

# Welcome to DevOps

# Who am I ?



2020

Elias Arellano  
"DevOps Engineer"



# Evaluation of the course

- Multiple choise test = **37 questions** -> **0,55 points each question**  
(20,35 total points) | **1h30 duration**

*Question type : What is .....? A,B,C,D -> The answers are explicitly written in the course*

- Practice work =
  - TD Git commands and workflow -> 2 parts = 2 Bonus points for the Multiple choise test
  - TD CI/CD -> 3 parts -> 4 Bonus points for the Multiple choise test.

Part 1 – 2 point

Part 2 – 1 points

Part 3 – 1 points

All the practice work must be done here at school.

- Free time to study = 1h30 before the Multiple choise test
- **Timing**
  - 8h Course
  - 12h Practice work
  - 1h30 Study
  - 1h30 Multiple choise test

# Requirements

- Practice work – Preparing the VM is suggested:

- TD :

[https://drive.google.com/drive/folders/1eSg01rjL\\_HaJrQbemKIXGZYqzP8dqig?usp=share\\_link](https://drive.google.com/drive/folders/1eSg01rjL_HaJrQbemKIXGZYqzP8dqig?usp=share_link)

- Course :

[https://drive.google.com/drive/folders/1077m7R3VBkV0qJXv0S6dOtN6fHyocDVa?usp=share\\_link](https://drive.google.com/drive/folders/1077m7R3VBkV0qJXv0S6dOtN6fHyocDVa?usp=share_link)

- Github account
- Docker account

# Summary

- What is DevOps ?
- Management-Humain and DevOps
- Tools – Technical concepts and DevOps
- Conclusion

# What is DevOps?

# What is NOT DevOps?

Not a methodology

Not a job title

Not a framework

Not a technology

Not only a culture

Not only automation

Not a tool or set of tools

DevOps Engineer does not mean nothing, as Agile engineer too.

# What is DevOps?

- This question will be answered with all the following slides.
- DevOps is a **culture and a set of best practices** applied in an IT team (even in a company) to deliver good quality products and services in a **fast and easy way**. Where the **well-being, communication and progress of team members are important**.

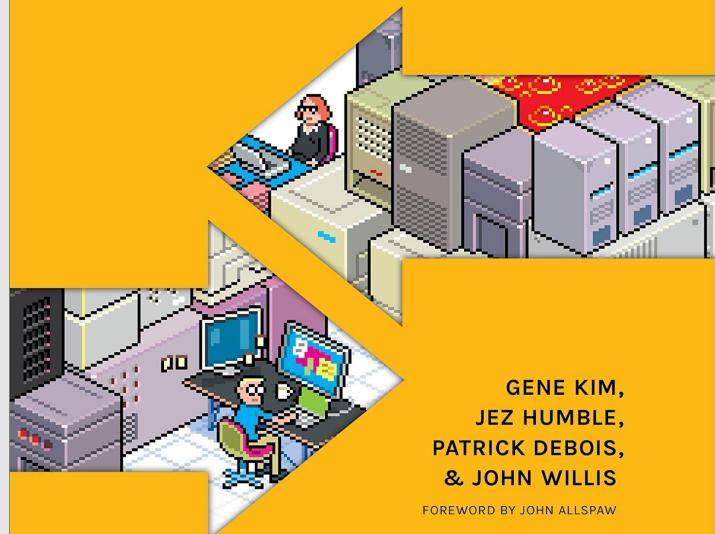
*Elias  
Arellano*

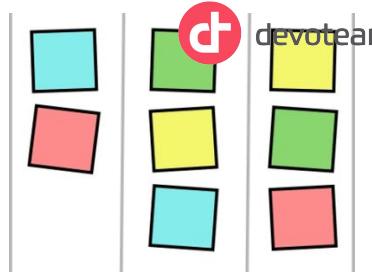
“Imagine a world where product owners, Development, QA, IT Operations and infosec **work together, not only to help each other, but also to ensure that the overall organization succeeds**. By working towards a common goal, they enable the fast flow of planned work into production, **while achieving world-class stability, reliability, availability and security**”

The

# DevOps Handbook

HOW TO CREATE WORLD-CLASS AGILITY, RELIABILITY, & SECURITY IN TECHNOLOGY ORGANIZATIONS





Communication



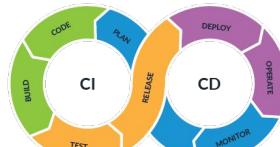
Well-being



Innovation



Sharing



Learning



Progress



Jenkins



# Why DevOps ?

- Comfortable for Developers and Operations
- Comfortable for business
- Comfortable as “culture”
- Consequence :
  - Companies & Teams like DevOps
- But:
  - Good but NOT THE BEST for all the cases

# Why DevOps today?

- Time to value must accelerate
- Migrating to cloud , Cloud infrastructure and Agile software are increasing.
- The data becomes bigger
- IT can no longer work in a silo culture

# A bit of history

Belgium - 2006



Patrick Dubois

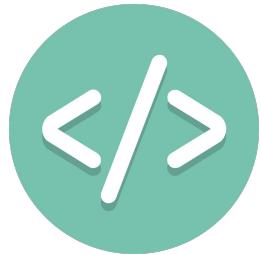
**Creator of the word DevOps**



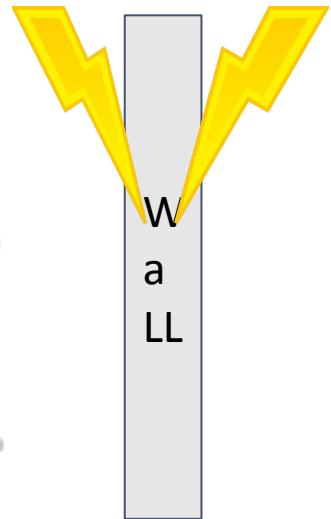
Migration of  
datacenters



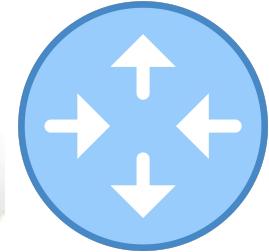
Dev - Development



Functionalities



Ops - Operations



Stability

Misunderstanding - Confusion

Differents Goals

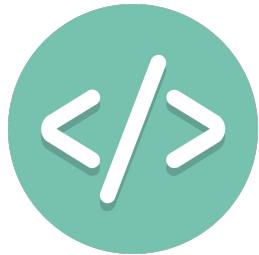
2008 - Toronto conference “Agile infrastructure” By Andrew Shafer



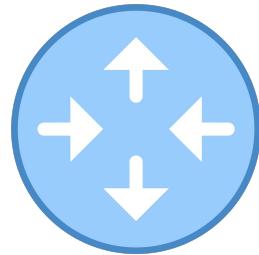
But only one guy was in the conference  
(Literally only one guy )

Then ..... DevOps community begun to grow

Dev - Development



Ops - Operations



Sharing

Learning

Same goals

# DevOps Stakeholders

- Dev includes all the people in charge of developing software and services  
(Architects, Quality assurance – QA, testers, customers, etc)
- Ops include all the people in charge of managing, delivering and maintaining software and services  
(Network engineers, system administrators, IT operations engineers, database administrators –DBAs, etc)
- DevOps is not only about Dev and Ops



# Not Only for Dev & Ops

In DevOps we include everybody

It touches every aspect of IT



And this is evolving

# Books DevOps

DevOps does not have a single source of knowledge

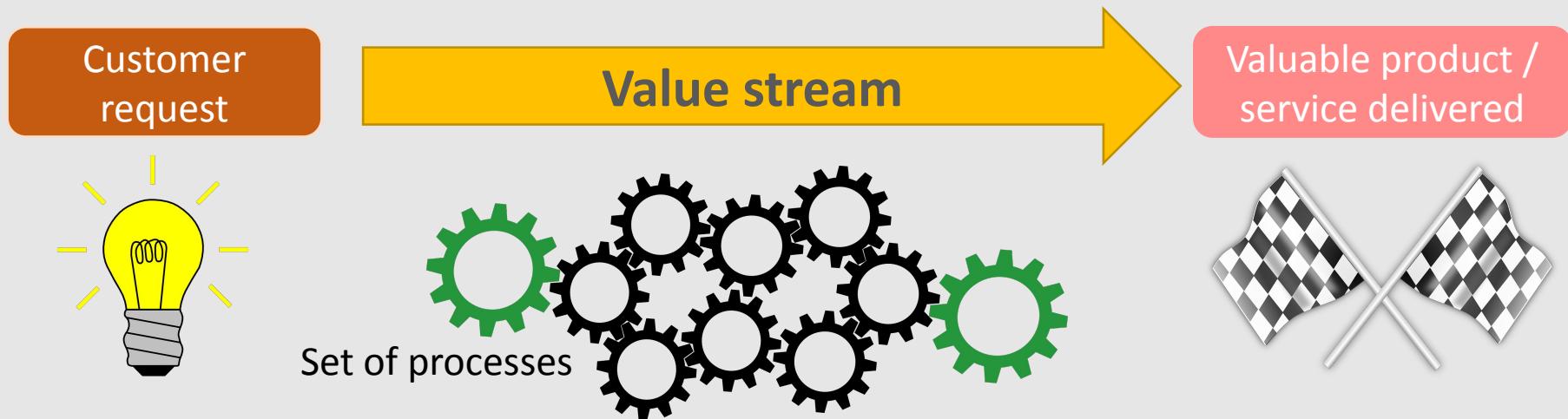
DevOps is built on the collective experience shared in books, webinars, conferences



# **Management-Humain and DevOps**

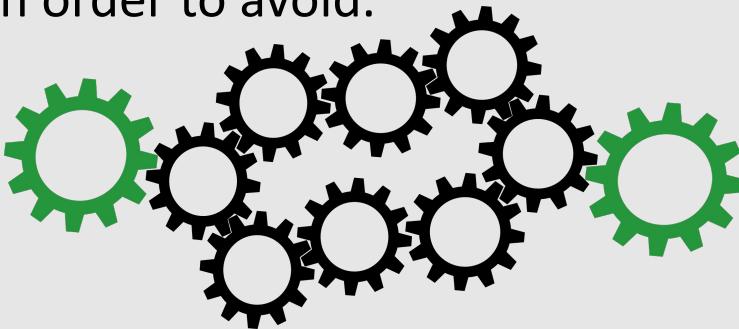
# Value Stream

- Value stream is a concept coming from LEAN.
- This is the whole end-to-end process that delivers value to the customers.



# Value Stream Mapping?

- How can we improve the process in order to avoid.
  - Waste of time
  - Waste of resources
  - Repeated manually some tasks
  - Errors and faults
- Identify non-values area.
- In DevOps , we must map the process by analysing and designing the flows.
- Don't forget to improve.

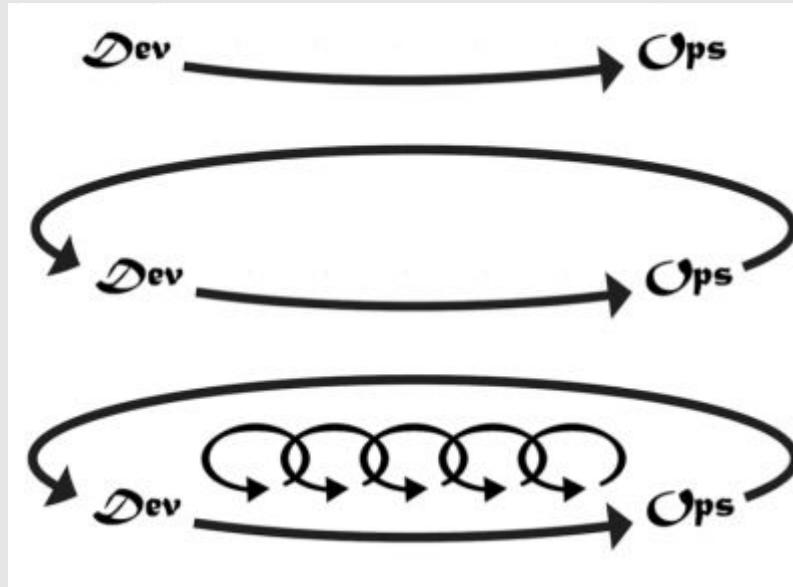


# Example of elements to improve in value streams

- Pipelines (Continuous deployment , Continuous Integration, Continuous delivery)
- Orchestration
- Automation
- Improved testing tools.
- Improved stability of the flows.

# The three ways

- The three ways describe **a set of actions** that Dev and Ops must take in order to apply the DevOps principals



# The First way

- From Dev to Ops
- Improve/Increase the flow by
  - Understanding constraints
  - Removing constraints
- Don't ignore known defects
- Understand the entire system



# Theory of constraint

Every process has at least one constraint or bottleneck that affects the capacity to meet the goals

It's important to identify the constraints and resolve them in order to meet the goal

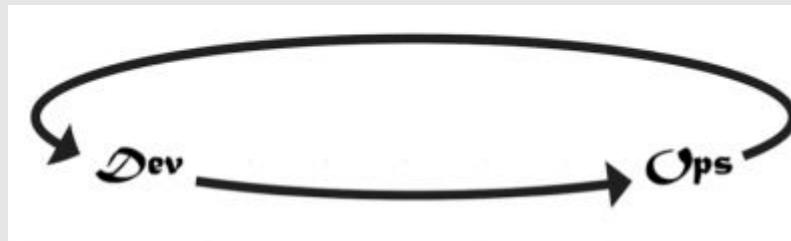


# Common constraints

- Development delays
- Code deployments
- Difficult bureaucratic process
- Security
- Environment's installation (Test, staging, pre-prod)
- Product management

# The second way

- Sending feedbacks related to the products/services from customers (internal or external customers)
- Sending feedbacks should be easy to do
- No secrets, the feedbacks should be easy to see and catch (dashboards)
- Very important to improve products and everything around

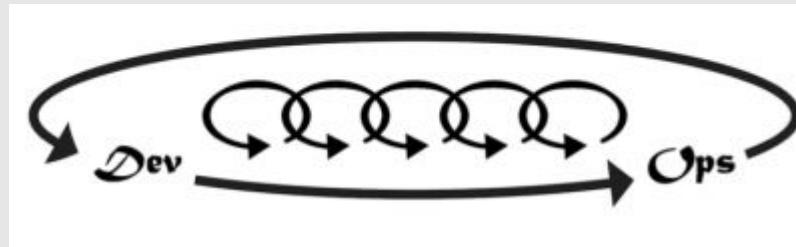


# Example of feedbacks

- Monitoring
- Automated tests
- Dashboard
- Measurements and metrics

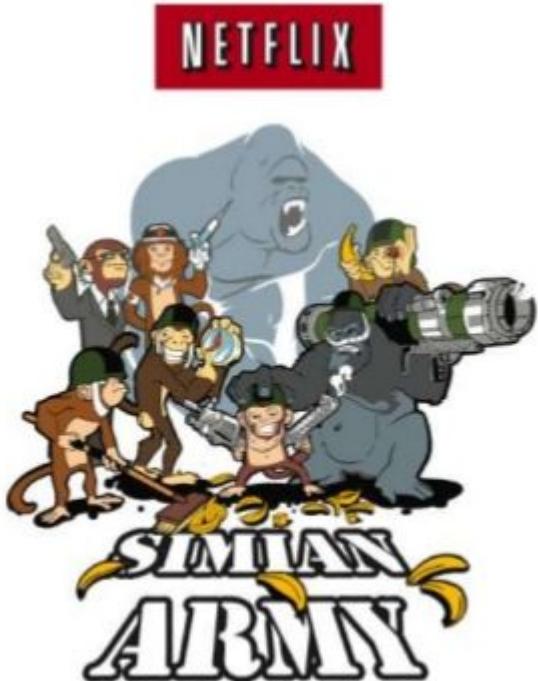
# The third way

- Continual learning and experimentation
- Learn from failure
- Reserve time to improve something in the daily work,
- Reserve time for experimentation
- Take the risk – Introduce faults in the system to increase resilience
- Plan





# Chaos Monkey



The Chaos monkey is a tool ( free and open source) that randomly disables the instances in production to make sure that the company can survive without any customer impact.

Conditions to use this

- Running in business days during the work day,
- Engineers must be prepared.



Useful because

- You identify the weakness of the system.
- Prepare a system that can recover automatically the service in production

# DevOps Values : CALMS

Culture

Automation

Lean

Measurement

Sharing

# Culture

- Good behaviour
- Improve the relationship and communication with your team
- Ask yourself about your behaviour, how to improve this in order to respect a DevOps culture

# DevOps culture in a nutshell

- Recognition
- Collaboration
- Respect
- Trust
- Transparency
- Continuous improvement
- Continuous learning
- Safe

# Maybe you must change



People don't resist  
change. They resist  
being changed

You can't change people,  
they can only change by  
themselves.

Peter senge

# Automation

- If a task (or a set of tasks) is repeated, you should think about automating it.
- This feature is part of all the DevOps tools
  - Control tools (Orchestration – Monitoring – Configuration tools)
  - Management tools (ITSM)
  - etc.
- Avoid humains errors

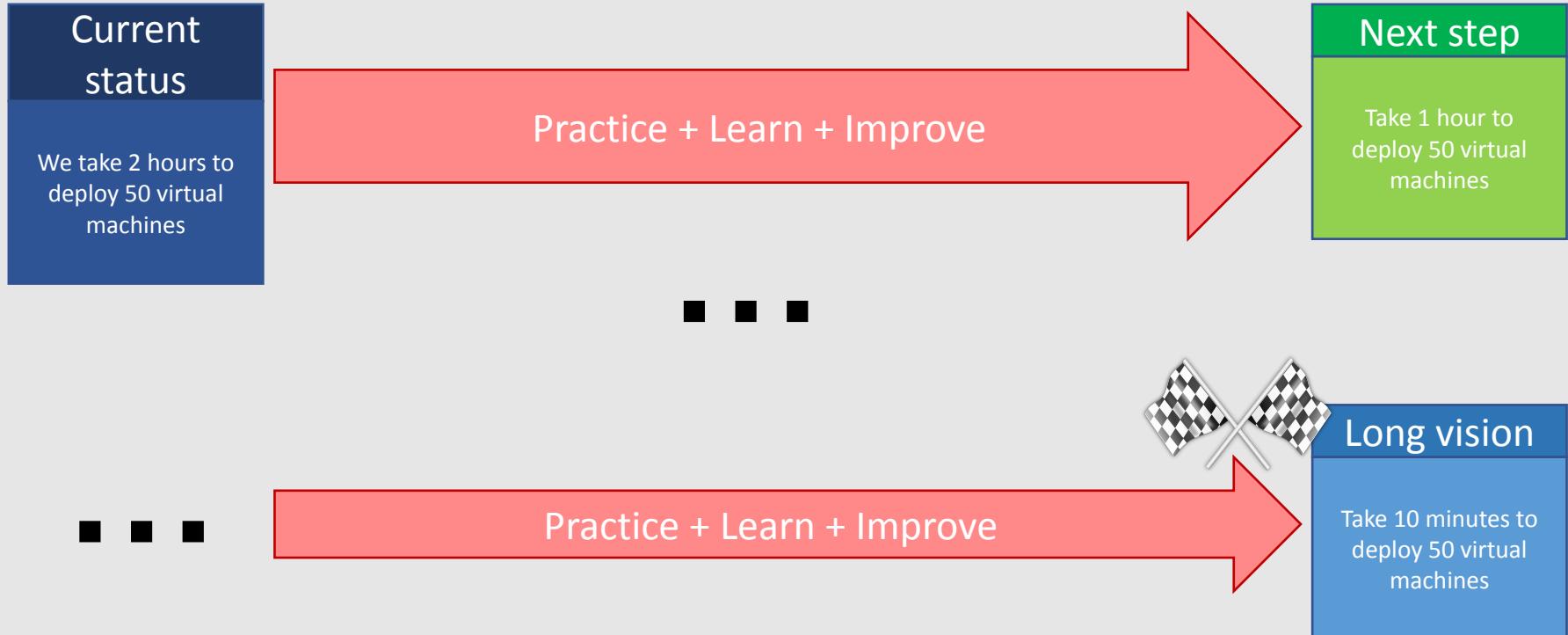
# Lean

- Delete waste (time, resources, energy, etc), improve flow and process.
- Maximise customer value.

# Improvement KATA

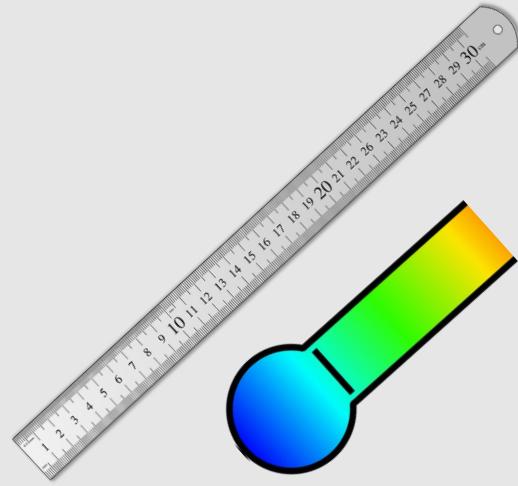
- Martial arts term
- Kata is a way of thinking and acting that you repeat activities until it becomes a habit.
- In DevOps this is part of continuous learning, continuous improvement and Lean.
- Members define a long-term vision that must be accomplished. From the current status, we define the little goals to achieve until to get the main goal.
- The goal is an improvement.

# KATA : example



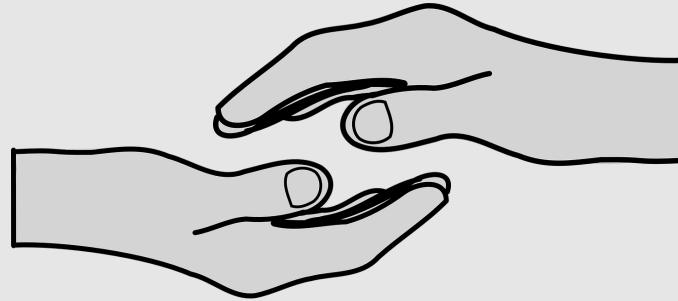
# Measurement

- We must measure everything as much as we can
- If you can't measure, you can't improve.
- Metrics , Analysis tools
- Monitoring tools, BUT Monitoring is not equal to measure -> Some Monitoring data can measure,
- Monitoring : Check the status of the system.
- Measure : Compares data and gives a result



# Sharing

- Be open to all around you.
  - Share what you know to do.
  - Teach to your around
- 
- You always have something to share, even so tinies ideas.
  - Share the problems encountered -> Send feedbacks if you receive them.



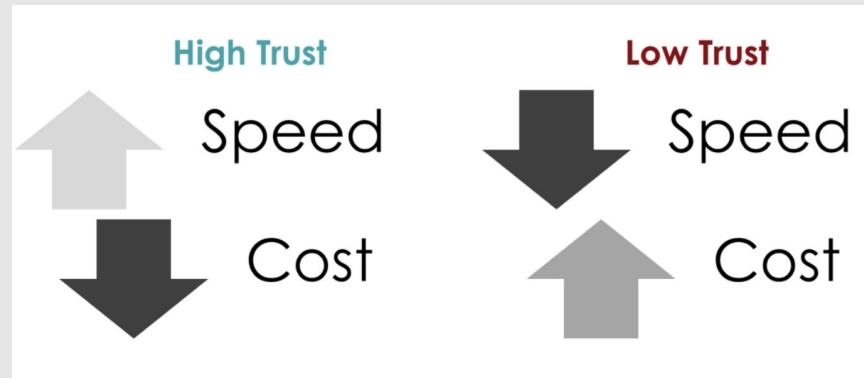
# Continuous learning

- Be curious, the world is changing everyday
- This is also a way to :
  - Improve your projects
  - Improve yourself
- Learning sessions within the team:
  - Establish a learning session in the sprints
  - Participate in conferences or watch it.

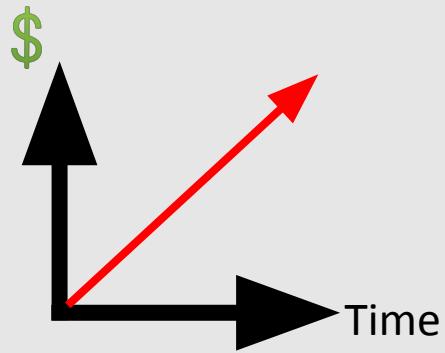


# DevOps : High Trust

- High trust encourages to:
  - Collaborate with anyone
  - Shared responsibilities
  - Good information flow
  - Take action quickly
- Avoid hiding information
  - More the people have access to the information, better is the relationship between people.
  - Be transparent.



# Without DevOps



Innovation



Dev



Ops

# With DevOps



Winning  
through  
Innovation



Dev



Ops

# Agile – The Manifesto

Remember Agile values

- Interactions & individuals
- Working software
- Customer collaboration
- Responding to change



- Tools and processes
- Long and Comprehensive documentation
- Following plans and procedures
- Contract & Negotiations



# Scrum

Scrum is a simple framework for effective team collaboration on complex projects. Scrum provides a small set of rules that create « just enough» structure for teams to be able to focus their innovation on solving what might otherwise be an insurmountable challenge.

*Scrum.org*

- Today Scrum is the most used framework for Agile developments.
- Scrum helps to deliver products quickly and efficiently.
- Scrum makes people to be more related and opened each other.

# Scrum

## Roles

- Product Owner
- Scrum Master
- Development Team

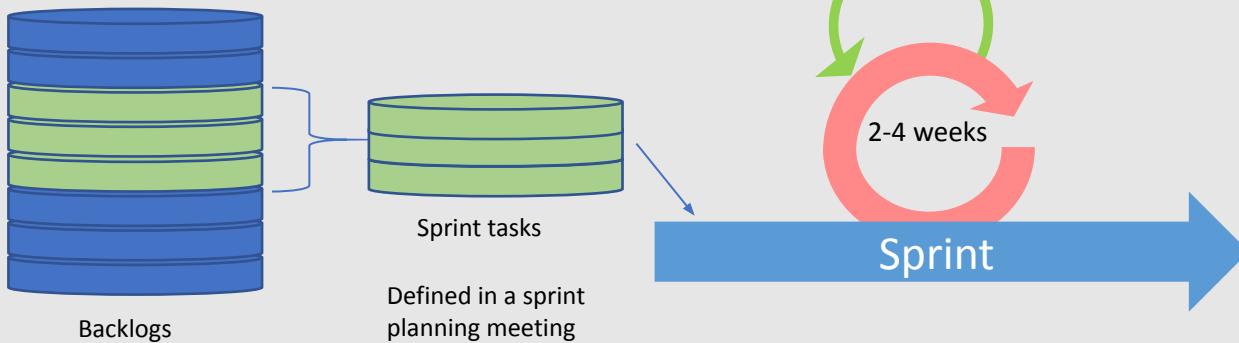
## Artifacts (Objects)

- Backlog
- Sprint
- Increment (done)

## Meetings

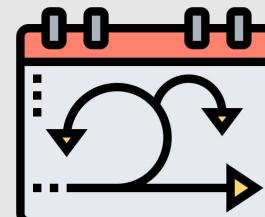
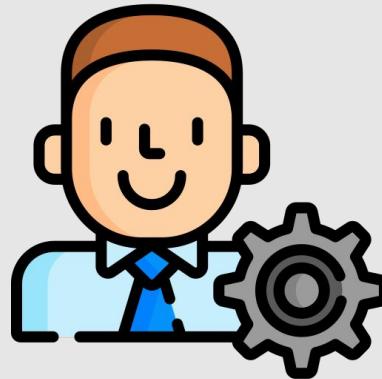
- The Sprint
- Sprint Planning
- Daily scrum
- Sprint review
- Sprint retrospective

## Where the Ops are involved?



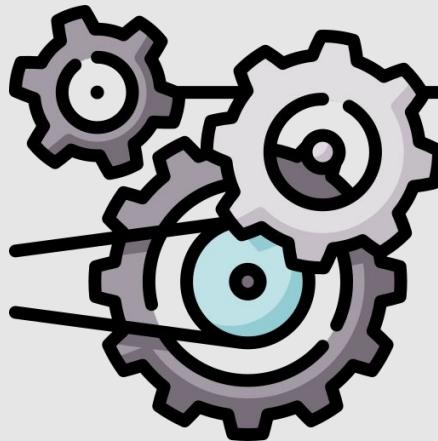
# The Ops are involved in the sprints

- 2 Possibilities
  - Meeting with the IT operators in some planned frequency in the sprints (Twice, three times per week, etc).
  - Have a representative from the IT operation team.
- In function of the features of the project , you should hire a Network expert in the team.
- Remember , be open minded, take the opportunity to learn different subjects.
- Doesn't matter if you are not an expert, all that you know can be important for the team.



# DevOps and automation

- DevOps is not about automation, just as astronomy is not about telescopes  
Christopher Little, quoted in the DevOps Handbook



# Why automation?

- Actions are done faster (releases, tests, configurations, recoveries, etc)
- Higher quality of anything
- Better security
- Fewer errors
- Result:
  - Happy customer



**Full DevOps engineer doesn't exist**

# **Tools – Technical concepts and DevOps**

# Why Cloud in DevOps?

- The cloud service depends on the provider (IaaS, PaaS, SaaS)
- Dynamic deployment of IT resources
- Fast and easy configuration
- Cheaper than on-premise
- Think on your business more than maintaining the infrastructure



# Why DevOps tools in DevOps

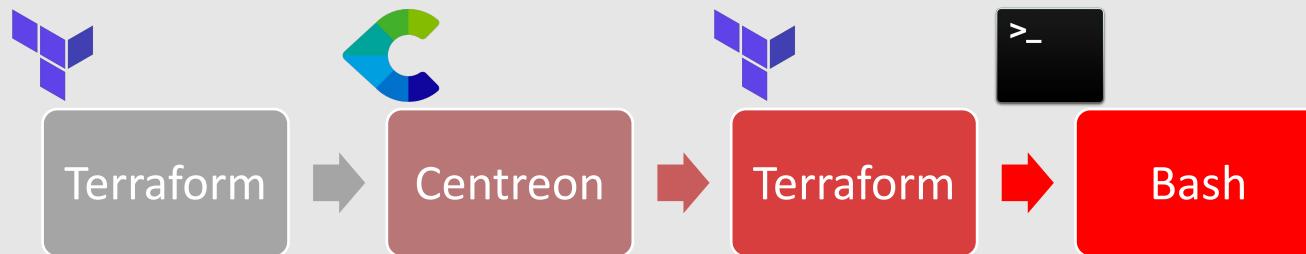


- DevOps tools are tools that are the most adapted for applying DevOps principles
- Don't take care if the tool is not a DevOps tool, take care if you use it as a DevOps tool.



# DevOps toolchain

- A DevOps toolchain is a set of IT tools synchronized each other in order to automate processes.
- Don't build a toolchain thinking that it's a definitive version, the toolchain should be modified in function of the time, updates, requirements, etc.



# DevOps toolchain : Elements

- Orchestration
- Virtualization
- Automated tests
- Monitoring
- Version control
- Automated deployment

# DevOps toolchain

- The toolchain evolves in function of the time , there is no definitive model for it
- This is required for continuous improvement

# DevOps tools

- I apply DevOps principals, because I use these DevOps tools
- I apply DevOps principals, so I could use these DevOps tools
- We call DevOps tools to the set of tools that are known to be the most adapted to apply DevOps principals like.
  - Automation.
  - Stability.
  - Security.
  - etc.

# DevOps tools : The periodic table

<b>Aja</b> Atlassian Jira Align	<b>Tp</b> Digital.ai Targetprocess	<b>Pv</b> Planview	<b>In</b> Instana	<b>Dd</b> Datadog	<b>Ja</b> JFrog Artifactory	<b>Aws</b> AWS	<b>Sl</b> Slack	<b>Mt</b> Microsoft Teams	<b>Rha</b> Red Hat Ansible	<b>Ht</b> HashiCorp Terraform	<b>Dk</b> Docker	<b>Rho</b> Red Hat OpenShift	<b>Lb</b> Liquibase	<b>Dp</b> Delphix	<b>Ud</b> UrbanCode Deploy	<b>Sni</b> Sonatype Nexus IQ	<b>Aq</b> Aqua Security	<b>Cfr</b> CloudBees Flow	<b>Brl</b> BMC RLM	<b>Gls</b> GitLab SCM	<b>Gi</b> Git
<b>Alo</b> Atlassian Align	<b>Tpa</b> Digital.ai Targetprocess	<b>Pvw</b> Planview	<b>Ina</b> Instana	<b>Dda</b> Datadog	<b>Jaa</b> JFrog Artifactory	<b>Awsa</b> AWS	<b>Sla</b> Slack	<b>Mta</b> Microsoft Teams	<b>Rhaa</b> Red Hat Ansible	<b>Hta</b> HashiCorp Terraform	<b>Dka</b> Docker	<b>Rhoa</b> Red Hat OpenShift	<b>Lba</b> Liquibase	<b>Dpa</b> Delphix	<b>uda</b> UrbanCode Deploy	<b>sni</b> Sonatype Nexus IQ	<b>aq</b> Aqua Security	<b>cfr</b> CloudBees Flow	<b>brl</b> BMC RLM	<b>gls</b> GitLab SCM	<b>gi</b> Git
<b>AlOps/Analytics</b>	<b>Artifact/Package Management</b>	<b>Cloud</b>	<b>Collaboration</b>	<b>Configuration Automation</b>	<b>Containers</b>	<b>Enterprise Agile Planning</b>	<b>Issue Tracking/ITSM</b>	<b>Release Management</b>	<b>Deployment</b>	<b>Database Management</b>	<b>Testing</b>	<b>Value Stream Management</b>	<b>Continuous Integration</b>	<b>Security</b>	<b>Serverless/PaaS</b>	<b>Source Control Management</b>	<b>Digital AI App Protection</b>	<b>Digital AI Release</b>	<b>AWS CodePipeline</b>	<b>Github</b>	
<b>Os</b>	<b>Fr</b>	<b>Fm</b>	<b>En</b>	<b>Pd</b>	<b>En</b>	<b>En</b>	<b>En</b>	<b>Os</b>	<b>Os</b>	<b>Os</b>	<b>Os</b>	<b>Os</b>	<b>En</b>	<b>En</b>	<b>En</b>	<b>En</b>	<b>En</b>	<b>En</b>	<b>Os</b>		
<b>Open-source</b>	<b>Free</b>	<b>Freemium</b>	<b>Paid</b>	<b>Enterprise</b>																	

<https://digital.ai/periodic-table-of-devops-tools>

<b>Jn</b> Jenkins	<b>Azc</b> Azure DevOps Code	<b>Glc</b> GitLab CI	<b>Tr</b> Travis CI	<b>Cc</b> CircleCI	<b>Mv</b> Maven	<b>Ab</b> Atlassian Bamboo	<b>Gd</b> Gradle	<b>Acb</b> AWS CodeBuild	<b>Aj</b> Atlassian Jira	<b>Bi</b> iHHC Helix ITSM	<b>At</b> Atlassian Trello	<b>Sw</b> ServiceNow	<b>Td</b> TOPdesk	<b>Pd</b> PaperDuty
<b>Tt</b> Tricentis Tosca	<b>Nn</b> Neotys NeoLoad	<b>Se</b> Selenium	<b>Ju</b> JUnit	<b>Sl</b> Saucelabs	<b>Ct</b> Compuware Topaz	<b>Ap</b> Appium	<b>Sq</b> Squash TM	<b>Cu</b> Cucumber	<b>Jm</b> JMeter	<b>Pa</b> Parasoft	<b>Dai</b> Digital.ai Desktop	<b>Tp</b> Digital.ai Desktop	<b>Pr</b> Plutora	<b>Gl</b> GitLab

# ITSM

# servicenow

- IT service management (ITSM)
- System that allows to manage any information regarding to the IT park of the environment (company, etc).
- Example of use in DevOps:
  - - Have an overview of the IT architecture
  - - Take Tickets from users
  - - Take Feedbacks from users
- In DevOps, we must know very well our IT parc, and to know what is happening, and its status



# ITIL

- IT infrastructure library
- It's a set of best practices to manage IT services within a company/organization
- Practices to add value to the IT services
- If DevOps is implemented in the company, you could check ITIL.
  - Continuous improvement
- You can be certified in ITIL, today we have ITIL.v4

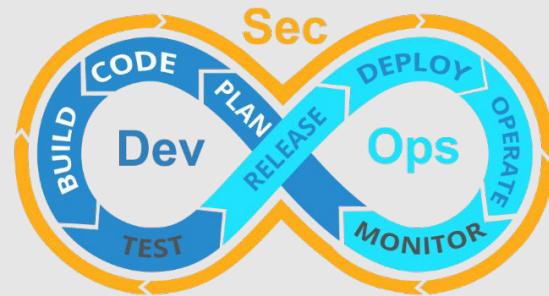
# Monitoring

- Everything must be monitored
  - Reaction of problems should be automatically applied
  - Alerts should be triggered automatically.
- 
- 1- The monitoring tool sends the alert and apply solution automatically
  - 2- If not resolved, mark the issue to be manipulated by the admin
- 
- Monitoring is useful to find failures, resolve them as soon as possible, then improve the product ( improve resilience)



# DevSecOps

- Its purpose is to build the mindset that « everyone is responsible for security ».
- Security as code.
- As everyone is responsible for security, the security should become easier to discuss, take decisions, etc.
- Break security constraints.
- Automate security actions for resilience, testing, detection and audits.



# FinOps

- This is the concept of analysing, monitoring and studying the costs that are spent on the cloud's resources in order to optimise the costs in function of the use of the cloud.



# DataOps

- It's just about handling data by applying DevOps principles
- Not a methodology
- Example :
  - Modeling in real time the incoming data (No stop for delivering results)
  - Automate the process of data treatment

# Safety culture

- DON'T BLAME to no one, just resolve it.
- Be fearless about reporting problems
- Be wise
  
- In DevOps , find ways, techniques and tools to prevent/predict incidents.



# ChatOps

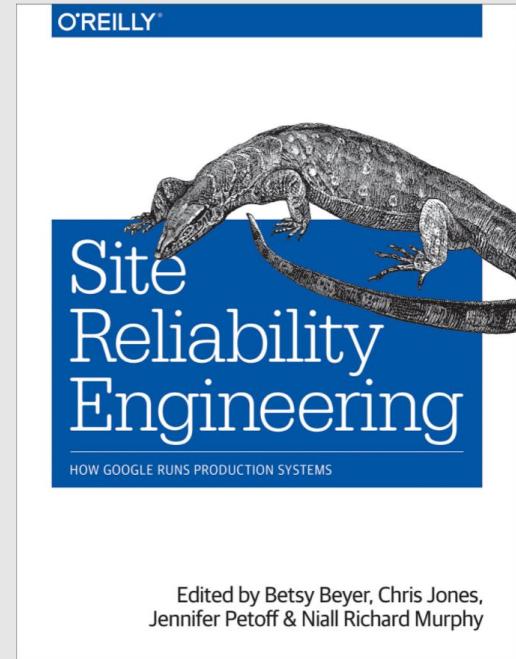
- This is the fact of providing a way to have communication with customers or team members.
- Conversation between humans or humans with bots
- The chat with the customers are used for
  - Support
  - Feedbacks
  - etc.
- The chat between team members is used for anything that regards the project

# Chat ops : example



# Site reliability engineering (SRE)

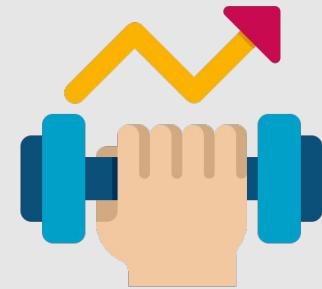
- Concept created by Ben Treynor Sloss from Google in 2003.
- Job title : Site reliability engineer.
- A SRE candidate is a developer that cares about stability and performance of the infrastructure. The programmer has a behavior, a culture and a set of practices that make him take care of the systems.



# Conclusion

# DevOps is useful because we can

- Deliver products/services fast
- Guarantee good quality
- Team members feel good
- Build more in less time
- Have the opportunity to learn
- Cover all the aspect of security
- Improve everything (even ourselves)



# Business value

- Why the Business is interested in DevOps?
  - Tech business is increasing (Most of the business counts a lot with IT).
  - Time to value is replacing to time to market.
  - Consumer thinks always about app solutions.



We need « DevOps engineers »

# DevOps and Covid-19

# DevOps and Covid-19

Products were deployed faster in a good quality

# Certification

- DevOps foundation



# Questions?