UNIVERSITÉ DU QUÉBEC À MONTRÉAL

TP2

Énoncé de planification initiale

PRÉSENTÉ À L’UNIVERSITÉ DU QUÉBEC À MONTRÉAL

COMME EXIGENCE DU COURS   
  
INF6150.030 – Génie logiciel III : Conduite de projets informatiques (Hiver 2024)

Chargé de cours : Ammar Hamad

PAR L’ÉQUIPE #06:

(LES TOURISTES)

Front-End - KEVIN DA SILVA - ( DASK30049905 ),

Back-End - THAI MINH DANNY NGUYEN - ( NGUT71060209 ),

Back -End - SAAFI ACHREF - (SAAA87070201),

Back -End - BENHAMAD ALI - (BENA21119407),

Back -End - JEAN-CHRISTOPHE CLOUÂTRE - (CLOJ73020201)

Front-End - EMMANUEL EXCELLENT - (EXCE23099402)

Département d'informatique

04 Mars 2024

# TP2 - Énoncé de planification initiale (20%)

La planification d'un projet de développement logiciel s'effectue à 3 niveaux de détails distincts (stratégique – tactique – opérationnel) et pour lesquels il existe au moins un livrable.

## Livrables et leurs éléments demandés

### Charte de projet (niveau Stratégique)

#### Mission :

Notre équipe s'est donné pour mission de développer une application destinée à inspirer et à motiver les athlètes de tous niveaux. Nous souhaitons offrir un outil précis et stimulant pour suivre leurs progrès en course à pied et à vélo. En facilitant l'accès à des données claires et en proposant des programmes d'entraînement personnalisés, notre objectif est de transformer l'expérience d'entraînement de nos utilisateurs. Notre contribution unique réside dans notre capacité à fournir des entraînements adaptés à chaque individu, que ce soit des séances personnalisées ou des conseils d'experts, afin d'améliorer à la fois leurs performances sportives et leur bien-être. Nous croyons que notre application joue un rôle essentiel dans le succès global de l'organisation, en répondant aux besoins des athlètes et en les aidant à atteindre leurs objectifs. Sans notre équipe, l'organisation risque de perdre un outil précieux pour motiver et soutenir ses clients dans leur pratique sportive.

#### Client :

Notre équipe cible les athlètes amateurs et semi-professionnels passionnés de course à pied et de vélo, désireux de bénéficier d'un suivi précis et personnalisé de leurs performances sportives.

#### Objectifs SMART :

1. Spécifique : Notre objectif est de lancer une version bêta fonctionnelle de l'application, intégrant des fonctionnalités clés pour la course à pied et le vélo, d'ici la fin de la session.

2. Mesurable : Nous visons à obtenir un minimum de 100 utilisateurs actifs et à recueillir leurs retours sur les fonctionnalités de l'application pour évaluer notre succès.

3. Atteignable : Nous concentrerons nos efforts sur le développement des fonctionnalités essentielles telles que le suivi GPS, l'analyse des performances et une interface utilisateur intuitive, car elles sont réalisables avec le niveau d'expérience de notre équipe et le temps disponible.

4. Réaliste : Notre objectif est réaliste compte tenu des ressources disponibles et du calendrier de la session. Nous nous concentrerons sur la création d'une expérience utilisateur fluide et des fonctionnalités de suivi précises pour garantir le succès de notre projet.

5. Temporel : Nous nous engageons à atteindre ces objectifs avant la fin de la session d'hiver, en nous assurant de respecter les délais fixés.

Outils utilisés pour évaluer les progrès : Chaque membre de l'équipe se fixera des objectifs spécifiques à accomplir avant chaque réunion d'équipe, qui se tiennent une à deux fois par semaine. Ces objectifs serviront de points de référence pour mesurer notre avancement et identifier les éventuels obstacles à surmonter.

#### Portée du Projet :

Inclus : Le projet comprend actuellement le développement du suivi GPS, de l'analyse des performances et de l'interface utilisateur.

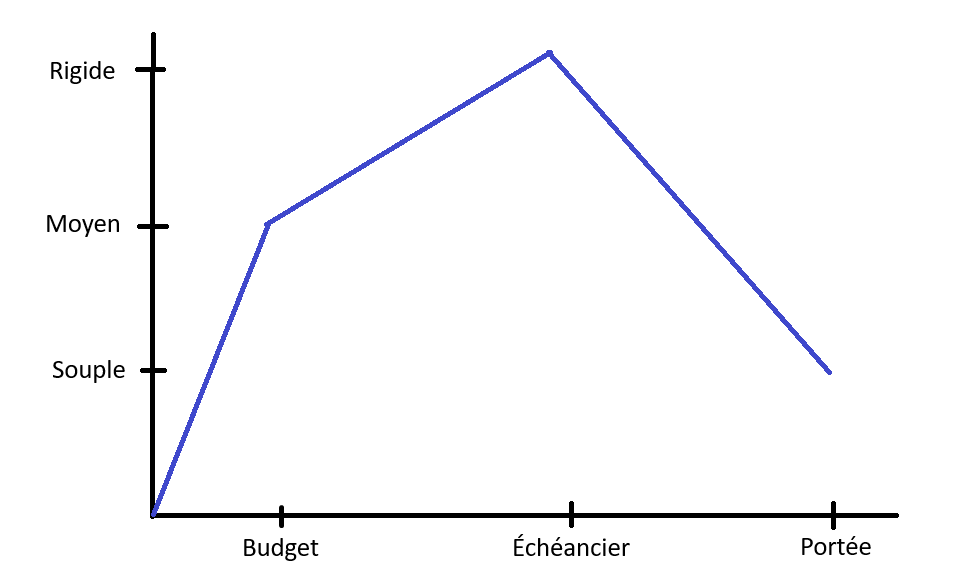
Exclus : L'équipe a délibérément exclu les fonctionnalités de réseautage social ainsi que l'intégration avec des équipements de fitness tiers.

#### Matrice des leviers:

Budget: Si on y va d’un point de vue hypothétique, le budget alloué pour la conception du projet Budget : Nous avons attribué un niveau de flexibilité moyen au budget, ce qui signifie qu'il existe une marge de manœuvre raisonnable pour ajuster les dépenses en fonction des besoins du projet, tout en respectant les contraintes financières globales.

Échéancier : Pour l'échéancier du projet, nous avons opté pour une flexibilité rigide. Cette décision repose sur le fait que nous avons une date limite stricte à respecter, correspondant à la fin de la session, ce qui rend difficile toute modification significative de la planification temporelle.

Portée : En ce qui concerne la portée du projet, nous avons décidé d'adopter une approche souple. Cela signifie que la portée du projet peut être ajustée et évoluer au fil du temps en fonction des commentaires des utilisateurs et des besoins du marché. Cette flexibilité nous permettra de répondre de manière agile aux changements et aux nouvelles exigences qui pourraient surgir au cours du développement.

Graphe de la matrice des leviers

#### Résumé des coûts anticipés :

En se basant sur un total estimé de 600 heures, comprenant 100 heures par personne, et en utilisant un tarif de 120 $/heure, le coût total anticipé s'élève à 72 000 $. Ces coûts sont exclusivement basés sur l'effort estimé en heures-personne, conformément à la méthodologie de gestion de projet en génie logiciel. Aucun autre coût, tel que des licences logicielles ou des services informatiques, n'est prévu dans ce contexte.

#### Critères de succès :

Notre équipe évalue le succès du projet à travers plusieurs critères clés définis en consultation avec le client. Ceux-ci incluent le nombre d'utilisateurs actifs de l'application, mesuré régulièrement pour suivre l'engagement des utilisateurs. En outre, nous utiliserons des sondages de satisfaction utilisateur pour recueillir des retours directs sur l'expérience utilisateur et identifier les domaines à améliorer. Enfin, nous considérons également l'amélioration mesurable des performances des utilisateurs comme un indicateur crucial de succès, en surveillant les progrès et les réalisations des utilisateurs dans leurs activités sportives grâce à notre application.

#### Liste des **risques :**

##### Risque : Perte de membres clés de l'équipe en raison de congés prolongés ou de départs inattendus.

###### Plan de mitigation (M) : Mettre en place une documentation approfondie et des processus de partage des connaissances pour assurer la continuité du travail en cas d'absence d'un membre clé.

###### Plan de contingence (C) : Prévoir des ressources supplémentaires ou des remplacements temporaires pour combler les lacunes laissées par les membres absents et maintenir la productivité de l'équipe.

##### Risque : Changements fréquents des exigences client.

###### Plan de mitigation (M) : Établir des canaux de communication clairs avec le client pour clarifier les exigences dès le départ et encourager une communication ouverte tout au long du projet.

###### Plan de contingence (C) : Mettre en place des procédures flexibles pour gérer les changements de portée, en priorisant les fonctionnalités essentielles et en planifiant des itérations de développement itératives.

##### Risque : Manque de participation aux tests unitaires.

###### Plan de mitigation (M) : Mettre en place une politique de développement axée sur les tests unitaires dès le début du projet, en mettant l'accent sur l'importance de l'écriture et de l'exécution de tests unitaires pour chaque fonctionnalité développée.

###### Plan de contingence (C) : Former l'équipe sur les bonnes pratiques de test unitaire et sur l'importance de leur participation, en fournissant des ressources et un soutien supplémentaire si nécessaire pour garantir que les tests unitaires sont effectués de manière exhaustive.

##### Risque : Dépassement des délais, ne pas atteindre les objectifs minimaux avant la fin de la session.

###### Plan de mitigation (M) : Effectuer des suivis réguliers du progrès de chaque membre de l'équipe pour identifier et résoudre les problèmes potentiels à un stade précoce.

###### Plan de contingence (C) : Réajuster le niveau de la portée du projet pour réduire le temps nécessaire à sa conception et respecter les délais impartis.

##### Risque : Conflits internes au sein de l'équipe ou désaccords sur la direction du projet.

###### Plan de mitigation (M) : Favoriser une communication ouverte et transparente au sein de l'équipe, en encourageant la résolution proactive des conflits et en facilitant la collaboration.

###### Plan de contingence (C) : Impliquer un médiateur externe ou un tiers neutre pour aider à résoudre les conflits et à rétablir un environnement de travail harmonieux si nécessaire.

### Charte d'équipe (niveau Tactique)

Nom de l'équipe : LES TOURISTES  
liste des membres : KEVIN DA SILVA, THAI MINH DANNY NGUYEN, SAAFI ACHREF, BENHAMAD ALI, JEAN-CHRISTOPHE CLOUÂTRE & EMMANUEL EXCELLENT

Référence : <https://gitlab.info.uqam.ca/inm5151-h24/touristes-de-salon-expires/-/milestones>

#### Sous-ensemble des fonctionnalités :

##### Gestion d'activité

###### Créer une activité

###### Modifier une activité

###### Supprimer une activité

###### Enregistrer les activités par saisie manuelle

###### Enregistrer les activités par fichier GPX

###### Géolocaliser l'emplacement actuel du client

#### Contraintes

L'équipe se retrouve confrontée à plusieurs limites imposées, sur lesquelles elle n'a aucun contrôle direct, mais qui encadrent son champ d'action. Parmi celles-ci, la connexion et la gestion des comptes clients constituent un défi majeur. En effet, l'équipe doit opérer dans le cadre de ces connexions, en s'adaptant aux différents profils d'utilisateurs et en assurant la sécurité et la confidentialité des données.

Un autre aspect crucial est le journal et l'analyse des activités. L'équipe doit travailler à l'intérieur des données fournies par ces journaux, ce qui peut parfois limiter leur marge de manœuvre. De plus, la dépendance à savoir quel utilisateur est connecté afin d'afficher le contenu du client spécifique représente une contrainte supplémentaire. Cette dépendance restreint les possibilités de personnalisation et de flexibilité dans la présentation des informations.

Par ailleurs, chaque requête de fusion (merge) doit être validée par l'équipe de DevOps avant d'être envoyée au "Main". Cette étape de validation constitue une limite importante, car elle ajoute un processus supplémentaire qui peut ralentir le développement et introduire des retards dans le déploiement des fonctionnalités.

Enfin, le serveur sur lequel est situé le projet appartient à la maintenance de l'UQAM. Cette dépendance envers un autre service implique que l'équipe n'a pas le contrôle total sur l'infrastructure sur laquelle elle opère. Cela peut entraîner des limitations en termes de personnalisation, de configuration et de gestion des ressources.

Les limites imposées à l'équipe représentent des défis significatifs, mais la collaboration avec d'autres équipes et le développement de stratégies d'adaptation permettent de les surmonter tout en maintenant un travail de qualité.

#### Responsabilités déléguées à l’équipe

L'équipe se voit confier un ensemble de responsabilités cruciales liées à la gestion des activités et à l'interaction directe avec les clients. Ces responsabilités comprennent la gestion d'activités, qui englobe la création, la modification et la suppression d'activités, ainsi que l'enregistrement des activités par saisie manuelle ou par fichier GPX. De plus, l'équipe est chargée de géolocaliser l'emplacement actuel du client, ce qui est essentiel pour la planification et la prestation efficaces des services. Ces tâches nécessitent une compréhension approfondie des besoins des clients et une capacité à travailler de manière autonome pour répondre à ces besoins de manière proactive.

En revanche, certaines responsabilités demeurent la responsabilité des autres équipes ou fonctions de l'organisation. La connexion et la gestion des comptes, notamment des comptes clients et des comptes entraîneurs, ainsi que les actions telles que la connexion/déconnexion et la gestion des profils utilisateur et client, relèvent de ces autres équipes. De même, la gestion du journal et des analyses, y compris la liste des activités avec filtrage et tri, la planification des activités, les notifications, ainsi que les statistiques et graphiques par type et par unité de temps, sont des responsabilités qui restent entre les mains de ces autres équipes ou de leurs gestionnaires. Ces responsabilités sont cruciales pour assurer la sécurité des données, la cohérence des informations et la fourniture de rapports et d'analyses approfondis pour guider les décisions stratégiques de l'organisation.

Ainsi, la répartition des responsabilités entre l'équipe et les autres entités de l'organisation permet une collaboration efficace et une répartition équilibrée des tâches, garantissant que chaque partie contribue de manière significative à la réalisation des objectifs globaux de l'organisation tout en exploitant ses domaines d'expertise spécifiques.

#### Rôles et responsabilités

Pour une compréhension plus détaillée des responsabilités et des rôles spécifiques de chaque membre de l'équipe, je vous invite à consulter le tableau récapitulatif qui présente une vue d'ensemble claire et structurée. Ce tableau met en lumière les tâches assignées à chaque membre de l'équipe, ainsi que les responsabilités qui leur sont déléguées dans le cadre de leurs fonctions respectives. De plus, en cas d'indisponibilité d'un membre de l'équipe due à une maladie ou une urgence particulière, un autre individu désigné par le symbole "-" assumera les responsabilités de manière temporaire, agissant en tant qu'assistant pour garantir la continuité des opérations et le bon fonctionnement des activités. Cette flexibilité permet à l'équipe de faire face aux imprévus tout en maintenant un niveau élevé de performance et de collaboration.

#### Tableau des rôles et responsabilités :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nom\Responsabilité | Coordonnateur | Développeur | Responsable assurance qualité | Responsable gestion de configuration | Point de contact technique |
| **EMMANUEL EXCELLENT** |  | Coche avec un remplissage uni |  |  |  |
| **JEAN-CHRISTOPHE CLOUÂTRE** |  | Coche avec un remplissage uni |  |  | Coche avec un remplissage uni |
| **BENHAMAD ALI** | Coche avec un remplissage uni | Coche avec un remplissage uni |  |  |  |
| **SAAFI ACHREF** |  | Coche avec un remplissage uni |  |  |  |
| **THAI MINH DANNY NGUYEN** |  | Coche avec un remplissage uni | Coche avec un remplissage uni |  |  |
| **KEVIN DA SILVA** |  | Coche avec un remplissage uni |  | Coche avec un remplissage uni |  |

#### Définition des rôles :

**Coordonnateur –** Assure la coordination efficace des activités de l'équipe, facilitant la communication et l'organisation des tâches pour atteindre les objectifs fixés.

**Développeur –** Conçoit et crée des solutions logicielles, en utilisant des langages de programmation et des technologies appropriés, pour répondre aux besoins spécifiques du projet.

**Responsable assurance qualité –** Garantit la conformité aux normes de qualité établies en supervisant le processus de développement, en définissant les critères de "terminé" et en veillant à ce que ces critères soient respectés à chaque étape du projet.

**Responsable gestion de configuration -** Assure l'intégrité et la traçabilité des artefacts du projet en mettant en place et en maintenant des processus de gestion de configuration efficaces, garantissant ainsi la cohérence et la fiabilité du développement logiciel.

**Point de contact technique –** Assure la liaison avec d'autres équipes dépendantes en fournissant un support technique, en résolvant les problèmes potentiels résultant de cette interdépendance, et en facilitant la collaboration entre les différents acteurs du projet.

#### Normes de fonctionnement de l’équipe

Processus de développement

Tableau Kanban verticale : Ce tableau mettra en évidence les différentes activités à réaliser, en cohérence avec la définition de "terminé" de manière généraliser.

|  |  |  |
| --- | --- | --- |
| Phase du Processus de Développement | Activités | Justification |
| Analyse des Besoins | Identification des exigences. | Assurer une compréhension claire des besoins et des attentes des utilisateurs. |
|  | Élaboration des spécifications fonctionnelles. | Définir les fonctionnalités attendues et les critères de succès pour le produit. |
|  | Revue et validation des spécifications. | S'assurer que les spécifications sont complètes, claires et alignées avec les attentes du client. |
| Développement | Écriture du code | Implémenter les fonctionnalités en suivant les meilleures pratiques de codage et en respectant les normes établies. |
|  | Tests unitaires | Valider le bon fonctionnement des différentes parties du code de manière isolée. |
|  | Intégration du code | Tester l'interaction entre les différentes parties du système pour garantir leur compatibilité. |
|  | Tests d'intégration | S'assurer que les différentes fonctionnalités du système fonctionnent correctement ensemble. |
|  | Revue par les Pairs | Examen du code par d'autres membres de l'équipe pour identifier les erreurs, les lacunes et garantir la qualité du code. |
| Validation | Tests fonctionnels | Vérifier que le système répond aux exigences fonctionnelles spécifiées et aux besoins des utilisateurs. |
|  | Revue par les Parties Prenantes | Présentation du produit aux parties prenantes pour obtenir leurs retours et leur validation. |
|  | Corrections et améliorations | Effectuer les ajustements nécessaires en fonction des retours reçus et des tests effectués. |
| Livraison | Finalisation et préparation du déploiement | Préparer le produit pour son déploiement en production, en effectuant les derniers tests et vérifications nécessaires. |
|  | Approbation finale | Obtention de l'approbation finale de toutes les parties prenantes avant le déploiement en production. |
|  | Déploiement en production | Lancement du produit en production, prêt à être utilisé par les utilisateurs finaux. |

#### Listes des outils et référence:

##### **Visual studio code**: <https://code.visualstudio.com/>

###### *Usages* : Un environnement de développement intégré (IDE).

##### **IntelliJ**: <https://www.jetbrains.com/idea/>

###### *Usages* : Un environnement de développement intégré (IDE).

##### **Npm**: <https://www.npmjs.com/>

###### *Usages* : Un gestionnaire de paquets JavaScript largement utilisé pour l'installation et la gestion des dépendances de projet.

##### **Svelte**: <https://svelte.dev/>

###### *Usages* : Un framework JavaScript moderne pour la construction d'interfaces utilisateur réactives et performantes, en générant un code optimisé qui s'exécute efficacement dans le navigateur.

##### **Svelte Kit**: <https://kit.svelte.dev/>

###### *Usages* : Un framework basé sur Svelte, facilitant la création d'applications web avec un routage intégré, un rendu côté serveur et des fonctionnalités de construction optimisées.

##### **Eslint**: <https://eslint.org/>

###### *Usages* : Un outil d'analyse statique du code JavaScript qui identifie les erreurs de syntaxe, les problèmes de style et les pratiques non recommandées, aidant ainsi à maintenir la cohérence et la qualité du code.

##### **Mysql**: <https://www.mysql.com/>

###### *Usages* : Un système de gestion de base de données relationnelle open source, largement utilisé pour le stockage et la manipulation de données structurées.

##### **NodeJs** : <https://nodejs.org/en>

###### *Usages* : Un environnement d'exécution JavaScript côté serveur, permettant le développement d'applications web évolutives et rapides.

##### **Powershell**: <https://learn.microsoft.com/en-us/powershell/>

###### *Usages* : Un shell de ligne de commande et un langage de script développé par Microsoft, offrant des fonctionnalités avancées pour l'automatisation des tâches système sous Windows.

##### **Bash**: <https://www.gnu.org/software/bash/>

###### *Usages* : Un shell Unix standard, largement utilisé sur les systèmes d'exploitation Linux et macOS, offrant un environnement de ligne de commande puissant et flexible.

##### **Docker**: <https://www.docker.com/>

###### *Usages* : Une plateforme de conteneurisation qui simplifie le déploiement et la gestion d'applications dans des environnements isolés et portables, facilitant ainsi le développement et la livraison logicielle.

##### **API Google Maps**: <https://github.com/googlemaps>

###### *Usages* : Une interface de programmation permettant d'intégrer des fonctionnalités de cartographie et de géolocalisation dans les applications web et mobiles.

##### **Format Json**: <https://www.w3schools.com/js/js_json_intro.asp>

###### *Usages* : Un format de données léger et lisible par l'homme, largement utilisé pour l'échange de données structurées entre les applications.

#### Logistique de coordination

Nous nous réunissons chaque lundi matin en personne pendant 15 minutes, une fois par semaine. De plus, nous avons une réunion toutes les deux semaines, d'une durée approximative d'une heure, via Discord.

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Lieu/Outils | Heure | Durée |
| 17-Janvier-2024 | Discord (Appel) | 9h00 am | 1 heure |
| 04-Février-2024 | Discord (Appel) | 10h00 am | 45 minutes |
| 17-Février-2024 | Discord (Appel) | 9h00 am | 2 heures |
| 02-Mars-2024 | Discord (Appel) | 9h00 am | 1 heure 40 minutes |

#### Modes décisionnels principal et secondaire

Les modes possibles sont :

* Unanimité (nous sommes tous d’accord),
* Consensus (une majorité est d’accord et les autres n’en sont pas dérangés),
* Majorité (une majorité est d’accord mais une minorité sont contre),
* Autocratie par le coordonnateur (il n’y a que lui/elle qui prend la décision et le reste de l’équipe accepte par défaut)

Les décisions finales sont prises de manière démocratique, où la majorité l'emporte, mais précédées d'un débat approfondi avant le vote définitif.

Lors de la première réunion, nous avons créé l'équipe, réparti les rôles et les responsabilités, et mis en place le serveur. Nous avons également rédigé le CONOPS (Concept of Operations), et discuté et choisi le modèle de développement agile. [[Lien vers le PDF de la réunion](https://uqam-my.sharepoint.com/personal/fh691920_ens_uqam_ca/Documents/hiver%202024/inf6150/devoir/tp2/RemiseTP2/1-ProcesVerbaux.pdf) 1-ProcesVerbaux.pdf]

Lors de la deuxième réunion, nous avons structuré les tâches en issues, établi le backlog, et discuté du langage de programmation à utiliser. Nous avons également affiné le CONOPS, débattu de la syntaxe de programmation, et décidé du langage et des outils de développement à adopter. [[Lien vers le PDF de la réunion](https://uqam-my.sharepoint.com/personal/fh691920_ens_uqam_ca/Documents/hiver%202024/inf6150/devoir/tp2/RemiseTP2/2-ProcesVerbaux.pdf) 2-ProcesVerbaux.pdf]

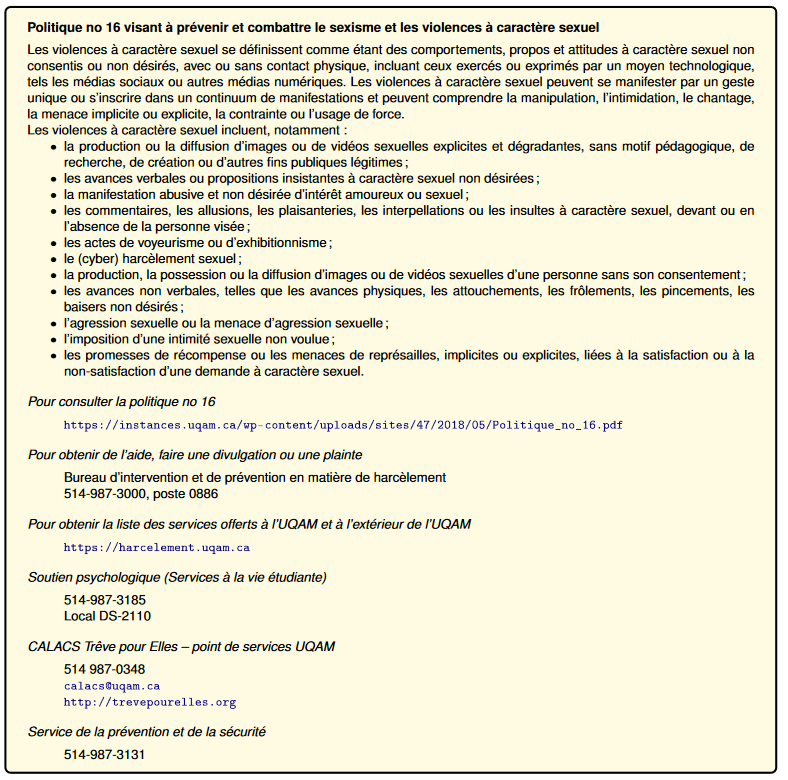
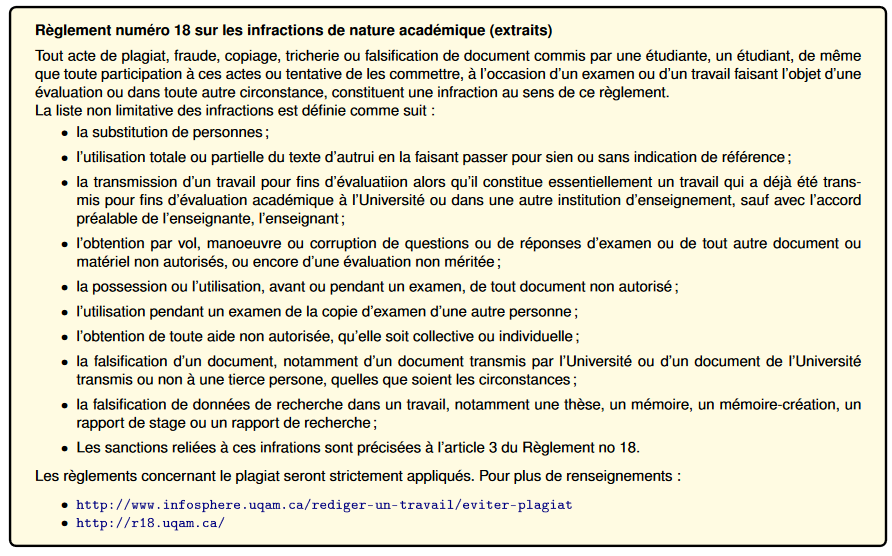
La troisième réunion a été consacrée à la compréhension et à la structuration des branches, à la réalisation des tâches demandées, ainsi qu'à la familiarisation avec les responsables des merge requests et à la résolution des erreurs de programmation. [[Lien vers le PDF de la réunion](https://uqam-my.sharepoint.com/personal/fh691920_ens_uqam_ca/Documents/hiver%202024/inf6150/devoir/tp2/RemiseTP2/3-ProcesVerbaux.pdf) 3-ProcesVerbaux.pdf]

Lors de la quatrième réunion, nous avons élaboré la planification initiale, défini les livrables, rédigé la charte de projet et la charte d'équipe, et clarifié les rôles et responsabilités de chacun. Nous avons également défini ce qu'une tâche considérée comme terminée impliquait, dressé la liste des activités, et établi un gabarit Excel pour estimer et suivre les activités. [[Lien vers le PDF de la réunion](https://uqam-my.sharepoint.com/personal/fh691920_ens_uqam_ca/Documents/hiver%202024/inf6150/devoir/tp2/RemiseTP2/4-ProcesVerbaux.pdf) 4-ProcesVerbaux.pdf]

#### Comportements attendus et Comportements non tolérés

Politique d’absence due à une maladie ou une urgence particulière :

En cas d'absence d'un membre de l'équipe en raison d'une maladie ou d'une urgence spécifique, il est impératif que l'individu informe immédiatement son assistant pour assurer une continuité efficace du travail et maintenir une productivité optimale. L'assistant prendra alors temporairement en charge les responsabilités de l'absent.



#### Résolution de conflits

##### **Non-livraison des engagements d'un membre de l'équipe** :

###### *Communication ouverte* : Engager une discussion avec le membre concerné pour comprendre les raisons derrière le retard ou le non-respect des engagements.

###### *Réaffectation des tâches* : Si possible, réaffecter les tâches pour alléger la charge de travail ou identifier des ressources supplémentaires pour aider le membre en difficulté.

###### *Suivi régulier* : Mettre en place des points de contrôle réguliers pour suivre la progression des travaux et identifier les problèmes potentiels à un stade précoce.

##### **Conflit d'interprétation des rôles et responsabilités** :

###### *Clarification des attentes* : Organiser une réunion d'équipe pour clarifier les rôles et responsabilités de chacun, en mettant l'accent sur les domaines d'overlap ou de confusion.

###### *Établissement de règles de communication* : Mettre en place des directives claires sur la manière dont les membres de l'équipe doivent communiquer leurs attentes et leurs besoins, afin d'éviter toute ambiguïté.

###### *Médiation* : Si nécessaire, faire appel à un médiateur neutre pour faciliter la discussion et trouver un terrain d'entente entre les parties en conflit.

##### **Divergences dans la prise de décision** :

###### *Brainstorming* : Organiser une session de brainstorming pour explorer différentes options et idées, en encourageant la participation de tous les membres de l'équipe.

###### *Vote démocratique* : En cas d'impasse, organiser un vote démocratique pour prendre une décision basée sur la majorité, en veillant à ce que toutes les voix soient entendues et respectées.

###### *Compromis* : Encourager les membres de l'équipe à rechercher des compromis et des solutions alternatives qui répondent aux besoins de chacun tout en permettant d'avancer.

##### **Communication inefficace au sein de l'équipe** :

###### *Établissement de canaux de communication clairs* : Définir des canaux de communication efficaces pour différents types de messages (e-mails, réunions en personne, plateformes de collaboration en ligne, etc.).

###### *Formation en communication* : Organiser des sessions de formation ou des ateliers sur la communication efficace pour aider les membres de l'équipe à développer leurs compétences en communication et à améliorer leur compréhension mutuelle.

###### *Feedback régulier* : Encourager les membres de l'équipe à donner et à recevoir un feedback constructif sur leur communication, en identifiant les points forts et les domaines à améliorer.

##### **Conflits de personnalité au sein de l'équipe** :

###### *Médiation individuelle* : Organiser des entretiens individuels avec les membres en conflit pour comprendre leurs points de vue et leurs préoccupations.

###### *Team building* : Organiser des activités de renforcement d'équipe pour favoriser la collaboration et renforcer les liens entre les membres de l'équipe.

###### *Établissement de normes de comportement* : Définir des normes de comportement claires et acceptées par tous pour favoriser un environnement de travail respectueux et harmonieux.

##### **Difficultés liées à la répartition des ressources ou du temps** :

###### *Planification stratégique* : Revoir et ajuster la planification des ressources et du calendrier pour mieux répondre aux besoins et aux priorités de l'équipe.

###### *Priorisation des tâches* : Identifier les tâches critiques et prioritaires, et allouer les ressources en conséquence pour garantir que les objectifs principaux soient atteints en premier.

###### *Communication transparente* : Communiquer ouvertement sur les contraintes de ressources et de temps, et impliquer les membres de l'équipe dans le processus de prise de décision pour trouver des solutions viables.

### Définition de « terminé » (niveau Tactique)

Pour qu'un livrable logiciel soit considéré comme "terminé" tout en assurant une qualité logicielle cohérente, les conditions suivantes doivent être remplies :

#### AQ logicielle :

##### Revue par les Pairs : Chaque portion de code doit avoir été examinée par au moins un autre développeur pour valider la qualité du code, son alignement avec les standards de codage établis, et l'absence d'erreurs évidentes.

##### Tests Unitaires et d'Intégration : Des tests couvrant à la fois les composantes individuelles (unités) et leurs interactions (intégration) doivent être écrits et réussis. Un taux de couverture de code défini doit être atteint pour assurer que le logiciel fonctionne comme prévu.

##### Validation des Fonctionnalités : Les fonctionnalités développées doivent être validées contre les exigences spécifiées, à travers des tests fonctionnels (manuellement ou via automatisation) pour s'assurer qu'elles répondent aux besoins des utilisateurs.

Pour l'assurance qualité de la documentation, les conditions suivantes doivent être satisfaites :

#### AQ de la documentation :

##### Exhaustivité : La documentation doit couvrir tous les aspects nécessaires pour comprendre, déployer et maintenir le livrable. Cela inclut les guides d'utilisation, les spécifications techniques, et les manuels d'opération.

##### Revue par les Pairs : La documentation doit être revue par une personne autre que l'auteur pour garantir la clarté, la précision et la cohérence des informations fournies.

##### Validation des Parties Prenantes : La documentation doit être approuvée par les parties prenantes concernées (par exemple, les gestionnaires de projet, les équipes de support, et les utilisateurs finaux) pour s'assurer qu'elle répond à leurs besoins et attentes.

La définition de "terminé" est un outil crucial pour assurer l'alignement de l'équipe sur les critères de qualité et la complétude des livrables avant le déploiement en production.

#### Liste des activités (niveau Opérationnel)

[Voir lien vers Excel des activités, obstacles et risques : LesTouristes-Liste\_Activites-Gabarit\_1](https://uqam-my.sharepoint.com/personal/fh691920_ens_uqam_ca/Documents/hiver%202024/inf6150/devoir/tp2/RemiseTP2/LesTouristes-Liste_Activites-Gabarit_1.xlsx)