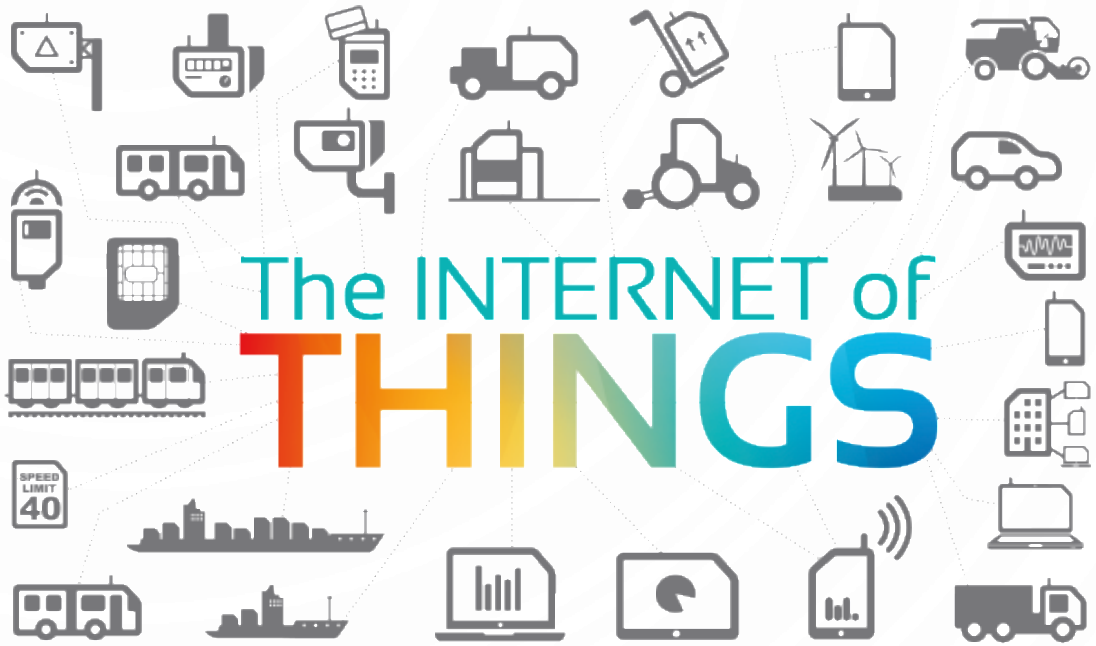
PROJECT MANAGEMENT

BARBAUX--ANQUETIL Sébastien
DEVINEZ Mathieu
KOSTULSKI Johan
MISHRA Shrey
SIRET Mélanie

eXia Nancy fourth promotion: **Atlantis Project**

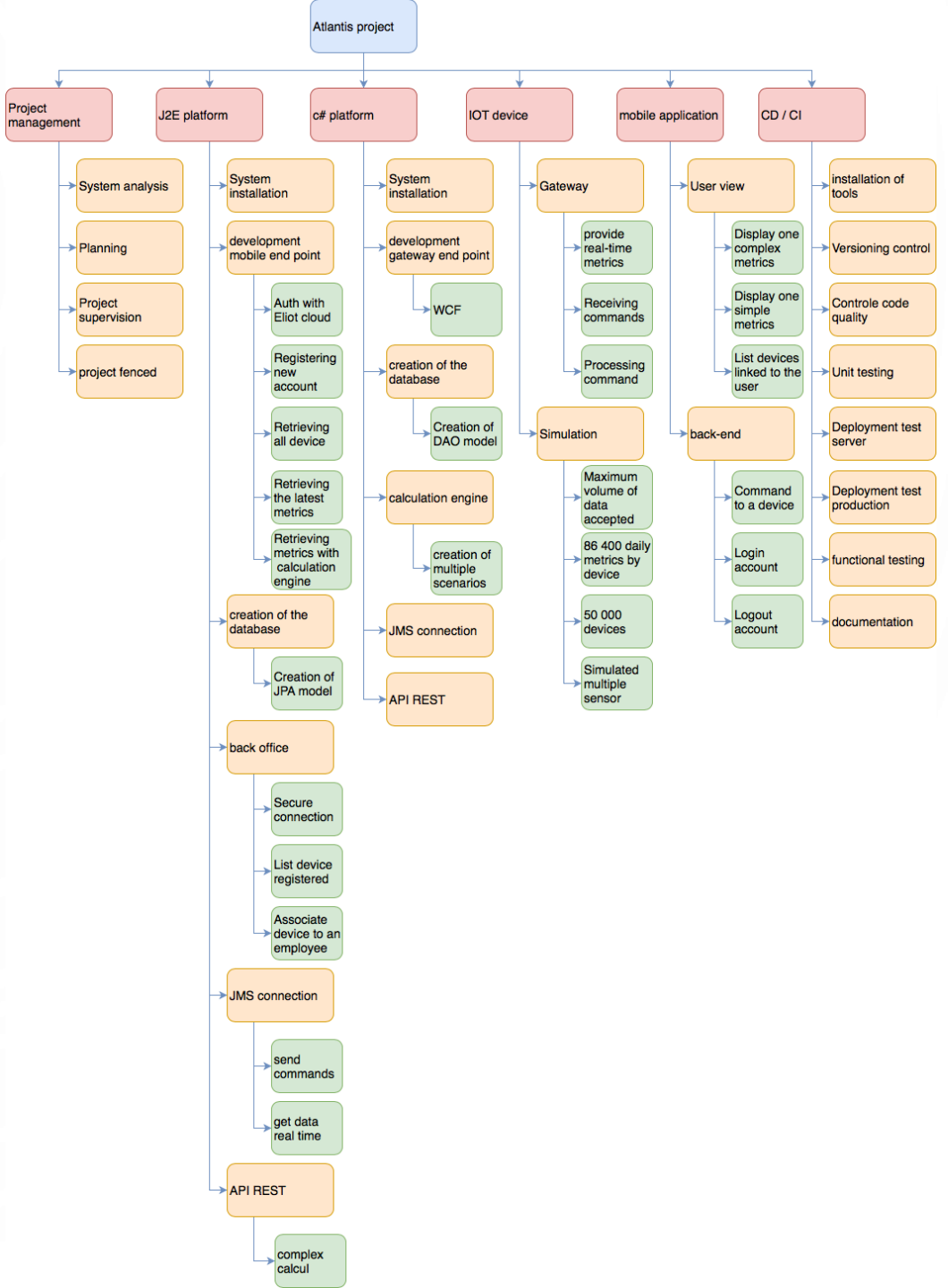
Promotion: **2017 - 2018**

SUMMARY



WORK BREAKDOWN STRUCTURE

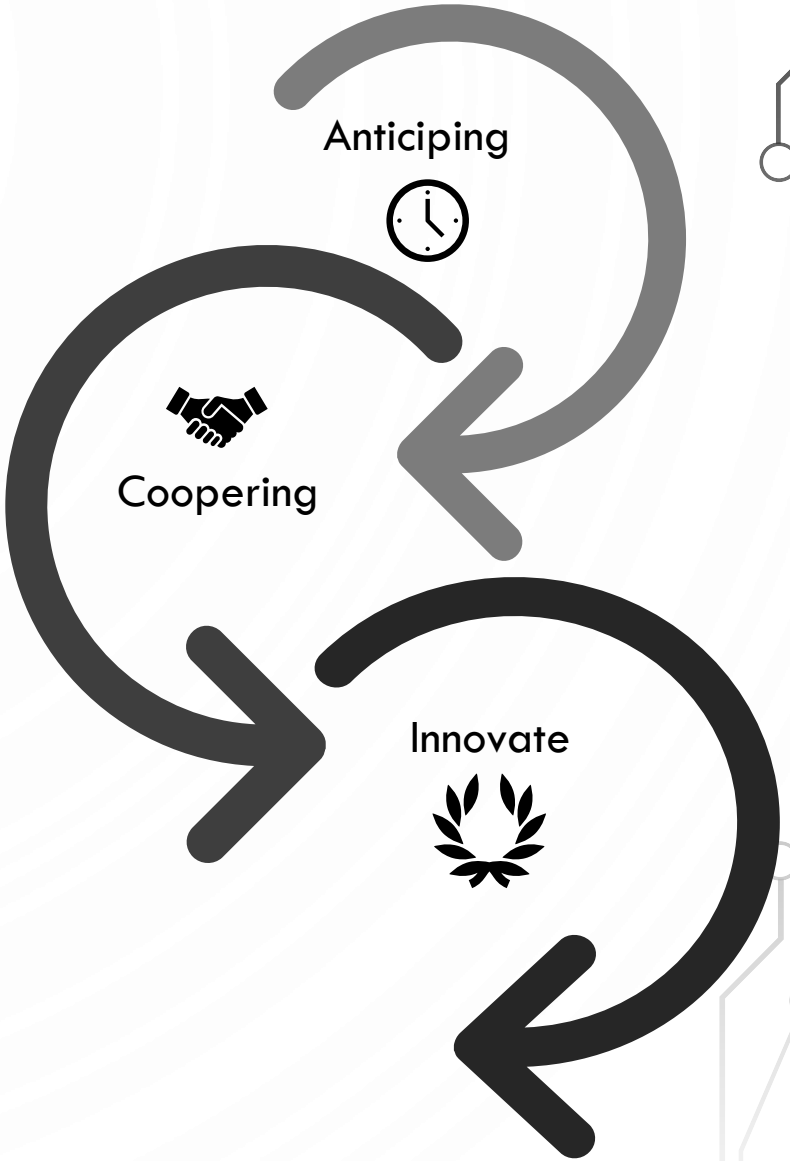
- WBS should contain all tasks for the project that is why we decided to represent all tasks by platform and kind of task.
- To do this WBS, we decided to start by splitting all major task (**red**) that we need, after we write all major deliveries (**orange**) that the client need, then we expand all sub-task (**green**) to be sure all task is developed for the client.
- As we can see we split the WBS in 6 part.
  - Project management
  - JE2 platform
  - C# platform
  - IOT Device
  - Mobile application
  - CI



Using a WBS is called a “Directive line for management of a project” is the **ISO 21500**

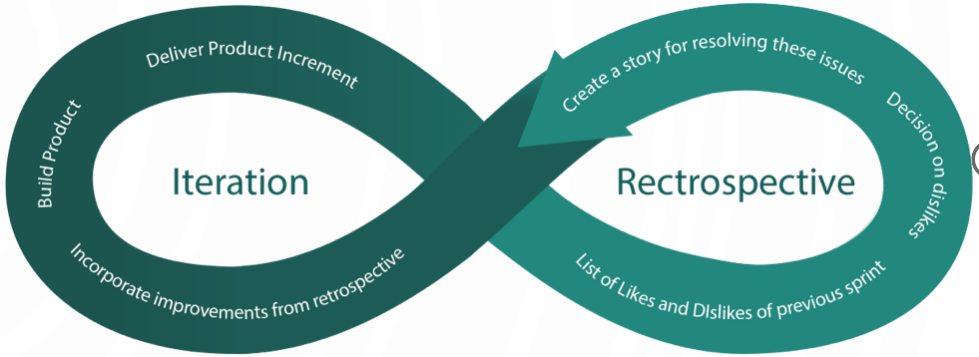
AGILE EXPLANATION

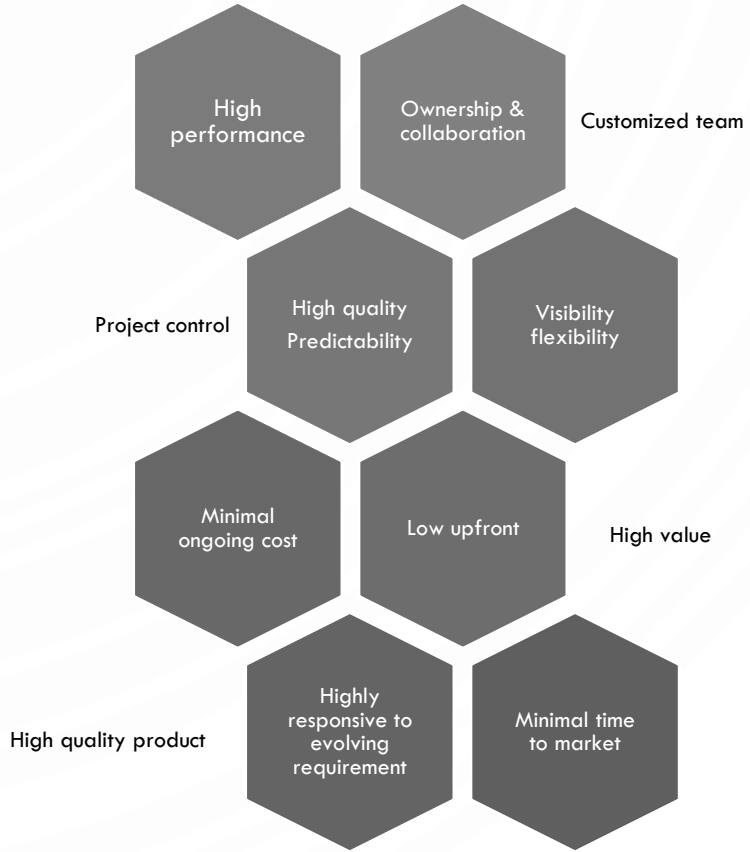
- Agile methodology based on an iterative and collaborative approach. Agile is used to accelerate the development.
- The main purpose of this methodology is to make a minimal version of the solution and add functionalities by iterative processes. To do this, the development should follow four rules:
  - Make unit test after each iteration
  - Develop
  - Make a integration
  - Optimisation and review of the code every time
- Agile is based on four principles:
  - **Collaboration**: Communication and team cohesion go before tools and processes.
  - **Team**: The privilege of the team / client relationship is put forward rather than the contractual negotiation.
  - **Application**: Prefer a well built application to a detailed documentation.
  - **Acceptance**: The choice of acceptance of change and flexibility at the expense of a rigid plan.



JUSTIFICATION OF USING AGILE

- Our project need a strong consistency
- The client is placed between all process during the development
- The development had to be the gain of the collaboration between all branches of development to ensure a perfect quality of the project
- Agile method allow us to adapt a functionality during the development
- Possibility to make management by iterative cycle





Advantages:

Flexibility of the product

Pragmatic (Need a concrete use to be develop)

Development oriented client, ensure the quality of the product

Comfortable for the client (see all advances)

SCRUM EXPLANATION

- After analysing the whole project and need of the client we made the choice to use the scrum methodology. The Scrum framework is for more than just software development. It was born from the need to create software, but is much more about enabling rapid intense human collaboration. Because Scrum enables such rapid feedback, it has proven itself in many domains other than software and has become generally applicable to cognitively intense knowledge work for teams.

- **What is Scrum ?**

  Scrum is an agile methodology that has for purpose to improve the productivity of the team.

  He's based on three pillars:

  - **Transparency**: Scrum ensures that all team had a common language to be sure all understand the project.

  - **Inspection**: Regularly the team need to make a point on the different artefact to ensure the project doesn't deviate from the main objective

  - **adaptation**: If a deviation is noticed during the inspection, the process must be adapted. Scrum provides events, during which this adaptation is possible. This is the sprint planning meeting, the daily scrum, the sprint review and the sprint retrospective.

In Scrum we found 3 different parts:

1. Scrum master: This member of the project ensures that all scrum principles are followed, moreover he facilitates the communication in the team and tries to improve the productivity of the team every time.

2. Product owner: This member of the project is like an expert for the project, he represents the client and validates all functionalities developed by the team.

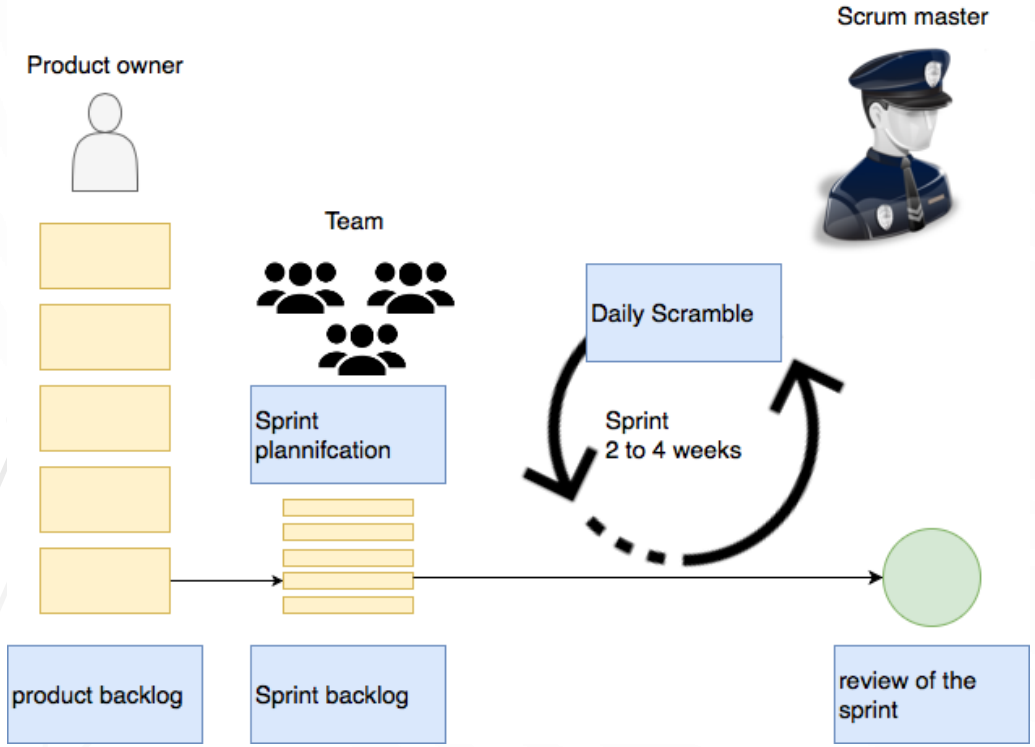
3. The team: The members of the team don't have a specific role in the project. All member can help another member and do his task, moreover the team must not exceed 10 people.

SCRUM EXPLANATION

- Scrum is the most used methodology in the world. The scrum method is based on "sprints". A sprint is Short time interval (1 month maximum, often called iteration), during which the development Team will design, build and test new features.

For our project due to his short deadline we have only two sprint.

- The first one was the response to the tender request
- The seconds was the development of the solution



This schema resume perfectly how scrum works we can find :

- **Product backlog**: Scheduled (prioritized) list of requirements of the project.
- **Sprint backlog**: Product Owner and Dev Team working meeting to select the elements of the Product Backlog to be performed during the starting Sprint.
- **Daily Scramble**: A daily meeting of less than 15 minutes allowing the Dev Team to synchronize, identify potential obstacles and measure progress on the current Sprint.
- **Sprint Retrospective**: Work meeting of the Scrum Team with the aim of learning the lessons of the expired Sprint in order to benefit the following ones. See Concrete Principle of Collective Continuous Improvement.
- **Scrum Master:** Guarantor of respect for Scrum methodological framework. He also has a role of coach regarding to development Team and the Product Owner in order to help them play their full role.

EVIDENCE OF IT'S USE

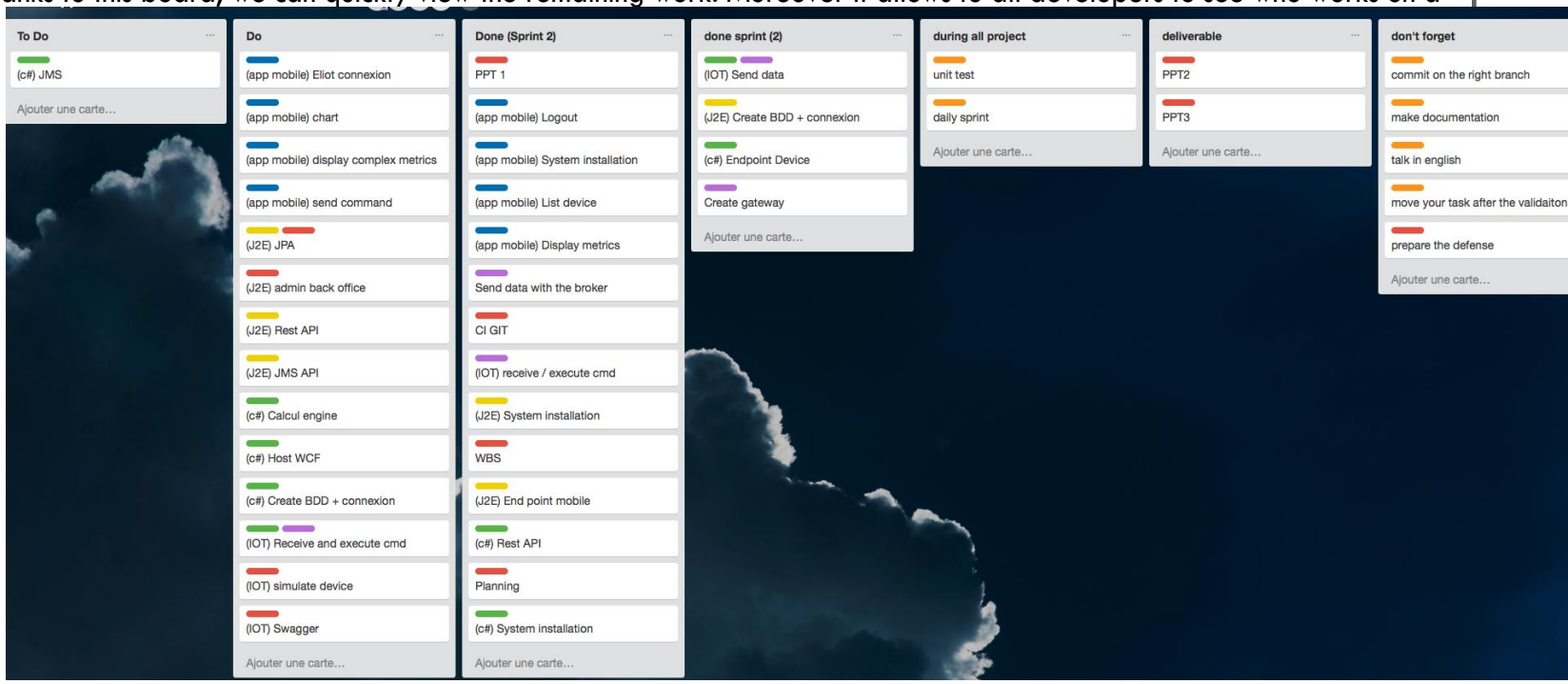
To use Scrum, we decide to put in place a board called "kandan board".

It's separated in three column "To do", "Doing", "Done". Thanks to this board, we can quickly view the remaining work. Moreover it allows to all developers to see who works on a special task.

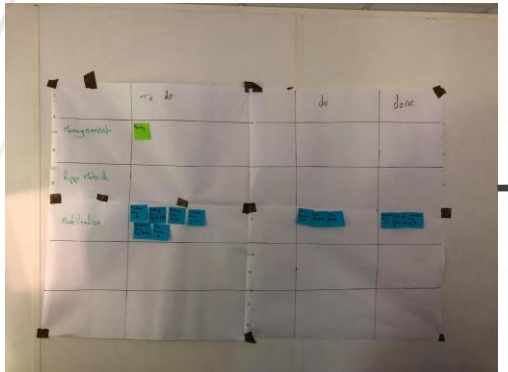
Moreover, we use 3 colours to make the differences between all task:

- **Rose**: Deliverable
- **Green**: Development tasks
- **Blue**: task to do during all project

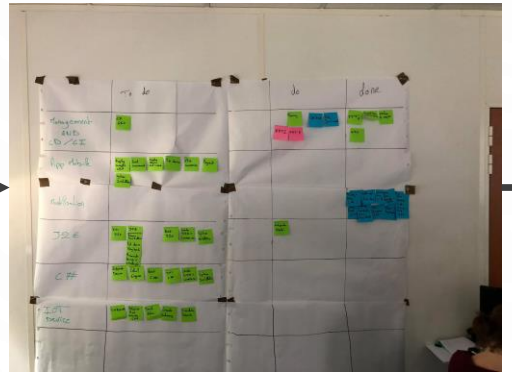
We also use the kandan board in 'real', but we also use a board online for the work the night and the week-end. Both was used by the whole team.



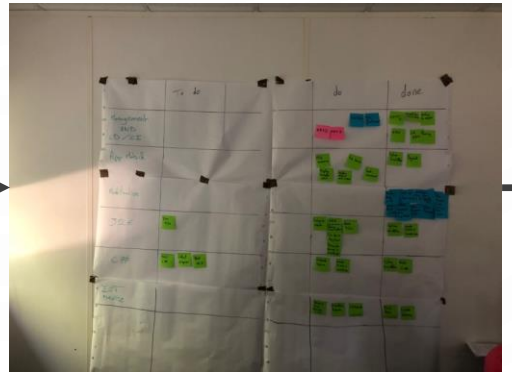
Sprint 1



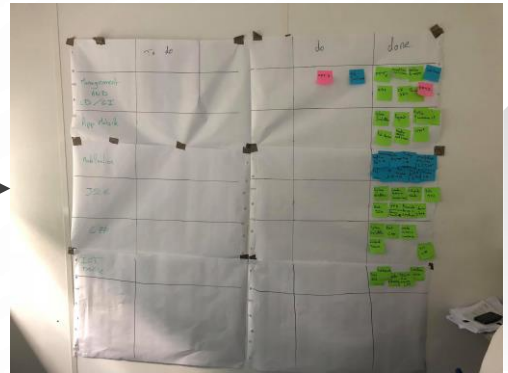
End sprint 1 / start sprint2



Middle sprint 2



End sprint2

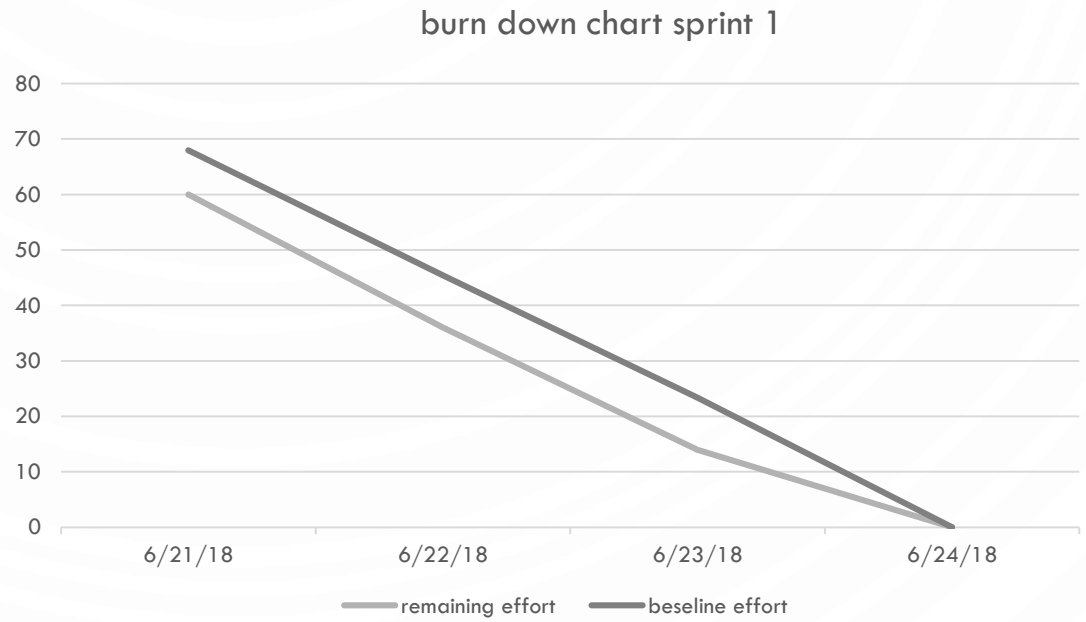


EVIDENCE OF ITS USE

- During the whole project, every morning the group made a little meeting (5-7minutes) to defined all tasks to do during the sprint. Nobody has a special ”job”, all of us are polyvalent. After all end of a day we make a review of the task done by the member of the group to be sure the project ensures to the client needs.
- Moreover we represent our work with a burndown chart, that allows us to follow the progress of the project with all point distributed during the planning poker. A burn down graph is generally used to track Sprint progress (short term horizon). It makes it possible to follow the evolution of the remaining work as a function of time. The goal is to reach the zero level as soon as possible, hence the term "Down".

		21/06/2018	22/06/2018	23/04/2018	24/04/2018
analyse rao	4	4	0	0	0
logical architecture	8	4	4	0	0
(c#) component diagram	5	0	5	0	0
(j2e) component diagram	5	0	5	0	0
(app mobile) component diagram	5	0	5	0	0
(iot) component diagram	5	0	5	0	0
(c#) package diagram	4	0	0	4	0
(j2e) package diagram	4	0	0	4	0
(app mobile) package diagram	4	0	0	4	0
(c#) class diagram	6	0	0	2	4
(j2e) class diagram	6	0	0	2	4
(app mobile) class diagram	6	0	0	2	4
physical architecture	4	0	0	4	0
power point validation	2	0	0	0	2

	21/06/2018	22/06/2018	23/06/2018	24/06/2018
remaining effort	60	36	14	0
beseline effort	68	45,4	23,4	0



COLLABORATIVE TOOLS

Name of the tools: Discord

Explanation: Discord allow us to make multiple channel to communicate.

Name of the tools: Trello

Explanation: Our Trello is like our scrum board but online.

Name of the tools: Google Drive

Explanation: Google drive was used when we make documentation for API or share document, power point etc.

Name of the tools: Appear.in

Explanation: Appear is a tools online that can allow us to record our screen (mobile and computer) and share it with the other member of the team.

Name of the tools: Merlin

Explanation: Merlin was used to create the provisional and the effective planning

Name of the tools: GitLab

Explanation: GitLab was used by all member of team, it's thanks to git we can all gather all code, create a test and see if the code is correct etc.

Name of the tools: Git Cracken

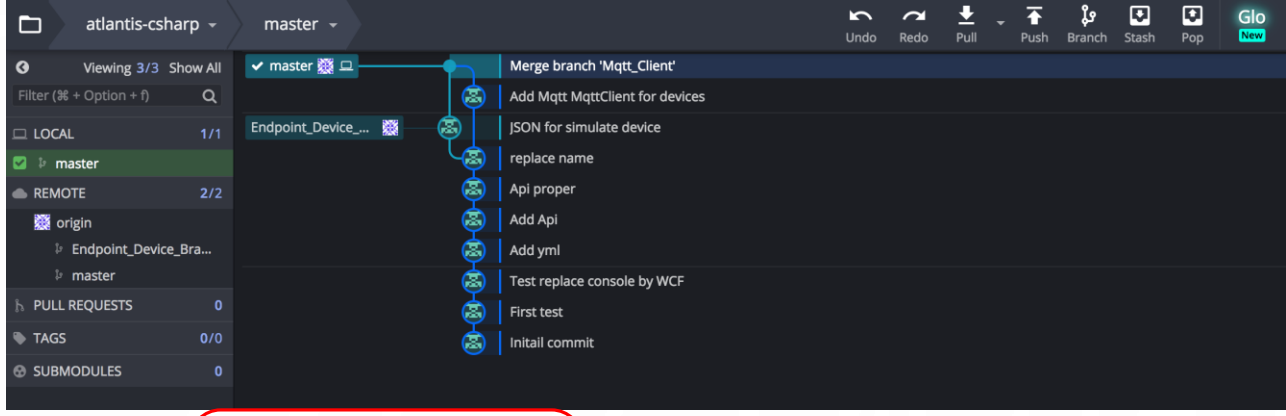
Explanation: Git kraken was mainly used by the manager of the group to push all code in the master branch.

Name of the tools: Amazon web services

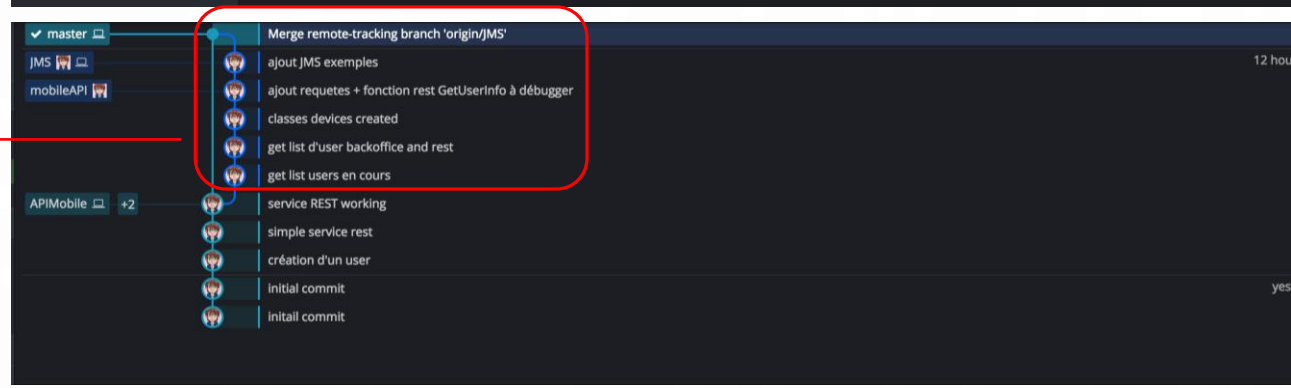
Explanation: We used AWS to manage our Jenkins server.

COLLABORATIVE TOOLS

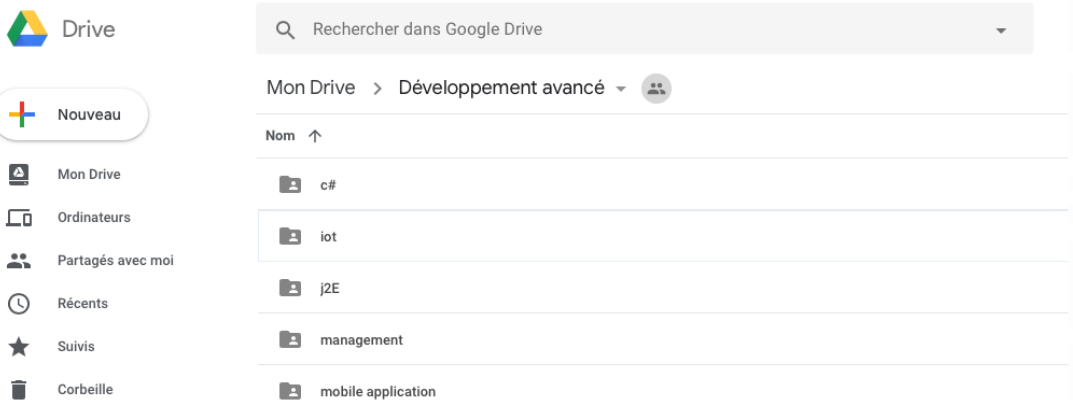
The picture on the right represent our gitckracken, that allow us to manage the merge and the push of the project



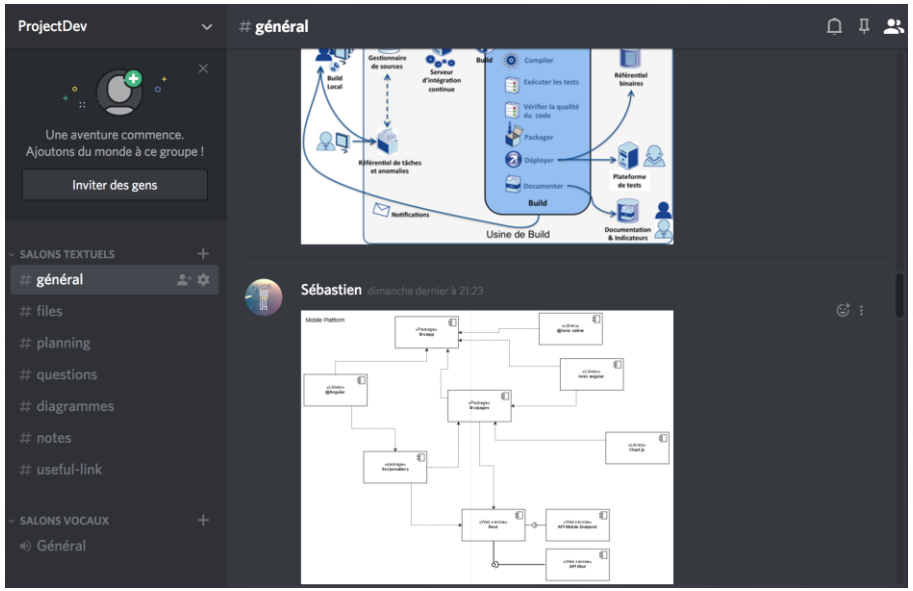
When a developer wants to push on the master before the developer creates a new branch with the name of the update, then the project manager will test it and push on the master.



On the bottom you can see a screen of our google drive. This Google drive contains all documentation and information on our project.



Google Drive



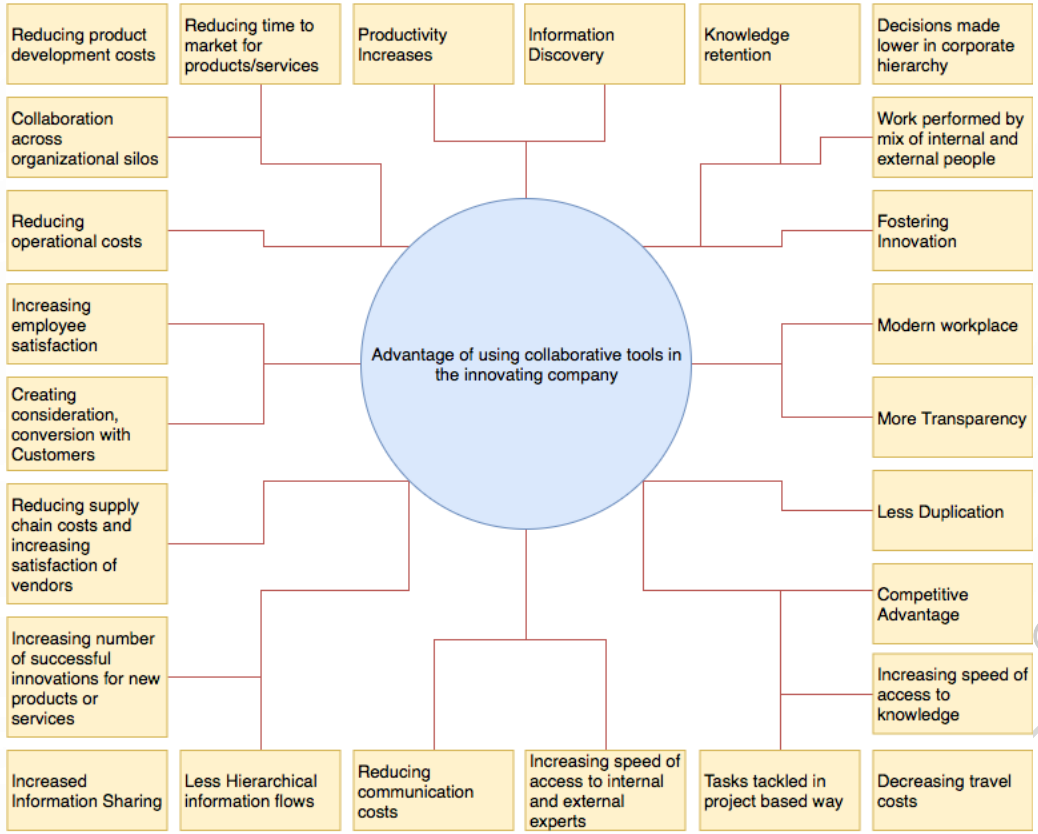
Discord

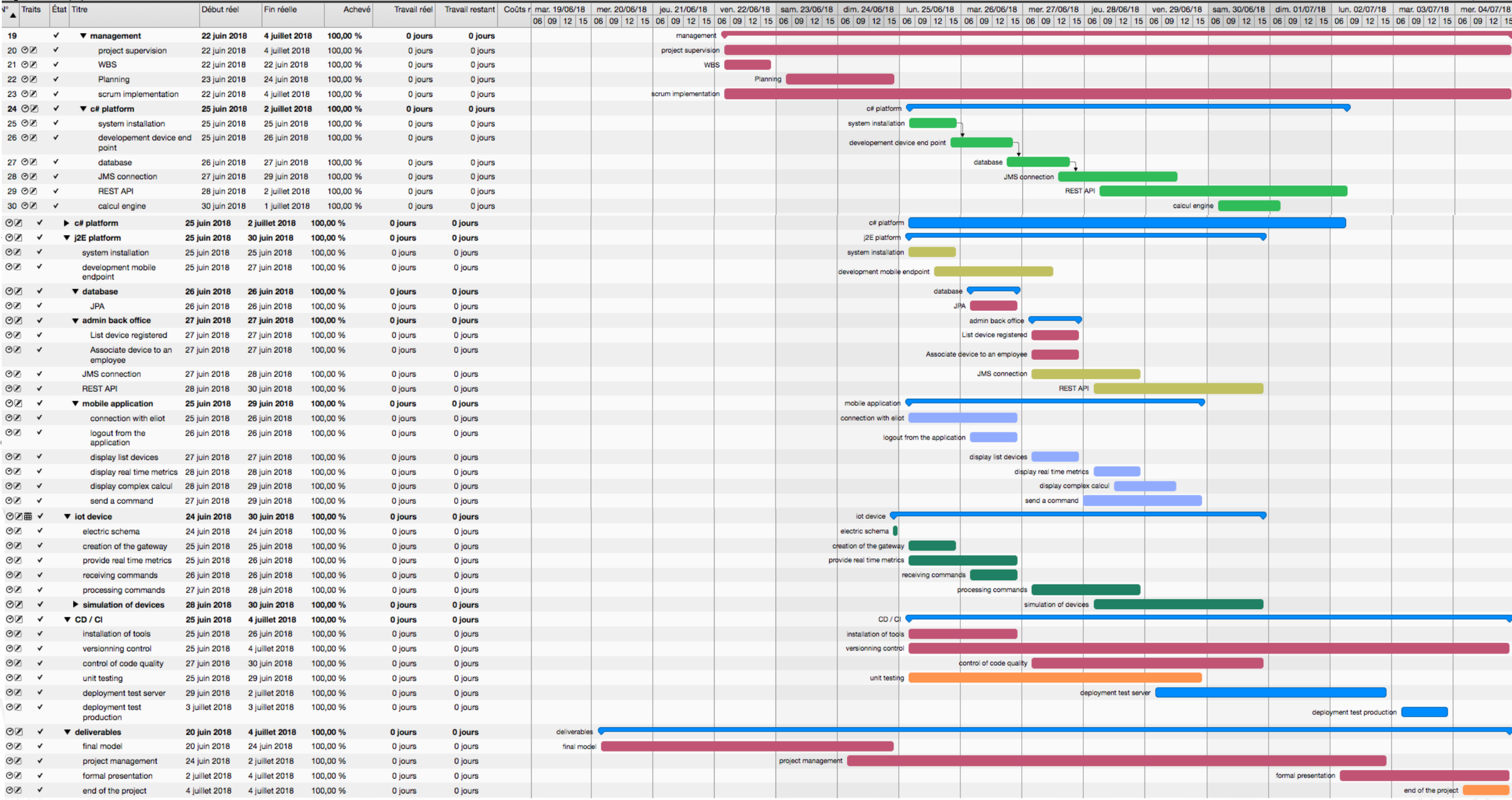
We also created a discord which different channels, that allows us to directly discuss on a specific discussion with the right person

GAIN TO USE COLLABORATIVE TOOLS

- **Saving time**: it does not create existing documents.
- **Foster social climate**: Team members are always interacting. This helps to regulate social confidences and improve trust between team members.
- **Increased productivity**: Team members are more efficient with collaborative work tools.
- **Improve team autonomy**: team members are free to carry out their project.
- **Knowledge Sharing**: Through collaborative work, team members share information consistently.
- **Saves time** by eliminating meetings

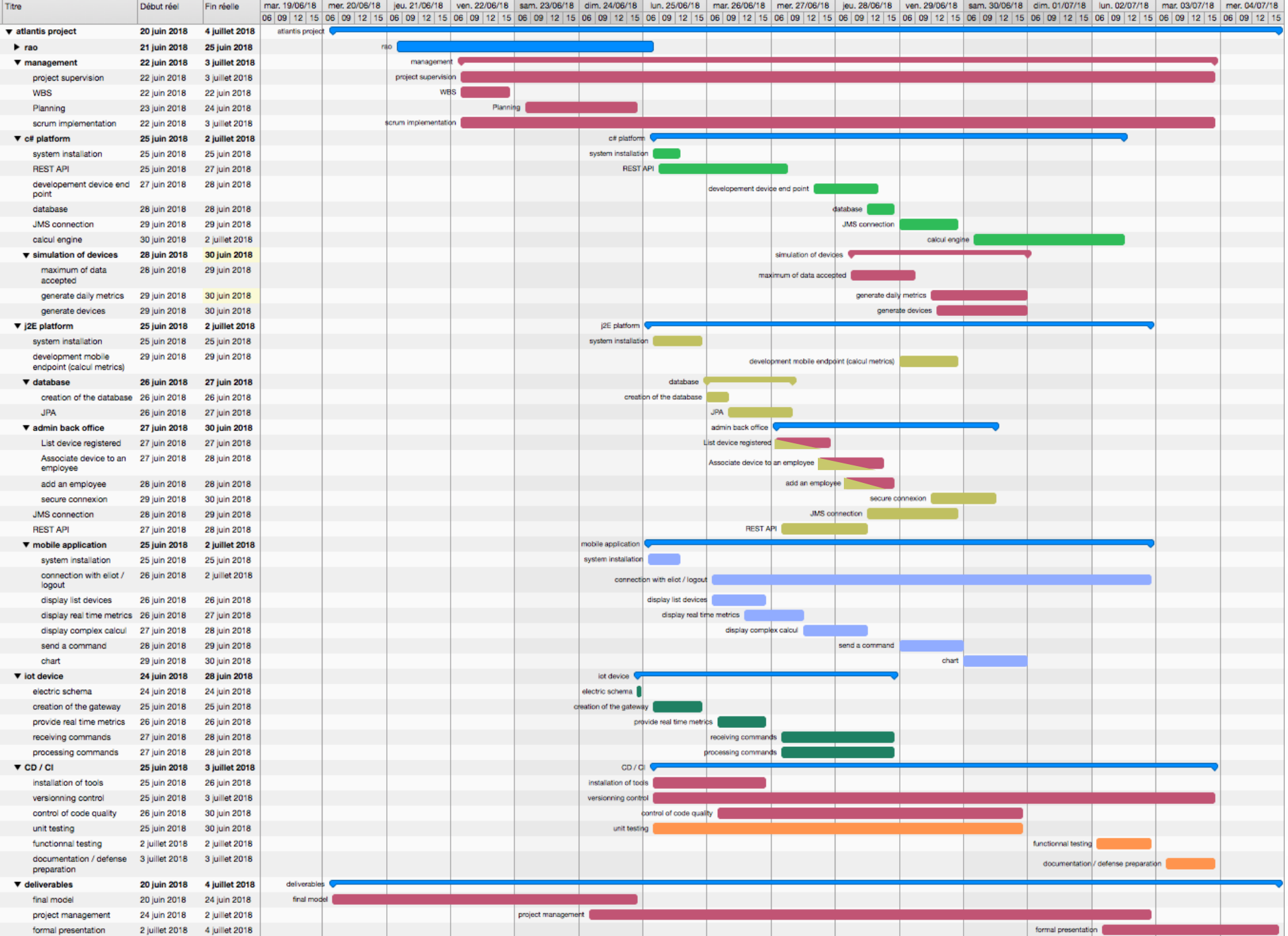
You can find more advantages to use collaborative tools in a international company.





N°	Traits	État	Titre	Début réel	Fin réelle	Achevé	Travail réel	Travail restant	Coûts r
19		✓	▼ management	22 juin 2018	4 juillet 2018	100,00 %	0 jours	0 jours	
20		✓	project supervision	22 juin 2018	4 juillet 2018	100,00 %	0 jours	0 jours	
21		✓	WBS	22 juin 2018	22 juin 2018	100,00 %	0 jours	0 jours	
22		✓	Planning	23 juin 2018	24 juin 2018	100,00 %	0 jours	0 jours	
23		✓	scrum implementation	22 juin 2018	4 juillet 2018	100,00 %	0 jours	0 jours	
24		✓	▼ c# platform	25 juin 2018	2 juillet 2018	100,00 %	0 jours	0 jours	
25		✓	system installation	25 juin 2018	25 juin 2018	100,00 %	0 jours	0 jours	
26		✓	developement device end point	25 juin 2018	26 juin 2018	100,00 %	0 jours	0 jours	
27		✓	database	26 juin 2018	27 juin 2018	100,00 %	0 jours	0 jours	
28		✓	JMS connection	27 juin 2018	29 juin 2018	100,00 %	0 jours	0 jours	
29		✓	REST API	28 juin 2018	2 juillet 2018	100,00 %	0 jours	0 jours	
30		✓	calcul engine	30 juin 2018	1 juillet 2018	100,00 %	0 jours	0 jours	
		✓	▶ c# platform	25 juin 2018	2 juillet 2018	100,00 %	0 jours	0 jours	
		✓	▼ j2E platform	25 juin 2018	30 juin 2018	100,00 %	0 jours	0 jours	
		✓	system installation	25 juin 2018	25 juin 2018	100,00 %	0 jours	0 jours	
		✓	development mobile endpoint	25 juin 2018	27 juin 2018	100,00 %	0 jours	0 jours	
		✓	▼ database	26 juin 2018	26 juin 2018	100,00 %	0 jours	0 jours	
		✓	JPA	26 juin 2018	26 juin 2018	100,00 %	0 jours	0 jours	
		✓	▼ admin back office	27 juin 2018	27 juin 2018	100,00 %	0 jours	0 jours	
		✓	List device registered	27 juin 2018	27 juin 2018	100,00 %	0 jours	0 jours	
		✓	Associate device to an employee	27 juin 2018	27 juin 2018	100,00 %	0 jours	0 jours	
		✓	JMS connection	27 juin 2018	28 juin 2018	100,00 %	0 jours	0 jours	
		✓	REST API	28 juin 2018	30 juin 2018	100,00 %	0 jours	0 jours	
		✓	▼ mobile application	25 juin 2018	29 juin 2018	100,00 %	0 jours	0 jours	
		✓	connection with eliot	25 juin 2018	26 juin 2018	100,00 %	0 jours	0 jours	
		✓	logout from the application	26 juin 2018	26 juin 2018	100,00 %	0 jours	0 jours	
		✓	display list devices	27 juin 2018	27 juin 2018	100,00 %	0 jours	0 jours	
		✓	display real time metrics	28 juin 2018	28 juin 2018	100,00 %	0 jours	0 jours	
		✓	display complex calcul	28 juin 2018	29 juin 2018	100,00 %	0 jours	0 jours	
		✓	send a command	27 juin 2018	29 juin 2018	100,00 %	0 jours	0 jours	
		✓	▼ iot device	24 juin 2018	30 juin 2018	100,00 %	0 jours	0 jours	
		✓	electric schema	24 juin 2018	24 juin 2018	100,00 %	0 jours	0 jours	
		✓	creation of the gateway	25 juin 2018	25 juin 2018	100,00 %	0 jours	0 jours	
		✓	provide real time metrics	25 juin 2018	26 juin 2018	100,00 %	0 jours	0 jours	
		✓	receiving commands	26 juin 2018	26 juin 2018	100,00 %	0 jours	0 jours	
		✓	processing commands	27 juin 2018	28 juin 2018	100,00 %	0 jours	0 jours	
		✓	▶ simulation of devices	28 juin 2018	30 juin 2018	100,00 %	0 jours	0 jours	
		✓	▼ CD / CI	25 juin 2018	4 juillet 2018	100,00 %	0 jours	0 jours	
		✓	installation of tools	25 juin 2018	26 juin 2018	100,00 %	0 jours	0 jours	
		✓	versionning control	25 juin 2018	4 juillet 2018	100,00 %	0 jours	0 jours	
		✓	control of code quality	27 juin 2018	30 juin 2018	100,00 %	0 jours	0 jours	
		✓	unit testing	25 juin 2018	29 juin 2018	100,00 %	0 jours	0 jours	
		✓	deployment test server	29 juin 2018	2 juillet 2018	100,00 %	0 jours	0 jours	
		✓	deployment test production	3 juillet 2018	3 juillet 2018	100,00 %	0 jours	0 jours	
		✓	▼ deliverables	20 juin 2018	4 juillet 2018	100,00 %	0 jours	0 jours	
		✓	final model	20 juin 2018	24 juin 2018	100,00 %	0 jours	0 jours	
		✓	project management	24 juin 2018	2 juillet 2018	100,00 %	0 jours	0 jours	
		✓	formal presentation	2 juillet 2018	4 juillet 2018	100,00 %	0 jours	0 jours	
		✓	end of the project	4 juillet 2018	4 juillet 2018	100,00 %	0 jours	0 jours	

Provisional planning : Melanie is **yellow,** Sébastien is **Bleu,** Mathieu is **red,** Johan is **light green,** Shrey is **dark green,** the group is **orange.**



Effective planning :

Melanie is **yellow**

Sébastien is **Bleu**

Mathieu is **red**

Johan is **light green**

Shrey is **dark green**

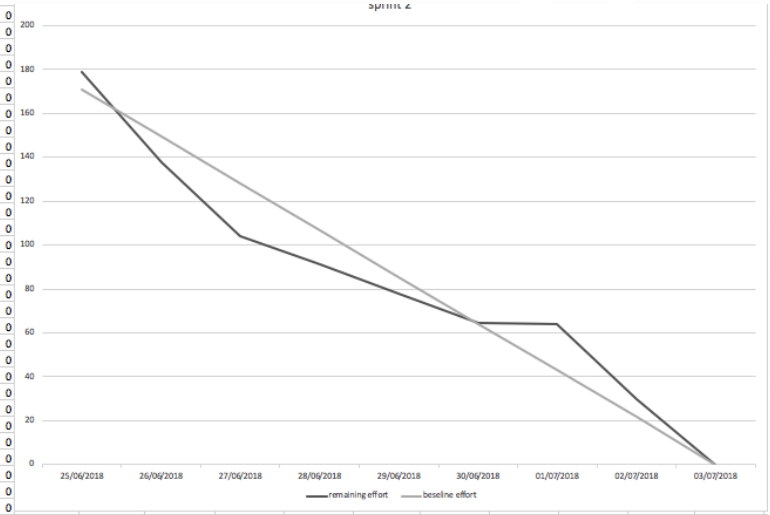
Group is **orange**

PLANNING EXPLANATION

- Feedback sprint 2
  - We try to put a Continuous deployment in place, however this task took more time than expected. Try to put Jenkins with a GitLab not in localhost was very difficult, that is why we chose to put in place Jenkins on an **Aws server** and made the connexion with a GitLab online.
  - Jpa was more longer than expected
  - The API to make connexion with Eliot was different that the documentation we had, that is why the mobile developer took more time to make the connexion.

Burndown chart sprint 2

(Project management) system analysis	3	3	0	0	0	0	0	0	0	0
(Project management) Planning	5	4	1	0	0	0	0	0	0	0
(J2E) System installation	3	3	0	0	0	0	0	0	0	0
(J2E) Development mobile end point	4	0	2	1	1	0	0	0	0	0
(J2E) Registering new account	4	0	0	0	0	2	2	0	0	0
(J2E)Retrieving all device	4	0	0	0	0	2	2	0	0	0
(J2E) Retrieving the latest metrics	4	0	0	0	0	2	1	0	1	0
(J2E) Retrieving a metrics with calculation engine	4	0	0	0	0	0	2	0	2	0
(J2E) Creation of the database	2	0	0	2	0	0	0	0	0	0
(J2E) JPA	6	0	0	3	3	0	0	0	0	0
(J2E) back office - secure connection	2	0	0	0	0	2	0	0	0	0
(J2E) back office - list device registered	3	0	0	1	2	0	0	0	0	0
(J2E) associate device to an employee	2	0	0	0	0	2	0	0	0	0
(J2E) JMS - send command	8	0	0	0	0	4	2	0	2	0
(J2E) JMS - get data real time	8	0	0	0	4	2	2	0	0	0
(J2E) API REST - complex calcul communication	8	0	3	5	0	0	0	0	0	0
(C#) System installation	2	2	0	0	0	0	0	0	0	0
(C#) development gateway end point (WCF)	9	0	4	5	0	0	0	0	0	0
(C#) creation of the database	1	0	0	0	1	0	0	0	0	0
(C#) creation of dao model	4	0	0	3	1	0	0	0	0	0
(C#) calculation engine	4	0	0	0	0	0	4	0	0	0
(C#) creation of multiple scenario for calculation engine	3	0	0	0	0	0	2	0	1	0
(C#) JMS connection	8	2	2	0	0	4	0	0	0	0
(C#) API REST - complex calcul communication	8	0	0	0	7	0	0	0	1	0
(IOT Device) Provide real time metrics	8	4	4	0	0	0	0	0	0	0
(IOT Device) Receiving commands	6	0	0	2	4	0	0	0	0	0
(IOT Device) Processing command	6	0	0	2	4	0	0	0	0	0
(IOT Device) Simulation with swagger	7	0	2	0	5	0	0	0	0	0
(IOT Device) 86 400 daily metrics by device	7	0	2	4	0	0	0	0	1	0
(IOT Device) simulate 50 000 devices	7	0	2	3	0	1	0	0	1	0
(IOT Device) Simulated multiple sensor	7	0	2	3	0	1	0	0	1	0
(mobile application) systeme installation	3	3	0	0	0	0	0	0	0	0
(Mobile application) Display one complex metrics	3	0	1	1	1	0	0	0	0	0
(Mobile application) Display one simple metrics in real time	5	0	0	4	0	0	0	0	1	0
(Mobile application) List all device	3	0	1	1	1	0	0	0	0	0
(Mobile application) command to a device	3	0	0	0	0	3	0	0	0	0
(Mobile application) login account	8	0	0	2	2	2	0	0	2	0
(Mobile application) logout account	2	0	0	2	0	0	0	0	0	0
(CI) installation of tools	5	4	1	0	0	0	0	0	0	0
PPT management	8	0	0	1	1	2	4	0	0	0
(CI) versionning control	8	2	6		0	0	0	0	0	0
end of the project	1	0	0	0	0	0	0	0	0	1



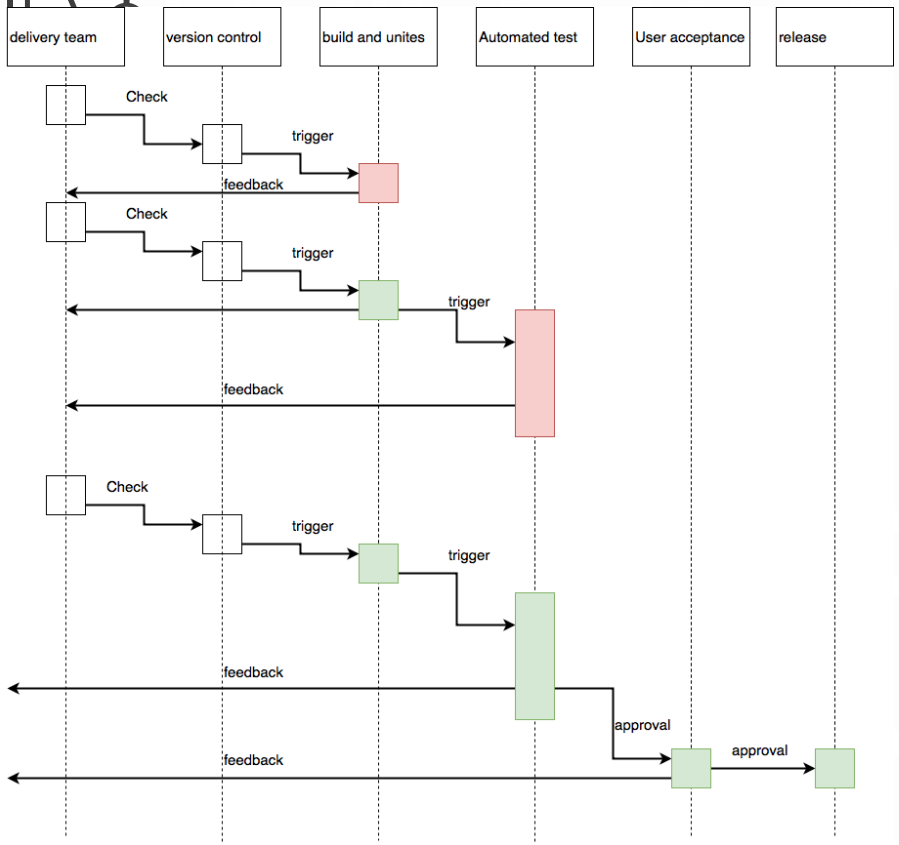
206									
	25/06/2018	26/06/2018	27/06/2018	28/06/2018	29/06/2018	30/06/2018	01/07/2018	02/07/2018	03/07/2018
remaining eff	179	137,7	104,3	91,1	77,8	64,5	64,2	29,9	0
beseline effo	170,7	149,3	128,1	106,8	85,5	64,2	42,9	21,6	0

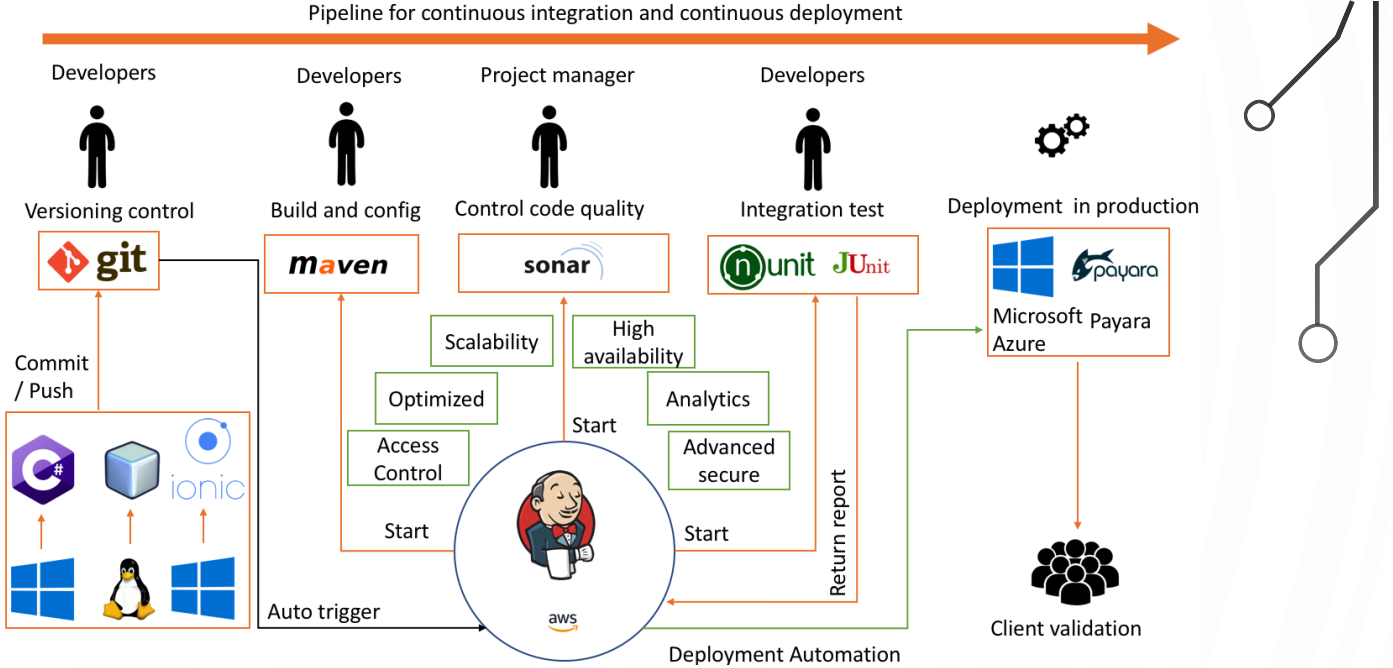
As we can see this burndown chart is very relevant, we saw the group very efficient during the middle of the week, and when the week-end come we can see a little relaxation.

Furthermore, nobody works on the Sunday, due to the good progression of the work, then we can see the curve ascent on the Sunday and go down the Monday.

Moreover, we took a little delay at the start but quickly caught by the whole team

CI CHAIN





- Here you can find our CI chain. The delivery team, also the development team push our project on our versioning control (git) if no error append after their push then the project are build and tested by our solution (**Jenkins / git LAB ci**) if no error append, the project is deployed and tested by the project manager whom represents the client. If the project is validate then the code is released.
- During the whole process, if the build or the test does not work, the product is automatically redirected to the delivery team.

We decided to use Jenkins as an automation server, Jenkins will really help us for the continuous integration part and the advantage is the number of plugins it has.

SonarQube will be our control code quality, it will provides the capability to show the health of our application (Bugs & vulnerabilities, coverage etc…)

Maven will be our build automation tool. The compilation and distribution process will be clearly defined by it in the build lifecycle

Speaking about plugins, NUnit and JUnit will be integrated to our Jenkins. NUnit will test our code in C# and JUnit our code in Java, both of them are really important because they can save us a lot of time and it will allow us to test instantly our code to be sure that everything works just fine.

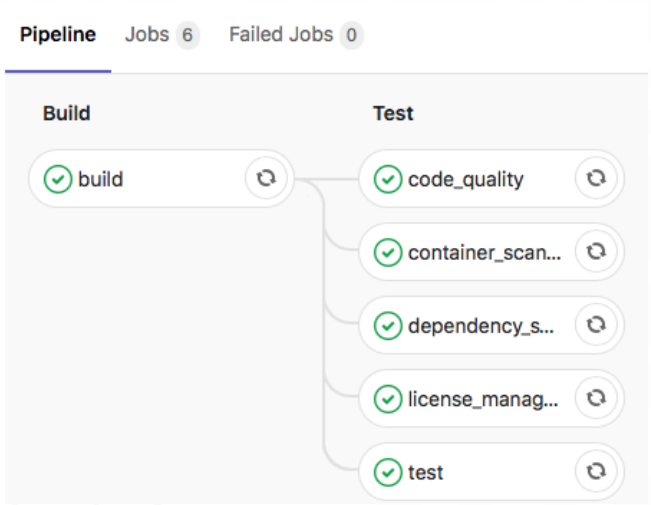
CI CHAIN

When a code is pushed we have created a file called ' .gitlab-ci.yml' this file contains a configuration to use multiple tools that can:

- Test the quality of the code
- Test the security with the 'Container Scanning' that will create a docker image and analyse the code
- Test all dependencies of the project
- Test the license Management to see all dependencies used by the provided source.
- Test automatically runs the appropriate tests unitest for our application using 'herokuish' by analysing our project to detect the language and framework.

Moreover, every 15 minutes the code is caught on GitLab with our server Jenkins on the AWS. And that will try to build the code.

Exemple of gitlab



Exemple of Jenkins Server