

# DevOps and Agile in control

(New report published)

June 24<sup>th</sup>, 2020



# Introduction

Sandeep Gangaram Panday, MSc RE CISA

Manager Internal Audit at Schuberg Philis

Guest lecturer

Chair working group Software Development NOREA

Email: [sgangarampanday@schubergphilis.com](mailto:sgangarampanday@schubergphilis.com)

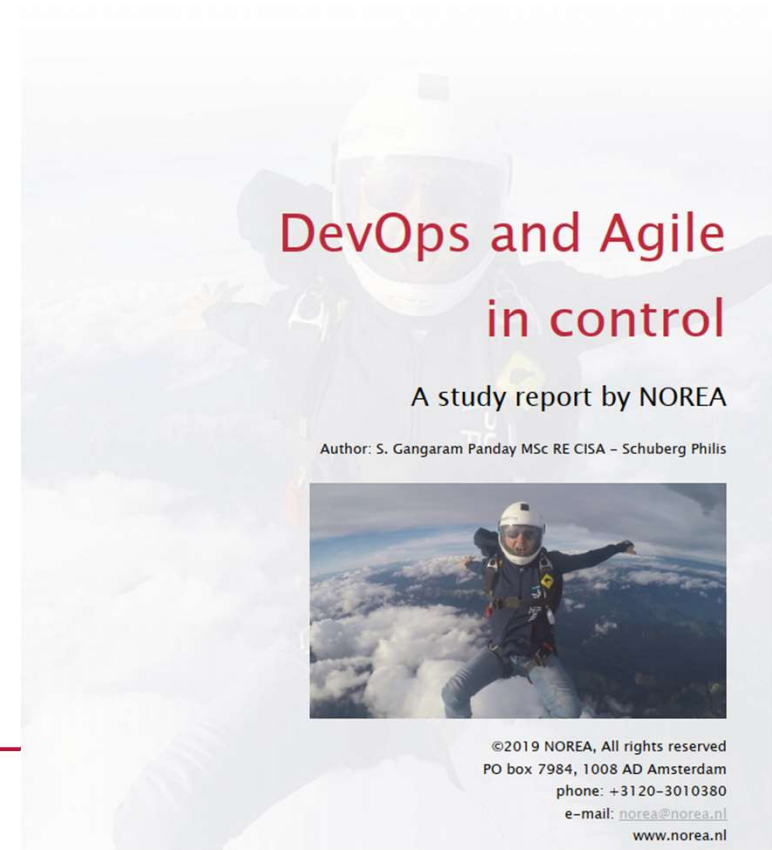
SCHUBERG  
PHILIS



---

## Norea DevOps paper

- ▶ <https://www.norea.nl/download/?id=6047>



# Schuberg Philis



Technology company



Mission critical environments only



Highly-regulated customers



For 9 years 100% customer recommendation



30+ audits per year



Agile/DevOps teams only



0 high risk findings since 2013

achmea



moneyou



ENEXIS



HEINEKEN



Rabobank



LeasePlan

BNP PARIBAS



AIR FRANCE KLM

JUMBO

ING



ARGENTA



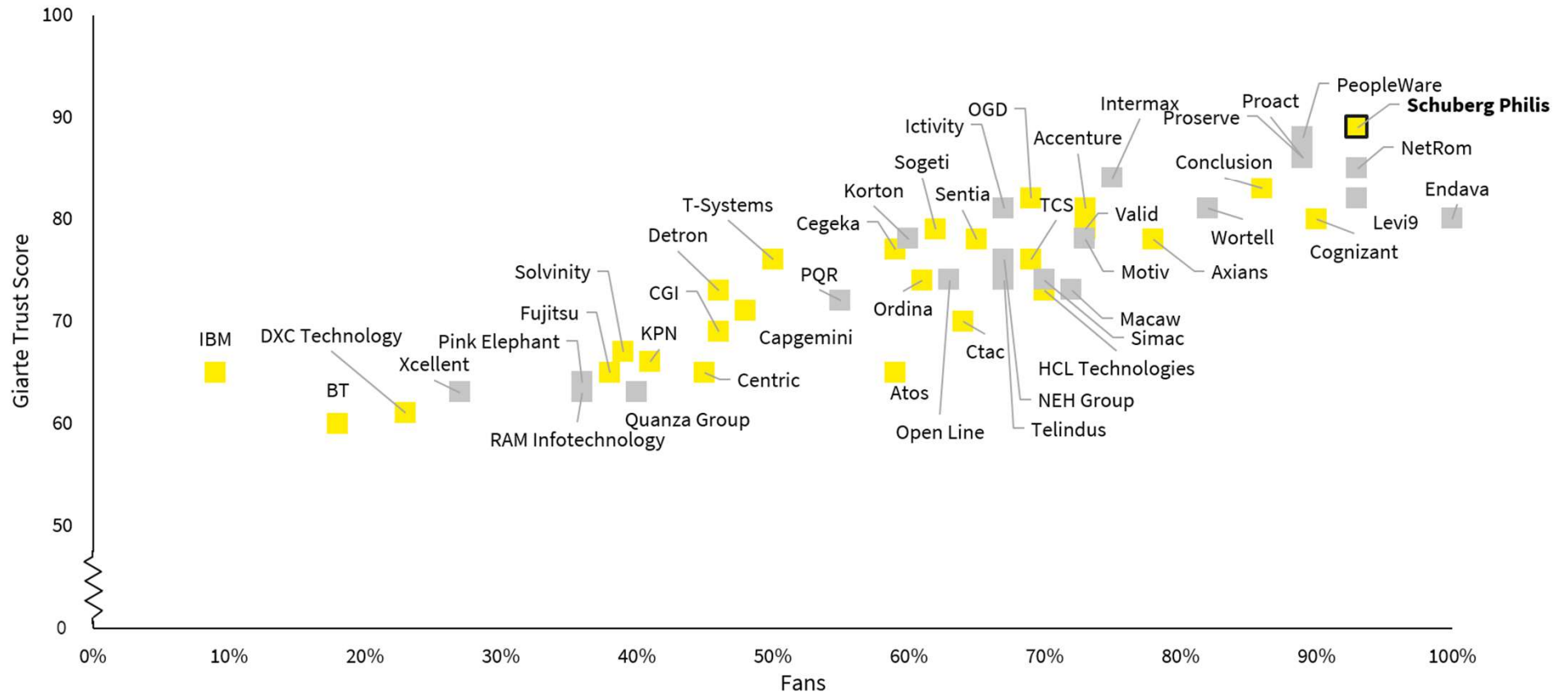
Eneco



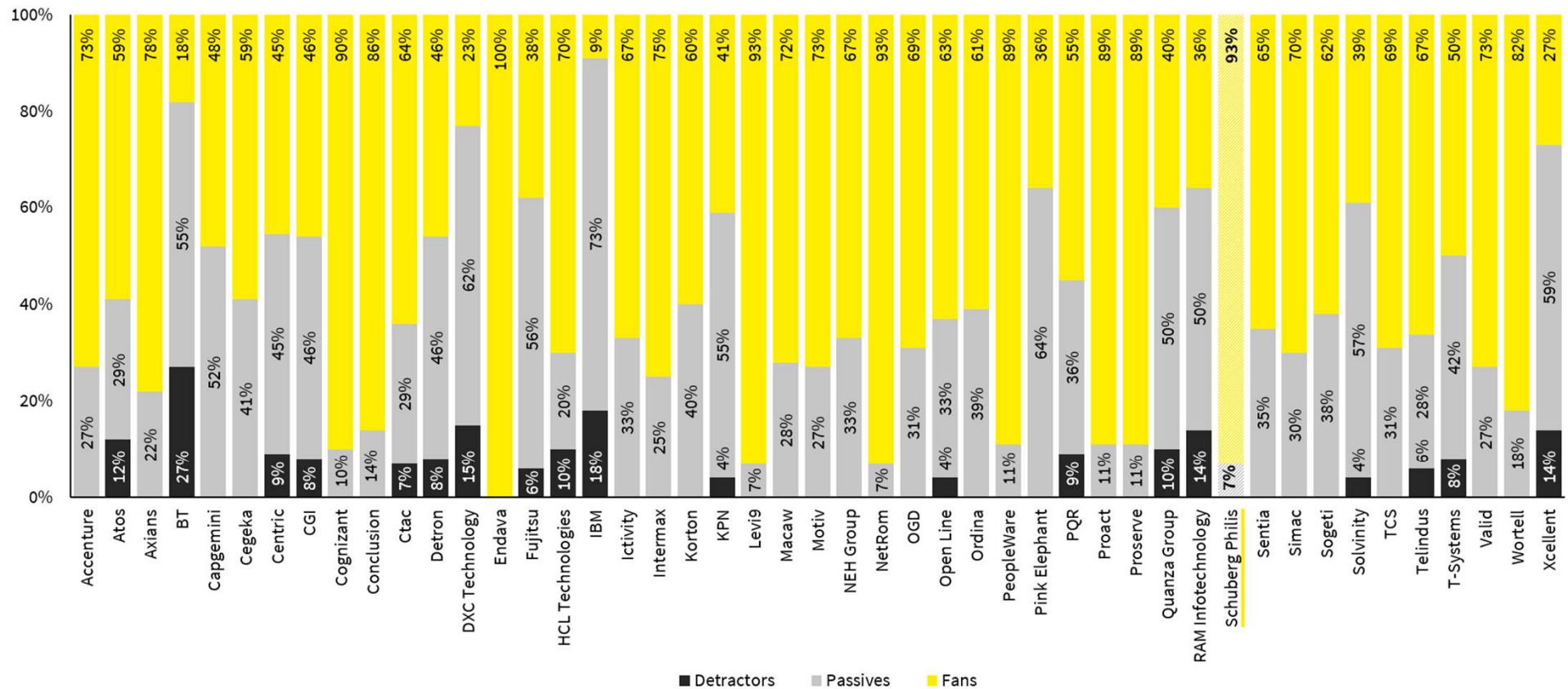
geldmaat

SCHUBERG  
PHILIS

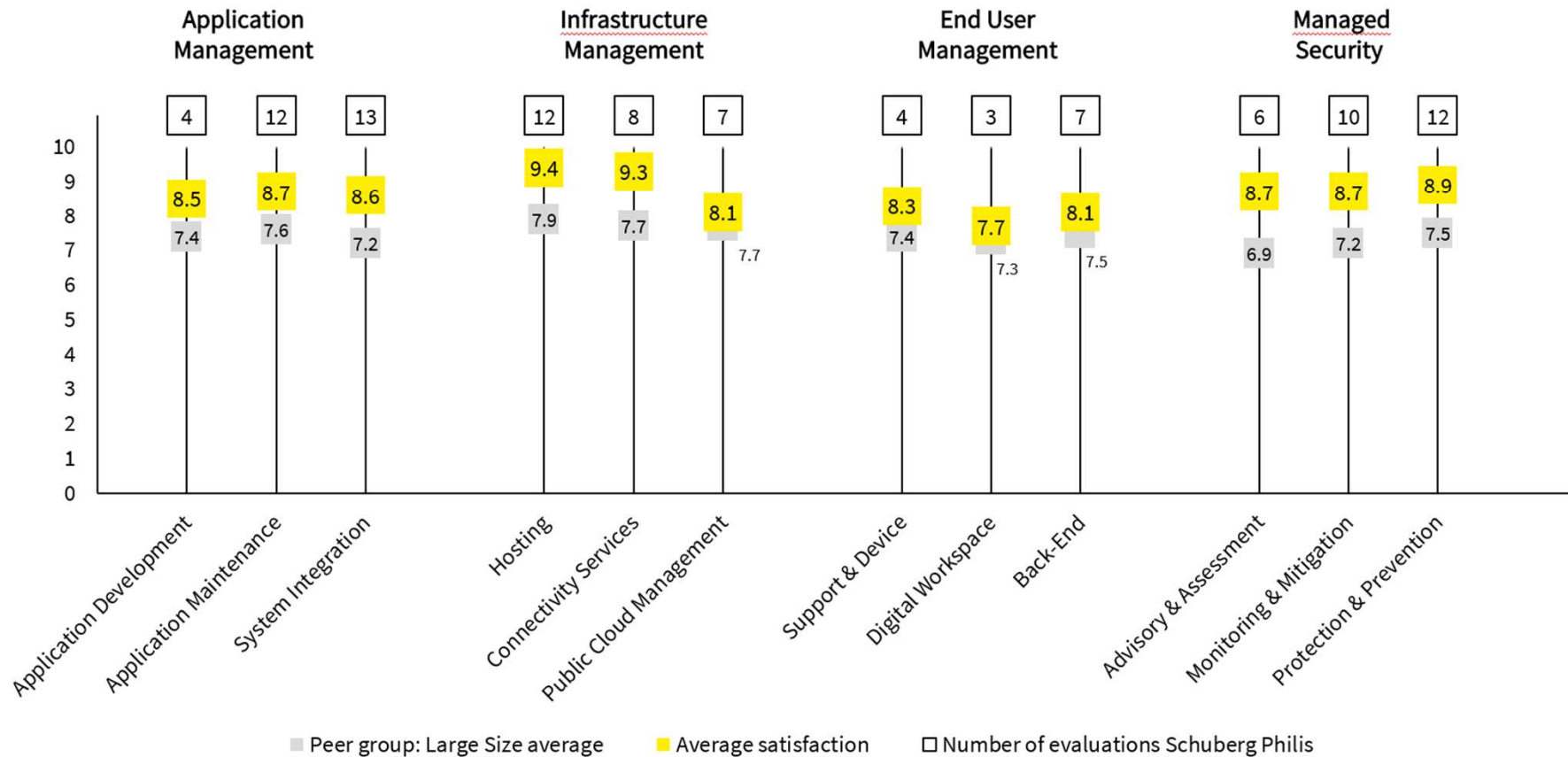
# GIARTE TRUST SCORE AND FAN SCORE | ALL SERVICE PROVIDERS



# RECOMMENDATION BENCHMARK | ALL SERVICE PROVIDERS



# SERVICE SATISFACTION | PEER GROUP COMPARISON





## Setting the scene

What is the impact for us  
as auditors/risk  
professionals??

“

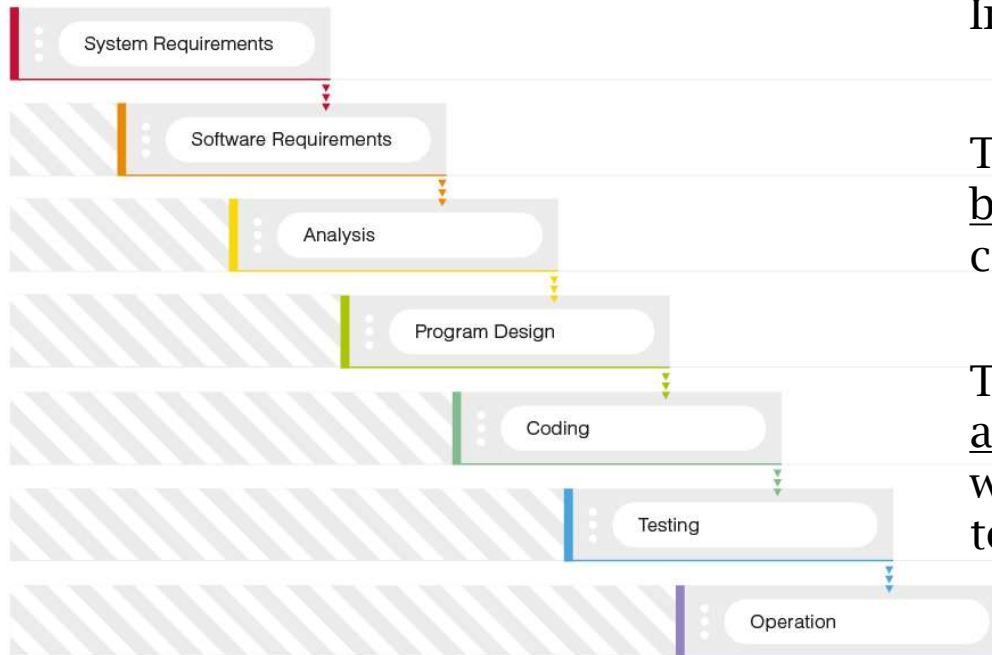
“Every company is a software company. You have to start thinking and operating like a digital company. It’s no longer just about procuring one solution and deploying one. It’s not about one simple software solution. It’s really you yourself thinking of your own future as a digital company.”

SATYA NADELLA  
CEO  
Microsoft



---

# Waterfall – was it meant to be sequential?



Introduced in 1956 by Herbert D. Benington

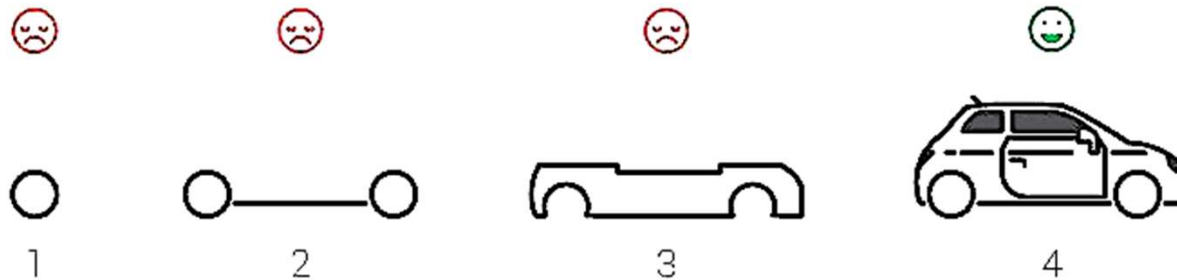
The waterfall top-down approach is not to be interpreted too literally: “This attitude can be terribly misleading and dangerous”.

The biggest mistake his team made: the attempt to make a too large release. He would now focus on smaller changes and test and evolve the system from there.

---

## Waterfall characteristics

- ▶ Project only completed after phase 4
- ▶ Requirements cannot change
- ▶ Separated teams per phase
- ▶ Need for extensive documentation



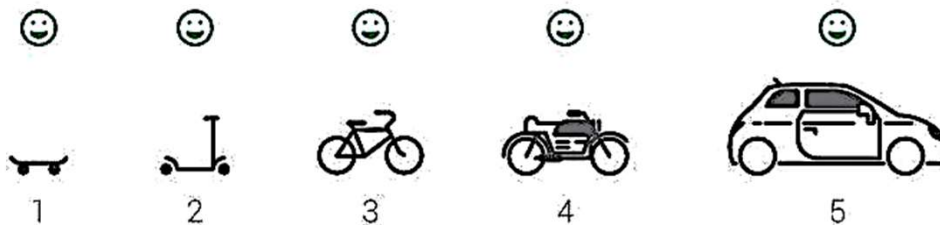
---

## Agile characteristics

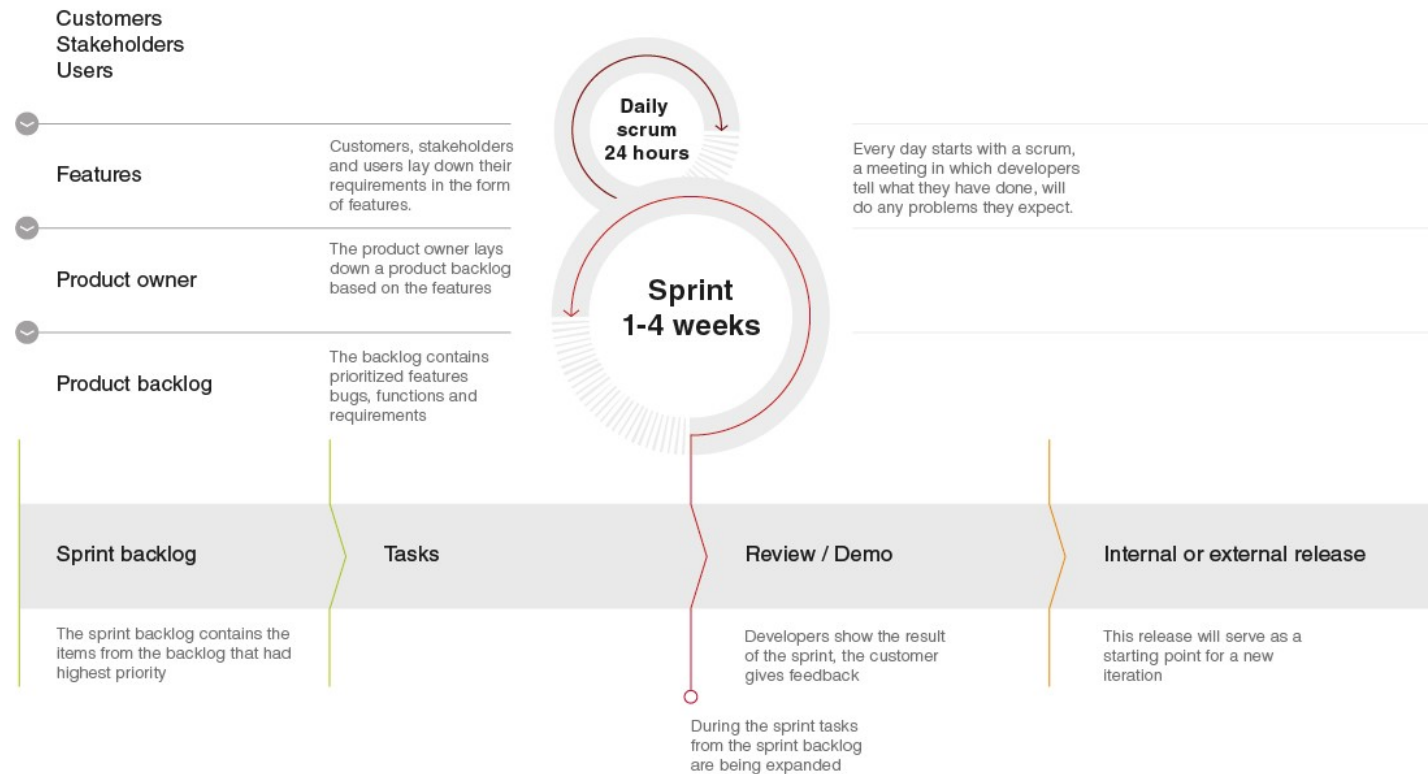
- ▶ A MVP after phase 1
- ▶ After each sprint the priorities can be re-visited
- ▶ Focus on constant improvement
- ▶ Importance of interaction and team dynamics
- ▶ Quicker feedback

Don't get demotivated  
or 'colored' by the  
Agile Manifesto!

————— How to build a minimum viable product —————



# SCRUM as implementation method (**one of the many**)

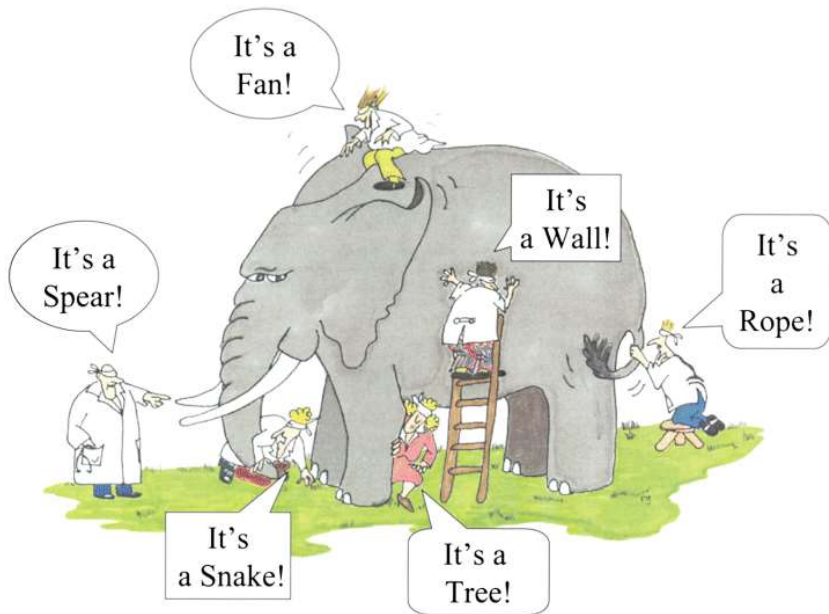


---

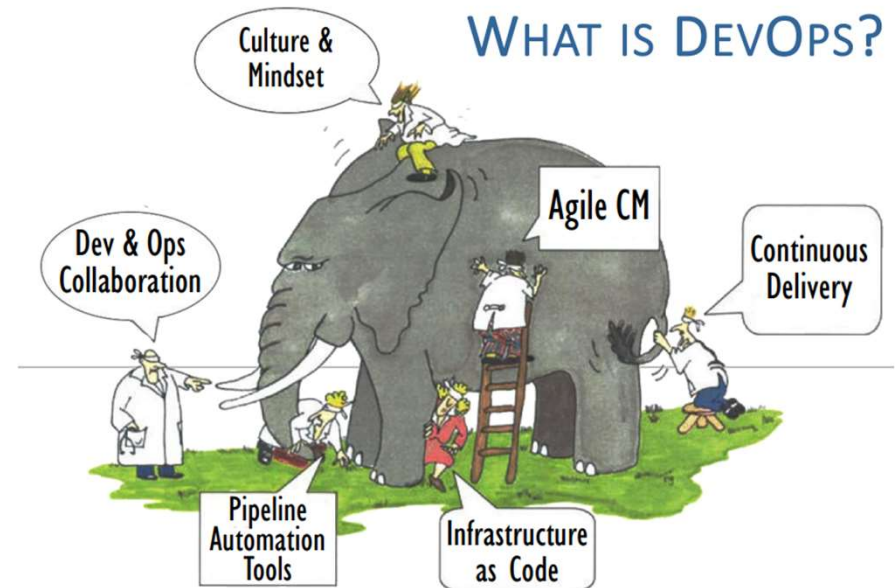
# What is DevOps?

1. Tool?
2. Process?
3. Philosophy?
4. Methodology?
5. Way of working?

# What is DevOps?

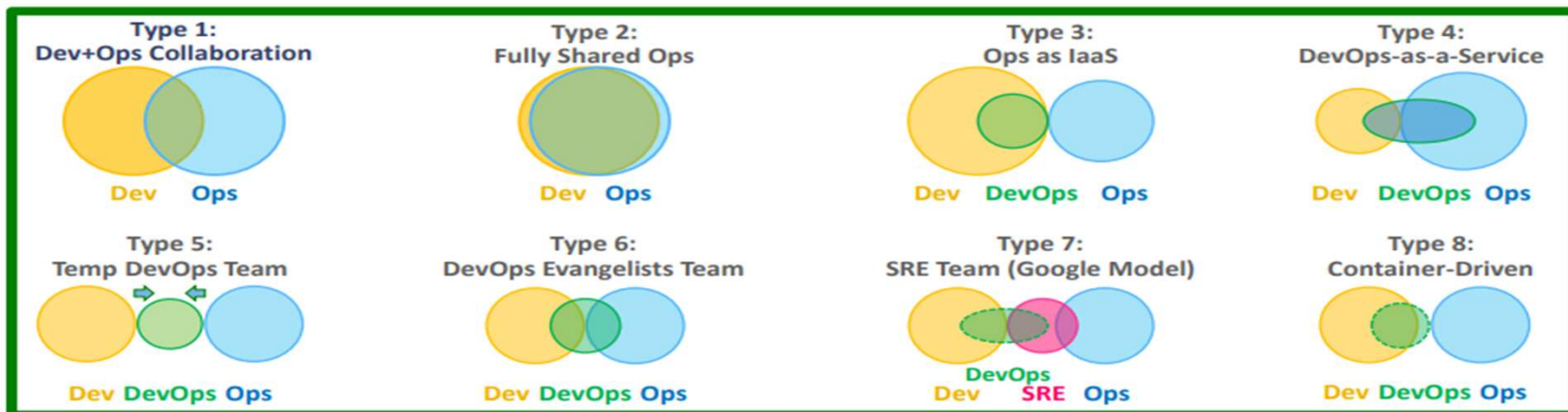


Source: Blind men and the elephant



---

## DevOps types from [www.devopstopologies.com](http://www.devopstopologies.com)





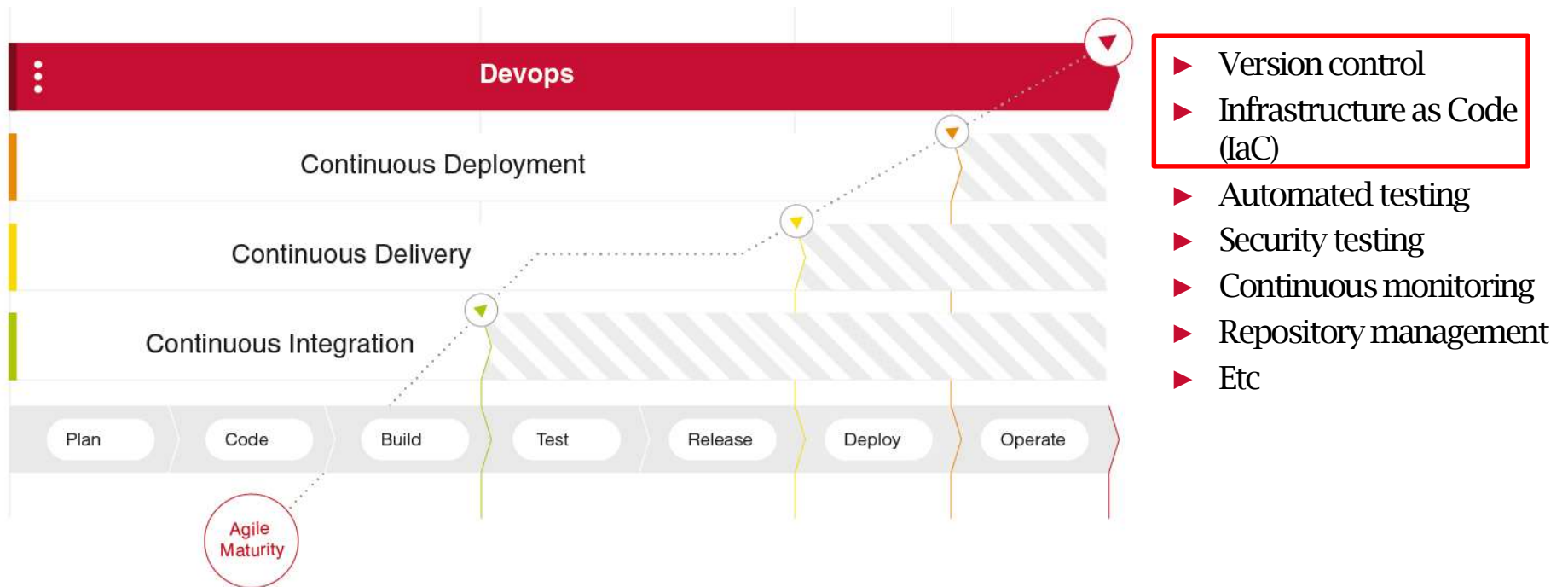
---

## Our definition

- ▶ “DevOps is the union of, at least, software development and IT operations activities in an environment that has incorporated the accompanying cultural and technical principles to deliver business value at a high frequency.”
- ▶ Source: Norea study report



# Technical principles



# Version control example

The screenshot shows a GitHub pull request interface. At the top, there's a navigation bar with links for Code, Issues (19), Pull requests (6), Projects (0), Wiki, Security, Insights, and Settings. The main title of the pull request is "Add Slack Channel Name to Domain config #864". Below the title, it says "Merged" and "sanderv32 merged 2 commits into master from feature/domain-slack 6 days ago". There are statistics for Conversation (0), Commits (2), Checks (0), and Files changed (18). A green bar indicates +197 -13 changes. A comment by remibergsma is shown, containing a screenshot of a form with fields for Name (Test), Email (test@cosmic.io), and Slack Channel Name (cosmic). Below the comment, the commit history is visible, showing "remibergsma added 2 commits 12 days ago" and "sanderv32 merged commit cd90995 into master 6 days ago". The commit details show "Add Slack Channel Name" and "UI: Add and edit SlackChannel name". The pull request status shows "1 check passed" and "mccd jenkins build All is well". On the right side, there are sections for Reviewers, Assignees, Labels, Projects, Milestone, Notifications, and Participants.

Code Issues 19 Pull requests 6 Projects 0 Wiki Security Insights Settings

## Add Slack Channel Name to Domain config #864

Merged sanderv32 merged 2 commits into master from feature/domain-slack 6 days ago

Conversation 0 Commits 2 Checks 0 Files changed 18 +197 -13

remibergsma commented 9 days ago • edited Member

Details

Name Test

Email test@cosmic.io

Slack Channel Name cosmic

remibergsma added 2 commits 12 days ago

- Add Slack Channel Name 255a38b
- UI: Add and edit SlackChannel name b54078c

sanderv32 merged commit cd90995 into master 6 days ago Hide details Revert

1 check passed

✓ mccd jenkins build All is well Details

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications: Customize Unsubscribe

2 participants

Lock conversation

# Infra as Code example

Source: <https://www.slideshare.net/AmazonWebServices/devops-on-aws-deep-dive-on-infrastructure-as-code>

```
{
  "AWSTemplateFormatVersion" : "2010-09-09",

  "Description" : "AWS CloudFormation Sample Template
  EC2InstanceSample: **WARNING** This template an Amazon EC2 instances.
  You will be billed for the AWS resources used if you create a stack
  from this template.",

  "Parameters" : {
    "KeyName" : {
      "Description" : "Name of an existing EC2 KeyPair to enable SSH
      access to the instance",
      "Type" : "String"
    },
    "Environment": {
      "Type" : "String",
      "Default" : "Dev",
      "AllowedValues" : ["Mgmt", "Dev", "Staging", "Prod"],
      "Description" : "Environment that the instances will run in."
    }
  },
  "Mappings" : {
    "RegionMap" : {
      "us-east-1" : { "AMI" : "ami-7f418316" },
      "us-west-2" : { "AMI" : "ami-16fd7026" }
    }
  },
  "Conditions" : {
    "EnableEBSOptimized" : { "Fn::Equals" : [{"Ref" : " Environment
```

```
    }, "Prod"]}],
  },
  "Resources" : {
    "Ec2Instance" : {
      "Type" : "AWS::EC2::Instance",
      "Properties" : {
        "KeyName" : { "Ref" : "KeyName" },
        "EbsOptimized" : { "Fn::If": [ " EnableEBSOptimized ",
        {"true"}, {"false"} ] },
        "ImageId" : { "Fn::FindInMap" : [ "RegionMap", { "Ref" :
        "AWS::Region" }, "AMI" ] },
        "UserData" : { "Fn::Base64" : "80" }
      }
    }
  },
  "Outputs" : {
    "InstanceId" : {
      "Description" : "InstanceId of the newly created EC2 instance",
      "Value" : { "Ref" : "Ec2Instance" }
    },
    "PublicDNS" : {
      "Description" : "Public DNSName of the newly created EC2
      instance",
      "Value" : { "Fn::GetAtt" : [ "Ec2Instance", "PublicDnsName" ] }
    }
  }
}
```

# CI/CD example

+ Step 1: Validate prompted values

+ Step 2: Pre-Deploy HealthCheck

+ Step 3: Slack -Start Deployment

















+ Step 4: Enable the Maintenance page in Chef

+ Step 5: Activate Maintenance Page on Webservers

+ Step 7: Set new release version for SQL Servers

+ Step 8: Chef-Client on Database Servers

## Release

3.7.1.57733	 3.7.1.57733 Jan 23, 2020 9:03 AM	 3.7.1.57733 Jan 23, 2020 10:53 AM	DEPLOY...	DEPLOY...	
3.7.0.57565	 3.7.0.57565 Jan 21, 2020 7:07 PM	 3.7.0.57565 Jan 21, 2020 8:43 PM	DEPLOY...	DEPLOY...	
3.6.5.56860	 3.6.5.56860 Jan 8, 2020 2:59 PM	 3.6.5.56860 Jan 9, 2020 5:17 PM	 3.6.5.56860 Jan 16, 2020 11:47 AM	 3.6.5.56860 Jan 16, 2020 12:23 PM	 3.6.5.56860 Jan 23, 2020 6:34 AM
3.5.12.55389	 3.5.12.55389 Nov 28, 2019 9:54 AM	 3.5.12.55389 Nov 26, 2019 5:33 PM	 3.5.12.55389 Nov 27, 2019 10:01 AM	 3.5.12.55389 Nov 27, 2019 10:29 AM	 3.5.12.55389 Nov 28, 2019 7:13 AM

set new release version for SQL Servers

lient on secondary Database Servers

ase for ALL servers

+ Step 13: Chef-client on MSMQ servers

+ Step 14: Enforce set release for ALL servers - AMQ Servers

# CI/CD example

## Build #2846 (09-Sep-2019 14:01:23)

<a title="Add Slack Channel Name to Domain config" href="https://github.com/MissionCriticalCloud/cosmic/pull/864">PR #864</a>: Add Slack Channel Na



### Changes

1. handle non-existing key ([commit: 7697cb0](#)) ([detail / githubweb](#))
2. Add Slack Channel Name ([commit: 255a38b](#)) ([detail / githubweb](#))
3. UI: Add and edit SlackChannel name ([commit: b54078c](#)) ([detail / githubweb](#))



GitHub pull request #864 of commit b54078c1768a80f44fae9a962c240efc0619b092, no merge conflicts.



















Revision: b54078c1768a80f44fae9a962c240efc0619b092

- detached



[Test Result](#) (no failures)

S	R	Job	Build #	Duration	Console
<i>Full Build</i>					
		<a href="#">0020-full-build</a>	<a href="#">build #3888</a>	( 3 hr 35 min )	
<i>Build maven project and prepare infrastructure for integrations tests</i>					
		<a href="#">9997-maven-build</a>	<a href="#">build #3883</a>	( 5 min 0 sec )	
		<a href="#">0200-prepare-infrastructure-for-integration-tests</a>	<a href="#">build #3871</a>	( 1 min 59 sec )	
<i>Setup infrastructure for integration tests</i>					
		<a href="#">0300-setup-infrastructure-for-integration-tests</a>	<a href="#">build #3438</a>	( 1 min 25 sec )	
<i>Deploy datacenter</i>					
		<a href="#">0400-deploy-datacenter-for-integration-tests</a>	<a href="#">build #3324</a>	( 4 min 57 sec )	
<i>Run integration tests</i>					
		<a href="#">0500-run-integration-tests</a>	<a href="#">build #3036</a>	( 3 hr 15 min )	
<i>Sonar analysis</i>					
		<a href="#">9998-maven-sonar-build</a>	<a href="#">build #2682</a>	( 6 min 48 sec )	
<i>Report, Archive and Cleanup</i>					
		<a href="#">0600-collect-artifacts-and-cleanup</a>	<a href="#">build #2693</a>	( 1 min 32 sec )	

# Testing pyramid

- ▶ Unit Tests: testing a single method, class or function in isolation.
- ▶ Acceptance Tests: testing the application as a whole.
- ▶ Integration Tests: testing the correct interaction with other applications and services

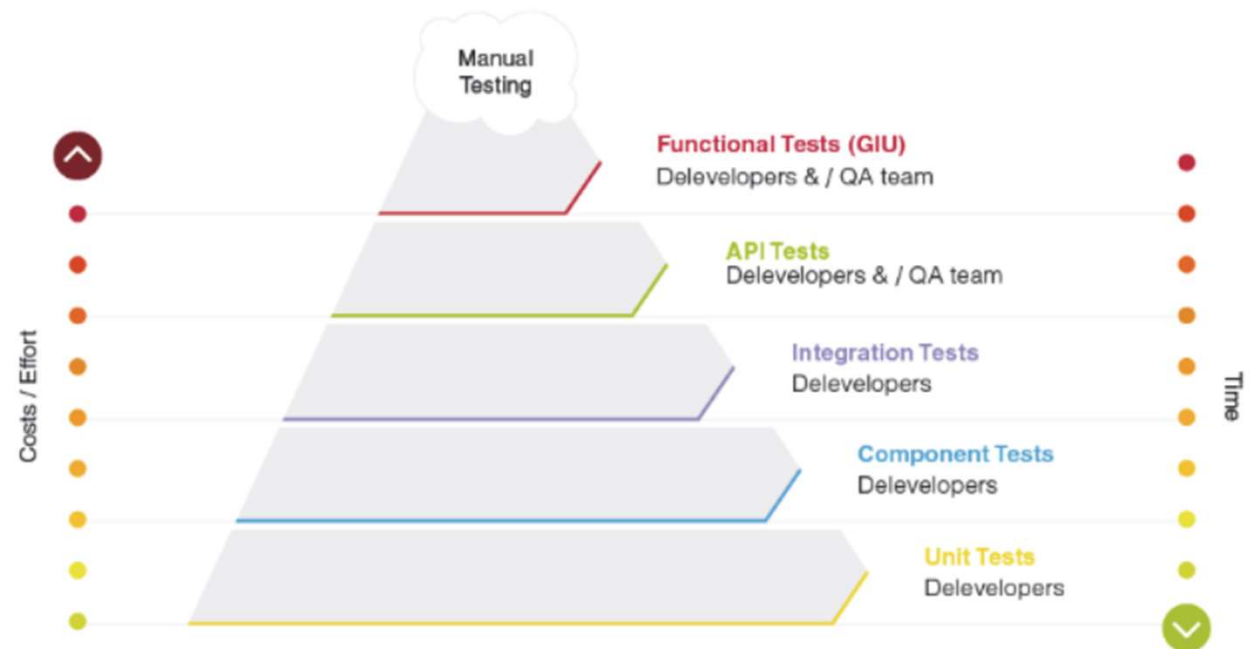


Figure 5 The ideal test pyramid [43]



---

## Take away

- ▶ IaC gives us insight in **admin configuration activities** which were previously almost unavailable for us.
- ▶ The high-level of automation goes together with a lot of source code maintained in the VCS -> This is the **new audit documentation**. We have more documentation than ever.
- ▶ Almost each implementation is a unique implementation therefore its imperative that we understand the concept and techniques to be able to tailor our approach -> **See CobiT 2019**.

PERIODIC TABLE OF DEVOPS TOOLS (V2)																		Aws Amazon Web Services																																																																																						
Os Open Source		SCM		Database Mgmt		Build																																																																																																		
Fr Free		CI		Repo Mgmt		Testing																																																																																																		
Fm Freemium		Deployment		Config / Provisioning		Containerization																																																																																																		
Pd Paid		Cloud / IaaS / PaaS		Release Mgmt		Collaboration																																																																																																		
En Enterprise		BI / Monitoring		Logging		Security																																																																																																		
1 Fm Gh Github	2 Fm Aws Amazon Web Services	3 Os Gt Git	4 En Dm DBmaestro	5 En Ch Chef	6 En Pu Puppet	7 Os An Ansible	8 Os Sl Salt	9 Os Dk Docker	10 Pd Az Azure	11 Fm Bb Bitbucket	12 Os Lb Liquibase	13 Os Ot Otto	14 En Bl BladeLogic	15 Os Va Vagrant	16 Fr Tf Terraform	17 Os Rk Rkt	18 En Gc Google Cloud Platform	19 Os Gl GitLab	20 En Rg Redgate	21 Os Mv Maven	22 Os Gr Gradle	23 Os At ANT	24 Os Fn FitNesse	25 Fr Se Selenium	26 Os Ga Gatling	27 Fr Dh Docker Hub	28 Os Jn Jenkins	29 Pd Ba Bamboo	30 Os Tr Travis CI	31 Pd Gd Deployment Manager	32 Os Sf SmartFrog	33 Os Cn Consul	34 Os Bc Bcf2	35 Os Mo Mesos	36 En Rs Rackspace	37 Os Sv Subversion	38 En Dt Datomic	39 Os Gt GrunT	40 Os Gp Gulp	41 Os Br Broccoli	42 Fr Cu Cucumber	43 Os Cj Cucumber.js	44 Fr Qu Quint	45 Os Npm npm	46 Fm Cs Codeship	47 Pd Vs Visual Studio	48 Fm Cr CircleCI	49 Fr Cp Capistrano	50 Fr Ju JuJu	51 Os Rd Rundeck	52 Os Cf CFEngine	53 Fr Ds Swarm	54 Os Op OpenStack	55 Os Hg Mercurial	56 En Dp Delphi	57 Fr Sb sbt	58 Os Mk Make	59 Os Ck CMake	60 Fr Jt JUnit	61 Fr Jm JMeter	62 Fr Tn TestNG	63 Os Ay Artifactory	64 Fm Tc TeamCity	65 Fm Sh Shippable	66 Os Cc CruiseControl	67 En Ry RapidDeploy	68 Fm Cy CodeDeploy	69 En Oc Octopus Deploy	70 En No CA Nolo	71 Os Kb Kubernetes	72 Fm Hr Heroku	73 En Cw ESPW	74 En Id Idera	75 Os Msb MSBuild	76 Os Rk Rake	77 Fr Pk Packer	78 Os Mc Mocha	79 Fr Km Karma	80 Os Jm Jasmine	81 Os Nx Nexus	82 Os Co Continuum	83 Fm Ca Continua CI	84 Pd So Solano CI	85 Pd Xld XL Deploy	86 En Eb ElasticBox	87 Fm Dp Deploybot	88 En Ud UrbanCode Deploy	89 Os Nm Nomad	90 En Os OpenShift	91 En Xlr XL Release	92 En Ur UrbanCode Release	93 En Bm BMC Release Process	94 En Hp HP CodeR	95 En Au Automic	96 En Pl Plutora Release	97 En Sr Serena Release	98 Pd Tfs Team Foundation	99 Fm Tr Trello	100 Pd Jr Jira	101 Fm Rf HipChat	102 Fm Sl Slack	103 Fm Fd Flowdock	104 Pd Pv Pivotal Tracker	105 En Sn ServiceNow

XebiaLabs

Follow @xebialabs

---

# How to audit

## 3. Agile and DevOps in control

Based on our research and as introduced in the preceding paragraphs we advise a 3-step approach for auditing DevOps environments:

1. Determining the software development methodology or principles being used
2. Cultural maturity assessment
3. Control assessment

---

# 1. Development approach in use

Delivery frequency	Methodology/practice	Description
Quarterly or less	Waterfall	The software development is done in phased steps leading to large planned software releases.
Monthly	Agile (principles and procedures)	The software development process follows an Agile approach, but deployments are still performed manually.
(bi-)weekly	Agile+	A CI/CD pipeline is implemented and used to deploy software to the production environment, but manual steps are still required.
Daily or more	DevOps / Continuous Deployment	Every change that is accepted is automatically build, tested and delivered by the automated delivery pipeline and possibly also deployed to the production environment.

Table 1: Guidance to determine software development method

## 2. Cultural maturity



## Examples on culture assessments @ Schuberg Philis

- ▶ Do you feel comfortable brainstorming in front of each other (also about possible issues)?
- ▶ Is it easy to get help from your coworkers when you need it?
- ▶ Do you think that you have good visibility into project priorities or progress?
- ▶ Do you actively ask feedback?





# Examples of tools to measure

The screenshot shows a document titled "TEAMS Guide: Understand team effectiveness". On the left is a sidebar with a table of contents including: Introduction, Define what makes a "team", Define "effectiveness", Collect data and measure effectiveness, Identify dynamics of effective teams, Tool: Help teams determine their own needs, Tool: Foster psychological safety, and Help teams take action. The main content area is titled "Introduction" and contains text about collaborative work at Google and the success of Google's Project Oxygen research. It mentions that Google researchers applied a similar method to discover the secrets of effective teams, code-named Project Aristotle. At the bottom, there is a "NEXT" button and a link to "Define what makes a 'team' ->".

GOOGLE

## The DORA Technology Performance Assessment

A unique, holistic, scientific tool to drive technology performance improvement



<http://devops-research.com>



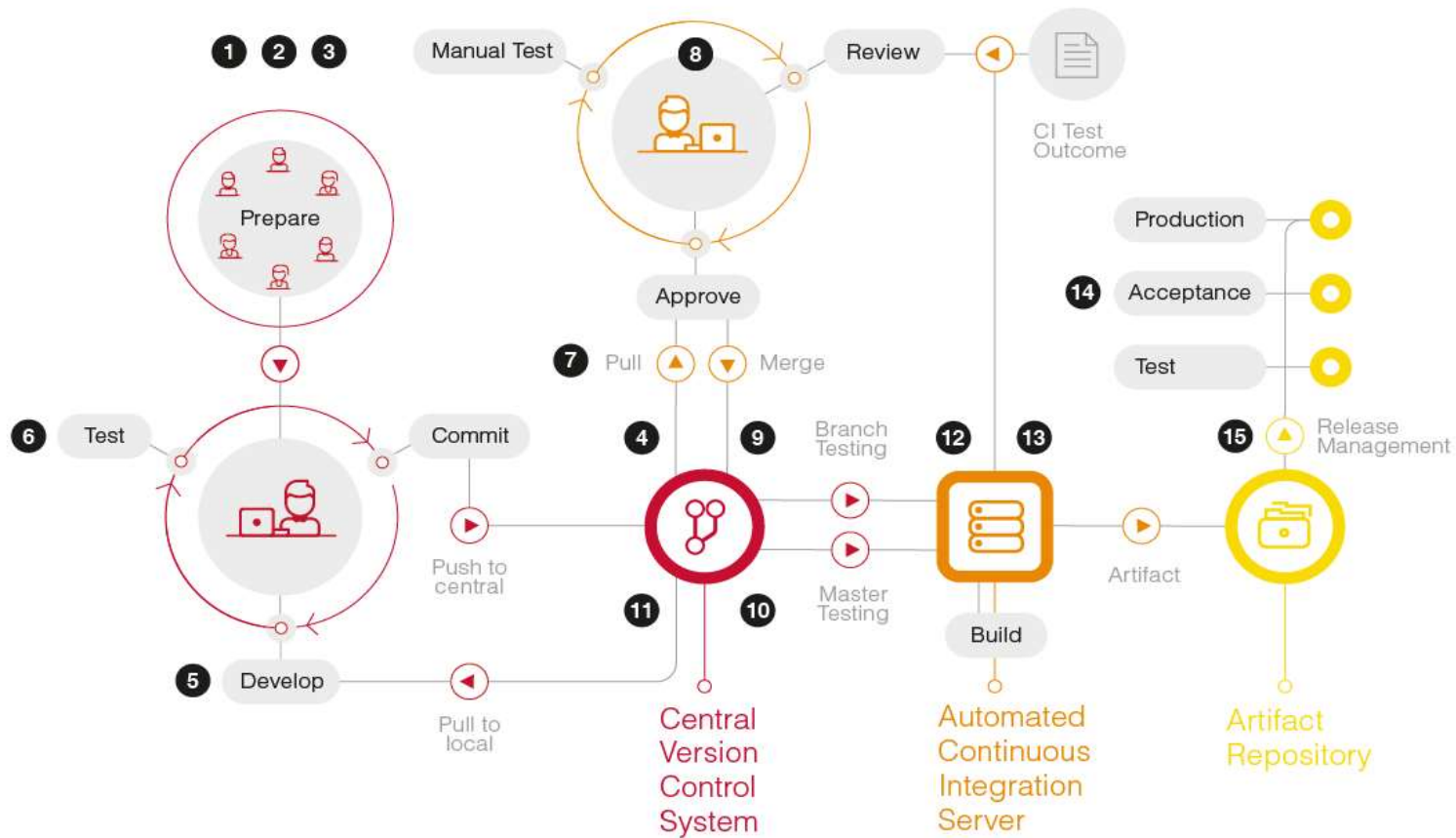
DORA

The screenshot shows the Microsoft DevOps Self-Assessment tool. At the top, the Microsoft logo is on the left, and the text "DevOps Self-Assessment" is on the right, with the tagline "Helping you become a high-performer" below it. The main heading is "DevOps Self-Assessment". Below this, a paragraph explains that the ability to develop and deliver software is an important piece of any organization's ability to deliver value to customers, pivot when necessary, beat competitors to market, and respond to regulatory and compliance requirements. It states that delivering value with software often requires a technology transformation, and these transformations necessitate improving key capabilities. Below this paragraph, it says "The assessment has questions that touch on several key areas. These areas include:" followed by a bulleted list: Process, Technology and automation, Culture, Measurement, and Outcomes.

MICROSOFT

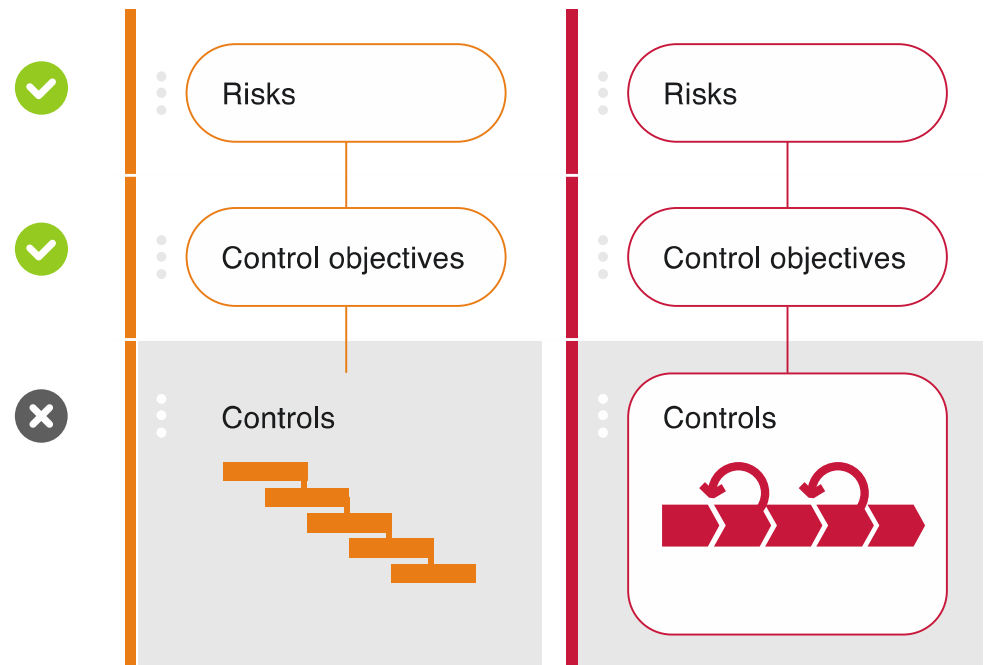


### 3. Control assessment



# What changed?

- ▶ Same risks:
  - Confidentiality, Integrity, Availability
- ▶ Same control objectives :
  - IT entity-level, Change management, Security management, Operational management.
- ▶ Different controls

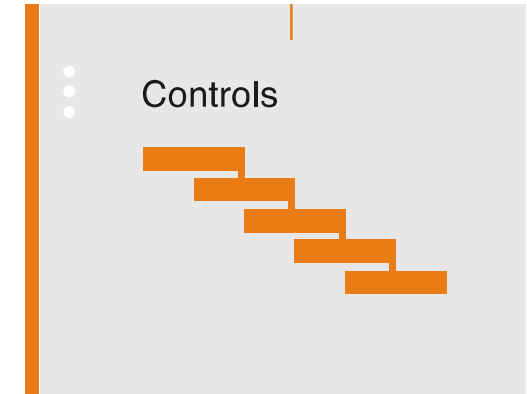


---

# Example

**C1:** All changes are reviewed by the Change Control Board (CCB) prior to release.

- a) The changes are submitted for review at least two weeks prior to the next CCB meeting.
- b) The submitter must complete the Change Control Form (CCF), documenting the changes to be made, which environments the change should be applied to, what risks are associated with the change, and rollback procedures.
- c) If the CCB approves the change, the change will be scheduled for the next release window with the IT Operations team.



## CS1 evidence:

- a) Documentation of CCB procedures.
- b) CCB meeting agendas for the last year.
- c) CCFs for each CCB meeting for the last year.
- d) Record of approval for each CCF.
- e) Record of changes applied for each production release window, along with CCF for each of those changes.
- f) Record of which changes were applied successfully and which failed.
- g) For change failures, record of rollback procedures applied and outcome of the rollback.



## Example cont'd

### Controls

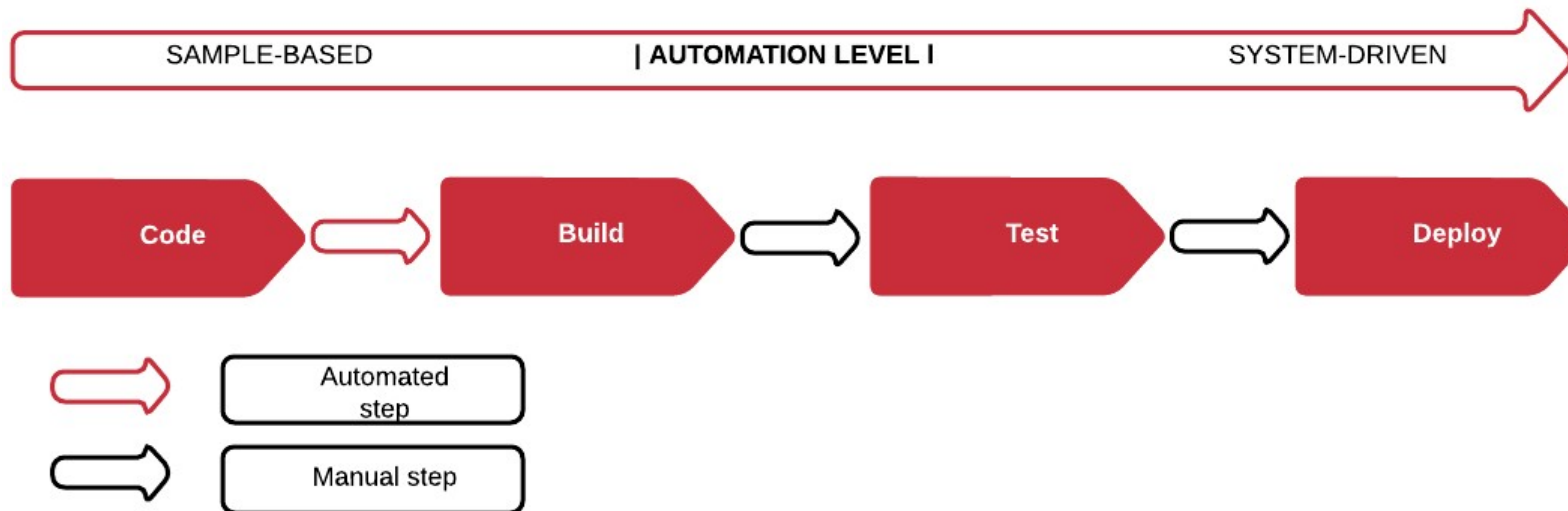


7	Develop	<p>A <u>peer review of the code</u> is mandatory for the code changes based on code review guidelines.</p>	<ol style="list-style-type: none"><li>1. The team has a documented their code review <u>guidelines</u> for performing the peer-review e.g. based on best practices such as Google Style Guide or, based on the application context, enriched with security checks from the OWASP Application Security Verification Standard (level 1 through 3).</li><li>2. Once committed, the developer can push the <u>local branch</u> to the CVS. Ensure the developed code remains a branch in this stage, until further testing and merging/approval is completed.</li><li>3. The VCS enforces a <u>peer review</u> of the code change by another developer of the team who can pull the new code change for review.</li></ol>
---	---------	--	---

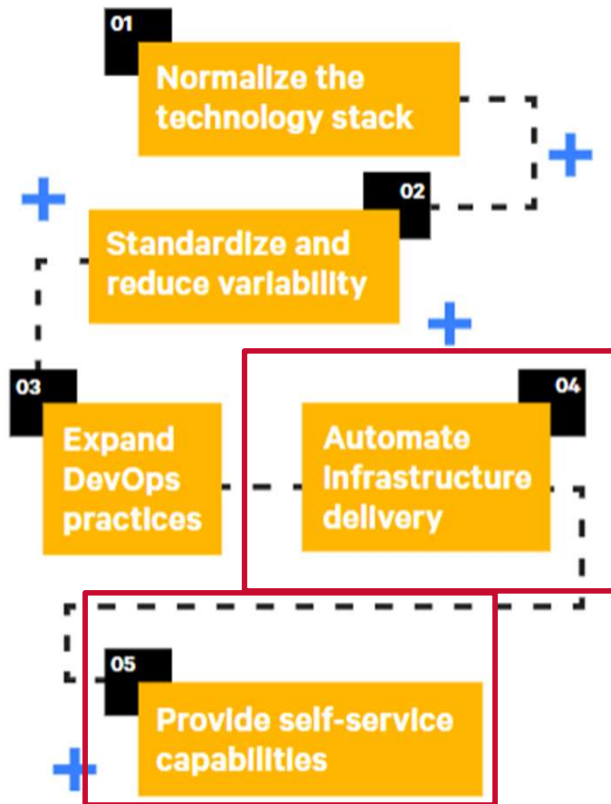
---

## Test strategies

- Sample based
- System driven (reperformance of one event)

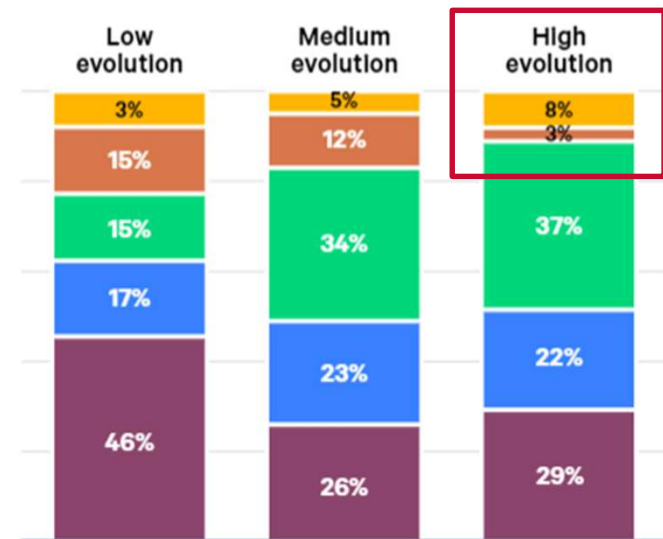


# Rome wasn't built in a day



Automation progress by evolutionary scale

- Most services are available via self-service.
- A few key services are available via self-service.
- Teams are collaborating to automate services for broad use.
- Teams are automating services they control, for others' needs.
- Teams are automating services they control, for their own need.



---

## Introducing a new test strategy

# Full-population Exceptional Analysis Testing (FEAT)

### Controls

- *Determine key controls to be tested*
- *Determine live data source per control*

### Logic

- *Create scripts with success/fail logic for automated testing*
- *Implement scripts in CI/CD pipeline*

### Automated testing

- *Continuous automated testing on full population in CI/CD pipeline*

### Exception analysis

- *Analysis of deviations (root-cause)*
- *Determine control effectiveness*



# What to test?

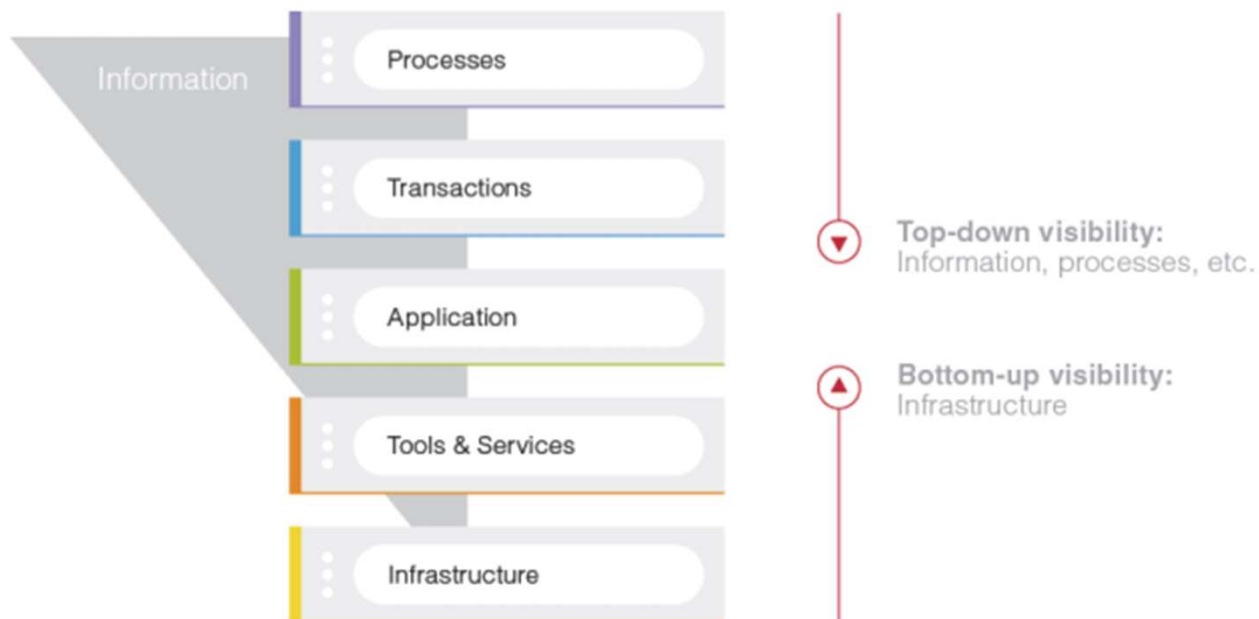


Figure 7: IT stack divided between DevOps and Shared Services teams [30]

---

## Summary

- ▶ Don't stop thinking:
  - New controls
  - Every implementation is unique, no standard control framework
  - DevOps is not a fixed methodology but a moving destination
  - System-driven, sample-based or FEAT test approach?
  - Culture is just as important as the technical practices
- ▶ The audit has changed: more technical & inclusion of cultural assessment
- ▶ Its already here: Technology and Financial Services firms are the largest applicants of Agile & DevOps.