
Gestion de projet informatique

OpenClassrooms

Cyber 2A



08-09-2020

Table des matières

1	Cahier des charges	3
2	Planification d'un projet	3
3	Méthodologies de gestion de projet	3
4	Méthodologie séquentielle	4
4.1	Définition du projet	4
4.2	Conception de l'architecture du programme	4
4.3	Développement et test	5
5	Méthodologie agile	5
6	Scrum : la méthodologie agile populaire	6

Liste des tableaux

Table des figures

1 Cahier des charges

- besoins et attentes du client ;
- budget, contraintes et deadlines ;
- rédigé initialement coté client ;
- transfert au prestataire et affinement du contenu ;
- aspect **contractuel** : base du contrat, accord mutuel ;
- structure du contenu :
 - ♦ **contexte** : objectifs, responsables, procédures de validation ;
 - ♦ **besoins** fonctionnels, techniques, organisationnels et contraintes ;
 - ♦ prestations et **livrables** attendus ;
 - ♦ contexte de réponse : appel d'offres, règles ;

2 Planification d'un projet

- **communication** : dialogue, approfondissement, mise en accord ;
- **sérénité** : se concentrer sur l'essentiel au quotidien ;
- **qualité** : coordination facilitée -> travail de qualité.

3 Méthodologies de gestion de projet

- parties prenantes :
 - ♦ **MOA** (maîtrise d'ouvrage) : le client ;
 - ♦ **MOE** (maître d'oeuvre) : le concepteur ;
 - ♦ **AMO** (assistance à maîtrise d'ouvrage) : aide au client ;
- méthodologie **séquentielle** : réaliser les phases dans l'ordre ; document à remettre en fin de phase ; livrable code en fin de projet :
 - ♦ définition du projet : définition des fonctionnalités (spécifications fonctionnelles) ;
 - ♦ conception de l'architecture (transformation des spécifications fonctionnelles en spécifications techniques) ;
 - ♦ code de l'architecture : concrétisation de l'application ;
 - ♦ recette : test des fonctionnalités (manuel ou tests unitaires codés) ;
- méthodologie **agile** : issue de Toyota (Lean Software et Kanban) :
 - ♦ itérative et incrémentale : définition, conception, code et test pour chaque incrément ;
 - ♦ équipes responsables et collaboratives ;
 - ♦ cérémonial minimal ;
 - ♦ répondre aux besoins changeants ;

4 Méthodologie séquentielle

4.1 Définition du projet

Document : spécifications fonctionnelles.

Présentation :

- fonctionnalités du projet :
 - ◆ aperçu global des fonctionnalités ;
 - ◆ détail de chaque fonctionnalité ;
- document de validation :
 - ◆ support de communication entre les parties prenantes ;
 - ◆ présentation des membres et de leurs rôles.

Contenu :

- des spécifications avec :
 - ◆ auteur ;
 - ◆ scénarios utilisateurs ;
 - ◆ aperçu sous forme de diagramme ;
 - ◆ détails des actions utilisateurs.
 - ◆ décharge : vision évolutive ;
 - ◆ non-buts : ce qui n'est pas prévu ;
 - ◆ annotations pour certaines équipes ;

CONSEILS : s'adapter à l'audience ; appliquer l'acronyme KISS (Keep It Simple, Stupid) ; inclure les collaborateurs.

4.2 Conception de l'architecture du programme

Document : spécifications techniques.

Présentation : préparer en amont tous les éléments techniques nécessaire au développement de l'application.

Contenu :

- architecture technique générale : framework, pattern, langage, UML ;
- batchs / interface : natures des données entrantes et sortantes ;
- description des données : traitement des données ;
- description du code : version du langage, framework et guides de style ;
- plateforme matérielle : hardware utilisé / conseillé ;

4.3 Développement et test

Cycle : local -> test -> production :

1. développement en local ; collaboration via Git ;
2. recette -> staging / préprod ; tester toutes les fonctionnalités :
 1. tests unitaires : petite fonctionnalité / portion de code ;
 2. tests d'intégration : interconnexion des fonctionnalités ;
 3. tests manuels : rare car chronophages ;
3. production / livraison.

5 Méthodologie agile

Manifeste Agile : années 2000 ; valeurs et principes de l'agilité.

Les **valeurs** :

- individus et interactions > processus et outils ;
- logiciel opérationnel > documentation exhaustive ;
- collaborer avec le client > négociation contractuelle ;
- adaptation au changement > suivre un plan ;

Les **principes** :

- satisfaire le client en priorité ;
- accueillir favorablement les demandes de changement ;
- livrer le plus souvent possible des versions opérationnelles de l'application ;
- assurer une coopération permanente entre le client et l'équipe projet ;
- construire des projets autour de personnes motivées ;
- privilégier la conversation en face à face ;
- mesurer l'avancement du projet en matière de fonctionnalité de l'application ;
- faire avancer le projet à un rythme soutenable et constant ;
- porter une attention continue à l'excellence technique et à la conception ;
- faire simple ;
- responsabiliser les équipes ;
- ajuster à intervalles réguliers son comportement et ses processus pour être plus efficace ;

Les **pratiques** :

- **user stories** : fonctionnalité écrite du point de vue de l'utilisateur ;
- **pair programming** : collaboration dans le codage ;
- **continuous integration** : nouvelle fonctionnalité intégrée à la précédente ;
- **acceptance testing** : conditions à réunir pour déclarer une story terminée ;
- **iteration planning (sprint)** : durée courte estimée pour réaliser chaque story ;
- **daily scrum** : courte réunion qui passe en revue la veille, ajd et les points bloquants ;

6 Scrum : la méthodologie agile populaire

Présentation :

Traduit de l'anglais par *mêlée* ; valeurs et esprit issus du rugby (sens de l'équipe, contact, travail bien fait) :

Processus Scrum :

1. **backlog** : liste priorisée des fonctionnalités ;
2. **release** : mise en production régulière ;
3. **sprint** : phase de maximum un mois pour développer une ou plusieurs fonctionnalités ;

Etapes d'une story :

1. manuel utilisateur ;
2. documentation technique ;
3. code ;
4. tests ;

Rôles :

- **Product Owner** : rpz utilisateur final ; gestion des priorités ; alimentation du backlog en nouvelles fonctionnalités ; définition d'un release ;
- **Scrum Master** : faire appliquer les principes de Scrum ; éliminer tout obstacle au projet ;
- **Team** : réaliser le produit ; auto-gestion ;

Débuter un projet : sprint 0 : organisation du projet (prise de connaissance, recrutement de l'équipe et création du backlog).