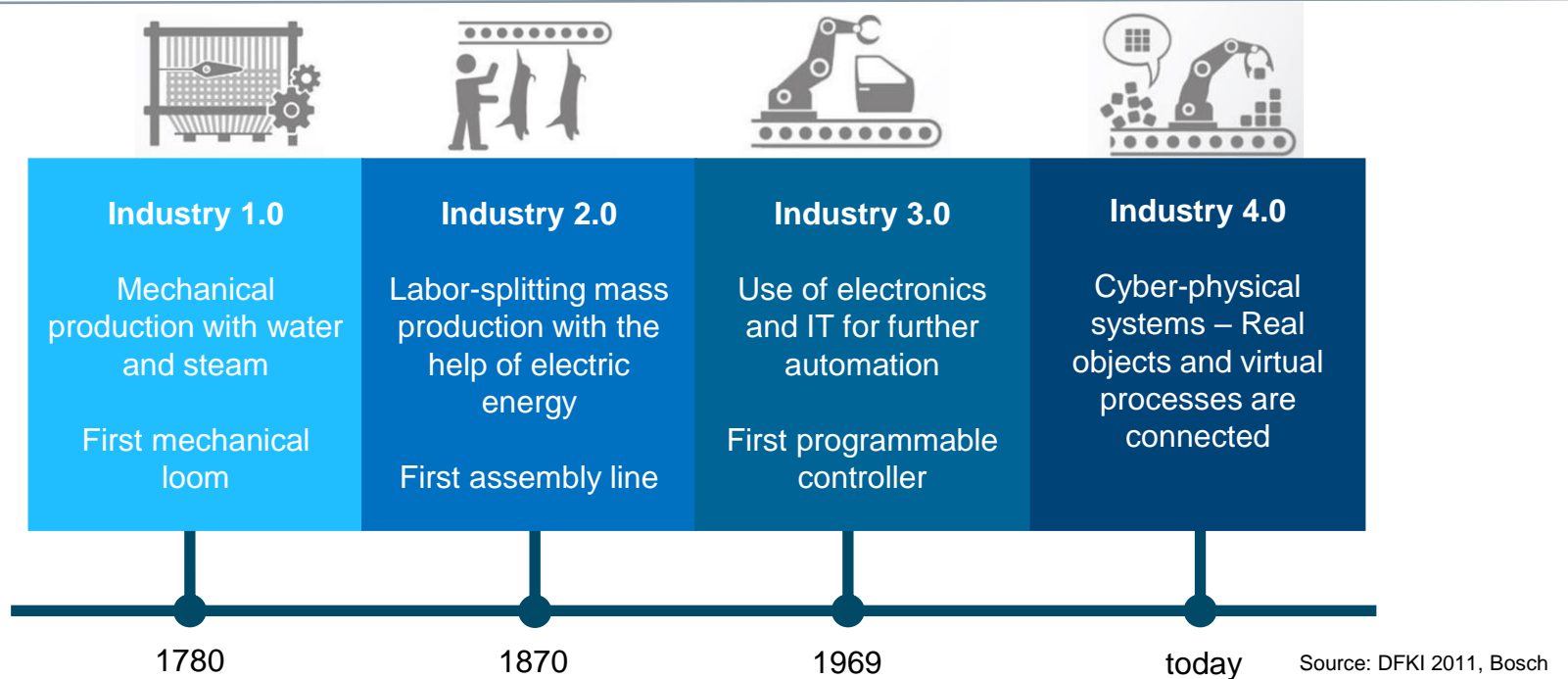

Management and Leadership in a Digital Transformation

About Me

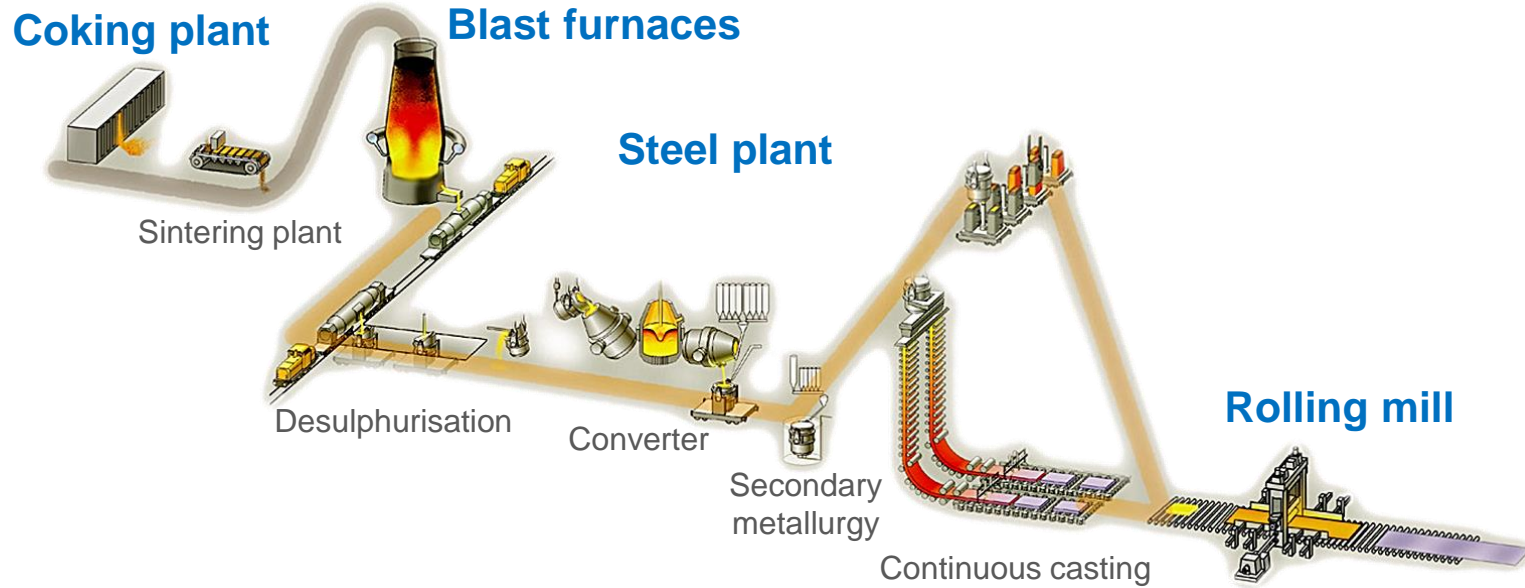
- Michael Replinger
 - Master in computer science
 - Over 20 years experience in developing distributed systems
 - Over 16 years experience in leading and developing teams
- Hobbies
 - My family, biking, new technologies
- Disclaimer
 - Content is inspired by a lot of conferences, papers, books (that then my teams explained to me)
 - It represents my personal view and no general purpose instruction
 - I hope this talks helps you to get some new ideas or new viewpoints

Industrial Age



- Technical progress was starting point for significant economic and social changes
- Management strongly influenced by Taylorism
 - Knowing exactly what you want men to do, and then seeing that they do it in the best and cheapest way
 - Hierarchical management ensures productivity by stable processes

Integrated Steel Production Site at Dillinger



- Run and optimized steel production for ~338 years (founded in 1685)
- Lived through all industrial revolutions

Digital Information Age

- Switch from productivity to individual customer experience
- From retail business to facilitate marketplaces
 - Digital platforms such as alibaba, fundrise, and diem disrupt retail market
 - Affected business: Banks, retail companies, traditional service providers, ...
- From ownership to access
 - Music and video streaming, computer games, ...
 - Candidates: Products whose life cycle exceeds the personal usage period, such as cars, houses, clothes, ...
- From product to service
 - Instead of selling a product, sell what it does
 - From transportation vehicle to tons/km
 - From insulin to good live with diabetes

➔ *“If companies are not digital of fast becoming digital they will not survive.” (Mrs. Vestager, EU commissioner for A Europe Fit for the Digital Age)*

Challenge for Industry

- Companies and management needs to become ambidexter
 - Manager to tightly manage existing business
 - Leader to enable innovative thinking for constant reinvention of the business
- Management vs. Leadership according to Warren Bennis

Manager	Leader
Manages and maintains	Renews and develops
Is classic good soldier	Is completely himself
Relies on control	Inspires confidence
Focuses on systems and structures	Focuses on people
Asks "how?" and "when?"	Asks "what?" and "why?"
Accepts the status quo	Challenges the status quo

➔ *"In the new world, it is not the big fish which eats the small fish, it's the fast fish which eats the slow fish."* (Klaus Schwab)

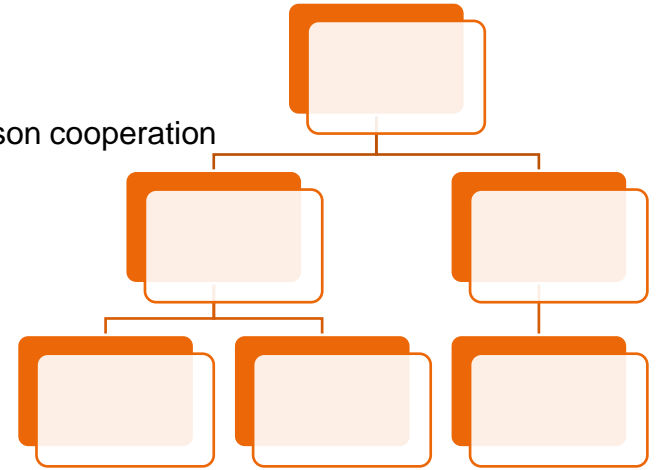
Typical mistakes in a digital transformation

- See a “transformation” as one time “change”
 - Change has a beginning and an end
 - A transformation is a process that, once initiated, cannot be stopped
- The IT department is responsible for digitalization and digital transformation
- Introduce new frameworks
 - Become agile by introducing Scrum, Kanban, or SAFE®
- Reduce the burden on or speed up your IT development department
 - Introduce LowCode or NoCode, standardize development tools
 - Functional IT organization to improve efficiency of single teams
- Underestimate culture change on management and operational level
 - Handing over and acceptance of responsibility and power

→ *“We cannot solve our problems with the same thinking we used when we created them.” (Albert Einstein)*

Major Problem: “Org Chart Thinking”

- Organizational structures in reality (Niels Pfläging)
 - Formal structure – static org chart that facilitates compliance
 - Informal structure – “realm of influence” between individuals
 - Value creation structure – how work gets done based on inter-team/person cooperation
- Reorganization is typically done on a static org-chart view
 - Do not represent real communication structure
 - Allow only local optimization and might help dedicated teams
 - Do not improve *overall* delivery of value to customers



- ➔ “*Organization should be viewed as complex and adaptive organism rather than mechanics and linear systems*” (Naomi Stanford)
- ➔ To start a transformation focus on informal and value creation structures and make them explicit

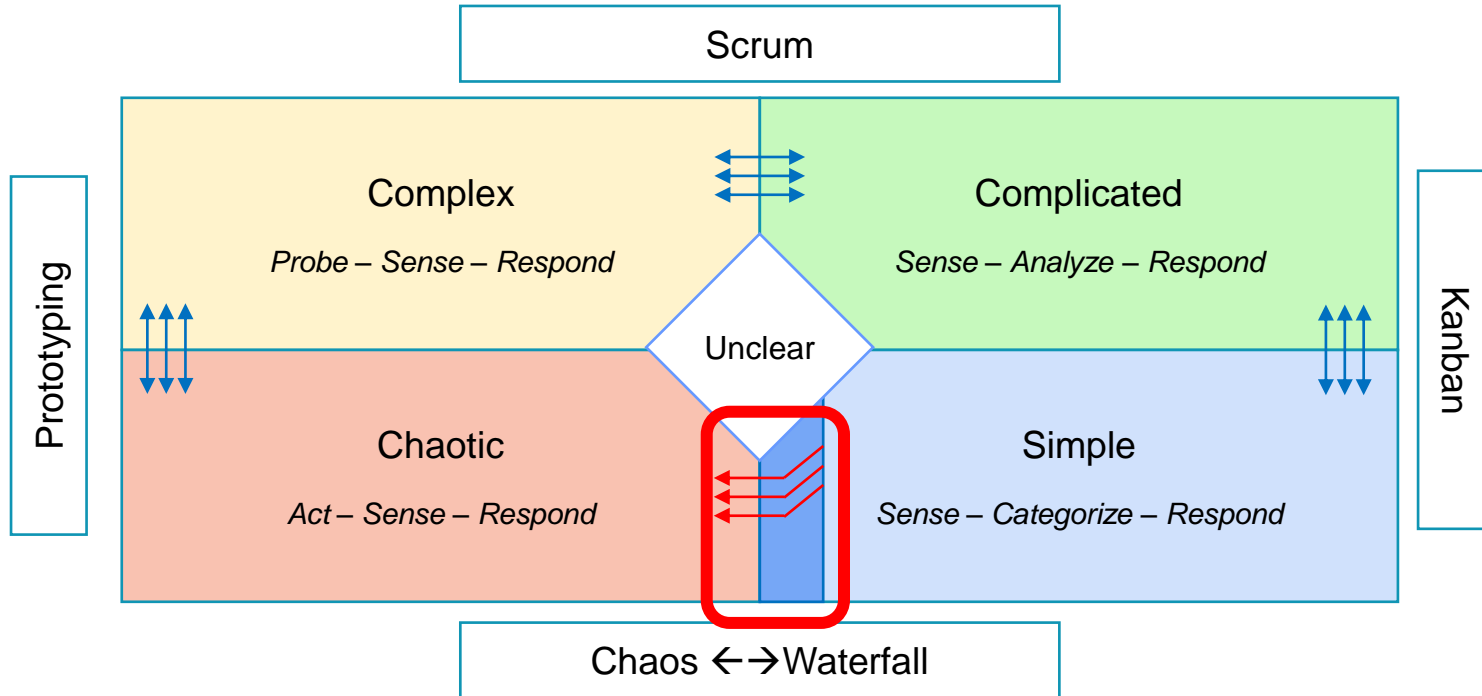
Some Old Laws from IT ...

- Lehman's laws (1974):
 - *“Continuing change: A software system must be continually adapted or it becomes progressively less satisfactory”*
 - *“Increasing Complexity: As a software system evolves, its complexity increases unless work is done to maintain or reduce it”*
- Conway's Law (1967):
 - *“Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.”*
 - The way people communicate closely aligns with the way your code communicates.
 - Business codebases tend to be tightly coupled (especially compared to open-source projects)
 - Inverse Conway Maneuver: Use Conway's Law to structure an organization that fits in the desired system architecture

→ Consequences

- Design systems that can be constantly extended and optimized in an efficient way
- Design organizational structures to reduce complexity and consider emerging technologies
- Organization and team design requires technical expertise
- Consequences belong to different problem domains

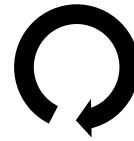
Cynefin-Framework – A Sense Making Framework



- Allows moving between different problem domains
- Simple → Chaotic: Apparent clarity of the simple domain tempts to overlook or ignore emerging changes!

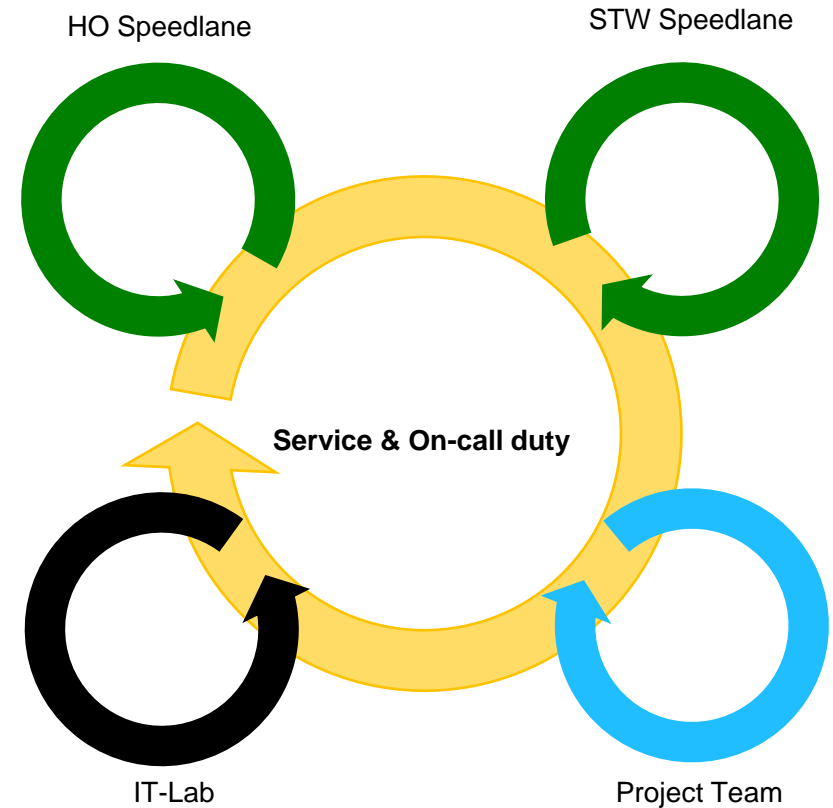
Separate Teams by Problem Domain

- Teams can understand why to use different approaches
- Speedlane
 - Problem Domain: “Complicated”
 - Expert team responsible for optimizations and minor changes
 - Responsible for a specific business domain
- Project Team
 - Problem Domain: “Complex”
 - Cross-functional team
 - Responsible for designing new systems with domain experts
- IT-Lab
 - Problem Domain: “Chaotic”
 - Students and Trainees
 - Evaluate, present, and train in evolving technologies
- Service Team
 - Problem Domain: “Simple/Complicated”
 - Contact for supported business domains for service and on-call duty
 - Rotating members of the department



TI-HO/STW – Derive Department Structure

- Establish a service team
 - Make daily requests and problems from customer visible
- Add a Speedlane for each business domain
 - Measure influence on service requests
 - Supporting service helps to see cross-domain problems
- Start building a project team
 - Solve new complex problems in business domain
 - Reduce complexity by migrating legacy systems
- IT-Lab
 - Support service to “feel” problems (aka “best practices”)
 - Evaluate new approaches how to do things differently
 - Communicate it to other teams



Results (Personal View)

- Starting such a transformation needs time (~4 years)
- The quality of collaboration within the department increased significantly
 - Sub-teams understand why they use different approaches
 - It gives trust to the entire department
 - Impacts caused by pandemic crises could be compensated easily
- Internal weekly meeting (Friday wins) to present added customer values
 - Makes customer needs transparent to all members
 - Also used for reporting to IT-management and customers
- The constant changes by IT-lab caused a significant technology boost
- Lessons Learned
 - Lone wolfs: Either you find some niche or they should change the department
 - Hierarchy: Even such a small transformation can hardly be stopped without damage → Be sure management really understands consequences

Summary

- Digital Information Age typically requires a transformation
 - Switch from Org-Chart thinking to a system and domain thinking
 - Requires top-down and bottom up support

- IT industry already run into similar problems and provides methods to start a transformation
 - Consider Conway's Law and Lehmann's Law, agile manifesto, ...
 - Ensure Craftsmanship and continuous learning

- Cynefin framework is essential for decision-making about management strategies

- Be aware of silver bullets and cargo cult
 - If a team reports everything is OK, then everything is in trouble

- Recommended books
 - *Slack: Getting Past Burnout, Busywork, and the Myth of Total Efficiency*, Tom DeMarco
 - *Team Topologies*, Matthew Skelton and Manuel Pais
 - *Elastic Leadership: Growing self-organizing teams*, Roy Osherove

Questions

Die große Veränderung planen ...

$$(1,00)^{365} = 1,00$$

$$(1,01)^{365} = 37,78$$

... oder jeden Tag für eine kleine
Verbesserung sorgen?

