□ kpi

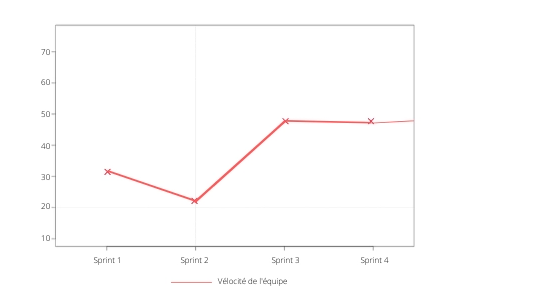
Kpi = indicateurs de performances

En agile, les indicateurs de performance concernent l’équipe.

L'efficacité d’une équipe Scrum se caractérise par la réussite de l’équipe à développer un certains nombre d’User Stories durant un Sprint.

L’efficacité en agilité peut se mesurer avec deux indicateurs de performance Agile:

1) La vélocité de l'Équipe



La vélocité mesure la quantité de travail (un certain nombre d’US) accompli dans un sprint.

Bien qu'il ne s'agisse pas d'un outil de prédiction ou de comparaison, la vélocité donne aux équipes une idée de la quantité de travail pouvant être effectuée lors du prochain sprint.

Elle se mesure en prenant en compte les Story Points des US considérées comme “Terminées” dans un Sprint.

Cette vélocité est unique pour chaque équipe et pour chaque produit.

Cependant ses variations sont intéressantes à regarder d’un sprint à l’autre.

Si l’équipe Scrum est nouvelle ou que le projet vient de démarrer, la vélocité des premiers sprint ne sera sûrement pas fiable.

Au fur et à mesure que les équipes acquièrent de l'expérience, les sprints se stabilisent et le calendrier des cérémonies Scrum se fluidifie, la vélocité atteint son pic.

Si, après analyse, on voit une diminution de la vélocité, cela indique des problèmes et un besoin de changement.

Pour analyser les performances de l’équipe, utilisez la moyenne des trois sprints précédents.

□ Découper un projet consiste ainsi à identifier des sous-ensembles quasi autonomes,

présentant les caractéristiques suivantes :

• chaque sous-ensemble du projet donne lieu à un résultat bien identifié ;

• la charge propre à chacun peut être évaluée ;

• les contraintes d’enchaînement entre les sous-ensembles sont repérables :

certains sous-ensembles peuvent être réalisés parallèlement, d’autres sont

liés entre eux par des contraintes d’antériorité ;

• le découpage est fait à des mailles différentes, un sous-ensemble étant souvent

à son tour décomposé.

On utilise deux grands critères pour découper un projet : l’un est temporel, l’autre structurel. Ces deux critères ne sont pas exclusifs.

Le critère temporel est utilisé dans la plupart des projets. Il permet de répartir

le travail dans le temps : la décomposition fait apparaître une succession d’étapes

et de phases. À chacune, on attache une date de début prévue et une date de fin

visée.

Un projet se compose de phases ; chaque phase comprend un certain nombre d’activités1 ; une activité est définie par une ou plusieurs tâches à effectuer.

À chaque élément de décomposition on attache un résultat à atteindre, appelé

livrable, qui peut faire l’objet d’un engagement contractuel.

L’ensemble ordonnancé des phases d’un projet s’appelle le cycle de vie du

projet.

Chaque date représente un jalon permettant de marquer les points de décision du

parcours.

Le client et le chef de projet ont tous deux

intérêt à découper le projet dans le temps, car ainsi :

*• Le client peut valider et orienter le projet.*

Le découpage temporel permet au client de s’assurer progressivement que

ce qui a été conçu traduit bien les objectifs généraux, de faire des choix,

éventuellement de réorienter le travail. En général, la fin d’une phase se

traduit par la livraison d’une fourniture contractuellement définie. La fin

d’une activité donne lieu à la remise de produits intermédiaires.

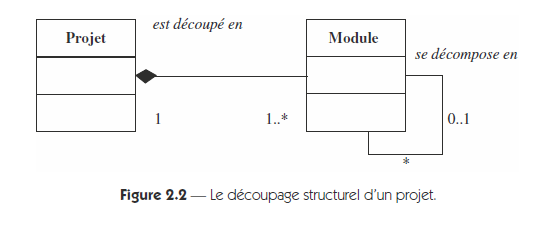
*• Le chef de projet peut baliser le déroulement du projet.*

S’il a découpé son projet en tranches, le chef de projet peut effectuer une planification et en suivre pas à pas l’avancement. La fin de chaque tranche

est comme un jalon où il regarde plus particulièrement s’il n’y a pas des

signaux inquiétants concernant l’état de santé du projet.

Le critère structurel permet d’organiser le travail en se basant sur la structure du produit final : la décomposition fait apparaître les différents modules qu’il faut obtenir.



Le découpage temporel standard ne peut guère être utilisé tel quel dans un projet système d’information.

Ensuite, dans le découpage temporel standard, on postule la possibilité pour le

client d’établir une description complète de ce qu’il attend. Or, dans notre

domaine, l’élaboration des spécifications, c’est-à-dire la détermination des besoins

et des solutions adéquates, est un problème majeur. Il y a souvent une construction

progressive, qui s’appuie sur des allers-retours entre une solution de gestion et des

possibilités techniques. Les besoins ne préexistent pas, ils émergent. Pour le chef

de projet, les étapes d’analyse et conception sont risquées. Une part non négligeable

du budget y est consacrée — jusqu’à 40 % — sans toujours atteindre la qualité

espérée. Par conséquent, la gestion de projet commence dès le début du projet.

□ LES MODÈLES DE DÉVELOPPEMENT

On considère aujourd’hui qu’on ne peut plus avoir une démarche unique, mais

qu’il faut construire le découpage temporel en fonction des caractéristiques de

l’entreprise et du projet. On s’appuie pour cela sur des découpages temporels

génériques, appelés *modèles de développement* (*process models*) ou modèles de cycle

de vie.

Les principaux modèles sont :

• le modèle du *code-and-fix* ;

• le modèle de la transformation automatique ;

• le modèle de la cascade ;

• le modèle en V ;

• le modèle en W ;

• le modèle de développement évolutif ;

• le modèle de la spirale.

***Le modèle de la cascade***

Le modèle de la cascade (*waterfall model*) est totalement opposé au modèle du

*code-and-fix*. Il a comme objectif majeur de jalonner rigoureusement le processus

de développement et de définir de façon précise les rôles respectifs du fournisseur

— qui produit un livrable — et du client — qui accepte ou refuse le résultat.

Le découpage temporel se présente donc comme une succession de phases

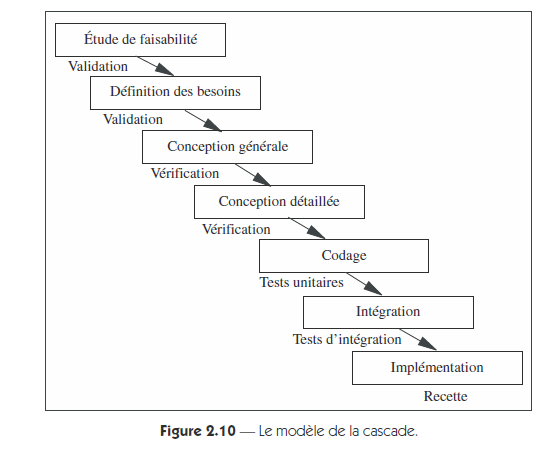
correspondant à une approche descendante (figure 2.10). Chacune donne lieu à

une validation officielle : on ne passe à la suivante que si le résultat du contrôle

est satisfaisant. Sinon, on modifie le livrable pour qu’il devienne acceptable. En

revanche, il n’y a pas de retour possible sur les options validées à l’issue des

phases antérieures.



***Le modèle en V***

Ce modèle (figure 2.11) est une amélioration du modèle de la cascade.

Il vise, d’une certaine façon, à réduire ce que l’on a appelé l’« effet tunnel » :

à partir d’un moment donné, les clients perdent la visibilité sur le projet. Quand

ce dernier ressort du tunnel, on découvre des livrables qui ne sont pas toujours

ceux que l’on attendait, non pas qu’ils ne soient pas conformes aux spécifications,

mais parce les spécifications sont parfois impuissantes à décrire les attentes. La

seule validation de documents est donc insuffisante.

Dans le modèle en V, on va donc s’attacher dans chacune des phases de la

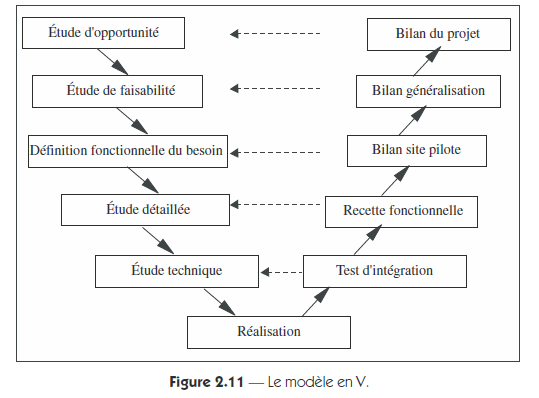
première branche du V à expliciter les critères d’appréciation et d’acceptation du

système aux étapes correspondantes de la deuxième branche du V. Par exemple,

l’étude détaillée produira un jeu d’essai qui sera utilisé pour effectuer la recette fonctionnelle.

L’installation sur un site pilote permettra de valider la définition

fonctionnelle du besoin, d’après les critères exprimés dans cette étape. Le bilan global du projet vérifiera que les objectifs initiaux formulés dans l’étude d’opportunité ont bien été atteints.



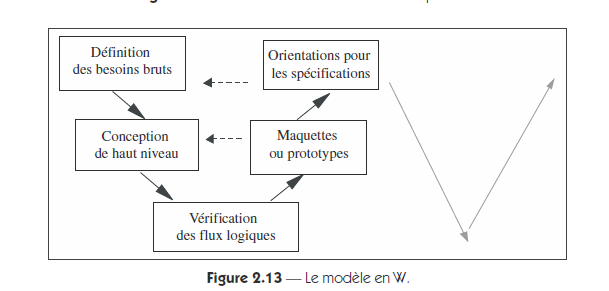
***Le modèle en W***

Ce modèle enrichit le modèle en V (figure 2.13) dans le même esprit d’anticipation

sur le livrable final.

La première partie du W vise à dégager avec les clients des orientations solides

pour la conception ou bien à explorer les possibilités d’une nouvelle technique.



Le développement de maquettes ou prototypes permet une validation plus concrète,

voire une expérimentation.

LES MODÈLES DE CYCLE DE VIE DES MÉTHODES AGILES

Toutes les méthodes agiles prennent en compte dans leur modèle de cycle de vie

trois exigences : une forte participation entre développeurs et utilisateurs, des

livraisons fréquentes de logiciel et une prise en compte de possibles changements

dans les besoins des utilisateurs au cours du projet. C’est pourquoi toutes font

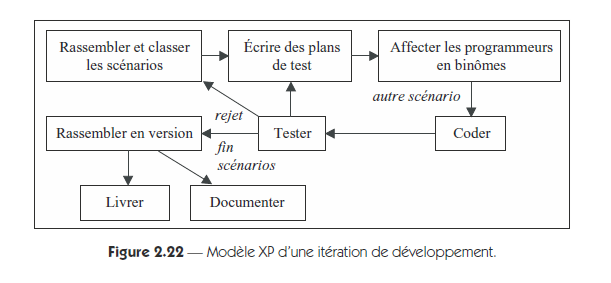
appel, d’une façon ou d’une autre, à un modèle itératif et incrémental.

De plus, elles préconisent en général des durées de cycle de vie des projets ne

dépassant pas un an.

# méthode agile XP = Extreme Programming

Les plans de test sont écrits, les développeurs sont répartis en binôme, ils codent les tâches qui leur sont affectées, puis effectuent avec les utilisateurs des tests d’acceptation.



***Le modèle SCRUM***

Une itération, dont la durée oscille entre une et quatre semaines, est appelée un Sprint.

L’estimation des charges

On mesure la taille des projets à leur charge. Les ordres de grandeur généralement

retenus sont les suivants :

• si la charge est inférieure à 6 mois-personne, c’est un très petit projet ;

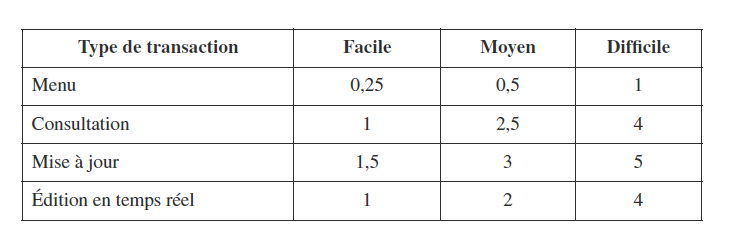
• si la charge est comprise entre 6 et 12 mois-personne, c’est un petit projet ;

• si la charge est comprise entre 12 et 30 mois-personne, c’est un projet moyen ;

• si la charge est comprise entre 30 et 100 mois-personne, c’est un grand projet ;

• si la charge est supérieure à 100 mois-personne, c’est un très grand projet,

souvent mesuré en années-personne.



Les techniques

de planification des délais

L’ESTIMATION DANS LES MÉTHODES AGILES

Elles donnent une place importante aux utilisateurs durant tout le

cycle de vie. S’ils sont autorisés à demander des modifications fonctionnelles

jusque dans les phases aval du projet, ils sont tenus d’accompagner au quotidien

le travail des développeurs par des indications et des évaluations des différents

composants du produit en cours de construction. De ce fait, ils sont parties

prenantes des estimations de charge.

Après une première itération de développement, le nombre de points produits par l’équipe est considéré comme une bonne mesure de sa productivité.

XP parle de *vélocité* de l’équipe. Cette valeur correspond à la capacité de production

de l’équipe à chaque itération, les itérations étant en général d’une durée

fixée.

La vélocité de l’équipe permet de déterminer le travail possible et de faire des

ajustements. En début de chaque itération de développement, l’équipe rassemble

un nombre de scénarios, les évalue et vérifie que leur somme en points correspond

à sa capacité de production, sinon le client doit indiquer ses priorités. Si la

définition d’un scénario excède la vélocité de l’équipe, il faudra le scinder en

plusieurs scénarios de taille réduite.

Ensuite, les scénarios sont éclatés en tâches, qui sont à leur tour estimées et

réparties entre les développeurs de façon à équilibrer leur charge. Si la somme de

ces estimations élémentaires est différente de la valeur retenue en début d’itération, l’équipe se retourne vers le client pour ajouter ou retirer un scénario de l’itération

de développement.

En conclusion, l’estimation des charges apparaît, dans un contexte agile,

avoir un objectif un peu particulier. Bien sûr, comme dans tous les projets, l’estimation

initiale sert à avoir une enveloppe et déterminer un budget. Mais

ensuite, il ne s’agit pas de faire une planification détaillée des différentes tâches

et d’en suivre l’avancement. Les estimations permettent principalement d’une

part de négocier sur des bases très précises avec les clients/utilisateurs pour qu’ils

ajustent leurs exigences ; d’autre part de maintenir un rythme de production

stable dans l’équipe de développement.

L’UTILISATION DE LA PLANIFICATION

Dans un second temps, il s’agit d’établir un *calendrier de travail*. La durée minimum

obtenue précédemment est à rapprocher du délai « normal » ou « raisonnable »

proposé par certaines méthodes d’estimation des charges (paragraphe 3.7). On

utilise ici le diagramme de Gantt. Pour cela, il faut faire une hypothèse sur les

ressources dont on disposera et, parfois, prendre en compte les contraintes de

disponibilité attachées à ces ressources. On peut établir plusieurs scénarios qui

correspondent à différentes hypothèses : les diagrammes aident à décider quel est

le scénario souhaitable (figure 4.1).

LES GRAPHES D’ORDONNANCEMENT

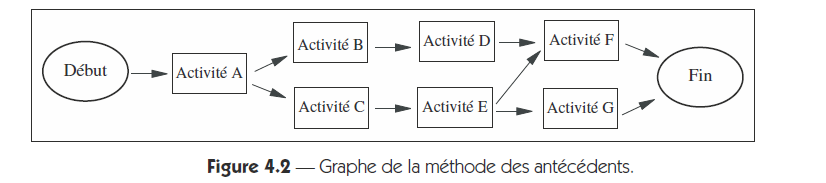
Il existe deux formalismes de représentation de l’ordonnancement des activités :

la méthode des antécédents et la méthode du diagramme fléché.

Dans le graphe de la méthode des *antécédents* (figure 4.2), les activités figurent

sur les rectangles, les flèches ne représentant que les liens. Les ronds représentent

des jalons, c’est-à-dire des activités de durée nulle.



Le formalisme du graphe des antécédents est relativement simple et est retenu par la majorité des progiciels de planification.

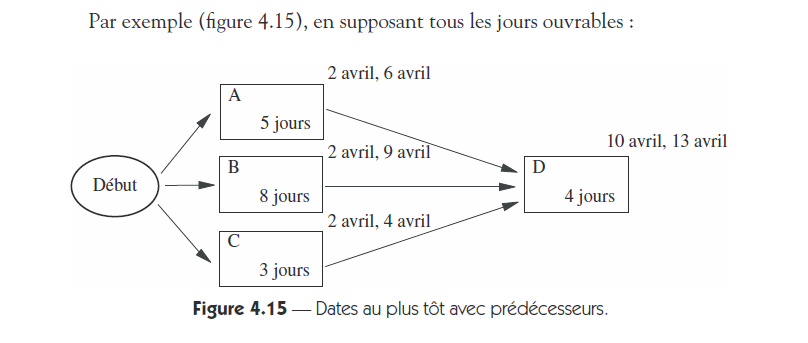
LA MÉTHODE DU CHEMIN CRITIQUE

L’analyse d’un graphe s’effectue avec une méthode appelée la méthode du chemin

critique1, car elle permet de mettre en évidence des chemins qui comportent des

tâches critiques, dans le sens où elles vont retarder la fin du projet si elles sont ellesmêmes

en retard.

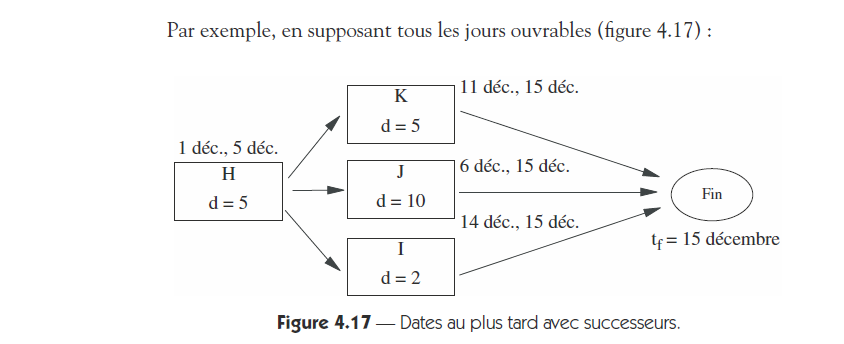


Si la tâche Ti ne se situe pas en fin de projet, elle a des successeurs. Sa date de

fin au plus tard est égale à la plus petite des dates de début au plus tard de tous ses

successeurs moins 1. Sa date de début au plus tard est obtenue en soustrayant la

durée de la tâche et en ajoutant 1.



La *marge* attachée à chaque tâche est la différence entre date au plus tard (Ti)

et date au plus tôt (Ti). En l’absence de liens autres que des liens fin-début, elle

peut être calculée indifféremment sur les dates de début ou sur les dates de fin.

Sinon, on peut avoir deux marges différentes sur une tâche, l’une attachée au

début de la tâche, l’autre attachée à la fin de la tâche. La marge représente la latitude

dont on dispose quand on élabore le planning.

On peut faire des simulations avec différentes dates de fin du projet. La marge

ne doit jamais être négative : sinon, il faut revoir le graphe, en modifiant la tâche

(éclatement de la tâche en deux pour réduire la durée de la tâche et augmenter

le parallélisme), en levant certaines contraintes (découplement de tâches) ou en

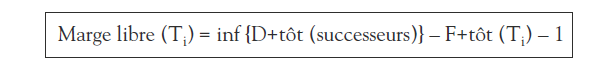
modifiant la date de fin du projet visée.

La marge, telle que définie ci-dessus, est parfois appelée *marge totale*, lorsque

l’on veut l’opposer à une marge plus réduite appelée *marge libre*. On dispose d’une

marge libre *ml* sur Ti si on peut planifier la Ti à la date (D+tôt (Ti) + ml) sans que

cela ait de conséquence sur ses successeurs (c’est-à-dire qu’on peut toujours les planifier au plus tôt).



Le schéma de la figure 4.19 illustre la différence entre marge libre et marge

totale. La tâche A dure deux périodes : sa planification au plus tôt se fait aux

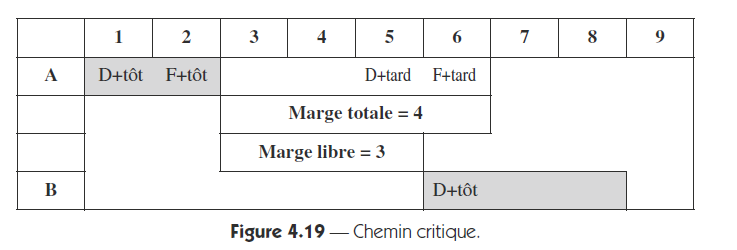
périodes 1 et 2, et sa planification au plus tard aux périodes 5 et 6. La marge

totale de A est donc de 4. La tâche B commence au plus tôt en période 6. Si l’on

planifie la tâche A aux périodes 4 et 5, cela est sans conséquence sur les choix de

la planification pour la tâche B. En revanche, si l’on fait une planification en

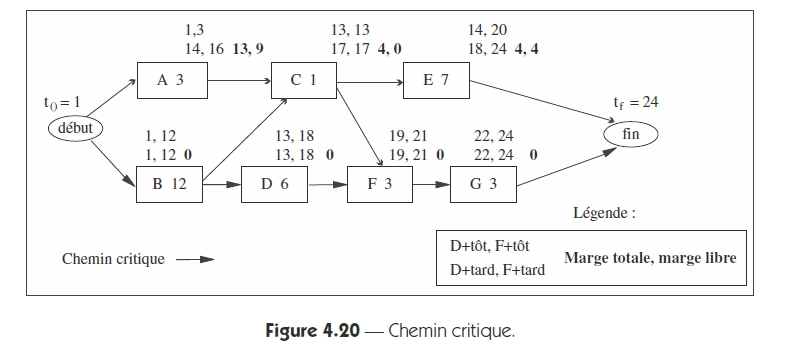
utilisant la marge totale, c’est-à-dire aux périodes 5 et 6, on ne pourra planifier la tâche B qu’à partir de la période 7. La marge libre de A est donc de 3.



Les marges (totale et libre) figurent sur le réseau de la figure 4.20.

Le *chemin critique* est le chemin du graphe sur lequel les marges totales sont nulles. La marge libre de toutes les tâches du chemin critique est donc nulle.

S’il n’y a que des liens fin-début, le chemin critique est le chemin le plus long : il est donc à surveiller tout particulièrement.



S’il y a d’autres liens que des liens fin-début dans un réseau, le chemin critique

peut ne pas être complet, c’est-à-dire ne pas parcourir le graphe du début à la fin.

C’est également le cas si l’on a des contraintes temporelles, c’est-à-dire des

dates imposées (début imposé ou fin imposée) pour certaines tâches. Ce point

sera illustré au chapitre 11 (paragraphe 11.9).

LE DIAGRAMME DE GANTT

Le graphe des antécédents permet de faire apparaître les possibilités de parallélisme

dans l’exécution des tâches et donne les dates de fin du projet possibles en dehors

des contraintes de ressources. Pour passer à un planning (calendrier du projet), il

faut faire des hypothèses de ressources et affecter les tâches à des personnes ou à

des profils de personnes. On pourra faire plusieurs simulations selon la taille de

l’équipe envisagée. On prend également en compte les contraintes de calendrier

(jours non ouvrables, jours fériés…).

On utilise, pour représenter le planning, le diagramme de Gantt, qui se construit de la façon suivante :

• en abscisse, on a l’axe du temps ;

• en ordonnée, on peut avoir soit les tâches, soit les personnes affectées aux

tâches.

Selon que l’on utilise ou non les marges pour effectuer la planification, on

parlera de *planification au plus tôt* ou de *planification au plus tard*.

LA PLANIFICATION OPÉRATIONNELLE

***4.6.1 La prise en compte des contraintes***

Pour établir un diagramme Gantt en vue d’une planification opérationnelle, on doit prendre en compte toutes les contraintes, en commençant par celles qui sont les plus fortes et en terminant par les tâches sur lesquelles on a la latitude la

plus élevée. Souvent, on cherche une planification satisfaisante, qui, souvent,

n’est pas la seule possible.

On distingue plusieurs types de contraintes :

• Les *contraintes de liens entre tâches* ont été mises en évidence sur le graphe

des antécédents, et en particulier le chemin critique. Si l’on veut terminer

dans le délai minimum, on planifie en premier les tâches qui sont sur le

chemin critique.

• Les *contraintes temporelles*, c’est-à-dire les dates imposées pour une ou plusieurs

Tâches

•Les *contraintes liées à la disponibilité des ressources* correspondent soit à une

ressource spécialisée, seule à même d’effectuer certaines des tâches du

projet, soit à une pénurie de ressources.

•Les *contraintes d’exclusion* indiquent que des tâches indépendantes ne

doivent pas être planifiées simultanément, souvent pour des raisons de

sécurité. Ces contraintes sont assez rares dans notre domaine. On peut

donner comme exemple les deux tâches suivantes : Tests volumétriques et

Recette fonctionnelle. Si elles s’effectuent sur la même machine de test, on risque de pénaliser la recette et d’indisposer le client.

***L’utilisation des marges***

Pour limiter les risques de retard provoqué par des aléas, le chef de projet a

souvent tendance à gonfler la durée des tâches, c’est-à-dire à introduire de la

marge cachée un peu partout dans la planification. E. Goldratt1 appelle cette

marge un *tampon (buffer)* et conseille de ne pas répartir les tampons sur toutes les

tâches, mais de le faire judicieusement pour qu’ils contribuent effectivement à la

maîtrise des délais.

Sinon, ces marges seront systématiquement consommées

sans amélioration de la performance du projet.

Il fait en particulier trois recommandations.

D’abord, il est souhaitable de prévoir un tampon en fin de projet pour absorber

les fluctuations du chemin critique.

Ensuite, il est prudent de placer un tampon en fin des tâches qui sont des

prédécesseurs des tâches du chemin critique. En effet, certaines de ces tâches peuvent présenter des risques de dérapages, dont les conséquences toucheront le chemin critique. Si le tampon n’est pas entièrement consommé, cela pourra éventuellement permettre d’anticiper sur le démarrage de la tâche critique.

Le tampon se présente comme une tâche fictive, liée à la tâche qu’elle doit protéger, comme prédécesseur ou comme successeur.

***Le nivellement***

La technique du nivellement consiste à maintenir le nombre de personnes

travaillant simultanément sur le projet en dessous d’une certaine limite. On va

donc, en général, augmenter la durée du projet. Le nivellement vise l’ensemble

des ressources du projet.

Plusieurs raisons peuvent conduire à utiliser cette technique. Le nivellement

évite d’avoir une taille d’équipe de projet trop importante par rapport à la durée

totale du projet. Une première hypothèse de planification qui exploite au maximum

le parallélisme peut conduire à une taille d’équipe risquant de générer des

surcharges de coordination. La disponibilité des ressources (personnes, matériel,

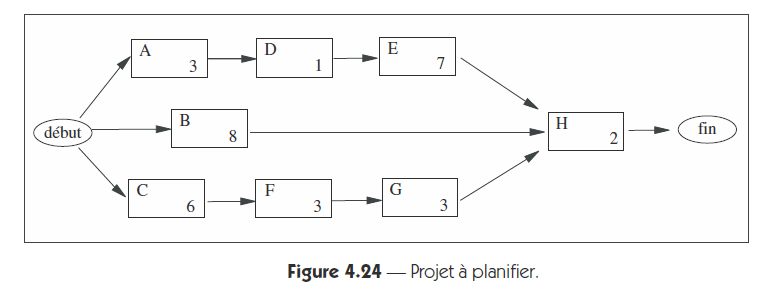
locaux…) peut être telle que l’on doit renoncer à utiliser toutes les possibilités

d’exécuter des tâches en parallèle, telles qu’elles figurent sur le graphe des antécédents.

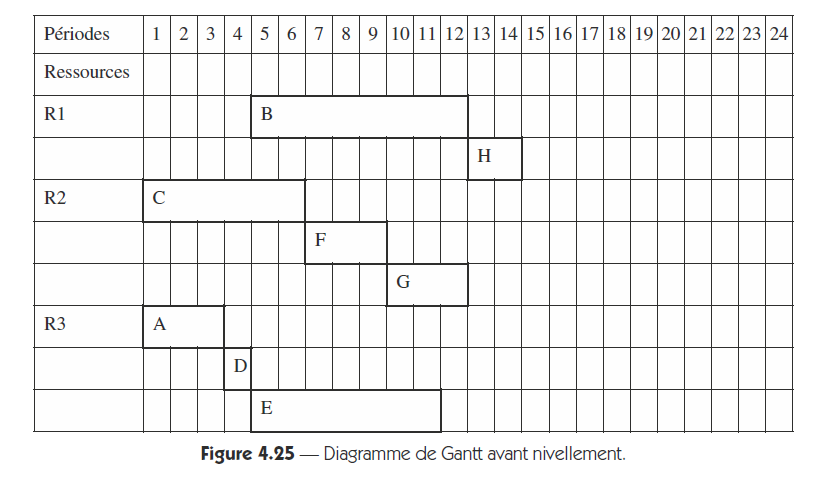
Le nivellement permet enfin d’étaler dans le temps les dépenses liées

au projet.

Soit par exemple le projet structuré en tâches selon le réseau de la figure 4.24.



Le graphe fait apparaître trois chemins en parallèle. On va donc faire une première planification avec trois ressources (figure 4.25).

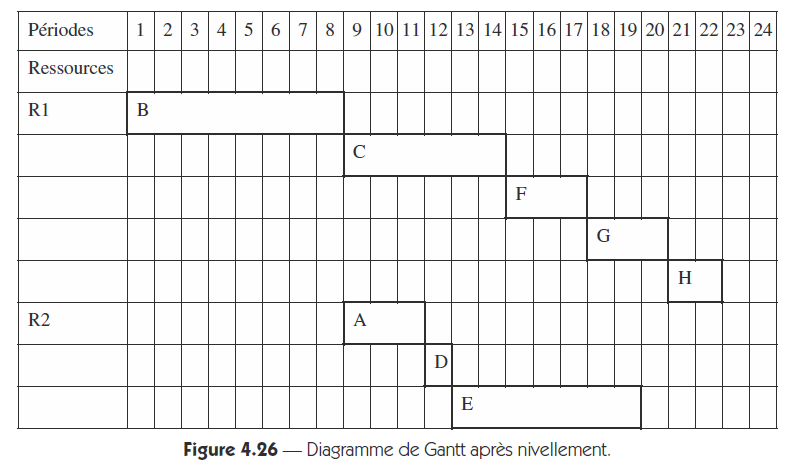


Supposons qu’on veuille limiter la taille de l’équipe à deux personnes : on va

donc niveler le diagramme, en allongeant la durée du projet (figure 4.26). On en

profite pour étaler la montée en charge, en intégrant les deux ressources non pas

simultanément mais successivement : R2 n’arrive sur le projet qu’en fin de période 8, quand R1 achève sa première tâche.



***Le lissage***

La technique du lissage consiste à répartir pour chaque ressource sa charge de

travail, de telle façon qu’elle ne se trouve à aucun moment en surcharge ou en sous-charge.

Une opération de lissage peut conduire à allonger les délais.

Les raisons du lissage sont le plus souvent des contraintes liées à la disponibilité réduite de personnes.

LA PLANIFICATION DES DÉLAIS DANS LES MÉTHODES AGILES

Les méthodes agiles présentent trois principales particularités dans la planification des délais : la négociation des délais, la planification sélective et l’utilisation de la technique de la *timebox*.

La dimension humaine

d’un projet

La MOE est responsable de la conduite du projet.

La MOA assure un suivi de l’avancement du projet.

À chaque fourniture contractuelle par la MOE, la MOA procède à la recette et prononce l’acceptation ou le refus. La MOA pilote la mise en oeuvre du projet.

LE RÔLE DU CHEF DE PROJET

***5.3.1 Définition du chef de projet***

Compte tenu de la définition d’un système d’information, le chef de projet devrait toujours être du côté maître d’ouvrage, avec un chef de projet informatique comme interlocuteur.

***Les responsabilités du chef de projet***

Le chef de projet doit faire en sorte que le projet réussisse. Pour cela, sa responsabilité

est multiple : il a la charge d’une bonne gestion du groupe et

des individus ; il pilote la production des livrables pour un achèvement en temps

et en heure ; il participe à la gestion du changement, en particulier auprès des

acteurs clés du projet ; il lui revient enfin de veiller à ce que les processus de

décision ne soient pas bloqués.

L’équipe prend corps lorsque l’objectif (le projet) devient collectif. Pour obtenir cette adhésion et cette synergie,

il faut que chacun soit — et se sente — responsable en propre d’une partie du travail.

Un retard sur une tâche peut être dû à une cause exogène (panne de la machine de développement) ou organisationnelle (demande de fonctionnalité supplémentaire), qui est du ressort du chef de projet.

Si le chef de projet associe individuellement chacun des membres et recherche une décision satisfaisante pour le plus grand nombre de personnes, on le

nomme style *participatif individuel*.

Le *style directif* donne au chef de projet une place prépondérante dans la prise

de décision : il décide seul de l’organisation du travail, des règles de fonctionnement

et des solutions à apporter aux différents problèmes et aléas. On distingue

parfois deux formes.

Si le chef de projet étudie les problèmes et prend seul les décisions, on parle

du style *directif seul*.

Si le chef de projet recueille des éléments d’information auprès de l’équipe, puis décide seul, il s’agit du style *directif après information*.

Le *style consultatif* consiste à susciter l’avis des membres de l’équipe. Le chef de

projet prend ensuite une décision qui n’est pas forcément le reflet des opinions

du groupe. Comme le précédent, il peut prendre deux formes.

Si le chef de projet recueille les opinions en discutant individuellement avec

chacun, on l’appelle style *consultatif individuel*.

Si le chef de projet recueille les avis de l’équipe lors d’une discussion de

groupe, c’est le style *consultatif groupe*.

Le *style participatif* conduit le chef de projet à associer les membres de l’équipe

aux décisions, notamment sur l’organisation du travail et la gestion des aléas. Cela

demande au chef de projet des capacités d’écoute, de collaboration et de recherche

de consensus. Ce style peut comme les précédents prendre deux formes.

Si le chef de projet associe individuellement chacun des membres et recherche

une décision satisfaisante pour le plus grand nombre de personnes, on le

nomme style *participatif individuel*.

Si le chef de projet réunit son équipe pour une recherche et une évaluation

collective des différentes solutions, on parle de style *participatif groupe*.

Le *style persuasif* vise à augmenter l’adhésion des membres de l’équipe au

projet. Le chef de projet passe un temps important à parler à l’équipe des objectifs

du projet, de son importance, du rôle de chacun et de la pertinence de l’organisation

et des décisions prises au cours du projet. Il suscite des réactions et

cherche à convaincre.

Le *style par délégation* laisse une grande part à la responsabilité personnelle notamment dans la conduite des travaux à mener : le chef de projet fournit les

informations à un membre de l’équipe et lui laisse complète initiative de traiter

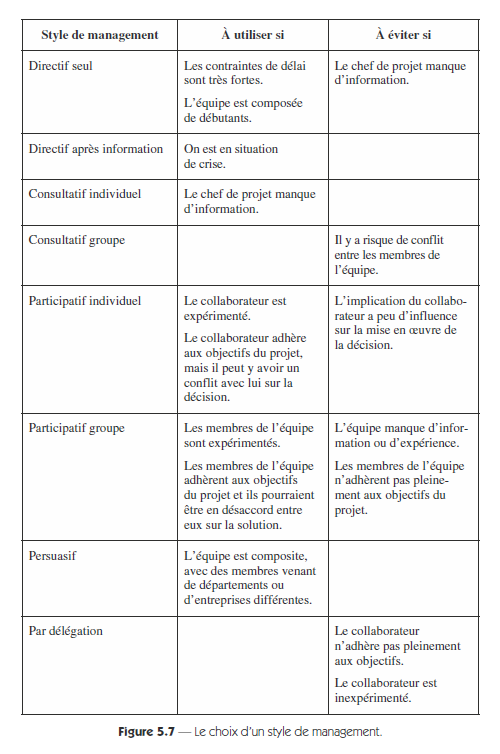
le problème. Ce style nécessite une confiance partagée, ce qui ne signifie pas

aveuglement car il s’accompagne d’un système de suivi précis. Il favorise l’apprentissage et l’autonomie des membres de l’équipe.

LES MÉTHODES AGILES ET LA GESTION DES RISQUES

***Méthodes agiles et profil de risque***

Un projet agile doit durer moins de 18 mois.



***Le management de l’équipe de développeurs***

Les méthodes agiles, en particulier XP, proposent d’organiser le développement en binômes, alliant souvent un développeur confirmé avec un autre moins expérimenté.

***Les rôles dans les méthodes agiles***

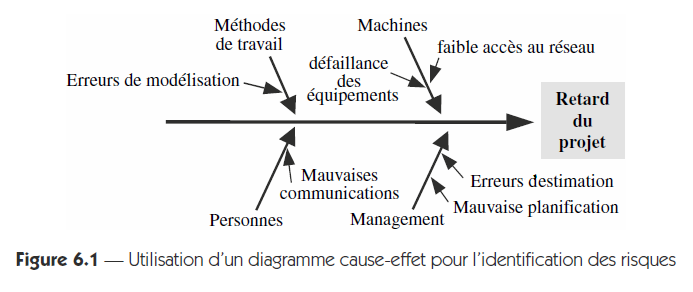
user stories = scénarios du client

SCRUM recommande comme XP une brève réunion quotidienne, appelée « mêlée » (scrum) car elle rassemble le propriétaire, membres de l’équipe et son capitaine. Animée par le scrum master, elle permet de répondre à trois questions : quelle a été la production de la veille ? quelle est la production prévue pour la journée ? qui rencontre un problème ?

L’ANALYSE DES RISQUES

***L’approche généralisée***

On peut éventuellement utiliser un diagramme cause-effet, dit d’Ishikawa, pour rechercher les sources d’un effet redouté (figure 6.1).



LE CONTRÔLE DES RISQUES

***L’audit en cours de projet***

Le Standish Group a proposé, à partir d’une soixantaine d’interviews approfondies

menées avec des responsables informatiques, une grille d’évaluation du

potentiel de réussite d’un projet en cours1. Cette grille comprend dix critères,

pesant chacun d’un certain poids dans la réussite du projet (figure 6.8). Il apparaît ainsi que l’engagement de la maîtrise d’ouvrage, utilisateurs et décideurs, avec une définition claire des besoins pèse pour moitié dans le succès d’un projet.

***Les conditions d’utilisation d’une méthode agile***

Les défis1 que représente le passage à une méthode agile, que l’on peut regrouper en plusieurs catégories sont :

1. La culture de l’entreprise et ses méthodes de management, en particulier

les pratiques de délégation et de collaboration, ainsi que les styles de leadership privilégiés.

1. Les caractéristiques des équipes de développement peuvent être un obstacle

au passage à des approches agile. Le travail en binôme, les tests croisés

entre développeurs ou l’intégration fréquente des composants développés

modifient profondément le quotidien d’un développeur travaillant

habituellement en solitaire. Par ailleurs, un travail efficient entre développeur et utilisateur suppose une confiance de la part du chef de projet, qui traditionnellement est responsable de la plupart des décisions sur le contenu du projet.

1. L’appui systématique sur des procédures entre en partie en conflit avec

l’esprit d’adaptation promu par les méthodes agiles. Plus précisément,

l’approche itérative et la tolérance aux changements de spécifications

requièrent une souplesse, avec laquelle sont peu familiers les acteurs des

projets gérés selon un cycle de vie qui est planifié et piloté de façon plus fine.

1. Même si les approches agiles sont essentiellement des principes et techniques d’organisation, elles présupposent un environnement technique

particulier. D’une part, la décomposition du futur système doit pouvoir

aller jusqu’à un niveau élémentaire de développement, sans générer un

besoin de coordination insoutenable. Ceci est possible avec des technologies

orientées objet, mais difficilement compatible avec des grands

systèmes. D’autre part, le modèle itératif suppose l’utilisation d’outils de développement rapide.

LE TABLEAU DE BORD DU CHEF DE PROJET

La planification détaillée va servir de repère pour suivre l’avancement des travaux. Suivre l’avancement, pour un chef de projet, c’est pouvoir répondre à

n’importe quelle question sur :

• ce qui a été produit : c’est l’avancement réel du projet ;

• ce qui a été consommé : ce sont les ressources utilisées ;

• les écarts entre le planifié et le réalisé ;

• l’origine des écarts, que ce soit une cause ayant des effets sur plusieurs

tâches, par exemple l’indisponibilité d’une machine, ou un problème ponctuel lié à une tâche ou à une personne ;

• ce qu’il reste à faire.

Le dispositif doit inclure un suivi individuel, afin de responsabiliser chacun des membres de l’équipe : la réussite collective passe par l’engagement individuel

dans le projet commun.

Le tableau de bord contient ainsi deux niveaux :

• le *suivi individuel,* qui permet de détecter d’éventuelles difficultés pour un

intervenant ou sur une tâche ;

• le *suivi du projet*, qui sert de base à un point d’avancement périodique avec

le maître d’ouvrage.

***Le suivi individuel***

Il se base sur la *liste des tâches*, qui sont affectées individuellement. Le descriptif de

chaque tâche comprend un identifiant — chaque tâche est repérée de façon unique

— et les éléments ayant servi pour l’évaluation de sa charge (méthode, unités

d’oeuvre, poids standard, ratio…). À chaque tâche, on associe trois types de charge :

• la *charge initiale* : c’est celle de l’estimation qui a servi à faire la planification

détaillée. Cette valeur doit toujours être conservée. Sa modification priverait l’entreprise d’une possibilité d’apprentissage ;

• la *charge affectée* : c’est la personnalisation de la charge initiale en fonction

de l’expérience et de la compétence de celui qui va l’effectuer. Elle peut être supérieure ou inférieure à la charge initiale. Cette valeur représente le

contrat entre le chef de projet et l’acteur concerné. Elle n’est en général

pas visible pour le maître d’ouvrage. Elle doit être utilisée avec précaution ;

• la *charge actualisée* : en cours de déroulement du projet, mais toujours

avant que la tâche ne soit commencée, des éléments nouveaux peuvent

conduire à revoir l’estimation initiale. Par exemple, l’étude technique a

fait apparaître des difficultés imprévues et l’on sait maintenant que la

charge de programmation a vraisemblablement été sous-estimée de 20 %.

On ne modifie pas la charge initiale, mais on négocie avec le maître

d’ouvrage une actualisation. La valeur de la charge actualisée est alors prise en compte pour une nouvelle planification.

La base d’alimentation du tableau de bord est le *compte rendu d’activité*, aussi

appelé *compte rendu d’avancement*, rédigé périodiquement, en général en fin de

semaine, par chaque intervenant affecté au projet. Le compte rendu doit être

régulier. Les chiffres doivent être le plus exacts possible, sinon tous les éléments

calculés du tableau de bord seront faussés. Il comprend, par intervenant et par

tâche :

• le *temps passé* T : c’est la consommation qui sera imputée au projet ;

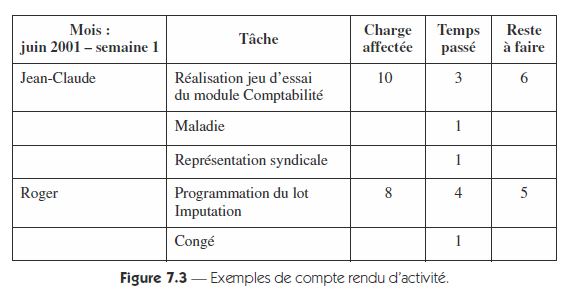
• le *reste à faire* R : c’est l’estimation par l’intervenant du temps nécessaire à

l’achèvement de la tâche. Ce chiffre peut être égal, inférieur ou supérieur

à la différence (charge affectée – temps passé).

Les tâches hors projet figurent également sur le compte rendu d’activité

(figure 7.3).



Le *récapitulatif mensuel* permet un suivi au plus fin. On y trouve pour chaque tâche et chaque semaine du mois :

• le temps passé ;

• le reste à faire ;

• l’*avancement*, calculé comme une différence entre les deux dernières évaluations

du reste à faire :

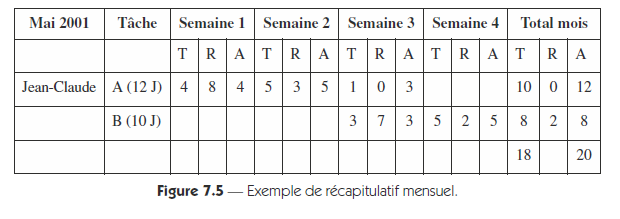
*avancement en fin de période n =*

*reste à faire en fin de période (n – 1) – reste à faire en fin de période n*

Un exemple de récapitulatif est donné à la figure 7.5 pour deux tâches A et B

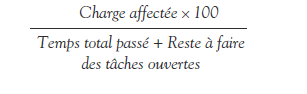
dont la charge affectée est respectivement de 12 et 10 jours. Dans la partie droite

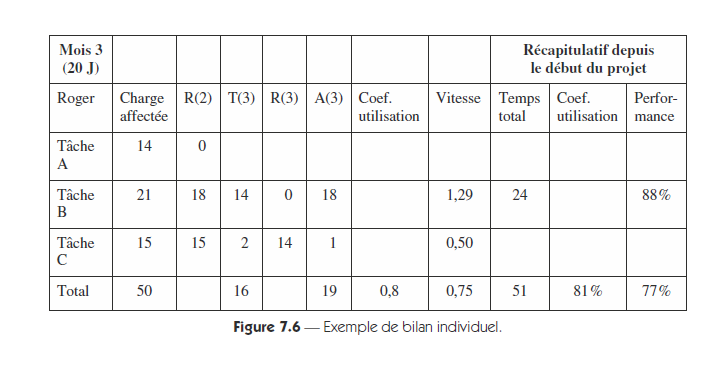
du tableau, on a effectué le total du mois. L’attention du chef de projet est attirée quand l’avancement est inférieur au temps passé.



**bilan individuel mensuel**

La *performance* : ce ratio mesure le degré d’atteinte des objectifs. Il n’a de sens que pour la totalité des tâches en cours ou achevées. On n’y inclut pas les tâches non encore ouvertes. Il compare la charge affectée avec la charge qui a été ou qui sera vraisemblablement consommée :





***Le suivi du projet***

Le chef de projet a besoin d’avoir périodiquement une vue de synthèse de l’état du projet. C’est sur cette synthèse que le maître d’oeuvre, responsable contractuel

du projet, fera le point avec le maître d’ouvrage.

C’est ce qu’on appelle le *tableau d’avancement du projet*. En général, la maille est plus large que pour le suivi individuel : on ne raisonne plus sur des tâches,

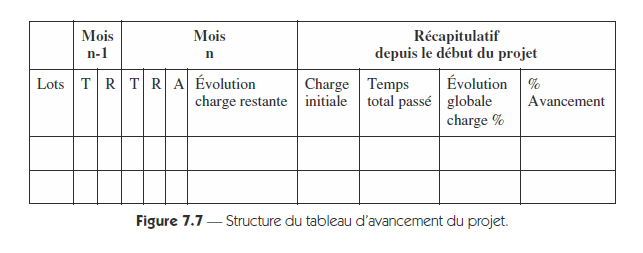
mais sur des *lots* de tâches (figure 7.7). Un lot est un regroupement de tâches

donnant lieu à des livrables qui forment un ensemble cohérent. Un lot correspond

à une ou plusieurs fonctionnalités et peut faire l’objet d’une recette. Les

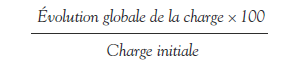
engagements contractuels de livraison portent le plus souvent sur des lots. Un

exemple de répartition en lots est donné au chapitre 10 (paragraphe 10.1.3).



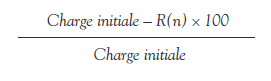
On calcule enfin deux ratios, le *pourcentage d’évolution* qui donne le pourcentage

de l’avance ou du dépassement par rapport à la charge initiale :



et le *pourcentage d’avancement* qui compare l’avancement à la fin du mois n avec

la charge initiale :



LE PILOTAGE D’UN PROJET EN MODE AGILE

***Le suivi d’un projet agile***

La formalisation du suivi n’est pas un aspect préconisé par les méthodes agiles. Il ne faudrait pas en conclure que le suivi est négligé, bien au contraire.

Le suivi individuel est assuré de façon étroite, notamment lors des itérations de développement, par le principe des brèves réunions quotidiennes, préconisées notamment par XP et SCRUM, qui permettent de faire un point sur la progression.

Certains rôles favorisent une proximité avec l’équipe, comme le *tracker*

dans XP, *le scrum master* ou le chef d’équipe dans DSDM.

Selon le Manifeste Agile, l’indicateur d’avancement le plus pertinent est la

mise à disposition de logiciel qui marche. Cette position se justifie dans

la mesure où la structure de découpage du projet a permis d’identifier, à une

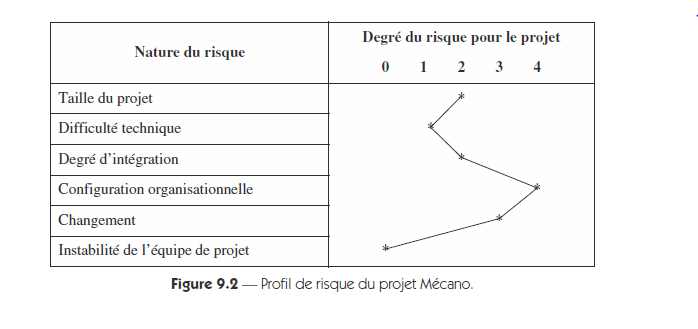
maille fine, des composants pouvant être testés de façon indépendante. Certains

tableaux simples, comme le *burndown chart* de SCRUM qui visualise la diminution du reste à faire, assurent une certaine formalisation de l’avancement.

La maîtrise des risques

9.1 LE CAS MÉCANO

***9.1.4 Profil de risque du projet Mécano***



10.La pratique de l’estimation

des charges

11

L’application des techniques

de planification des délais

12.5 ÉNONCÉ DU BILAN INDIVIDUEL MENSUEL DU PROJET PARKING

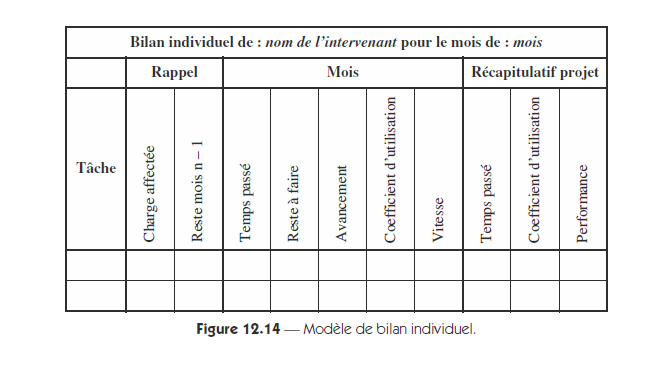
Le bilan individuel mensuel se présentera comme à la figure 12.14 avec :

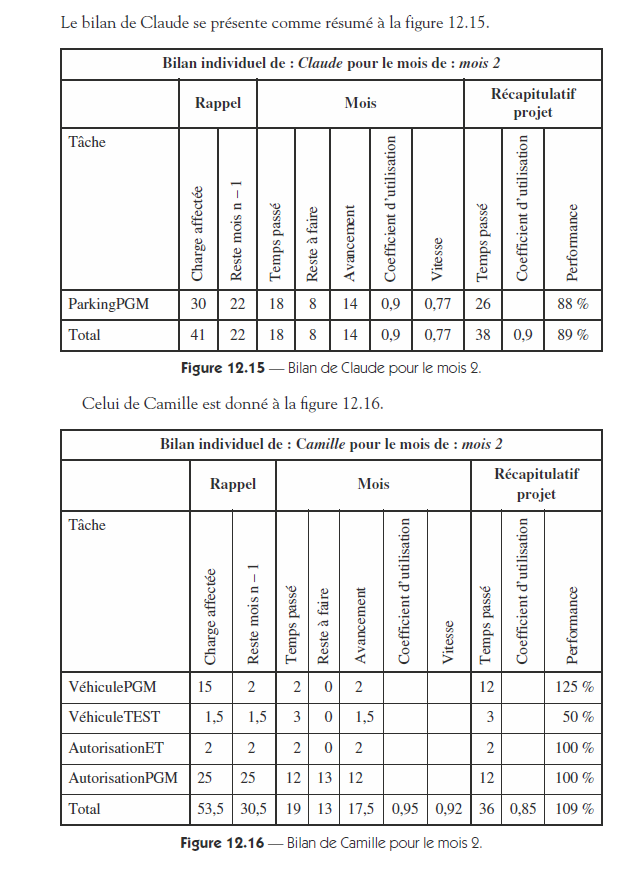
coefficient d’utilisation = temps passé/durée planifiée

(pour le mois ou depuis le début du projet)

vitesse = avancement/temps passé

performance = charge affectée ¥ 100/(temps passé + reste à faire)





Les deux intervenants ne travaillent pas à temps plein sur le projet, comme le

montrent leurs coefficients d’utilisation. Ceci est à surveiller de près.

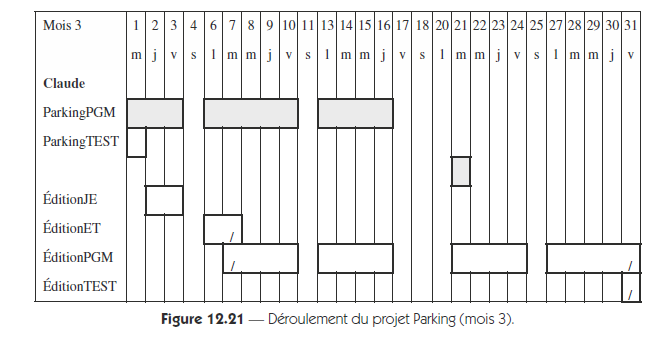
La performance de Camille est satisfaisante, elle équilibre des lenteurs sur

certaines tâches par une rapidité sur d’autres.

La vitesse de Claude sur la programmation Parking doit nous alerter : il faut

faire le point avec lui et voir les difficultés qu’il rencontre.

12.8 CORRIGÉ DE LA GESTION DES ALÉAS DU PROJET PARKING



Où en sommes-nous le mercredi 22 au matin (figure 12.21) ? Claude aurait

dû commencer les tests de Parking le dernier jour du mois 2, le mardi 28. En fait,

il termine la programmation Parking le mardi 21 du mois 3. Par rapport au planifié,

il a 14 jours de retard. Comme il est sur le chemin critique, cela signifie que

si l’on ne fait rien, le projet est en retard de près de trois semaines calendaires.

12.9 ÉNONCÉ DE L’AVANCEMENT DU PROJET PARKING

Le tableau d’avancement du projet sera présenté par lots. Il aura la forme du

tableau de la figure 12.27, avec :

Évolution de charge pour le mois écoulé

= temps passé – avancement

Évolution de charge en jours depuis le début du projet

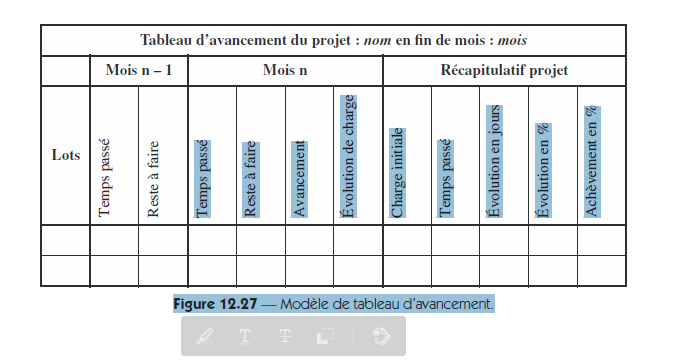
= temps passé + reste à faire à ce jour – charge initiale.

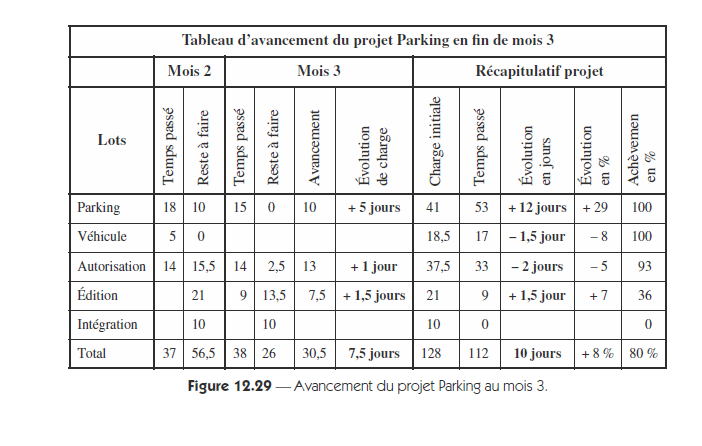
Évolution de charge depuis le début du projet en pourcentage

= évolution de charge en jours ¥ 100/charge initiale.

Achèvement en pourcentage

= (Charge initiale – reste à faire à ce jour ¥ 100/charge initiale





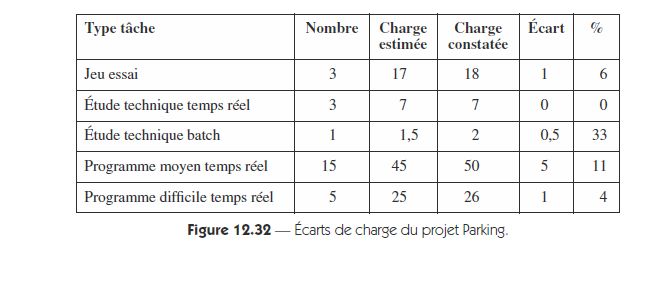
12.12 CORRIGÉ DU BILAN DU PROJET PARKING

Globalement, le projet a consommé 146 jours hors encadrement, soit 14 %

de plus que prévu. Nous allons étudier par type de tâche où s’est créé l’écart

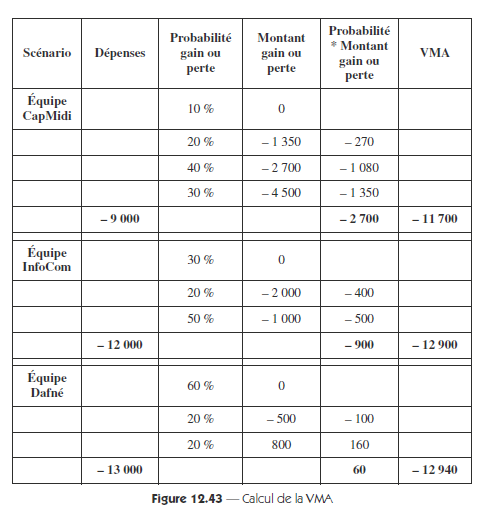
(figure 12.32). Nous reprenons pour cela la typologie qui a servi de base à l’estimation

analytique.



12.16 LA VALEUR MONÉTAIRE ATTENDUE

12.16.2 Corrigé de la VMA



Pour les détails, voir document.

□ commentaires de Nicolas

Un plan de charge n’est pas lié à des données financières.

Ton plan de charge permet de définir les charges nécessaire et le déroulement à mettre en place pour atteindre une date butoire

[18:07, 22/10/2020] Perso Nicolas Rouillé: tu en déduis les coûts en fonction

[18:07, 22/10/2020] Perso Nicolas Rouillé: et généralement, le financier va te dire que tu dois faire avec 2 fois moins de ressources pour tenir le budget

A ajouter dans les plans de charge:

• Mise à jour les scénarios de tests fonctionnels,

• Création des jeux de test,

• Exécution des jeux de test,

• Rédaction spécification technique ou fonctionnelle