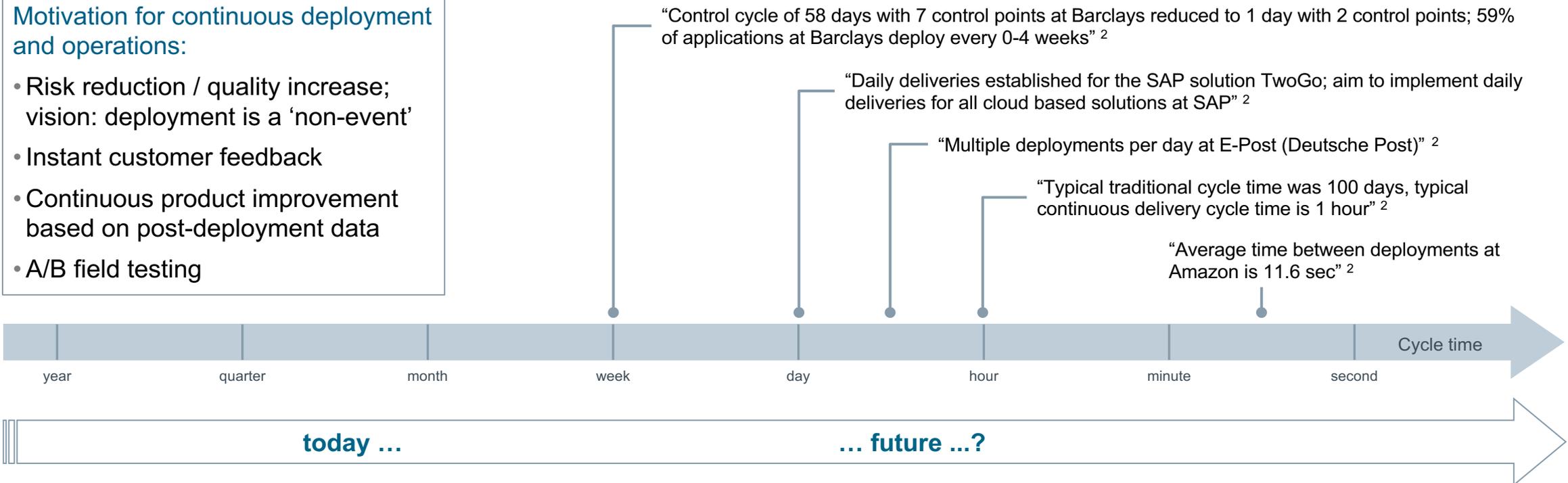
**“Industrial-grade DevOps” for a digitalized industrial world**

# Daily and more frequent enhancements <sup>1)</sup> of the live system are already common practice across many businesses outside industrial environments



Outside view: paradigm shift towards continuous deployment and operations

- Motivation for continuous deployment and operations:**
- Risk reduction / quality increase; vision: deployment is a 'non-event'
  - Instant customer feedback
  - Continuous product improvement based on post-deployment data
  - A/B field testing

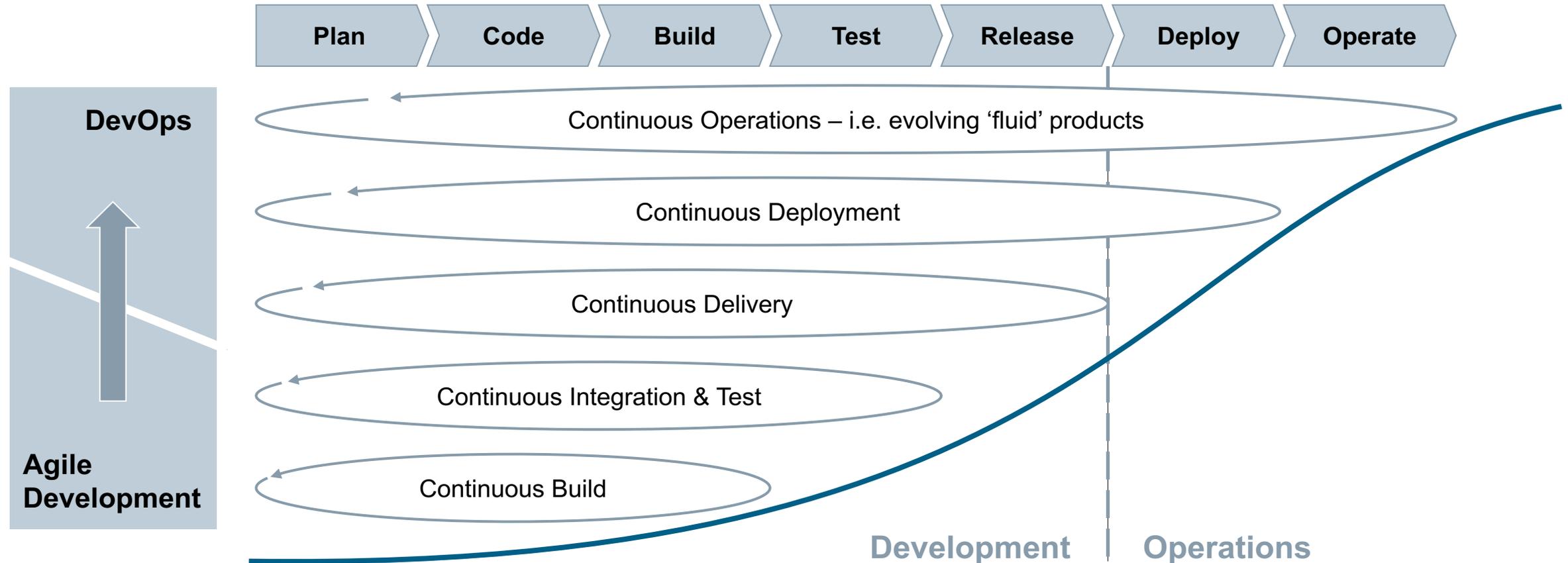


**In the web domain – and increasingly in traditional business as e.g. retail or finance – mastering continuous deployment of enhancements into operations is a key competitive advantage today**

1) Enhancements includes e.g. bug fixes, updates, functional enhancements; 2) Quotes from 'Delivery of Things' conference, Berlin, April 2016

# Fast evolving part is driven by increasing software capabilities, enabling the application of concepts from continuous deployment and DevOps

Scope of continuous activities (e.g. daily / multiple times per day)



Source: Adapted from collab.net; for definitions of the five levels see next slide

Restricted © Siemens AG 2016

# Fast evolving part is driven by increasing software capabilities, enabling the application of concepts from continuous deployment and DevOps

## Definitions and characteristics

- **Continuous Build**

Ability to compile new or changed code whenever desired.

- **Continuous Integration & Test**

Ability to integrate and test new or changed code whenever desired.

- **Continuous Delivery**

Ability to release customer and / or product value to a target environment whenever desired, but one may decide not to for business reasons.

- **Continuous Deployment**

Ability to release and install customer and / or product value into a production environment / customers device whenever desired, but not necessarily activating it in the live system.

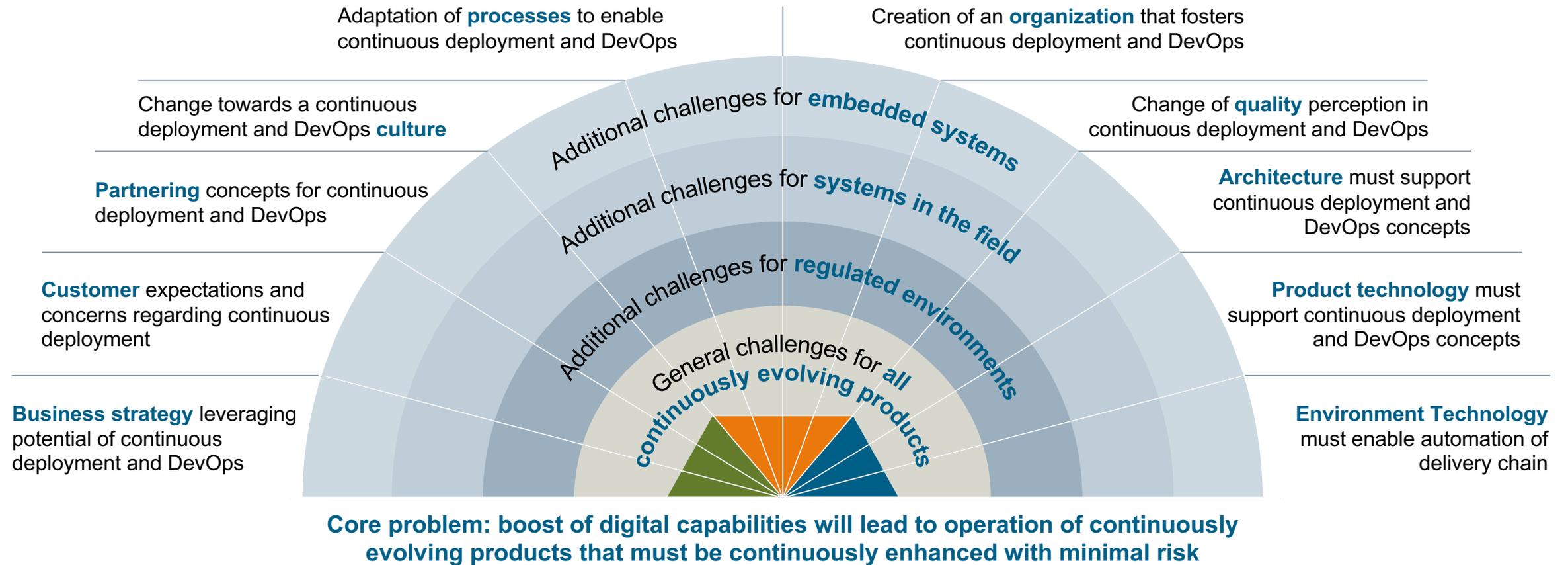
- **Continuous Operations – i.e. evolving ‘fluid’ products**

Ability to update / enhance a running system at customer site with customer and / or product value whenever desired.

# There are significant challenges for continuously evolving products and PLM has to be further innovated to enable application of DevOps concepts



## Challenge categories and 'layers'



Thanks for your attention! Questions?

**Frances Paulisch**

Siemens AG  
CT BE&QM OE  
Otto-Hahn-Ring 6  
80200 Munich  
Germany

E-mail:

[frances.paulisch@siemens.com](mailto:frances.paulisch@siemens.com)

**Internet**

[siemens.com/corporate-technology](http://siemens.com/corporate-technology)

**Intranet**

[intranet.ct.siemens.com](http://intranet.ct.siemens.com)