

Cours : thème 4 - Question 9

Question 9 : en quoi un projet de système d'information est-il une réponse au besoin d'évolution de l'organisation ?

Notions abordées :

- Projet, ressources, phases, tâches et parties prenantes ;
- Planification et diagramme GANTT ;
- Coût d'un projet et budgétisation.

1. Qu'est-ce qu'un projet ?

1.1. Définitions

Projet : un projet est une série de tâches exécutées pour atteindre un ou plusieurs objectifs spécifiques répondant à un ou plusieurs besoins. Il se caractérise par un délai de réalisation, et *a fortiori* par un début et une fin identifiés, par un coût et une équipe en charge de sa réalisation.

Ressource : une ressource est une personne, un équipement, un matériel ou encore un coût nécessaire à la réalisation d'une tâche.

Donneur d'ordre : le donneur d'ordre est une dénomination souvent employée pour qualifier la personne à l'initiative d'un projet. Le donneur d'ordre est donc le demandeur.

Livrable : un livrable consiste en toute production, à destination du demandeur, matérialisant les réalisations de l'équipe projet. Un livrable peut être : un document (exemples : cahier des charges, spécifications, cahier de tests, etc.) ou encore tout ou partie de la réalisation (exemple : module d'un logiciel).

Lot : lorsqu'un projet est de grande envergure (coût ou délai), on le divise en lots indépendants. On dit qu'on lotit le projet (lot 1, lot 2, etc.). L'équipe projet livrera les lots les uns après les autres (livraison du lot 1, puis livraison du lot 2 et ainsi de suite).

1.2. Nature et contraintes d'un projet

Un projet doit être explicite. Il ne doit pas consister dans une simple idée informelle. Il est nécessaire de le décrire clairement. On peut le décrire par exemple en utilisant la méthode **QQOQCCP** :

- **Quoi ?** Quels sont les besoins, les résultats attendus, l'ergonomie ou encore la qualité souhaitée ?
- **Qui ?** Quelles sont les acteurs concernés par le projet ? Quelle sera l'équipe à le prendre en charge ?
- **Où ?** Quels sont les domaines touchés par le projet, voire les lieux ?
- **Quand ?** Quels sont les délais de réalisation ? Quel sera le planning ?
- **Comment ?** Comment le projet se déroulera-t-il et avec quels moyens ?
- **Combien ?** Quel est le budget ? (du projet lui-même, de la formation, de la maintenance, etc.).
- **Pourquoi ?** Quels sont les motifs du projet ? Quels sont les objectifs poursuivis ?

1.3. Risques en cours de projet

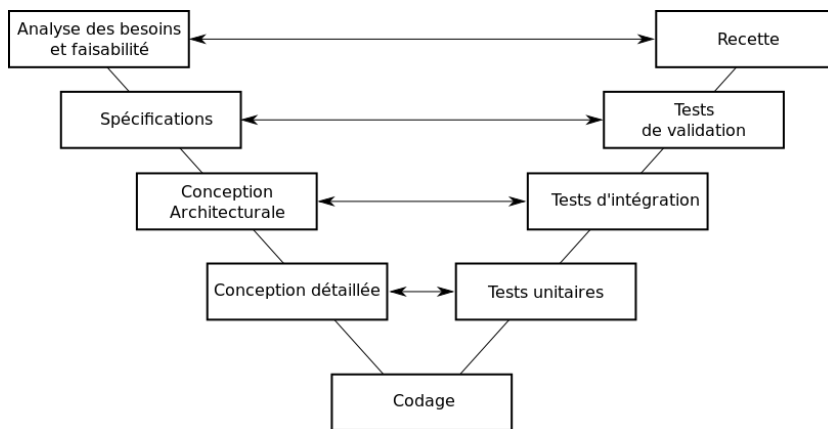
Un projet est guidé par plusieurs contraintes. On parle souvent de **contraintes QCD** : qualité, coût et délai. Un projet présente toujours un risque de non-respect de ces contraintes :

Contrainte	Risques associés
Qualité	<p><u>Risques internes</u> :</p> <ul style="list-style-type: none"> - cahier des charges imprécis, incomplet ou incohérent ; - évolution du cahier des charges en cours de projet ; - risques techniques : incompatibilité, complexité, performances insuffisantes, etc. ; <p><u>Risques externes</u> :</p> <ul style="list-style-type: none"> - évolution des réglementations en cours de projet ; - obsolescence des solutions choisies en cours de projet ; - indisponibilité des ressources (dont les personnels compétents).
Coût	<ul style="list-style-type: none"> - insuffisance initiale de budget ; - dépassement du budget en cours de projet.
Délai	<ul style="list-style-type: none"> - délais initiaux trop courts ; - dépassement de dates butoirs en cours de projets.

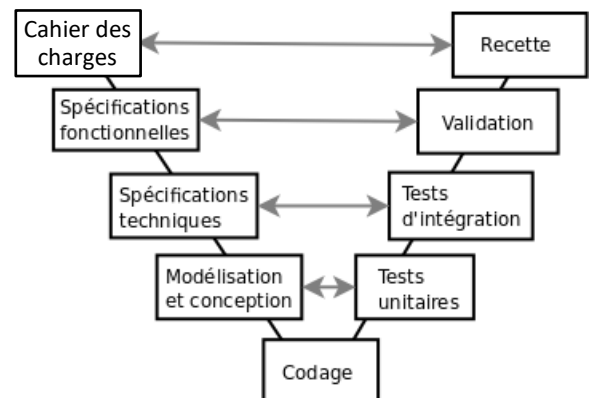
1.4. Déroulement d'un projet (informatique)

Une méthode classique de réalisation de projet s'appelle le **cycle en V**. Cette méthode met en évidence le fait qu'un projet comporte plusieurs phases, et qu'il ne comprend pas seulement la production d'un résultat. En effet, il faut également : savoir ce que le demandeur veut (analyse des besoins et faisabilité), définir la solution d'un point de vue utilisateur (spécifications), définir la solution d'un point de vue technique (conception architecturale et conception détaillée), réaliser la solution (codage pour un projet de logiciel). Finalement, il faut tester la réalisation pour s'assurer que le résultat fonctionne bien et répond aux exigences du demandeur. A chaque phase du projet correspond typiquement un livrable.

Phases d'un projet géré en cycle en V



Livrables correspondants



En-dehors du **cycle en V**, il existe d'autres méthodes de gestion de projet : les méthodes AGILE ou encore le *lean management*.

2. Gestion de projet

2.1. Introduction

On notera que la réussite d'un projet ne tient pas qu'à sa seule réussite technique. Une enquête du cabinet d'études Standish Group (2009) fait apparaître que :

- 24% des projets sont des échecs : abandon pur et simple ;
- 44% des projets sont douteux : dépassement des délais, des coûts et/ou absences de fonctionnalités, à savoir le projet est non conforme au cahier des charges ;
- 32% des projets sont des succès.

Les **facteurs clefs de succès** de la conduite d'un projet sont les facteurs contribuant à ce qu'un projet soit un succès. L'implication, l'engagement des parties prenantes (=acteurs) est un élément déterminant.

Un projet conduit, mené, de manière méticuleuse suit typiquement la **roue de Deming** :



2.2. Planifier

2.2.1. Identifier les tâches et leur ordre

Un projet s'inscrit dans le temps : il a un début et une fin. La planification (prévisionnelle) d'un projet consiste à diviser le projet en tâches élémentaires. Une tâche a une durée (de quelques heures à quelques jours*). Elle doit être réalisée par une ou plusieurs ressources humaines. La réalisation d'une tâche peut être subordonnée à la réalisation d'une ou plusieurs autres (**prédécesseurs**).

* Jours : **jours ouvrés** (lundi-vendredi hors jours fériés), **jours ouvrables** (lundi-samedi hors jours fériés).

Des tâches peuvent potentiellement être effectuées simultanément. On parle de **tâches parallèles**.

Exemple d'ordonnancement :

NUMERO	TACHES	DUREE	PREDECESSEUR
A	Analyse du logiciel	5 jours	-
B	Définition des chapitres	4 jours	-
C	Présentation de l'objectif	9 jours	A , B
D	Utilisation des différentes commandes	20 jours	C
E	Création d'exemples	20 jours	C
F	Fiches résumés	5 jours	D
G	Vérification des exemples	4 jours	E
H	Tests de clarté et de compréhension	8 jours	F , G
I	Correction des erreurs	4 jours	H
J	Organisation de la mise en page	2 heures	I
K	Envoi des impressions	2 heures	J

2.2.2. Planning

Il existe principalement trois représentations d'un planning de projet : le diagramme Gantt, le réseau PERT et le réseau MPM. **Le diagramme Gantt** (exemple d'outil logiciel : Gantt Project) fournit une représentation visuelle et intuitive d'un planning. Il permet de facilement visualiser le **chemin critique**.

Réf. tâche	Nom tâche	Durée	Prédécesseurs
A		2	-
B		4	-
C		4	A
D		5	A, B
E		6	C,D

Réf. Tâche	Nom Tâche	Durée	Durée															
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A		2	■	■														
B		4	■	■	■	■												
C		4			■	■	■	■										
D		5					■	■	■	■	■							
E		6										■	■	■	■	■	■	

Le chemin critique (en rouge dans l'exemple ci-dessus) correspond à la séquence de tâches qui détermine la durée totale du projet. Ce chemin est continu depuis le début jusqu'à la fin du projet. Tout retard

affectant une tâche du chemin critique affecte directement la durée du projet et donc sa date de fin. Une **tâche critique** est une tâche du chemin critique. Toute modification de la durée de l'une des tâches critiques impacte d'autant la durée totale du projet.

La **marge** est la possibilité qu'à une tâche d'être retardée sans impacter la date de fin du projet. Les tâches qui sont sur le chemin critique ont une marge nulle (=0). On utilise également la terminologie suivante : date de début au plus tôt, date de début au plus tard, date de fin au plus tôt, date de fin au plus tard.

Il advient que l'on ajuste le planning prévisionnel en cours de projet. Cela peut arriver en outre en cas de dépassement. Afin d'éviter des dépassements budgétaires et/ou afin d'éviter un dépassement de la *deadline* (date de fin du projet), on peut : utiliser des heures supplémentaires, augmenter les ressources humaines*, utiliser des solutions moins onéreuses, faire appel à du personnel moins onéreux, etc.

* L'augmentation des ressources humaines peut être interne ou externe. Elle est externe si on fait appel à du personnel extérieur et/ou à la sous-traitance.

2.2.3. Budget

Budgétiser ou **chiffrer** un projet, c'est établir le budget nécessaire à sa réalisation. On parle de **chiffrage**. En d'autres termes, il s'agit d'établir le prix d'un projet. Dans le cadre d'un projet à forte valeur ajoutée (exemple : projets logiciels), c'est le coût du travail, à savoir le coût des ressources humaines, qui représente très vite la plus grande partie du budget.

$$\text{Budget projet} = \text{coût des investissements} + \text{coût des ressources humaines}$$

Coût	Evaluation	Exemples
Investissement matériel	Prix de l'investissement	Serveurs, Ordinateur, Matériel réseaux, etc.
Investissement immatériel	Prix de l'investissement	Licences logiciels, brevets, etc.
Ressources humaines	Prix du travail : - prix du jour/homme fois nombre de jours/homme ; - le tarif journalier peut varier en fonction de la prestation ou encore de la qualification du personnel.	Projet de 40 jours/hommes : Développeurs : 400€ HT/jour, 19 jours Testeurs : 400€ HT/jour, 14 jours Chef de projet : 500€ HT/jour, 5 jours Formation : 900€ HT/jour, 2 jours Prix : 17500€ HT

2.3. Réaliser et suivre

Il convient bien entendu de réaliser le travail, à savoir de réaliser une-à-une les tâches planifiées. Afin d'assurer le suivi du projet (état d'avancement, écarts au prévisionnel, etc.), on utilise parfois un tableau de bord appelé *backlog* (exemple d'outil logiciel : Trello) sur lequel figure :

- les tâches à réaliser ;

- l'état des tâches à réaliser (exemple : à faire, en cours, à tester, terminé). Il y a par exemple une colonne par état et chaque tâche figure dans la colonne correspondant à son état ;
- le pourcentage d'avancement de chaque tâche ;
- le « **fait** » (temps total de travail effectué) et « **reste-à-faire** » (temps total de travail restant, souvent exprimé en jours-hommes).
- d'autres informations.

2.4. Vérifier et recetter

Dans tout projet, il convient de vérifier les résultats obtenus. C'est ce qu'on appelle les **tests** ou encore le *testing*. Il y a plusieurs niveaux de tests : tests unitaires, tests d'intégration, tests de validations et recette. En particulier, en fin de projet, on procède à la **recette**. On parle de **tests d'acceptations**. La recette consiste à vérifier la conformité de la réalisation au regard du cahier des charges. Dans le cadre d'un projet de logiciel, cela revient à tester toutes ses fonctionnalités. Les **tests d'acceptations** sont les tests nécessaires afin de vérifier la conformité de la réalisation. On distingue :

- les **tests d'acceptation utilisateurs** (UAT, *User Acceptance Tests*) effectués par les utilisateurs même du logiciel ;
- les **tests d'acceptation d'usine** (FAT, *Factory Acceptance Tests*) effectués par les testeurs de l'équipe projet. On préfère en général que le développeur ne soit pas le testeur (il est difficile d'être juge et partie).

Pour faciliter le *testing* et l'analyse des résultats :

- on peut établir des **rapports de tests** (éventuellement générés automatiquement) ;
- on automatise les tests. En matière de logiciel, cela revient à dire qu'on crée des programmes qui testent le programme (i.e. qui testent le logiciel) ;
- on crée des **tests de non régression**. On parle de régression lorsque, en passant à une nouvelle version (évolutions), des fonctionnalités qui fonctionnaient ne fonctionnent plus. Les tests de non régression visent à vérifier qu'il n'y a pas eu de régression.

2.5. Mettre en production, suivre, améliorer, accompagner

Une fois le projet validé par le demandeur, il est mis en production (il est livré). Mais un projet n'est jamais réellement clos. Il convient ensuite d'assurer :

- la **maintenance** : consiste dans la mise en place d'actions préventives et correctives visant à assurer le bon fonctionnement de la réalisation. On parle de continuité de service ;
- la prise en charge des **évolutions** et **corrections** : un projet continue à vivre à s'améliorer au fil du temps (amélioration continue). Des anomalies de fonctionnement peuvent par ailleurs être détectées une fois la réalisation en production. Il existe des normes et méthodes de suivi de projet, souvent qualifiées de gestion de ticket. Exemple : méthode ITIL ;
- la **formation**, la **documentation** et l'**accompagnement** : il ne suffit de livrer la réalisation. Il faut encore que les utilisateurs puissent l'utiliser. Pour ce faire, il faut fournir les modes d'emplois, à savoir la documentation utilisateur et/ou la documentation technique. Il convient, le cas échéant, de mettre en place des formations et/ou un accompagnement via une assistance technique/téléphonique.