Conduite de projet – part 2

Autocontrôler et rendre compte de l’avancement du projet

Sommaire

- Des approches classiques à l’agilité
- Les approches agiles :
  - Scrum
  - Kanban
- Suivit et pilotage d’un projet :
  - Le planning
  - L’avancement

Des approches classiques à l'agilité

La gestion de projet avec des processus classiques prévoit un **enchaînement linéaire d'activités**, du cahier des charges jusqu'à la livraison du produit final.

⇒La qualité du livrable dépend donc de la qualité du cahier des charges qui doit être très **rigoureux**.

⇒Dans la réalité le **besoin peut être évolutif**, ou alors la plateforme ou doit être déployé l'appli entraine une adaptation de cette dernière.

⇒Chaque **étape n'est exécutée qu'une fois** (une seule spécification, une seule phase de codage, un seul cycle de tests…).

Des approches classiques à l'agilité

Les approches agiles sont :

- Un ensemble de méthodes
- Un état d'esprit :

  On cherche à avoir une plus grande **flexibilité** dans le processus de la gestion de projet.
- Centrer sur la satisfaction du client

On va procéder à la réalisation du produit par **petit bloc** (étapes) que l'on va **itérer et incrémenter** jusqu'à aboutir au résultat rechercher.

- Cela va permettre l'élaboration d'un cahier des charge ou les étapes sont décrite de manière moins rigoureuse.
- On va pouvoir visualiser et adapter le produit au fil de l'eau.

Des approches classiques à l'agilité

Les méthodes de gestions de projet on pour but de maitriser **la qualité, le coût, les délais** (classique) et **la satisfaction client** (agile).

En générale en approche classique on commence par demander :

- Qu'es ce que vous voulez faire ?
- Pour quand ?
- A quel prix ?

En mode agile on demandera plutôt :

- Quel sont les fonctionnalités les plus importante du produit final ?

	Approche classique	Approche Agile
Qualité du livrable	Dépend d'un cahier des charges complet et précis, d'une exécution rigoureuse et sans heurt. La qualité est **déterminée** à l'avance.	Le client est invité à prioriser régulièrement ses demandes qui ne sont pas détaillées à l'avance, mais complétées au fur et à mesure. La qualité se **reconnaît** à chaque itération dans la limite de ce qui a été traité.
Coût du livrable	Fonction de la durée du projet et des moyens alloués. Le coût est **estimé** à l'avance.	Il ne s'agit pas d'une grandeur objective. Le coût est **révisé** périodiquement.
Délais du livrable	Fonction de la durée du projet et des moyens alloués. Sur la base d'un chiffrage adéquat, le délai est **fixé** d'avance.	Ce n'est pas une grandeur objective. Le délai dépend de la rapidité de **convergence** vers un résultat satisfaisant.
Satisfaction du client	Elle est atteinte quand l'exécution du projet est conforme au **plan**. Le client accepte l'effet tunnel puisqu'il s'attend à un livrable en tous points conforme à la spécification.	Le projet est piloté de façon à développer la satisfaction du **client**.

Les approches agiles : Scrum

La méthode **Scrum** (qui signifie mêlée) est l'une des méthodes phare de la gestion de projet en mode agile.

- C'est surtout dans cette méthode que l'on retrouve le principe d'itération que l'on appel **sprint.**

Principe de base :

Les **sprints structurent les projets Scrum**. Ce sont des cycles d'activités généralement assez courts où se succèdent des réunions appelées **cérémonies** et des phases de production.

Scrum est aussi un cadre se composant de plusieurs élément fondamentaux :

- des **rôles**,
- des **événements**,
- des **artefacts**,
- des **règles**.

Les approches agiles : Kanban

Kanban est plutôt une méthodologie qui permet de **contrôler le flux de travaille** que l'on appel couramment **Worflow** dans le but de **produire seulement sur demande** et de gaspiller un minimum de ressources.

On distingue cinq bonnes pratiques :

- **La visualisation** : Pour le workflow (peut prendre la forme d'un tableau ou l'on va répertorier les étape avec leurs états (à faire, en cours, en test, terminé)).
- **La limitation du nombre de tâches en cours**: Chaque étape du tableau ne peut contenir qu'un nombre maximum de tâches en même temps, défini en fonction des capacités de l'équipe. Lorsqu'une tâche est terminée, une nouvelle peut alors être ajoutée.
- **La gestion du flux** : il est essentiel de suivre, mesurer et consigner le déroulement du travail à travers chaque étape du tableau. Le but est de connaître la vitesse et la fluidité du travail.
- **L'explicitation des normes de processus** : les règles du système Kanban doivent être formulées clairement et sans ambiguïté afin de s'assurer que l'équipe comprenne le travail réalisé et les améliorations futures.
- **L'identification des opportunités d'amélioration**: une fois que l'équipe a compris les théories sur le travail, les processus et les risques, elle sera capable de discuter d'un problème ou d'un blocage auquel elle est confrontée et de trouver des améliorations à mettre en place.

Suivit et pilotage d’un projet

- Le **planning** organise le déroulement des opérations.
- Chaque opération est **une tâche** (unité d’œuvres) qu’il convient de réaliser.
- Il convient ensuite de faire évoluer le planning en fonction de **l’avancement** de chacune de ces tâches (éventuelle retard).

Suivit et pilotage d’un projet

1. <u>Recenser les tâches (unités d’œuvres) :</u>
   - Les tâches doivent être **étiquetées** pour garantir un suivi efficace. C’est-à-dire des noms clairement définis permettant de les reconnaitre dans n’importe quel documents.
   - Chaque tâches de développement doit intégrer un « **test unitaires** » pour valider son fonctionnement.
   - Une tâche peut être liée à une ou plusieurs autres tâches. En effet ça réalisation peut être conditionné par la réalisation d’autre tâches au préalable.
   - Une tâche est également définie par ses objectifs et ses critères d’achèvement.

Suivit et pilotage d'un projet

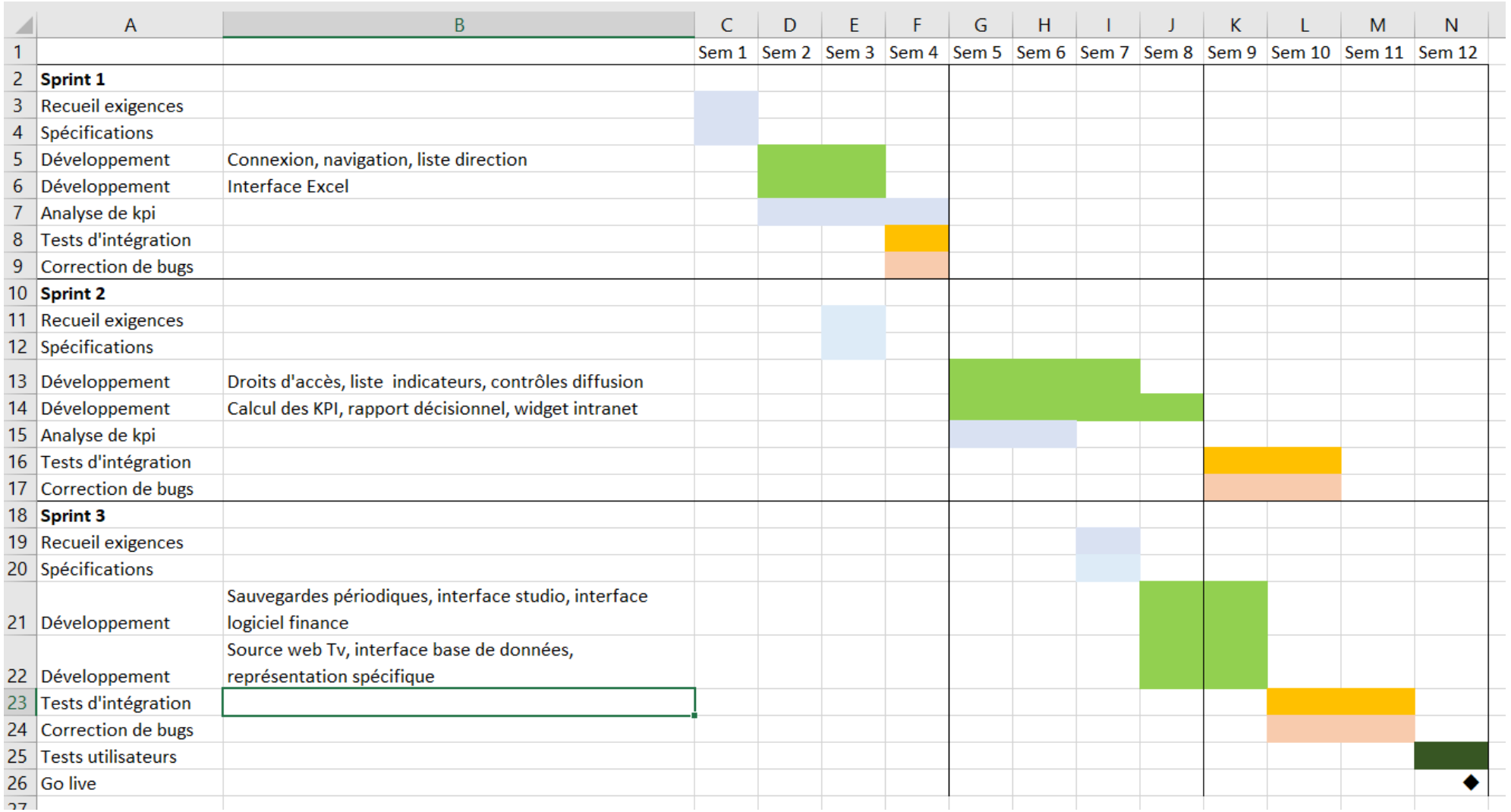
N°	Tâche	Durée jours	Tâches requises	Objectifs	Critères d'achèvement	Livrables
100	Modèle logique de données	1	99	Décrire la manière dont les données seront structurées	Tout les attributs sont classé dans une tables avec clés prim. étrg.	Schéma logique
101	Modèle physique de données	0,5	100	Retranscrire le modèle logique en langage SQL	Possibilité d'implémenter des données	Code SQL de la base de données
102	Programme de nettoyage et mise en forme des données	2		Traiter les valeurs manquantes, format date num, symbole no désiré, accent.	Données formalisées et uniformisées.	Code python

Suivit et pilotage d'un projet

2. La représentation des plannings : le diagramme de Gantt

- Le Gantt présente les **périodes d'activité par des barres colorées** limitées par le temps. Chaque barre correspond à l'exécution d'une tâche laquelle peut être discontinue.
- Le diagramme de Gantt peut **indiquer également les dépendances** entre les tâches
- Le Gantt n'est initialement **pas adapté au travail en mode agile** car le formalisme du planning va l'encontre de la flexibilité souhaité.
- Toutefois on peut le rendre agile en y faisant apparaitre les **Sprints**.
- Transformer alors son macroplanning en un Gantt donne plus de **possibilités d'analyse et d'anticipation**.

Suivit et pilotage d’un projet



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1			Sem 1	Sem 2	Sem 3	Sem 4	Sem 5	Sem 6	Sem 7	Sem 8	Sem 9	Sem 10	Sem 11	Sem 12
2	**Sprint 1**													
3	Recueil exigences													
4	Spécifications													
5	Développement	Connexion, navigation, liste direction												
6	Développement	Interface Excel												
7	Analyse de kpi													
8	Tests d'intégration													
9	Correction de bugs													
10	**Sprint 2**													
11	Recueil exigences													
12	Spécifications													
13	Développement	Droits d'accès, liste indicateurs, contrôles diffusion												
14	Développement	Calcul des KPI, rapport décisionnel, widget intranet												
15	Analyse de kpi													
16	Tests d'intégration													
17	Correction de bugs													
18	**Sprint 3**													
19	Recueil exigences													
20	Spécifications													
21	Développement	Sauvegardes périodiques, interface studio, interface logiciel finance												
22	Développement	Source web Tv, interface base de données, représentation spécifique												
23	Tests d'intégration													
24	Correction de bugs													
25	Tests utilisateurs													
26	Go live													

Suivit et pilotage d’un projet

3. <u>L’avancement de tâches :</u>

Dans un tableau d’avancement on peut faire apparaitre :

- **L’état d’une tâche** peut prendre la valeur « non démarrées », « en cours », ou « terminées ».
- On fait figurer une **estimation du nombre de jours** qu’il faudra pour la réaliser.
- Le nombre de **jours consommés** à une date t pour ça réalisation.
- Le nombre de **jours restants** pour terminer la tâches.
- **L’avancement** (en %) = (Consommé / (Consommé + Restant)) (* 100)

Suivit et pilotage d’un projet

	A	B	C	D	E	F	G	H	I
1	Avancement	22/12/2014							
2									
3	**Itération**	**Tâche**	**Etat**	**Estimation**	**Consommé**	**RAF**	**Avancement**	**Commentaire**	
4	Sprint 1	Recueil exigences	Terminé	5	4	0	100%		
5	Sprint 1	Spécifications	Terminé	5	6	0	100%		
6	Sprint 1	Développement Connexion, navigation, liste direction	Terminé	7,5	8	0	100%		
7	Sprint 1	Développement interface Excel	Terminé	10,5	9		100%		
8	Sprint 1	Analyse de kpi	En cours	15	15	2	88%	Retard 2 jours sans impact planning	
9	Sprint 1	Tests d'intégration	En cours	15	3	12	20%		
10	Sprint 1	Correction de bugs	Non démarré	15	15	15	50%		
11	Sprint 2	Recueil exigences	Terminé	5	5	0	100%		
12	Sprint 2	Spécifications	En cours	5	5	3	63%	Retard 3 jours sans impact planning	
13	Sprint 2	Développement Droits d'accès, liste indicateurs, contrôles diffusion	Non démarré				#DIV/0!		
14	Sprint 2	Développement Calcul des KPI, rapport décisionnel, widget intranet	Non démarré				#DIV/0!		
15	Sprint 2	Analyse de kpi	Non démarré				#DIV/0!		
16	Sprint 2	Tests d'intégration	Non démarré				#DIV/0!		
17	Sprint 2	Correction de bugs	Non démarré				#DIV/0!		
18	Sprint 3	Recueil exigences	Non démarré				#DIV/0!		
19	Sprint 3	Spécifications	Non démarré				#DIV/0!		
20	Sprint 3	Développement Sauvegardes périodiques, interface studio, interface logiciel finance	Non démarré				#DIV/0!		
21	Sprint 3	Développement Source web Tv, interface base de données, représentation spécifique	Non démarré				#DIV/0!		
22	Sprint 3	Tests d'intégration	Non démarré				#DIV/0!		
23	Sprint 3	Correction de bugs	Non démarré				#DIV/0!		
24	Sprint 3	Tampon	Non démarré				#DIV/0!		
25	Sprint 3	Tests utilisateurs	Non démarré				#DIV/0!		
26	Sprint 3	Limite de prise en compte de bugs	Non démarré				#DIV/0!		
27	Sprint 3	Go live	Non démarré				#DIV/0!		