

Visulog

Team Dino

Before you read this document	1
Goals	1
Workflow	2
Tools	3
Planning	3
Links & Resources	3
Code Culture	4

Before you read this document

- This document is only used for organizational purposes.
- Nothing here is concrete or final, there are propositions and we can change them anytime, so feel free to share your feedback and suggestions.
- Please, use the comment or suggestion tool of google docs to propose, comment and others.
- This is a "dynamic" file, i.e. it will change frequently so use the shared link to consult the file.
- I've used google translate to write in French...Hope is understandable.

Paris Mollo

Goals

1. **(Final Goal)** Développer un outil d'analyse et de visualisation des logs git.
Visulog est un outil d'analyse des contributions des membres d'une équipe

travaillant sur un même projet donné hébergé sur un référentiel git. Son objectif est d'aider les enseignants à noter individuellement les élèves travaillant en équipe.

2. (Project Features)

- a. Calculer quelques indicateurs pertinents tels que :
 - i. nombre de lignes ou de caractères ajoutés/supprimés/modifiés
 - ii. nombre de commits
 - iii. nombre de commits de fusion
- b. Analyser les variations de ces indicateurs dans le temps : par exemple sommer puis en une semaine, calculer une moyenne journalière ou une moyenne dans une fenêtre glissante.
- c. Visualiser les indicateurs sous forme de graphiques (histogrammes, camemberts, etc.) intégrés dans une page Web générée.

3. (Project Documentation)

- a. Requirements – Statements that identify attributes, capabilities, characteristics, or qualities of a system. This is the foundation for what will be or has been implemented.
- b. Architecture/Design – Overview of software. Includes relations to an environment and construction principles to be used in design of software components.
- c. Technical – Documentation of code, algorithms, interfaces, and APIs.
- d. End user – Manuals for the end-user, system administrators and support staff.

Workflow

1. Nous allons utiliser le processus Scrum, pour en savoir plus sur le processus, vous pouvez voir ce lien ici: [Scrum : de quoi s'agit-il, comment cela fonctionne-t-il et en quoi est-ce génial ? \(atlassian.com\)](https://atlassian.com/fr/scrum)
2. Pour chaque **feature**, nous utiliserons la méthodologie Agile.
 - a. requirements → design → develop → test → deploy → review
3. Nous pouvons faire des "sprints" de 7 jours. Créer le backlog représentant toutes les features (ensemble de story dédiées à une feature) puis s'assurer

d'un découpage des story suffisamment petit/fin pour que chaque story puisse être faite dans un sprint.

4. Pratiques communes :

Pour éviter les problèmes d'intégration et l'apparition de bugs nous choisissons une approche de développement basé sur les tests au préalable:

- Nous définissons une story
- La façon de tester cette story

Les deux seront commités et pushés.

5. Phases du projet

- a. Constitution du backlog du produit visulog.
- b. Development : réalisation des storys depuis leur mise en oeuvre jusqu'au test et leur intégration sur le master
- c. Post development : post sprint : demo et rétrospective (présentation des storys et amélioration de la méthode agile et réunion)

6. Travail en paire et travail de toute l'équipe

- a. Au niveau paire, on livre des story et on les intègre avec le reste.
- b. Au niveau de toute l'équipe, sprint planning (exemple du tableau excel), les démo, les retro.

Tools

1. Gitlab for code issues and code storage
2. Google Sheets for non-code tasks and planning
3. Google Docs for documentation and information
4. Discord and Whatsapp for communication
5. Nous aurons un dossier sur google drive avec tous les documents du projet.

Planning

1. Chaque semaine, le planning sera partagé avec l'équipe, ainsi que les tâches de sprint et les informations respectives sur les tâches.

Links & Resources

- [Gitlab repo](#)
- [Drive](#)
- [Discord](#)
- <https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1443814-gitlab-definition-api-pricing-ci-cd/>

Code Culture

1. Modular Code !
2. Small commits !
3. Comment your code !
4. Test before commit!

Paris Mollo
parismolloch@gmail.com