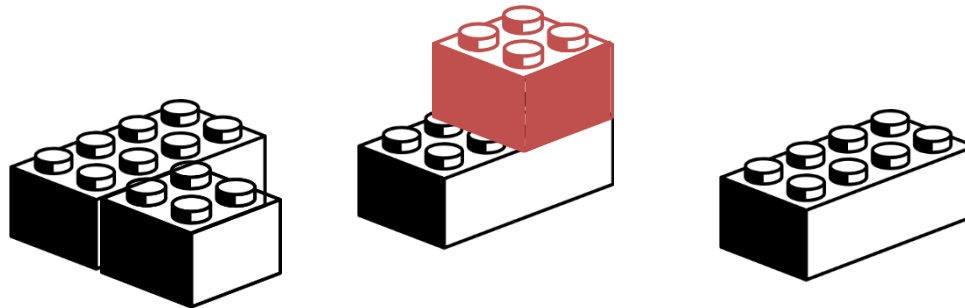


# Collaboration et Robustesse

Mise en œuvre par le langage Python

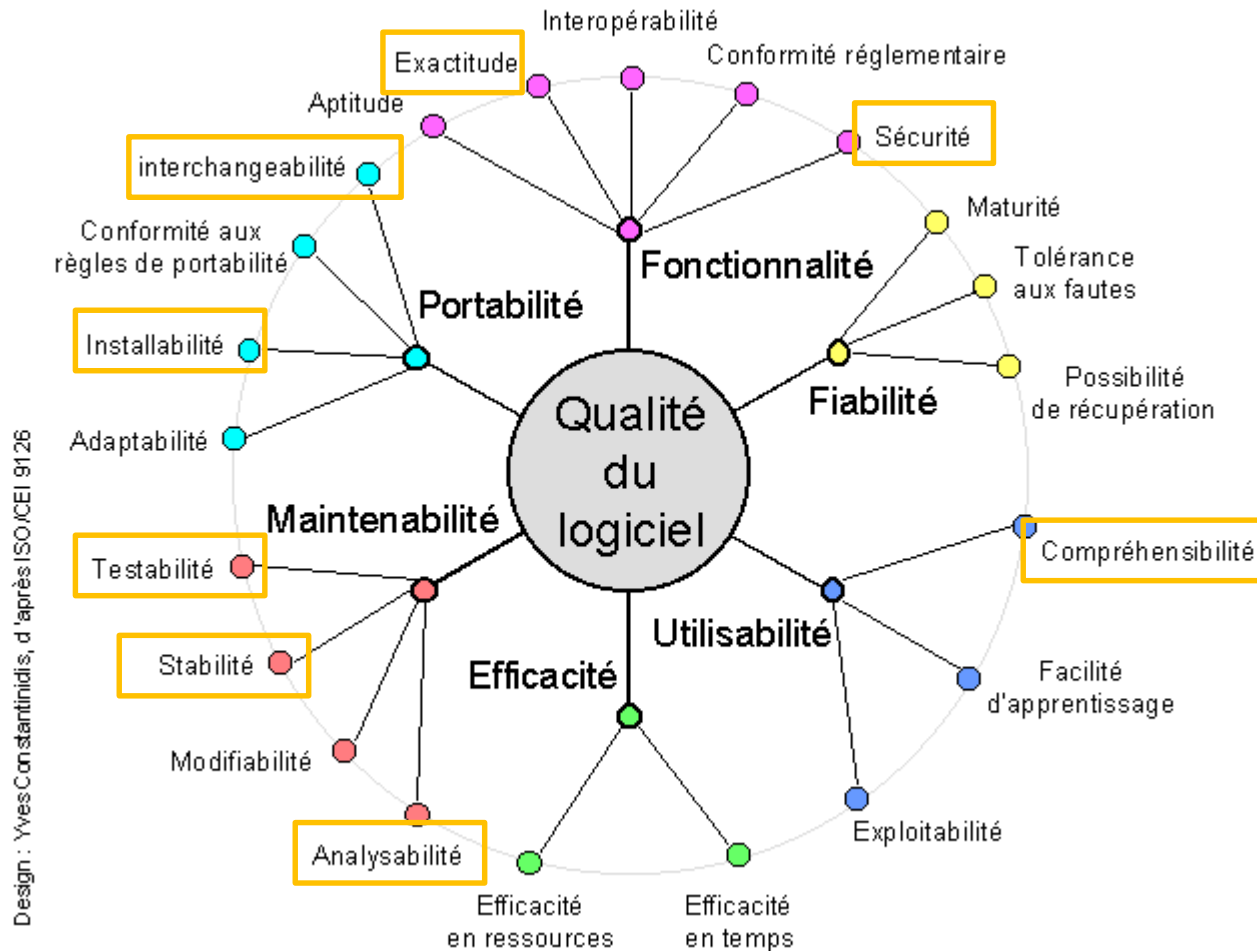
J. Saraydaryan





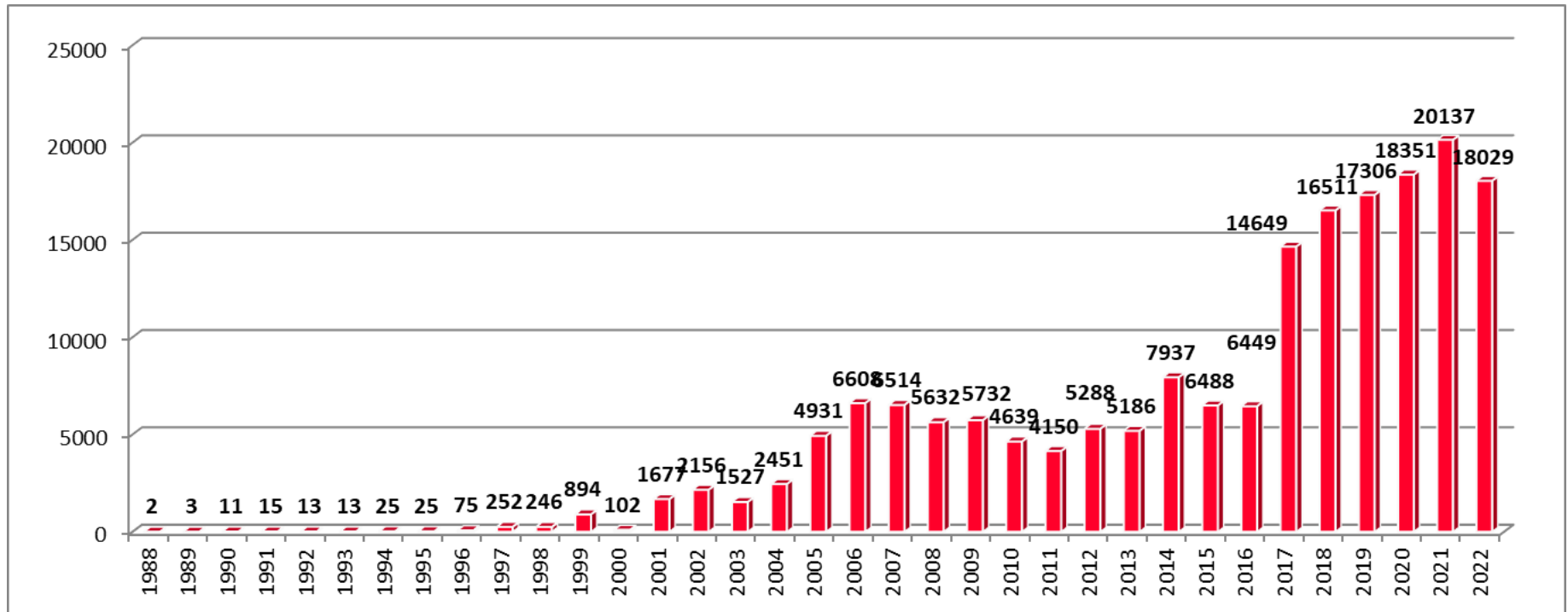
# Motivation

# Qualité d'un logiciel



# Qualité d'un logiciel

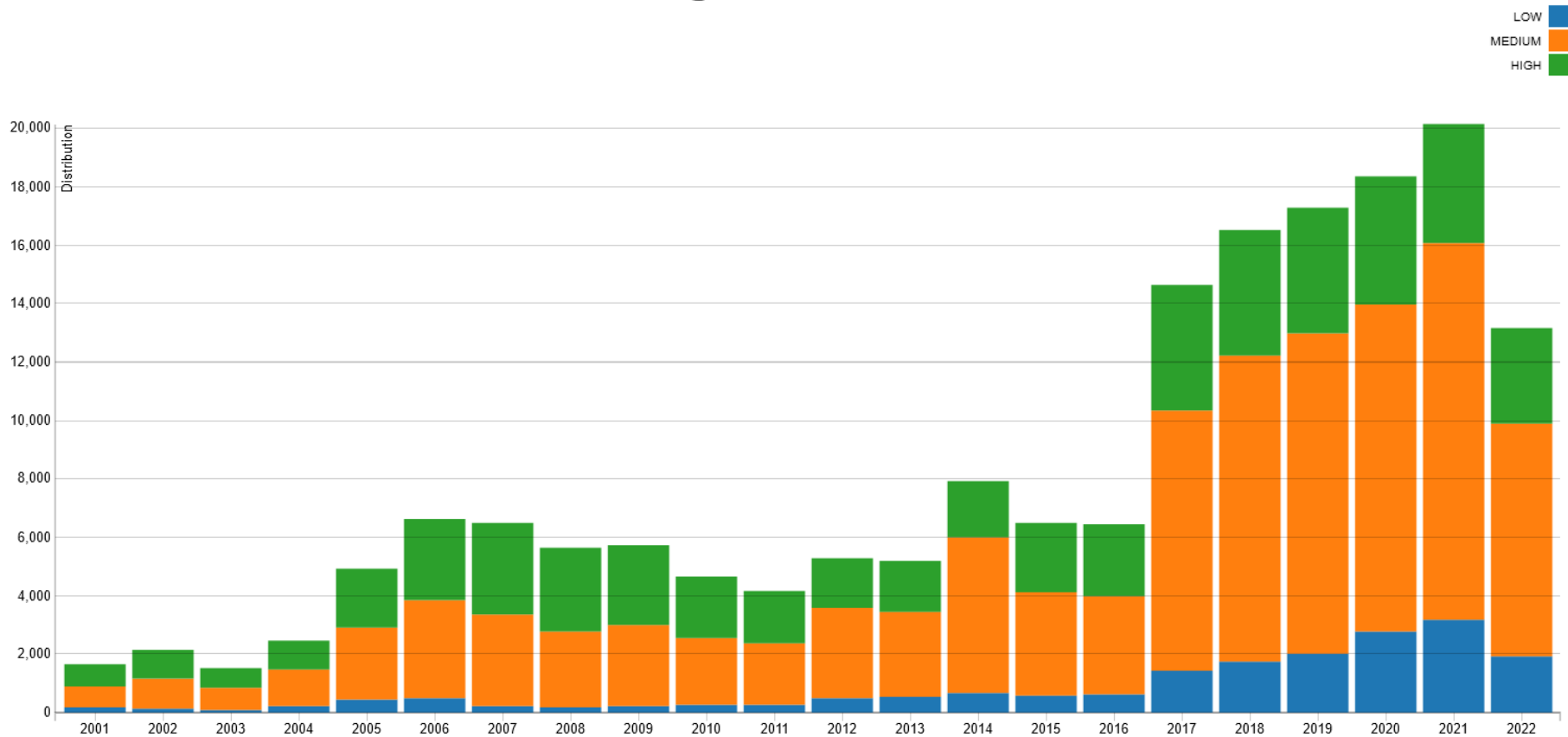
Evolution du nombre de vulnérabilités



<http://web.nvd.nist.gov>



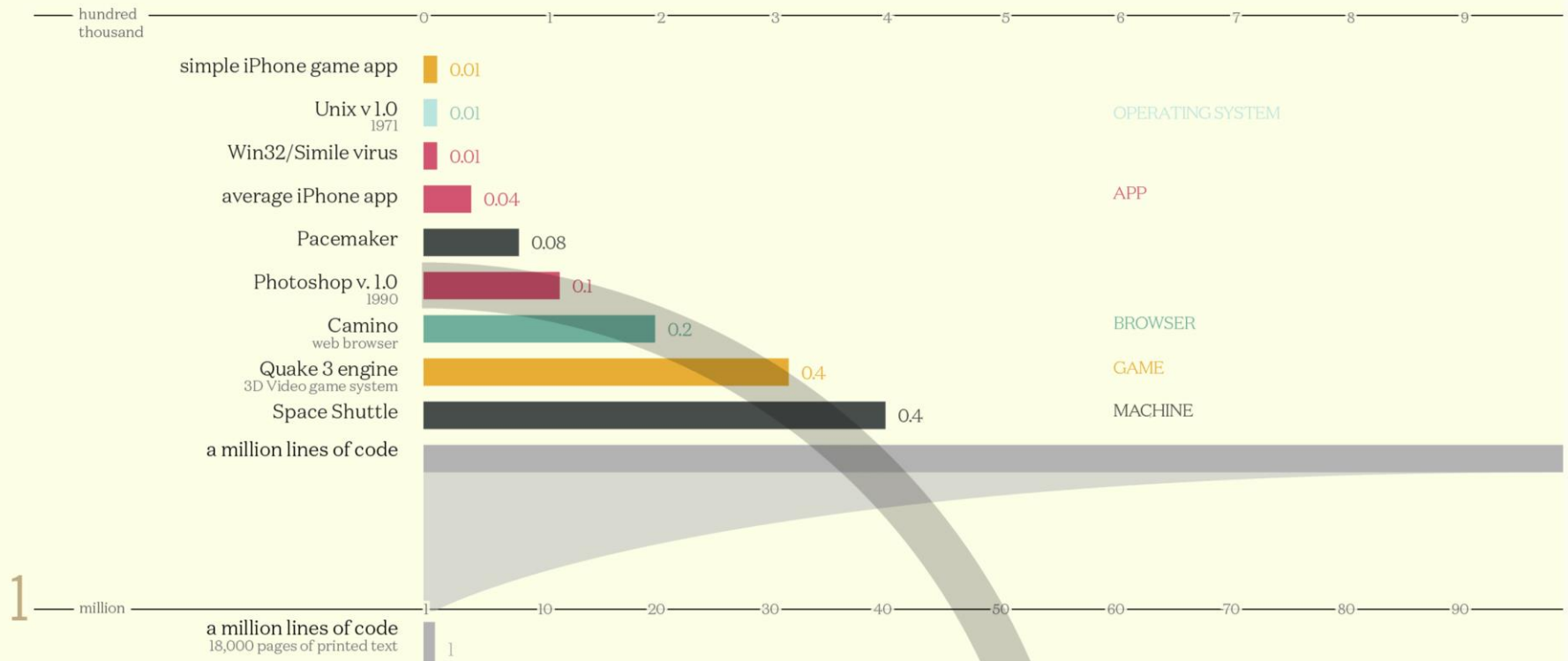
# Qualité d'un logiciel



<https://nvd.nist.gov/vuln-metrics/visualizations/cvss-severity-distribution-over-time>

## Codebases

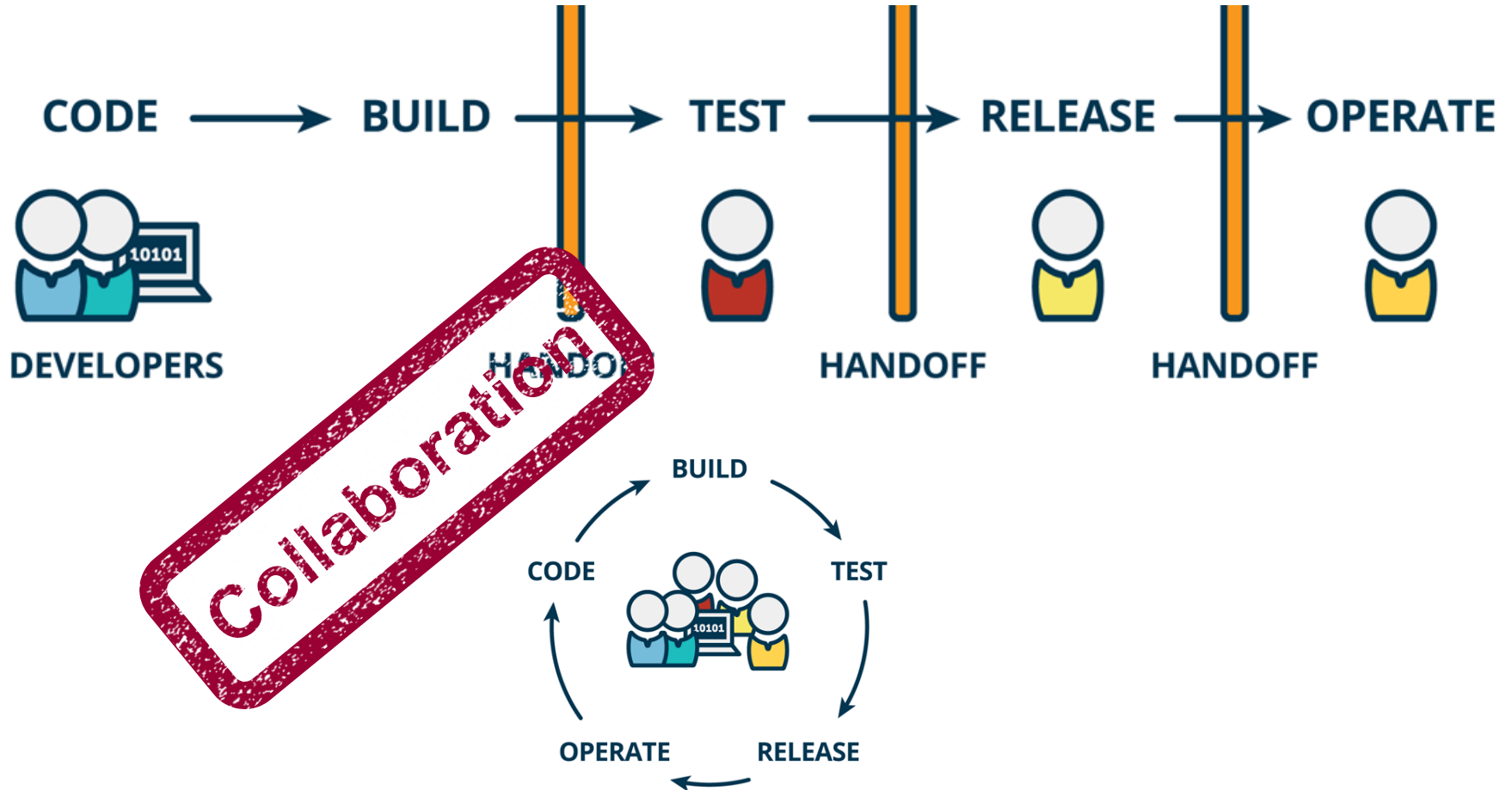
Millions of lines of code



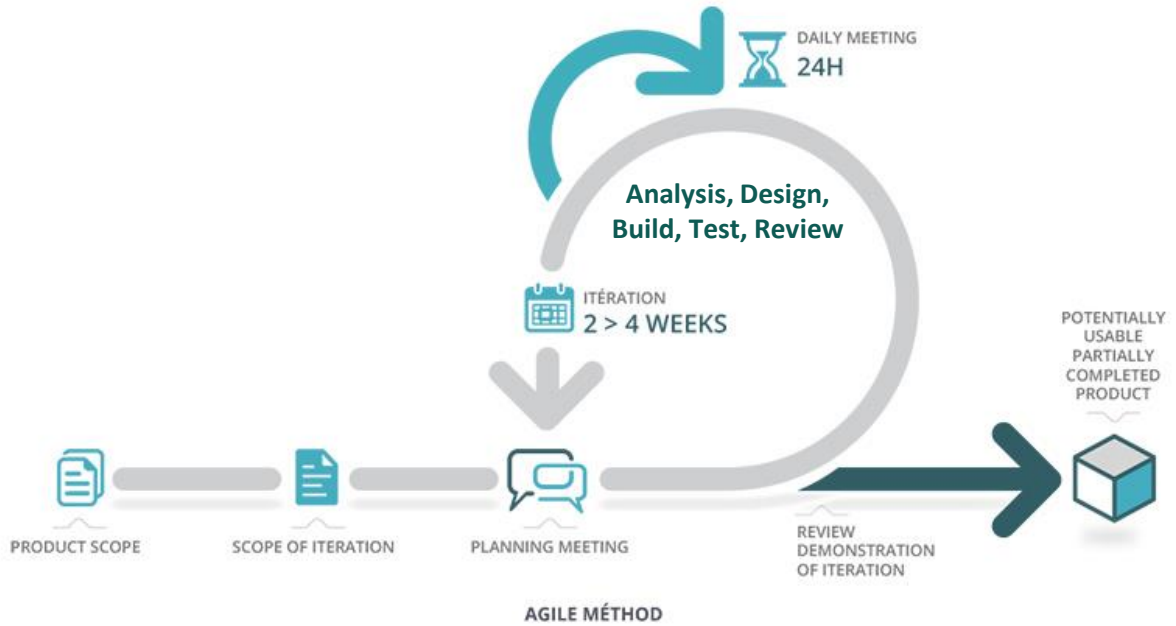
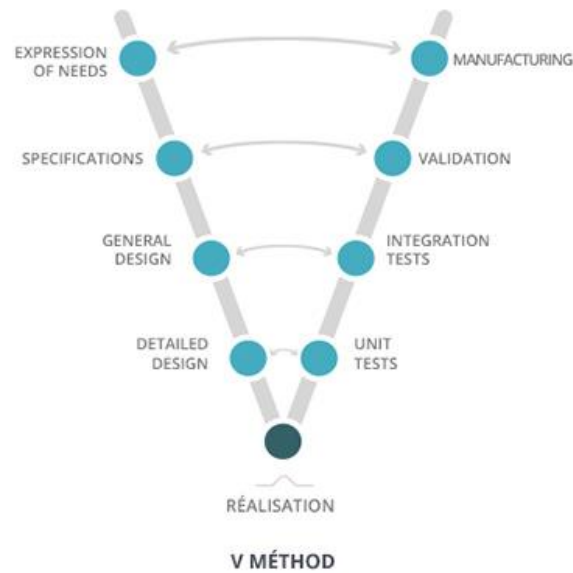
<https://informationisbeautiful.net/visualizations/million-lines-of-code/>



# Cycle de production



# Cycle de production

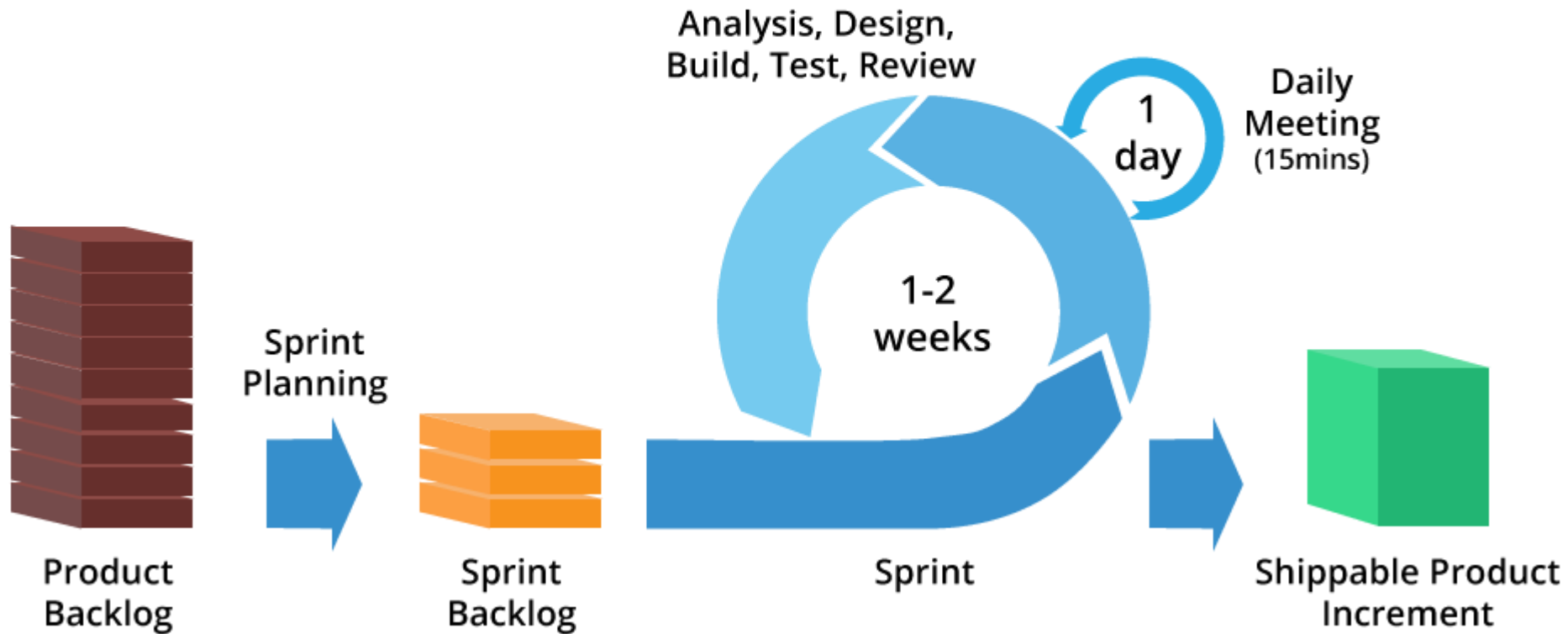


<https://www.cegedim-insurance.com/en-EN/solutions-services/our-added-value/Pages/agile-methodology.aspx>



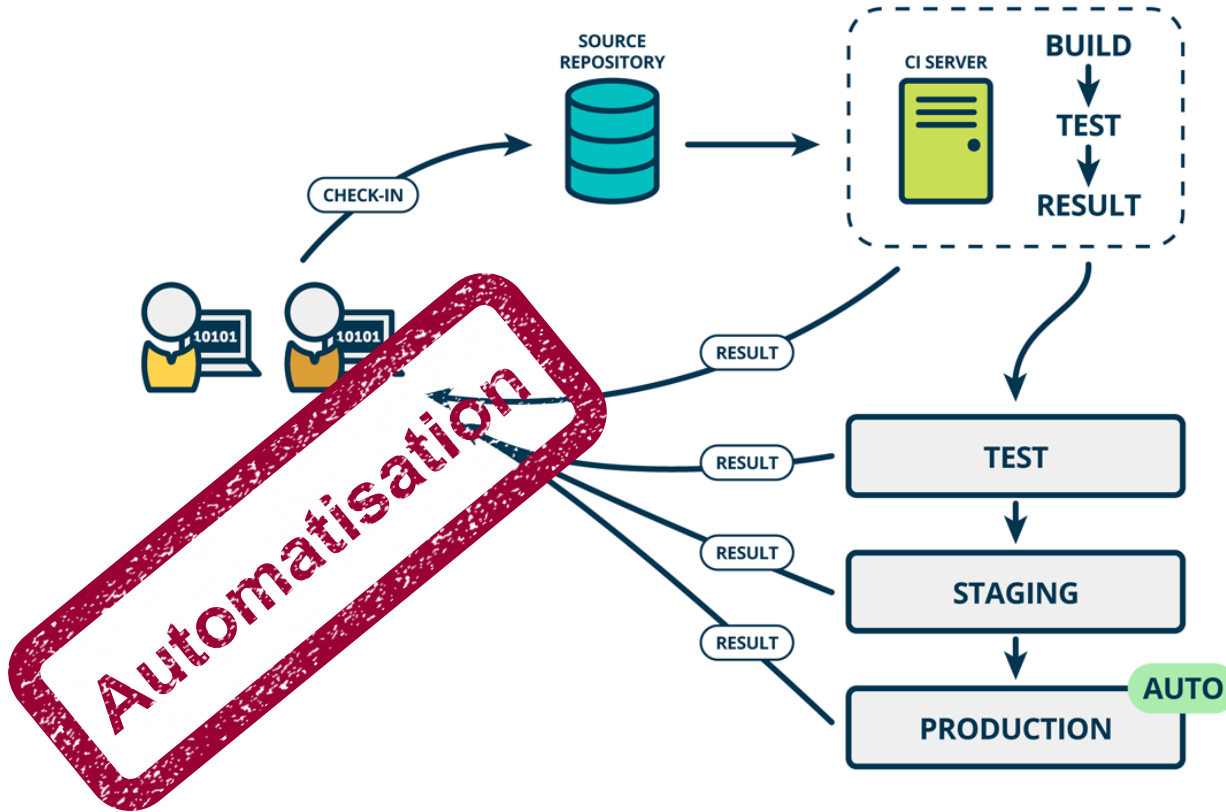
# Cycle de production

## Agile Software Development



<https://www.codingmart.com/blog/what-is-agile-what-is-scrum>

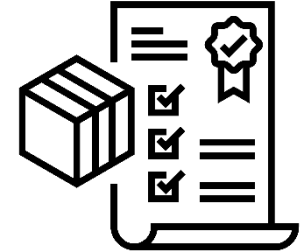
# Cycle de production



# Besoin

## ☐ Qualité logicielle

- Test
- Fiabilité
- Sécurité



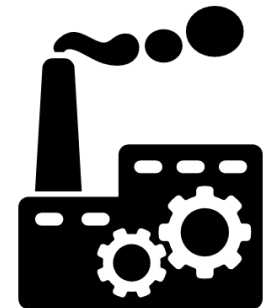
## ☐ Collaboration

- Travail en équipe
- Compréhension
- Partage, gestion de version



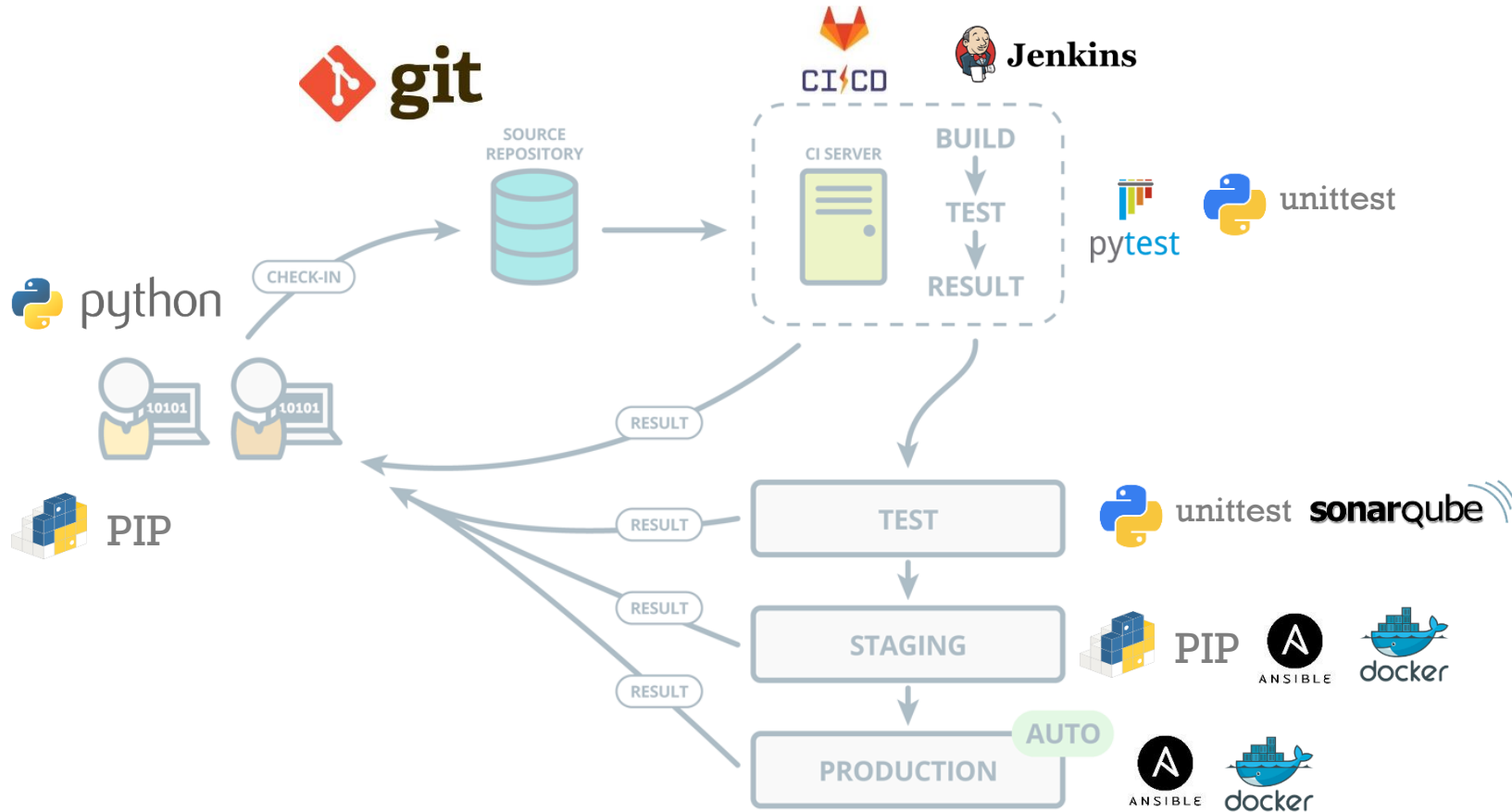
## ☐ Automatisation

- Test, build
- Mise en production
- Communication de l'état du logiciel





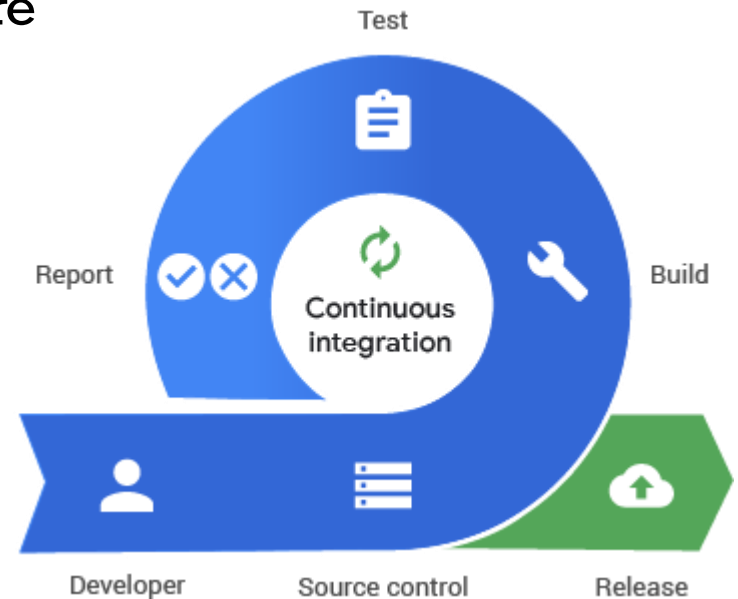
# Cycle de production



# Intégration Continue



Prendre en compte des **changements** apportés à un code informatique d'un logiciel en **garantissant** la **non régression** de l'application. Ce mécanisme est **continue** et permet la **génération** du code final (**binaire**) de l'application à chaque modification de cette dernière



<https://cloud.google.com/solutions/continuous-integration/>

# Intégration Continue

Outil de tests  
Automatique

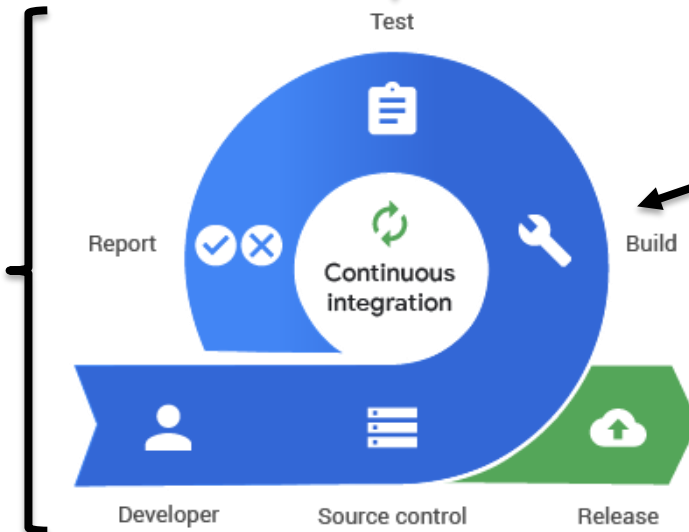


Test

Environnement  
de Build



Framework de  
coordination  
des actions



<https://cloud.google.com/solutions/continuous-integration/>

Logiciel de versioning



# Questions ?