

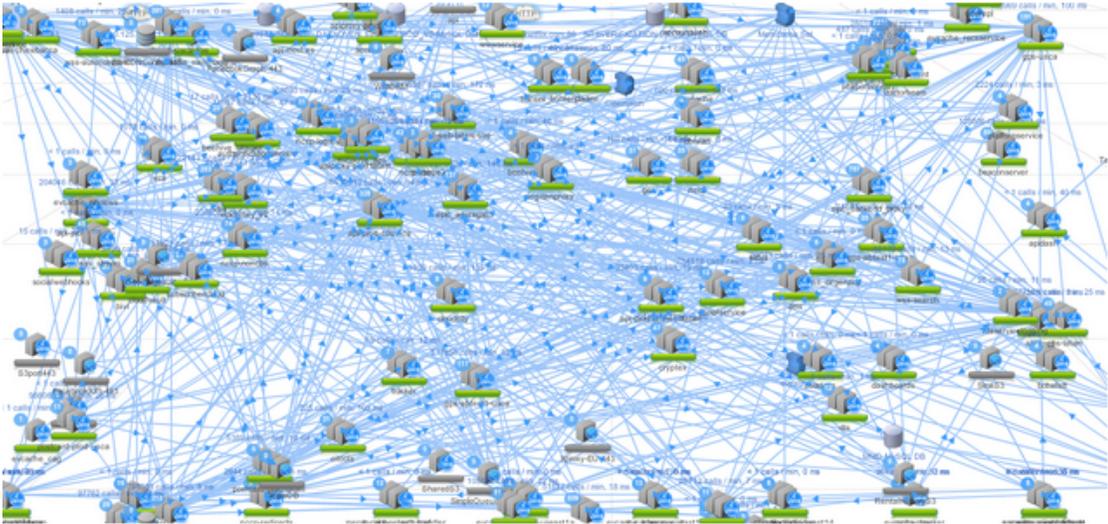
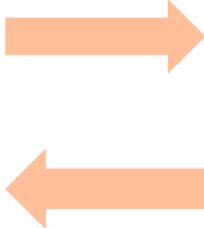
# The last agile mile

Amsterdam

Henk Kolk, Chief Engineer, ING

May 12<sup>th</sup>, 2017

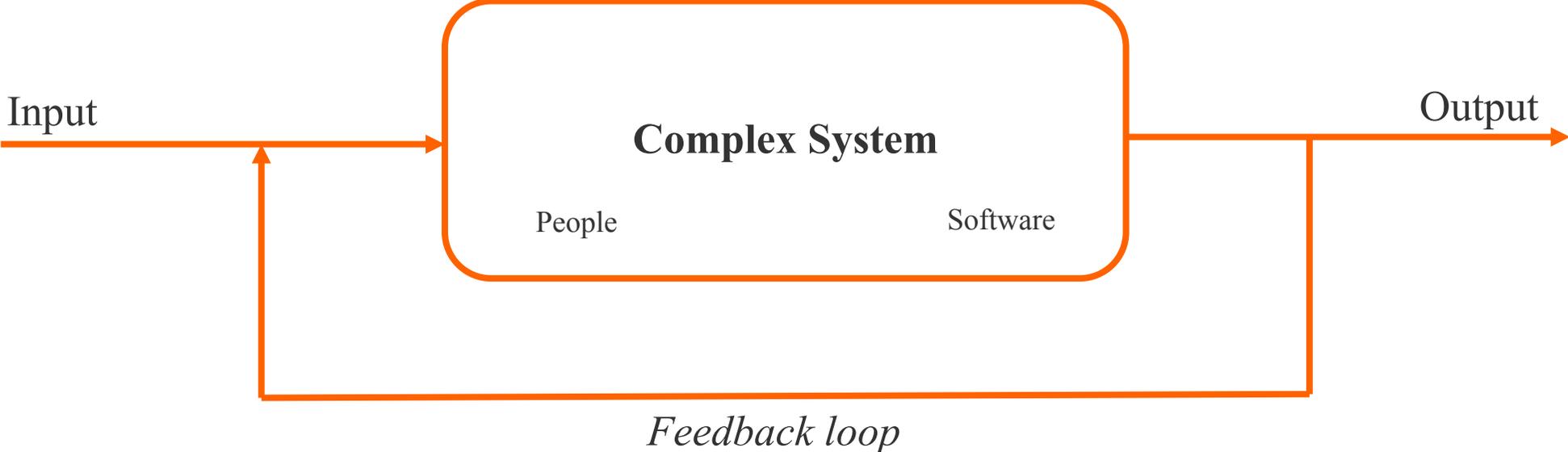
# Engineering takes place in the context of changes being made to a complex, interrelated, existing system of people and software



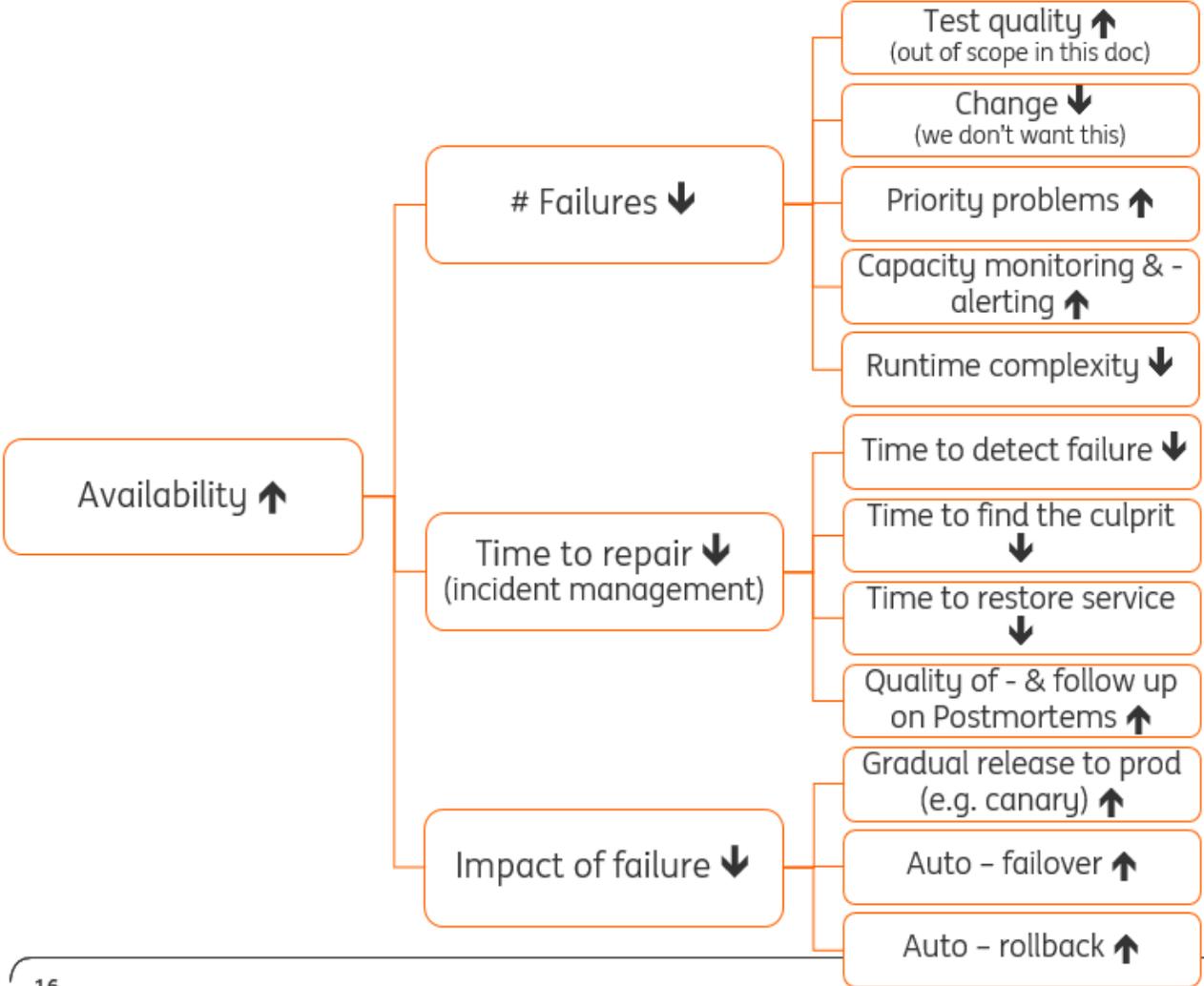
**We make changes to achieve better outcomes for our customers, society, employees and shareholders. Objectives and Key Results are outcomes defined, by ING Product Owners\*)**



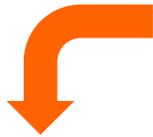
**Feedback loops are at the heart of “Agile”, “Lean Start-up”, “DevOps” and Continuous Delivery as the means to control changes to a Complex (non-linear, unpredictable) System.**



# Feedback loops need to be designed. For example, if we want to reach an availability of 99,9% for our customers, how do we achieve that particular outcome?



# Agility<sup>\*)</sup> is ...



- **Assess** where you are

- **Take a small step**

- **Evaluate**

- **Repeat**



\*) As intended in the Manifesto for Agile Software Development, 1999  
According to “Pragmatic” Dave Thomas (“Agile is Dead” - Goto talk)

# Agile Software Development is recycling some pretty solid (old) ideas

- **Scientific Method**

Francis Bacon (1620)

- **Plan – Do – Study – Act loop of Total Quality Management**

Shewart (1939) and W. Deming (1950)

- **The New New Product Development Game**

Hiroataka Takeuchi and Ikujiro Nonaka (1986)

# Definition of Done (test): the current manual documents and tools are prevented, or replaced with Robots and Feedback Loops.

## Current tools and Deliverables

- BIA
- ITRA
- SA- Chapter 6
- OSG-A
- OSG-I
- Authorization Matrix
- SOLL-IST
- TSM
- Internal RCECs
- Provisioning VMs
- iRisk
- ...



## Tools and Deliverables

- Guiding Principles
- Control Framework
- Assembly Robot & Module Warehouse
- Audit Robot
- Release Robot
- Masking Robot
- Integration with Private Cloud IaaS
- Integration with ServiceNow (autofilled CMDB)

## Resulting in

- Globally standardized way of working
- Time to market across countries and datacenters of mere hours

# Outcomes over Impositions

Jeff Sussna, Designing Delivery

What is the **problem** that we're trying to solve?

# Software is eating the world

Marc Andreessen

**Speed is market share**

Adrian Cockroft

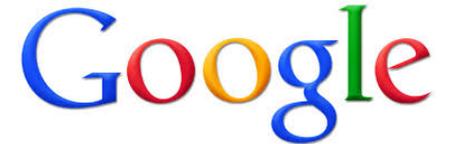
# Platforms eat Pipelines

Ron Kersic, ING

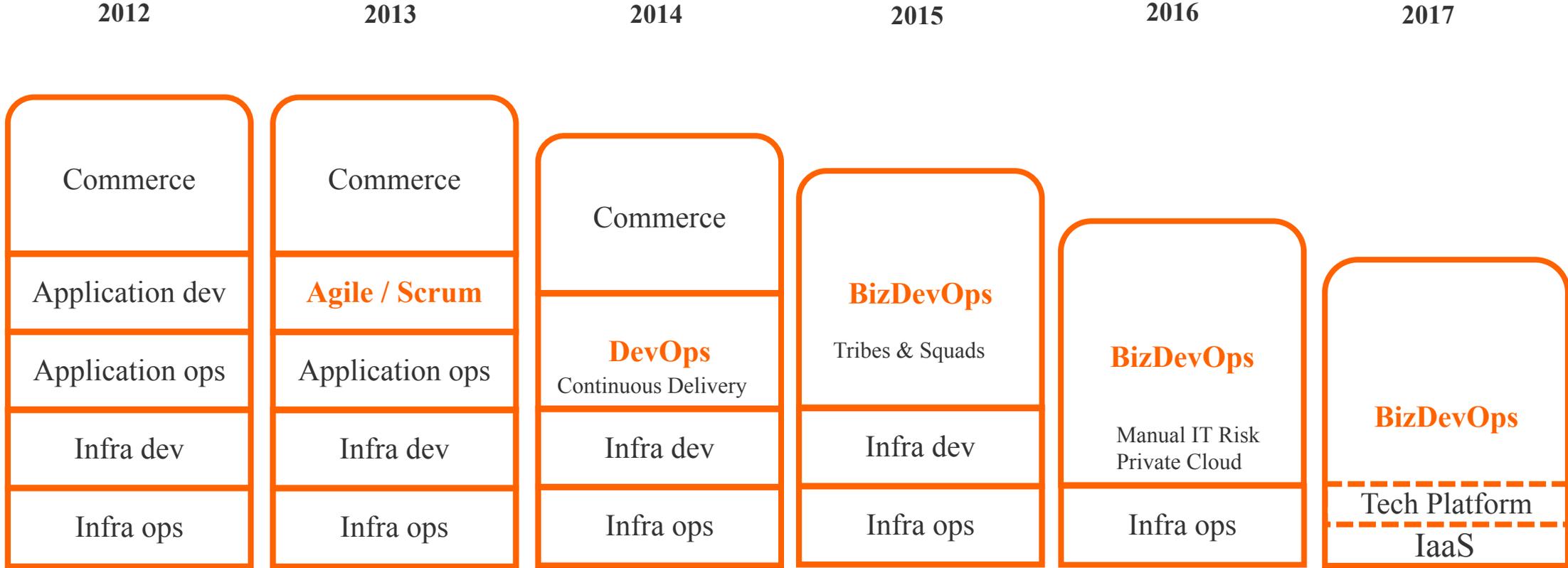
**The platform business model underlies the success of many of today's biggest, fastest growing and most disruptive companies. From Google, Amazon and Microsoft to Uber, Airbnb and eBay.**

## Platforms ...

- **Serve an ecosystem of external producers and consumers**
- **Unlock new sources of value creation and supply**
- **Eliminate gatekeepers to scale efficiently**
- **Run on a not-even-mine inventory**
- **Create community feedback loops**
- **Are designed for global scalability**



# In the past five years, ING has been reorganizing for speed and skill. Roles and responsibilities have shifted radically



**Engineer:** From single discipline to **full stack engineers:** designing, coding, test engineering, infra engineering, etc  
**Product Owner:** From writing PIDs to product vision and backlog to **end to end bizdevops responsibility**  
**IT Manager:** from delivery manager to perhaps the most differentiating role: **skill and competency coach.**

Just like Max Verstappen, we can only achieve high velocities if we are in control



# Technology has evolved to improve control at higher speeds

1990 Steering Wheel  
(McLaren)



2010 Steering Wheel  
(Mercedes)



We must start by leveraging today's technology, language, way of thinking and today's technology best practices for IT Control to assure ourselves and our auditors that we are in control



**“BizDevSecRiskOps”**

- Control Framework shifts left
- IT Risk is controlled 95% by design

**Envisioned outcome**

- CIRM / CORM set the conditions for teams to be in control
- Control is auditable, globally

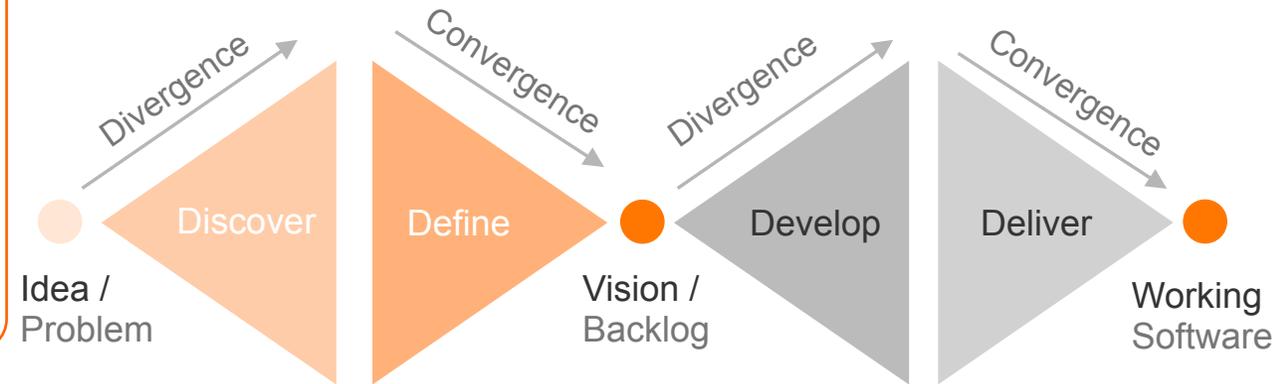
# Shift-Left

John Sharratt, Jon Lee, ING

# Shift Left

## 1. Design Thinking (solve the right problem)

- Discover & define the right problem
- Define the right vision
- Design the right solution
- Test your vision as early as possible
- Test your design as early as possible



## 2. Hypothesis Driven Development (Francis Bacon)

- Every “requirement” is just a hypothesis
- Every hypothesis needs to be proven by data

## 3. Build quality in (W. Deming)

- Problem prevention over detection,
- Begin testing earlier than ever before



**Shift left to the Max**

**The more we shift the problem to the left, the cheaper it is to solve the problem and the faster we go**

# Humans vs Robots

Isaac Azimov

# Immutable Servers

Martin Fowler

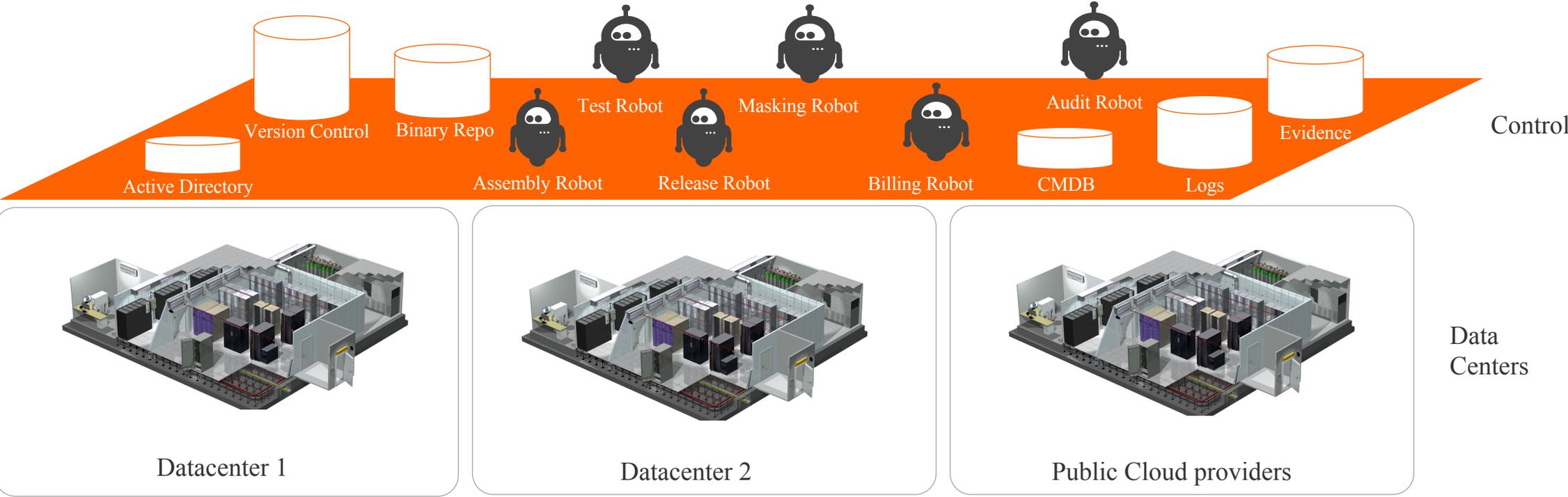
# Cattle and Pets

John Sharratt, Jon Lee

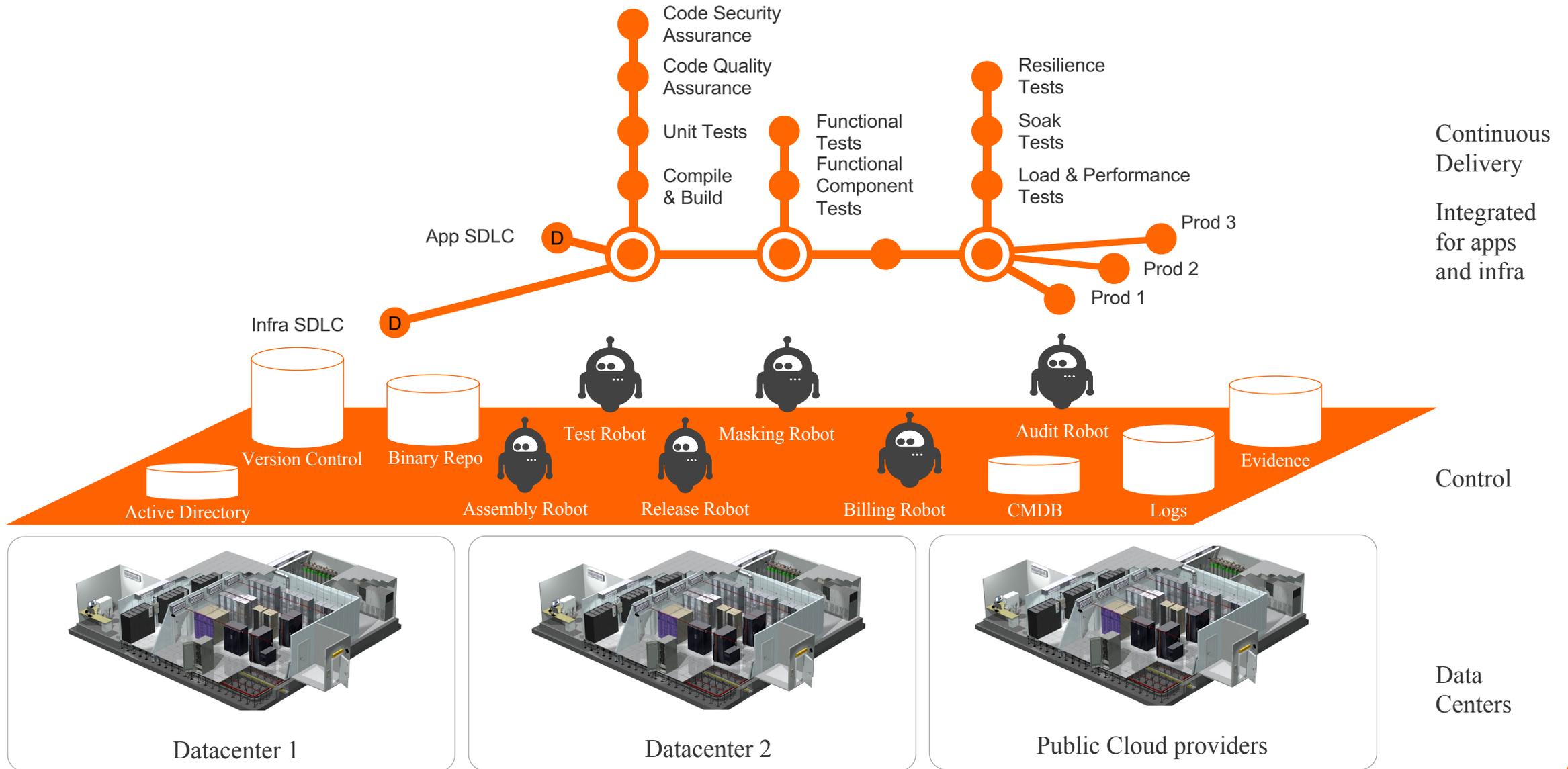
**Infrastructure = code**

John Sharratt, Jon Lee

# Robots (=software) are taking center stage in release engineering (of software)



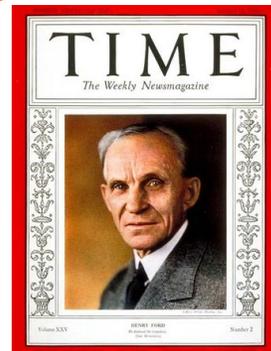
# Humans leverage automated pipelines to deliver software



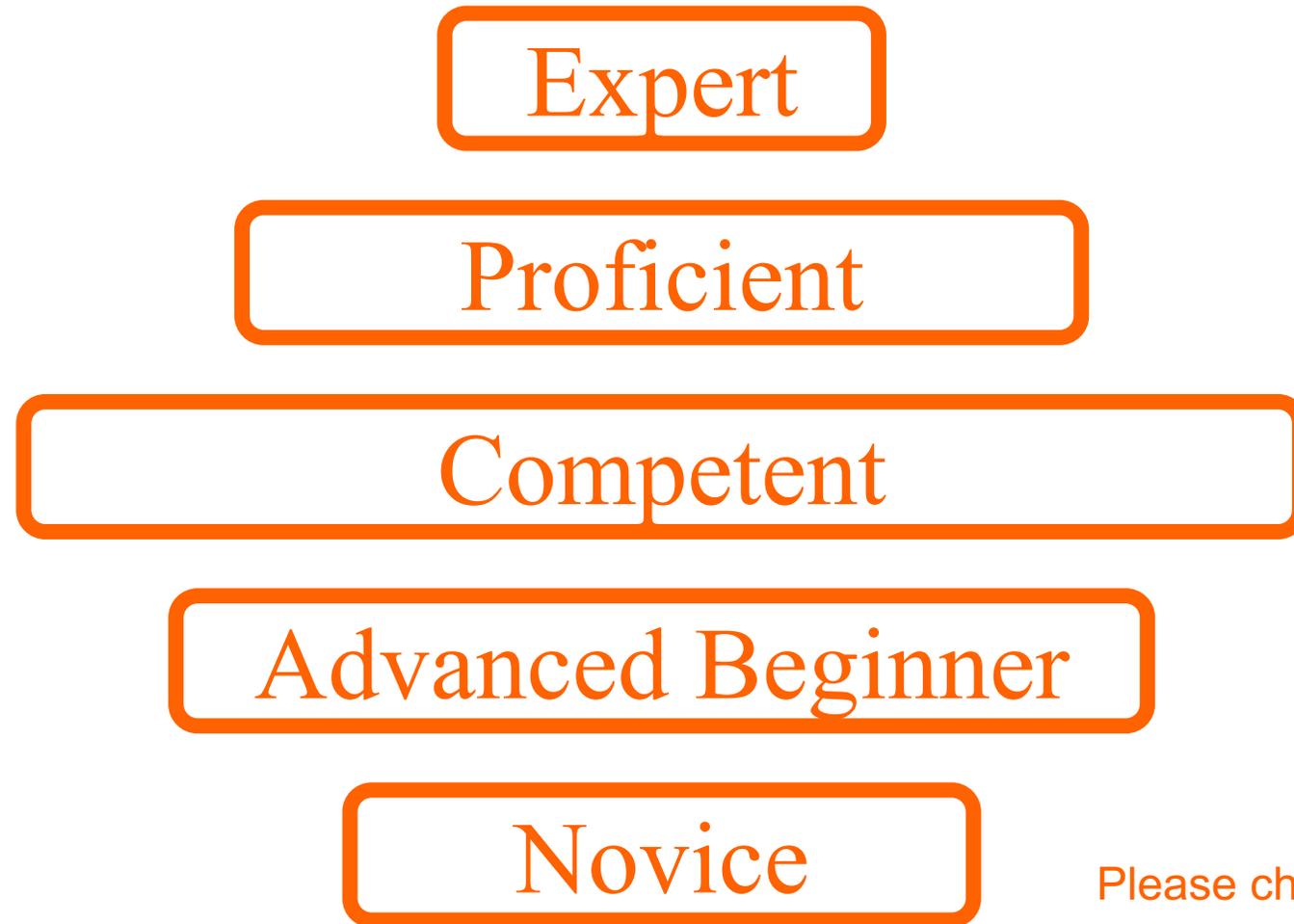
# Learning organization

# Weird assumptions about the roles of software engineers

- Business Architect
- Solution Architect
- Requirements Specifier
- Designer
- Coder
- Tester
- Deployer
- Chief Engineer 😊

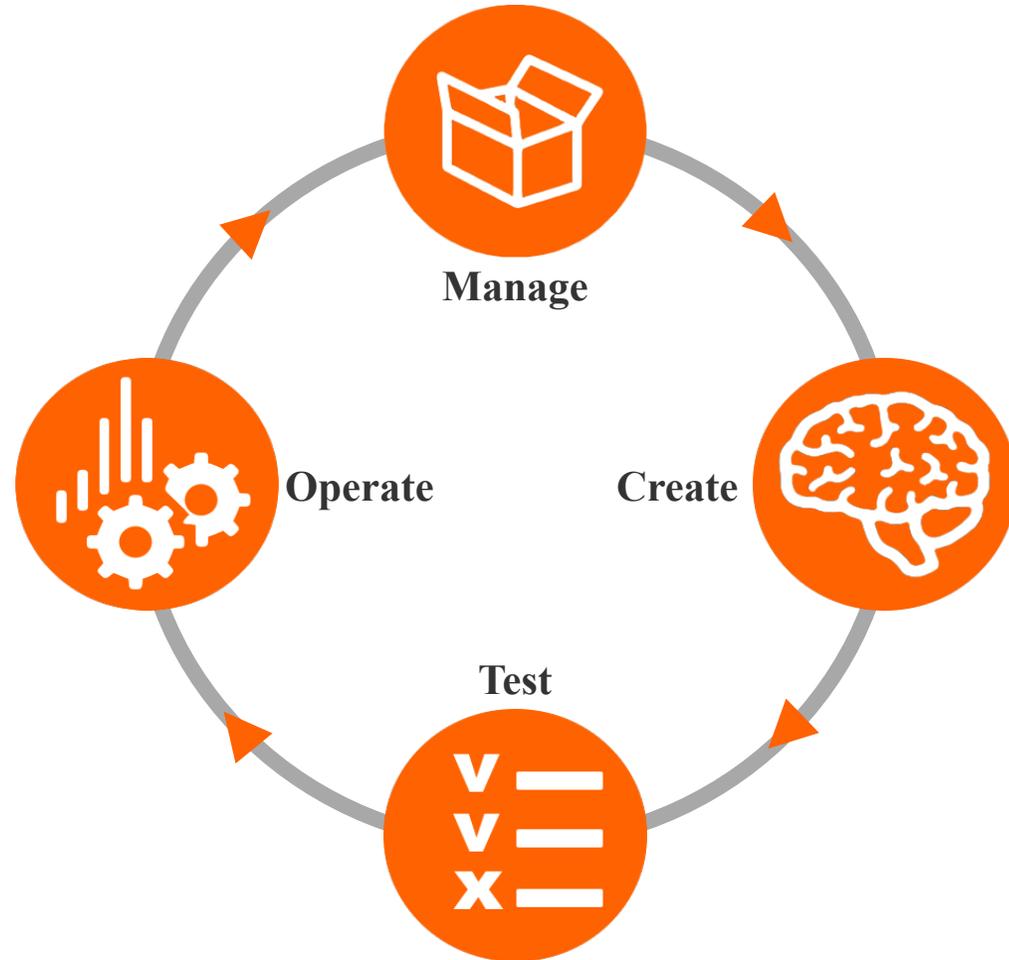


# ING radically changed its HR strategy to allow for the skill growth and creative liberation of multi skilled engineers



Andy Hunt  
Please check out his GROWS method

# INGs Way of Working is supported by an “Technology Platform”, that is designed as a giant Plan-Do-Study-Act (PDSA) cycle, adapted for agile software product management at scale.



**Manage (Plan)** - Digital product management. e.g. portfolio management, key objectives and results (OKRs), backlog management, IT service management, virtual Obeya rooms (feedback on strategic results)

**Create (Do)** - design, share (knowledge management), engineering laptops, continuous integration (version control, build services, binary repositories), release engineering (assembly - and deployment robots)

**Test (Study)** – test engineering: control framework, feedback loops, test environments, integrated test environments, test data, data masking, unit testing, functional testing, integration testing, code quality testing, code security testing, pen testing, load testing, performance testing, resilience testing, test robot, audit robot, evidence repository

**Operate (Act)** - Monitoring, Alerting, Paging, Master Control Room, ChatOps, Run-time Immune Systems, Site Reliability Engineering

**In the words of the great W. Edwards Deming**

**“Learning is not compulsory... neither is survival.”**

Be(come)  
**AWESOME**  
everybody

Twitter: [@henkkolk](#)

E-mail: [henk.kolk@me.com](mailto:henk.kolk@me.com)

LinkedIn: [www.linkedin.com/in/henkkolk](http://www.linkedin.com/in/henkkolk)