

TIE-23536 Cloud Applications

Lecture 1: introduction

Kari Systä
30.08.2019

Today's content

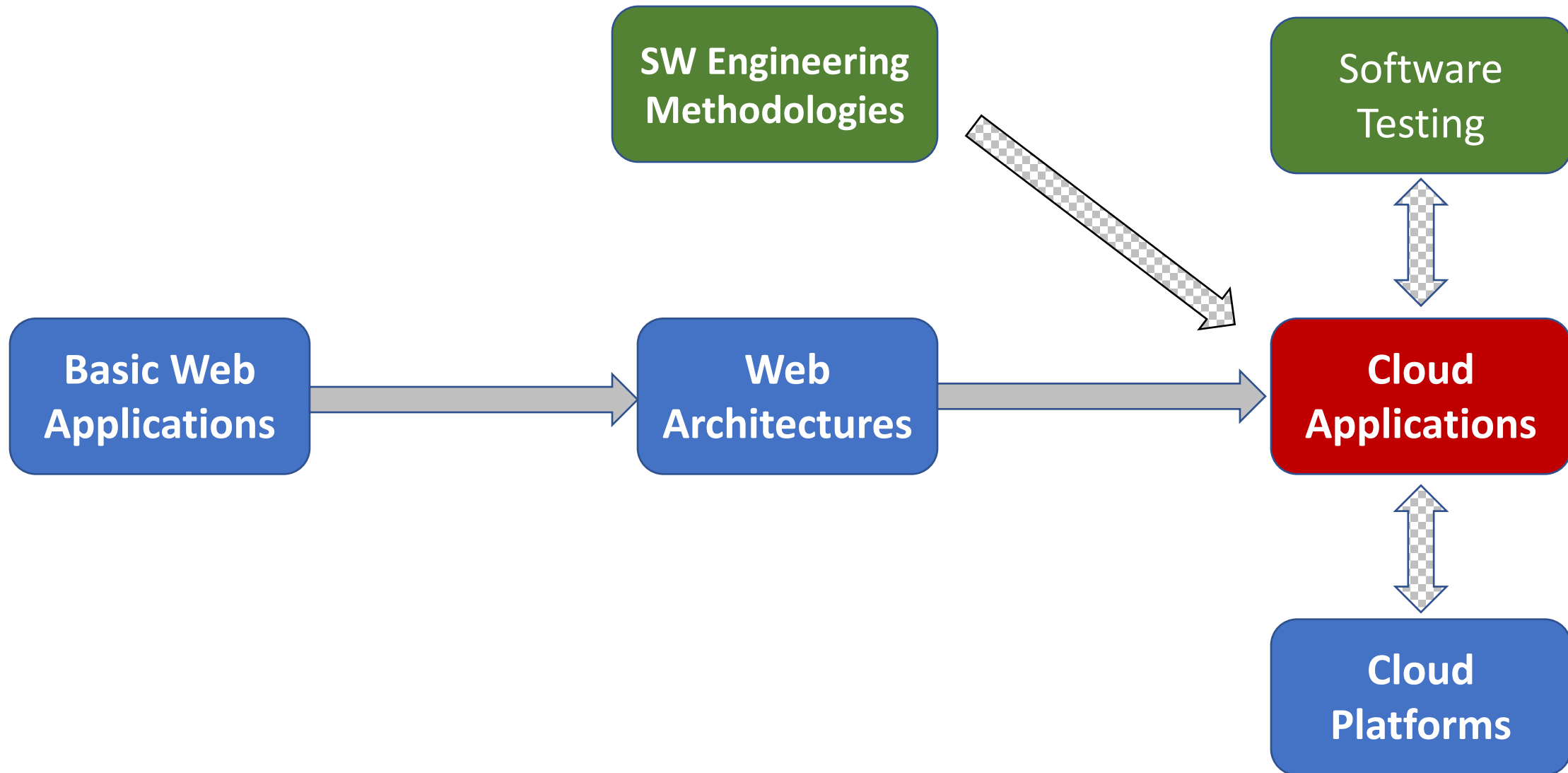
- Introduction to the course
- Introduction to DevOps
- Next steps in the course

Student background

- Passed **TIE-23526 Web Architectures ?**
 - Passed **TIE-23546 Cloud Platforms ?**
 - Now starting **TIE-23546 Cloud Platforms ?**
 - Passed **TIE-21107 Software Engineering Methodologies ?**
 - Passed **TIE-21201 Software Testing ?**
-
- Practical experience with Docker?
 - Practical experience with Kubernetes?
 - Practical experience with AWS?
 - Practical experience with Azure Cloud?

What is this course about

- How to design, implement, deploy and operate cloud applications.
- So this is a DevOps course
- A lot about automation of the above



Rought plan – subject to change

- 30.08: Intro to the course and DevOps
- 06.09: Recap (from earlier courses) on virtualization technologies;
- 13.09: Cloud and scalability
- 20.09: Continuous Deployment – what & why
- 27.09: Continuous Deployment – tools and techniques
- 04.10: Security issues; recap on internet networking (from earlier courses)
- 11.10: Guest lecture on AWS
- 23.10: Issues on cloud-SW: isolation, dependency management etc,
- 30.10: Cloud-native architectures: microservices
- 06.11: Cloud-native architectures: FaaS/serverless
- 13.11: About automation: testing and beyond
- 20.11: Hosting and deployment issues
- 27.11: Guest lecture on Kubernetes
- 04.12: Recap

06.09: Recap (from earlier courses) on virtualization technologies;

- What and why
- Hypervisor vs containers?
- Technology examples
 - Vagrant
 - Docker

13.09: Cloud and scalability

- Technical and business perspectives
- Technologies

20.09: Continuous Deployment – what & why

- Software engineering and business perspectives

27.09: Continuous Deployment – tools and techniques

- Design and implementation of the pipeline
- Technology examples with the xenialab periodic table

1OsGI
GitLab

2EnSp
Splunk

3FmGh
GitHub

4EnDt
Datical

5EnXlr
XebiaLabs
XL Release

6FmAws
AWS

7PdAz
Azure

8EnGc
Google Cloud

9EnOp
OpenShift

10FmSl
Sumo Logic

11OsSv
Subversion

12EnDb
DBMaestro

13OsDk
Docker

14EnUr
UrbanCode
Release

15PdAf
Azure
Functions

16PdLd
Lambda

17EnIc
IBM Cloud

18OsFd
Fluentd

19EnCw
ISPW

20EnDp
Delphix

21OsJn
Jenkins

22FmCs
Codeship

23OsFn
FitNesse

24FrJu
JUnit

25FrKa
Karma

26OsSu
SoapUI

27EnCh
Chef

28FrTf
Terraform

29EnXld
XebiaLabs
XL Deploy

30EnUd
UrbanCode
Deploy

31OsKu
Kubernetes

32FmCc
CA CD
Director

33EnPr
Plutora
Release

34PdAl
Alibaba Cloud

35OsOs
OpenStack

36OsPs
Prometheus

37OsAt
Artifactory

38EnRg
Redgate

39PdBa
Bamboo

40FmVs
VSTS

41FrSe
Selenium

42FrJm
JMeter

43OsJa
Jasmine

44PdSl
Sauce Labs

45OsAn
Ansible

46OsRu
Rudder

47EnOc
Octopus
Deploy

48OsGo
GoCD

49OsMs
Mesos

50PdGke
GKE

51FmOm
OpenMake

52PdCp
AWS
CodePipeline

53OsCy
Cloud
Foundry

54EnIt
ITRS

55OsNx
Nexus

56OsFw
Flyway

57OsTr
Travis CI

58FmTc
TeamCity

59OsGa
Gatling

60FrTn
TestNG

61FmTt
Tricentis
Tosca

62PdPe
Perfecto

63EnPu
Puppet

64OsPa
Packer

65FmCd
AWS
CodeDeploy

66EnEc
ElectricCloud

67OsRa
Rancher

68PdAks
AKS

69OsRk
Rkt

70OsSp
Spinnaker

71PdIr
Iron.io

72PdMg
Moogsoft

73FmBb
BitBucket

74EnPf
Perforce
HelixCore

75FmCr
Circle CI

76PdCb
AWS
CodeBuild

77FrCu
Cucumber

78OsMc
Mocha

79OsLo
Locust.io

80EnMf
Micro Focus
UFT

81OsSl
Salt

82OsCe
CFEngine

83EnEb
ElasticBox

84EnCa
CA Automate

85EnDe
Docker
Enterprise

86PdAe
AWS ECS

87FmCf
Codefresh

88OsHm
Helm

89OsAw
Apache
OpenWhisk

90OsLs
Logstash

Os

Fr

Fm

Pd

En

Open Source

Free

Freemium

Paid

Enterprise

Source Control Mgmt.

Database Automation

Continuous Integration

Testing

Configuration

Deployment

Containers

Release Orchestration

Cloud

AIOps

Analytics

Monitoring

Security

Collaboration

EMBED

DOWNLOAD

91 En Xli XebiaLabs XL Impact	92 Os Ki Kibana	93 Fm Nr New Relic	94 En Dt Dynatrace	95 En Dd Datadog	96 Fm Ad AppDynamics	97 Os El ElasticSearch	98 Os Ni Nagios	99 Os Zb Zabbix	100 En Zn Zenoss	101 En Cx Checkmarx SAST	102 En Sg Signal Sciences	103 En Bd BlackDuck	104 Os Sr SonarQube	105 Os Hv HashiCorp Vault
106 En Sw ServiceNow	107 Pd Jr Jira	108 Fm Tl Trello	109 Fm Sl Slack	110 Fm St Stride	111 En Cn CollabNet VersionOne	112 En Ry Remedy	113 En Ac Agile Central	114 Pd Og OpsGenie	115 Pd Pd Pagerduty	116 Os Sn Snort	117 Os Tw Tripwire	118 En Ck CyberArk Conjur	119 En Vc Veracode	120 En Ff Fortify SCA

04.10: Security issues; recap on internet networking (from earlier courses)

- A lot of misc. Stuff
- Technology example
 - NGINX

11.10: Guest lecture on AWS

23.10: Issues on cloud-SW: isolation, dependency management etc,

- First scare the students with the complexity
- Then, give some solutions

30.10: Cloud-native architectures: microservices

- OO vs SOA
- Traditional SOA vs microservices
- Typical solutions
 - API gateway
 - Message bus

06.11: Cloud-native architectures: FaaS/serverless

- What and why

13.11: About automation: testing and beyond

- What and why to automate
- Research directions

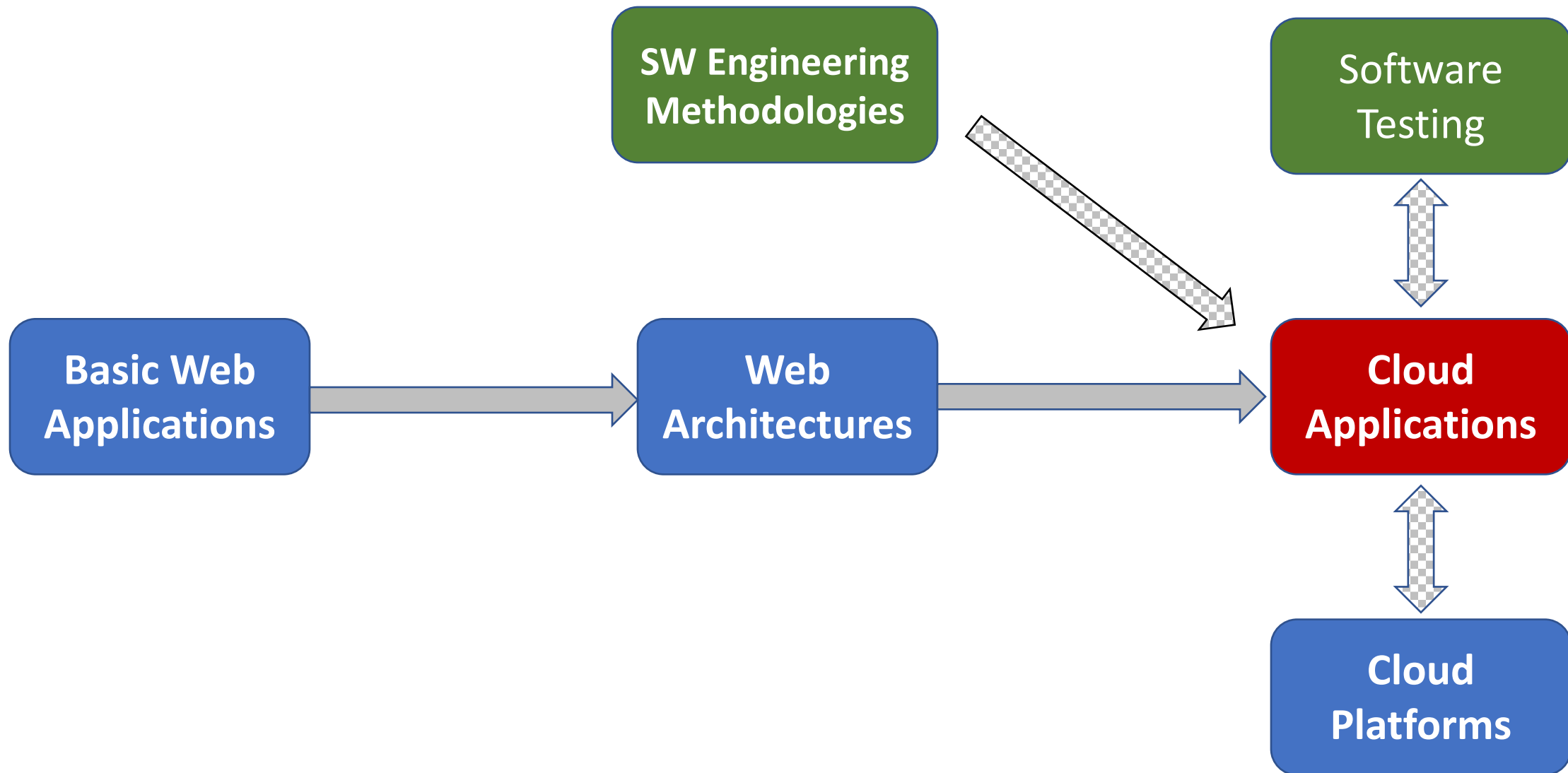
20.11: Hosting and deployment issues

- Focus on continuity – systems should not break when updated
- Technology examples
 - Blue-green
 - Canary

27.11: Guest lecture on Kubernetes

- A tool that is currently widely used

04.12: Recap



You

- Automation: 4
- Doctoral school: 9
- Exchange students: 7
- Electrical engineering: 3
- Science and engin.: 2
- Information techn.: 51
- TSK: 23
- TTK: 2
- Other: 6

What do I expect from your background

- 10-20 credits of software engineering after the first two programming courses
- Programming routine; typically in this phase you should be able to pick up a new programming language without specific teaching
- Basic knowledge of Linux/Unix operating system; command shell, file system etc.
- Some experience with virtualization, containers and docker.

The bad news

- This is a new course – that we are just ramping up
- We have a serious lack of teaching staff
- Unless some miracle happens we are forced to limit the number of participants.
- Priority is give to students who are
 - At least on master-level (do you BSc first!)
 - Major is software engineering

Passing requirements

- Exam
 - Electronic
- Project
 - Details will be published in couple of weeks
- A few compulsory on-line exercises
- Some face-to-face sessions – mainly to help

Course logistics

- Lectures
 - Fridays 10-12 in SA207
- Weekly/on-line (“alone”) exercises
 - Wednesdays 14-16 in SJ204
 - Fridays 14-16 in SJ204
 - Physical ones will not start before 11.09

Project

- In which language?
 - YAML
 - JavaScript, Python, Golang...
- The main part is building of pipeline

Course material

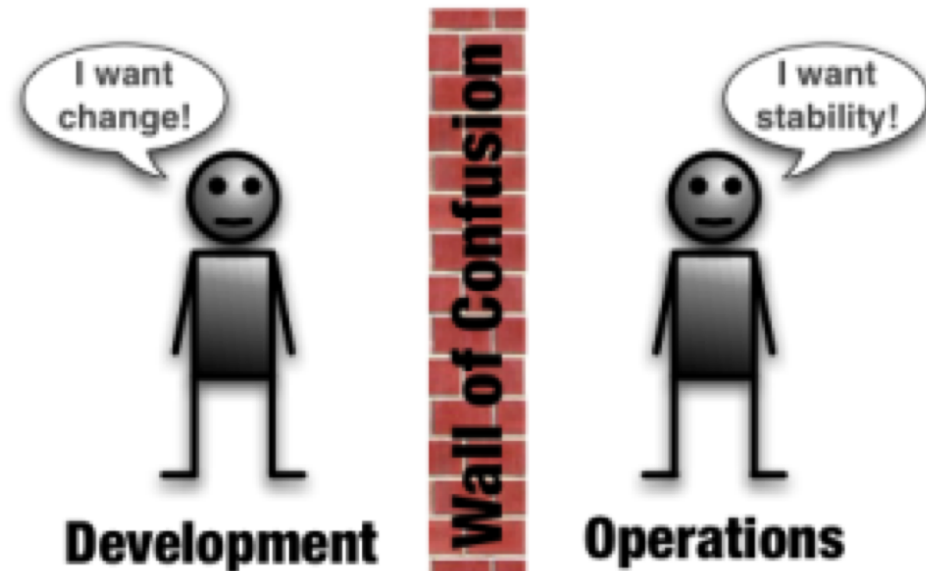
- Will be collected to
 - <https://plus.tuni.fi/tie-23536/autumn-2019/>
- Examples of recommended reading
 - Humble, Farley: Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Addison-Wesley Signature Series)
 - Summary part of “Lwakatare, Lucy Ellen: DevOps adoption and implementation in software development practice : concept, practices, benefits and challenges, ”, <http://urn.fi/urn:isbn:9789526217116>

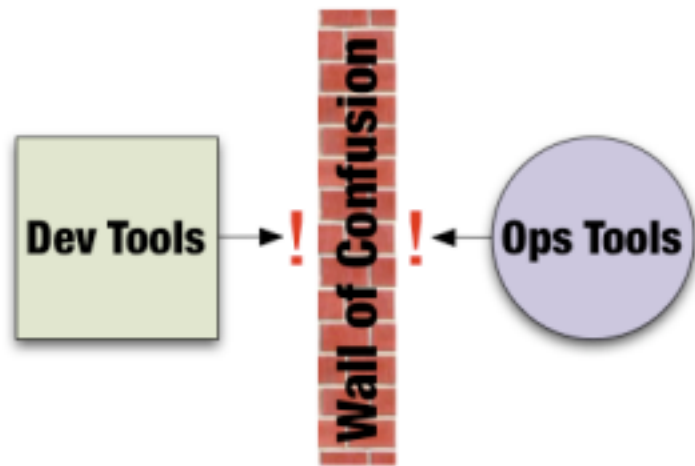
DevOps

DevOps – where it started

(<http://dev2ops.org/2010/02/what-is-devops/>)

- DevOps is a response to the growing awareness that there is a disconnect between what is traditionally considered development activity and what is traditionally considered operations activity. This disconnect often manifests itself as conflict and inefficiency.
- Wall of confusion

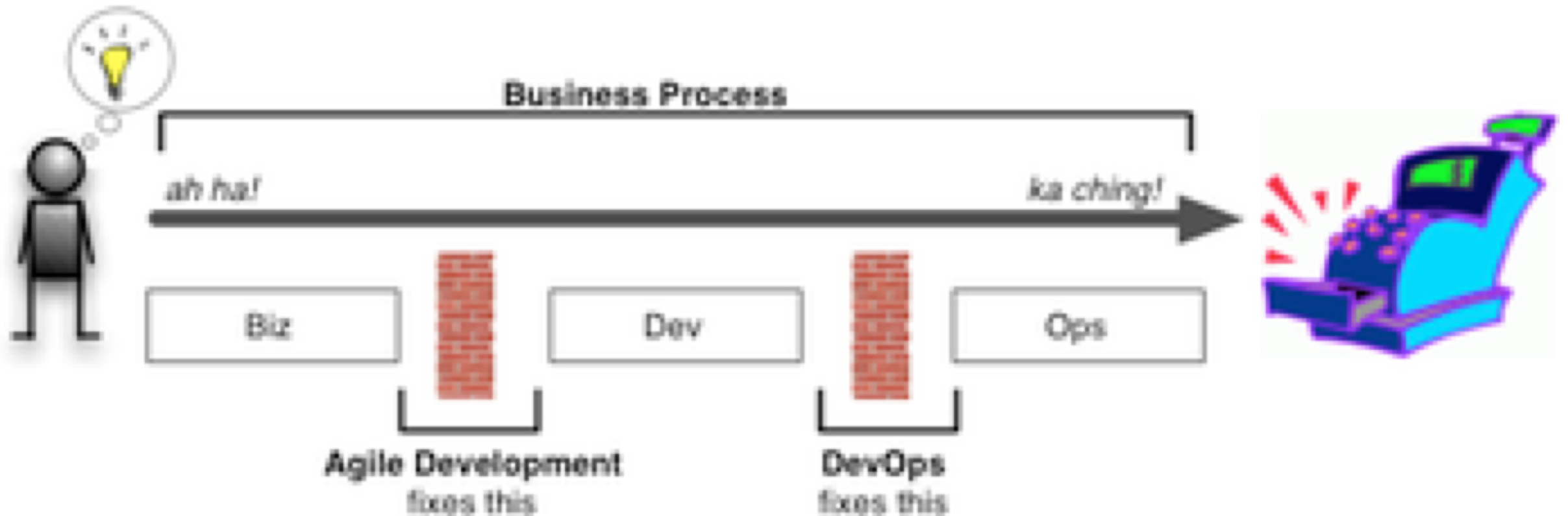




=



The lifecycle



What is DevOps

(there are several definitions)

- Lucy Lwakatare:
 - DevOps is a concept that embodies a **cultural and mindset change** that is substantiated with a **set of practices** to encourage **cross-disciplinary collaboration between software development and IT operations** within a software company. The main purpose for the collaboration is to enable the **fast release of quality software changes** while simultaneously **operating resilient systems**.
 - From a **socio-technical perspective**, DevOps practices are focused on the **automation practices** of software deployment and infrastructure management, specifically automation of configuration management and monitoring.

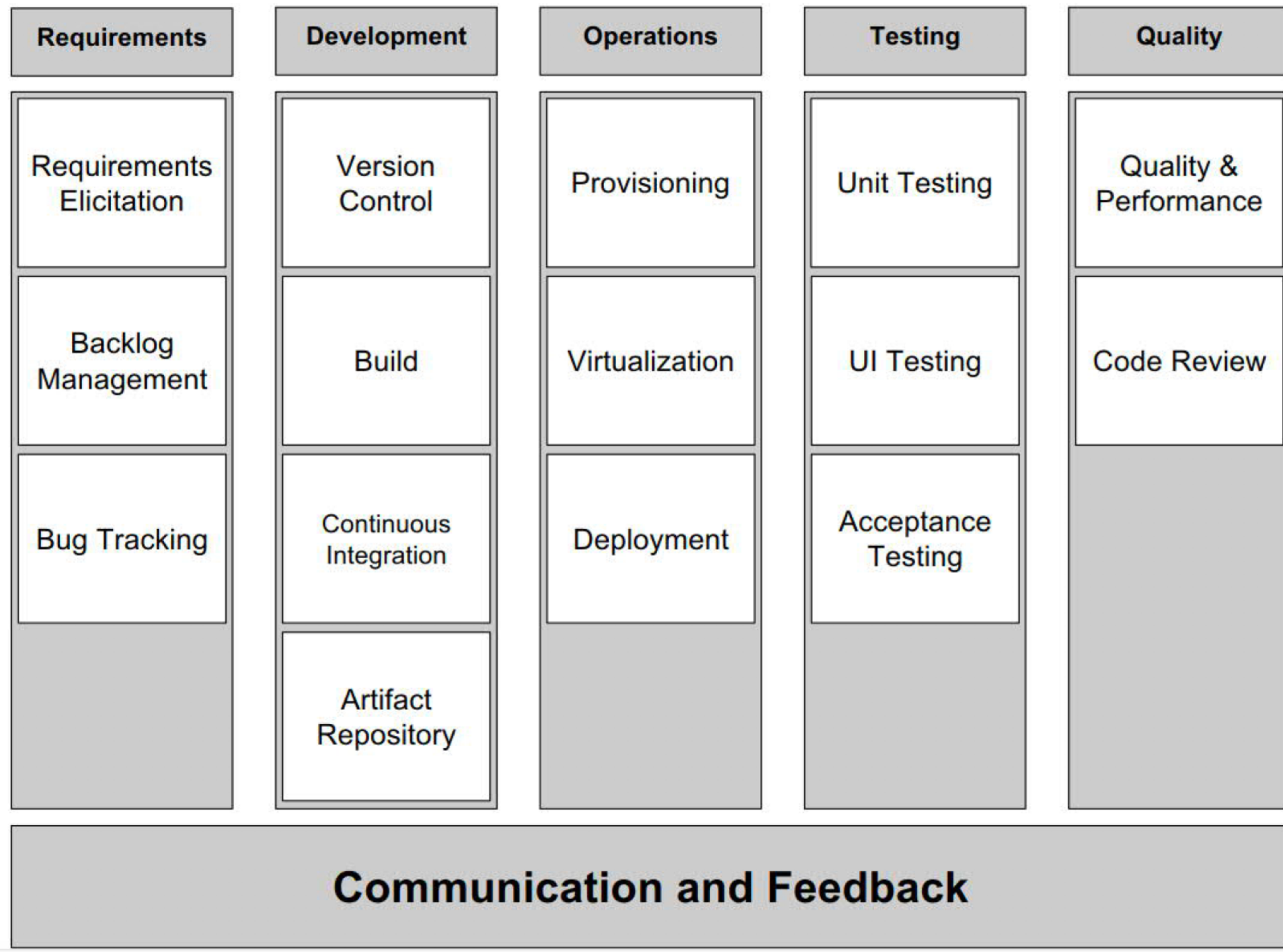
DevOps practices

- Organizational (draw a picture!)
 - increased scope of responsibilities for developers;
 - intensified cooperation between development and operations.
- Technical
 - automation,
 - monitoring
 - measurement

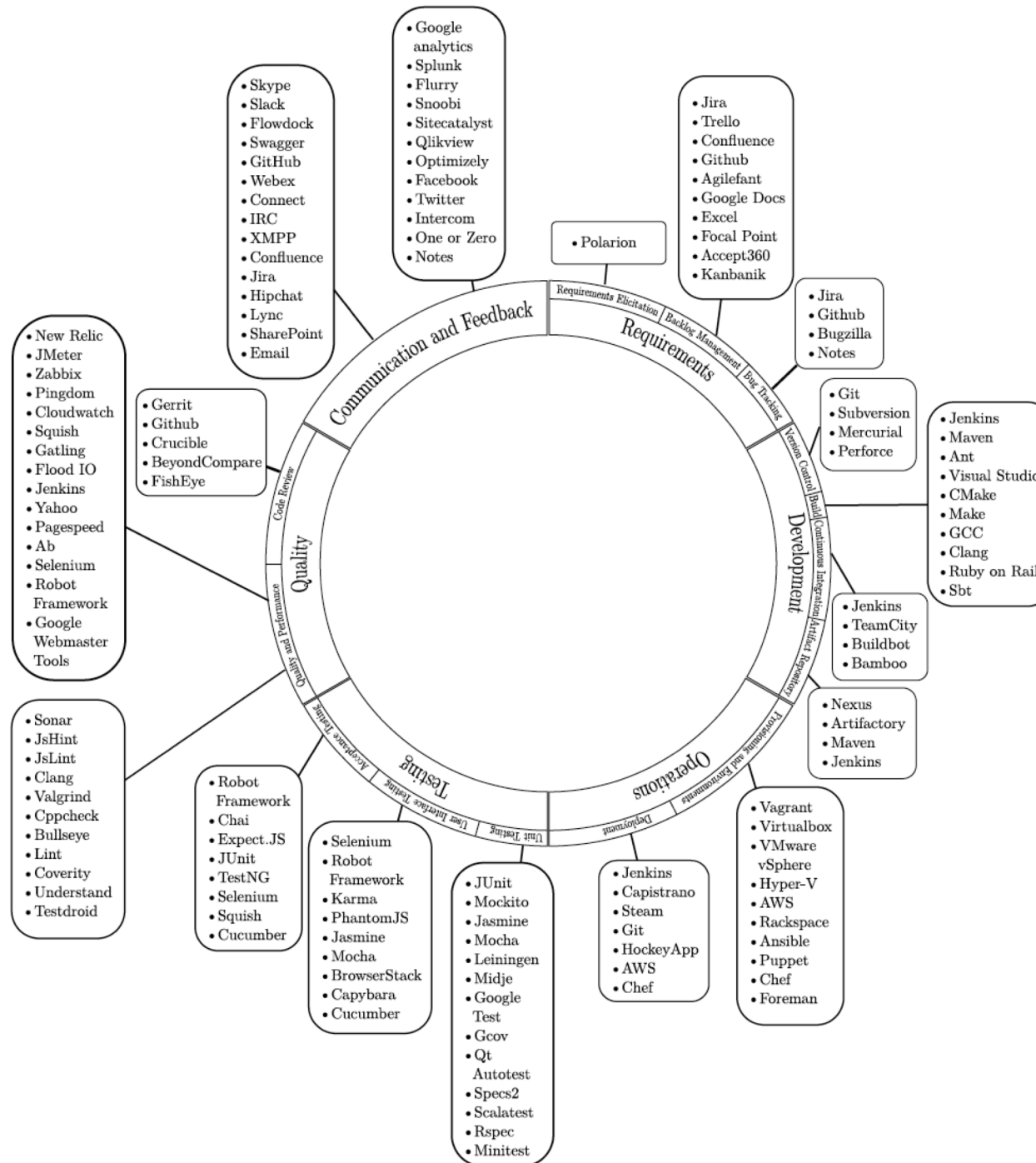
DevOps - benefits and challenges

- **improvement in speed (release cycle time)**
 - **continuous deployment of system changes**
 - **productivity of operations work**
 - **improved morale, knowledge and skills**
-
- **resource constraints;**
 - **insufficiencies in infrastructure management;**
 - **high demands for required skills and knowledge, and**
 - **difficulties in monitoring microservices**

Tools



Tools



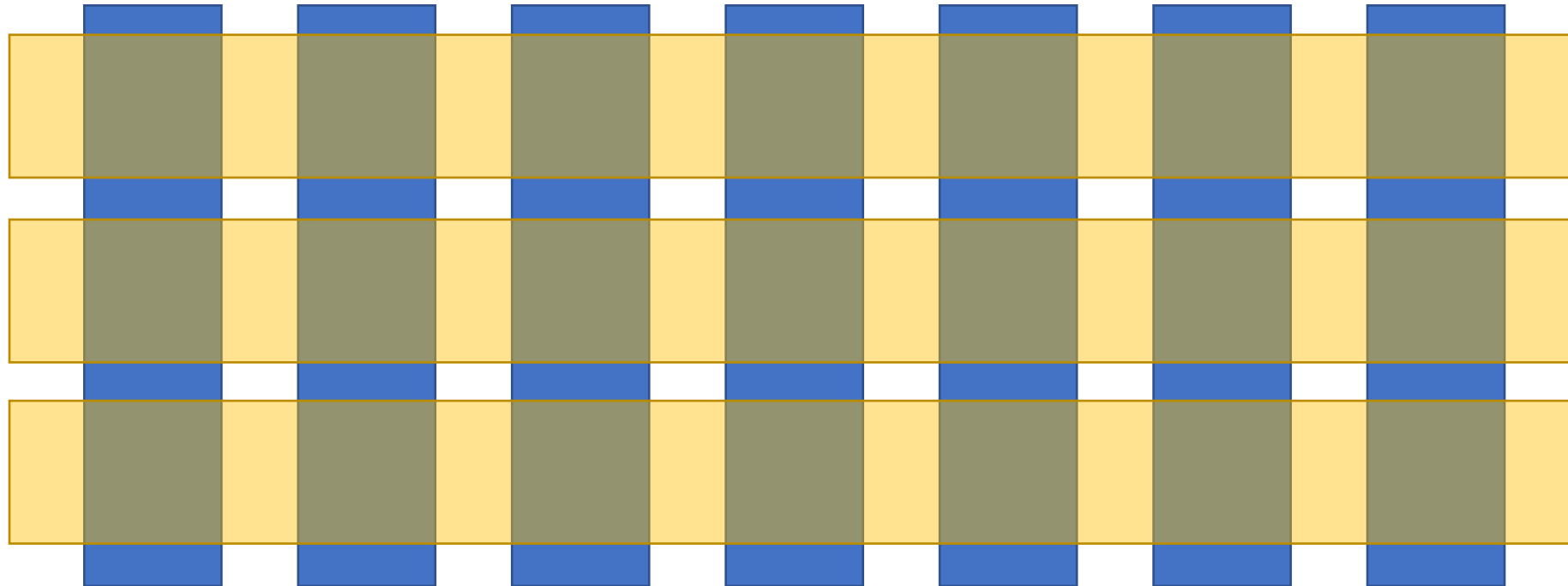
Back your course

What and why?

tools

ECO-systems

- Azure
- Amazon
- Google



Underlying principles

Course material will be in “plus”.

- Seminar from last Spring:
 - <https://plus.cs.tut.fi/cloudapps/spring-2019/>
- This implementation
 - <https://plus.tuni.fi/tie-23536/autumn-2019/>

Back to the bad news

The bad news

- This is a new course – that we are just ramping up
- We have a serious lack of teaching staff
- Unless some miracle happens we are forced to limit the number of participants.
- Priority is give to students who are
 - At least on master-level (at least in practice!)
 - Major is software engineering

Second bad news

- Our plan was to offer virtual machines for you to play with. Unfortunately, the system is still “under construction.”
- Thus you need a “Linux”.
 - I strongly recommend a virtual machine (and back-up snapshots)

Before limiting the number of students

- Let's see if I can get more teaching staff
- If some students realize that this course is not for them at this point
- Decisions done 06.09 (by the latest)

First plus-“exercise” is a background check

- Info will be sent as a “pop”-message
 - Deadline 06.09 @ 09:00
- Second will be hands-on with Docker
 - Deadline 13.09 @ 23:59