

A night photograph of a city skyline. In the foreground, a bridge or elevated walkway with a metal railing is visible, its structure partially illuminated. Behind it, several tall buildings are lit up from within, showing numerous windows. One building has a large digital sign that reads "ADDITIONAL FLOOR ON THE MARKET", "LUXURY RENTALS", and "FOR \$500,000". To the left, construction cranes are silhouetted against the dark sky. The overall atmosphere is industrial and urban.

Forge logicielle      Autour de la webapp

# Forge logicielle - TDD

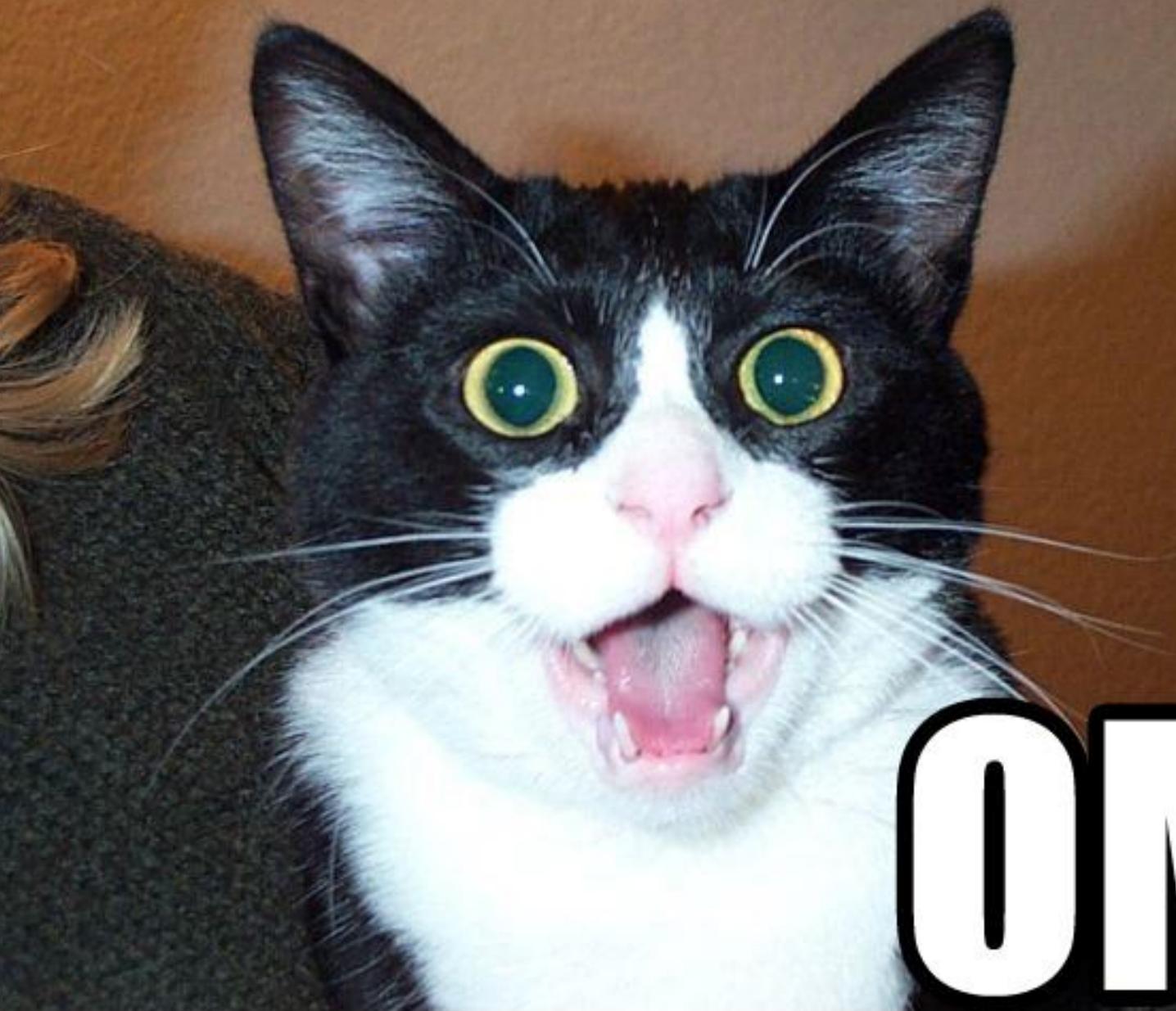
- Ma vie de chef de projet
- Travailler en équipe
- Forge
- Tests
- Déploiement



## Ma vie de chef de projet

“J'ai une super idée, on va se faire une machine à cambrer les bananes! On va défoncer la concurrence!”



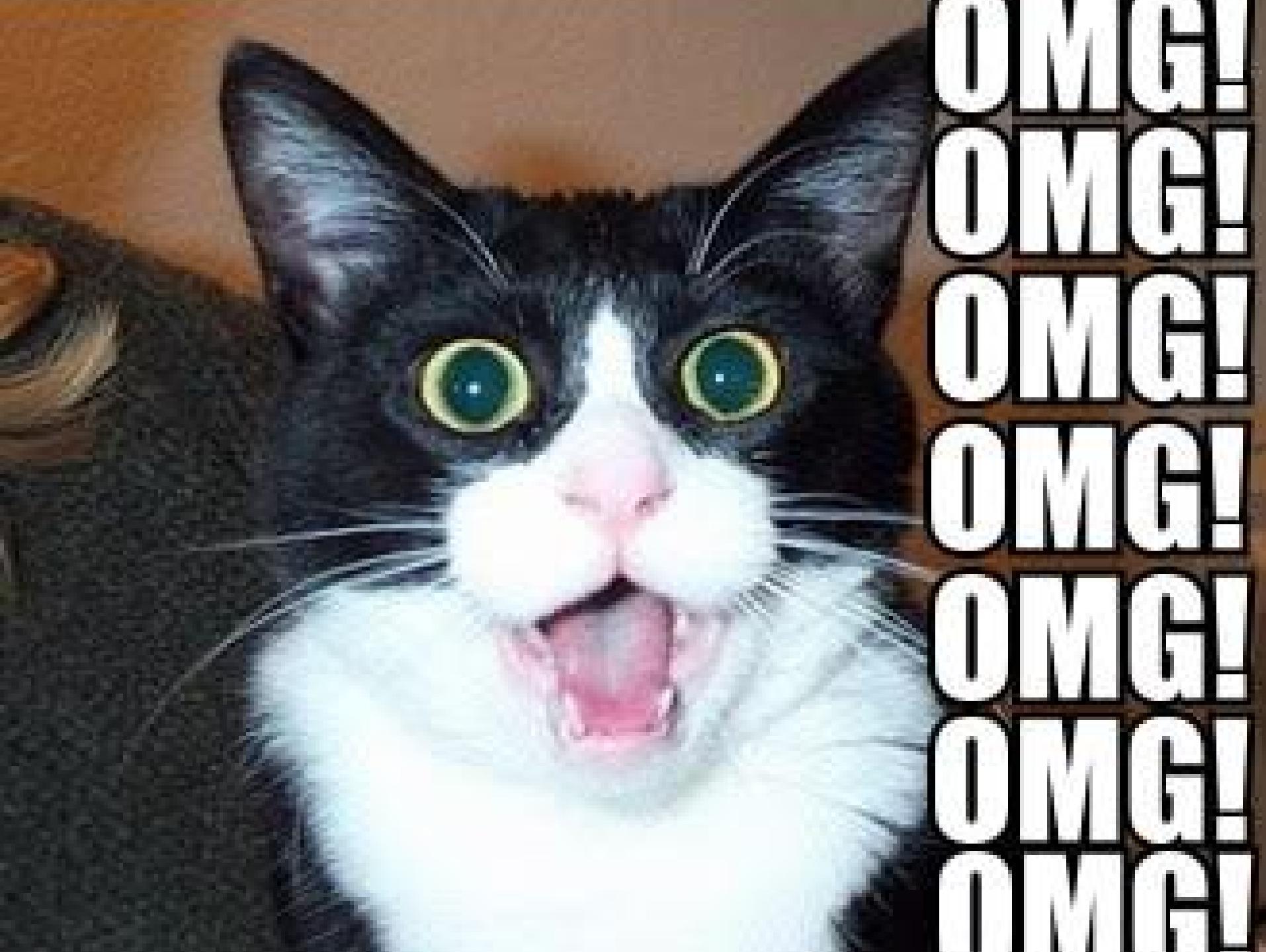


**OMG**

## Ma vie de chef de projet

"et on a la campagne d'octobre qui commence en septembre, il nous faut cette machine!"





OMG!  
OMG!  
OMG!  
OMG!  
OMG!  
OMG!  
OMG!  
OMG!  
OMG!

# Ma vie de chef de projet

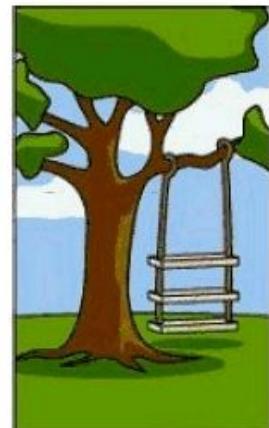
L'inventaire :

- un chiffrage
- s'organiser
  - des ~~développeurs~~ ressources
  - planifier
  - suivre
- construire
  - tester
  - mesurer
  - corriger
- livrer

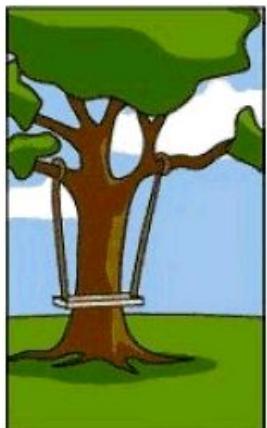
## CLIENT

« Un cahier des charges ? Pas la peine, faites directement un site qui permet tout, comme ça je ne vous dérange plus. »

# Ma vie de chef de projet



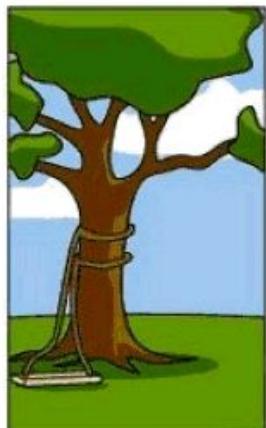
How the customer explained it



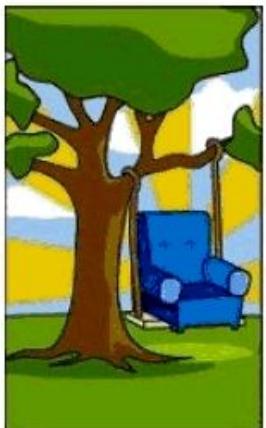
How the project leader understood it



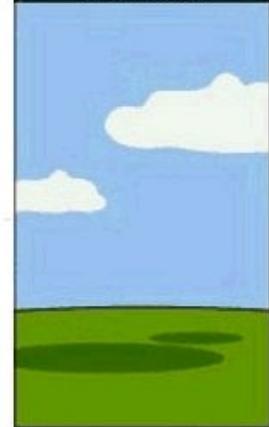
How the engineer designed it



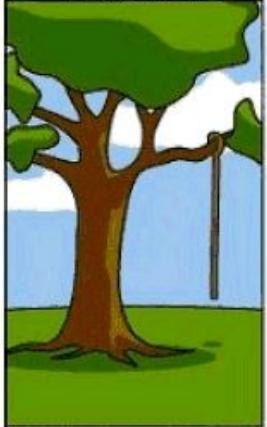
How the programmer wrote it



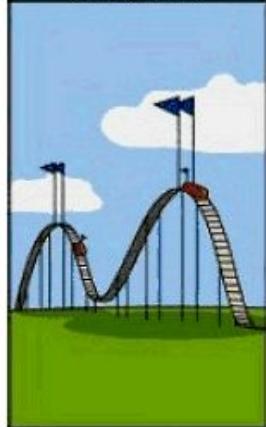
How the sales executive described it



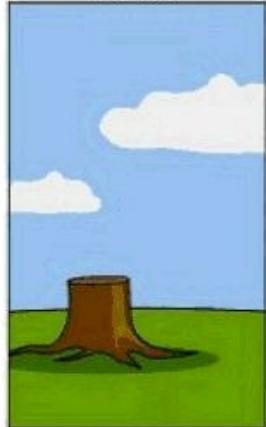
How the project was documented



What operations installed



How the customer was billed



How the helpdesk supported it



What the customer really needed

# Ma vie de chef de projet

## Approche Lean

Not like this....



1



2



3



4

---

Like this!



1



2



3



4



5

**Ma vie de chef de  
projet**

**Rapide**

ça risque d'être

**Laid**

c'est

**NON**

**Pas cher**

ça sera plus  
**Lent**

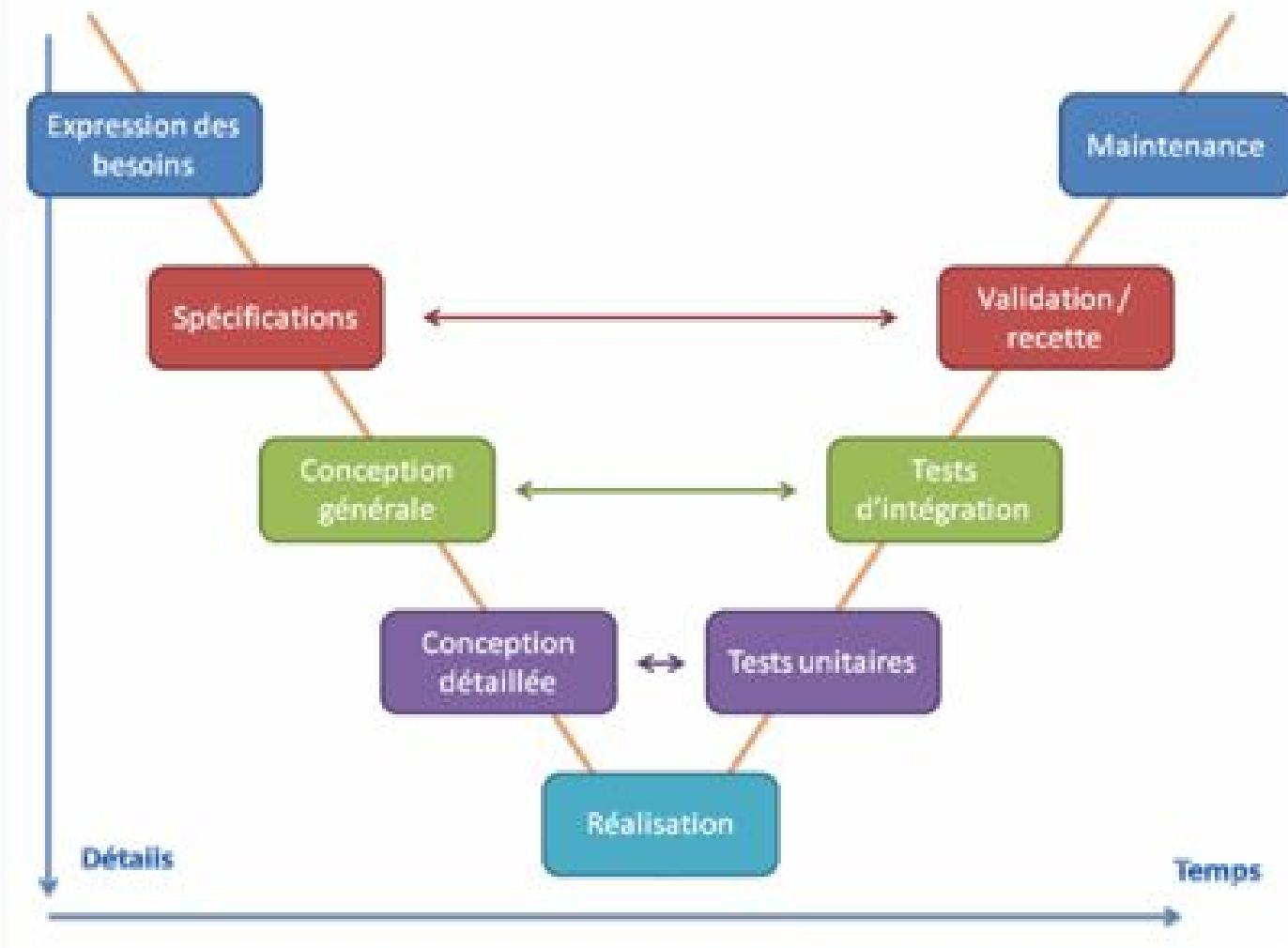
ça sera plus  
**Cher**

**De bonne**

# Ma vie de chef de projet

- Chiffrer = faire un devis
- ... Cycle en V “classique”
- comparer avec des projets similaires
- **découper en tâches atomiques**
- appliquer des ratios (des abaques peuvent exister)
  - pilotage
  - recette
  - tests
  - documentations
- Effet tunnel

# Ma vie de chef de projet



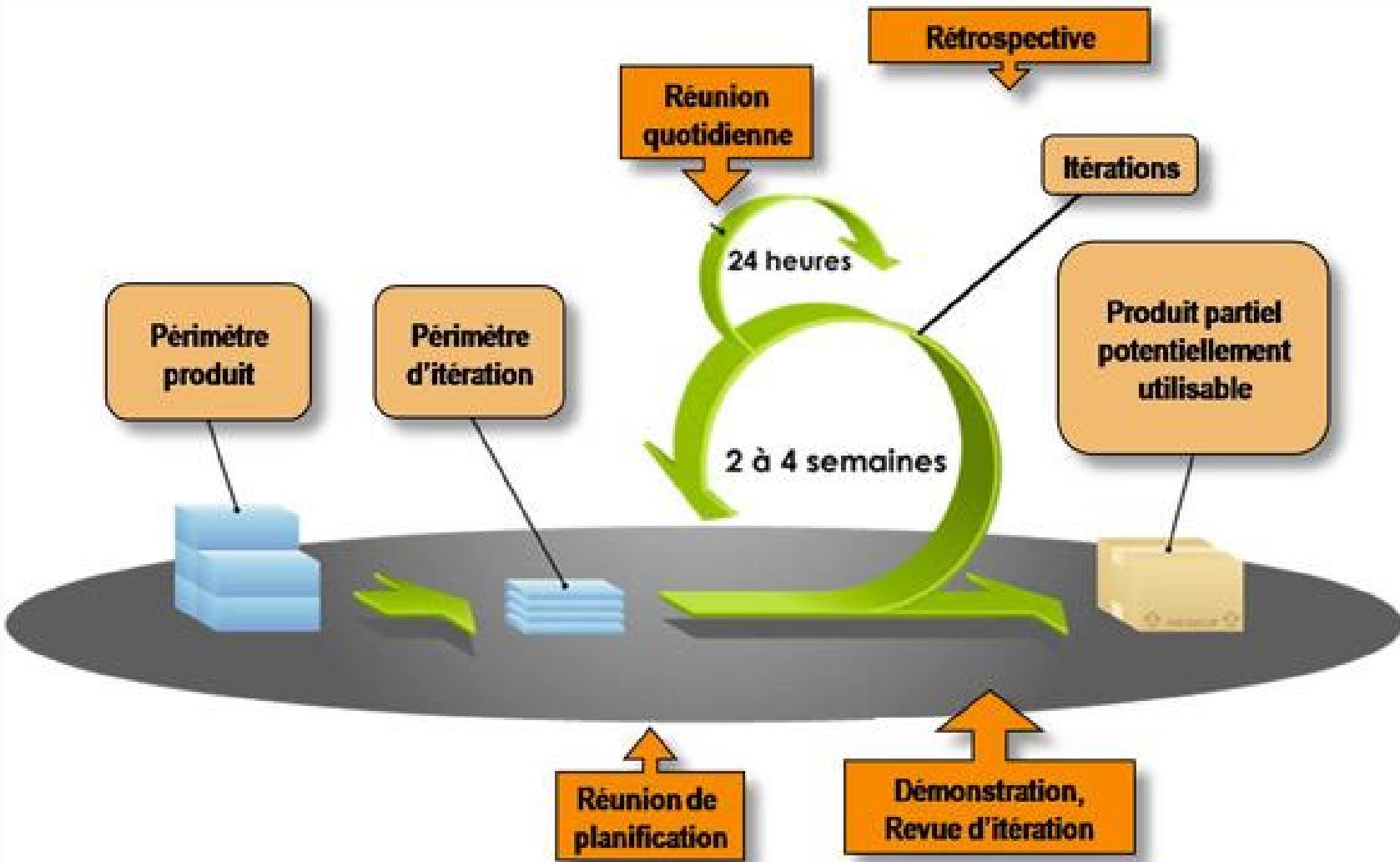
Chiffrer = faire un devis

... méthode agile = périmètre fixe

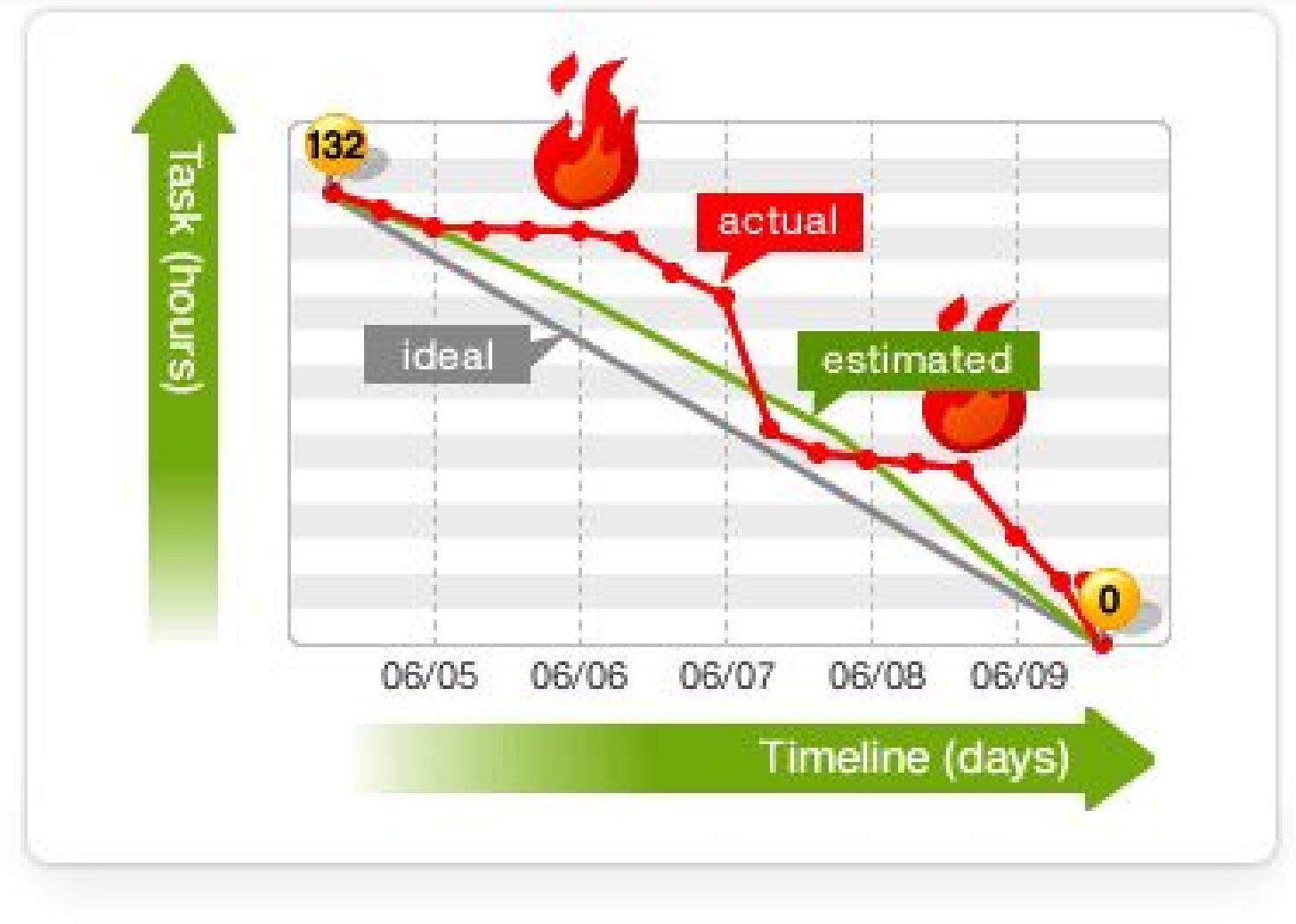
- découper en tâches atomiques
- catégoriser par valeur business
- chiffrer en points de complexité
- impliquer le client
- beaucoup de refactoring
- risque de manque de recul



# Ma vie de chef de projet



# Ma vie de chef de projet



# Ma vie de chef de projet

Client peu présent, peu mature -> Cycle en V

Client impliqué et réceptif -> Agile

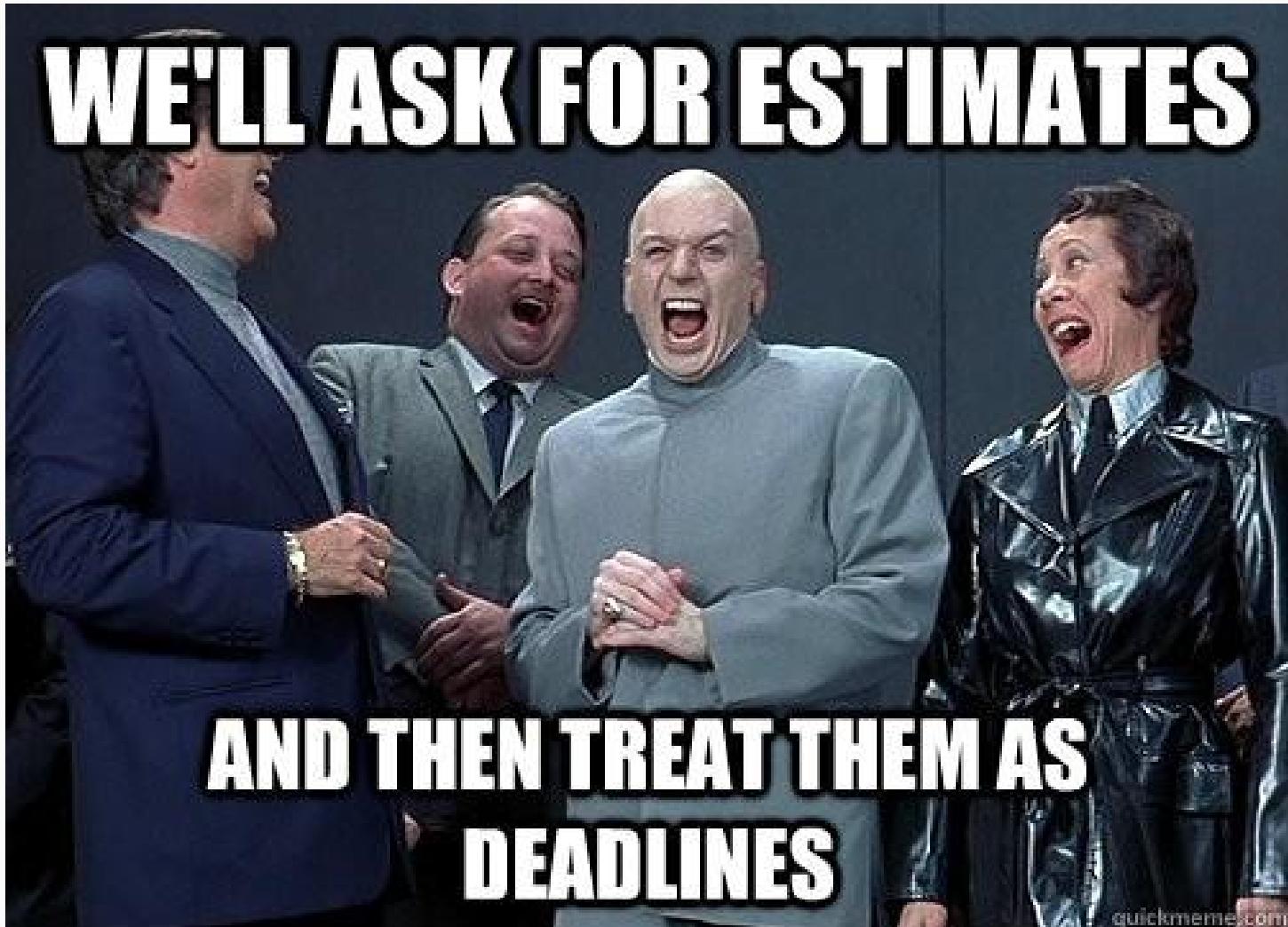
Un chiffrage c'est dur à faire

- abaques
- expérience
- analogie
- découpage en tâches atomiques

## Ma vie de chef de projet

découper en tâches atomiques







# Ma vie de chef de projet

Les estimations ne sont que des estimations

- suivi des consommés
- maîtrise des risques
  - plans d'actions
- pourcentage d'erreur
  - indisponibilités
  - problèmes techniques
  - ...



## La problématique

Des livrables configurables

Sortir de l'enfer des jars

Garder des livrables stables

Déploiement en un clic

Se préparer à l'avenir

## Travailler en équipe - Forge

- Référentiel exigences
- SCM
- Documentation
- Construction
- Tests
- Déploiements
- Bug tracker



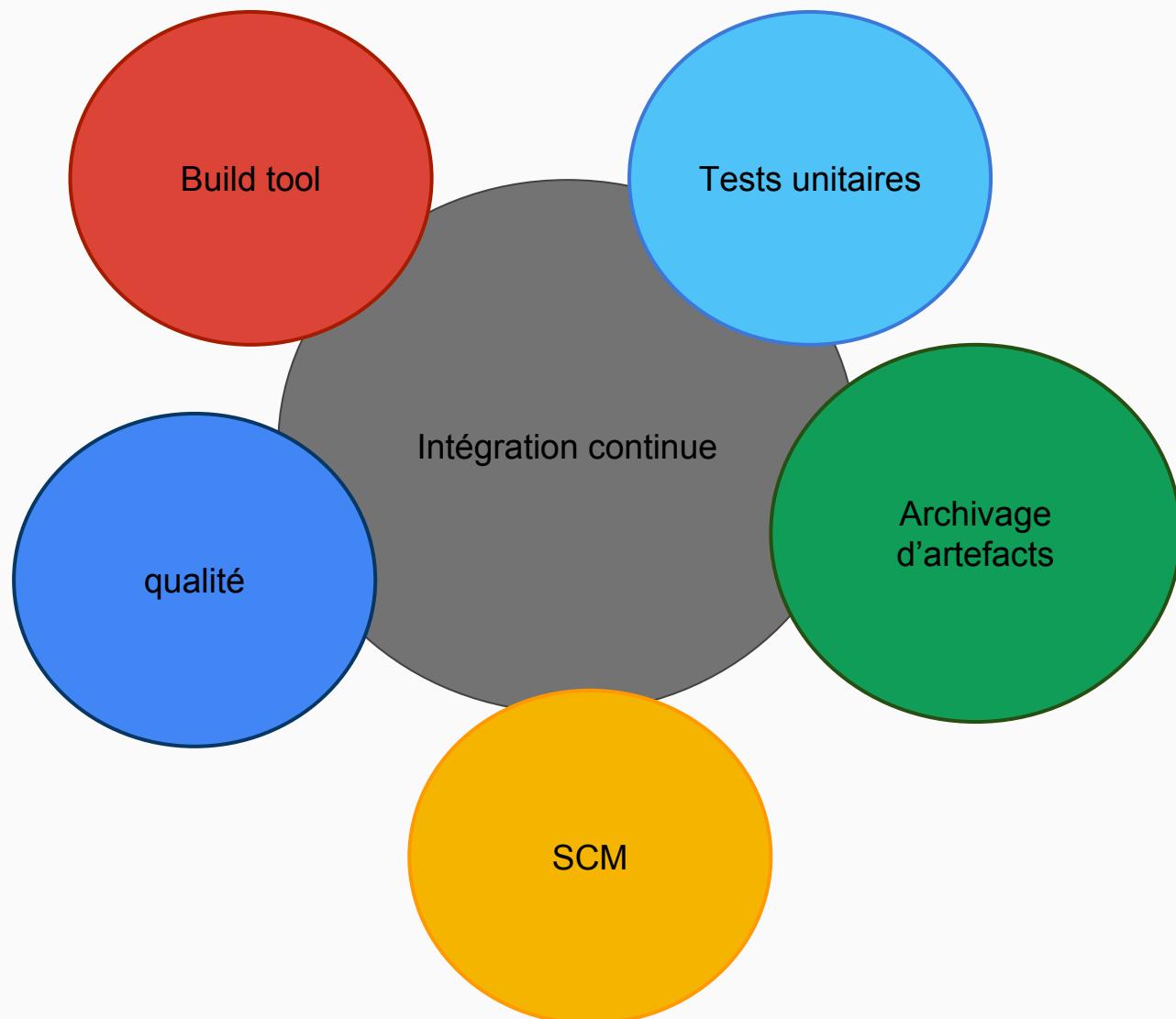
A photograph of a dark industrial interior, likely a foundry or steel mill. The scene is dominated by deep shadows, punctuated by bright orange and yellow light sources from overhead machinery and conveyor belts. A large mechanical arm or conveyor system is visible in the center, with a chain hanging from it. In the background, there are more industrial structures and what appears to be molten metal or glowing debris. The overall atmosphere is hazy and industrial.

# Les composants de l'écosystème

Une forge?

- Amélioration de la qualité des logiciels
- Amélioration de la traçabilité : du développement à la production
- Garantir la pérennité
- Amélioration de la productivité des développements

# Forge logicielle



# Travailler en équipe - Forge

## Un référentiel des exigences

- tracer les demandes client
- lister les epics / user-stories
- lier le code à la user-story et inversement
- aide à définir et comprendre le besoin du client
- priorisation des développements

Jira / IceScrum / ...

The screenshot shows a Jira Scrum board titled "Scrum: Teams in Space". The board is divided into four columns: To Do, In Progress, Code Review, and Done. The "To Do" column contains two issues: TIS-46 and TIS-40. The "In Progress" column contains two issues: TIS-40 and TIS-68. The "Code Review" column contains one issue: TIS-67. The "Done" column contains three issues: TIS-8, TIS-69, and TIS-49. On the right side of the board, there is a sidebar with details about the sprint: Reporter and Assignee are both Jennifer Evans; Created on 30/Jan/14 at 3:43 PM and Updated on 03/Feb/14 at 3:22 PM. The sidebar also shows development statistics: 5 branches, 3 commits, 1 pull request, and 1 build (status: latest 13/Feb/14). A "Create branch" button is also present.

Category	Value
People	Reporter: Jennifer Evans Assignee: Jennifer Evans
Dates	Created: 30/Jan/14 3:43 PM Updated: 03/Feb/14 3:22 PM
Development	5 branches 3 commits 1 pull request 1 build (status: latest 13/Feb/14)

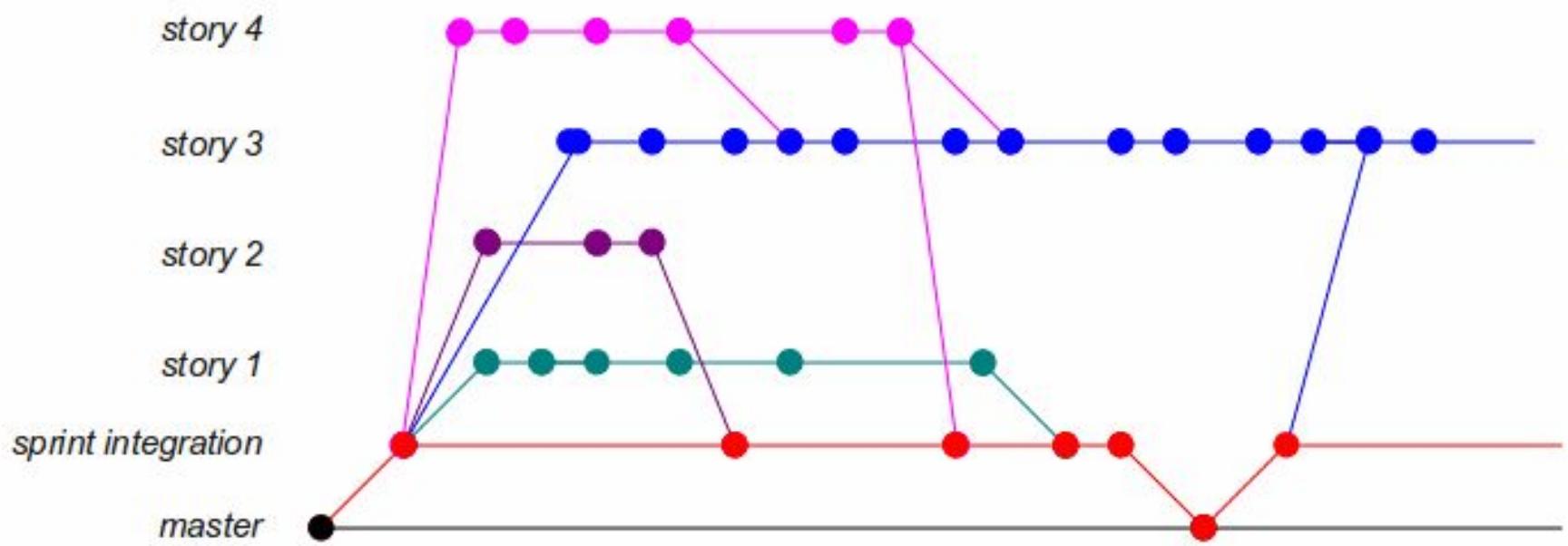
# Travailler en équipe - SCM

- Source Control Management
  - Git
  - Subversion
  - CVS
  - Mercurial
  - Clearcase
  - ...



# Travailler en équipe - SCM

- Branches
- Tags
- Commits



# Travailler en équipe - SCM

- Versionning (SEMVER)



<http://semver.org/>

# Travailler en équipe - Doc

- JavaDoc / JSDoc / APIdoc
- Wiki

“Any fool can write code that a computer can understand. Good programmers write code that humans can understand.”  
– Martin Fowler

“Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live.”  
– Martin Golding



# Travailler en équipe - Construire

Hey Jean-Claude, il paraît que tu as développé une super lib pour requêter le web-service...

Pas de soucis, je te l'envoie par mail

Et il y a besoin de quelles autres libs pour la faire fonctionner ?

HttpClient, Jackson, euuuh, Log4j, et euuuh

Quelles versions?

... j'ai une réunion, j'te laisse

# Travailler en équipe - Construire

Hey Jean-Claude, il paraît que tu as mis à jour  
ta super lib ...

Oui, hier, tu as quelle version?

la 0.2.3.00.25

Ah, on en est à la 5.25.665.14 ...

les librairies dépendantes sont à monter en  
version aussi?

... j'ai une réunion, j'te laisse

# Travailler en équipe - Construire

- Gestion de dépendances
- Archétypes
- Java
  - Dépôts Maven
    - Maven
    - Ivy
    - Gradle
- Javascript
  - NPM (Node Package Manager)
  - Yeoman
  - Bower (Basé sur Github)



maven



npm



# Travailler en équipe - Construire

```
<project>
...
<dependencies>
  <dependency>
    <groupId>group-a</groupId>
    <artifactId>artifact-a</artifactId>
    <version>1.0</version>
    <exclusions>
      <exclusion>
        <groupId>group-c</groupId>
        <artifactId>excluded-artifact</artifactId>
      </exclusion>
    </exclusions>
  </dependency>
  <dependency>
    <groupId>group-a</groupId>
    <artifactId>artifact-b</artifactId>
    <version>1.0</version>
    <type>bar</type>
    <scope>runtime</scope>
  </dependency>
</dependencies>
</project>
```



# Travailler en équipe - Construire

dependencies {

```
compile fileTree(dir: 'libs', include: ['*.jar'])
apt "org.androidannotations:androidannotations:$AAVersion"
compile "org.androidannotations:androidannotations-api:$AAVersion"
compile 'com.android.support:appcompat-v7:22.1.1'
compile 'com.google.android.gms:play-services:7.3.0'
compile 'com.android.support:support-v4:22.1.1'
compile 'org.apache.commons:commons-lang3:3.4'
compile 'com.google.code.gson:gson:2.3.1'
compile 'com.android.support:cardview-v7:21.0.3'
compile('au.com.datasymphony:EasyFlow:1.3.1') {
    exclude group: 'junit'
    exclude group: 'slf4j-api'
    exclude group: 'slf4j-log4j12'
}
```



# Travailler en équipe - Construire

- Gestion de dépendances
- Dépôt public sur le Web
- Cache d'entreprise (optionnel)
- Cache local
- Résolution des dépendances transitives ou non
- SEMVER



# Travailler en équipe - Construire

- Compiler / assembler
- Java
  - Maven
  - Gradle
  - Ant
  - ...
- Javascript
  - Gulp
  - Grunt



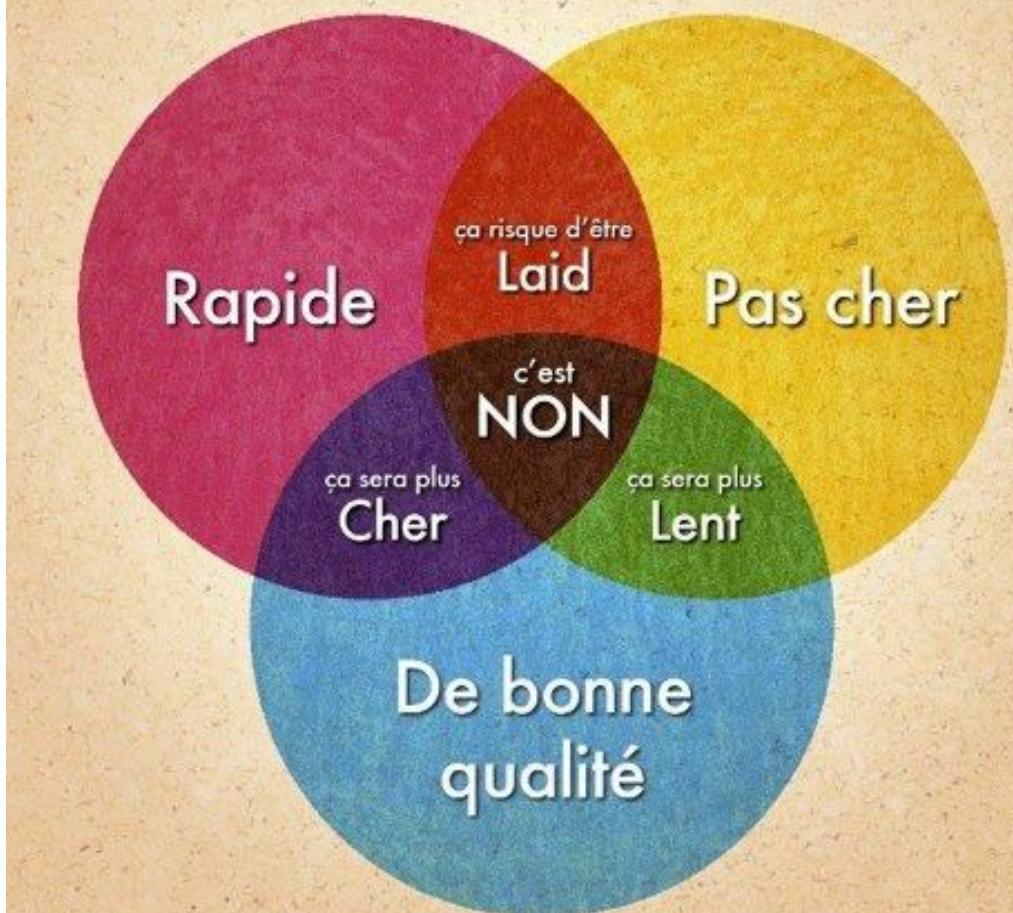
**maven**

The Maven logo is displayed in a white rectangular box. It features the word "maven" in a lowercase, bold, sans-serif font. The letter "a" is unique, composed of two overlapping shapes: a black shape on the left and an orange shape on the right.

## Travailler en équipe - Qualité

VOUS LE VOULEZ COMMENT  
VOTRE PROJET ?

(VOUS POUVEZ FAIRE JUSQU'À DEUX CHOIX)



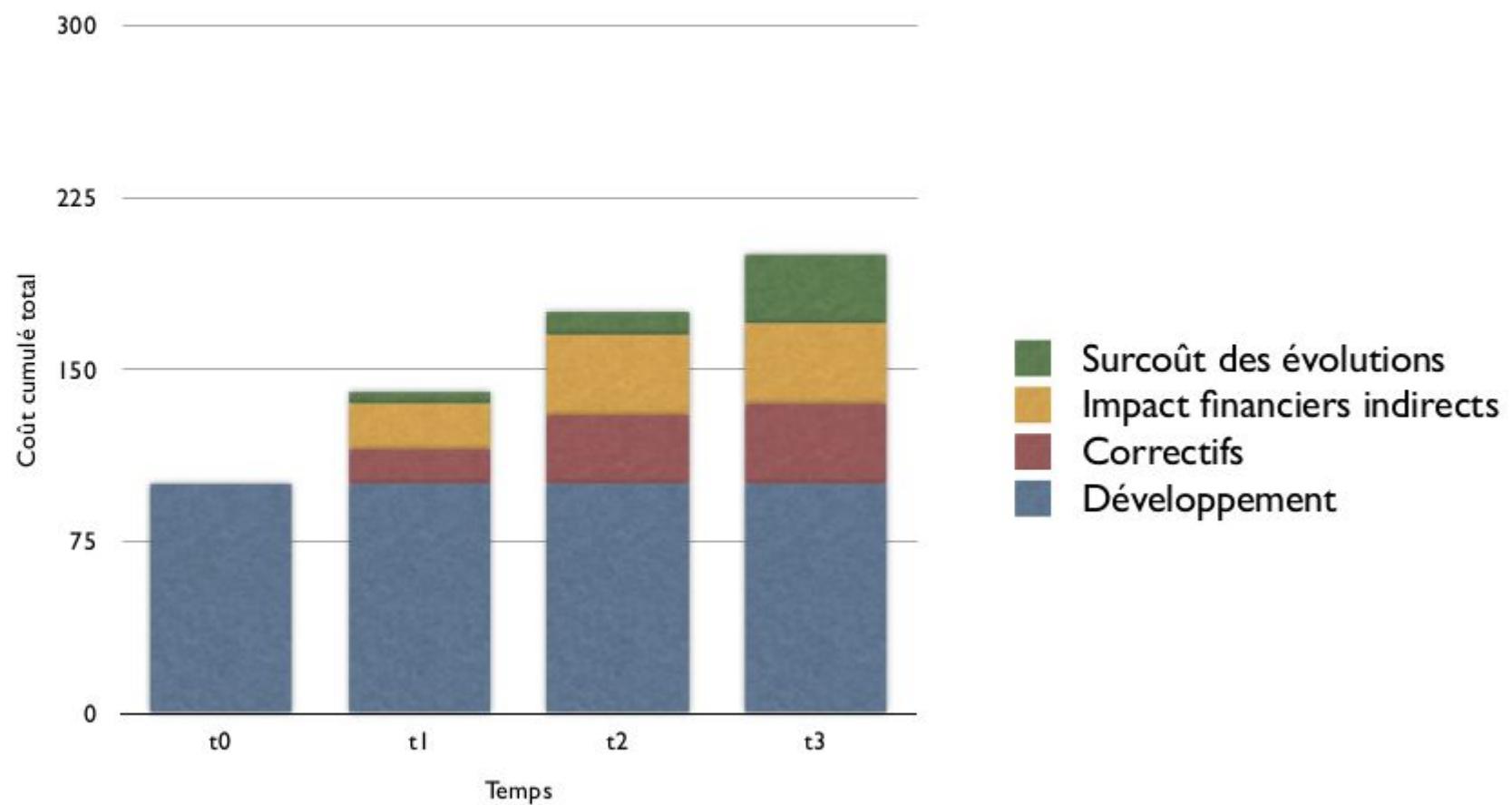
"Gratuit" n'est pas une option.

# Travailler en équipe - Qualité

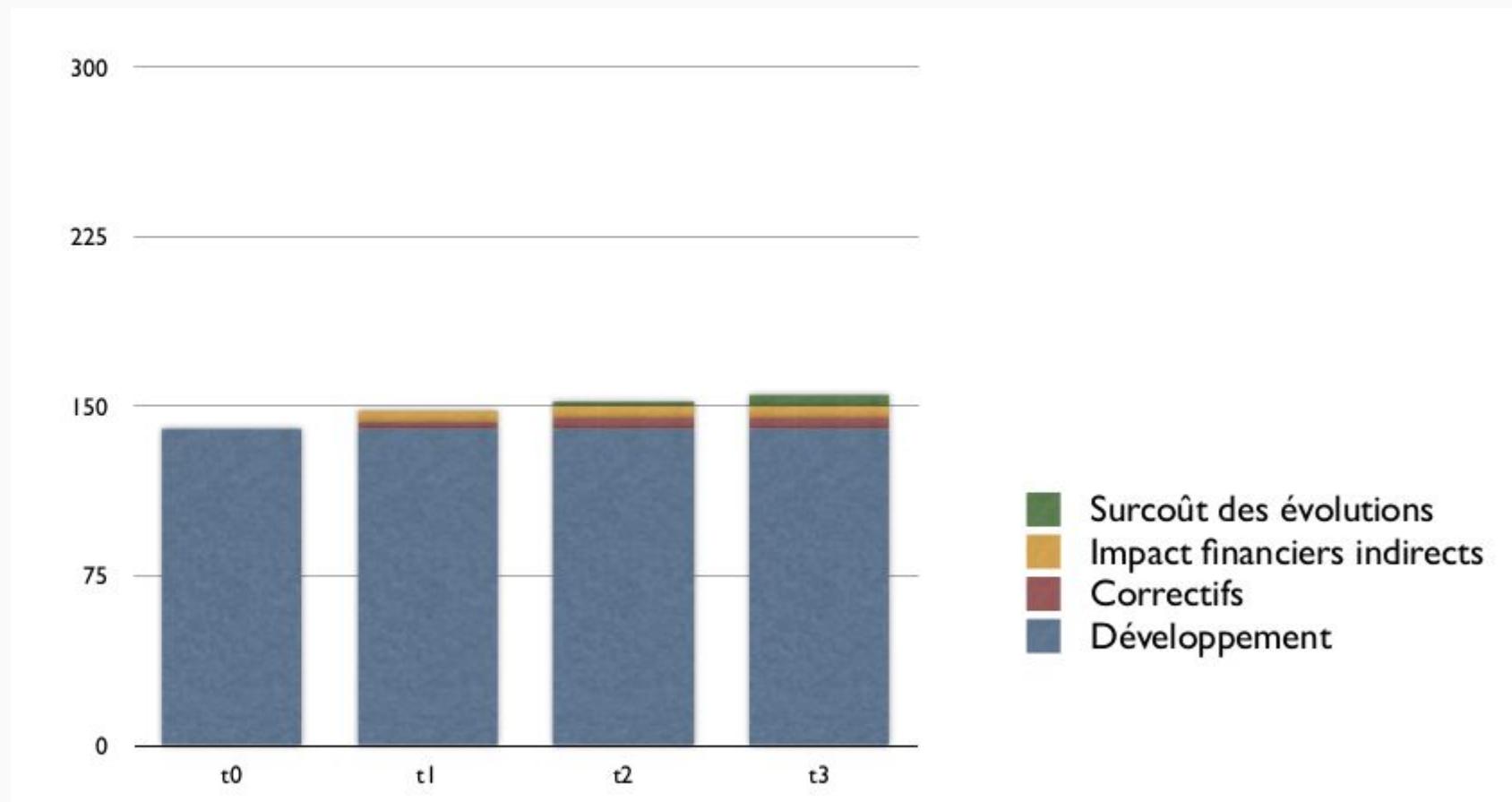
“La qualité fait chuter ma productivité”



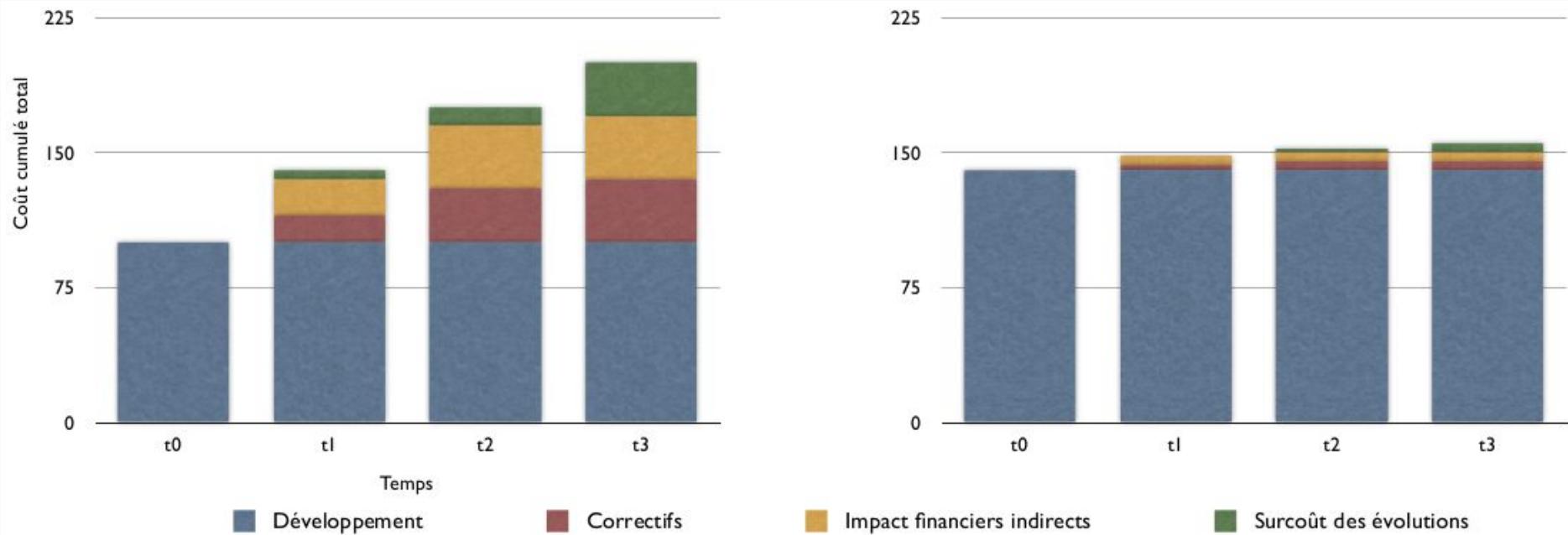
## Travailler en équipe - Qualité



## Travailler en équipe - Qualité



# Travailler en équipe - Qualité



# Travailler en équipe - Qualité

Coûts de la qualité



bugs

coûts de correction

performances

arrêt du projet

manque à gagner

notoriété

image

adoption

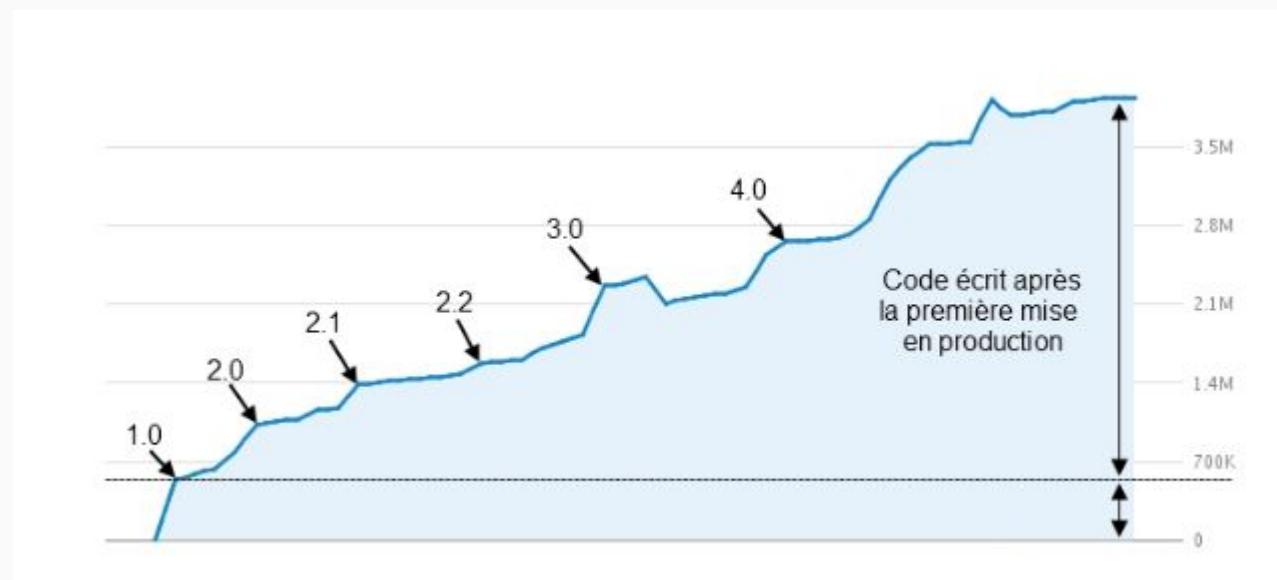
utilisation

surcoût des  
évolutions

Coûts de la non qualité

# Travailler en équipe - Qualité

- Votre logiciel va évoluer :
- 60% à 80% des coûts en maintenance
- dont 75% évolutive
- Source: Software Maintenance, par G. Ganfora



# Travailler en équipe - Qualité

- Augmenter la durée de vie de nos applications :
  - Améliorer la maintenabilité
  - Améliorer la documentation
  - Réduire le code dupliqué
  - Respecter des pratiques communes de développement
- Déetecter au plus tôt certains bugs classiques
- Améliorer les performances par la détection d'anti-patterns

# Travailler en équipe - Qualité

Pair programming

Tu as oublié un  
point-virgule...



C'est du Javascript !

# Travailler en équipe - Qualité

## Analyse statique / dynamique (Sonar)

Sonar logo

Commons Collections

Version 3.3-SNAPSHOT - Sun, 21 Feb 2010 18:12 - profile Sonar.woc

**Home** **Search** Configuration Log in

**Dashboard**

**Components**

**Violations drilldown**

**Time machine**

**Clouds**

**Design**

**Hotspots**

**sonar**

**Lines of code** **20,901**  
52,871 lines

**Classes** **371**  
13 packages  
3,161 methods  
463 accessors

**Comments** **42.4%**  
15,369 lines  
63.2% docu, API  
965 undocu. API  
74 commented LOKs

**Duplications** **3.3%**  
1,744 lines  
60 blocks  
30 files

**Complexity** **2.1 / method**  
**17.9 / class**  
6,956 cmplx  
8,956 statements

**Methods** **Classes**

**Code coverage** **79.8%**  
82.0% line coverage  
74.9% branch coverage  
11,021 tests  
55.7 sec

**Test success** **100.0%**  
0 failures  
0 errors

**Rules compliance** **82.3%**

**Violations** **2,285**

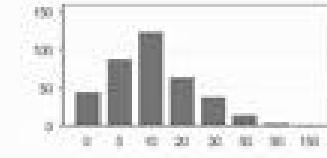
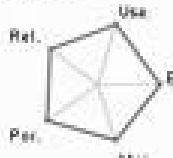
- Blocker: 0
- Critical: 24
- Major: 678
- Minor: 1,536
- Info: 47

**LCOM4** **2.2 / class**  
31.5% suspect files (LCOM4>1)

**RFC** **22 / class**

**Package tangle index** **49.3%**  
161 cycles

**Dependencies to cut**  
12 between packages  
135 between files



# Travailler en équipe - Qualité

## Analyse statique / dynamique (Sonar)

- **Apports**
  - Vérification des pratiques de codages
  - Détection de bugs
  - Référentiel qualité du code source
- **Non apports**
  - Ne valide pas la performance de l'application
  - Ne valide pas la sécurité
  - Ne valide pas la capacité du logiciel à répondre au besoin fonctionnel

# Travailler en équipe - Qualité

## Revue de code (Crucible)

Screenshot of a code review interface (Crucible) showing a Java applet named "Blink".

The code is as follows:

```
01. import java.awt.*;
02. import java.util.*;
03.
04. public class Blink extends java.applet.Applet {
05.     private Timer timer;          // Schedules the blinking
06.     private String labelString;   // The label for the window
07.     private int delay;           // the delay time between blinks
08.
09.     public void init() {
10.         String blinkFrequency = getParameter("speed");
11.         delay = (blinkFrequency == null) ? 400 :
12.             (1000 / Integer.parseInt(blinkFrequency));
13.         labelString = getParameter("lbl");
14.         if (labelString == null)
15.             labelString = "Blink";
16.         Font font = new java.awt.Font("Serif", Font.PLAIN, 24);
17.         setFont(font);
18.     }
19.
20.     public void start() {
21.         timer = new Timer();        //creates a new timer to schedule the blinking
22.         timer.schedule(new TimerTask() {      //creates a timertask to schedule
23.             // overrides the run method to provide functionality
24.             public void run() {
25.                 repaint();
26.             }
27.         });
28.     }
29. }
```

The code review interface includes a sidebar for navigation and user information, and a main area for viewing the code and adding comments.

Comments are visible on the right side:

- Brendan Humphreys (19:31): consider using HTML5. [Reply](#)
- Seb Ruiz (19:34): Absolutely. [Reply](#)
- Seb Ruiz (19:32): This will fail when blinkFrequency is an empty string. Consider pulling this out into an if statement for greater readability. [Reply](#) [Defect](#)

# Travailler en équipe - Tester

- Tests unitaires
  - Junit
  - Selenium
  - Karma
  - ...
- Tests end to end
- Tests fonctionnels
- Bug trackers
  - Jira
  - Mantis
  - ...

CLIENT

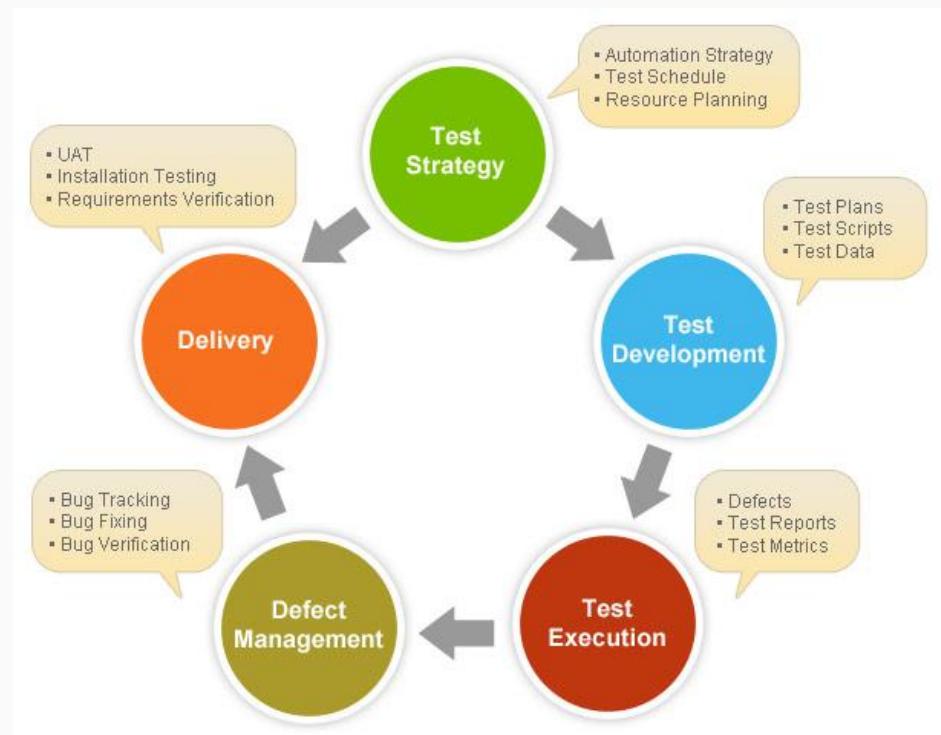
« Vous corrigez les bugs ? Pourquoi n'avez vous pas directement codé le site sans bug ? »

@webAgencyFAIL

PIWEE

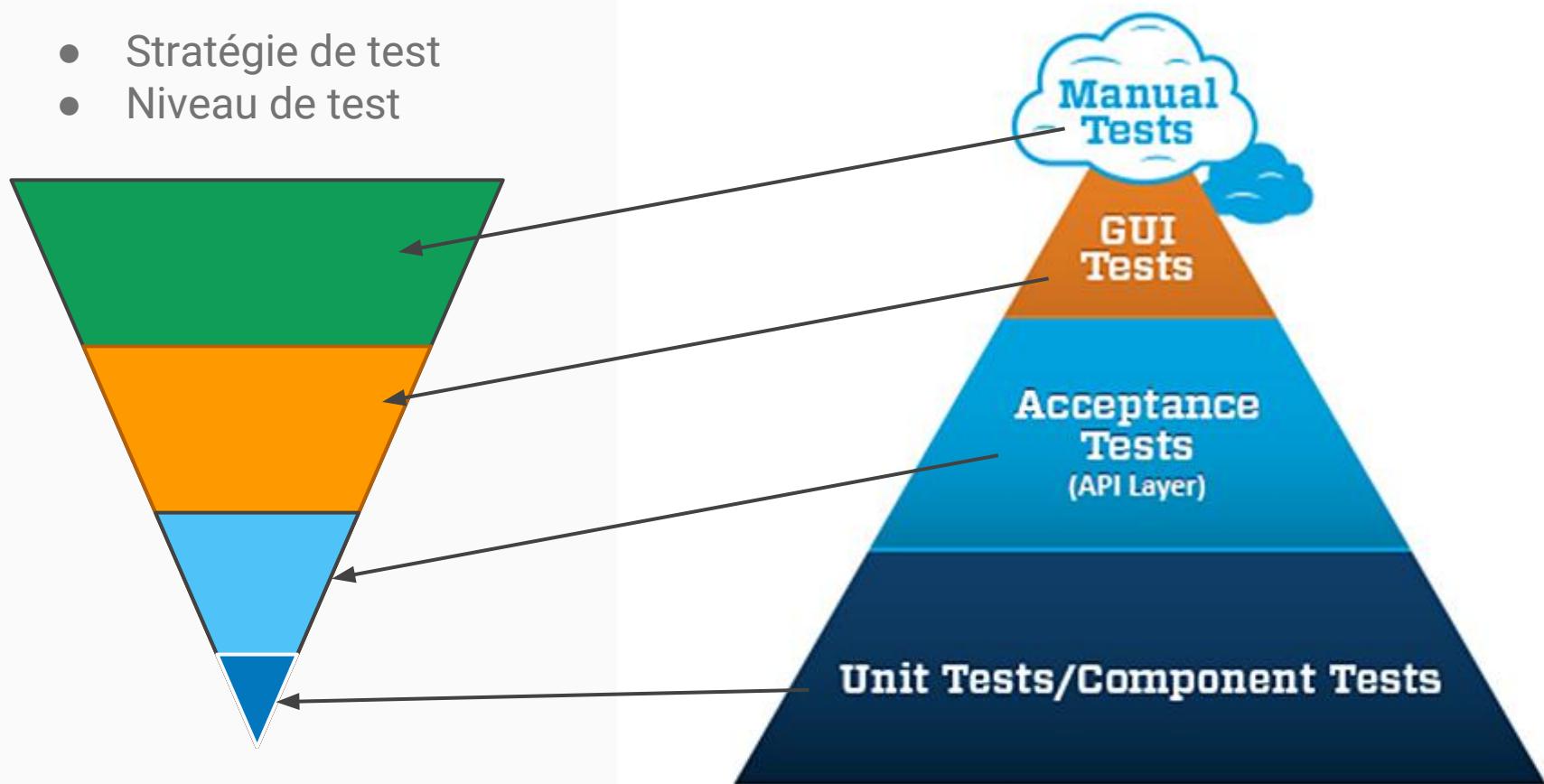
# Travailler en équipe - Tester

- Tests unitaires
  - Junit
  - Selenium
  - Karma
  - ...
- Tests end to end
- Tests fonctionnels

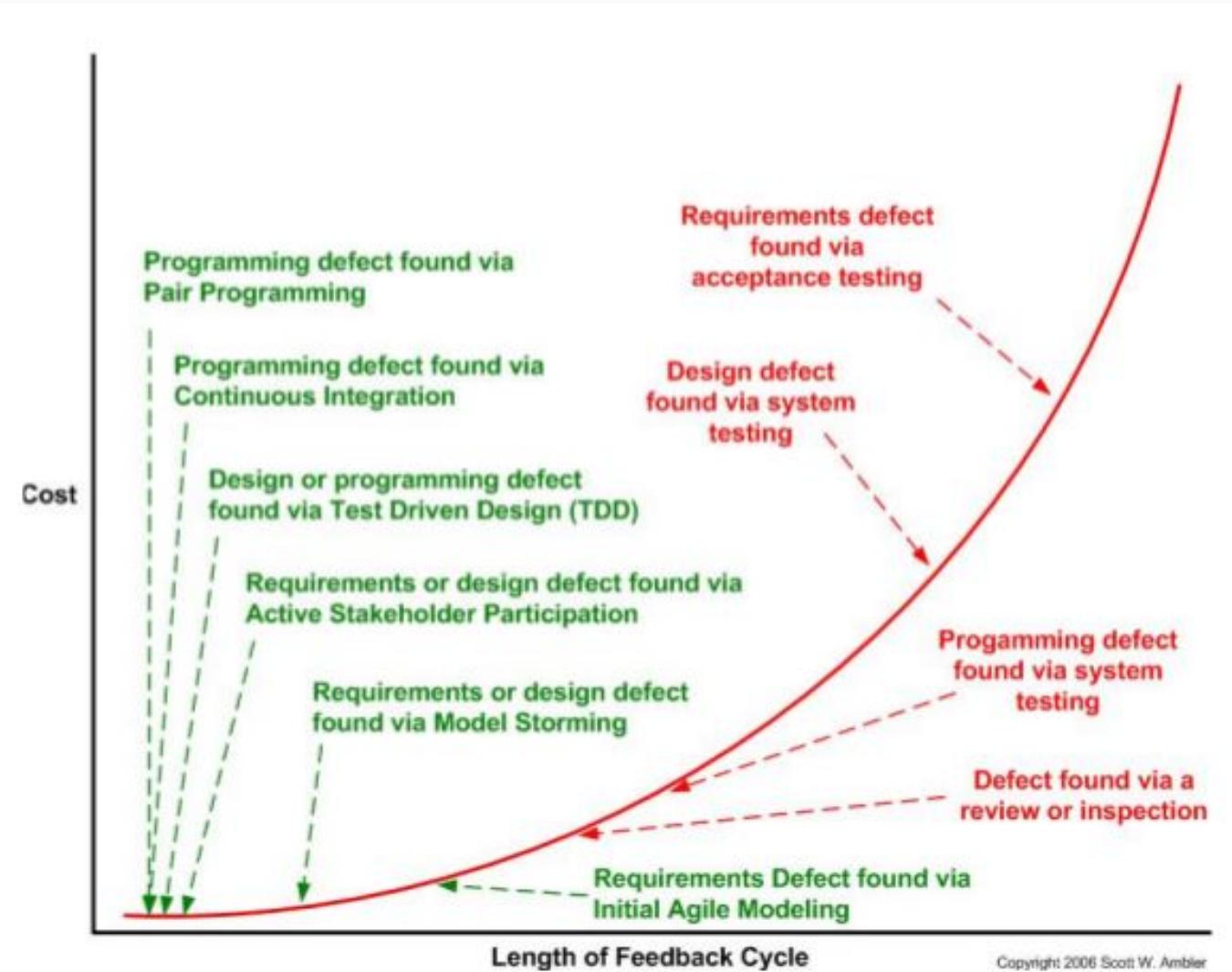


# Travailler en équipe - Tester

- Stratégie de test
- Niveau de test



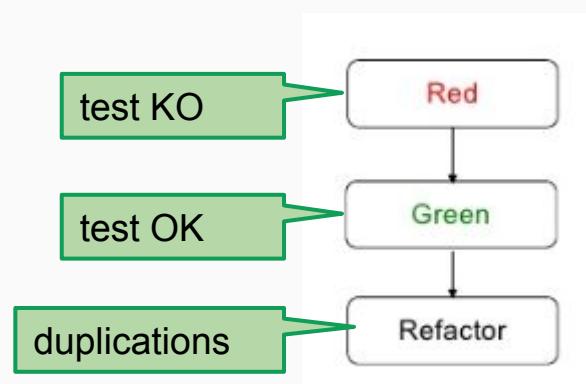
# Travailler en équipe - Tester



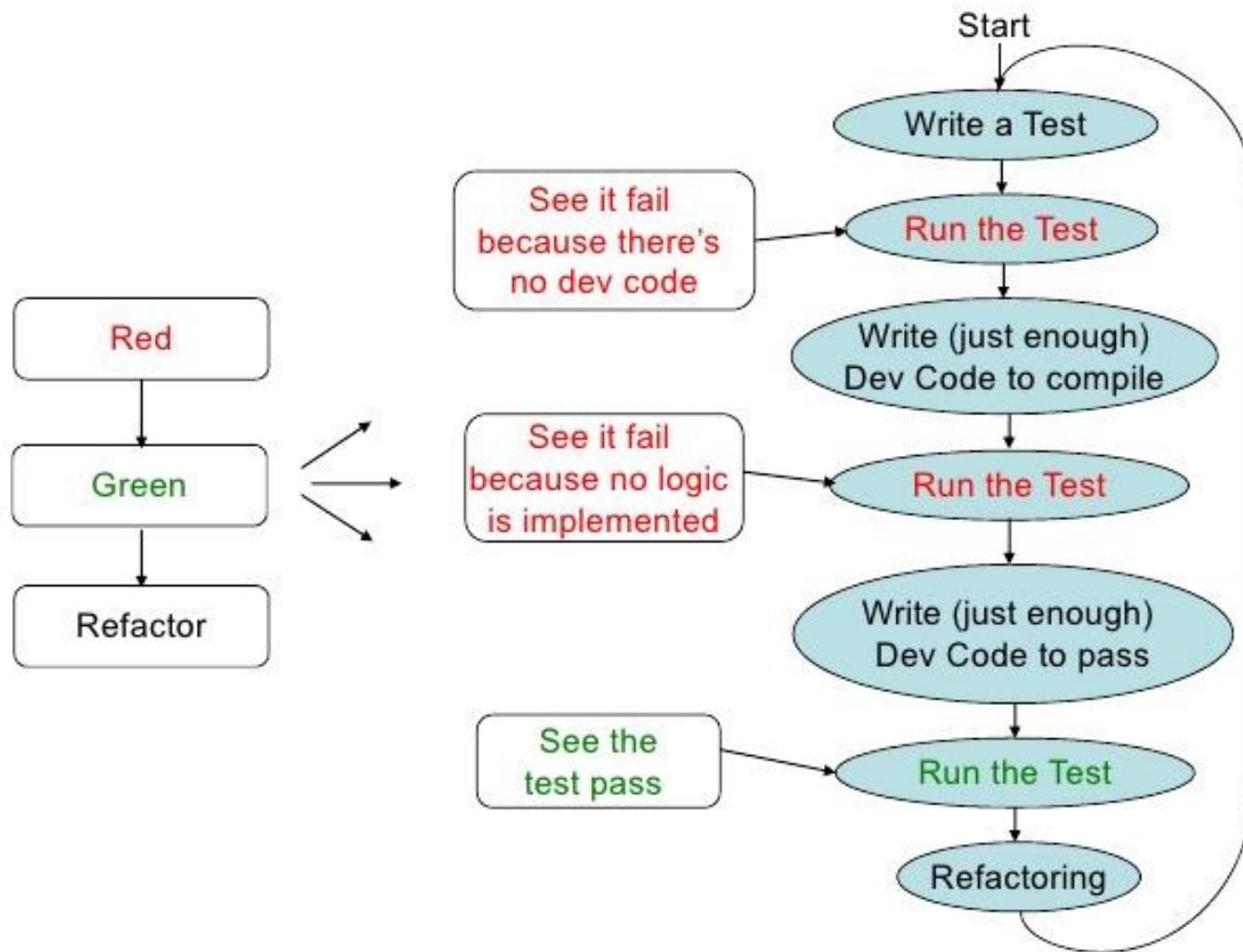
# Travailler en équipe - TDD

## Test Driven Developement

- Approche du dev par les tests
  - On code les tests puis on fait le dev
  - Comment utiliser un composant vs comment l'implémenter
  - Ne pas écrire une ligne de code sans un test ko
  - Éliminer les duplications



# Travailler en équipe - TDD



## Travailler en équipe - TDD

- Développeur
  - Tests unitaires
  - Implémentations des BDD
- Client (MOA / Product owner)
  - Tests BDD d'acceptation



<http://junit.org/>

```
@Test
public void newArrayListsHaveNoElements() {
    assertThat(new ArrayList().size(), is(0));
}

@Test
public void sizeReturnsNumberOfElements() {
    List instance = new ArrayList();
    instance.add(new Object());
    instance.add(new Object());
    assertThat(instance.size(), is(2));
}
```



- Business Driven Developpement
- Tests décrits avec une syntaxe et un vocabulaire fixe
- Traduit par le développeur

<https://cucumber.io/>

## Feature: Book search

To allow a customer to find his favourite books quickly, the library must offer multiple ways to search for a book.

### Scenario: Search books by publication year

Given a book with the title 'One good book', written by 'Anonymous', published in 14 March 2013

And another book with the title 'Some other book', written by 'Tim Tomson', published in 23 August 2014

And another book with the title 'How to cook a dino', written by 'Fred Flintstone', published in 01 January 2012

When the customer searches for books published between 2013 and 2014

Then 2 books should have been found

And Book 1 should have the title 'Some other book'

And Book 2 should have the title 'One good book'

# Travailler en équipe - BDD

- Junit + Cucumber

```
public class BookSearchSteps {  
    Library library = new Library();  
    List<Book> result = new ArrayList<>();  
  
    @Given(".+book with the title '(.+)', written by '(.+)', published in (.+)")  
    public void addNewBook(final String title, final String author, @Format("dd MMMMM yyyy") final Date published) {  
        Book book = new Book(title, author, published);  
        library.addBook(book);  
    }  
    @When("^the customer searches for books published between (\\d+) and (\\d+)\$")  
    public void setSearchParameters(@Format("yyyy") final Date from, @Format("yyyy") final Date to) {  
        result = library.findBooks(from, to);  
    }  
    @Then("(\\d+) books should have been found$")  
    public void verifyAmountOfBooksFound(final int booksFound) {  
        assertThat(result.size(), equalTo(booksFound));  
    }  
    @Then("Book (\\d+) should have the title '(.+)'$")  
    public void verifyBookAtPosition(final int position, final String title) {  
        assertThat(result.get(position - 1).getTitle(), equalTo(title));  
    }  
}
```

# Travailler en équipe - Intégration continue

- Déléguer les tâches répétitives
  - Compilation
  - Assemblage
  - Tests
  - Livraison
- Reporting
- Partager l'information
- Factuel



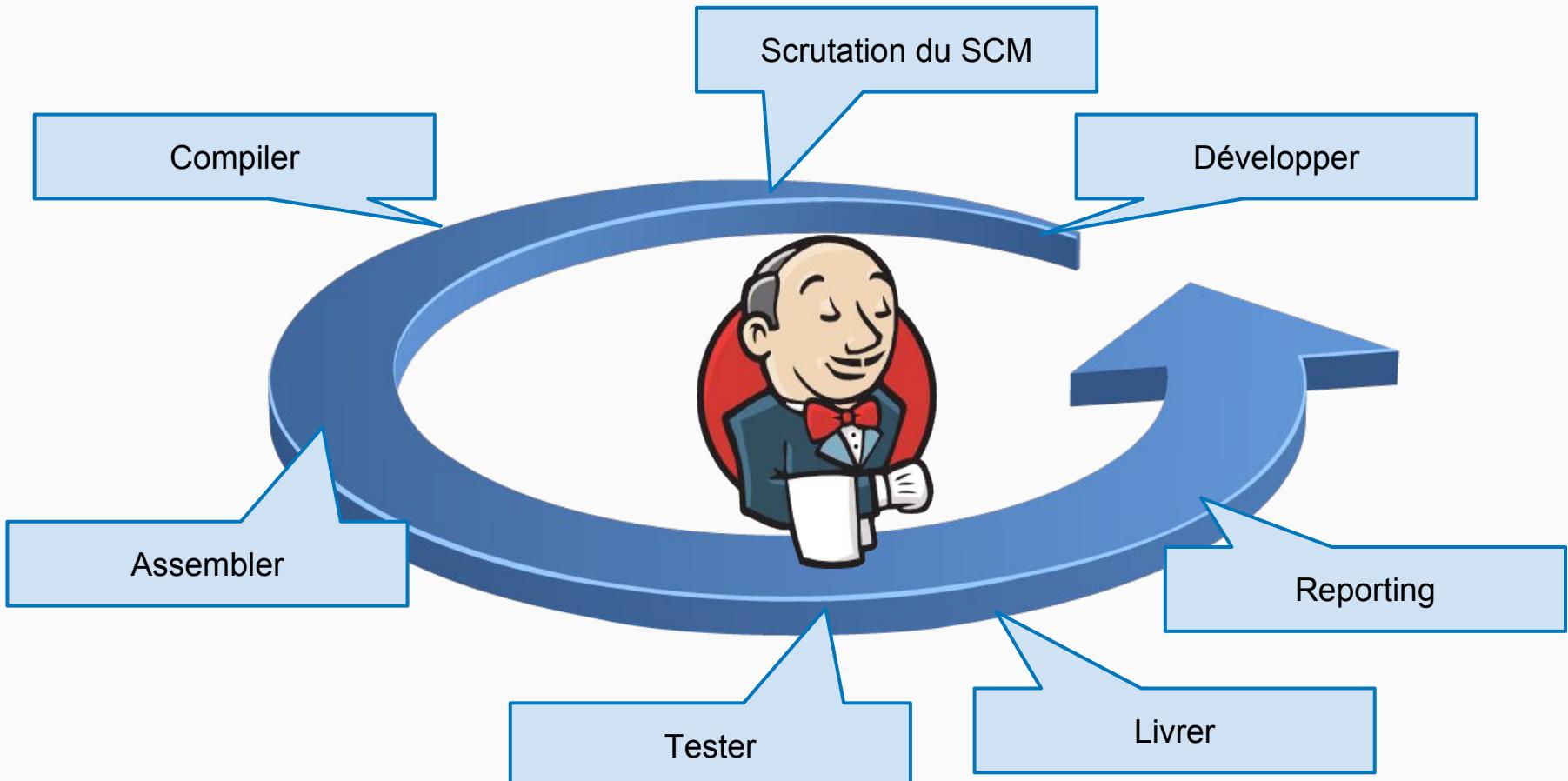
# Jenkins



# Travailler en équipe - Intégration continue

- Automatiser la construction et le lancement des tests
  - De manière périodique
  - Sur la base de la version courante sur le SCM
- Avertir l'équipe en cas de problèmes
  - Mail aux personnes concernées
- Présenter les résultats à l'équipe
  - Qualimétrie, résultats des tests, rapports de compilation
- Suivre l'impact des développements sur la qualité du build
  - Tendance des constructions

# Travailler en équipe - Intégration continue

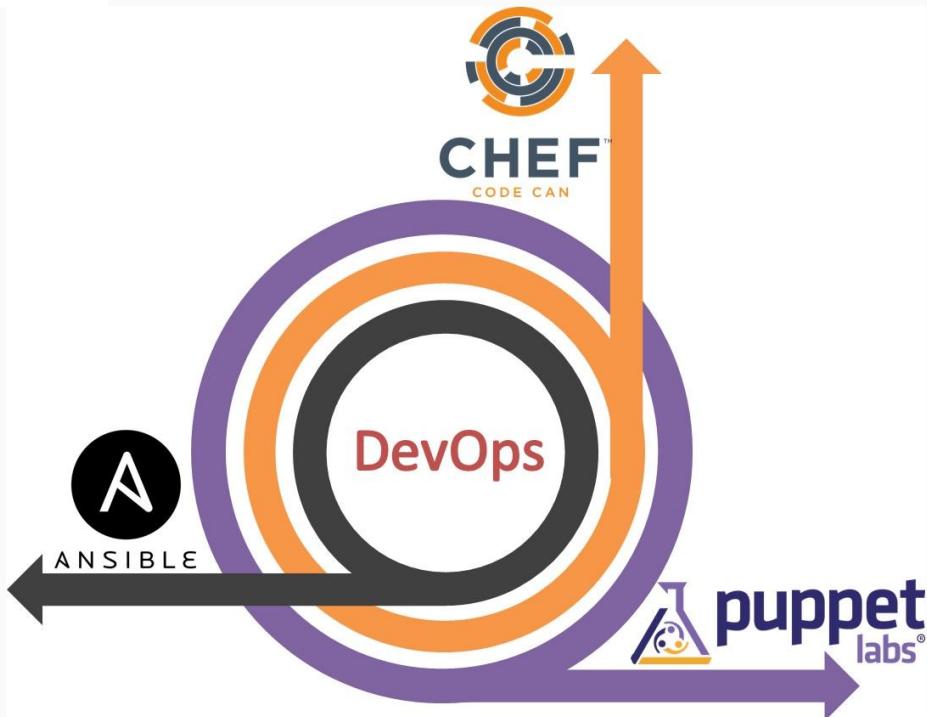
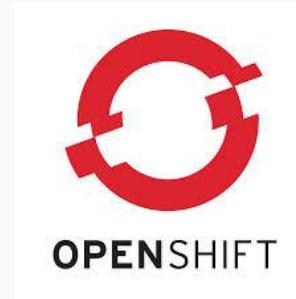


# Travailler en équipe - Intégration continue

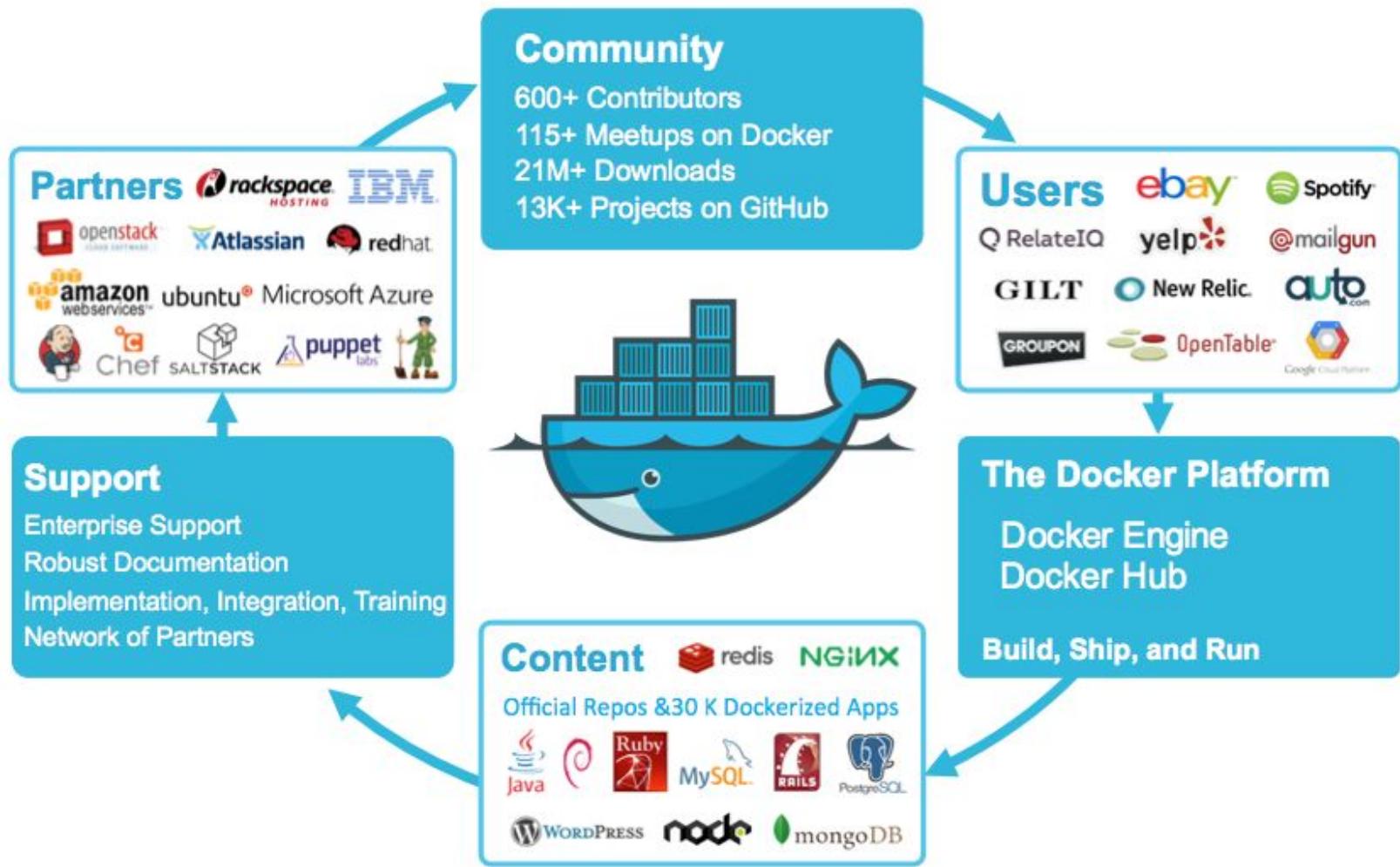
- Mise en œuvre des bonnes pratiques de développements
  - Commit régulier : retour rapide sur les erreurs
  - Écriture des tests unitaires : éviter les non-regressions
- Passer d'une démarche personnelle à une démarche d'équipe
- Vérifier en continu l'état des développements
- Libérer des tâches répétitives
- Avoir un référentiel de construction lors des développements

# Livraisons

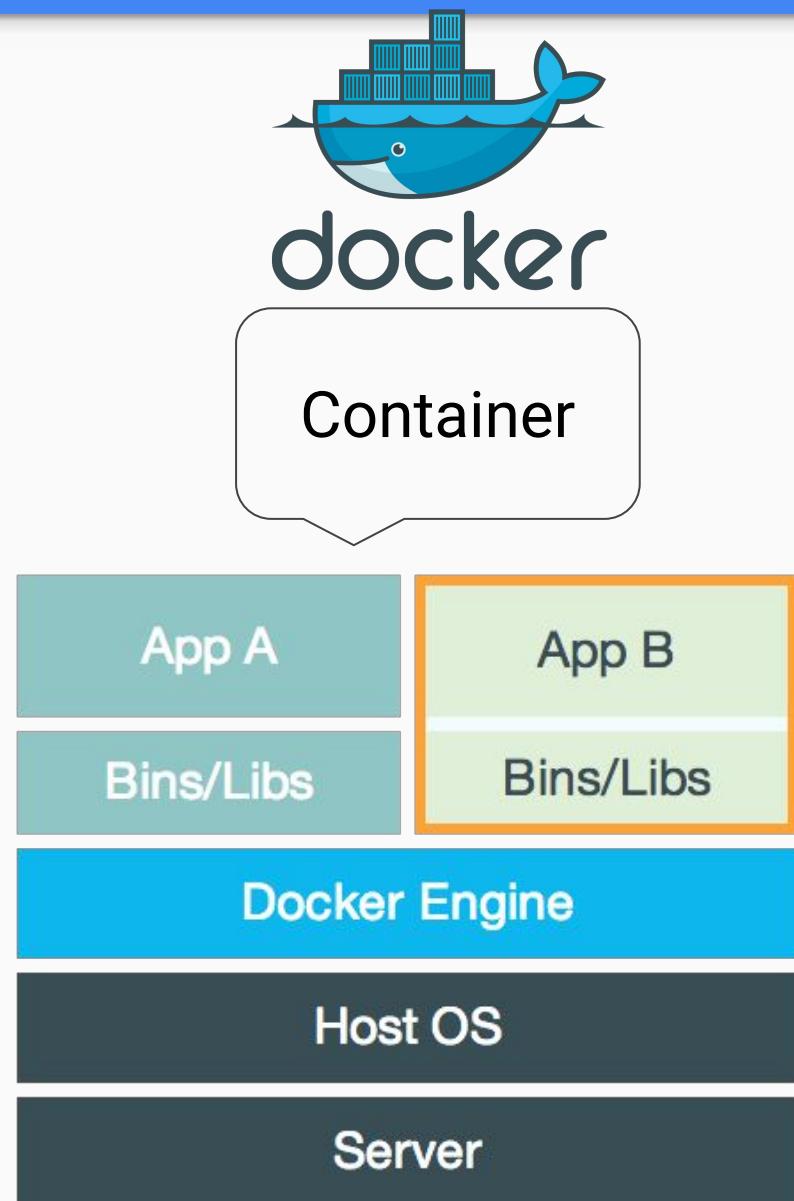
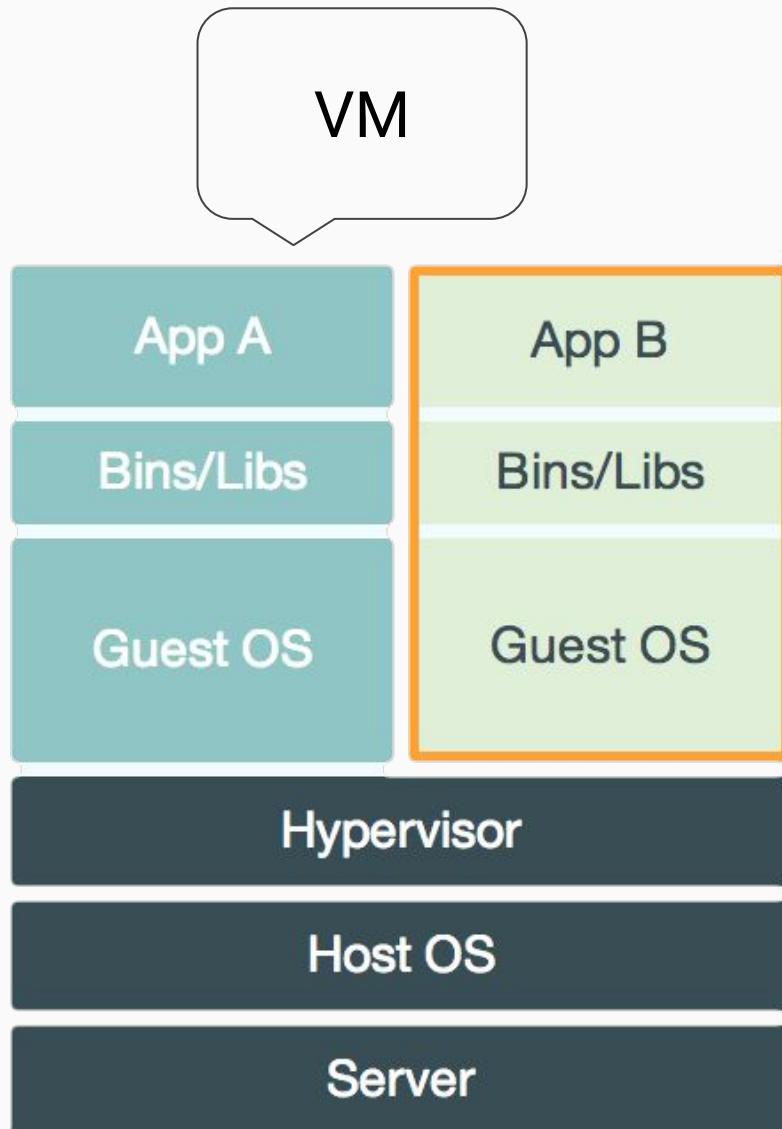
- A la main
- Dev/Ops
- Automatisée
  - OpenStack
  - OpenShift
  - Ansible
  - Puppet/Chef
  - ...
- Docker



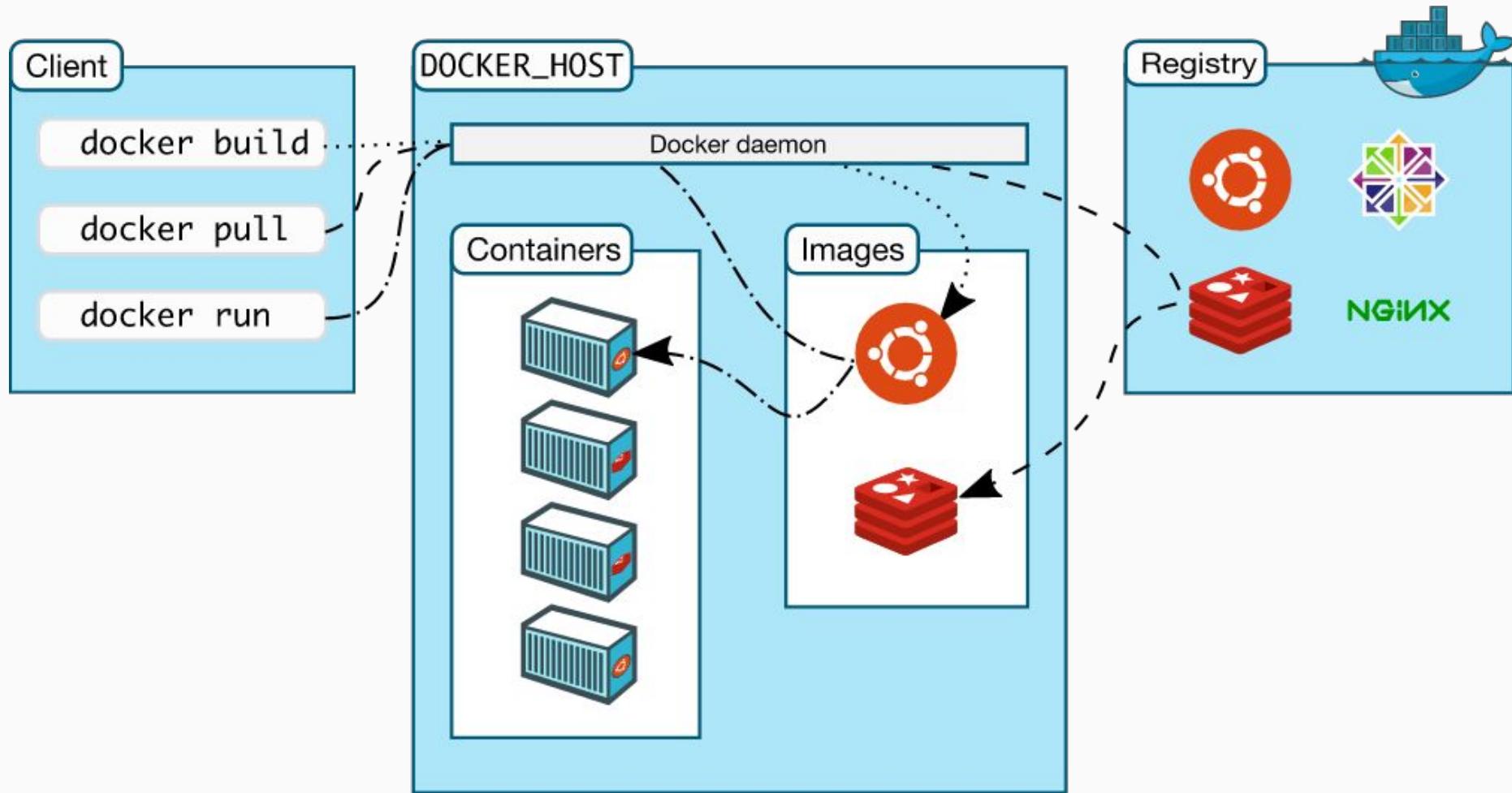
# Livraisons - Docker



# Docker



# Docker



# Docker - les images

The screenshot shows a web browser window displaying the Docker Hub page for the official CentOS repository. The URL in the address bar is [https://hub.docker.com/\\_/centos/](https://hub.docker.com/_/centos/). The page title is "Docker Hub". The search bar contains "centos".

**OFFICIAL REPOSITORY**

**centos**

Last pushed: 13 days ago

[Repo Info](#) [Tags](#)

**Note:** the description for this image is longer than the Hub length limit of 5000, so has been trimmed. The full description can be found at <https://github.com/docker-library/docs/tree/master/centos/README.md>. See [docker/hub-beta-feedback#238](#) for more information.

**Supported tags and respective Dockerfile links**

- latest , centos7 , 7 ([docker/Dockerfile](#))
- centos6 , 6 ([docker/Dockerfile](#))
- centos5 , 5 ([docker/Dockerfile](#))

**DOCKER PULL COMMAND**

```
docker pull centos
```

**DESCRIPTION**

The official build of CentOS.

# Docker - les images

```
$ docker search centos
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED		
centos	The official build of CentOS.	3351	[OK]			
jdeathe/centos-ssh	CentOS-6 6.8 x86_64 / CentOS-7 7.3.1611 x8...	69		[OK]		
nimmis/java-centos	This is docker images of CentOS 7 with dif...	26		[OK]		
gluster/gluster-centos	Official GlusterFS Image [ CentOS-7 + Glu...	19		[OK]		
kinogmt/centos-ssh	CentOS with SSH	13		[OK]		
torusware/speedus-centos	Always updated official CentOS docker imag...	8		[OK]		
nathonfowlie/centos-jre	Latest CentOS image with the JRE pre-installed	5		[OK]		
centos/mariadb55-centos7		5		[OK]		
harisekhon/centos-java	Java on CentOS (OpenJDK, tags jre/jdk7-8)	2		[OK]		
harisekhon/centos-scala	Scala + CentOS (OpenJDK tags 2.10-jre7 - 2...	2		[OK]		
[...]						

# Docker - les images

```
$ docker pull centos
```

```
Using default tag: latest
latest: Pulling from library/centos
343b09361036: Pull complete
Digest: sha256:bba1de7c9d900a898e3cadbae040dfe8a633c06bc104a0df76ae24483e03c077
Status: Downloaded newer image for centos:latest
```

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
centos	latest	8140d0c64310	12 days ago	193 MB
pico	latest	dced8bcb6d5d	2 weeks ago	274 MB
<none>	<none>	3d5e22b4b76e	2 weeks ago	274 MB
<none>	<none>	067ed313cc11	2 weeks ago	274 MB
<none>	<none>	591089767816	2 weeks ago	273 MB
<none>	<none>	32d23c8b5e6e	2 weeks ago	119 MB
<none>	<none>	5e09cdbf1c31	2 weeks ago	69.6 MB
php	7.1.3-fpm-alpine	e97e3208d019	2 months ago	69.6 MB

# Docker - Les containers

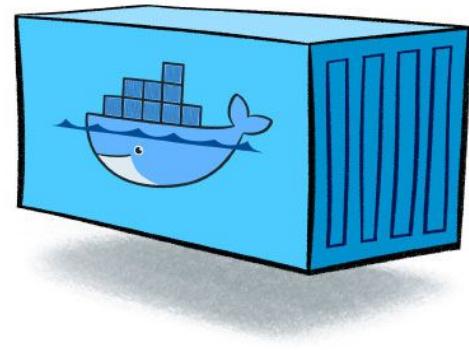
1 container = 1 processus

Mode interactif

```
$ docker run -it centos /bin/bash
```

```
[root@af40a9d95e82 /]#
```

- -t        Allocate a pseudo-tty
- -i        Interactive mode (STDIN)



# Docker - Les containers

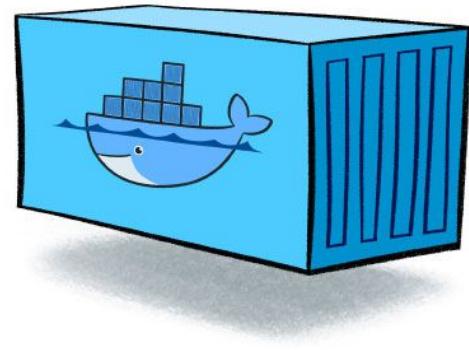
## Mode background

```
$ docker run -d --name hello centos /bin/bash -c "while  
sleep 1; done"  
ce7383fd92b9e15c9c96fa58bb99fe62fba4dee50c566abd5807804e
```

- -d        Daemon mode (Background)
- --name    nom du container

```
$ docker logs -f hello
```

```
hello  
hello  
hello  
hello  
hello
```

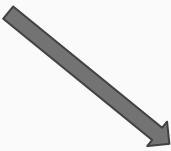


# Docker - Les containers

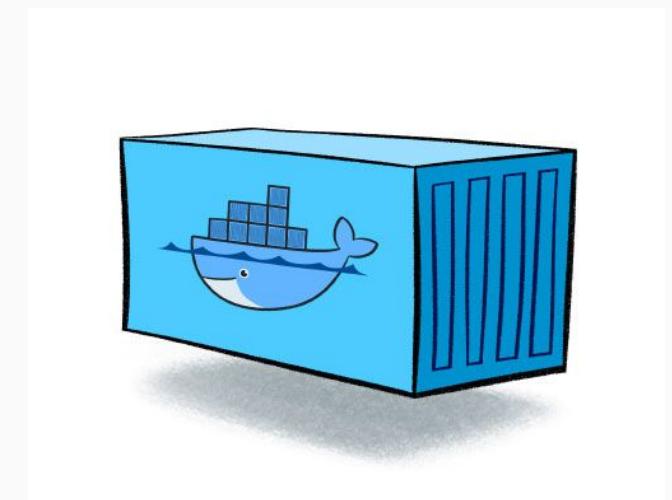
## Lister les containers

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6669ee75652a	centos	/bin/bash -c 'whi...'	4 seconds ago	Up 2 seconds		hello



```
$ docker stop 6669ee75652a  
$ docker start 6669ee75652a  
$ docker restart 6669ee75652a  
$ docker kill 6669ee75652a  
$ docker rm 6669ee75652a
```



# Docker - Créer une image

- Depuis une image existante, après avoir exécuté des commandes

```
$ docker commit -m "Say Hello" -a "Kate Smith" 0b2616b0e5a8 user/centos:v2  
4f177bd27a9ff0f6dc2a830403925b5360bfe0b93d476f7fc3231110e7f71b1c
```

- Depuis un Dockerfile, basé sur une image existante

```
$ docker build -t user/centos:v2 .
```

- From scratch

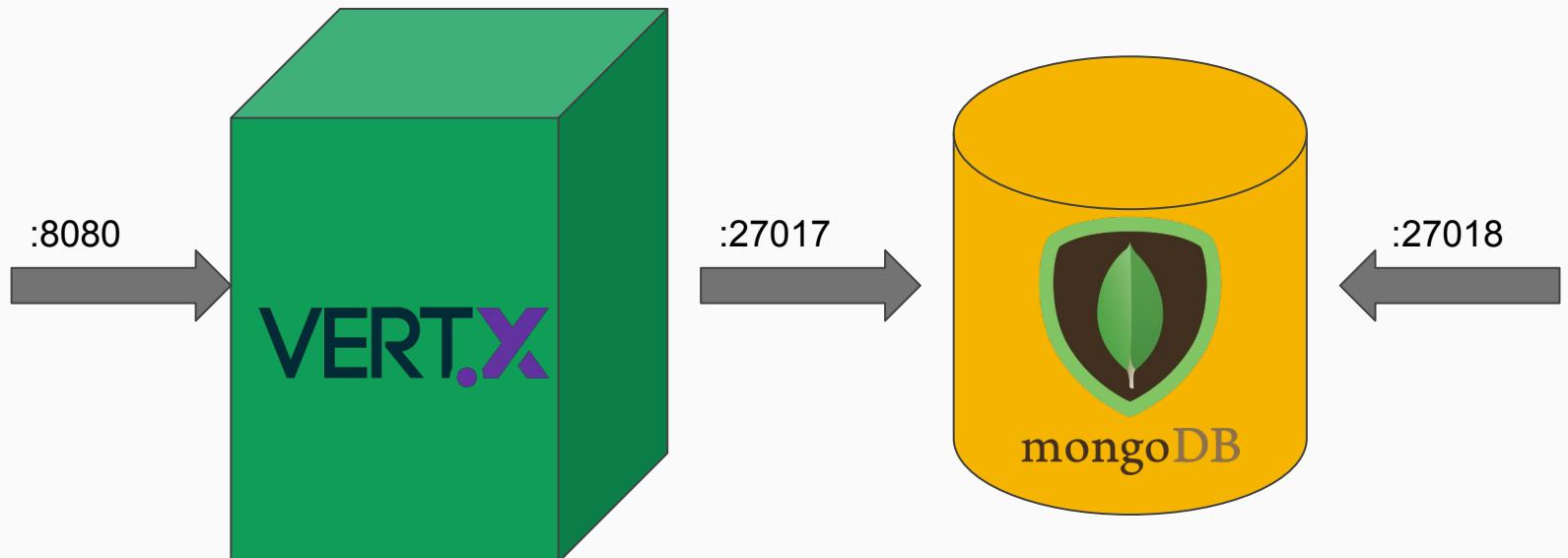
```
$ cat helloFromScratch.tgz | docker import - hello:new
```

# Dockerfile

```
FROM BaseImage:version
MAINTAINER Xavier MARIN "marin.xavier@gmail.com"
ENV myvar="value"
RUN yum install pkgs1 , pkgs2 -y
RUN mkdir -p /workdir/log
COPY ./conf/application.conf
ADD www.archive.com/myarchive.tar.gz /workdir/
EXPOSE 9000
VOLUME ["/workdir/log"]
WORKDIR ["/workdir"]
USER myuser
ENTRY-POINT ["/bin/bash"]
CMD ["echo" , "hello world"]

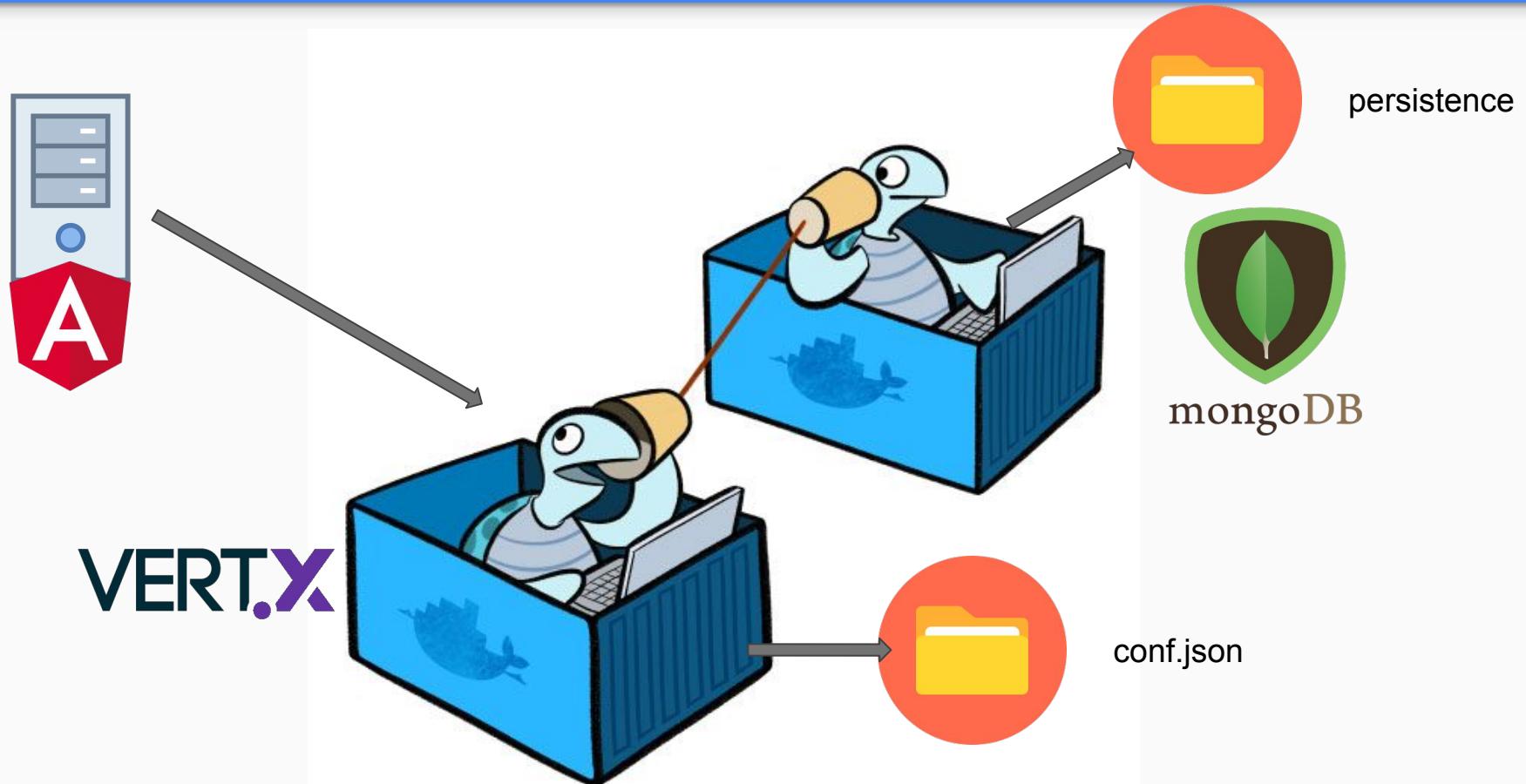
$ docker build -t hello:0.2.5 .
```

# Docker - cas pratique



```
$ export ENV=PROD  
$ java -jar monServeur.jar
```

## Docker - cas pratique



```
$ docker run -d -p 27018:27017 -v /home/moi/data:/var/lib/data --name mongo mongodb
$ docker run -d \
  -p 8080:8080 \
  -e ENV=PROD \
  -v /home/moi/conf:/opt/conf \
  --link mongo
serveur:0.0.25
```

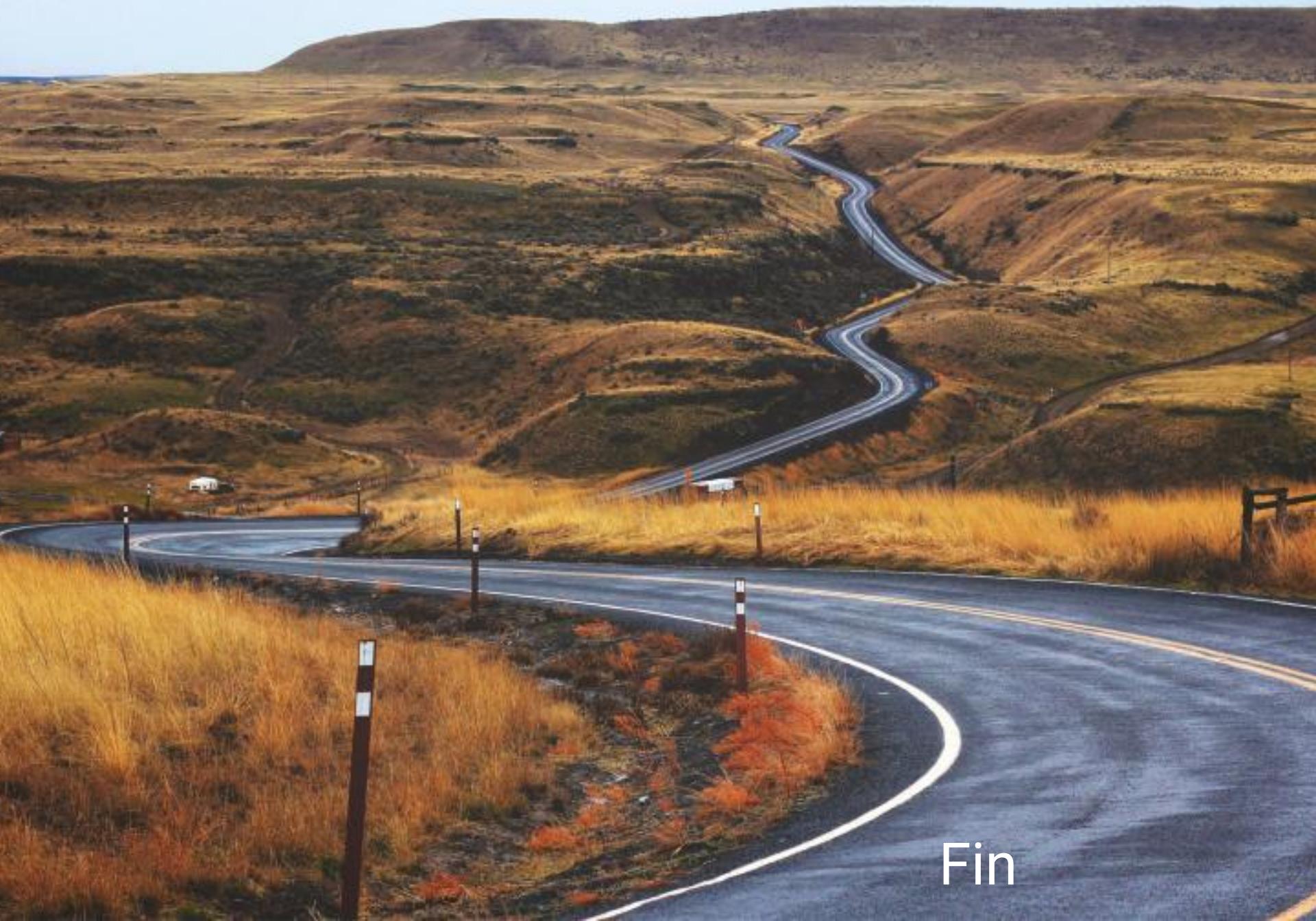
# Docker compose

## docker-compose.yml

```
version: '2'  
services:  
  mongodb:  
    image: mongodb  
    volumes:  
      - /home/moi/data:/var/lib/data  
    ports:  
      - 27018:27018  
  serveur:  
    image: serveur:0.0.25  
    environment:  
      ENV: PROD  
    volumes:  
      - /home/moi/conf:/opt/conf  
    ports:  
      - 8080:8080
```

\$ docker-compose up -d





Fin