-

DEVOPS

(ST2DCD)

Final project

July 2023

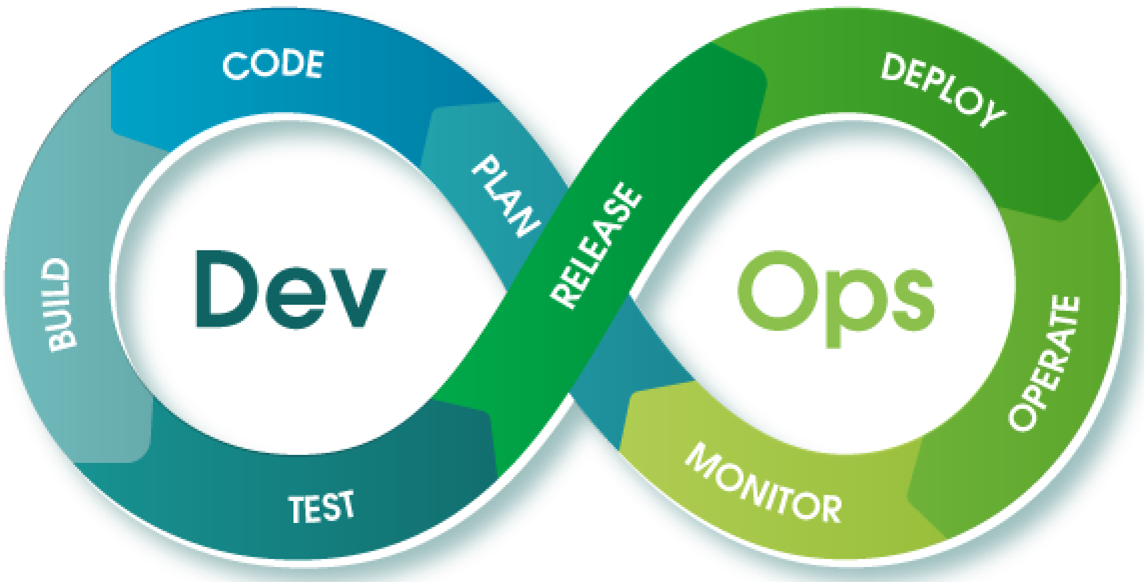

Table of Contents

1. INTRODUCTION

The client is **Agile Auto Parts (AAP)**, a major worldwide auto parts retailer. Its business has grown steadily for over 20 years. But since 2005 a decline in revenues has forced the CED to investigate and take actions to restore growth.

The board has clearly identified the IT department and services as the #1 suspect.

Several members of the board have heard about how DevOps has helped their rivals radically increase their Time to Market (TTM).

Thus they are open to adopt DevOps but would like specific input from you in order to be able to weigh the pros and cons.

Good luck!

2. CONTEXT

AAP is operating in 36 countries with 452 stores, nearly 6800 employees (staff, salespeople, personnel).

AAP is also continuously communicating with over 60 different partners (auto makers, parts manufacturers, other retailers...).

In 2018 the company reported 136 million EUR in revenues worldwide.

But the market and shareholders were very unhappy with this result.

Once a leader, now AAP is behind the main competitors.

The Time-To-Market is very bad (it is so awful they don't even want to communicate about it).

What's wrong then?

The board has identified the source of the problem: **the information system.**

The IT department has existed since the very beginning of AAP. At that time 2 developers coded the first system and were mostly doing all the work.

Now the situation has changed dramatically.

2.1. THE PROFILES

● Developers	:	28
● Integrators	:	2
● Designers	:	5
● Architects	:	2
● System administrators	:	3
● Database administrators	:	2
● Infrastructure technicians	:	5
● Release managers	:	3
● Security expert	:	1
● Chief Technology Officer	:	1

2.2. THE APPLICATIONS

- 1 public Web site for customers (online store) + backend

- 1 Web application for our stores (check parts availability, contact partners...)
- 1 standalone local application used in the retail stores
  both connected to the same backend

- 1 intranet Web application for HR purpose

2.3. ENVIRONMENTS

- Dev
- Integration
- User Acceptance Tests
- Security
- Performances
- Preproduction
- Production

2.4. INFRASTRUCTURE

- As of today, virtualization is a distant and vague concept at AAP. So nothing is virtualized.
- We are running and maintaining 21 physical servers (some are backup servers) :
  - HTTP (Apache and nginx)
  - Web (Tomcat)
  - Databases (MySQL and PostgreSQL)
  - Applications (PHP, Java EE, Java Swing)
  - WildFly application server
  - Debian, Ubuntu and CentOS

2.5. GEOGRAPHICAL LOCATIONS (DEVELOPERS AND SERVERS)

3 different sites: Bordeaux, Nantes and Brussels.

2.6. DELIVERY PROCESS

- When a release is due, each developer sends a zip file of the project on her/his workstation via *WeTransfer*
- There is no code repository
- The release managers get the different codes and do a manual merge
- Each deployment takes no less than 7 weeks
- There is never enough time to run all the tests, most of them are manual
- Environments are a not maintained and are often outdated

3. THE TARGET

Reach and maintain a TTM lower than our competitors'!

Apparently DevOps can help us achieve that goal.

To convince us, you need to provide a two-part reply to our RFP:

1) Write a **strategy roadmap** (document) in which you : (3,5)
   a. Make a blueprint of the existing system (resources, process, locations...)
   b. Make a blueprint of your target solution
      describe the architecture of your prototype
      explain the devops tools that you propose

2) Address our main concerns (see next section) (8)

3) Build a **Proof Of Concept (POC)** that shows your mastery of the delivery pipeline. And you will provide us will all the information needed to recreate the pipeline and run the base cases ourselves.
   Please choose any application of your choice to demonstrate your implementation of the pipeline. (8)

+ General appreciation (0,5)

1 Report per group

Send your PDF report to benjamin.fichera@intervenants.efrei.net

Before the 25th of July 2023 11pm CET

Add as much things as possible in your public code repository (or a .zip) to help me to find what I'm looking for when I try to higher your score.

You can benefit from a report review and commentary from the teacher :

- Unlimited report review with teacher until last lesson

4. OUR MAIN CONCERNS

4.1. PIPELINE/CI/CD (1)

What does the word *pipeline* refer to? How does it fit in our DevOps strategy? Do we need a pipeline in order to be DevOps compliant? What about Continuous Integration? What are your recommendations? Should we push til Continuous Deployment? What are the differences?

4.2. JENKINS (OR GITHUB-ACTION / GITLAB-CI) (1)

We are seriously considering adopting Jenkins. What it is? Where does it fit in the pipeline (diagram please)? Any best practices you'd recommend? Is Jenkins the only tool in its category? Give us an overview of the market in 2020.

4.3. VIRTUALIZATION / CONTAINERIZATION (DOCKER) (1)

Virtualization vs Containerization : we want to understand the differences before making any decision. What are the benefits and various scenarios of using Docker (Podman/ContainerD) instead of our good old VMs?

4.4. COMMUNICATION: SLACK (1)

We might choose Slack to improve the general communication between the teams. How could we do that? Please include a Communication tool demo in your POC.

4.5. TESTING & KPIS (1)

What do you recommend in order to industrialize, automate and increase the quality of our tests? Should and can we automate all the tests? What are the best practices in the field?

Please describe in great details your solution. It is extremely important to us.

DevOps strongly advocates the use of indicators to know if we are doing things right and how well we're doing things. Any clues on the indicators to use?

4.6. ORGANIZATION / CHANGE MANAGEMENT (3)

1) How will DevOps affect:
   - Our current team structure?
   - Our mentality?
   - Our IT delivery process?
   - Our technical architecture?
   - Our tool chain?

2) What are the risks in terms of our well-known human resistance to change?

3) Propose a comprehensive change management plan, recommend an approach that will help AAP quickly and efficiently adopt DevOps

5. FOCUS ON THE CI/CD PIPELINE

1. **Master/Slave** : Configure Jenkins (or equivalent tool… Gitlab-CI) in such a way that when the master accepts a build request, the execution is started on one of the slave (agent) machines (if the slave is dynamically created and deleted after the execution, you get bonus). The pipeline needs to include : build, unit test, deployment

2. Once the Jenkins build is complete, whether it succeeded or failed, a notification is sent to the developers (Slack / Discord / …).

3. The application is deployed to a final server that will contain only the application and nothing else.

4. All your code needs to be versioned and working

Reminder:

- A pipeline without **testing stages** is a nonsense.
- You can use **another tool than Jenkins** but it needs to meet the above requirements
- At least 3 hosts (VM or Container) are required: Pipeline orchestrator, Pipeline slave, Application server (that host the deployed application)

Going further (only if you want to earn bonus points)

Cette section n'est pas obligatoire

(Bonus) Develop an UAC (User Acceptance Test) or any "after deployment test" that give us confidence about the success. For example:

a. (easy) A python/shell script that check if the home page answer code 200
b. (great) A python/selenium/robotFramwork test that check if the expected html visual web elements are present https://www.lambdatest.com/blog/robot-framework-tutorial/