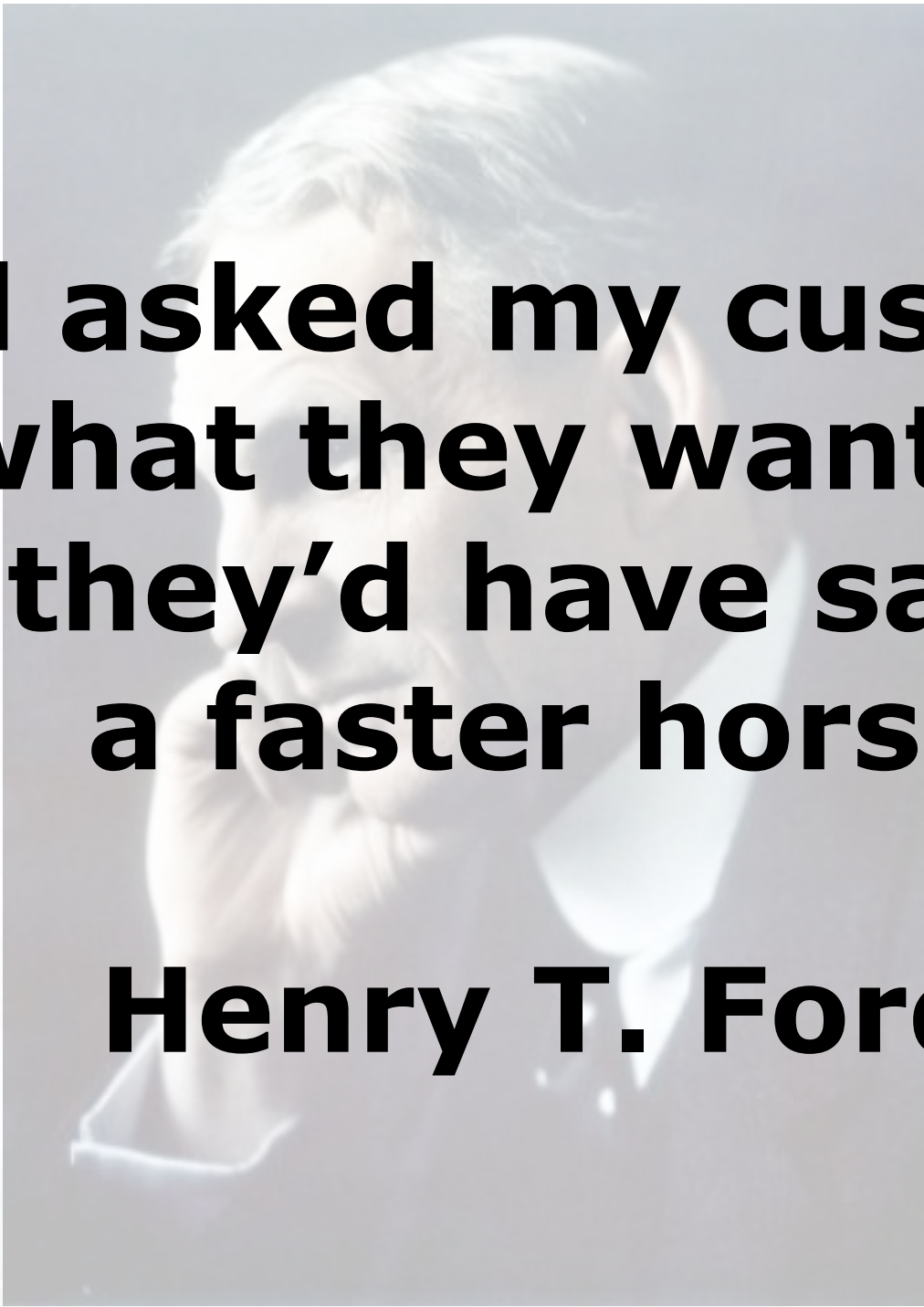# Do As I Say; Not As I Do?
Building Products Customers Want to Use

Jan Bosch

Director Software Center
www.software-center.se

Professor of Software Engineering
Chalmers University of Technology
Gothenburg, Sweden.

www.janbosch.com

February 2015
Industrial Engineering and Management
Guest lecture

If I'd asked my customers what they wanted, they'd have said a faster horse

Henry T. Ford

Customers don't know what they want. It's very hard to envision the solution you want without actually seeing it.

Marty Cagan

The critical failing of user interviews is that you're asking people to either remember past use or speculate on future use of a system

Jakob Nielsen

[The assumption that a] reasonably well-defined set of requirements exists, if only we take the time to understand them, is wrong

Dean Leffingwell

Customers don't know what's possible. Most have no idea about the enabling technologies involved

Marty Cagan

You can't just ask customers what they want and then try to give that to them. By the time you get it built, they'll want something new.
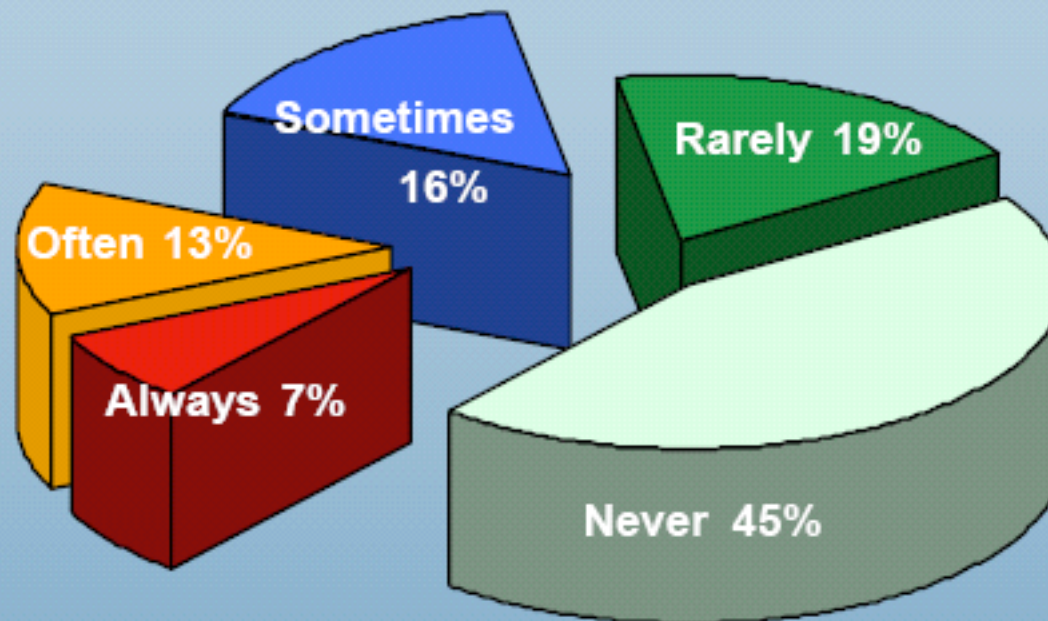
Steve Jobs

Featuritis

Our Research …
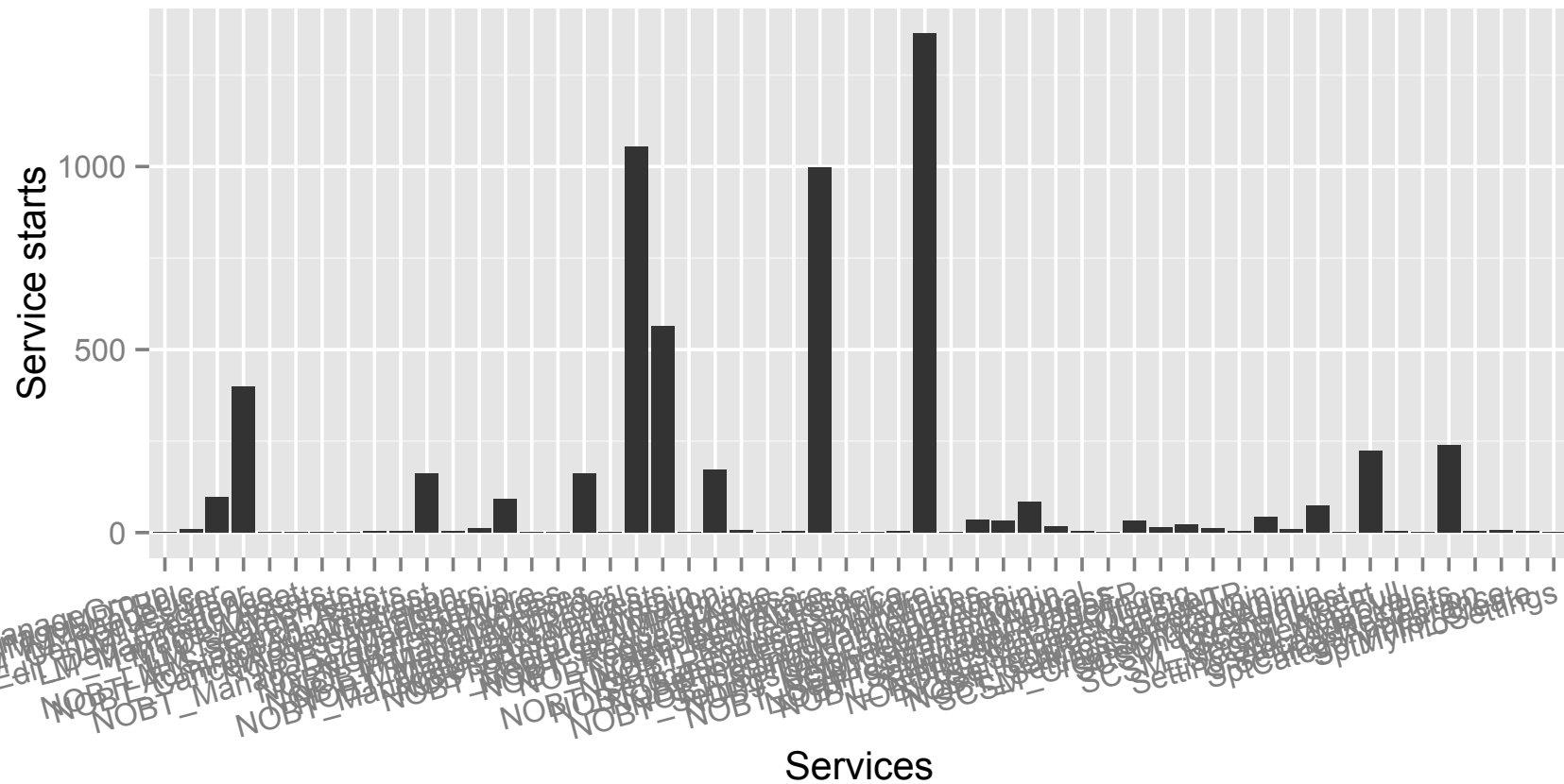
Three Key Take-Aways

- Customer don't know what they want until you show it to them – this requires **fast experimentation**

- Increasing **SPEED** gives you the short cycle times that allow for fast experimentation

- Continuous deployment allows you to organize R&D as **innovation experiment system**

Overview

- Vem är jag? Wie ben ik? Who am I?
- Trends in Software: Need for Speed
- Innovation Experiment Systems
- How to Get There: Stairway to Heaven
- Involving the Customer
- HYPEX Framework
- Case studies
- Conclusion

Software Center

Mission: Improve the software engineering capability of the Nordic Software-Intensive industry with an order of magnitude

Theme: Fast, continuous deployment of customer value

Success: Academic excellence

Success: Industrial impact

Overview

- Vem är jag? Wie ben ik? Who am I?
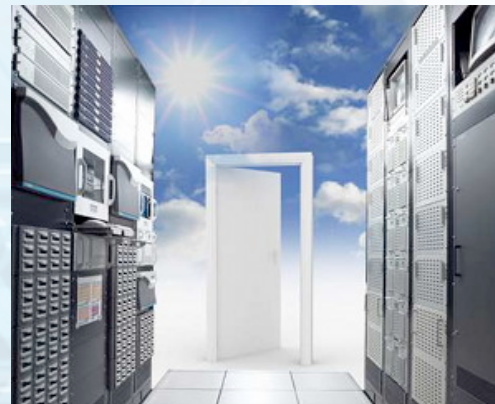- Trends in Software: Need for Speed
- Innovation Experiment Systems
- How to Get There: Stairway to Heaven
- Involving the Customer
- HYPEX Framework
- Case studies
- Conclusion

Trend: Products to Services
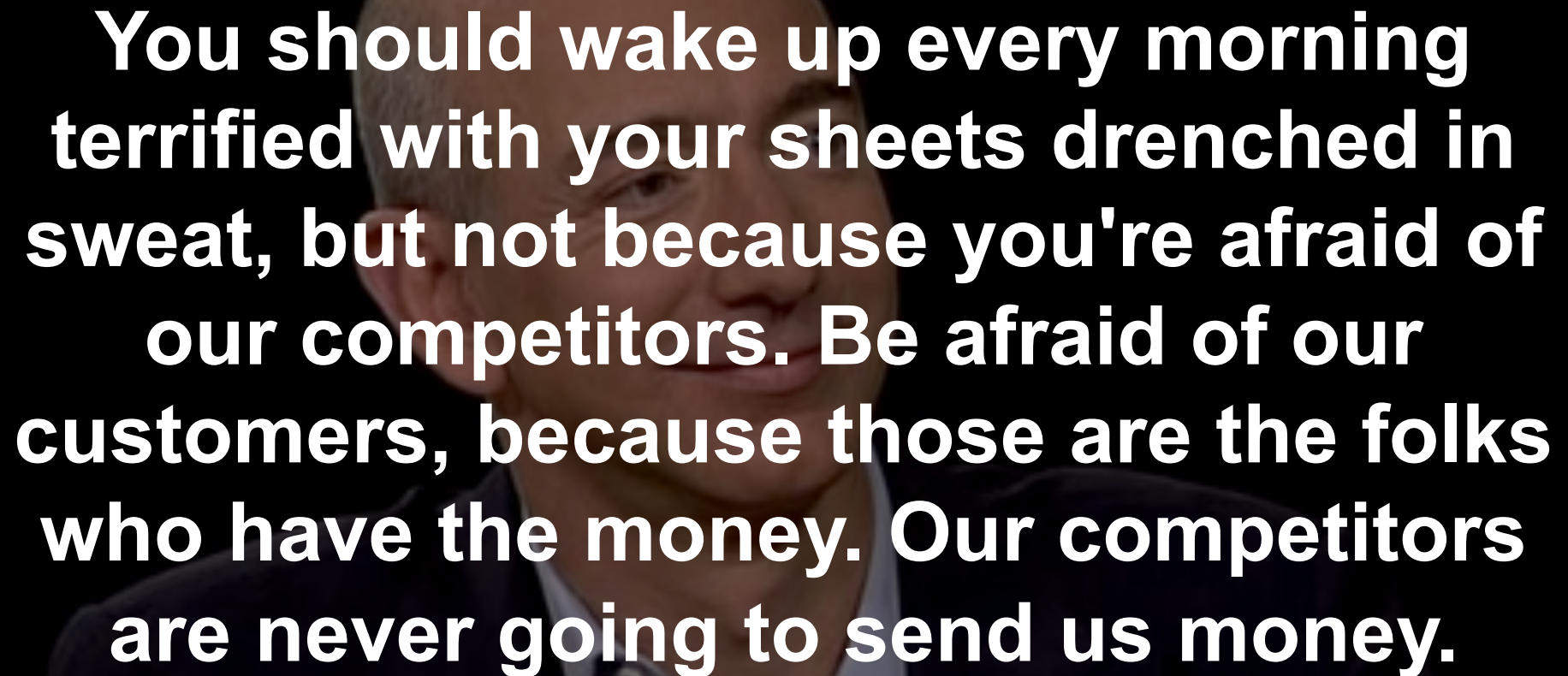
This requires continuous deployment throughout the lifetime of the product

Innovation Approaches
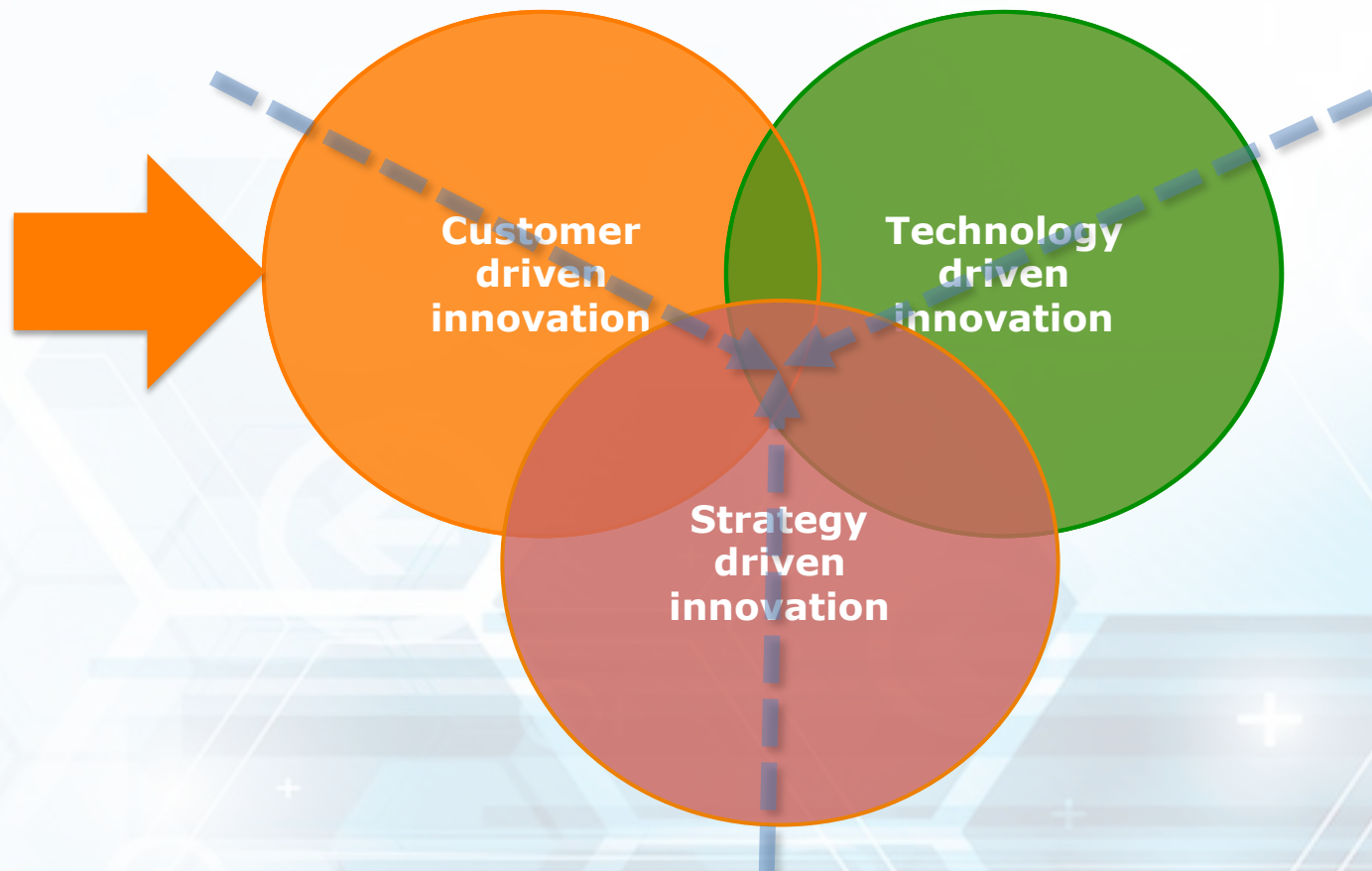
This requires continuous experimentation with customers

You should wake up every morning terrified with your sheets drenched in sweat, but not because you're afraid of our competitors. Be afraid of our customers, because those are the folks who have the money. Our competitors are never going to send us money.

- Jeff Bezos
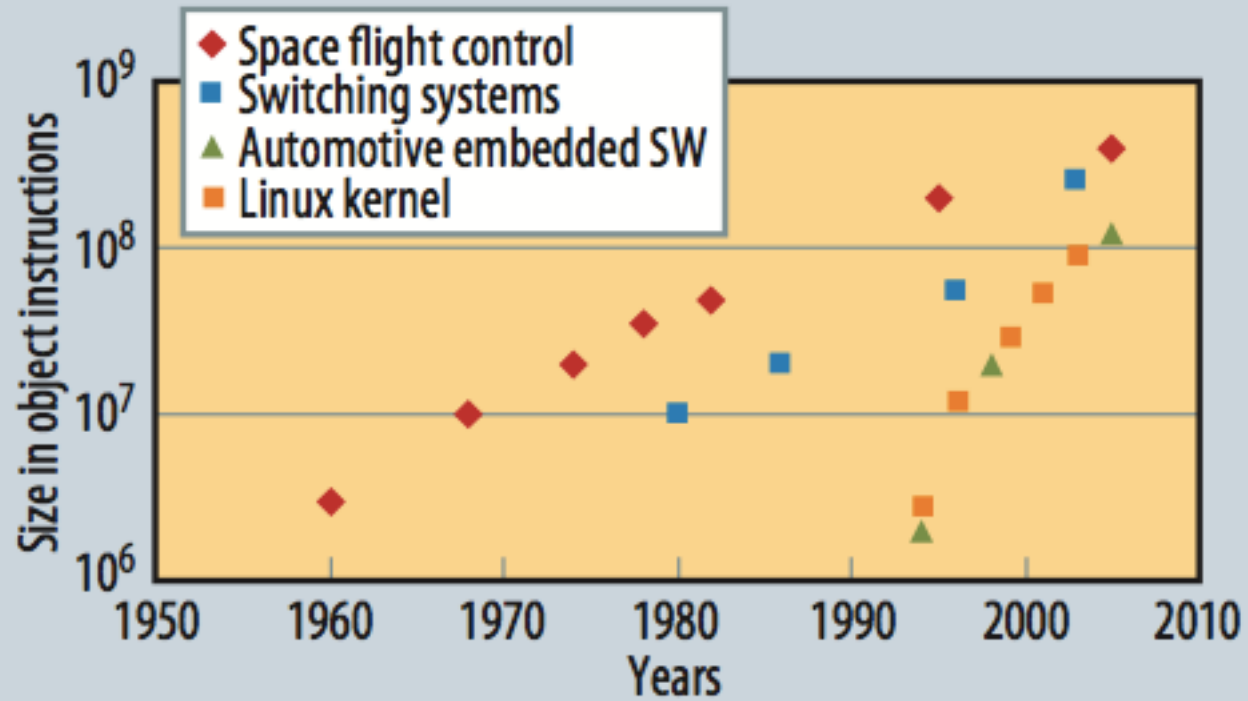
Figure 2. Complexity growth of embedded systems.

10x every ~7 years

Volvo XC 90
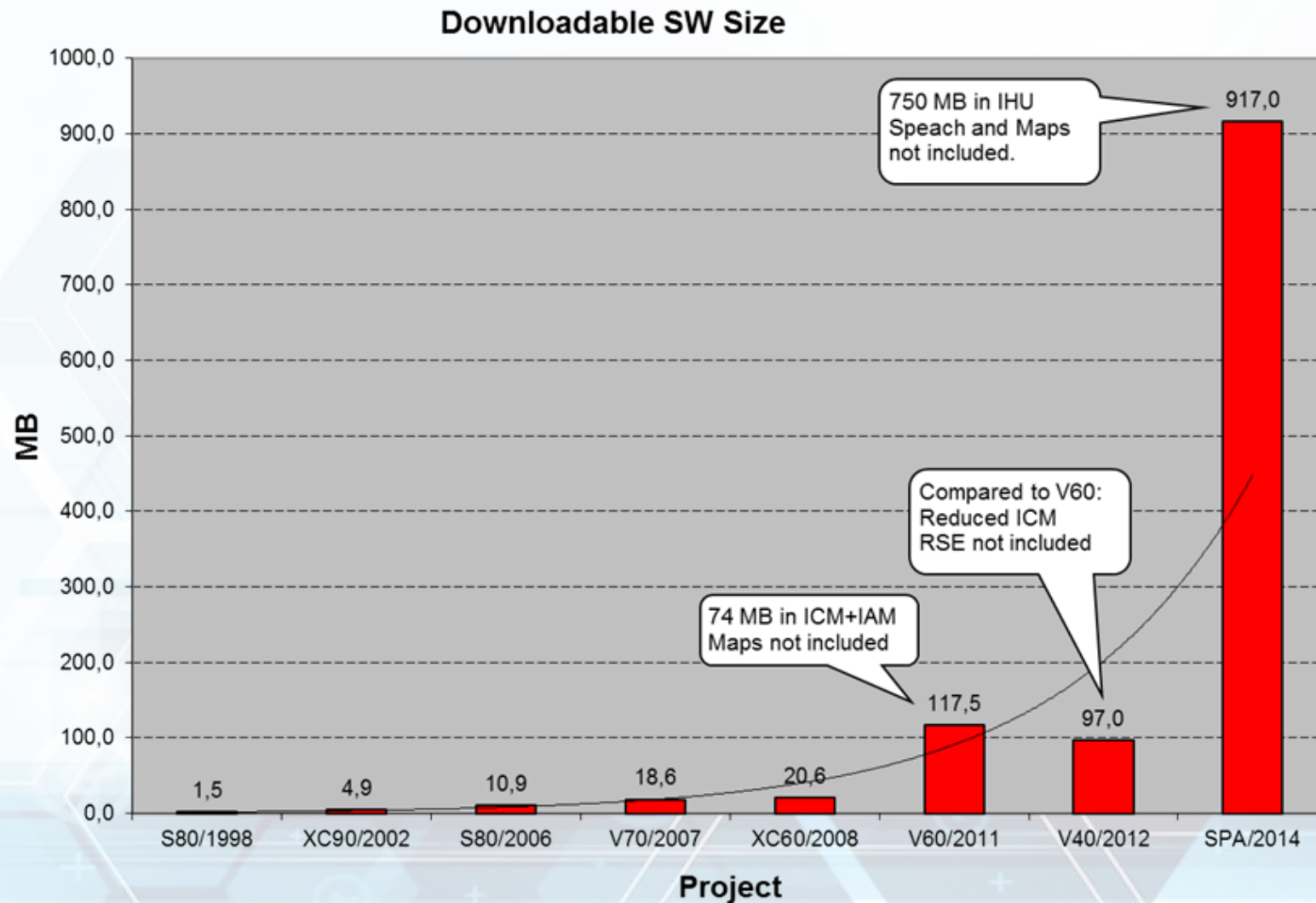
Trend: Need for Speed

Value Creation Shifts

Emerging companies highlight importance
of user contribution and social connectedness

Level of User Contribution

Founded	1984	1995	2004
1M users	~6 years	30 months	10 months
50M users	N/A	~80 months	~44 months

Need for Speed in R&D – An Example

- Company X: R&D is **10%** of revenue, e.g. 100M$ for a 1B$ product
- New product development cycle: **12 months**

- Alternative 1: improve efficiency of development with 10%
  - **10 M$** reduction in development cost
- Alternative 2: reduce development cycle with 10%
  - **100M$** add to top line revenue (product starts to sell 1.2 months earlier)

No efficiency improvement will outperform cycle time reduction

Need for Speed - Principles

Team
- 2 pizza's
- self-selected, directed and managed
- quantitative output metrics

Architecture
- simplicity – 3 API rule
- backward compatibility – no versions!
- focus on compositionality

Release process
- continuous, independent deployment
- all the way to customers – installed base
- measure usage to feed back into development

Overview

- Vem är jag? Wie ben ik? Who am I?
- Trends in Software: Need for Speed
- Innovation Experiment Systems
- Continuous deployment
- Web analytics
- HYPEX Framework
- Case studies
- Conclusion

What Do These Product Have in Common?
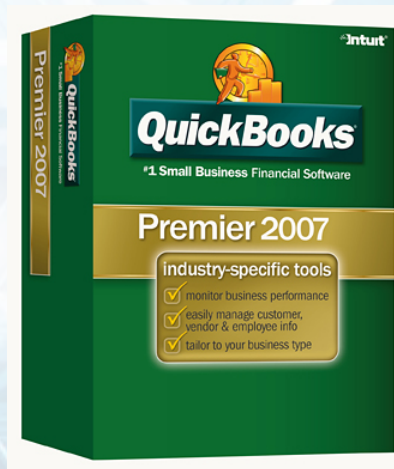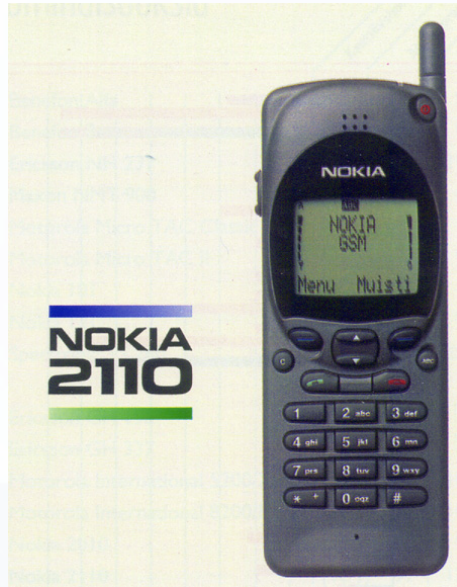
Example: Apple

The Myth	The Reality
Inspired innovation	Create and winnow 10 pixel-perfect prototypes
Inspired design	Build a better backstory (intricate layers of business design behind the products)
Brilliantly inspired marketing	Engineer the perfect customer experience to create customer experience and buzz
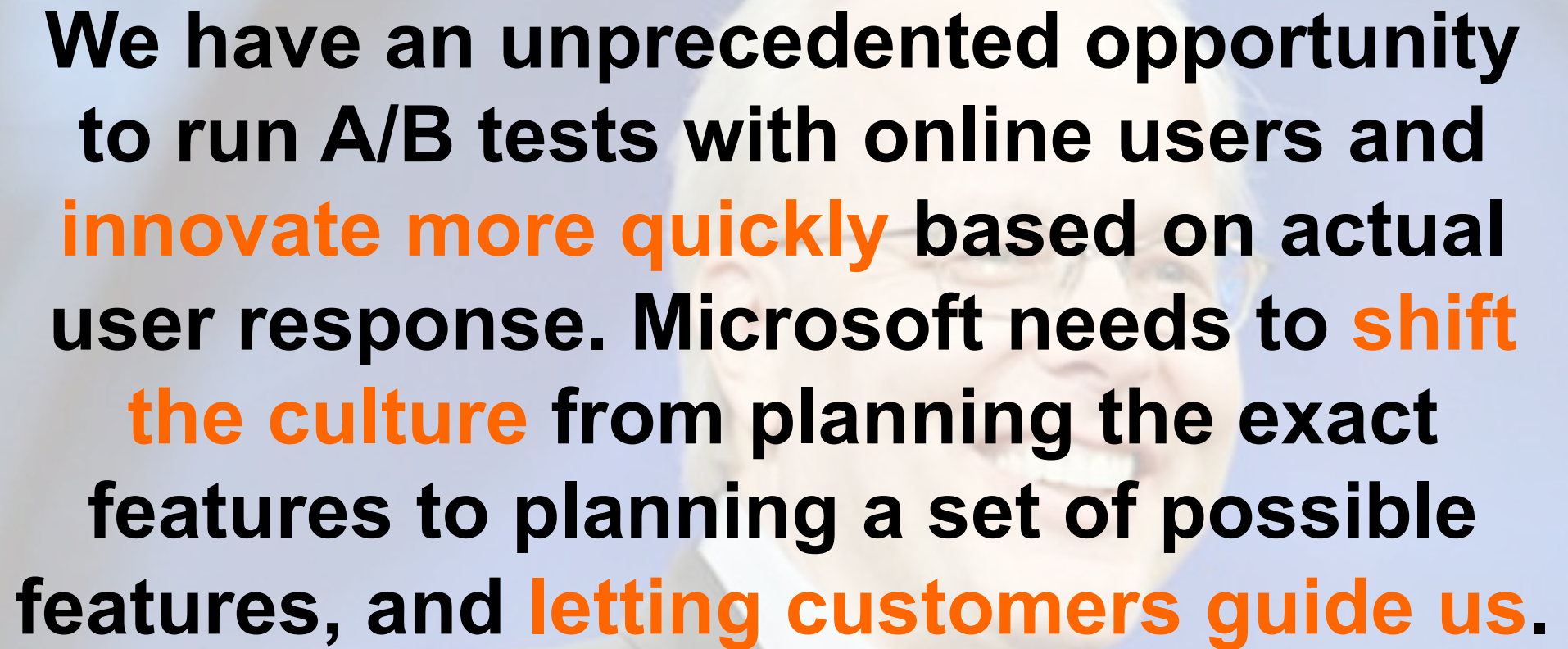
R&D as an Experiment System

Learning: the company running the most experiments against the lowest cost per experiment wins

Goal: increase the number of experiments (with customers) with an order of magnitude to ultimately accelerate organic growth

Usage and other data

Decide on new hypotheses to test based on data, ideas, strategy and customer requests

R&D iteration (2-4 weeks)

Three types of functionality
- Customer-requested
- Strategy driven
- Experiments

Installed Base (products @ customers)
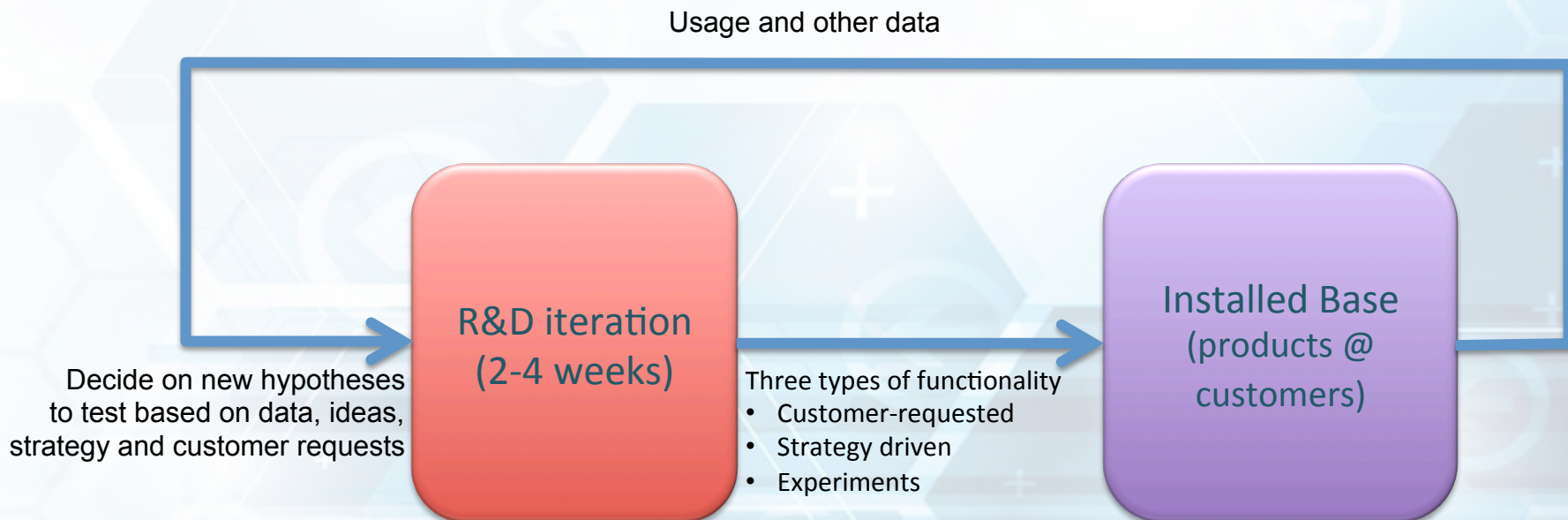
Decisions should be based on DATA, not opinions

We have an unprecedented opportunity to run A/B tests with online users and **innovate more quickly** based on actual user response. Microsoft needs to **shift the culture** from planning the exact features to planning a set of possible features, and **letting customers guide us**.

- Ray Ozzie

Overview

- Vem är jag? Wie ben ik? Who am I?
- Trends in Software: Need for Speed
- Innovation Experiment Systems
- How to Get There: Stairway to Heaven
- Involving the Customer
- HYPEX Framework
- Case studies
- Conclusion

Stairway to Heaven

Traditional Development

R&D Organization All Agile

Continuous Integration

Continuous Deployment

R&D as an Innovation System

product management → R&D → verification & validation → customer

Stairway to Heaven

Stairway to Heaven

Traditional Development

R&D Organization All Agile

Continuous Integration

Continuous Deployment

R&D as an Innovation System

product management → R&D

R&D → verification & validation → customer

Stairway to Heaven

Traditional Development

R&D Organization All Agile

Continuous Integration

Continuous Deployment

R&D as an Innovation System

product management

R&D

verification & validation

customer

Stairway to Heaven

Traditional Development

R&D Organization All Agile

Continuous Integration

Continuous Deployment

R&D as an Innovation System

10% waste

15% waste

20% waste

25% waste

30% waste

Rough estimation of waste and benefits

Financial Impact Potential

Ericsson

- R&D budget July 2011 – June 2012: 4,864 M$

- Software R&D (80%): 3891 M$

- Value of removing 5% waste:
  195 M$ (1280 MSEK)

AB Volvo

- Revenue 2011: 310 BSEK

- R&D budget 2011 (est. 5%):
  16 BSEK

- Software R&D (25%): 4 BSEK

- Value of removing 5% waste: **200 MSEK**

CI: Three Core Elements
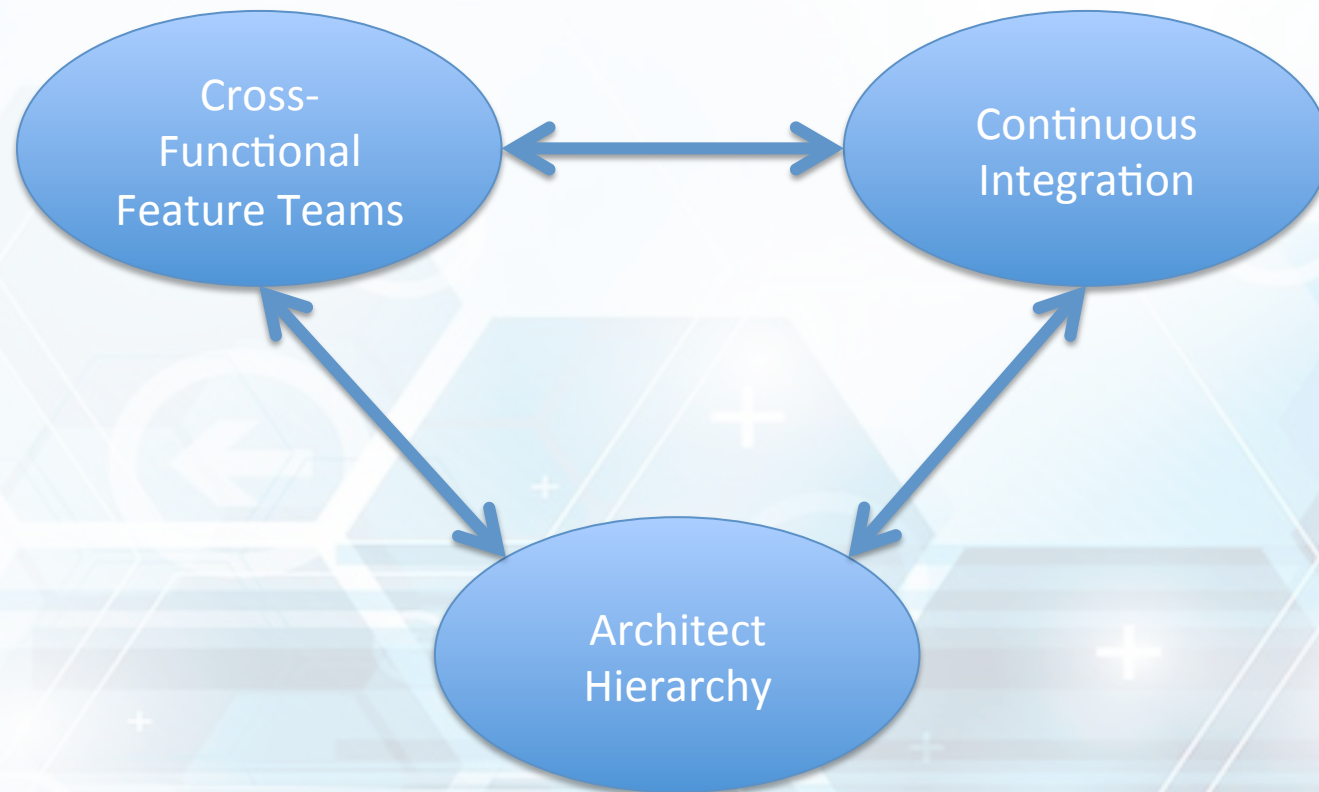
Organizing Development
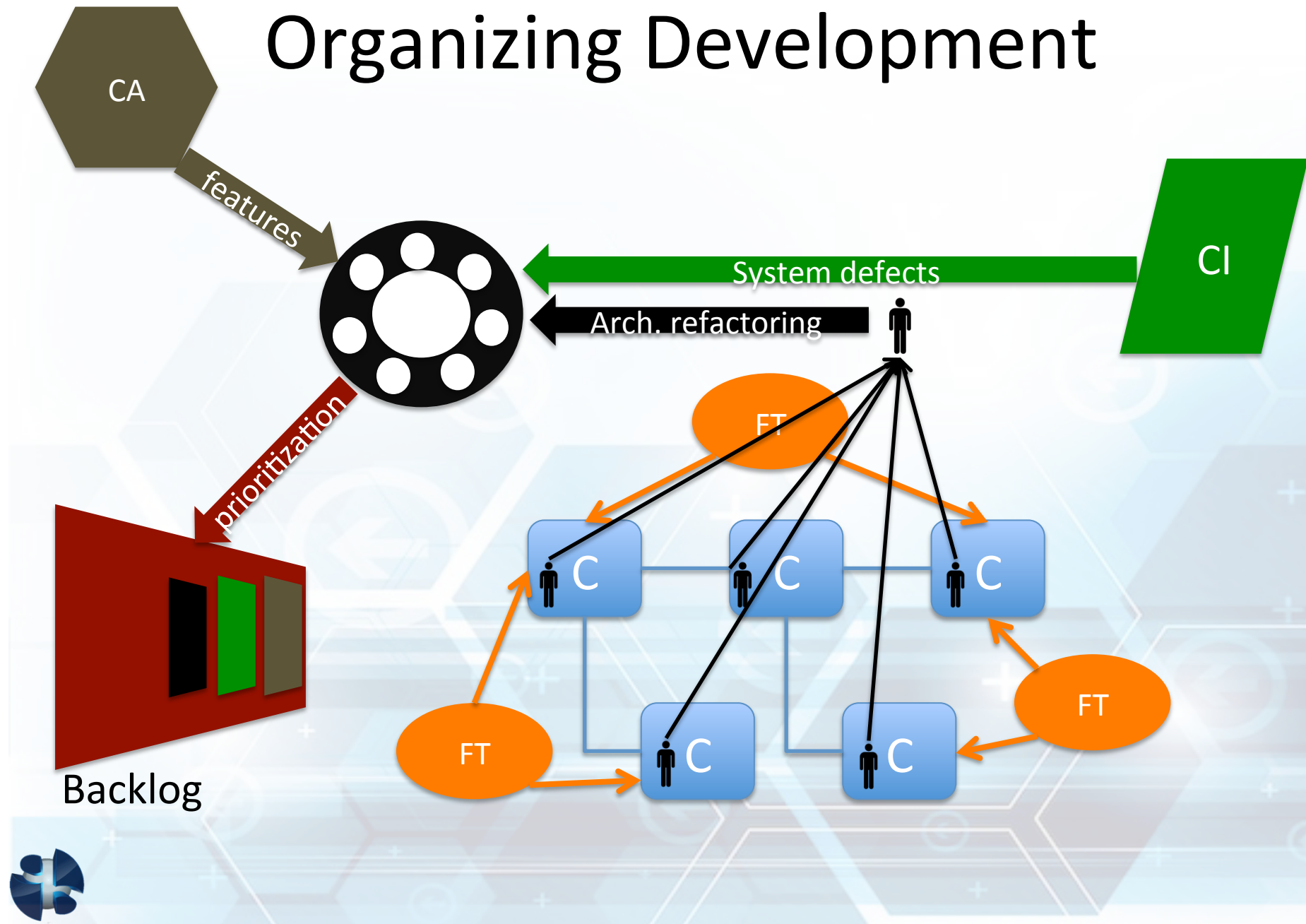
Continuous Deployment

- **Continuous Deployment** is the frequent (at least every 4 weeks) deployment of new software at products in the field initiated by the producer without (or with minimal) interaction with and approval from the customer.

CD: Two Loops

Release Team

(mostly) automated testing

Cross-Functional Feature Teams

Continuous Integration

Architect Hierarchy

Deployment infrastructure (+rollback)
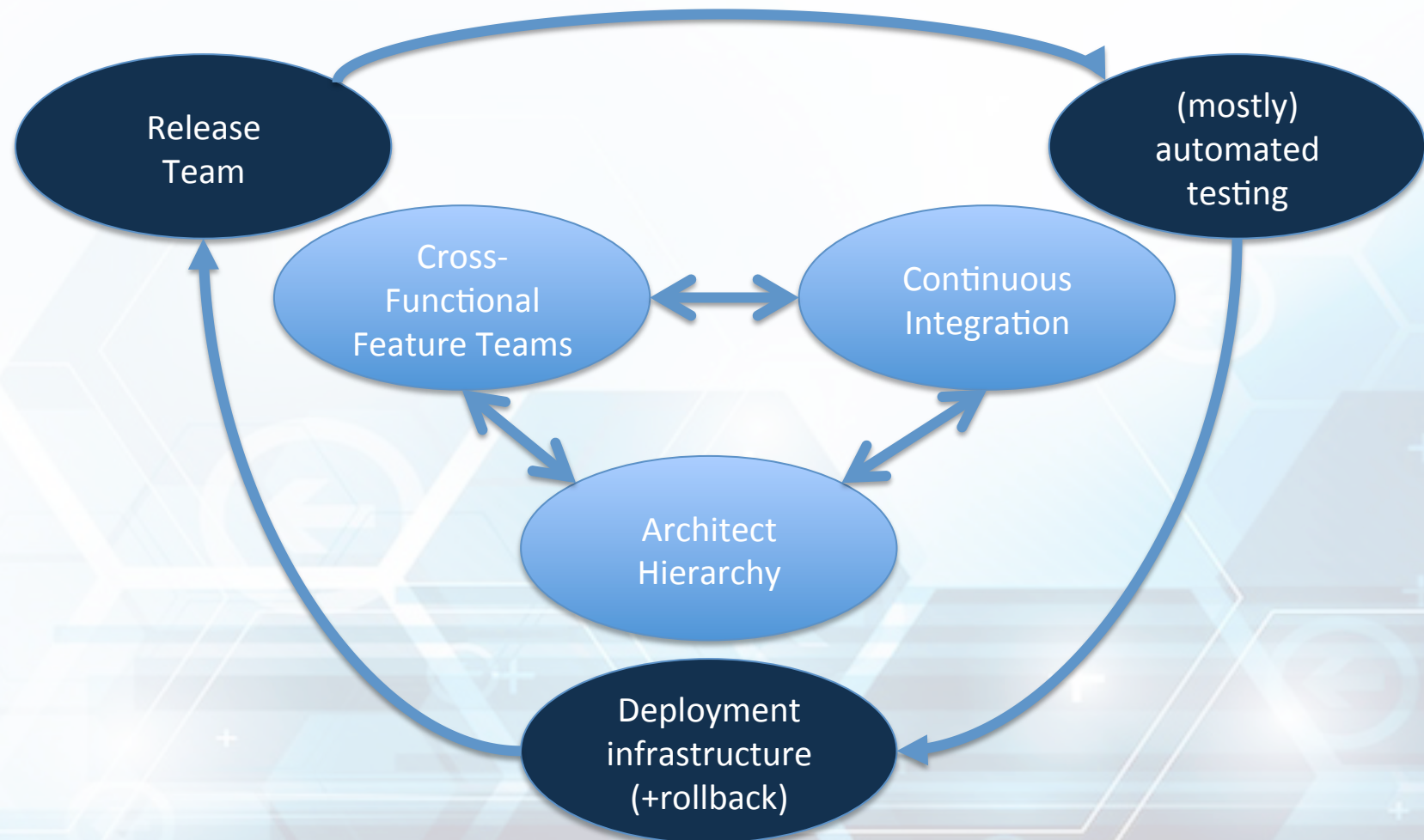
Later: collapse the loops

Overview

- Vem är jag? Wie ben ik? Who am I?
- Trends in Software: Need for Speed
- Innovation Experiment Systems
- How to Get There: Stairway to Heaven
- Involving the Customer
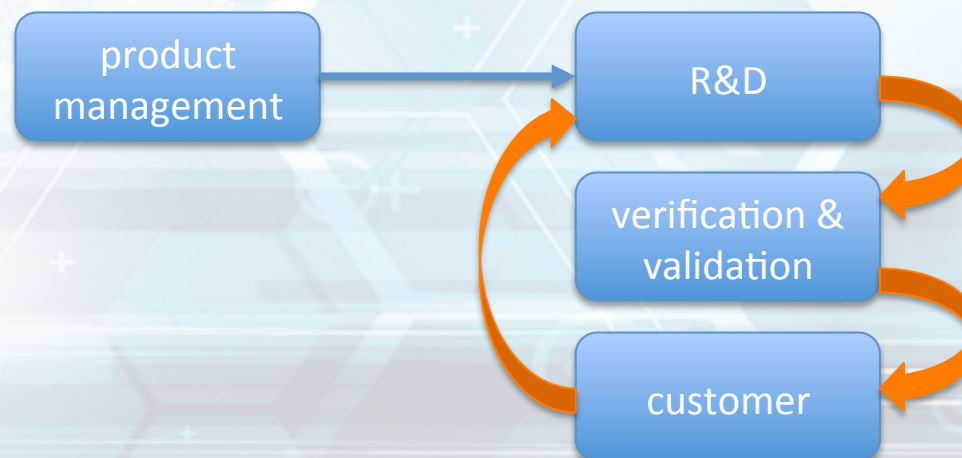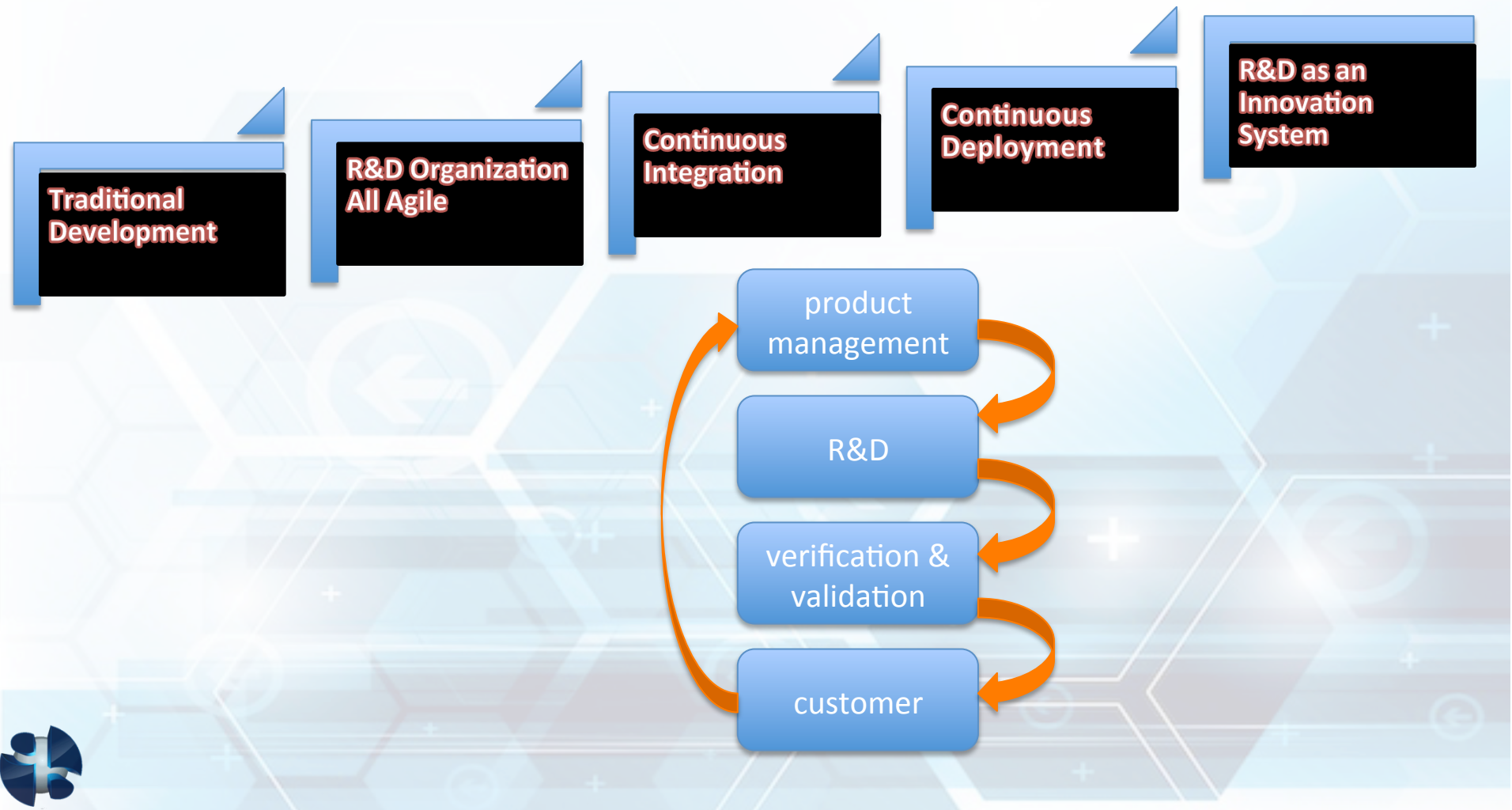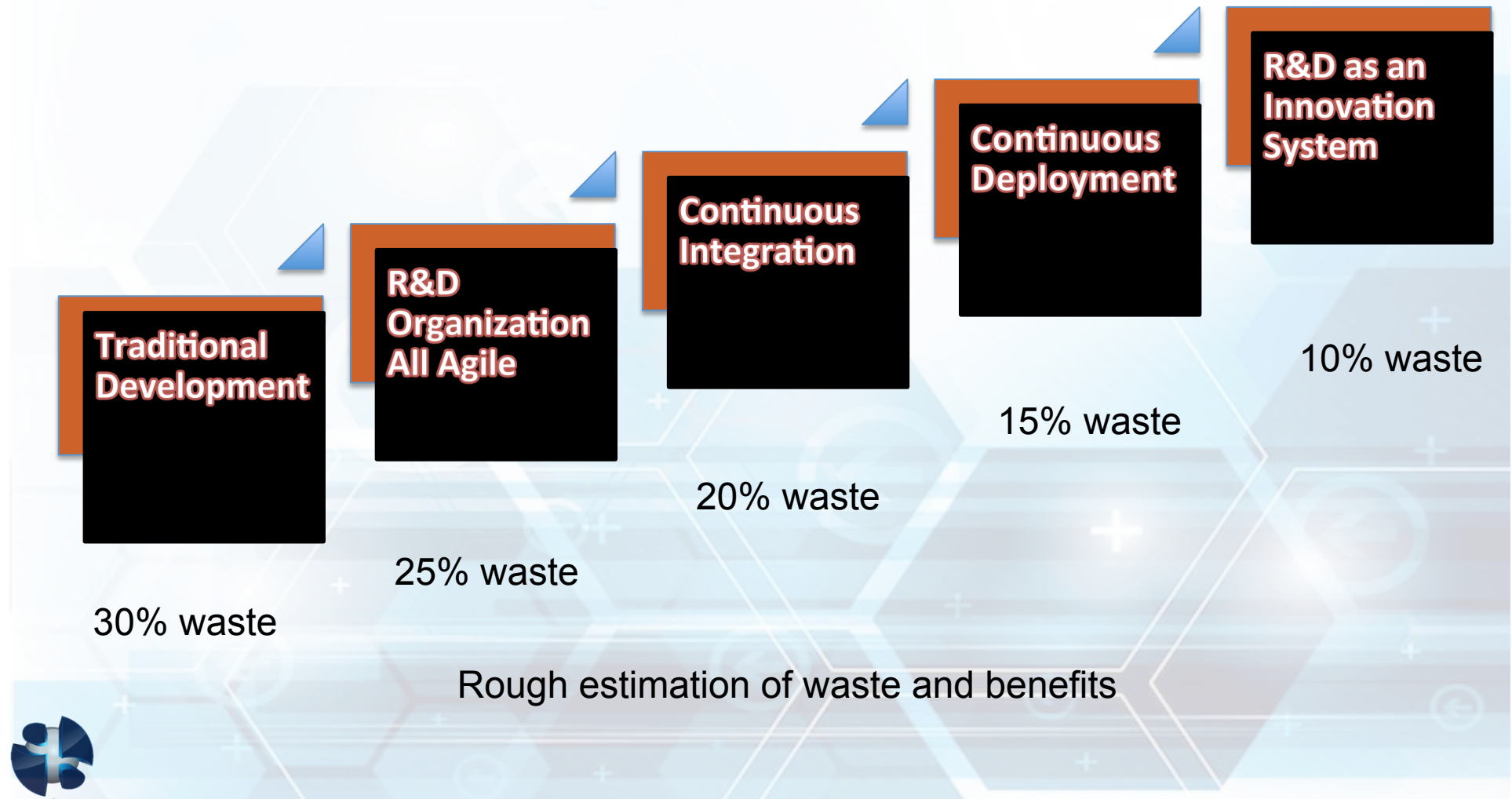- HYPEX Framework
- Case studies
- Conclusion

Interview Quotes

- "We DON'T know what features our customers use".

- "We have an idea on what functionality that is used...based on sales...but we DON'T really know...".

- "We can see some of the functionality that is used, but we CAN'T see how it is used".

- "Our development is affected in that we DON'T know what customers want".

- "We get feedback only on things that DON'T work...things that are problematic. This is not necessarily an indication of what is used the most...".

- "Does silence mean that things are OK? We DON'T know...".

Stages and Techniques

	Pre-Development	Development	Commercial deployment
Optimization	Ethnographic studies	Independently deployed extensions	Random selection of versions (A/B testing)
New features	Solution jams	Feature alpha In-product surveys	Instrumentation/ collecting metrics
New Products	Advertising Mock-ups BASES testing	Product alpha Labs website In-product advertising	Surveys Performance metrics

"In the field"   "In product"
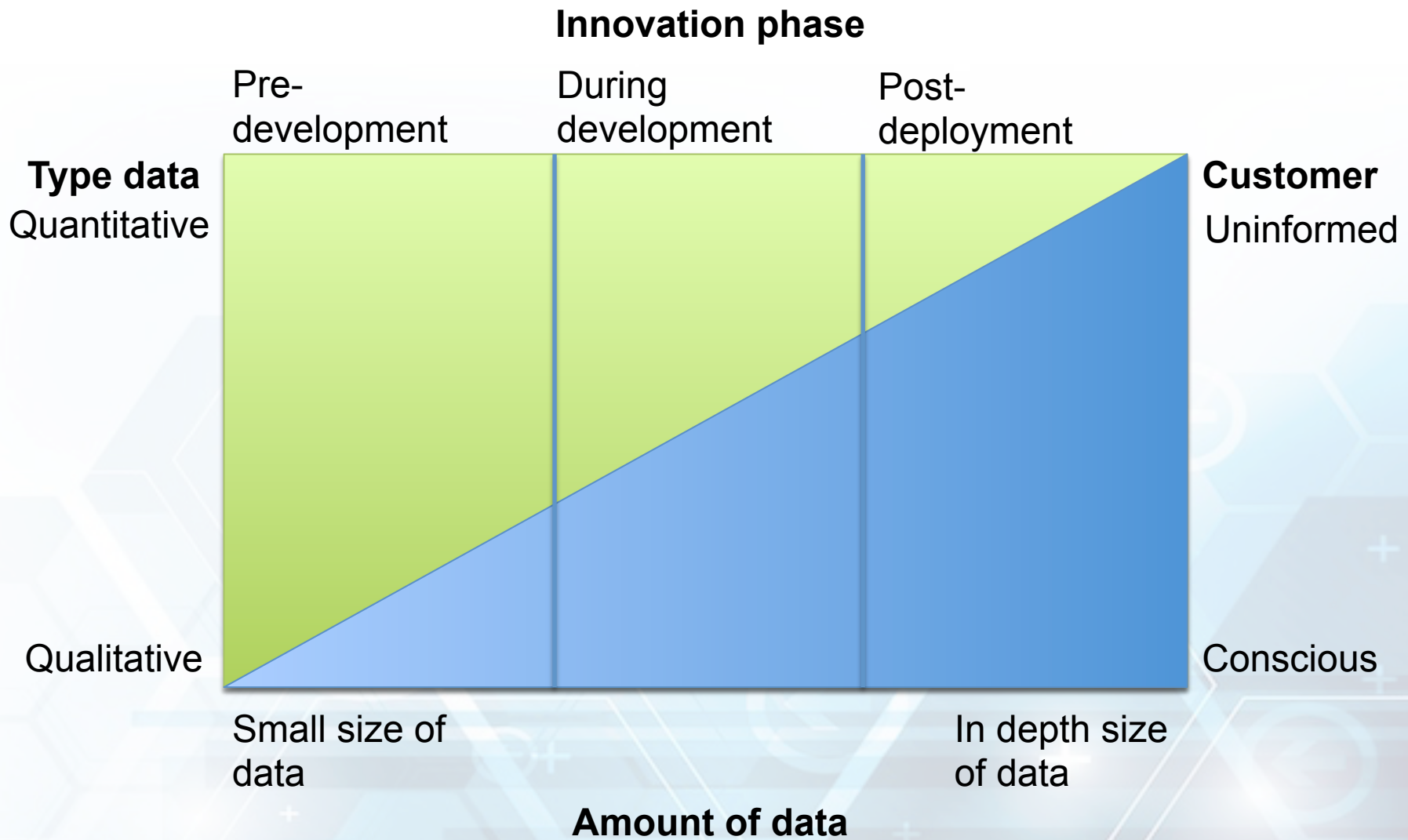
Source: Bosch-Sijtsema & Bosch, 2015

Conceptual Model
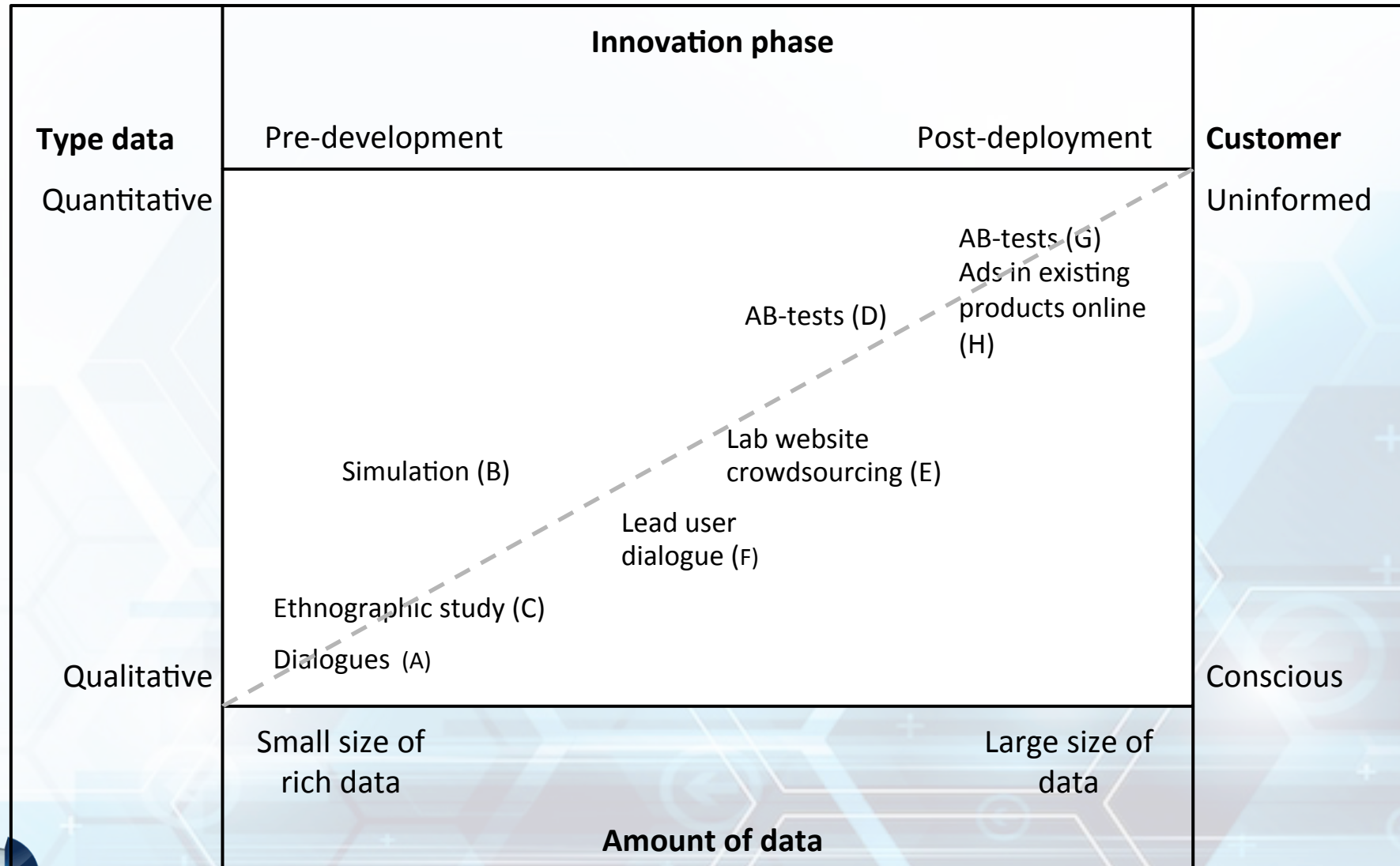
Pre-Development: Concept Testing

- What
  - Technique to test a concept before start of development.
  - Active Customer: Respondents are exposed to graphical or written product description. Answer a survey
  - Passive customer: Cross-sell, upsell or market a non-existing product (e.g. AdWords) to measure market interest
- To think about
  - Measure the conversion funnel
  - Consider A/B testing on your ad & pages

Quickbooks Online

- Cross-sell and upsell "ads" in the online product

- Measure conversion of those ads

- Decide on development of feature/product based on data

Evolution: A/B Testing

- What

  – A/B testing is a method of comparing a baseline control sample to a variety of single-variable test samples for improving some metric

- Alternatives

  – "Marketing" testing, e.g. colors, buttons and order of options

  – Alternative implementations of a feature

- To think about

  – Run multiple experiments simultaneously

  – Verify statistical relevance (free online tools exist)

Case: Open Infotainment Labs

Case: Open Infotainment Labs

- Feature development from a nominal lead-time of 1-3 years to 4-12 weeks?
- Working software was continuously validated in "real" environments
  - installed in both a driving simulator and real test cars
  - users evaluated the system
- 4th sprint: A/B experiment
  - Evaluating two layouts of the start screen
    - Implemented as two different launchers in Android
  - Mounted in a vehicle
  - 7 test drivers in total (3 used A, 4 used B)

A/B Testing Examples

- 37signals tested the headline on its pricing page. It found that "30-Day Free Trial on All Accounts" generated 30% more sign-ups than the original "Start a Highrise Account."

- Dustin found that "You should follow me on Twitter here" worked 173% better than his control text, "I'm on Twitter."

- A surprising conclusion from two separate A/B tests: putting human photos on a website increases conversion rates by as much as double.

- CareLogger increased its conversion rate by 34% simply by changing the color of the sign-up button from green to red.

- A software product company redesigned their product page to give it a modern look and added trust building elements (such as seals, guarantees, etc.). End result: they managed to increase total sales by 20%.

We don't have better algorithms.
We just have more data.

Peter Norvig
Chief Scientist, Google

Overview

- Vem är jag? Wie ben ik? Who am I?
- Trends in Software: Need for Speed
- Innovation Experiment Systems
- How to Get There: Stairway to Heaven
- Involving the Customer
- HYPEX Framework
- Case studies
- Conclusion

The HYPEX Model

- **WHAT?**
  - Process model for conducting feature experiments
- **HOW?**
  - A set of practices that allow for continuous validation of customer value
- **WHY?**
  - To significantly shorten feedback loops to customers
  - To increase accuracy of R&D investments

The HYPEX Model

Business strategy and goals

Strategic product goal

Feature backlog

generate

select

Feature: expected behavior (B_{exp})

implement MVF

B_{exp}

Gap analysis

actual behavior (B_{act})

Product

← no gap ($B_{act} = B_{exp}$)

↓ *relevant* gap ($B_{act} \neq B_{exp}$)

Experimentation

Develop hypotheses

implement alternative MVF

extend MVF

The HYPEX practices

4.
Implementation and instrumentation of code (MVF)

5.
Gap analysis between expected and actual behavior

3.
Identify expected behavior(s)

6.
Develop hypotheses on what causes the gap

2.
Select and specify feature

7.
New feature OR alternative feature development

1.
Generate feature backlog

8.
Hypotheses and analysis iteration

BUSINESS STRATEGY AND GOALS

Product

Overview

- Vem är vi? Wie zijn wij? Who are we?
- Trends in Software: Need for Speed
- Innovation Experiment Systems
- How to Get There: Stairway to Heaven
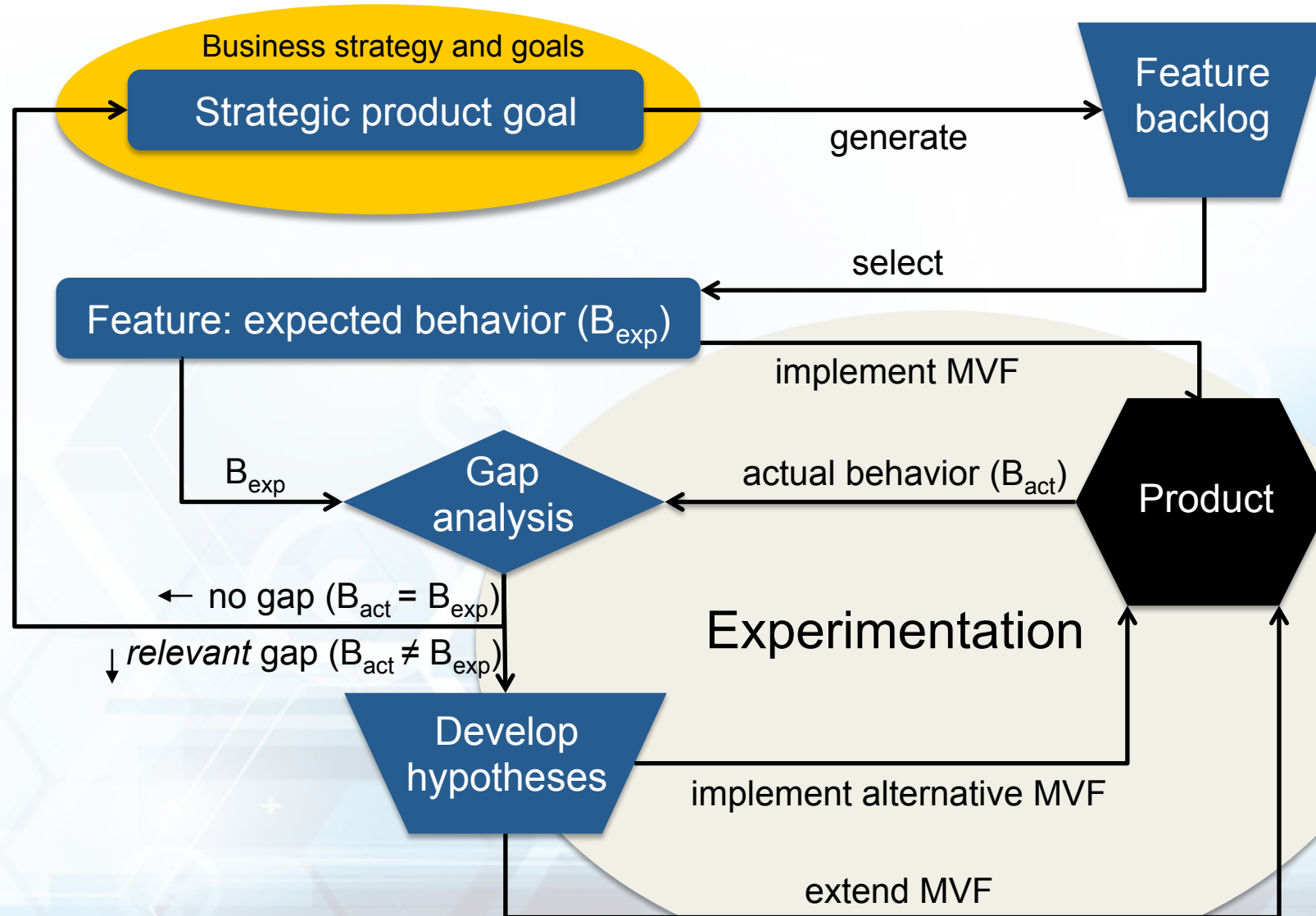- Involving the Customer
- HYPEX Framework
- Case studies
- Conclusion

Case: Company A

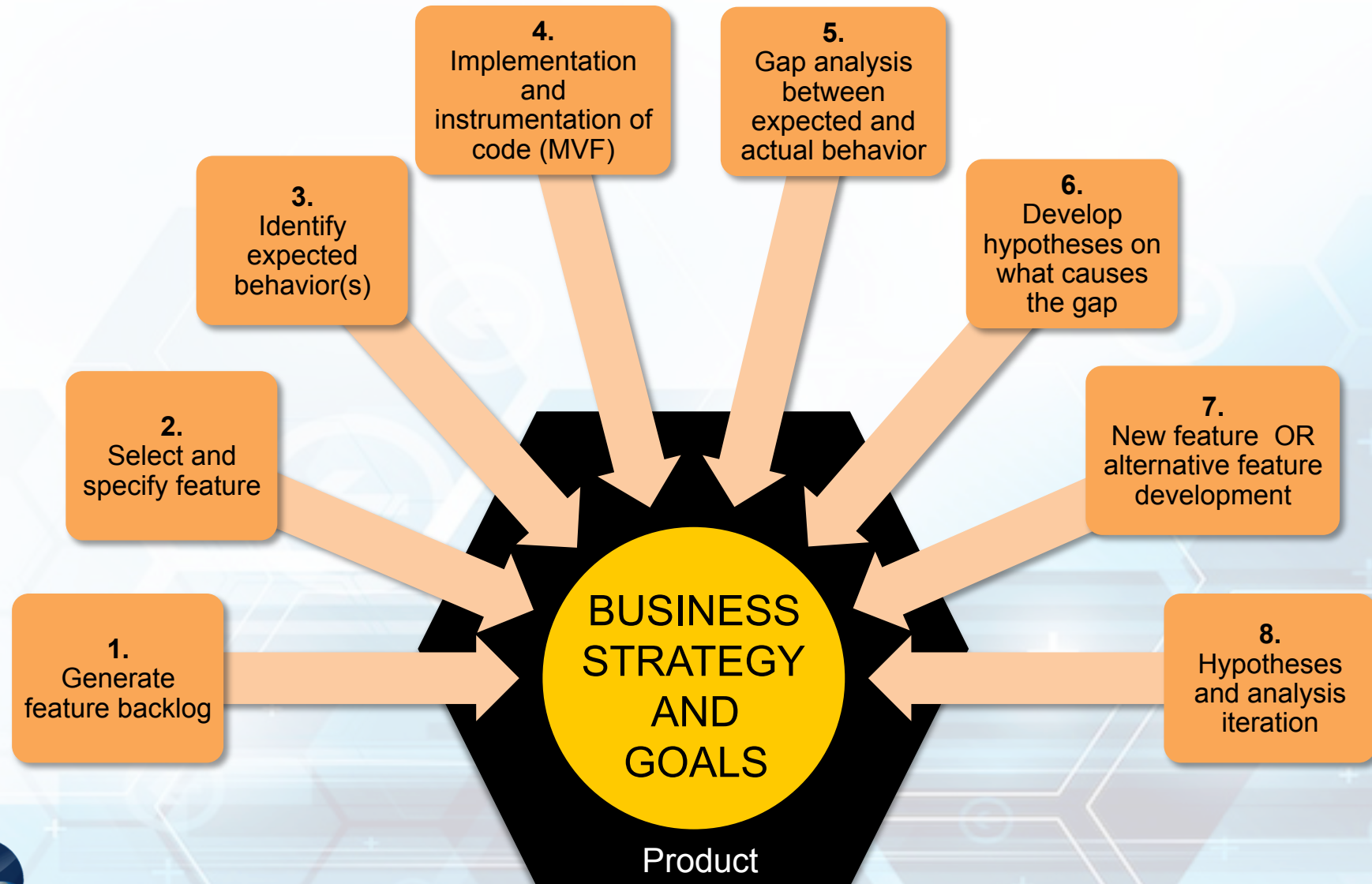- **Domain:**
  - Network cameras, video surveillance, video management SW

- **Feature experiment:**
  - New approach on an existing feature for accessing the cameras remotely by using mobile devices.
  - The new approach will improve the success rate of the 'one-click' workflow.
  - When accessing a camera from a remote device it will relay through a mediator that will authenticate the connection.
  - Initially, the feature experiment is conducted with company employees worldwide.

Lessons Learned

- ☺
  - Improved communication (HW/SW)
  - Rewarding definition of metrics
  - How to collect data from our products
  - What data to collect from our products
- ☹
  - Establishing an "experiment mind-set"
  - Identifying feature
  - Identifying customers to work with
  - Slow development lead-time
  - Slow distribution of cameras to customers

Case: Company B

- **Domain:**
  - Navigational IS, optimization solutions, crew and fleet management systems.

- **Feature experiment:**
  - Improve the ability to compare KPI's for optimization runs.

  - A new feature is developed that allows the user to tune input parameters in correspondence with KPI's, and to send new optimization runs based on these.

Lessons Learned

- ☺
  - Understanding of quality issues
  - Understanding for early user involvement
  - Communication between service managers and product development
  - Knowledge exchange within company ecosystem

- ☹
  - Definition of metrics – what data to log?
  - Deployment of builds to customers
  - Customer training
  - Uncertainty if customers attend training sessions

Overview

- Vem är jag? Wie ben ik? Who am I?
- Trends in Software: Need for Speed
- Innovation Experiment Systems
- How to Get There: Stairway to Heaven
- Involving the Customer
- HYPEX Framework
- Case studies
- Conclusion

Customers Don't Know

- Customer don't know what they want until you show it to them – this requires **fast experimentation**

- Test continuously with customers that you're on the right track

- Apply the right technique for the right stage and problem

- Focus on small, cross-functional teams, give them direction and get out of their way

Speed

- Increasing **SPEED** gives you the short cycle times that allow for fast experimentation

- If you're not a front-line engineer, there is only ONE measure that justifies your existence: how have you helped teams move faster?

- Don't optimize efficiency, optimize speed

Innovation Experiment Systems

- Continuous deployment allows you to organize R&D as **innovation experiment system**

- Legacy system != slow (necessarily)

- Traditional working methods no longer apply

- Decouple components, decouple teams and decouple organizations

- Lean and agile at scale

Not My Job?!

Strong LEADERSHIP needed from YOU

www.software-center.se
Chalmers University of Technology