Lecture 04

Cloud, CD, DevOps

Kari Systä

15.09.2020

Tampereen yliopisto
Tampere University

Schedule for coming weeks

Week	Lecture	Plussa exercises (deadlines)
-1		
0		17.08 Plussa open for students
1 / 35	25.08 Intro to the course and topic.	20.08 Background survey opens
2 / 36	01.09 Virtualization, what, why and how. Intro to containers and Docker.	04.09 Background survey closes
3 / 37	08.09 Cloud and scalability, implications to SW development and business	11.09/17.09 Docker exercise closes
4 / 38	15.09 Continuous deployment, what and why	17.09 Docker compose e. opens
5 / 39	21.09 Continuous deployment, tools and techniques	
6 / 40	28.09 Issues on cloud-SW: isolation, dependency management etc	01.10 Docker compose e. closes ??.10 Next exercise opens

Tampereen yliopisto
Tampere University

Course practicalities

- At the moment 62 students have returned the first exercise
  - Some have missed the two-deadline approach
- Nice collection of programming languages:
  Python, JavaScript, Golang, Java, C#, PHP, C, C++, Ruby, Racket,
  (sh, HTML)
- Some students have had difficulties in getting access to a Linux VM
  - I know two cases, but in case there are more let me know
  - My recommedation is Linux. You may use Docker on top of Windows, but under your own responsibility. The staff members cannot help in case of difficulties.

Tampereen yliopisto
Tampere University

Cloud computing - definition

- In 1997, Professor Ramnath Chellapa of Emory University defined Cloud Computing as the new *'computing paradigm, where the boundaries of computing will be determined by economic rationale, rather than technical limits alone.'*
- *NIST:* Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.
-

Tampereen yliopisto
Tampere University

Service models

SaaS

PaaS

IaaS

Infrastructure as a Service (IaaS). The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

Examples?

Tampereen yliopisto
Tampere University

Service models

SaaS

PaaS

IaaS

Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

Examples?

Tampereen yliopisto
Tampere University

Service models

SaaS

PaaS

IaaS

Software as a Service (SaaS). The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

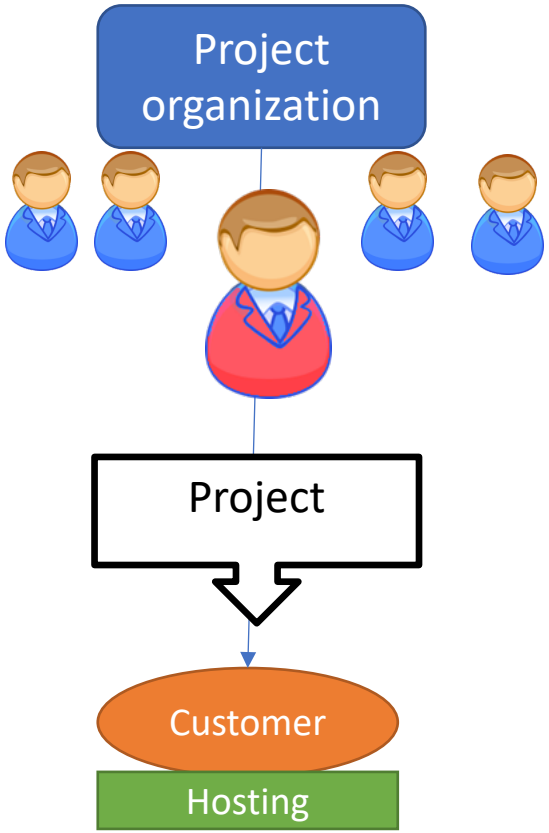
Examples?

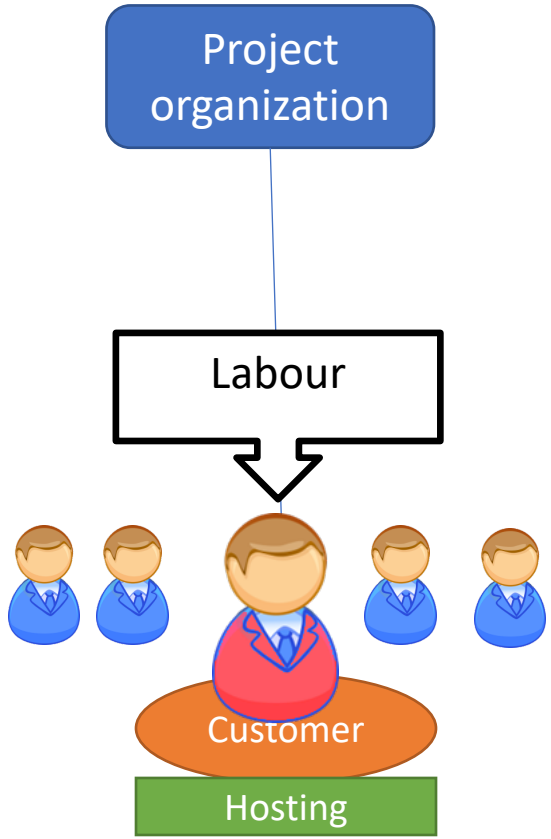
Tampereen yliopisto
Tampere University

Software development and bussiness

Tampereen yliopisto
Tampere University

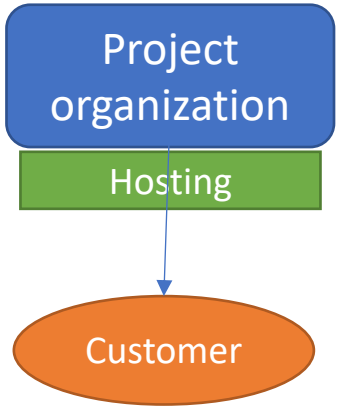
Ways to conduct SW business

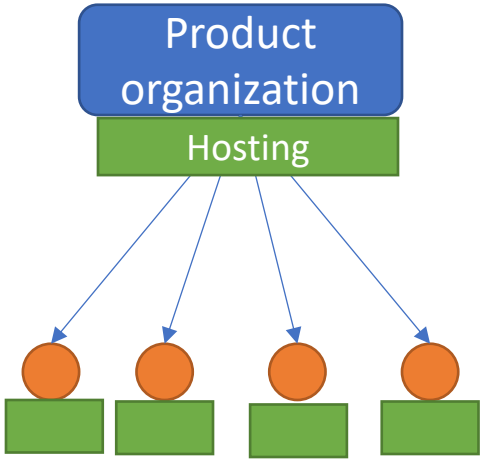




Tampereen yliopisto
Tampere University

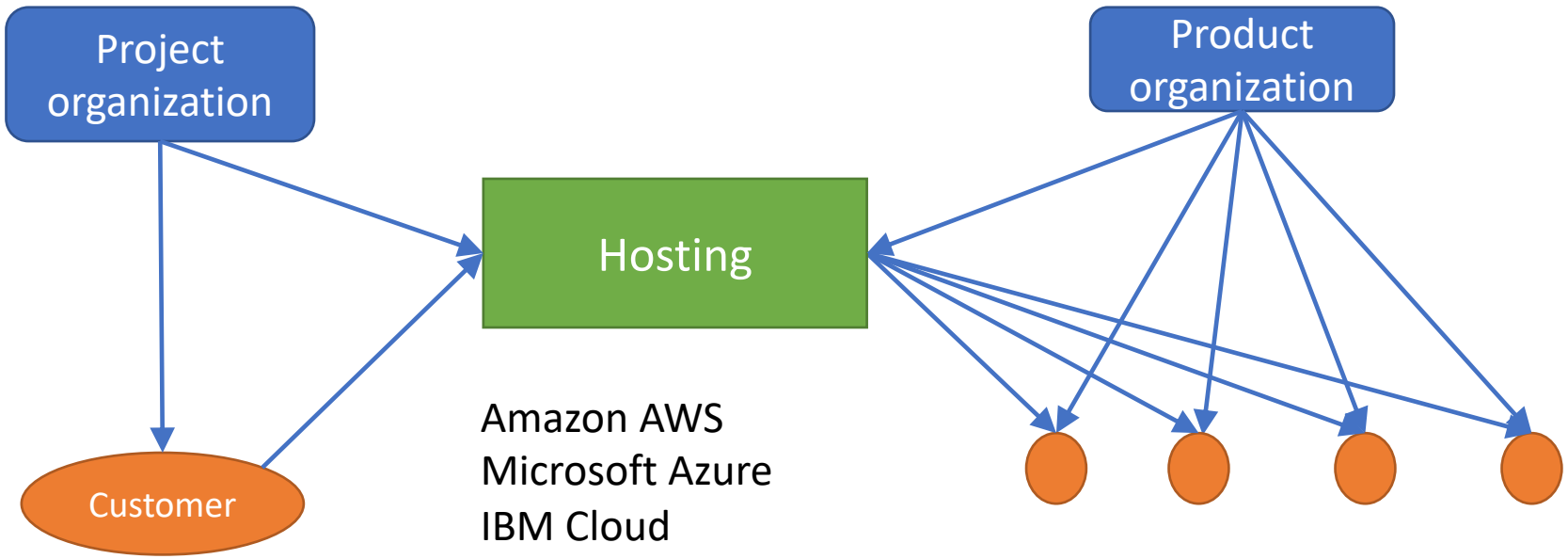
Software as a service (SaaS)





Tampereen yliopisto
Tampere University

Hosting can be a separate business



Amazon AWS
Microsoft Azure
IBM Cloud
Google Alphabet
OVH
…
Tieto
Cybercom

Tampereen yliopisto
Tampere University

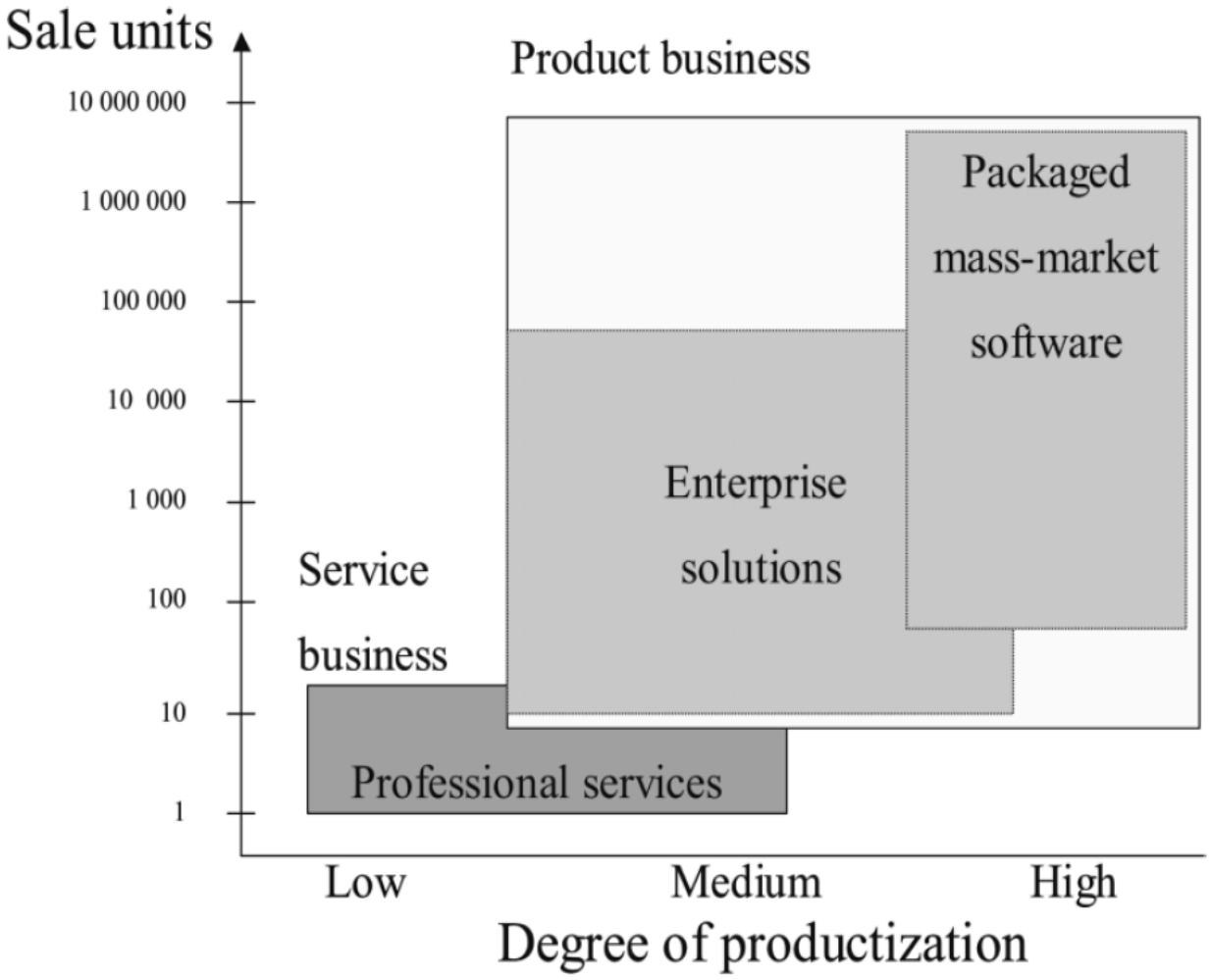
Implied changes to SW business

Teppo Yrjönkoski and Kari Systä. 2019. Productization Levels Towards Whole Product in SaaS Business. IWSiB ’19, August 26, 2019, Tallinn, Estonia

- Easy discovery and access through internet enable worldwide market for SaaS product vendors. Simultaneously, all players have same playground and competition is open and global.
- SaaS customers expect new technologies, systems and applications and faster reactions from their software vendors than customers of traditional software.
- The increased speed multiplies the risks.
  - Great success and failure overnight?
- Cash-flow and funding changes

Tampereen yliopisto
Tampere University

What is productization?



- ”a standardized process which aims to produce a high quality commercial good or service viable in the market from produced information”.
- Emphasis on activities beyond R&D.

Tampereen yliopisto
Tampere University

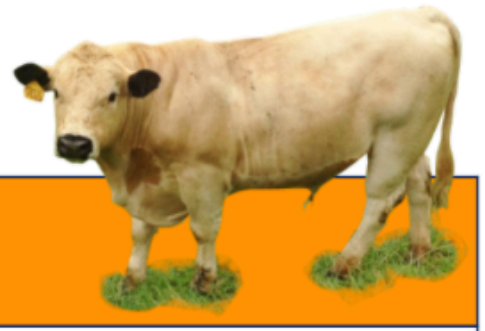
Productization is SaaS business

- Implications of fast cycles and uncertainty?
- Yrjönkoski proposes a three level model:
  - Proof of concept
  - Individual sales from 1st to 10th customer
  - Mass distribution

Tampereen yliopisto
Tampere University

Implications to developers

https://cloudrumblings.io/cloud-farm-pets-cattle-unicorns-and-horses-85271d915260



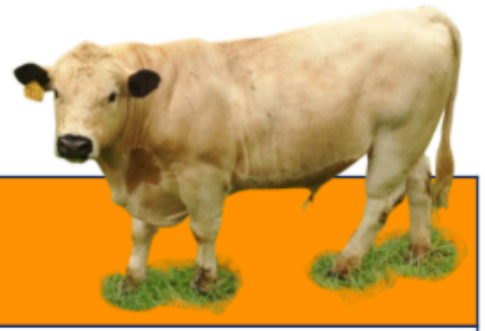
Pets Legacy Infrastructure	Cattle Cloud-Friendly Infrastructure
Pets are given names like grumpycat.petstore.com They are unique, lovingly hand raised and cared for When they get ill, you nurse them back to health	Cattle are given numbers like 10200713.cattlerancher.com They are almost identical to other cattle When they get ill, you replace them and get another
Infrastructure is a permanent fixture in the data center	Infrastructure is stateless, ephemeral, and transient
Infrastructure takes days to create, are serviced weekly, maintained for years, and requires migration projects to move	Infrastructure is instantiated, modified, destroyed and recreated in minutes from scratch using automated scripts
Infrastructure is modified during maintenance hours and generally requires special privileges such as root access	Infrastructure uses version-controlled scripts to modify any service without requiring root access or privileged logins
Infrastructure requires several different teams to coordinate and provision the full environment	Infrastructure is self-service with the ability to provision computing, network and storage services with a single click
Infrastructure is static, requiring excess capacity to be dormant for use during peak periods of demands	Infrastructure is elastic and scales automatically, expanding and contracting on-demand to service peak usage periods
Infrastructure is an capital expenditure that charges a fix amount regardless of usage patterns	Infrastructure is a operating expenditure that charges only for services when they are consumed

When your servers get sick, do you nurse them back to health or shoot them?

Tampereen yliopisto
Tampere University

Implications to developers

https://cloudrumblings.io/cloud-farm-pets-cattle-unicorns-and-horses-85271d915260



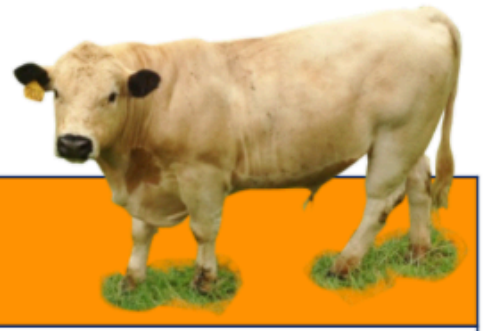
Pets Legacy Infrastructure	Cattle Cloud-Friendly Infrastructure
Pets are given names like grumpycat.petstore.com They are unique, lovingly hand raised and cared for When they get ill, you nurse them back to health	Cattle are given numbers like 10200713.cattlerancher.com They are almost identical to other cattle When they get ill, you replace them and get another
Infrastructure is a permanent fixture in the data center	Infrastructure is stateless, ephemeral, and transient
Infrastructure takes days to create, are serviced weekly, maintained for years, and requires migration projects to move	Infrastructure is instantiated, modified, destroyed and recreated in minutes from scratch using automated scripts
Infrastructure is modified during maintenance hours and generally requires special privileges such as root access	Infrastructure uses version-controlled scripts to modify any service without requiring root access or privileged logins
Infrastructure requires several different teams to coordinate and provision the full environment	Infrastructure is self-service with the ability to provision computing, network and storage services with a single click
Infrastructure is static, requiring excess capacity to be dormant for use during peak periods of demands	Infrastructure is elastic and scales automatically, expanding and contracting on-demand to service peak usage periods
Infrastructure is an capital expenditure that charges a fix amount regardless of usage patterns	Infrastructure is a operating expenditure that charges only for services when they are consumed

When your servers get sick, do you nurse them back to health or shoot them?

Tampereen yliopisto
Tampere University

Implications to developers

https://cloudrumblings.io/cloud-farm-pets-cattle-unicorns-and-horses-85271d915260



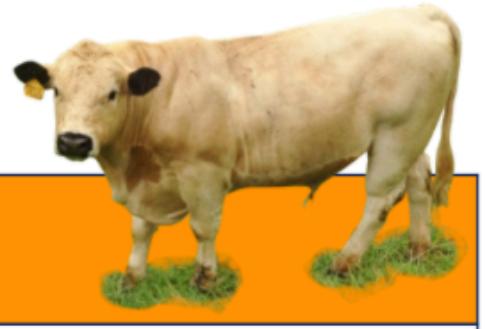
Pets Legacy Infrastructure	Cattle Cloud-Friendly Infrastructure
Pets are given names like grumpycat.petstore.com They are unique, lovingly hand raised and cared for When they get ill, you nurse them back to health	Cattle are given numbers like 10200713.cattlerancher.com They are almost identical to other cattle When they get ill, you replace them and get another
Infrastructure is a permanent fixture in the data center	Infrastructure is stateless, ephemeral, and transient
Infrastructure takes days to create, are serviced weekly, maintained for years, and requires migration projects to move	Infrastructure is instantiated, modified, destroyed and recreated in minutes from scratch using automated scripts
Infrastructure is modified during maintenance hours and generally requires special privileges such as root access	Infrastructure uses version-controlled scripts to modify any service without requiring root access or privileged logins
Infrastructure requires several different teams to coordinate and provision the full environment	Infrastructure is self-service with the ability to provision computing, network and storage services with a single click
Infrastructure is static, requiring excess capacity to be dormant for use during peak periods of demands	Infrastructure is elastic and scales automatically, expanding and contracting on-demand to service peak usage periods
Infrastructure is an capital expenditure that charges a fix amount regardless of usage patterns	Infrastructure is a operating expenditure that charges only for services when they are consumed

When your servers get sick, do you nurse them back to health or shoot them?

Tampereen yliopisto
Tampere University

Implications to developers

https://cloudrumblings.io/cloud-farm-pets-cattle-unicorns-and-horses-85271d915260



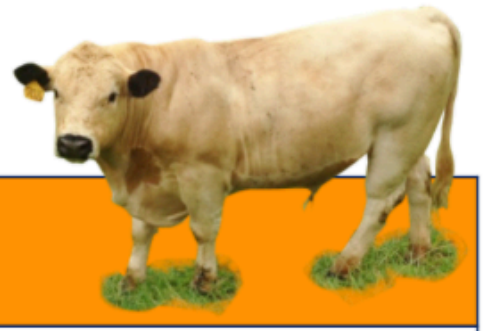
Pets	Cattle
Legacy Infrastructure	Cloud-Friendly Infrastructure
Pets are given names like grumpycat.petstore.com They are unique, lovingly hand raised and cared for When they get ill, you nurse them back to health	Cattle are given numbers like 10200713.cattlerancher.com They are almost identical to other cattle When they get ill, you replace them and get another
Infrastructure is a permanent fixture in the data center	Infrastructure is stateless, ephemeral, and transient
Infrastructure takes days to create, are serviced weekly, maintained for years, and requires migration projects to move	Infrastructure is instantiated, modified, destroyed and recreated in minutes from scratch using automated scripts
Infrastructure is modified during maintenance hours and generally requires special privileges such as root access	Infrastructure uses version-controlled scripts to modify any service without requiring root access or privileged logins
Infrastructure requires several different teams to coordinate and provision the full environment	Infrastructure is self-service with the ability to provision computing, network and storage services with a single click
Infrastructure is static, requiring excess capacity to be dormant for use during peak periods of demands	Infrastructure is elastic and scales automatically, expanding and contracting on-demand to service peak usage periods
Infrastructure is an capital expenditure that charges a fix amount regardless of usage patterns	Infrastructure is a operating expenditure that charges only for services when they are consumed

When your servers get sick, do you nurse them back to health or shoot them?

Tampereen yliopisto
Tampere University

Implications to developers

https://cloudrumblings.io/cloud-farm-pets-cattle-unicorns-and-horses-85271d915260



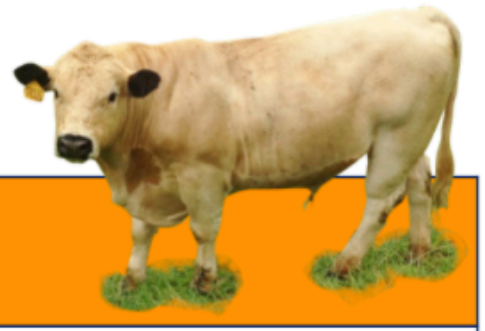
Pets	Cattle
Legacy Infrastructure	Cloud-Friendly Infrastructure
Pets are given names like grumpycat.petstore.com They are unique, lovingly hand raised and cared for When they get ill, you nurse them back to health	Cattle are given numbers like 10200713.cattlerancher.com They are almost identical to other cattle When they get ill, you replace them and get another
Infrastructure is a permanent fixture in the data center	Infrastructure is stateless, ephemeral, and transient
Infrastructure takes days to create, are serviced weekly, maintained for years, and requires migration projects to move	Infrastructure is instantiated, modified, destroyed and recreated in minutes from scratch using automated scripts
Infrastructure is modified during maintenance hours and generally requires special privileges such as root access	Infrastructure uses version-controlled scripts to modify any service without requiring root access or privileged logins
Infrastructure requires several different teams to coordinate and provision the full environment	Infrastructure is self-service with the ability to provision computing, network and storage services with a single click
Infrastructure is static, requiring excess capacity to be dormant for use during peak periods of demands	Infrastructure is elastic and scales automatically, expanding and contracting on-demand to service peak usage periods
Infrastructure is an capital expenditure that charges a fix amount regardless of usage patterns	Infrastructure is a operating expenditure that charges only for services when they are consumed

When your servers get sick, do you nurse them back to health or shoot them?

Tampereen yliopisto
Tampere University

Implications to developers

https://cloudrumblings.io/cloud-farm-pets-cattle-unicorns-and-horses-85271d915260



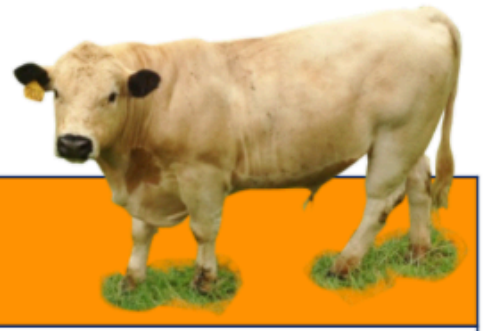
Pets Legacy Infrastructure	Cattle Cloud-Friendly Infrastructure
Pets are given names like grumpycat.petstore.com They are unique, lovingly hand raised and cared for When they get ill, you nurse them back to health	Cattle are given numbers like 10200713.cattlerancher.com They are almost identical to other cattle When they get ill, you replace them and get another
Infrastructure is a permanent fixture in the data center	Infrastructure is stateless, ephemeral, and transient
Infrastructure takes days to create, are serviced weekly, maintained for years, and requires migration projects to move	Infrastructure is instantiated, modified, destroyed and recreated in minutes from scratch using automated scripts
Infrastructure is modified during maintenance hours and generally requires special privileges such as root access	Infrastructure uses version-controlled scripts to modify any service without requiring root access or privileged logins
Infrastructure requires several different teams to coordinate and provision the full environment	Infrastructure is self-service with the ability to provision computing, network and storage services with a single click
Infrastructure is static, requiring excess capacity to be dormant for use during peak periods of demands	Infrastructure is elastic and scales automatically, expanding and contracting on-demand to service peak usage periods
Infrastructure is an capital expenditure that charges a fix amount regardless of usage patterns	Infrastructure is a operating expenditure that charges only for services when they are consumed

When your servers get sick, do you nurse them back to health or shoot them?

Tampereen yliopisto
Tampere University

Implications to developers

https://cloudrumblings.io/cloud-farm-pets-cattle-unicorns-and-horses-85271d915260



Pets Legacy Infrastructure	Cattle Cloud-Friendly Infrastructure
Pets are given names like grumpycat.petstore.com They are unique, lovingly hand raised and cared for When they get ill, you nurse them back to health	Cattle are given numbers like 10200713.cattlerancher.com They are almost identical to other cattle When they get ill, you replace them and get another
Infrastructure is a permanent fixture in the data center	Infrastructure is stateless, ephemeral, and transient
Infrastructure takes days to create, are serviced weekly, maintained for years, and requires migration projects to move	Infrastructure is instantiated, modified, destroyed and recreated in minutes from scratch using automated scripts
Infrastructure is modified during maintenance hours and generally requires special privileges such as root access	Infrastructure uses version-controlled scripts to modify any service without requiring root access or privileged logins
Infrastructure requires several different teams to coordinate and provision the full environment	Infrastructure is self-service with the ability to provision computing, network and storage services with a single click
Infrastructure is static, requiring excess capacity to be dormant for use during peak periods of demands	Infrastructure is elastic and scales automatically, expanding and contracting on-demand to service peak usage periods
Infrastructure is an capital expenditure that charges a fix amount regardless of usage patterns	Infrastructure is a operating expenditure that charges only for services when they are consumed

When your servers get sick, do you nurse them back to health or shoot them?

Pets are given names like grumpycat.petstore.com They are unique, lovingly hand raised and cared for When they get ill, you nurse them back to health	Cattle are given numbers like 10200713.cattlerancher.com They are almost identical to other cattle When they get ill, you replace them and get another
Infrastructure is a permanent fixture in the data center	Infrastructure is stateless, ephemeral, and transient
Infrastructure takes days to create, are serviced weekly, maintained for years, and requires migration projects to move	Infrastructure is instantiated, modified, destroyed and recreated in minutes from scratch using automated scripts
Infrastructure is modified during maintenance hours and generally requires special privileges such as root access	Infrastructure uses version-controlled scripts to modify any service without requiring root access or privileged logins
Infrastructure requires several different teams to coordinate and provision the full environment	Infrastructure is self-service with the ability to provision computing, network and storage services with a single click
Infrastructure is static, requiring excess capacity to be dormant for use during peak periods of demands	Infrastructure is elastic and scales automatically, expanding and contracting on-demand to service peak usage periods
Infrastructure is an capital expenditure that charges a fix amount regardless of usage patterns	Infrastructure is a operating expenditure that charges only for services when they are consumed

Tampereen yliopisto
Tampere University

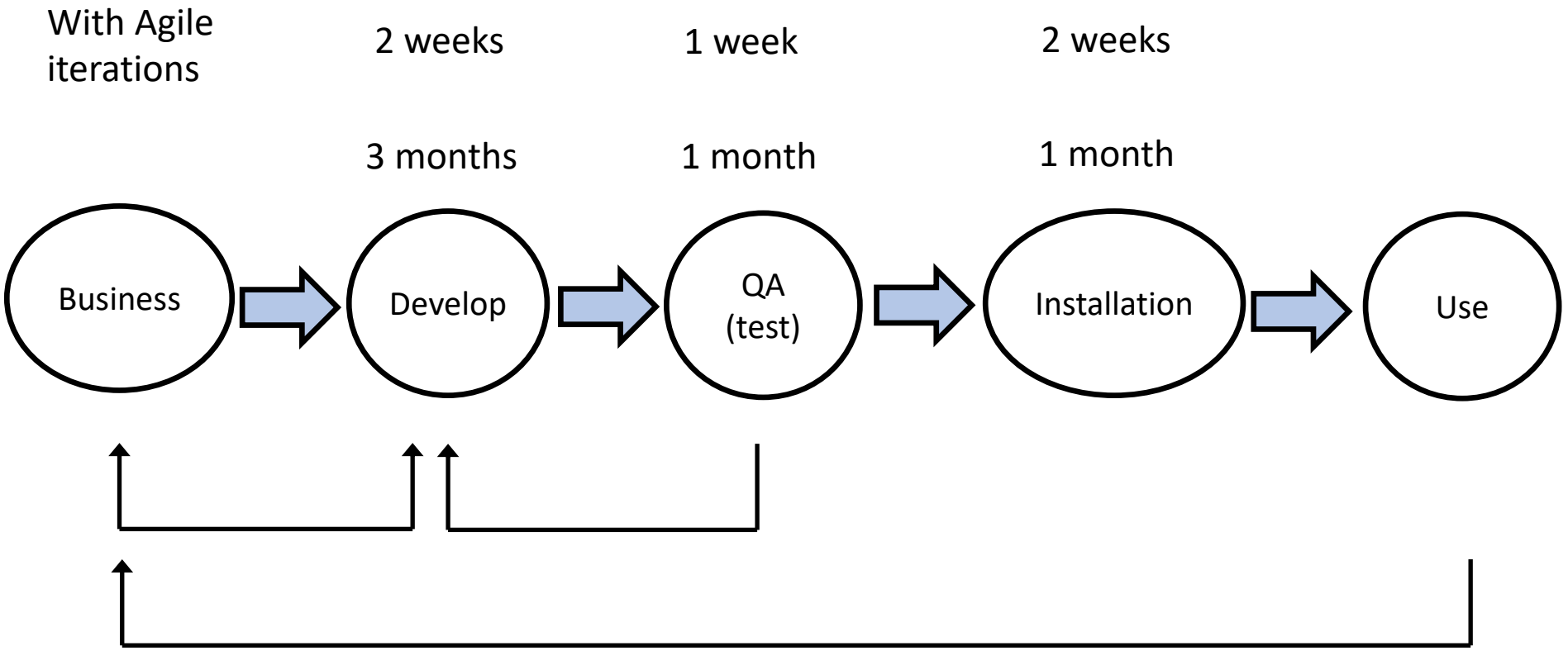
The example promised in the lectures

- https://blogs.tuni.fi/cs/research/cattle-instead-of-pets-end-of-carefully-crafted-software/

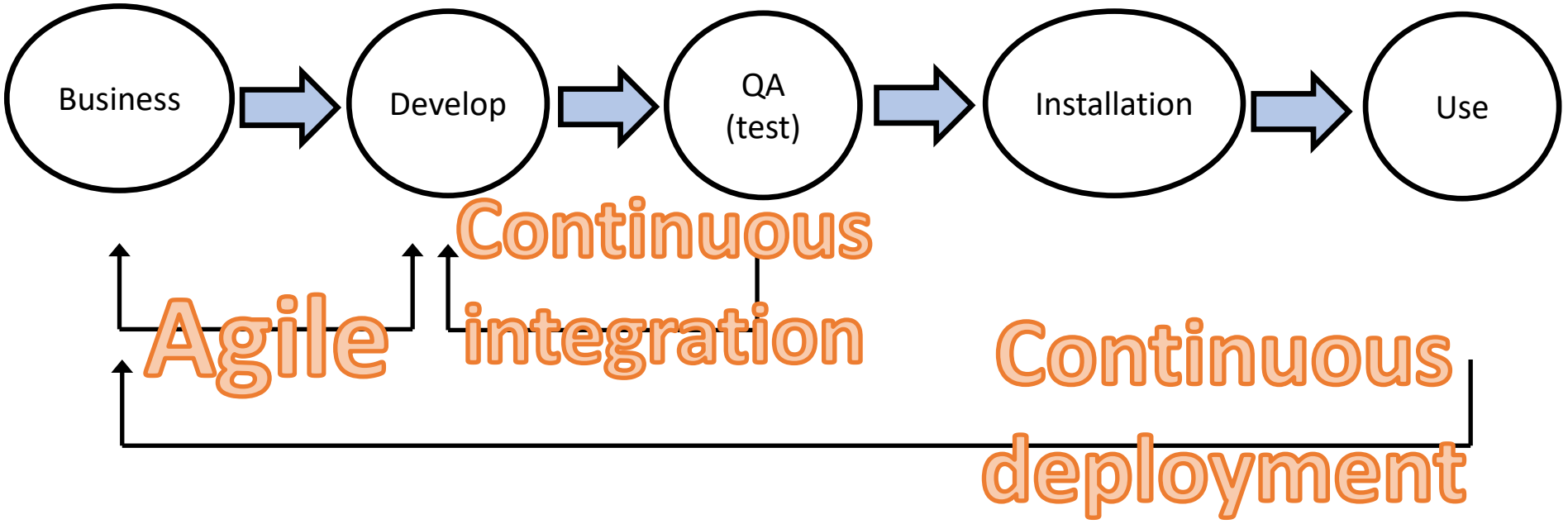
Tampereen yliopisto
Tampere University

Continuous Delivery and Deployment

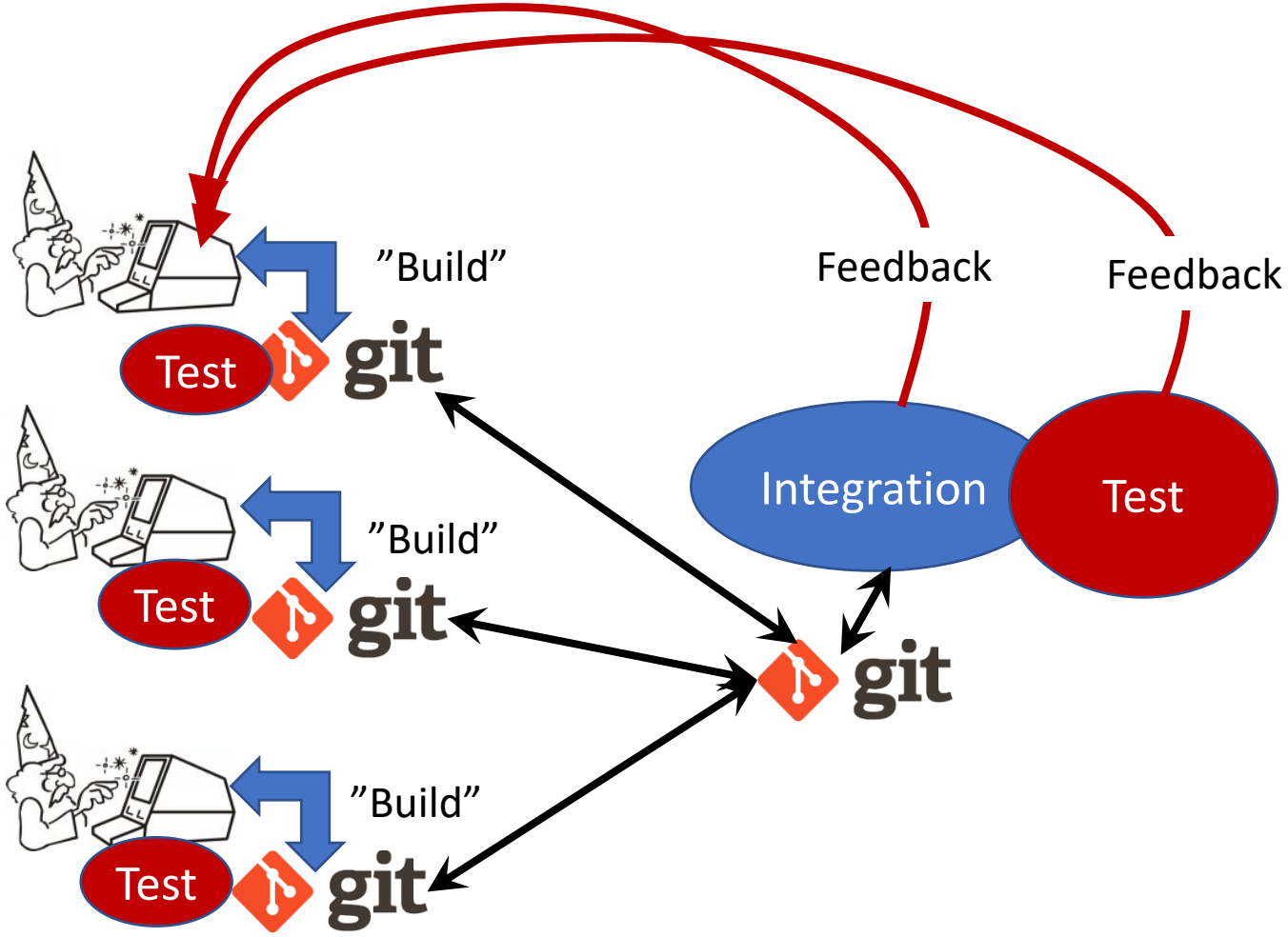
Feedback in traditional development
(Case: Internet-based service; based on slide by Antti Tirilä)



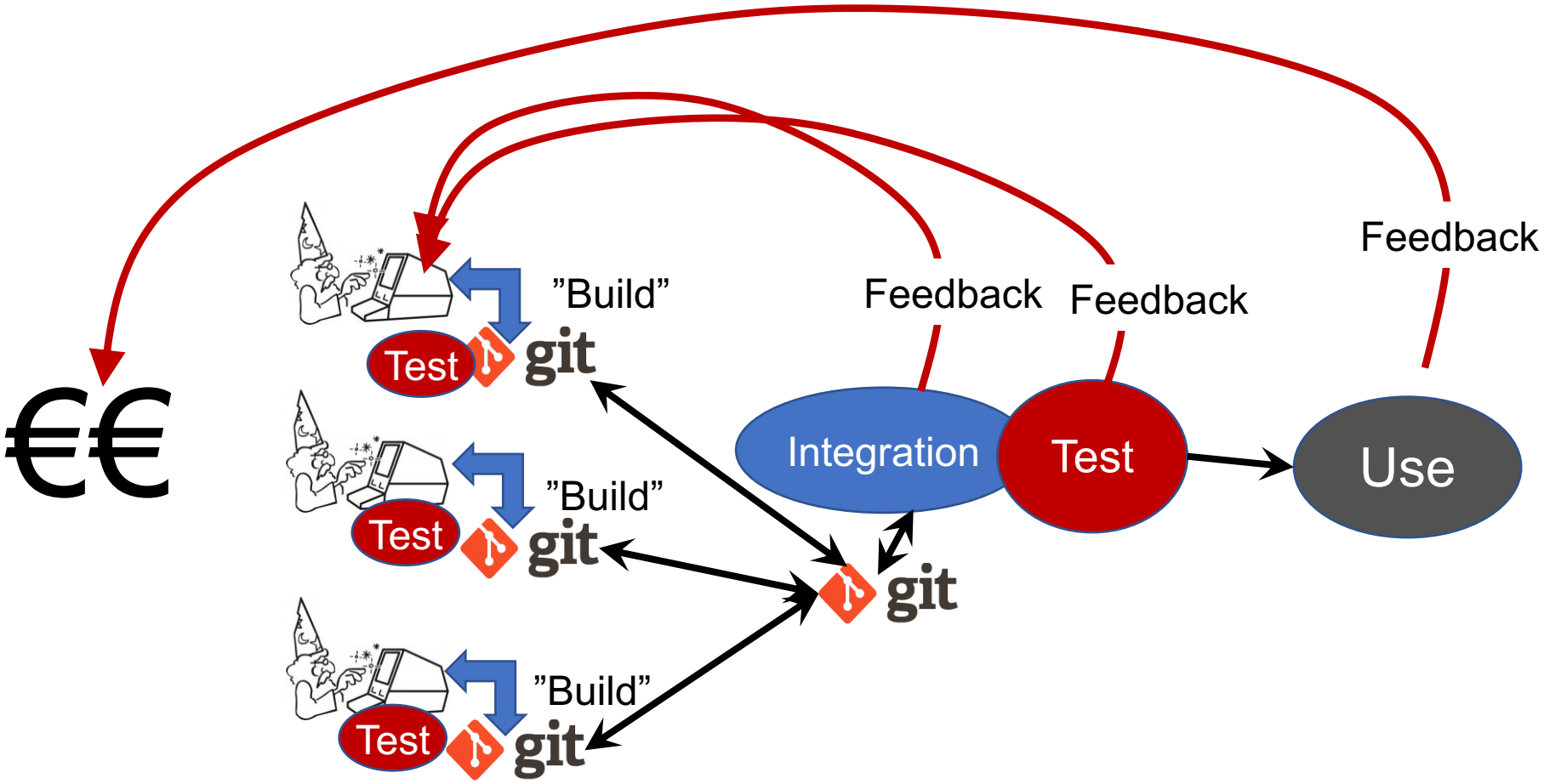
Feedback in traditional development (Case: Internet-based service; based on slide by Antti Tirilä)



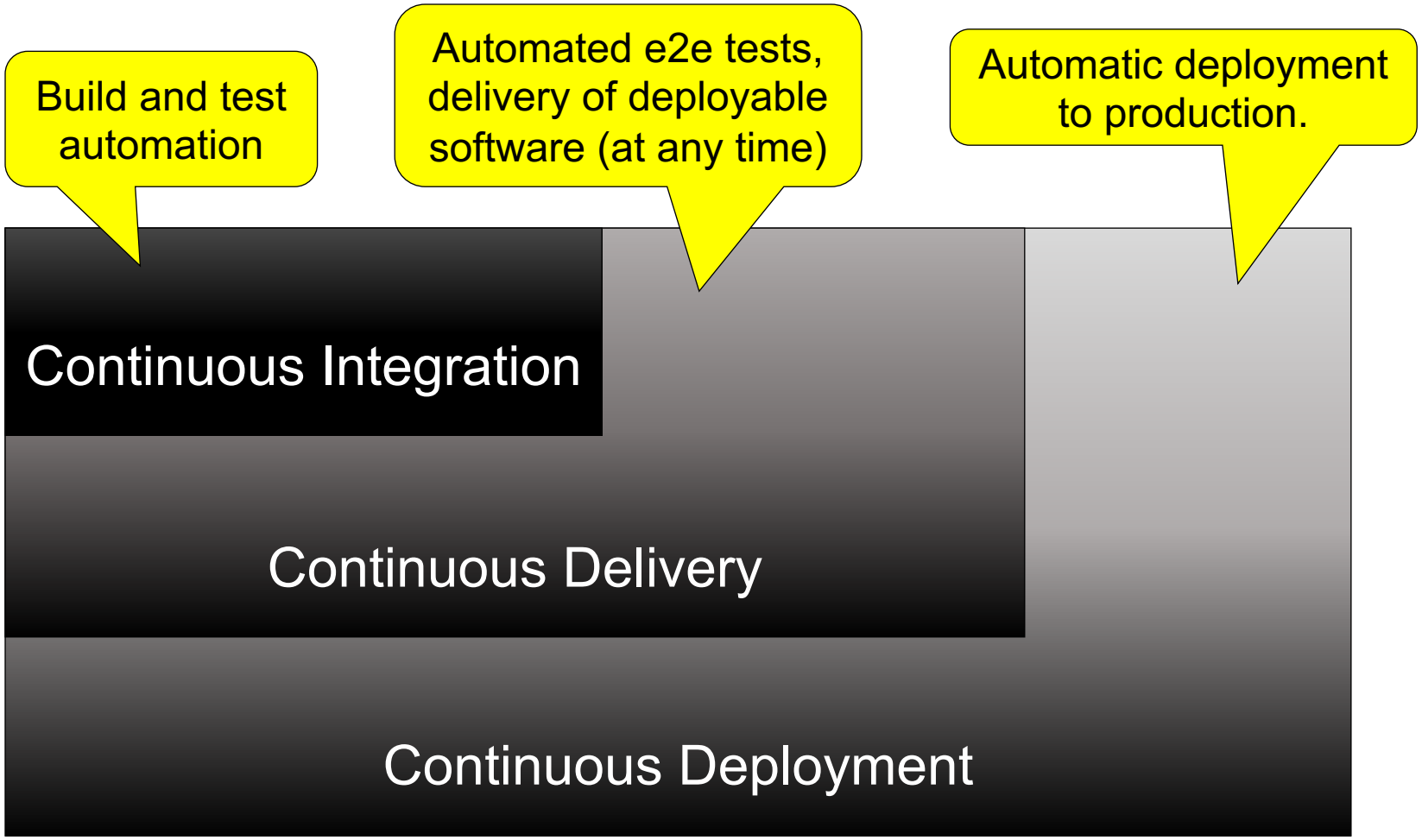
Continuous integration



Continuous deployment

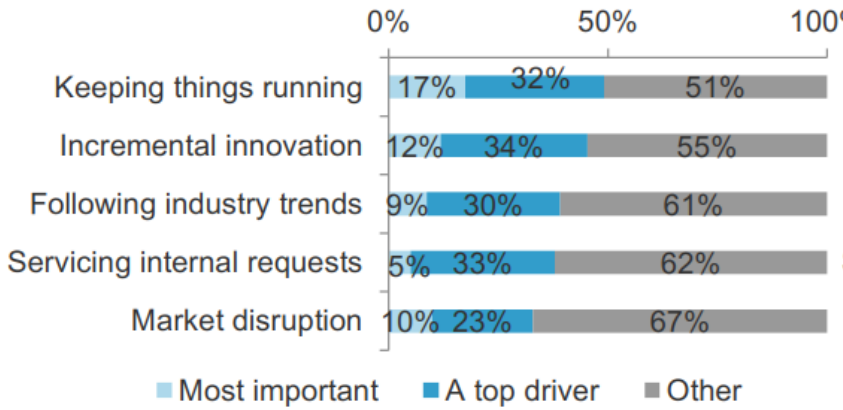


Continuous X

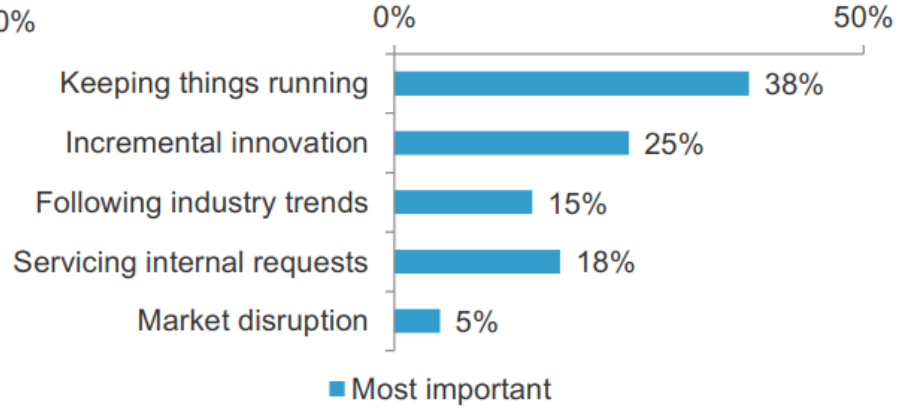


From Forrester report: Continuous Delivery: A Maturity Assessment Model: Building Competitive Advantage With Software Through A Continuous Delivery Process, 2013

"Please rank the importance of the business drivers behind your software development investments within your business area (current)."

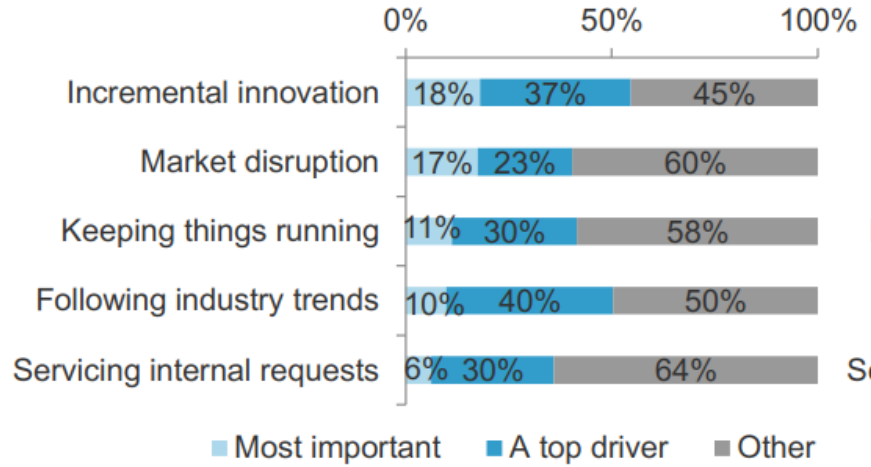


Base: 161 business decision-makers

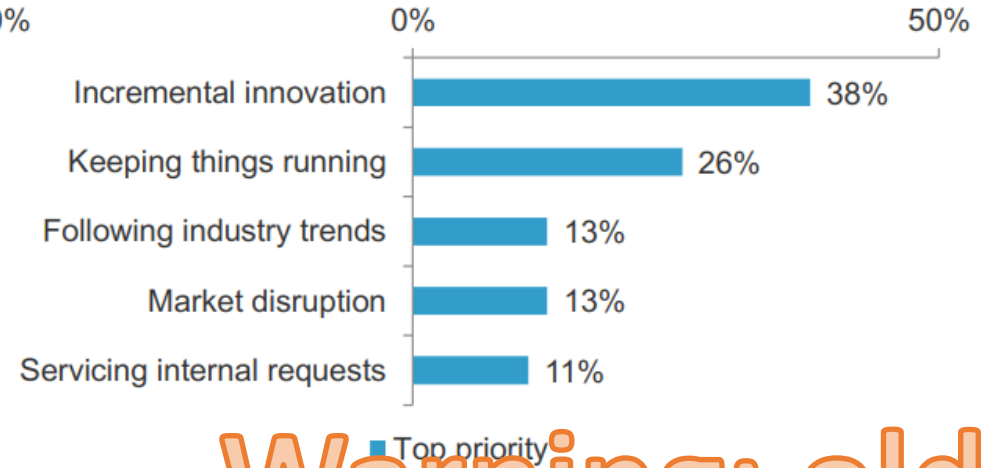


Base: 164 IT executives and managers

"Please rank the importance of the business drivers behind your software development investments within your business area (in two years)."



Base: 161 business decision-makers



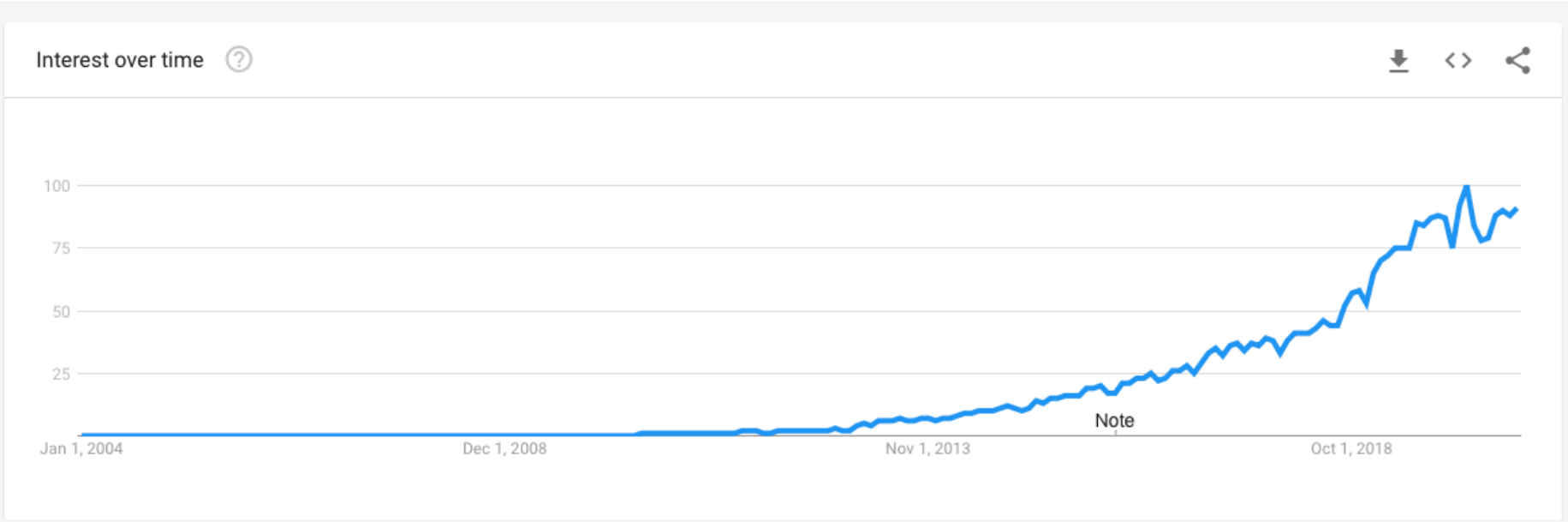
Base: 164 IT executives and managers

Source: A commissioned study conducted by Forrester Consulting on behalf of Thoughtworks, September 2012

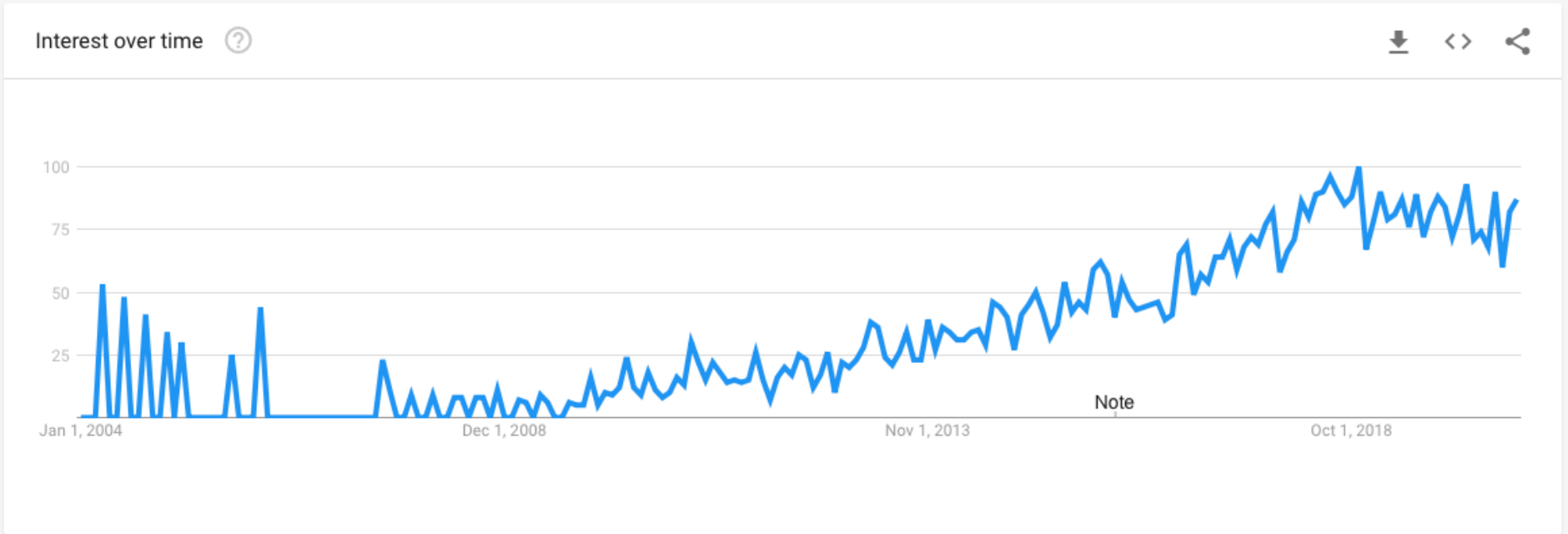
Warning: old stuff

Tampereen yliopisto
Tampere University

Google trends after that 2003



DevOps

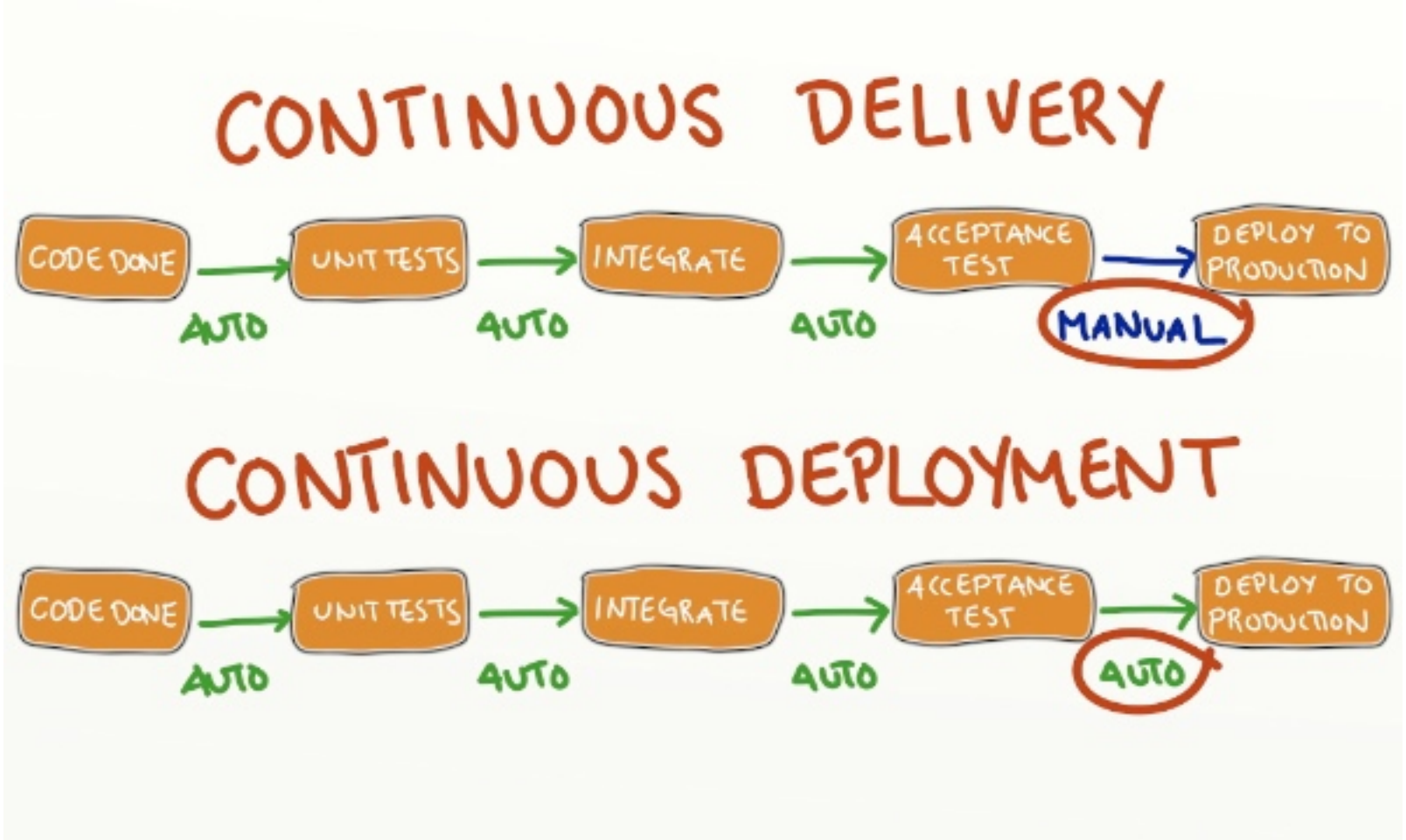


Continuous deployment

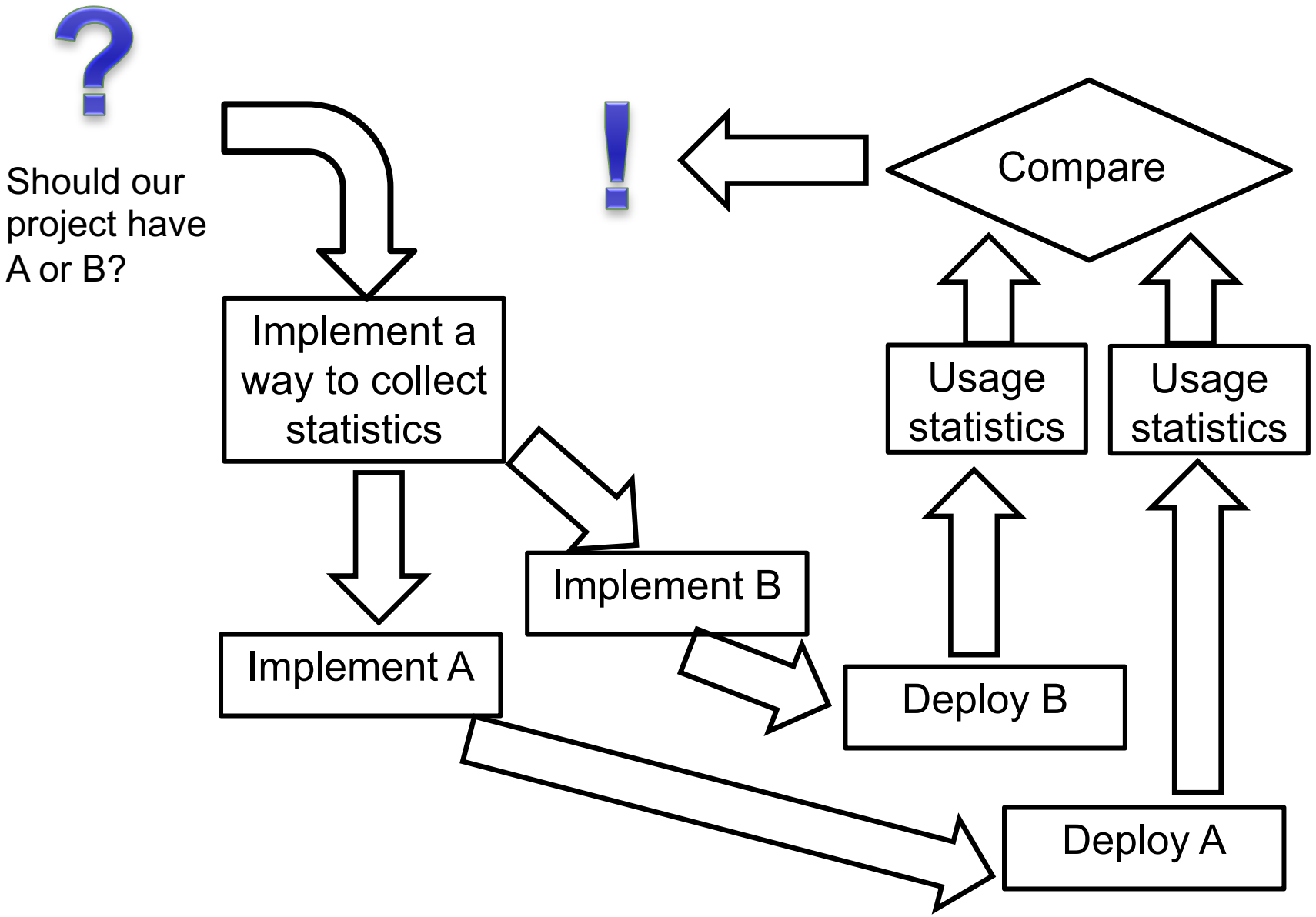
Tampereen yliopisto
Tampere University

Continuous delivery and deployment

(http://blog.crisp.se/2013/02/05/yassalsundman/continuous-delivery-vs-continuous-deployment)

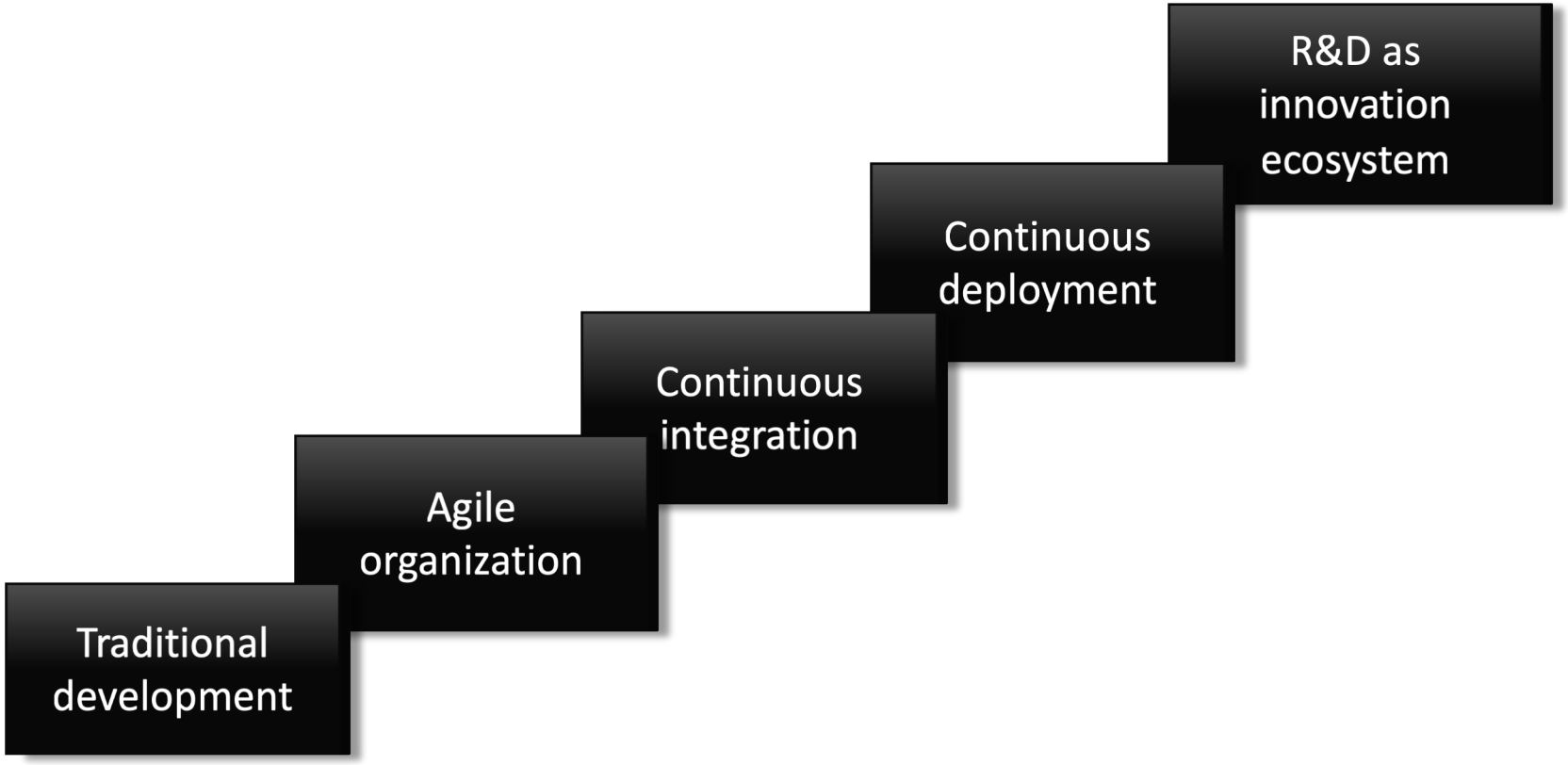


A/B Testing

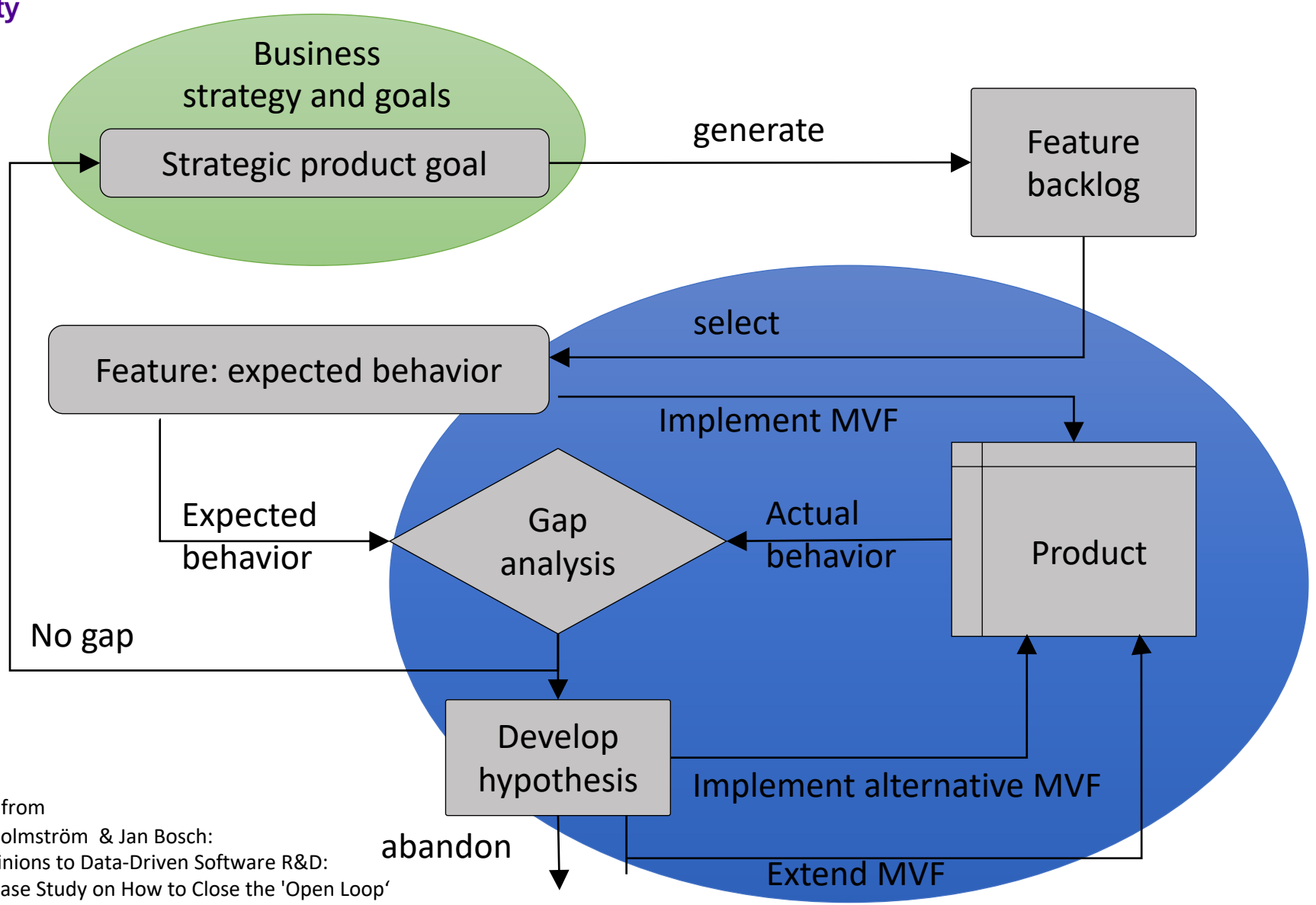

Should our project have A or B?
Implement a way to collect statistics
Implement A
Implement B
Deploy B
Deploy A
Usage statistics
Usage statistics
Compare

Tampereen yliopisto
Tampere University

Stairway to Heaven (As described by Jan Bosch)



The HYPEX model (**Hypothesis Experiment Data-Driven Development)**

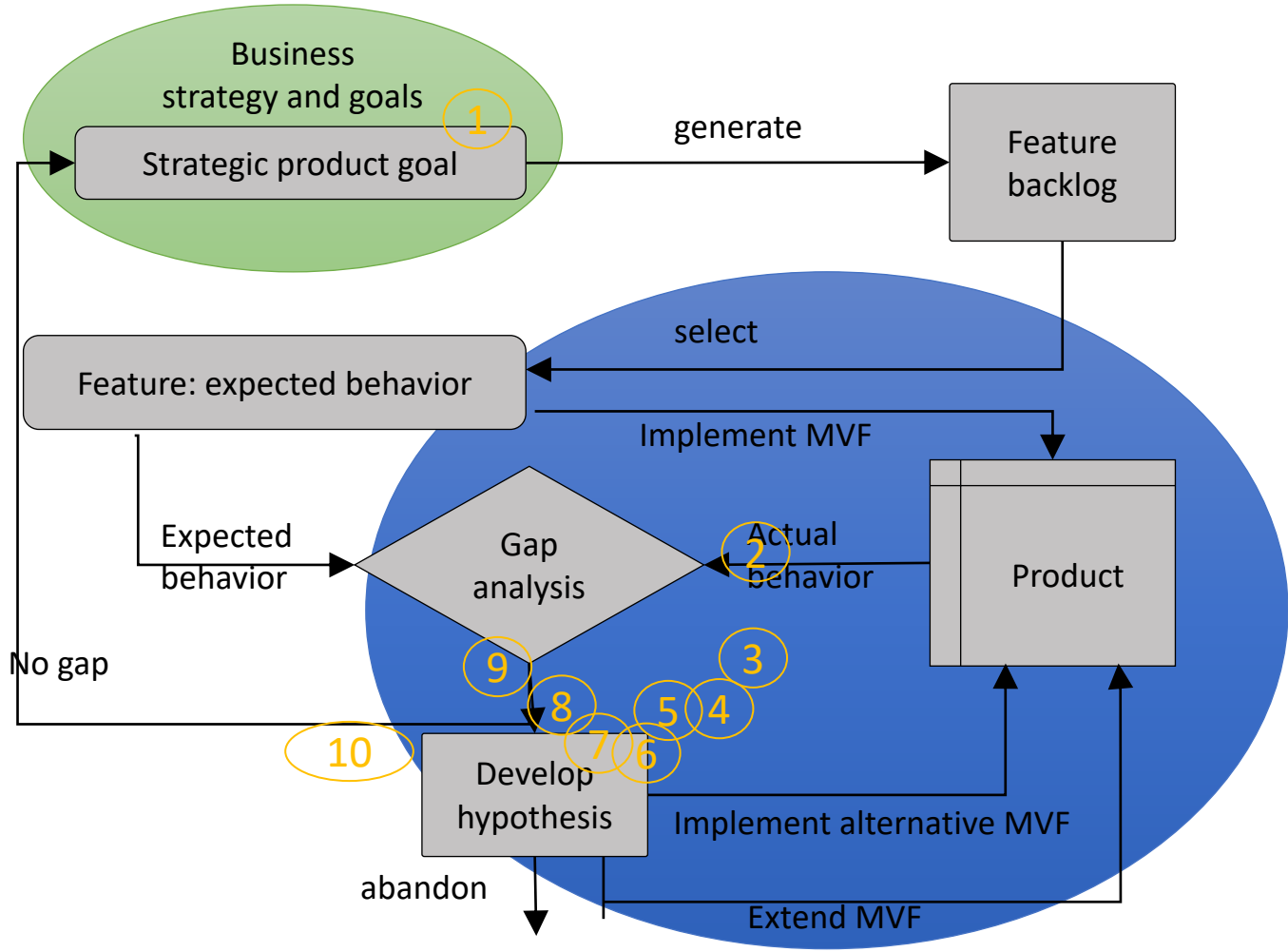


Adopted from
Helena Holmström & Jan Bosch:
From Opinions to Data-Driven Software R&D:
A Multi-case Study on How to Close the 'Open Loop‘
Problem

Tampereen yliopisto
Tampere University

Data-driven software development

1. Planning of the data collection
2. Deployment of data collection
3. Monitoring of the applications
4. Picking up the relevant data
5. Pre-processing – filtering and formatting – the data
6. Sending and/or saving the data
7. Cleaning and unification of the data
8. Storing the data
9. Visualizations and analysis
10. Decision making



Tampereen yliopisto
Tampere University

Main principles (https://continuousdelivery.com/principles/)

- Build quality in
- Work in small batches
- Computers perform repetitive tasks, people solve problems
- Relentlessly pursue continuous improvement
- Everyone is responsible

Sound familiar from somewhere?

Tampereen yliopisto
Tampere University

Reported HP case-study
(https://continuousdelivery.com/evidence-case-studies/)

They had three high-level goals:

- Create a single platform to support all devices
- Increase quality and reduce the amount of stabilization required prior to release
- Reduce the amount of time spent on planning

A key element in achieving these goals was implementing continuous delivery, with a particular focus on:

- The practice of continuous integration
- Significant investment in test automation
- Creating a hardware simulator so that tests could be run on a virtual platform
- Reproduction of test failures on developer workstations

Tampereen yliopisto
Tampere University

Reported HP case-study
(https://continuousdelivery.com/evidence-case-studies/)

They
- C
- I
- F

A
a p
- T
- S
- C
- F

Results:
- Overall development costs were reduced by ~40%.
- Programs under development increased by ~140%.
- Development costs per program went down 78%.
- Resources driving innovation increased eightfold.

Tampereen yliopisto
Tampere University

Let’s speculate the contribution of each

They had three high-level goals:

- Create a single platform to support all devices
- Increase quality and reduce the amount of stabilization required prior to release
- Reduce the amount of time spent on planning

A key element in achieving these goals was implementing continuous delivery, with a particular focus on:

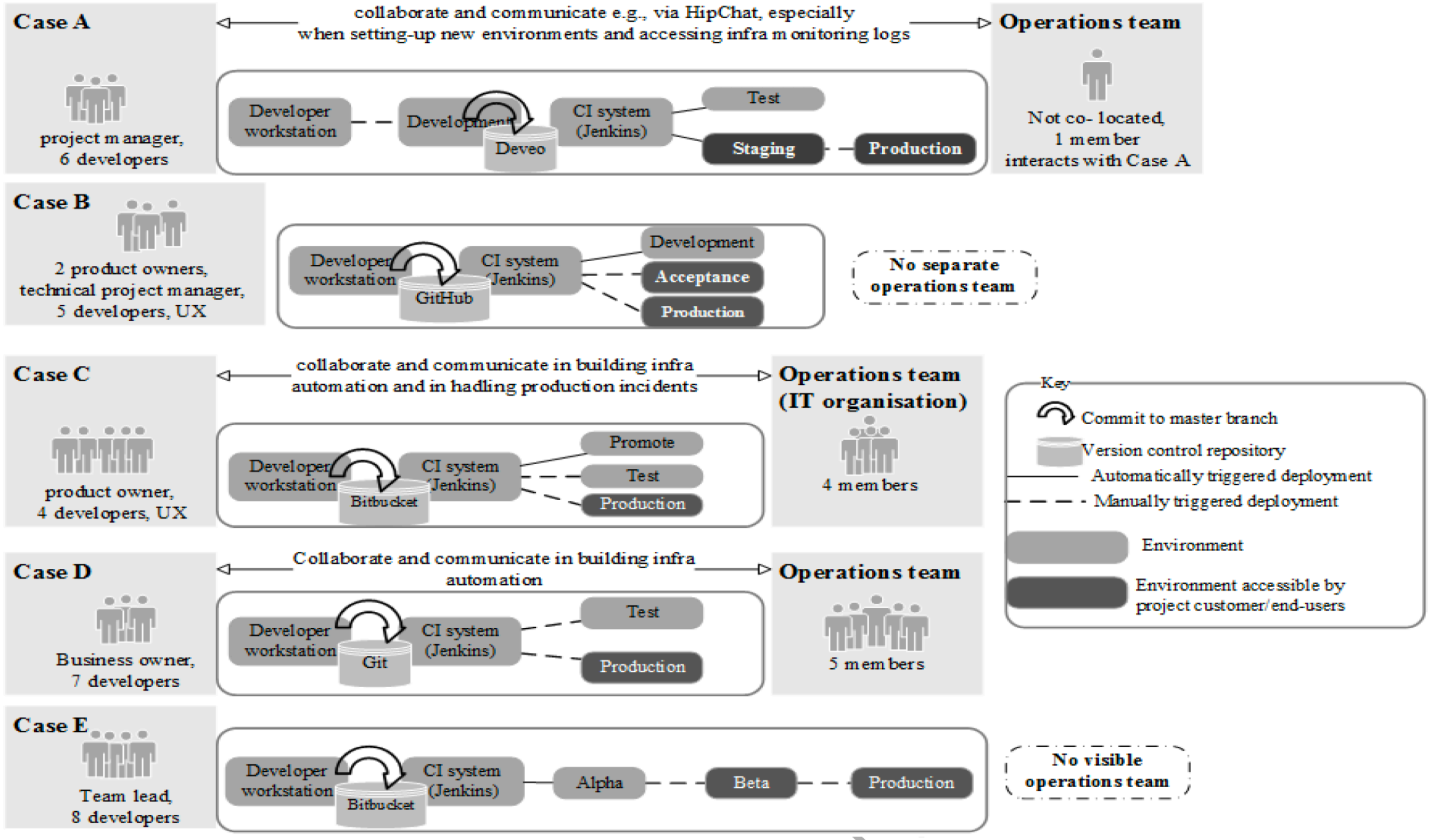
- The practice of continuous integration
- Significant investment in test automation
- Creating a hardware simulator so that tests could be run on a virtual platform
- Reproduction of test failures on developer workstations

Tampereen yliopisto
Tampere University

Couple of Finnish studies

Tampereen yliopisto
Tampere University

Lwakatare , Kilamo , Karvonen, Sauvola , Heikkilä, Itkonen, Kuvaja, Mikkonen, Oivo & Lassenius:
DevOps in practice : A multiple case study of five companies, Information and Software Technology , vol. 114 , pp. 217-230 . https://doi.org/10.1016/j.infsof.2019.06.010


Case A
project manager, 6 developers
collaborate and communicate e.g., via HipChat, especially when setting-up new environments and accessing infra monitoring logs
Operations team
Not co- located, 1 member interacts with Case A
Developer workstation
Development
Deveo
CI system (Jenkins)
Test
Staging
Production
Case B
2 product owners, technical project manager, 5 developers, UX
Developer workstation
GitHub
CI system (Jenkins)
Development
Acceptance
Production
No separate operations team
Case C
product owner, 4 developers, UX
collaborate and communicate in building infra automation and in hadling production incidents
Operations team (IT organisation)
4 members
Developer workstation
Bitbucket
CI system (Jenkins)
Promote
Test
Production
Case D
Business owner, 7 developers
Collaborate and communicate in building infra automation
Operations team
5 members
Developer workstation
Git
CI system (Jenkins)
Test
Production
Case E
Team lead, 8 developers
Developer workstation
Bitbucket
CI system (Jenkins)
Alpha
Beta
Production
No visible operations team
Key
Commit to master branch
Version control repository
Automatically triggered deployment
Manually triggered deployment
Environment
Environment accessible by project customer/end-users

Tampereen yliopisto
Tampere University

Perceived benefits

- **Improved delivery speed** of software changes Improved speed in the development and deployment of software changes to production environment.
- **Improved productivity in operations** work. Decreased communication problems, bureaucracy, waiting overhead due to removal of manual deployment hand-offs and organisational boundaries; Lowered human error in deployment due to automation and making explicit knowledge of operation-related tasks to software development
- **Improvements in quality**. Increased confidence in deployments and reduction of deployment risk and stress; Improved code quality; Improved product value to customer resulting from production feedback about users and usage.
- **Improvements in organisational-wide culture** and mind-set. Enrichment and wider dissemination of DevOps in the company through discussions and dedicated training groups ‘communities of practice’

Tampereen yliopisto
Tampere University

Perceived challenges

- Insufficiencies in infrastructure automation
- High demand for skills and knowledge
- Project and resource constraints
- Difficulties in monitoring, especially for microservice-based applications and in determining useful metrics
- Difficulties in determining a right balance between the speed of new functionality and quality.

Tampereen yliopisto
Tampere University

Summary of the findings

(i) software development team attaining ownership and responsibility to deploy software changes in production is crucial in DevOps.

(ii) toolchain usage and support in deployment pipeline activities accelerates the delivery of software changes, bug fixes and handling of production incidents. (ii) the delivery speed to production is affected by context factors, such as manual approvals by the product owner

(iii) steep learning curve for new skills is experienced by both software developers and operations staff, who also have to cope with working under pressure.

Tampereen yliopisto
Tampere University

Leppänen, Mäkinen, Pagels, Eloranta, Itkonen, Mäntylä, Männistö
The highways and country roads to continuous deployment,
IEEE Software, vol. 32, no. 2, pp. 64-72, Mar.-Apr. 2015.
doi: 10.1109/MS.2015.50

” Interviews with 15 information and communications technology companies revealed the benefits of and obstacles to continuous deployment. Despite understanding the benefits, none of the companies adopted a fully automatic deployment pipeline.”

Tampereen yliopisto
Tampere University

State of the practice (2014)

- Only one company had completely automatic pipeline to deployable product; no one really to production
- Fastest time from code change to production
  - 5min – 4 weeks
    (for web application developers longest time was 1 day)
- Cycle-time to potentially deployable software
  - 20min – 1 months
- Full deployment cycle
  - 1 hour – 1.5 years

Tampereen yliopisto
Tampere University

Perceived benefits 1/2

- Faster feedback
  - to development
  - From users to decision making
- More Frequent Releases
  - ” less waste because the features weren’t waiting in the development pipeline to be released.”
- Improved Quality and Productivity
  - robust automated deployment with a comprehensive test suite
  - reduced scope for each release

Tampereen yliopisto
Tampere University

Perceived benefits 2/2

- Improved Customer Satisfaction
  - new product features provided better customer service
  - (reported by 5 out of 15 interviewed organiations)
- Effort Savings
  - three interviewees reported
  - automation saved time
- Closer Connection between Development and Operations
  - only one reported !

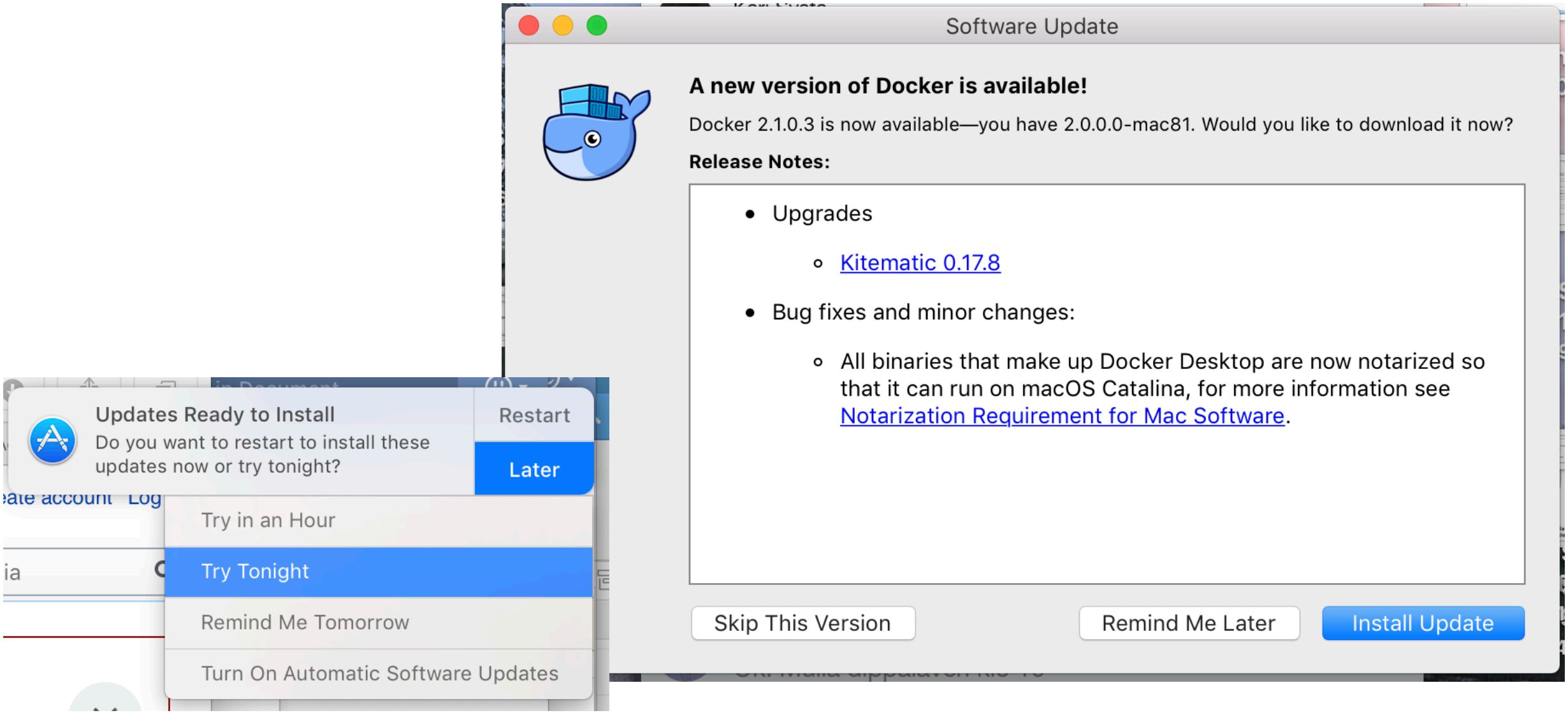
Tampereen yliopisto
Tampere University

Obstacles 1/2

- Resistance to Change
  - Organization culture, management, social relations, …
- Customer Preferences
  - Might be reluctant to deal with more frequent releases
- Domain Constraints
  - Telecom, Medical, Embedded, …
  - Distribution channels
- Developer Trust and Confidence
  - Proficiency and knowledge of typical continuous-deployment practices
  - Reliable automated testing (… even browser-bases apps)

Tampereen yliopisto
Tampere University

About resistance



Tampereen yliopisto
Tampere University

Obstacles 2/2

- Legacy Code Considerations
  - Quality has decreased over time
  - Not be designed to be automatically tested
- Duration, Size, and Structure
  - Effort to create the pipe-line and tests is big
  - In big projects the execution of tests will also take time
- Different Development and Production Environments
  - Especially ”embedded”
- Manual and Nonfunctional Testing