IT auditorsdag 2019
Digital transformation & control

DevOps and Agile in control

(New report published)

17 september 2019

NOREA
DE BEROEPSORGANISATIE VAN IT-AUDITORS

ISACA
Vertrouwen in en waarde uit informatiesystemen
Netherlands Chapter

Introduction

Sandeep Gangaram Panday, MSC CISA RE

Manager Internal Audit at Schuberg Philis

Guest lecturer VU & UvA

Chair working group Software Development

NOREA

Email:

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

Rabobank

ABN·AMRO

BNP PARIBAS

AIR FRANCE KLM

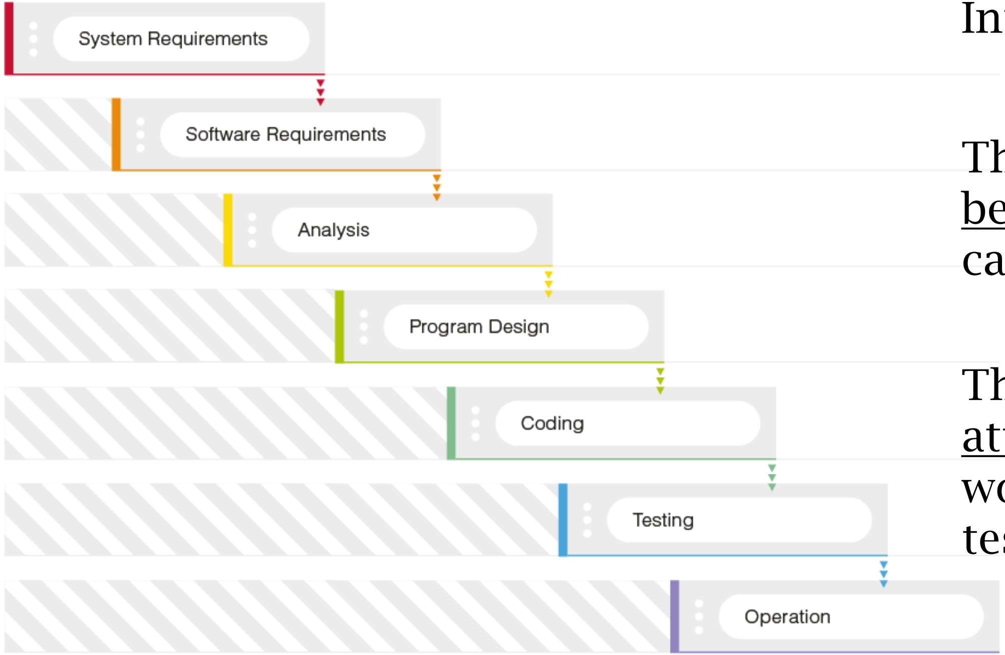
ING

ARGENTA

Eneco

geldmaat

Waterfall – was it meant to be sequential?



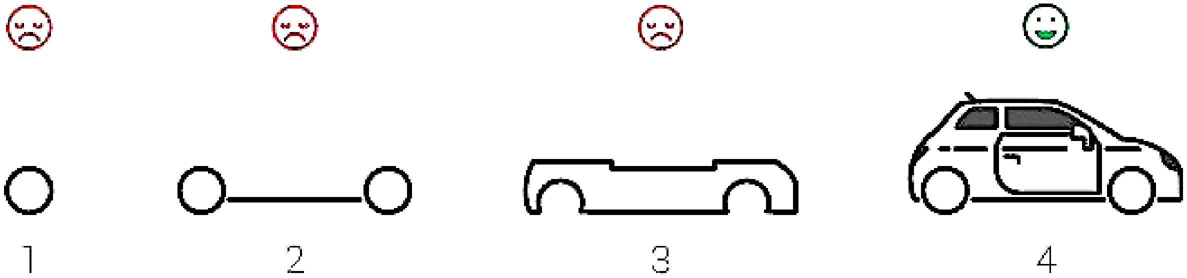
Introduced in 1956 by Herbert D. Benington

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.

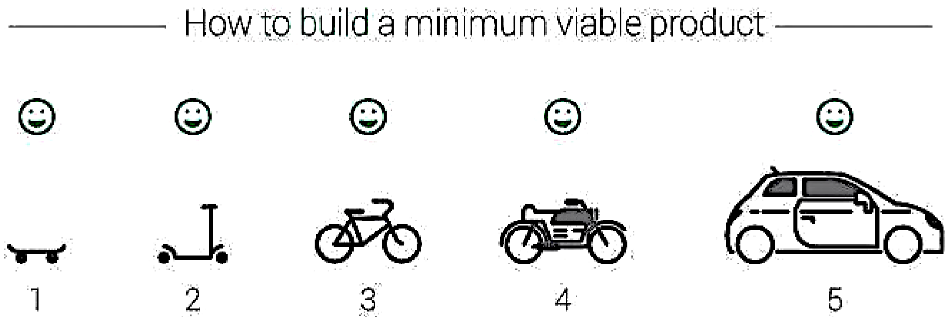
Waterfall characteristics

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



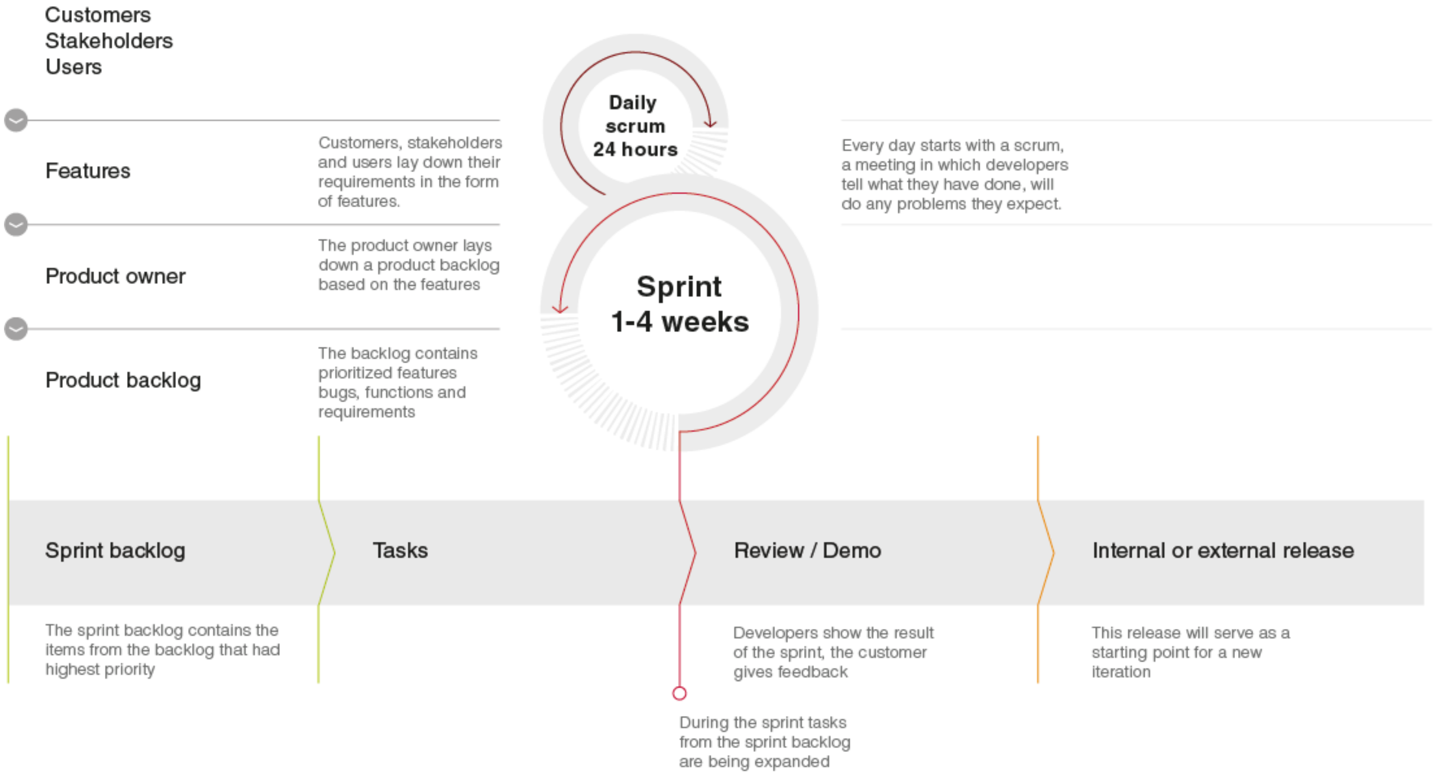
Agile characteristics

- 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



How to build a minimum viable product

SCRUM as implementation method



Customers
Stakeholders
Users

Features

Customers, stakeholders and users lay down their requirements in the form of features.

Product owner

The product owner lays down a product backlog based on the features

Product backlog

The backlog contains prioritized features bugs, functions and requirements

Daily scrum 24 hours

Every day starts with a scrum, a meeting in which developers tell what they have done, will do any problems they expect.

Sprint 1-4 weeks

Sprint backlog

The sprint backlog contains the items from the backlog that had highest priority

Tasks

During the sprint tasks from the sprint backlog are being expanded

Review / Demo

Developers show the result of the sprint, the customer gives feedback

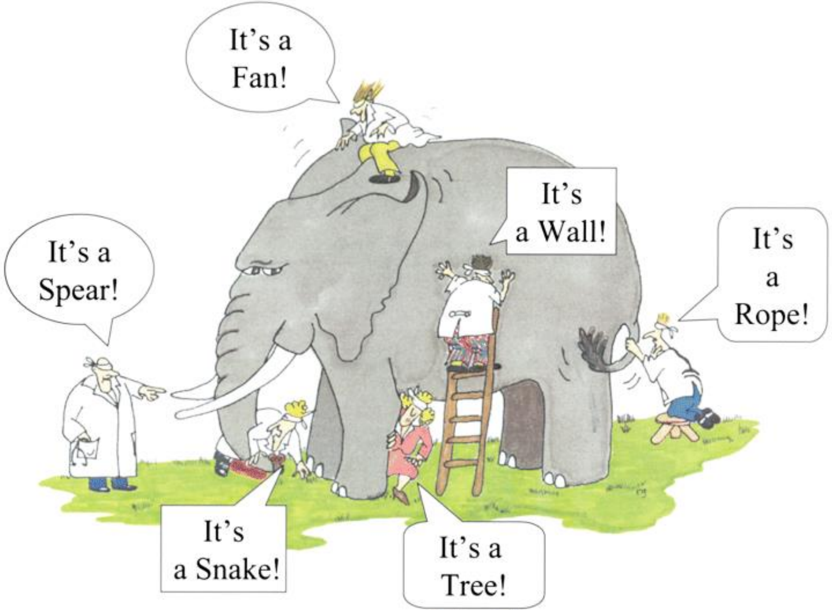
Internal or external release

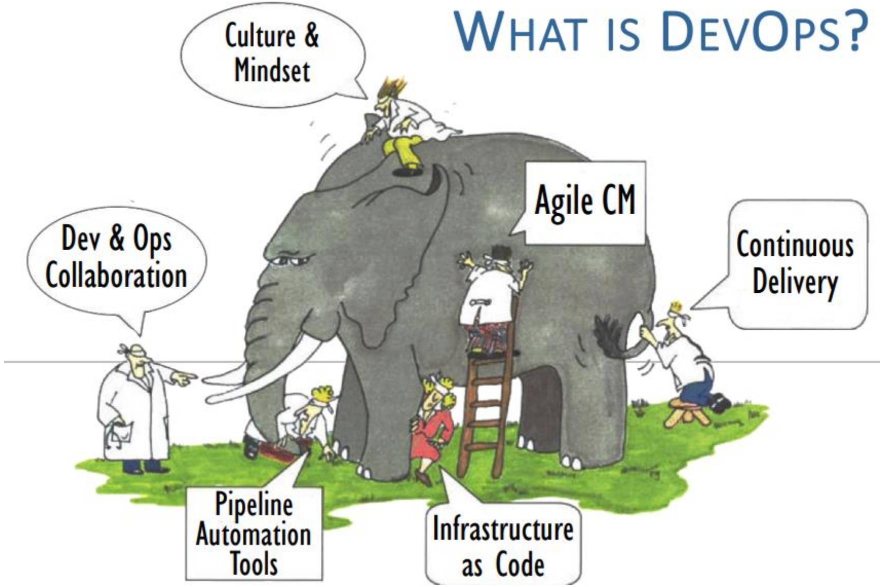
This release will serve as a starting point for a new iteration

What is DevOps?

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

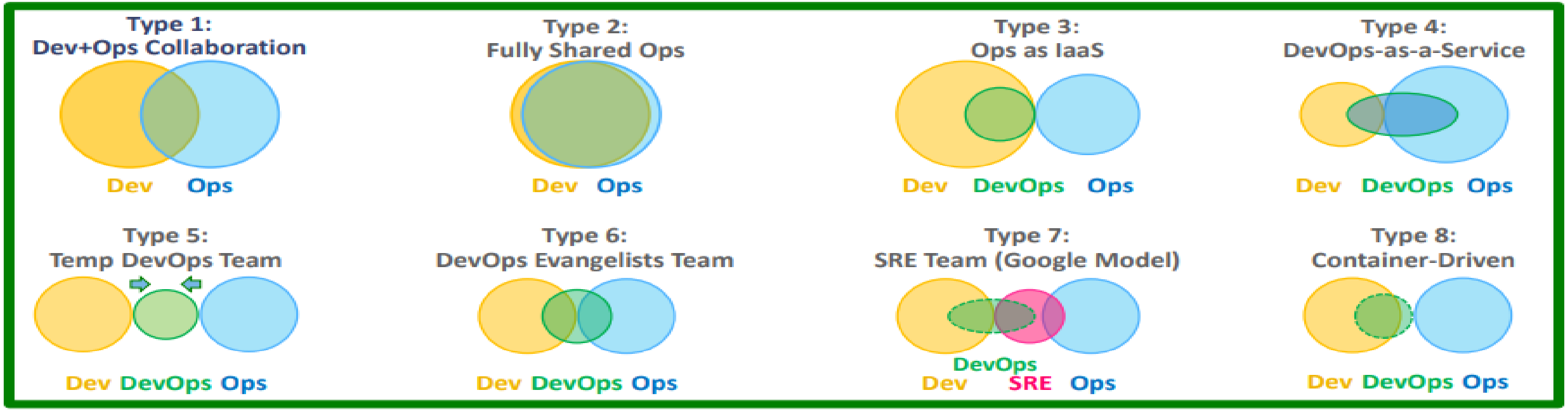
What is DevOps?





Source: Blind men and the elephant

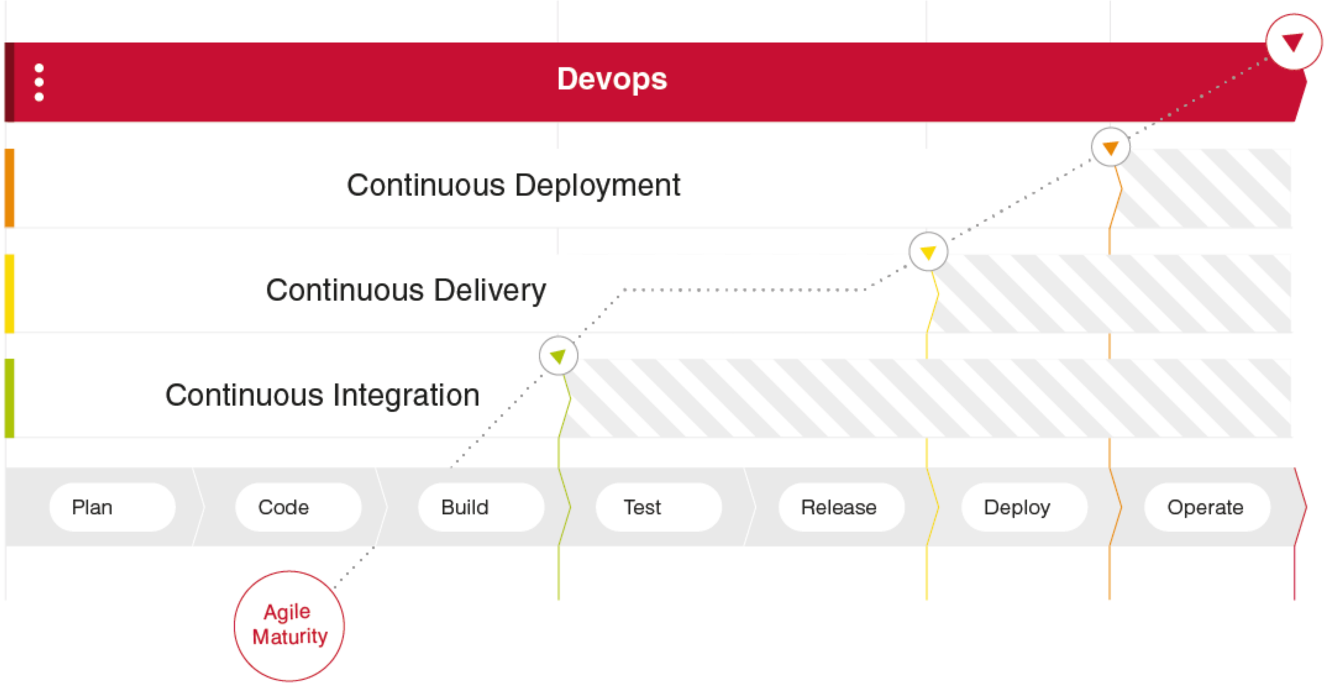
DevOps types from www.devopstopologies.com



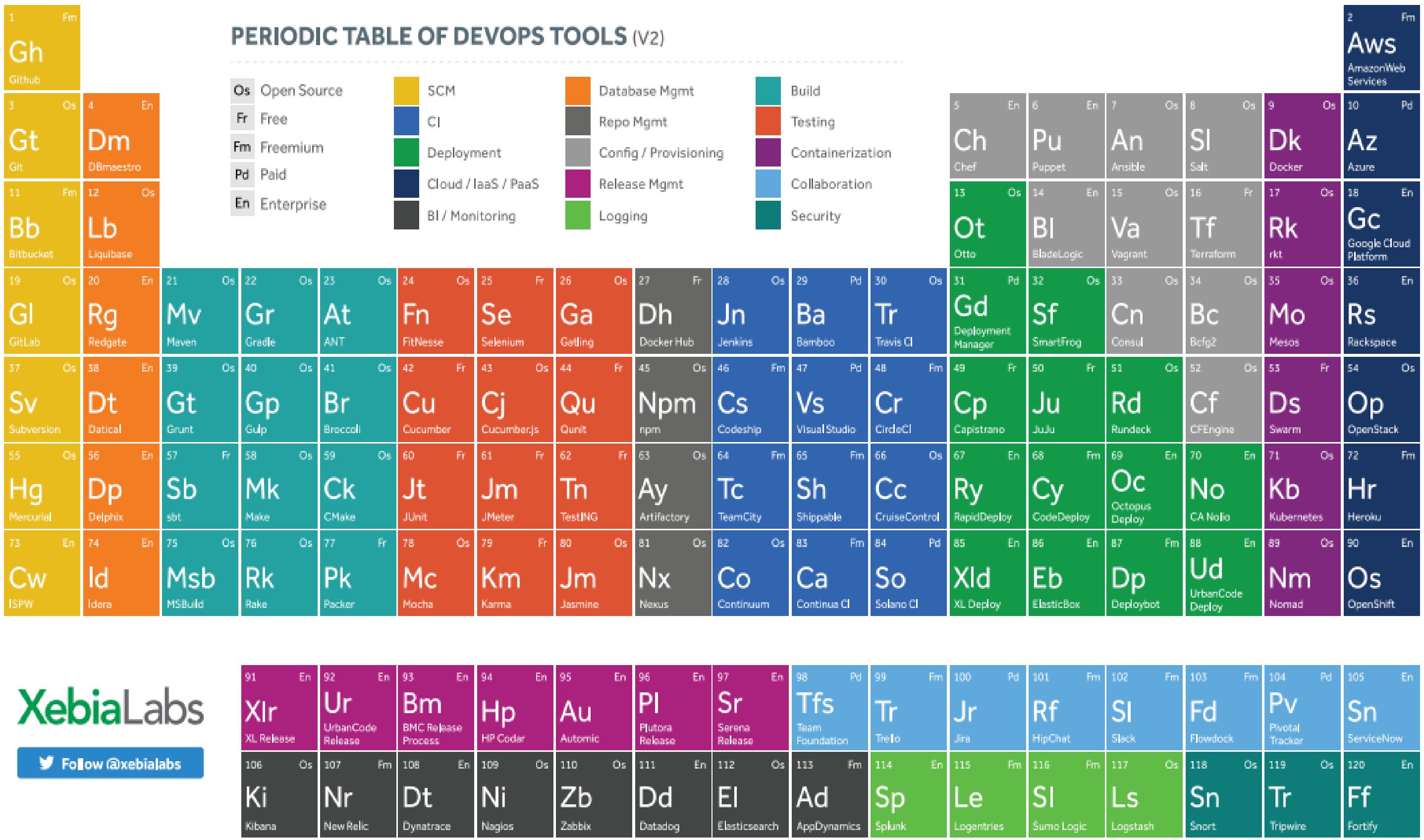
Our definition

- “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
- Infrastructure as Code (IaC)
- Automated testing
- Security testing
- Continuous monitoring
- Repository management
- Etc



PERIODIC TABLE OF DEVOPS TOOLS (V2)

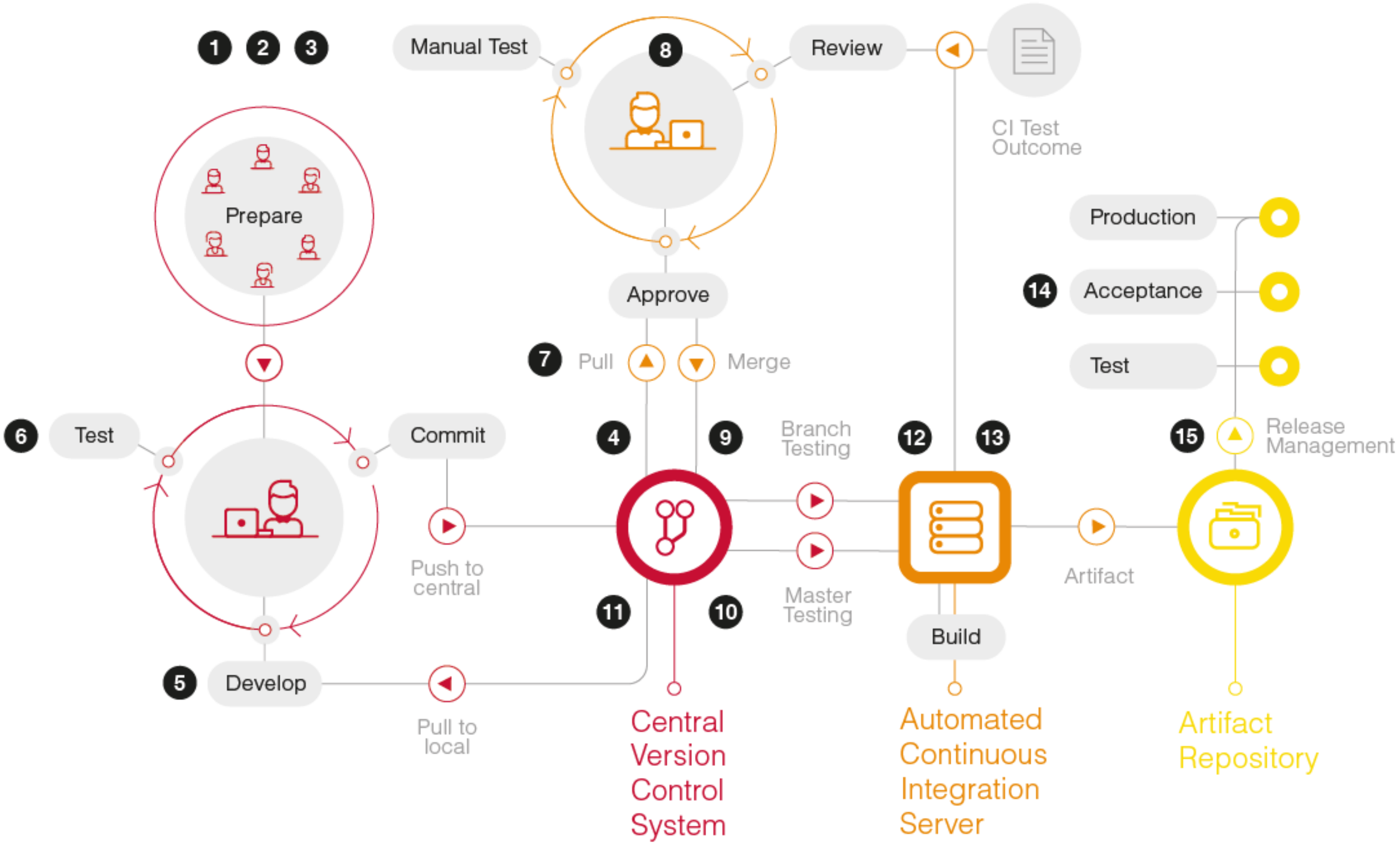
Os	Open Source	SCM		Database Mgmt		Build	
Fr	Free	CI		Repo Mgmt		Testing	
Fm	Freemium	Deployment		Config / Provisioning		Containerization	
Pd	Paid	Cloud / IaaS / PaaS		Release Mgmt		Collaboration	
En	Enterprise	BI / Monitoring		Logging		Security	

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 Bcfg2	35 Os Mo Mesos	36 En Rs Rackspace
37 Os Sv Subversion	38 En Dt Datical	39 Os Gt Grunt	40 Os Gp Gulp	41 Os Br Broccoli	42 Fr Cu Cucumber	43 Os Cj Cucumber.js	44 Fr Qu Qunit	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 Delphix	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 Nolio	71 Os Kb Kubernetes	72 Fm Hr Heroku
73 En Cw ISPW	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 En 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 Codar	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
			106 Os Ki Kibana	107 Fm Nr New Relic	108 En Dt Dynatrace	109 Os Ni Nagios	110 Os Zb Zabbix	111 En Dd Datadog	112 Os El Elasticsearch	113 Fm Ad AppDynamics	114 En Sp Splunk	115 Fm Le Logentries	116 Fm Sl Sumo Logic	117 Os Ls Logstash	118 Os Sn Snort	119 Os Tr Tripwire	120 En Ff Fortify

XebiaLabs

Follow @xebialabs

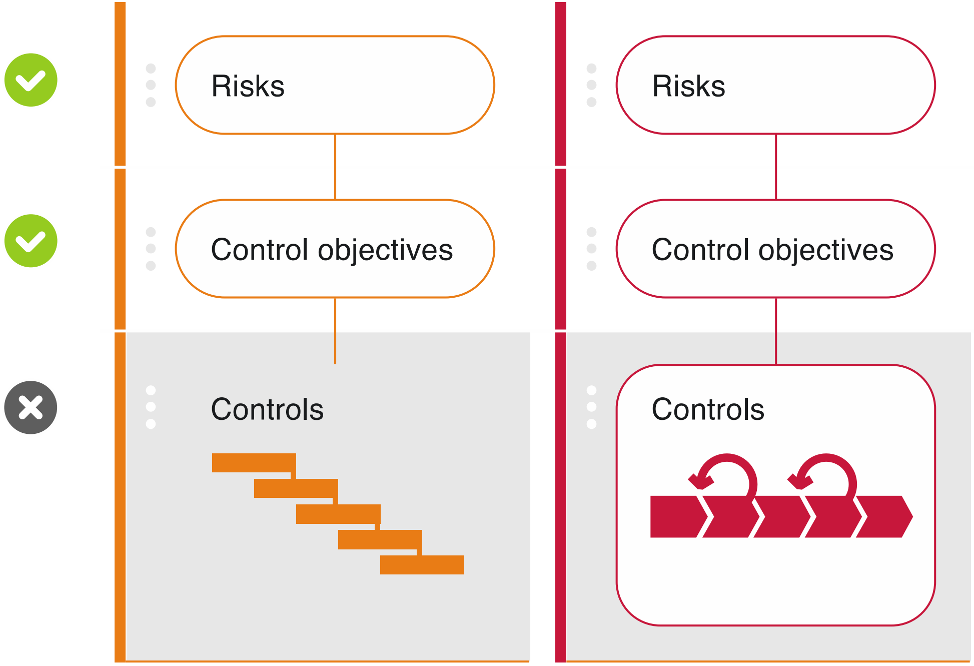
The control framework



NOREA
DE BEROEPSORGANISATIE VAN IT-AUDITORS

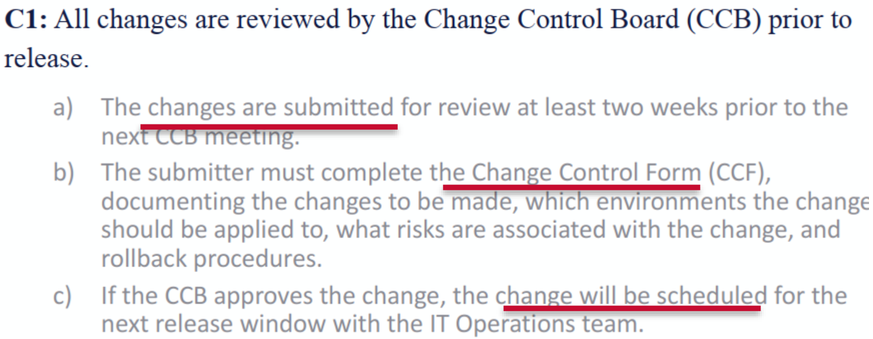
What changed?

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



NOREA
DE BEROEPSORGANISATIE VAN IT-AUDITORS

Example

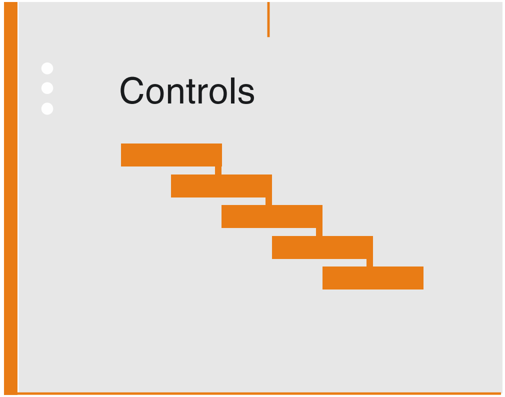


C1: All changes are reviewed by the Change Control Board (CCB) prior to release.

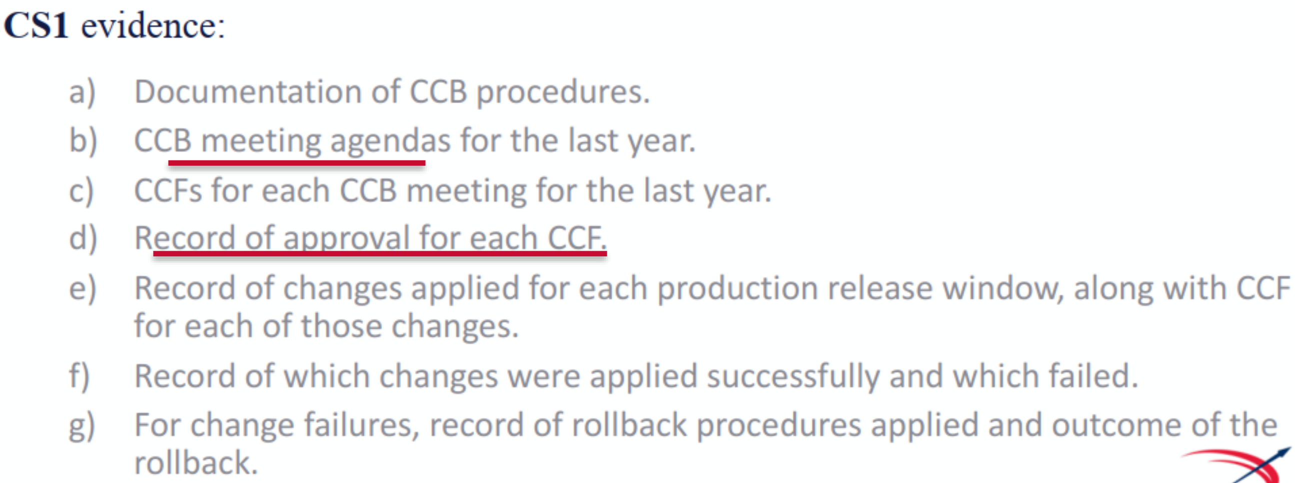
a) The changes are submitted for review at least two weeks prior to the next CCB meeting.

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.



Controls



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.

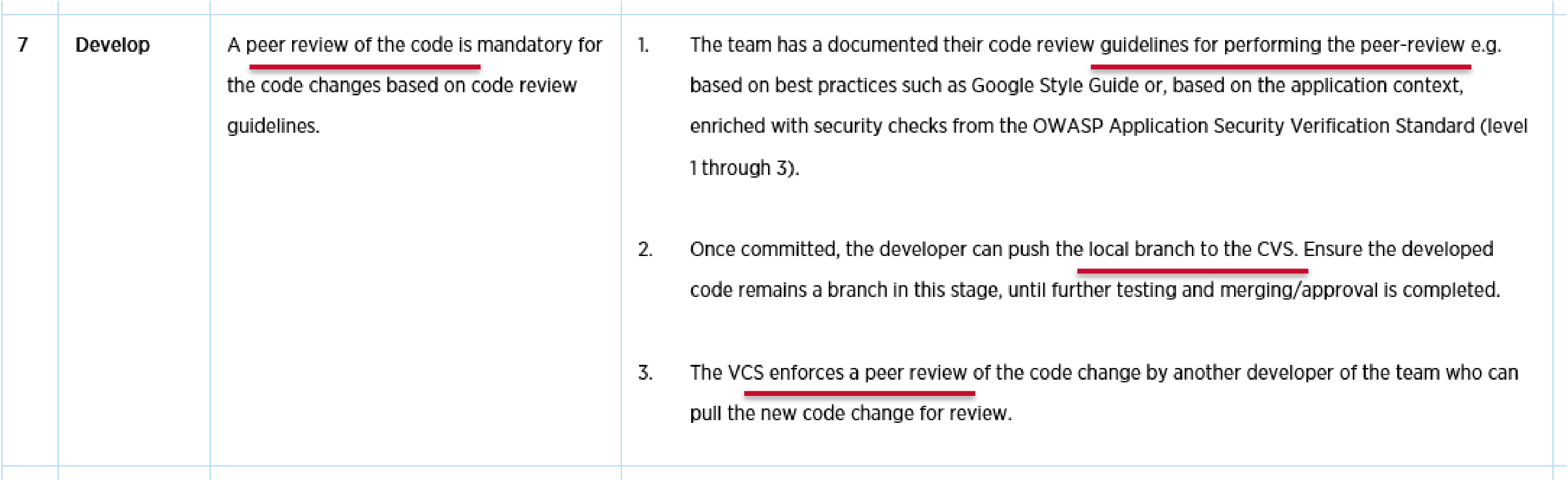
e) Record of changes applied for each production release window, along with CCF for each of those changes.

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.

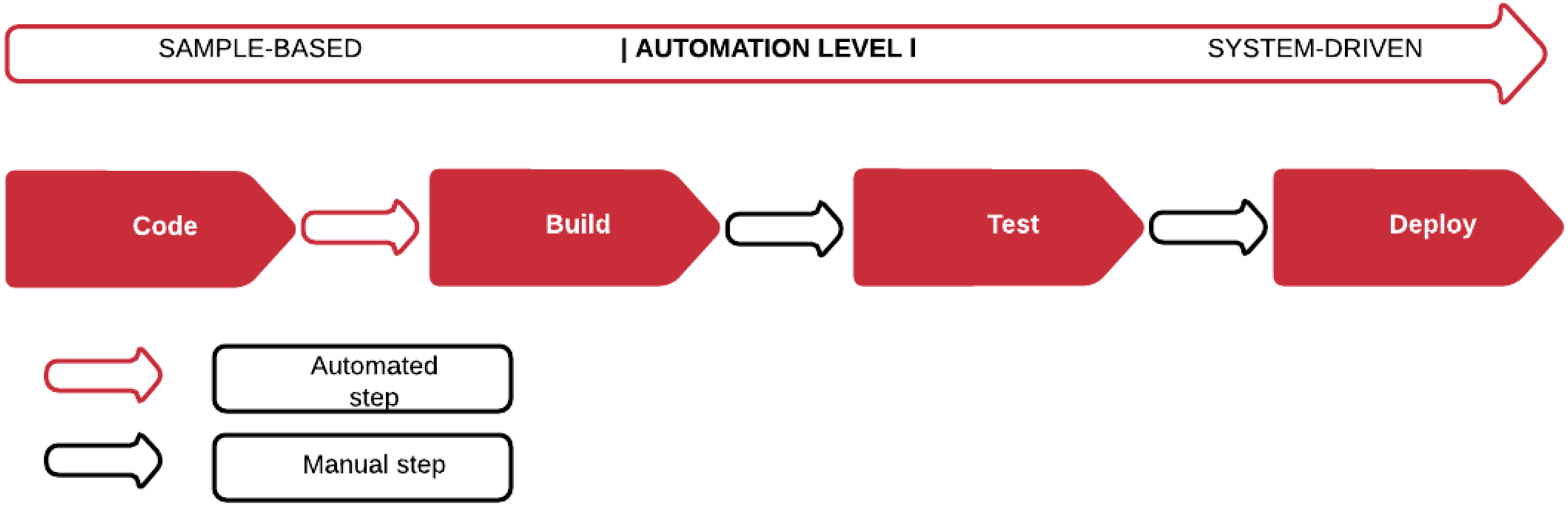
Example cont’d

Controls

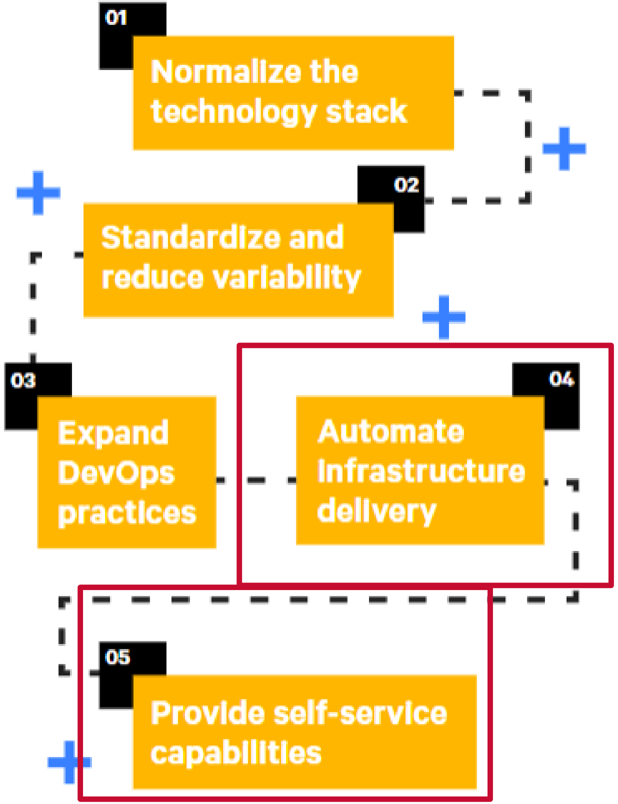


7	**Develop**	A peer review of the code is mandatory for the code changes based on code review guidelines.	1. The team has a documented their code review guidelines for performing the peer-review e.g. based on best practices such as Google Style Guide or, based on the application context, enriched with security checks from the OWASP Application Security Verification Standard (level 1 through 3). 2. Once committed, the developer can push the local branch to the CVS. Ensure the developed code remains a branch in this stage, until further testing and merging/approval is completed. 3. The VCS enforces a peer review of the code change by another developer of the team who can pull the new code change for review.

System–driven versus sample–based



Rome wasn’t built in a day



01 Normalize the technology stack

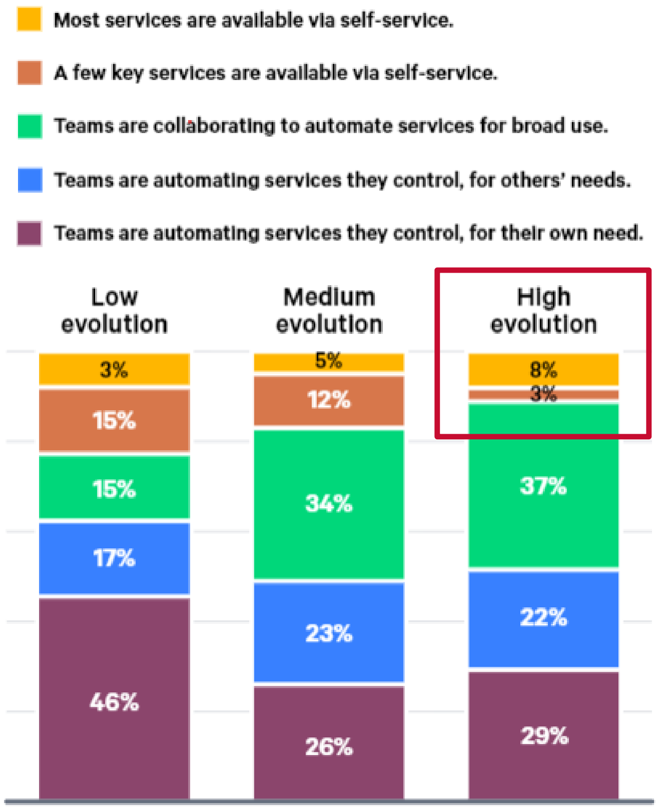
02 Standardize and reduce variability

03 Expand DevOps practices

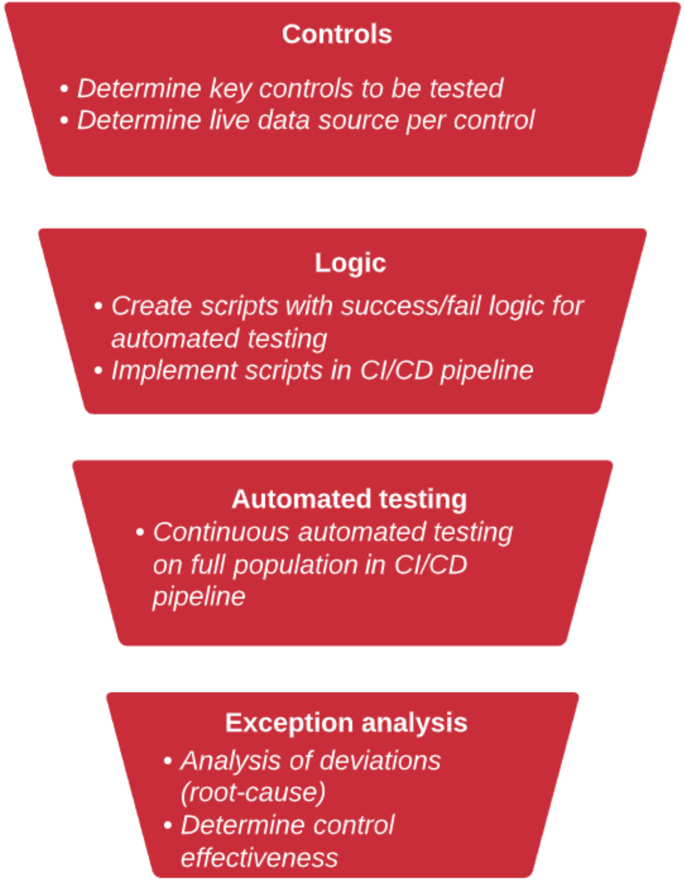
04 Automate Infrastructure delivery

05 Provide self-service capabilities

Automation progress by evolutionary scale



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*

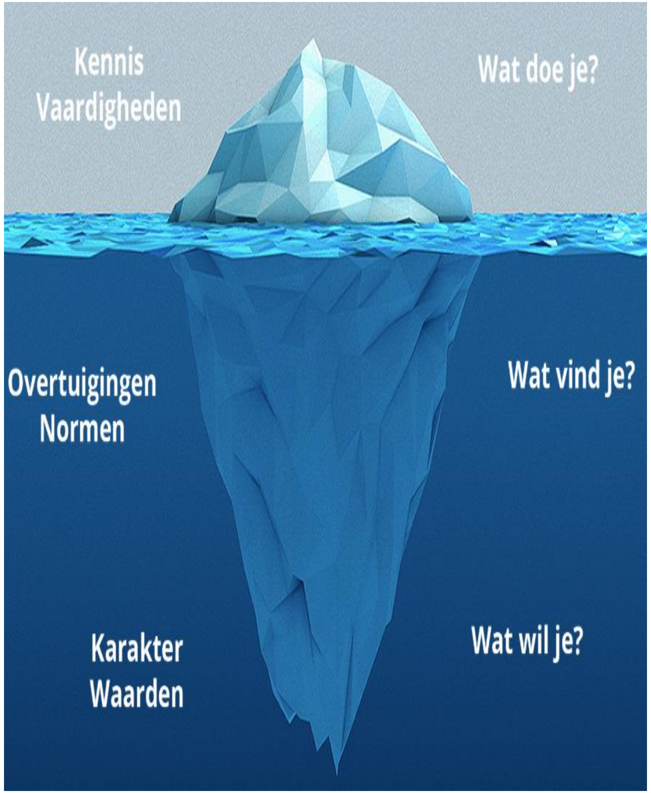
Automated testing

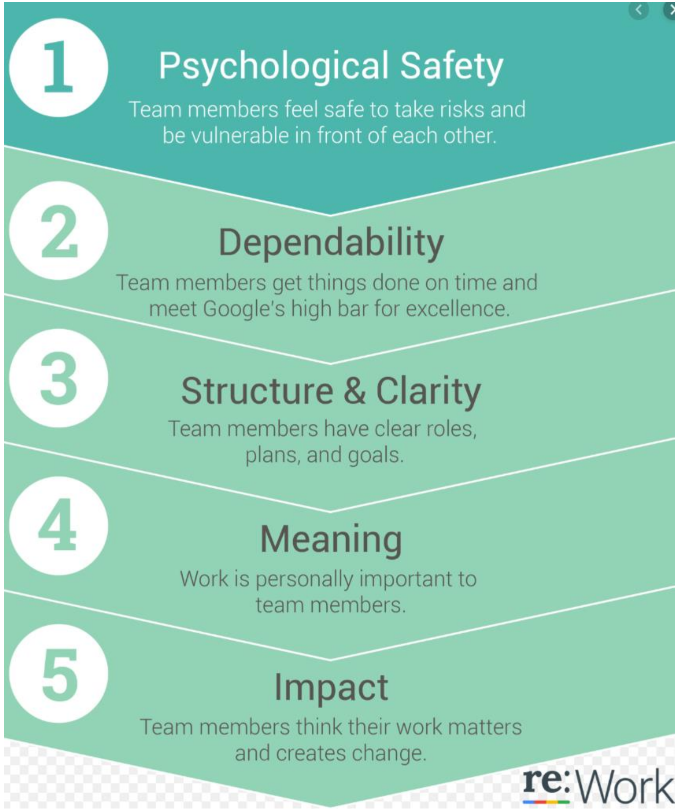
- *Continuous automated testing on full population in CI/CD pipeline*

Exception analysis

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

Cultural principles (Google project Aristotle)





1 **Psychological Safety**
Team members feel safe to take risks and be vulnerable in front of each other.

2 **Dependability**
Team members get things done on time and meet Google's high bar for excellence.

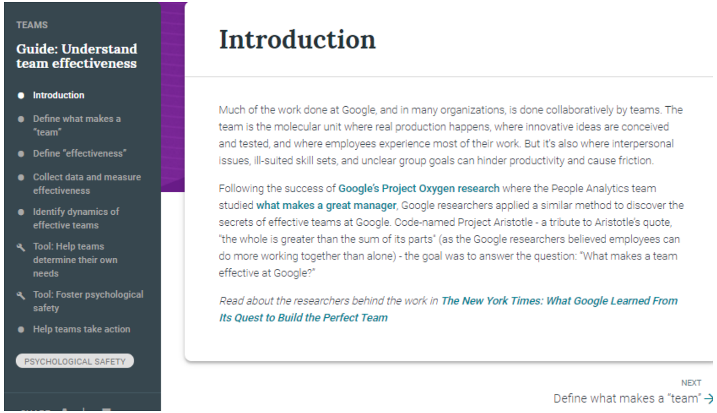
3 **Structure & Clarity**
Team members have clear roles, plans, and goals.

4 **Meaning**
Work is personally important to team members.

5 **Impact**
Team members think their work matters and creates change.

re:Work

Examples of tools to measure



TEAMS

Guide: Understand team effectiveness

- 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
- Help teams take action

PSYCHOLOGICAL SAFETY

Introduction

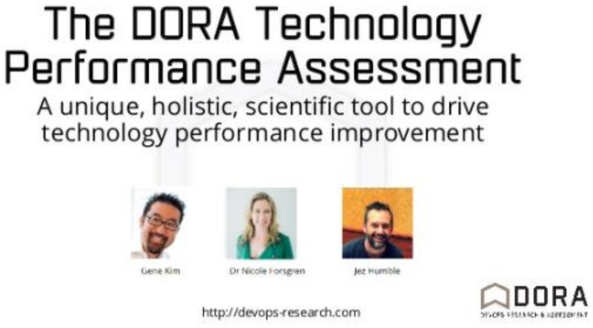
Much of the work done at Google, and in many organizations, is done collaboratively by teams. The team is the molecular unit where real production happens, where innovative ideas are conceived and tested, and where employees experience most of their work. But it's also where interpersonal issues, ill-suited skill sets, and unclear group goals can hinder productivity and cause friction.

Following the success of **Google's Project Oxygen research** where the People Analytics team studied **what makes a great manager**, Google researchers applied a similar method to discover the secrets of effective teams at Google. Code-named Project Aristotle - a tribute to Aristotle's quote, "the whole is greater than the sum of its parts" (as the Google researchers believed employees can do more working together than alone) - the goal was to answer the question: "What makes a team effective at Google?"

Read about the researchers behind the work in ***The New York Times: What Google Learned From Its Quest to Build the Perfect Team***

NEXT
Define what makes a "team" →

GOOGLE



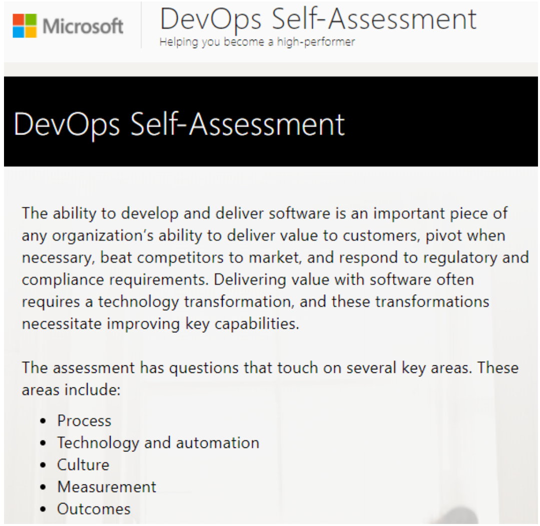
The DORA Technology Performance Assessment

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

Gene Kim
Dr Nicole Forsgren
Jez Humble
DORA

http://devops-research.com

DORA



Microsoft | DevOps Self-Assessment
Helping you become a high-performer

DevOps Self-Assessment

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. Delivering value with software often requires a technology transformation, and these transformations necessitate improving key capabilities.

The assessment has questions that touch on several key areas. These areas include:

- Process
- Technology and automation
- Culture
- Measurement
- Outcomes

MICROSOFT

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

The full report

- https://www.norea.nl/handreikingen:

Scan me

- www.linkedin.com/in/sandeep-panday:

Scan me