

Organisation du travail

Objectif : Créer une application permettant de générer des fichiers wave contenant des signaux simples (sinus, cosinus, carré, triangle).

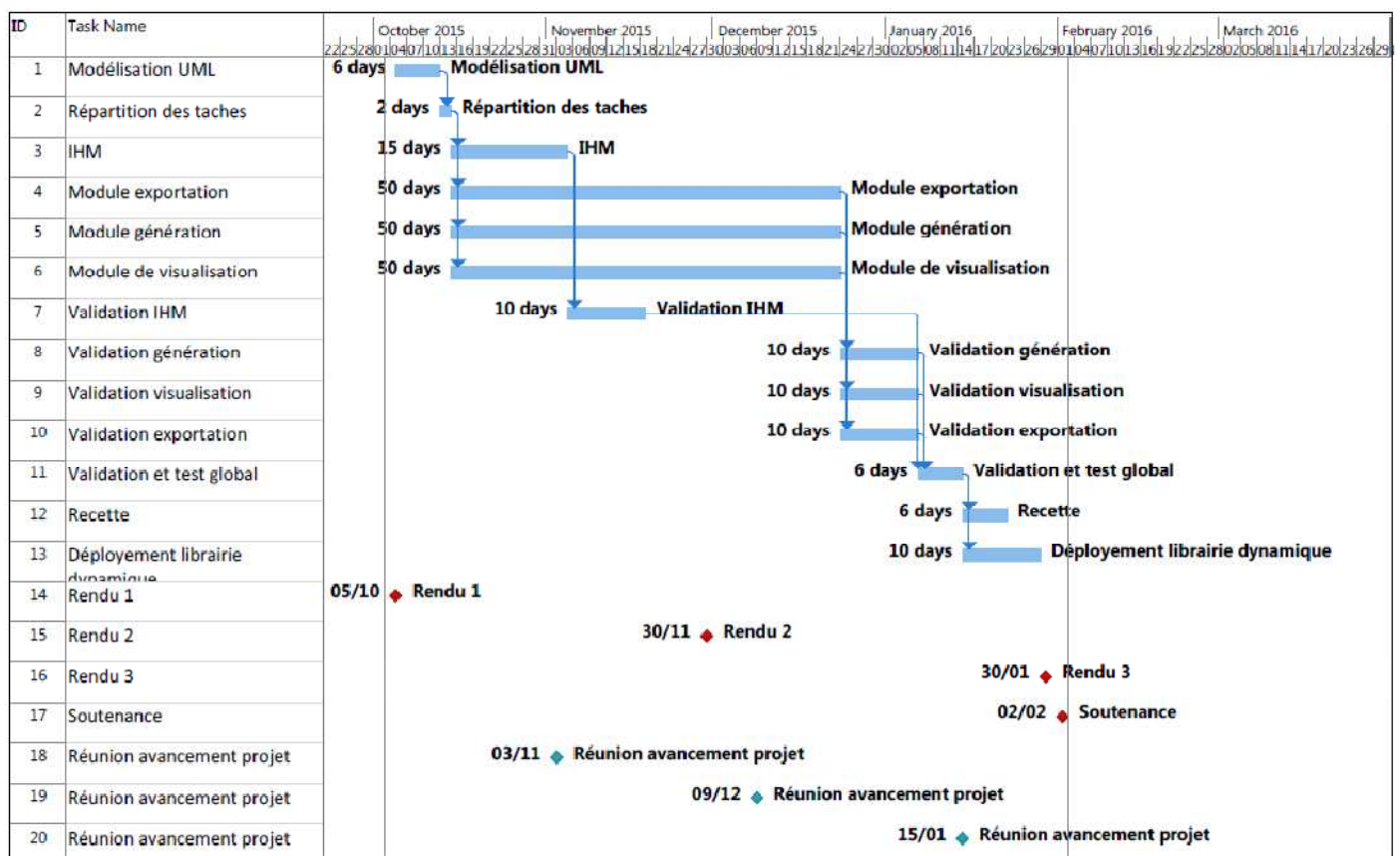
➤ Rappel des objectifs

Une interface graphique intuitive devra être créée pour régler les paramètres des signaux :

- Amplitude (en pourcent de la résolution max)
- Offset (en pourcent de la résolution max)
- Rapport cyclique (%)
- Fréquence (Hz)

Le programme pourra être compilé sous forme de librairie dynamique afin de permettre la réutilisation des fonctions par d'autres utilisateurs.

➤ Diagramme de Gantt



➤ Répartition des tâches

Tâche	RESULTAT
Modélisation UML	Étudiant A, B, C, D
IHM	Étudiant B
Module d'exportation au format WAVE	Étudiant C
Module de génération de signaux	Étudiant D
Module de visualisation	Étudiant A
Validation IHM	Étudiant A, B, C, D
Validation génération	Étudiant A, B, C, D
Validation visualisation	Étudiant A, B, C, D
Validation exportation	Étudiant A, B, C, D
Validation et test global	Étudiant A, B, C, D
Recette	Étudiant A, B, C, D
Déploiement librairie dynamique	Étudiant D, B, A, C

➤ Système de travail collaboratif et méthode de validation interne des contributions

Il est nécessaire de mettre en place un gestionnaire de version afin de pallier à l'éloignement géographique des différents membres constitutif de l'équipe projet. Nous avons donc mis en place un dépôt Git, afin de fournir un espace de travail collaboratif et évolutif.

Le choix de Git s'est fait par sa simplicité d'utilisation et son interface conviviale via TortoiseGit. Chaque membre dispose d'un accès au serveur de gestion de versions sur Git Hub et peut ainsi apporter sa contribution au projet.

Git permet la fusion automatique des modifications effectués par les différents membres de l'équipe, et enregistre également le nom de la personne ayant apporté les modifications. Ceci permettra de valider les différentes contributions lors des réunions d'avancement du projet.

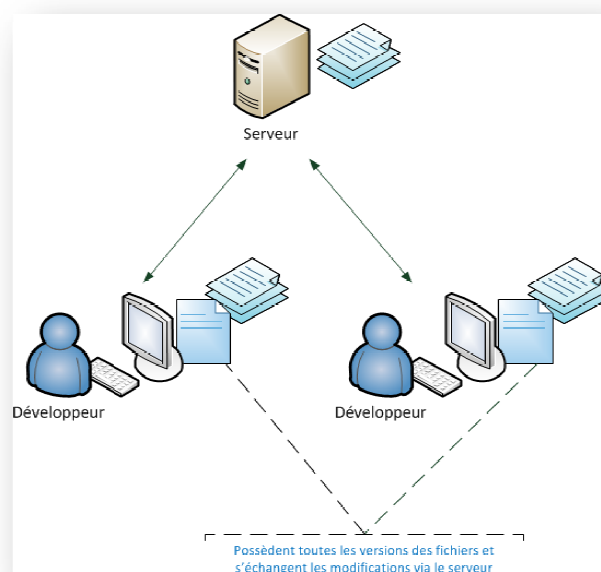


Figure 1 - Fonctionnement de Git