SIEMENS
Ingenuity for life

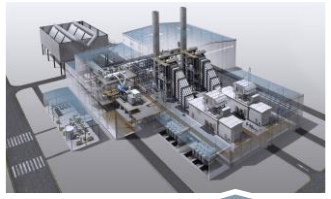
A long way into the future?

DevOps and Industry

DevOps in Industry

DevOps in Industry

Trains



Power plants

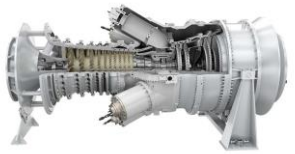
Industrial plants

Buildings

Power distribution

Medical Devices

Drives



Turbines

SIEMENS
Ingenuity for life

Electrification, Automation, Digitalization

DevOps in Industry

Today's update scenario*

- Product update
  - Packaged, approx. once per year
  - Safety functions have separate approval process (like other cross cutting concerns)
  - Products are regression tested with millions of test cases
- Engineering update
  - Product update is deployed in test environment
  - Plant engineering is migrated to updated product
  - Plant is simulated to verify operations
- Plant update
  - Plant is stopped
  - Updates are applied and made operational
  - Plant is restarted

No room for a DevOps vision?

*exemplified to illustrate possible steps to keep quality constraints

DevOps in Industry

iDevOps

Our iDevOps view (“i” stands for industrial-grade)

Product update
- Continuous
- Fast

Engineering update
- automatic

Plant update
- Update in run
- Self testing
- Automatic rollback

And…
- Automatic operations feedback

A long way into the future of iDevOps!

SIEMENS

1. How to update tightly coupled complex systems more frequently?
   - Without losing quality?
   - Without operational hick-ups?

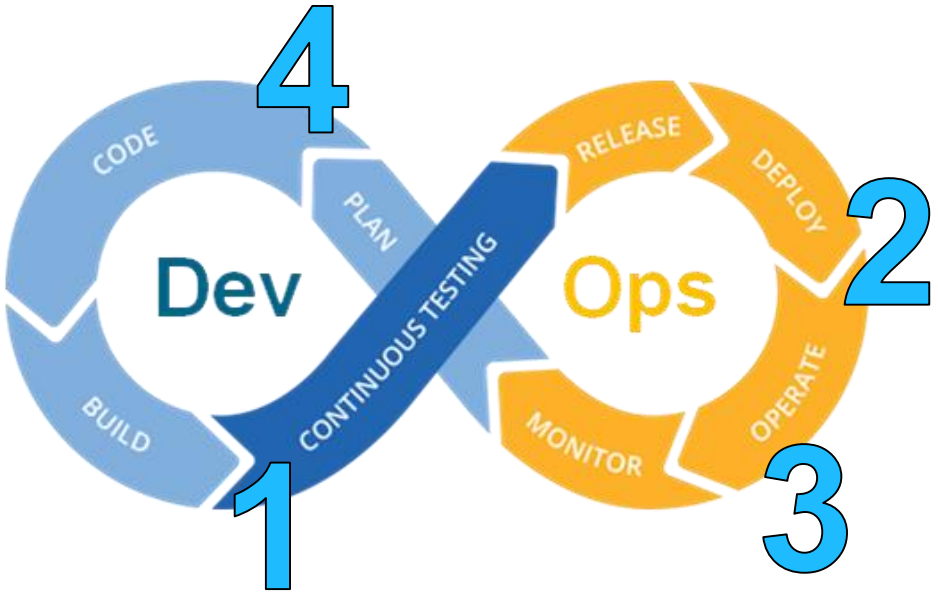
today's focus:
Product Creation

2. How to automatically migrate the engineering data?
   - Without losing functionality?
   - Without losing quality?

3. How to keep an operational plant up and running without quality overhead?

4. How to iteratively frontload the development pipeline?

Siemens research:
iDevOps Program



A long way into the future of iDevOps!
How to update tightly coupled complex systems more frequently?

SIEMENS

iDevOps - Product Creation

→ Strategically change the way we develop our complex systems

→ Main question: how to handle complexity in a scalable way?

Experienced Complexity Drivers

Feature related	Quality related	Product related	Infrastructure related	Organization related
• Frequent changes • Conflicting requirements • Promises • Legacy	• Safety • Security • Usability, Workflow • Installation/ Update	• Interdepenencies within the product • Cross cutting features • Testability, Test coverage • Migration/ Evolution	• Unsuitable task management • Complex building infrastructure • Security constraints • Slow evolution in infrastructure • Hickups in operations	• Technical debt, Feature pressure • Size • Brain drain • Conflicting strategies • Concurrent improvement programs

A long way into the future of iDevOps!
How to update tightly coupled complex systems more frequently?

SIEMENS

Feature related	Quality related	Product related	Infrastructure related	Organization related
• Frequent changes • Conflicting requirements • Promises • Legacy	• Safety • Security • Usability, Workflow • Installation/ Update	• Interdepenencies within the product • Cross cutting features • Testability, Test coverage • Migration/ Evolution	• Unsuitable task management • Complex building infrastructure • Security constraints • Slow evolution in infrastructure • Hickups in operations	• Technical debt, Feature pressure • Size • Brain drain • Conflicting strategies • Concurrent improvement programs

→ changes needed, in all kinds of different topics:

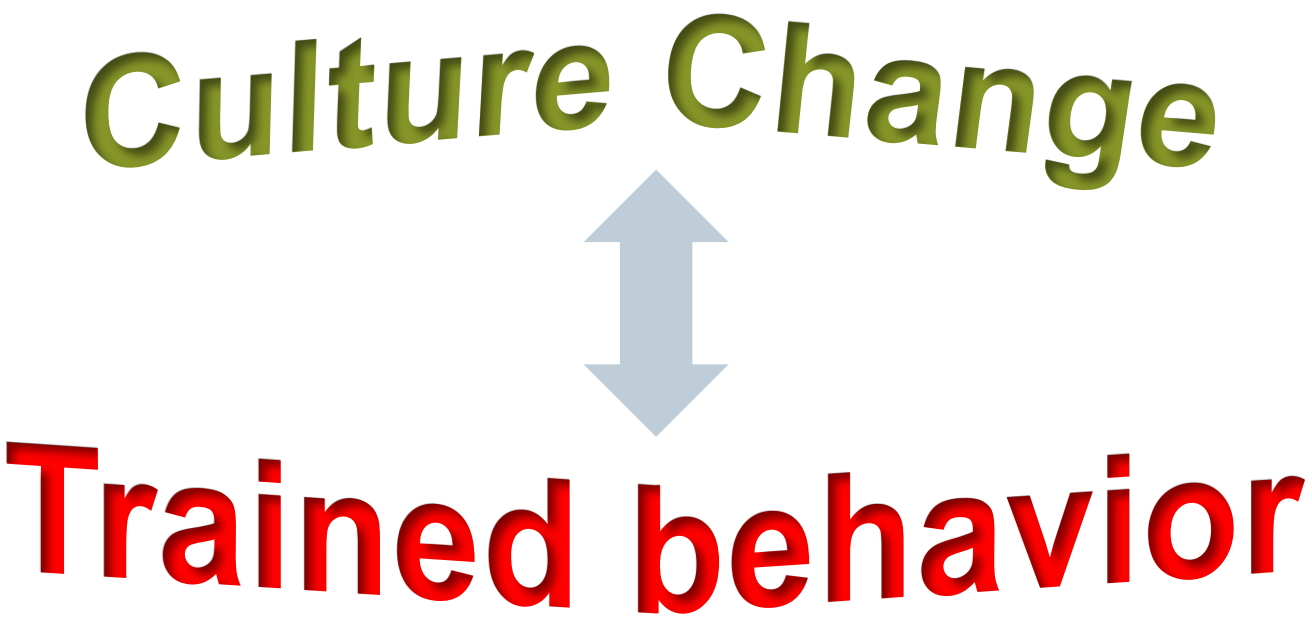
- Process and Organization, to allow for agility
- Product Architecture, to allow for more independent work
- Tools Development, to allow for fit-for-purpose process automation and support

→ Single changes in one area will no do the trick

If nearly all aspects of our PLM business are impacted by our goal, we obviously need to rethink on a global level. Otherwise things would have emerged in a different way.

Consequently we talk about “Culture Change”

Because: many aspects of what we did in the past need to be done differently. Our trained behavior needs to change.



Culture Change

SIEMENS

Questions to understand change:

1. What will be fundamentally different in iDevOps?
2. Where would our learned habits lead us into the wrong direction?
3. Which learned knowledge is to be changed then?

Agile?

Globalization?

Social media?

Digitalization?

Climate change?

Culture Change

SIEMENS

Answers for iDevOps:

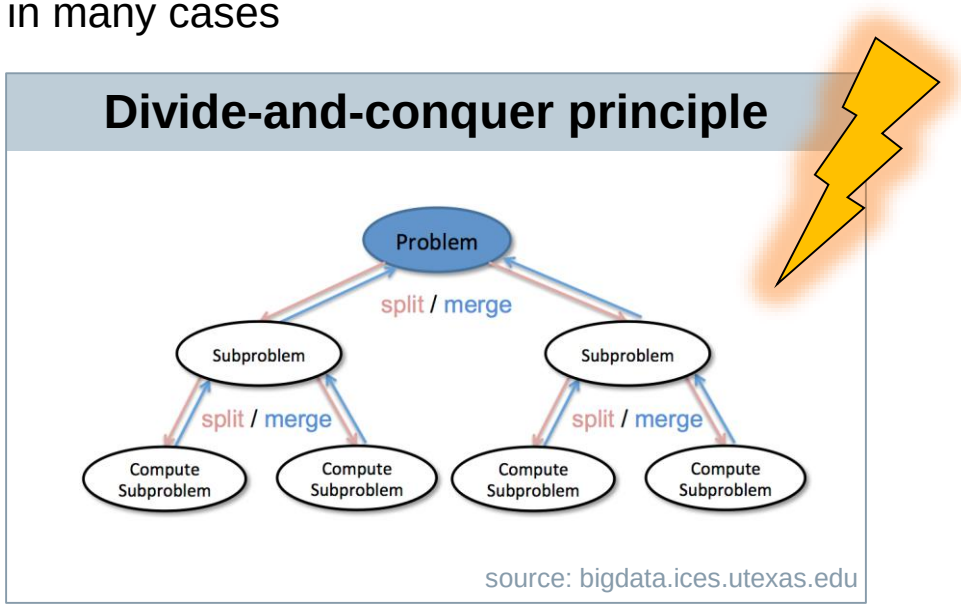
1. What will be fundamentally different in iDevOps?
   - Development and Operations are integrated somehow.
   - Everything seems to be automated.

2. Where would our learned habits lead us into the wrong direction?
   - Silo organizations, optimizing within their silo.
   - Tool development and operations treated separately from products.

3. Which learned knowledge is to be changed then?
   - What makes us work in silos?

SIEMENS

Lemma: Divide-and-conquer is wrong in many cases



- What can be wrong about this?
- I learned this in university!
- Divide et impera is very old and proven!
- What is the better paradigm then?

Culture Change – Divide and Conquer is wrong in many cases
About divide et impera

This principle originates from the Romans in order to control the empire.

The concept

- split up the empire
- give the parties contradictory and competitive goals
- parties may only have contracts with Rome

The effect

- The parties would rather fight against each other than fight against Rome
- Rome's position and power is strengthened

source: truthearth.org

This principle is designed to control people, not to maximize output or efficiency!
Divide-and-conquer should be removed from our solution portfolio. Split/Merge is OK.

Culture Change – Divide and Conquer is wrong in many cases
Paradigm shift – Starting Point

Drive Optimization by Divide and Conquer

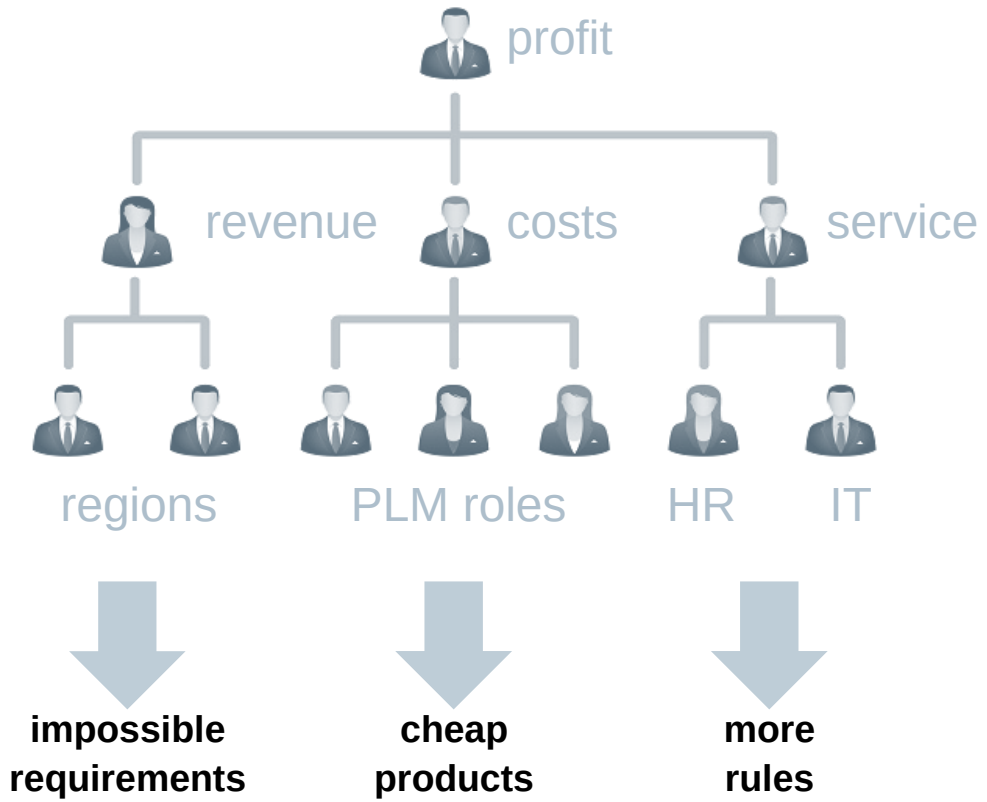
- Complexity managed with classical hierarchical divide-and-conquer command and control structures
- The overhead costs to drive these control structures pay off with optimizations provided
- Today's hierarchies follow optimization rules, like synergies or value stream optimization

Examples

- Synergy by pooling: Prod.Mgmt // R&D // Testing
- Flexibility by splitting: Prod 1 // Prod 2 // Prod 3

Problems

- If the optimization focus changes, we need to re-organize
- For more complex settings, a one dimensional hierarchical simplification no longer provides value
- Overhead costs (multiple managers, alignment meetings, controlling, reporting) do no longer pay off
- De facto we experience, several orthogonal hierarchical management structures co-exist (sometimes represented in powerless cross cutting functions)
- Silos, Silos, Silos



Culture Change – Divide and Conquer is wrong in many cases
Paradigm shift – Some thoughts on new paradigm needed

Enable quick adaption in a scaling organization

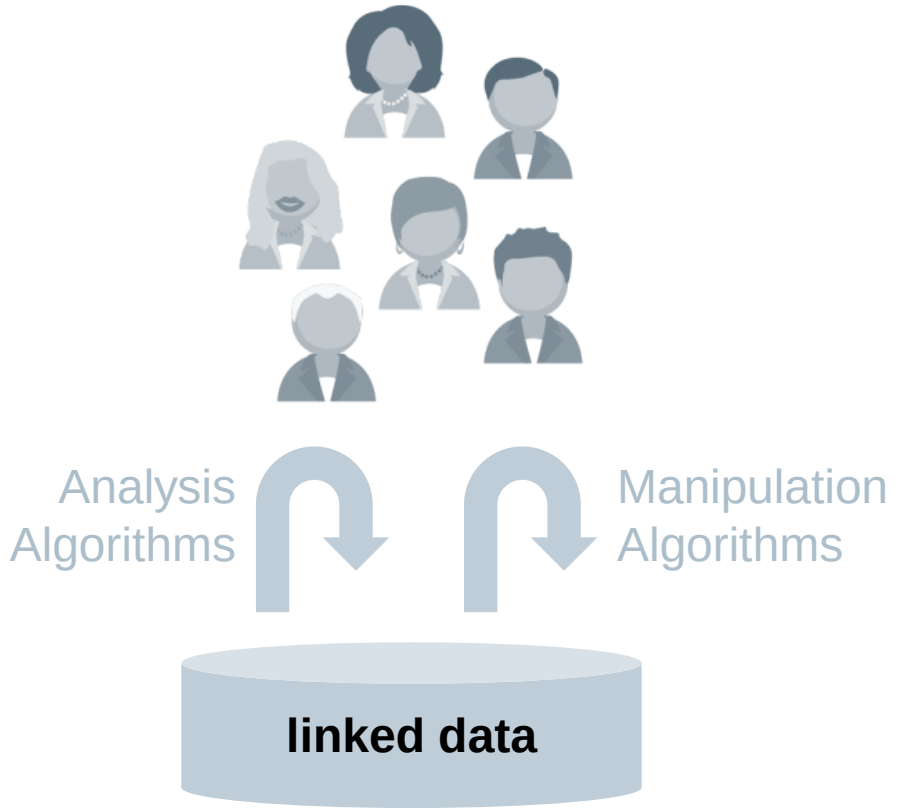
- No leading organizational hierarchy
- Multiple integration hierarchies co-exist
- Results are linked into each integration dimension
- Stakeholders are responsible to keep control within one dimensional hierarchy
- Dependencies are taken to manage changes
- Tools support stakeholders to keep control

Dimension examples

- Epics, User Stories
- Services, Components
- Qualities
- Value: Requirement, Concept, Estimation, Development, …

Problems

- Redefinition of responsibility needed
- How to manage a changing multidimensional hierarchical system?



SIEMENS

Lemma: Culture change also is a technical problem

- But we are good at engineering!
- Only our cultural and inter-human working behavior is error prone!
- Technology can do anything!

Culture Change – Culture change also is a technical problem
Research frame

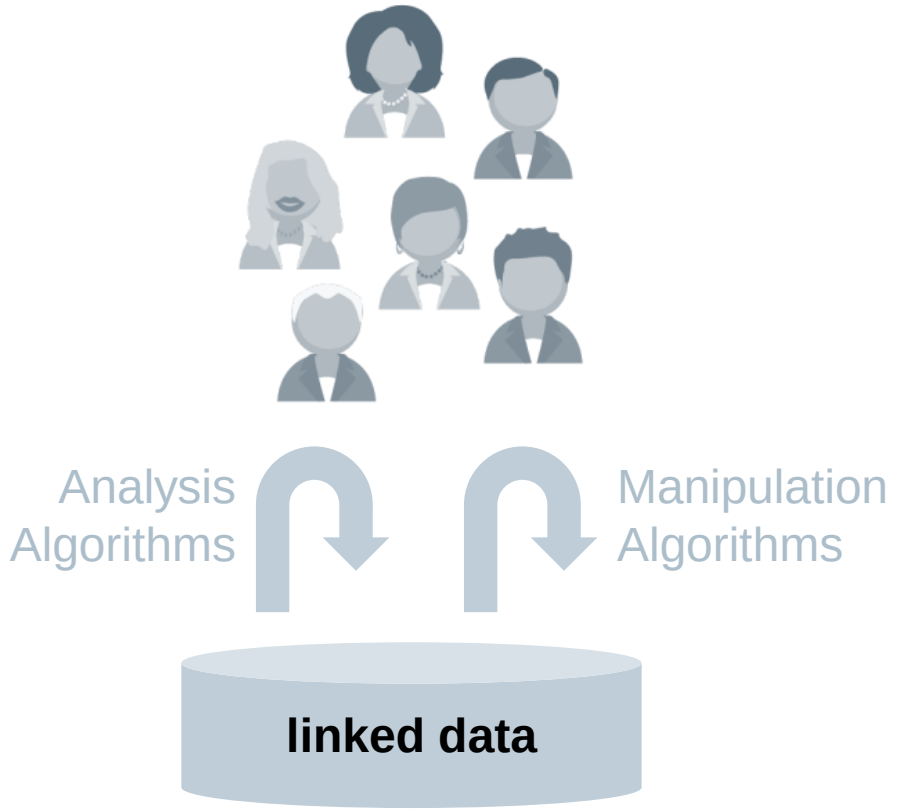
Technical Challenges

- Support data to be managed in different dimensions
- Link these data items into other systems
- Support change in the data
- Support change in the dimensions
- Create algorithms to analyze data
- Create algorithms to manipulate data
- Organize daily work of stakeholders
- Constantly be able to adapt priorities
- Automatically do reporting in different dimensions

Without such technology …

… we will lose control, and

… we will fall back into simplification by divide and conquer



Now…

… that I talked about **culture** and that is also is a **technology** topic …

… I come now to some …

… **technology**, we look into

Technological aspects in iDevOps
Architecture

SIEMENS

Some of the more technical trends, we invest in

Product architectures will move towards

- Microservices
- Cloud technologies
- Multiple deployment setups

Testing strategies might be innovated towards

- Self testing
- Concurrent testing
- Virtualized environments

Technological aspects in iDevOps
Tools/ Automation

SIEMENS

Changes in the way we deal with tools

- Tools behave like the products
- Tools need real ownership in the organization
- The tool landscape needs well done architecting

Tools massively automate many steps

- IDE, Workspace
- Building
- Deployment
- Testing
- …

We need to even more focus on the few humans left in the equation

Recursive mindset

- Tools are built with the tools

Thank you

SIEMENS

- DevOps is one major trend also in industrial applications
- DevOps marks many of our visions into the Digitalization age
- DevOps needs special attention to also ensure industrial fitness

- We need to change …
  - how we manage
  - how we change
- We need to invest into research and technology …
  - to be able to manage multidimensional hierarchical structures
  - to be able to manage changes
  - and to do all the rest of DevOps

- Lets get in touch! stefan.horn@siemens.com