

Projet d'évolution logiciel
Plan de projet

Version 2.2

Historique des révisions

Date	Version	Description	Auteur
2021-01-29	1.0	contributions à l'énoncé des travaux et à la gestion des exigences	Raphaël Lasalle
2021-01-31	1.1	création d'un organigramme pour le cahier de charge	Raphaël Lasalle
2021-02-01	1.2	début de la section gestion de risques	Jaafar Kaoussarani
2021-02-04	1.3	Ajout de risques	Jaafar Kaoussarani
2021-02-04	1.4	Ajout de risques concernant la cohérence des interfaces usager	Davi Fraval
2021-02-05	1.5	Ajouts aux sections 2.2, 4 et 5	Jaafar Kaoussarani
2021-02-06	1.6	Ajustement de la section solution proposée	Jaafar Kaoussarani
2021-02-06	1.7	Ajout de la section 3.4 Gestion des configuration	David Fraval
2021-02-09	1.8	Corrections à la section sur la gestion des risques	Jaafar Kaoussarani
2021-02-18	1.9	Révision des section 1, 2 et 3	Jaafar Kaoussarani
2021-02-18	1.10	Ajout des lots, des charges de travail, des phases et des jalons	Raphael Lasalle Jaafar Kaoussarani
2021-04-14	2.0	Ajustements aux sections 1, 2, 3, 5 et 6	Jaafar Kaoussarani
2021-04-14	2.1	Ajustements à la section 4	Jaafar Kaoussarani
2021-04-14	2.2	Ajout du type de contrat	Jaafar Kaoussarani

Table des matières

1. Introduction	4
2. Énoncé des travaux	4
2.1. Solution proposée	4
2.2. Hypothèses et contraintes	4
2.2.1 Hypothèses	4
2.2.2 Contraintes	4
2.3. Biens livrables du projet	4
3. Gestion et suivi de l'avancement	5
3.1. Gestion des exigences	5
3.2. Contrôle de la qualité	5
3.3. Gestion de risque	5
3.4. Gestion de configuration	6
4. Échéancier du projet	7
5. Équipe de développement	11
6. Entente contractuelle proposée	11

Plan de projet

1. Introduction

Ce document décrit de manière générale le processus utilisé pour la réalisation et la gestion de l'application *Fais-moi un dessin*. Pour se faire, il y aura d'abord un énoncé des travaux, où nous présenterons la solution proposée, les hypothèses et contraintes et les biens livrables du projet. Nous passerons ensuite à la gestion et le suivi de l'avancement, qui comportera la description de la gestion des exigences, du contrôle de la qualité, de la gestion des risques et de la gestion de la qualité. Nous terminerons finalement par une vue d'ensemble de l'échéancier du projet, une introduction à l'équipe de développement et l'entente contractuelle proposée.

2. Énoncé des travaux

2.1. Solution proposée

Notre équipe souhaite répondre à l'appel d'offre en livrant une application de dessin en réseau qui contient les fonctionnalités listées dans le document de vision ainsi que le document de spécification des requis. Le produit livré contiendra notamment des fonctionnalités de dessin multijoueur, de clavardage texte, la possibilité d'utiliser des joueurs virtuels avec des personnalités distinctes pouvant dessiner à partir de paires mot-image, un tutoriel interactif et divers effets visuels et sonores. Elle sera entièrement utilisable aussi bien sur ordinateur (client lourd) que sur tablette (client léger). La zone de dessin utilisera la base de code du projet *PolyDessin* et ce projet sera une en quelques sortes une évolution de ce dernier.

À cela s'ajoute des fonctionnalités proposées par notre équipe qui seront souhaitables à implémenter. Cela inclut notamment de donner une mémoire aux joueurs virtuels, ainsi que d'ajouter un nouveau mode de jeu, divers modificateurs et options additionnels pouvant être appliqués à une partie et un système social permettant d'ajouter des amis, de rejoindre leur partie et de consulter leur profil.

Les détails des fonctionnalités mentionnés précédemment sont tous inclus dans le document SRS.

2.2. Hypothèses et contraintes

2.2.1 Hypothèses

Nous supposons que tout les membres de l'équipe seront disponible jusqu'à la date de livraison du produit final, soit le 19 avril 2021.

Nous supposons que le client ne demandera pas à l'équipe d'ajouter des exigences qui ne sont pas spécifiées dans le SRS du premier livrable.

2.2.2 Contraintes

L'équipe de développement consistera de cinq développeurs et nous ne pouvions pas en engager plus.

Dû à la situation pandémique actuelle, l'équipe ne peut pas se procurer une tablette pour le développement du client léger. Ainsi, le développement se fera exclusivement à partir d'un émulateur de tablette. La tablette émulée sera une Samsung Galaxy Tab A 2019.

Le premier livrable doit être envoyé d'ici le 19 février 2021 et cette date ne peut pas être changée.

La date de livraison du produit final est fixée au 19 avril 2021 et cette date ne peut pas être changée.

Le nombre d'heures-personnes est de 900 pour l'entièreté du projet.

2.3. Biens livrables du projet

Le premier livrable à remettre doit comprendre, en plus du document présent, un document de spécifications des requis du système (SRS), un document d'architecture logicielle, un document de protocole de communication, ainsi qu'un prototype de communication entre le client lourd, le client léger et le serveur. Ce livrable sera remis avant le 19 février 2021 à 23h59.

Le deuxième livrable à remettre doit comprendre des versions mises à jour des artefacts du premier livrable, un plan de test, les résultats de tests, ainsi que le produit final. Ce dernier consiste en le code compilé et un exécutable pour le client lourd et le code source pour le client léger. Ce livrable sera remis avant le 19 avril 2021 à 23h59.

3. Gestion et suivi de l'avancement

3.1. Gestion des exigences

Le contrôle des exigences passe avant tout par une communication continue entre tous les membres de l'équipe. Tout changement significatif des exigences doit être amené en réunion, qui a lieu au moins deux fois par semaine. La réunion a pour but d'évaluer la charge de travail représentée par le changement, ainsi que la meilleure approche pour intégrer ce changement au processus. Lorsqu'un changement est approuvé, la charge de travail associée est enregistrée et répartie entre les différents membres grâce à l'outil Jira par un membre de l'équipe. Tout les artefacts concernés par ce changement seront aussi mis à jour.

Les changements aux exigences peuvent arriver dans deux situation. La première est suite à la rencontre en présentiel avec le client (soit la rencontre au L-4812) où, suite à une démo, il propose des changements au comportement ou aux éléments visuels des fonctionnalités de l'application. La deuxième situation serait que, lors de la planification du prochain Sprint de travail, l'équipe observe qu'il n'y a pas assez de temps pour accomplir tout ce qui a été planifié. Des exigences souhaitables seront alors coupés selon la processus du paragraphe précédent, soit en mettant à jour la liste des tâches sur Jira et en répartissant les heures de travail vers d'autres tâches existantes.

3.2. Contrôle de la qualité

Afin d'assurer la qualité du livrable, l'état d'avancement du projet sera amené en réunion à chaque semaine. Dans les cas où l'état réel du projet diffère des artefacts, un ou plusieurs membres de l'équipe auront comme tâche de mettre à jour le(s) document(s) concerné(s).

De plus, chaque artefact du livrable, incluant le document présent, doit être relu par un minimum de deux membres de l'équipe (excluant l'auteur) avant d'être rendu sur la branche principale. Cette révision se fait dès que l'auteur indique que l'artefact ou la tâche qui l'englobe est complété. Comme pour la gestion des exigences, les contributions de chaque membre seront enregistrées avec Jira.

3.3. Gestion de risque

La description des risques suit la convention suivante :

- Ampleur : sur une échelle de 1 à 10, 10 étant le risque le plus élevé. Cette analyse est basée sur la probabilité d'occurrence du risque, ainsi que ses impacts.
- Description : Une description textuelle du risque ainsi que les problèmes attendus.
- Impact : échelle définissant la portée du risque
 - C – critique (affecte le projet en entier)
 - E – élevé (affecte les fonctionnalités principales du système)
 - M – moyen (devrait être maîtrisable en appliquant une stratégie d'atténuation adéquate)
 - F – faible (l'acceptation du risque est une stratégie envisageable)
- Facteurs : aspects (métriques) du système pouvant être compromis.
- Stratégie de gestion : mesures à prendre afin de gérer le risque.

Ampleur	Description	Impact	Facteurs	Stratégie de gestion
7	Ce risque fait référence au fait que le serveur utilise les mêmes canaux de communication pour les deux clients. Étant donné qu'ils utilisent des langages de programmation différents, il peut arriver, par exemple, qu'une fonctionnalité marche bien quand les données proviennent du client lourd, mais pas lorsqu'elles proviennent du client léger, et vice versa.	C	Utilisabilité, Fonctionnalité, Fiabilité	Nous utiliserons un format uniformisé pour les paquets qui seront transférés des clients vers le serveur, soit le format JSON, qui permet de représenter des objets en <i>string</i> . En utilisant cela, on garantit que toutes les données qui entrent dans le serveur ont toujours le même format, minimisant ainsi ce risque.

2 - Retard au niveau d'un Sprint dû à une sous-estimation du travail

Ampleur	Description	Impact	Facteurs	Stratégie de gestion
6	Ce risque fait référence au fait de sous-estimer la charge de travail que demande une tâche, notamment dû à un manque de connaissances techniques ou une mauvaise gestion du temps. Cela peut non seulement causer un coupage de certaines fonctionnalités, mais aussi un manque de temps de testage, ce qui pourrait nuire au fonctionnement, à l'intégrité et à la fiabilité des fonctionnalités déjà implémentées.	E	Fonctionnalité, Intégrité, Fiabilité, Utilisabilité	Proactive: Considérer l'opinion de tout les développeurs lors de la planification des tâches à faire et réviser ces dernières régulièrement et en équipe Réactive: Lors de la réunion suivante il y aura analyse des raisons du retard, une priorisation et un nouveau plan adapté au retard qui risque d'être moins ambitieux et qui veillera à limiter les facteurs qui nous ont placés dans cette situation.

3 - Interface utilisateur esthétiquement trop différente entre les deux clients

Ampleur	Description	Impact	Facteurs	Stratégie de gestion
5	Les interfaces graphiques respectives des clients lourd et léger sont assez esthétiquement différentes pour qu'un utilisateur qui utilise les deux applications subséquemment mentionne que ça le dérange.	M	Utilisabilité	Utilisation régulière du client lourd par les développeurs du client léger, et vice-versa. Communication constante entre les développeurs du client lourd et du client léger.

4 - Fonctionnalités trop différentes entre les deux clients

Ampleur	Description	Impact	Facteurs	Stratégie de gestion
6	Les clients lourds et légers sont suffisamment différents en fonctionnalité pour que cela nuit à l'expérience de une de ces deux catégories d'utilisateurs.	E	Utilisabilité, Fonctionnalité	S'assurer de respecter tous les requis du SRS qui ne spécifient pas un seul client dans les deux clients Communication constante entre les développeurs du client lourd et du client léger.

3.4. Gestion de configuration

Afin de bien organiser les changements, nous avons opté pour une utilisation de git à 5 branches qui sont toujours présentes:

1 branche serveur, 1 branche client léger, 1 branche client lourd, 1 branche dev et 1 branche master. Cette configuration n'est pas laissée au hasard. Les 3 premières branches citées permettent à chaque personne responsable de sa partie respective du développement de commit des changements de comportement atomiques qui risquent d'engendrer des comportements inattendus par les autres (en particulier par le responsable du serveur) sans que cela se répercute directement sur les versions des autres. Elle permet d'expérimenter spontanément en gardant des sauvegardes granulaires. La branche dev permet de mettre le tout en commun entre les différentes parties (pour que les responsables du client lourd puissent avoir les derniers ajouts du serveur par exemple) sans pour autant être obligé de s'assurer que les solutions aient une qualité prête à la production. Enfin la branche master sert à sauvegarder les versions du logiciel étant passées à travers des tests de qualité digne d'une sortie. La logique sous-jacente de cette structure en 5 branches est le compromis entre intégration continue fréquente et stabilité qui résulte avec une logique de percolation filtrée des versions les plus stables du système.

Pour la mise en commun du code, il y aura des merge requests sur gitlab dont nous sommes automatiquement prévenus grâce à un webhook sur discord. On pourra voir ces requêtes de merge puis les approuver si tout va bien, ou bien parler des problèmes qu'on y trouve autrement pour pouvoir en faire collaborativement une meilleure version.

Le format des commits est "[type de tâche](ID-JIRA-DU-TICKET) Description".

Le numéro de version des artefacts prend la forme X.Y. Le Y est mis à jour à chaque fois que l'artefact est changé lors avant la remise d'un livrable. Le X est mis à jour lors du premier changement après la remise d'un livrable.

4. Échéancier du projet

On suppose que le lot de travail sera partagé également entre les membres de l'équipe de développement. L'estimation initiale du temps à allouer à ce projet est de 840 heures-personnes, soit une base de 42h/crédit/personne mais en considérant la charge de travail et la grande incertitude qui vient avec un projet de développement, nous nous fixons un seuil de travail de 960 heures-personnes, soit une base de 48h/crédit/personne. Ainsi, cela estime que chaque membre travaillera entre 168 et 192 heures pour la durée de ce projet, soit entre 12.9 et 14.8 heures par semaine. En moyenne, cela donnerait 900 heures-personnes, ce qui correspond à 180 heures de travail par membre, soit 13.8 heures par semaine. Nous prendrons le temps de travail moyen comme base d'estimation du travail à faire.

Les principaux lots de travail et leur pourcentage d'effort estimé sont la rédaction des artefacts (15%), l'écriture des tests (5%), le développement du système de clavardage (10%), le développement du système d'authentification et de profils utilisateurs (5%), le développement des joueurs virtuels (10%), le développement de la création d'une paire mot-image (15%), le développement du système de création et de recherche d'une partie (10%), le développement des parties elles-mêmes (20%) et le développement des fonctionnalités sociales (10%). Voir l'annexe A pour un organigramme détaillé de ces lots et de leurs sous-divisions.

Le projet est divisé en 4 sprints, qui excluent la phase "initiale". Cette dernière comprend la réalisation du SRS, du plan de projet, du document d'architecture, du protocole de communication, du prototype de communication ainsi que la liste d'exigences.

Itération	Durée
Sprint 1	19 février - 5 mars
Sprint 2	5 mars - 19 mars
Sprint 3	19 mars - 2 avril
Sprint 4	2 avril - 16 avril

Jalon	Taux d'achèvement du projet	Lots de travail à compléter	Date prévue
1er jalon	26 %	clavardage, tutoriel, authentification et profil utilisateur	5 mars
2ème jalon	56 %	recherche et création de partie, mécanismes de partie	19 mars
3ème jalon	81 %	joueurs virtuels, création de paire mot-image	2 avril
4ème jalon	100 %	fonctionnalités sociales, tests	16 avril

Lot de travail	Estimation du temps consacré (heures-personnes)	Date de remise prévue
Artefacts	99	19 février
Tutoriel	45	5 mars
Clavardage	45	5 mars
Profils utilisateurs et Authentification	45	5 mars
Création et recherche de partie	90	19 mars
Mécanismes de partie	180	19 mars
Joueurs virtuels	90	2 avril
Création de paire mot-image	135	2 avril
Fonctionnalités sociales	90	16 avril
Tests	81	16 avril
Total	900	

Les tableaux suivants donnent la décomposition en heures-personne prévue des tâches de chaque lot de travail. Les éléments en **rouge** ne s'appliquent qu'au client lourd.

Artefacts (avant le 19 février)	
Tâche	Estimation du temps consacré (heures-personne)

SRS	45
Plan de projet	22
Document d'architecture	22
Protocole de communication	10
Total	99

Sprint 1	
Tâche	Estimation du temps consacré (heures-personne)
Implémentation de l'affichage de la séquence d'instructions du tutoriel	15
Implémentation de la détection des interactions de l'utilisateur au tutoriel	30
Implémentation des modes d'affichage du chat	15
Implémentation des canaux de chat	30
Implémentation de l'authentification et de l'inscription	15
Implémentation des éléments du profil	30
Total	135

Sprint 2	
Tâche	Estimation du temps consacré (heures-personne)
Implémentation de l'affichage des parties disponibles	40
Implémentation de la création de partie	50
Implémentation de la logique du mode classique	40
Implémentation de la logique du mode chacun pour soi	20
Implémentation de du partage des modifications de dessin	40
Implémentation de l'expulsion d'un joueur	25
Implémentation des "power ups"	35
Implémentation de l'écran de fin de partie	20

Total	270
--------------	-----

Sprint 3	
Tâche	Estimation du temps consacré (heures-personne)
Implémentation de la reproduction des dessins par les joueurs virtuels	55
Implémentation de la logique des messages des joueurs virtuels	35
Implémentation des ordres de traçage d'une paire mot-image	60
Implémentation de la sauvegarde des informations de la paire mot-image	30
Implémentation de la conversion en dessin des images téléversés	45
Total	225

Sprint 4	
Tâche	Estimation du temps consacré (heures-personne)
Implémentation du système relationnel (ajouts d'amis, blocage, liste d'amis, etc)	29
Implémentation du statut connecté / déconnecté	26
Implémentation des interactions sociales lors de la recherche et du déroulement d'une partie	35
Document de plan de tests	18
Implémentation des tests	45
Document de résultats de tests	18
Total	171

Les dates de tombé des livrables sont les suivantes:

- Le 19 février 2021 pour la livraison des artefacts tels que ce plan de projet, le SRS, le document d'architecture logicielle et le protocole de communication, ainsi qu'un prototype de communication permettant d'envoyer des messages entre les clients.
- Le 19 avril 2021 pour la livraison du produit final. Cela inclut toutes les fonctionnalités essentielles et au

moins 50% des fonctionnalités souhaitables. On y inclut aussi les artefacts du premier livrable mis à jour, le plan de tests et les résultats de tests.

5. Équipe de développement

L'équipe de développement dispose de cinq développeurs au totale, chacun ayant ses responsabilités et points d'expertise:

- David Fraval: Responsable de la création et de la maintenance du client léger. Il est expert en développement Kotlin avec Android Studio.
- Hugo Palisson: Responsable de la création et de la maintenance du client léger. Il est expert en développement Kotlin avec Android Studio. Il est aussi responsable de l'assurance qualité du client léger.
- Jaafar Kaoussarani: Responsable de la création et de la maintenance du client lourd. Il est expert en développement Angular avec Electron.
- Raphael Lasalle: Responsable de la création et de la maintenance du client lourd. Il est expert en développement Angular avec Electron. Il est aussi responsable de l'assurance qualité du client lourd.
- Théo Turell: Responsable de la création et la maintenance du serveur et de la base de données. Il est expert en développement backend, notamment avec l'API Socket.io. Il est aussi responsable de l'assurance qualité de la communication client-serveur.

6. Entente contractuelle proposée

Étant donné que la commercialisation et donc les revenus liés à l'application sont incertains, le soumissionnaire propose une rémunération fixe à la réception de tous les livrables, dans le but de garantir un salaire pour les développeurs et les gestionnaires de projet.

Nous supposons tout d'abord que les cinq membres de l'équipe se partagent les responsabilités de développeur et de gestionnaire de projet. Nous estimons en plus un ratio de 4h de développement pour chaque heure de gestion de projet pour chaque individu. À partir d'une estimation moyenne de 900 heures-personnes pour une équipe de 5 personnes, nous estimons une moyenne de 180 heures de travail par personne pour la réalisation de ce projet, soit 144 heures consacrées aux activités de développement et 36 heures consacrées aux activités de gestion de projet. En considérant un taux horaire de 100\$/h pour un développeur et 125\$/h pour un gestionnaire de projet, notre équipe demande donc un prix de 94 500 \$ (CAD) pour le travail total de l'équipe. Voir la section 4 pour plus de détails sur cet estimation et les lots de travail.

En échange, notre équipe s'engage à remettre les livrables aux dates et heures convenues dans l'appel d'offre, soit le 19 février 2021 avant 23h59 pour le document de spécification des requis du système (SRS), la liste d'exigences, le document d'architecture logicielle, le document de protocole de communication ainsi que le document présent, et le 19 avril 2021 avant 23h59 pour le produit final, le plan des tests ainsi que le document de résultat des tests. Notre équipe s'engage également à respecter le format de remise spécifié pour les artefacts et le code.

Nous proposons de mettre en place un contrat à terme. En effet, notre équipe ne peut plus travailler sur le code de ce projet après le 19 avril. Comme mentionné précédemment, l'équipe s'engagera à livrer un produit contenant toutes les exigences essentiels et une bonne partie des exigences souhaitables, mais le nombre de ces derniers qui seront implémentées est avant tout dépendant du temps restant avant la date limite du 19 avril. Donc, l'équipe donnera suffisamment d'effort pour livrer un produit qui est à l' hauteur des attentes du client, mais sans possibilité d'extension de la date limite.

Annexe A

Cahier de charges pour l'application *Fais-moi un dessin*

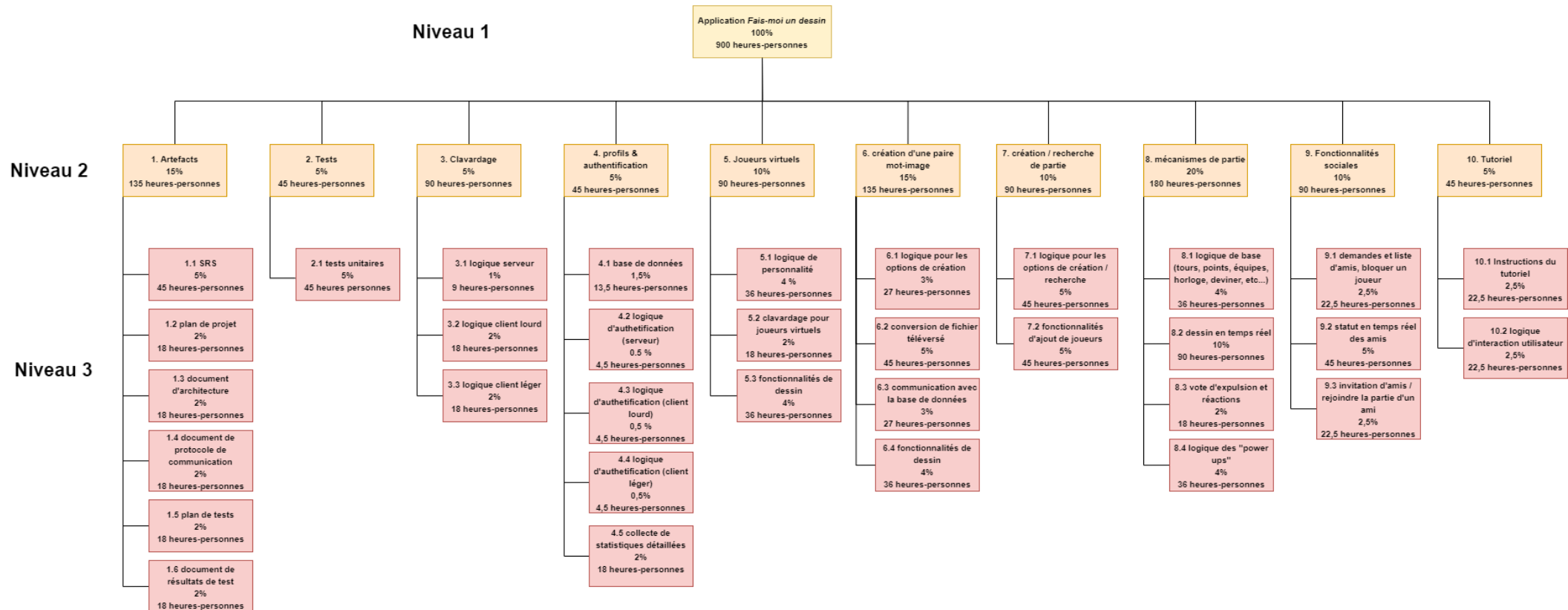


Figure 1 : Organigramme des lots de travail pour la réalisation de l'application *Fais-moi un dessin*