

Noowow community

Projet pédagogique (Facebook messenger)

Objectifs

Reproduire Facebook messenger sans les appels vidéos et audios dans le cadre d'un projet pédagogique.

Processus de développement

Processus unifié

Le Processus Unifié (UP, pour Unified Process) est un processus de développement logiciel « itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques »

- **Itératif et incrémental:** le projet est découpé en itérations de courte durée (environ 1 mois) qui aident à mieux suivre l'avancement global. À la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale.
- **Centré sur l'architecture:** tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML, en appliquant la célèbre règle des 80/20 qui consiste à modéliser 80 % des problèmes en utilisant 20 % d'UML.
- **Piloté par les risques:** les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.
- **Conduit par les cas d'utilisation:** le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés.

Les différentes phases du processus unifié

Initialisation: conduit à définir la « vision » du projet, sa portée, sa faisabilité, son business case, afin de pouvoir décider au mieux de sa poursuite ou de son arrêt. (On va sauter cette phase puisqu'on est dans un contexte de reproduction de système déjà établi.

Elaboration: poursuit trois objectifs principaux en parallèle :

- identifier et décrire la majeure partie des besoins des utilisateurs,
- construire (et pas seulement décrire dans un document !) l'architecture de base du système,
- lever les risques majeurs du projet.

Construction: consiste surtout à concevoir et implémenter l'ensemble des éléments opérationnels (autres que ceux de l'architecture de base). C'est la phase la plus consommatrice en ressources et en effort.

Transition: permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux. Les mots-clés sont : conversion des données, formation des utilisateurs, déploiement, bêta-tests.

Chaque phase est elle-même décomposée séquentiellement en itérations limitées dans le temps (entre 2 et 4 semaines). Le résultat de chacune d'elles est un système testé, intégré et exécutable.

EXtreme Programming (XP)

L'eXtreme Programming (XP) est un ensemble de pratiques qui couvre une grande partie des activités de la réalisation d'un logiciel, de la programmation proprement dite à la planification du projet, en passant par l'organisation de l'équipe de développement et les échanges avec le client.

Scrum

Le principe de base de Scrum est de focaliser l'équipe de façon itérative sur un ensemble de fonctionnalités à réaliser, dans des itérations de

deux, trois ou quatre semaines, appelées Sprints. Chaque Sprint possède un but à atteindre, défini par le directeur de produit (Product owner), à partir duquel sont choisies les fonctionnalités à implémenter dans ce Sprint. Un Sprint aboutit toujours sur la livraison d'un produit partiel fonctionnel. Pendant ce temps, le scrum master a la charge de réduire au maximum les perturbations extérieures et de résoudre les problèmes non techniques de l'équipe.

Un principe fort en Scrum est la participation active du client pour définir les priorités dans les fonctionnalités du logiciel, et choisir lesquelles seront réalisées dans chaque Sprint. Il peut à tout moment ajouter ou modifier la liste des fonctionnalités à

Roles et description

Architecte

L'architecte informatique s'occupe de construire la structure d'un système d'information, de la conception à la mise en place. Polyvalent, organisé, il prend en charge les contraintes et les attentes de chacun ainsi que les moyens à disposition afin de proposer une solution la plus efficace possible. Chef de projet, directeur d'équipe, il sait travailler rapidement et faire face aux imprévus, inévitables bugs et autres modifications de dernière minute...

Analyste

L'analyste en informatique analyse les besoins techniques des utilisateurs, conçoit les systèmes appropriés et assure leur formulation et leur implantation. Il effectue l'analyse des coûts et des économies qu'entraînera le système proposé. Il coordonne l'implantation des nouvelles technologies et applications.

L'analyste en informatique s'assure de l'utilisation optimale du parc informatique. Il organise les activités nécessaires à la surveillance de l'environnement technique.

Scrum master

Les Scrum Master sont les animateurs Scrum, le framework Agile léger qui met l'accent sur des itérations limitées dans le temps appelées sprints. En tant qu'animateurs, les Scrum Masters jouent le rôle de coachs pour le reste de l'équipe. Ce sont des « leaders-serviteurs », comme le dit le Guide Scrum. Les bons Scrum Masters s'engagent à respecter les bases et les valeurs Scrum, mais restent flexibles et ouverts aux possibilités d'amélioration du workflow de l'équipe. Ils remplissent souvent les fonctions suivantes:

- **Stand-ups** : animer les stand-ups quotidiens (ou Daily Scrums), si nécessaire.
- **Réunions de planification d'itération/de sprint** : protéger l'équipe contre les engagements excessifs et la dérive des objectifs. Aider à l'estimation et à la création de sous-tâches.
- **Revue de sprint** : participer à la réunion et recueillir le feedback.
- **Rétrospectives** : noter les domaines d'amélioration et les éléments d'action à adopter pour les futurs sprints.
- **Administration des tableaux** : travailler en tant qu'administrateur de tableau Scrum. S'assurer que les cartes sont à jour et que l'outil Scrum, Jira Software ou autre, fonctionne bien.
- **Réunions individuelles** : rencontrer individuellement les membres de l'équipe et les parties prenantes selon les besoins. Régler les désaccords au sein de l'équipe concernant le processus et les styles de travail. Bien que de nombreux experts Scrum soient opposés à cette méthode, car ils pensent que ces communications doivent avoir lieu pendant les stand-ups, certaines équipes, en particulier les nouvelles, préfèrent avoir ces interactions régulières en face à face avec des membres spécifiques de l'équipe. Le Scrum Master peut décider que ces interactions individuelles sont cruciales pour le développement de l'équipe et pour apprendre à se connaître.
- **Services de conseil internes** : les Scrum Masters doivent être prêts à consulter les membres de l'équipe et les parties prenantes internes sur la meilleure façon de travailler avec l'équipe Scrum.
- **Rapports** : analyse régulière des graphiques Burndown et d'autres outils de planification de portefeuille pour comprendre ce qui est développé et à quelle cadence.

- **Bloqueurs** : le Scrum Master aide l'équipe en éliminant les bloqueurs externes et en gérant les obstacles internes par l'amélioration des processus ou des workflows.
- **Travail prenant** : si l'équipe Scrum ne ronronne pas, c'est le problème du Scrum Master. Cela signifie peut-être qu'il faut réparer les ordinateurs cassés, déplacer les bureaux ou même régler le thermostat. Les Scrum Masters devraient être à l'aise pour faire à peu près n'importe quoi pour aider leur équipe, et ne devraient pas rechigner à aller chercher des cafés si c'est ce dont l'équipe a vraiment besoin.

Techlead

C'est une personne qui fait partie de l'équipe de développement, qui possède également une bonne connaissance de l'application afin de rapidement identifier certaines solutions techniques ou de bien présenter l'application à d'autres personnes. Il peut occuper ces rôles :

- Participer à l'élaboration du design technique des produits.
- Coacher les équipes informatiques et former les nouveaux arrivants.
- Collaborer à l'intégration des produits dans le SI avec l'architecte technique.
- Garantir la qualité des produits.
- Faire progresser les équipes.

Testeur QA

Le rôle du Testeur QA est de garantir le bon fonctionnement des logiciels en trouvant les bugs et problèmes potentiels d'un logiciel en les testant régulièrement. Le Testeur QA est également connu sous le nom testeur quality assurance ou de testeur informatique.

Développeur

À partir d'un cahier des charges défini comprenant l'ensemble des besoins et spécificités techniques, le Développeur Informatique se lance dans la création de programmes sur mesure. Il assure la conception des programmes informatiques ainsi que le suivi régulier après lancement, de la maintenance à l'évolution et l'amélioration du logiciel. Il peut également être amené à prendre en charge la formation des utilisateurs et rédiger des guides utilisateurs pour faciliter la prise en main d'une application.

Il peut être amené à:

- Identifier et analyser les besoins des utilisateurs pour présenter un projet au plus près des attentes du client ;
- Définir les solutions techniques possibles tout en respectant le cahier des charges ;
- Créer une version Beta du futur logiciel et réaliser une série de tests afin d'identifier les potentiels problèmes et les corriger ;
- Programmer la version finale du logiciel et participer au lancement ;
- Présenter un support d'aide technique pour les utilisateurs, et assurer si nécessaire des sessions de formation ;
- Réaliser un suivi pour la maintenance corrective ou évolutive ;
- Adapter ou améliorer les programmes informatiques aux besoins des utilisateurs.

Equipes et structure

Chaque équipe doit avoir:

- Un nom
- Un scrum master
- Un techlead
- Un ou plusieurs analystes
- Plusieurs développeurs
- Un ou plusieurs testeurs QA
- Un projet ou fonctionnalité à gérer

Liste des équipes

- Neptune
- Jupiter
- Mercure
- Mars

Produit

Perspective du produit

Le produit sera la reproduction d'une application web et mobile permettant à quiconque de tchatter, en l'occurrence Facebook messenger.

L'interface utilisateur

L'interface utilisateur doit être responsive, ce qui facilitera son utilisation sur les plateformes web et mobiles.

Les utilisateurs

Quiconque ayant un numéro de téléphone ou un email pourra utiliser l'application afin d'échanger des messages.

Les composantes logicielles

- HTML5
- CSS3
- Vanilla Javascript ES6+
- Node.js
- MongoDB / MySQL
- Websocket
- Tailwind CSS
- Swift pour l'implémentation native iOS
- Java / Kotlin pour l'implémentation native Android

Fonctions du produit

- Gestion des comptes utilisateurs
- Gestion des demandes d'amis
- Gestion des amis
- Gestion des conversations
- Gestion des recherches
- Gestion des status

Exigences fonctionnelles

A faire en équipe...

Cas d'utilisations

A faire en équipe...

Exigences non fonctionnelles

A faire en équipe...

Description de l'architecture

Architecture logique

Le système sera divisé en quatre sous-systèmes:

Frontend

- Regroupe toutes les fonctionnalités graphiques
- Indépendant du backend
- Doit être une SPA pouvant être hébergée sur un serveur statique

Backend

- Expose toutes les fonctionnalités via une API
- Doit être Restful
- Indépendant du frontend

IAM (Identity and access management)

- Deux sous systèmes, backend et frontend
- Regroupe toutes les fonctionnalités de gestion de compte utilisateur
 - Login
 - Logout
 - Changement de mot de passe
 - Enregistrement
 - Identity provider
 - Oublie de mot de passe
 - Confirmation d'email
 - Gestion de profil

Patron de conception

- MVC
- Singleton
- Facade
- Factory
- Proxy
- Observer
- DI (injection de dépendances)

Principes

- SOLID
- IOC
- DRY
- KISS

Diagramme de contexte

A faire en équipe...

Diagramme de classe

A faire en équipe...

Cadre de travail (framework)

Le cadre de travail définit une structure et des outils de base pour les projets. Il sera implémenté from scratch et sera divisé en deux composantes:

Frontend

- Services
- Core
 - router
 - Stubs
 - components
 - container
 - Store
 - Services
- Components
- Middleware
- Pages
- services
- models
- assets
- Store
- Test
- Public

- Lang
- Routes

Backend

- Services
- Core
 - router
 - Stubs
 - container
 - Services
 - controller
- Middleware
- services
- models
- assets
- Test
- Lang
- Routes
- Database
 - Migrations