



Le projet **DELIRE**
Développement par Equipe
de Livrables Informatiques
et Réalisation Encadrée

PStr8 – On itère



Où en est-on ?

Vous voilà proche de la fin de la phase de spécification et de structuration. Résumons ce qui s'est passé, dans un monde un tantinet idéal.

Vous avez reçu un appel d'offre, sous forme d'un CDC, de la part d'un client. Ce document était rédigé par le MOA (Maitre d'OuvrAge) et vous spécifiait : des besoins à couvrir, potentiellement des exigences, une date limite de livraison et un budget en € alloué pour l'opération.

A partir de votre expérience, vous avez construit un Business plan, pour vérifier que vous aviez la compétence technique, que vous aviez la disponibilité nécessaire, que le nombre d'heures de travail nécessaires était compatible avec la somme allouée par le client. Vous avez donc décidé de vous lancer dans l'opération.

Vous avez donc répondu au cahier des charges par un document de SFG (Spécification Fonctionnelles Générale) qui détaillait la liste des fonctionnalités qui seraient présentes dans le produit que vous vous proposez de développer. Ce développement est prévu en 3 phases : Basic, Standard et Advanced (certaines équipes n'auront prévu que 2 phases, ce n'est pas un problème). Vous ne vous êtes engagé que pour la phase Basic. Le budget alloué par le client peut ne couvrir que la phase Basic ; en ce cas, il faudra négocier par la suite le financement des phases Standard et Advanced. Mais la phase Basic répond néanmoins au cahier des charges. Nous faisons l'hypothèse que vous avez été retenu par le client.

En parallèle, vous avez défini le cycle de vie du projet pour le premier semestre : à quelle date doivent être fini les SFG, les architectures, les STD, la présentation de la soutenance... Vous avez d'autre part réparti les tâches dans l'équipe, et défini le nombre d'heures nécessaire pour chaque tâche. Vous avez commencé (ou auriez dû commencer) un suivi de planning et de budget.

Vous avez attaqué l'architecture fonctionnelle du produit : la liste des commandes élémentaires et l'enchaînement desdites commandes. Pour pouvoir valider que les commandes étaient nécessaires et suffisantes, vous avez écrit des scénarios fonctionnels, qui couvrent tous les cas d'utilisation du produit. Cette architecture fonctionnelle a pris en compte la logique des 3 phases.

Vous avez transcrit ces différentes commandes, grâce à un travail de design, en une maquette de votre site : le dessin des différents écrans. Armés de cette maquette vous avez fait (ou auriez dû faire) une campagne de validation de l'Interface Utilisateur auprès de candidats extérieurs : leurs remarques ont été prises en compte pour améliorer l'ergonomie du produit.

Vous avez défini l'architecture logique de votre produit : la liste des composants, et pour chaque composant :

1. L'objectif du composant
2. Le savoir-faire nécessaire pour développer le composant
3. Les services exposés (publics)
4. Les composants prérequis, pour en utiliser des services exposés.

Ceci définit la frontière des composants : tant que cette frontière, c'est-à-dire tant que cette architecture logique, n'est pas remise en cause, les évolutions du produit ne sont que des modifications locales.



L'architecture logique a pris en compte la logique des 3 phases.
Chaque composant peut maintenant être confié à une personne.

Le responsable de la gestion des risques a élaboré son document PGR qui définit les risques économiques, organisationnels, fonctionnels et technologiques encourus par le projet. Pour chaque risque il a identifié probabilité et impact, et pour chaque risque retenu il a élaboré un plan de levée, préventif ou correctif.

Le responsable qualité a élaboré son PQ :

1. Qualité produit (convivialité et performance essentiellement) sous forme d'objectifs chiffrés et partagés avec le client.
2. Qualité process sous forme de standard et de procédure à respecter par l'équipe projet durant la phase de spécification et de structuration, puis durant la phase de réalisation.

Le chef de projet a élaboré son plan management PM : organisation de l'équipe et du fonctionnement de celle-ci durant la phase de spécification et de structuration, mode fonctionnement nominal et de crise durant la phase de réalisation.

Chaque responsable de composant a pu alors construire les STD (Spécifications Techniques Détaillées) de son (ou ses) composant(s) : l'architecture interne, l'identification de tous les services (ou classes) à développer, l'organigramme de chaque service. Il a défini également chaque test unitaire, qui vise à valider les services développés dans le composant en faisant abstraction des autres composants. Ces STD ont débouché sur une fiche technique du composant.

Le responsable des tests a également élaboré son plan de tests. Il est essentiellement composé de la démarche utilisée pour valider la qualité du produit, de la liste des tests unitaire, et de la liste des scénarios d'intégration et des scénarios fonctionnels.

A partir des STD, des scénarios de tests et d'un peu de bons sens, vous avez pu alors définir l'Organigramme des tâches : la liste complète des tâches élémentaires à accomplir en phase de réalisation puis en phase de convergence au second semestre, leurs dépendances, une estimation crédible du coût de chaque tâche, et le nom du responsable de ladite tâche.

Vous étiez alors armés pour définir le planning et le budget du second semestre, ce que vous avez réalisé. C'est ce que j'appelle l'approche bottom-up du budget : par la sommation des tâches élémentaires.



On itère

Mais en parallèle je rappelle que le MOA vous a attribué un budget (en €, lequel a été convertit en heures de travail). Vous savez donc globalement, maintenant que vous connaissez combien a coûté (ou plutôt va coûter, puisqu'il faudra inclure le coût de la présente itération) le coût de la phase de spécification et de structuration, quel budget est encore disponible pour les phases de réalisation et de convergence.

Vous avez donc la capacité de définir un planning et un budget de macro tâches à l'identique de celui de la phase de spécification et de structuration

1. développement et tests unitaires de chaque composant (en dissociant chacune des 3 phases)
2. intégration et convergence
3. tests fonctionnels et convergence
4. rédaction des livrables
5. préparation de la soutenance et de la démonstration.

C'est ce que j'appelle la démarche top-Down.

Il y a peu de chance que le budget macro-tâches (top-down) et le budget obtenu en sommant les tâches élémentaires (bottom-up) coïncident. Nous allons analyser ici les différents cas possibles. Je travaille sur une hypothèse d'une équipe de 6 ingénieurs, dont le budget de la phase de réalisation est défini comme suit :

1. Codage : 120 heures
2. Tests unitaires et correction des bugs : 60 heures
3. Intégration, tests fonctionnels, et correction des bugs : 60 heures
4. Rédaction des livrables et préparation soutenance : 60 heures.

Je fais l'hypothèse que ce qui ne coïncide pas bien est la phase codage plus test unitaires. A priori, la sommation pour les phases Basic + Standard + Advanced est nettement plus chère que le budget de 180 heures allouées dans l'approche macro tâches.

Commencez par voir combien coutent la sommation pour les phases Basic + Standard. Si le budget est de l'ordre de 180 heures, ceci devient votre nouvel objectif, et vous faites définitivement une croix sur la phase Advanced.

Si le budget reste trop important, voyez combien vous coute la phase Basic. Si le budget est de l'ordre de 180 heures, ceci devient votre nouvel objectif, et vous faites définitivement une croix sur la phase Standard.

Oui, Antoine, et si le budget reste encore supérieur à 180 heures ? Là, je suis plus ennuyé, car la phase Basic a dû normalement être constitué par les fonctionnalités à forte valeur ajoutée pour le client, et à faible coût de développement pour le fournisseur.

Il va falloir enlever des fonctionnalités, ce qui conduit à devoir remettre en cause toute l'architecture, tant fonctionnelle que logique.

Bien sûr, vous avez le droit de penser que le travail va aller plus vite que ce que vous avez prévu. Laissez-moi simplement vous rappeler une nouvelle fois les résultats de l'étude, en 1995, du Standish Group portant sur 365 entreprises et 8 380 projets. Parmi les projets informatiques lancés par ces entreprises :



1. 16 % seulement peuvent être considérés comme des succès (délais, et coûts sont ceux qui ont été prévus)
2. 31 % sont arrêtés en cours de réalisation
3. 53 % aboutissent mais au prix d'un accroissement du délai et du coût tout en offrant moins de fonctionnalités que prévu, (le multiplicateur étant en moyenne de 2,89 pour le coût et de 3,22 pour le délai).

Bref, sachez-le, on a toujours tendance à être trop optimiste dans un projet informatique. Il faut des années d'expérience pour connaître très exactement le temps et le nombre d'heures nécessaires.

En gros, n'attendez pas de miracle.

Faisons l'hypothèse que le budget est globalement respecté. Ceci ne signifie pas que la répartition des tâches se fasse de façon homogène. Vous aviez prévu 60 heures pour la base de données et 20 heures pour l'UI, et la sommation des tâches vous montre que ce sera plutôt l'inverse. Il est temps de modifier la responsabilité de chacun et de prévoir si besoin des développements de compétences supplémentaires.

Quand vous élaborerez et itérerez sur votre planning et votre budget, un petit conseil : nous sommes dans la phase descendante de la courbe de Gauss, c'est donc au départ qu'il faut charger la mule

1. Beaucoup de travail dans les premiers mois, et particulièrement dans la phase de réalisation
2. Plus calme pendant la phase de convergence
3. Vraiment cool lorsque vous allez vous approchez de la date couperet de la soutenance.

Le fait de mettre un coup de collier en début de réalisation présente plusieurs avantages :

1. Dans la réalité, lorsqu'on est en phase d'intégration, la capacité à paralléliser des tâches diminue. Il sera alors trop tard pour mettre un coup de collier
2. S'il y a des erreurs d'architecture, elles apparaîtront rapidement. Il sera certes trop tard pour tout reconcevoir, mais vous aurez encore le temps de limiter les dégâts
3. Le fait de se garder du buffer en fin de projet est une sage précaution.
4. Le fait d'avancer rapidement va vous dynamiser pour la suite du projet, et ce pour une raison basique : vous ne vous serez pas donné le temps de déprimer.

La seule difficulté, c'est qu'il est difficile de se donner un coup de fouet en début de phase de réalisation. Il reste encore tellement de temps...

C'est encore plus vrai dans des grands programmes : comment faire comprendre à un acteur noyé dans un groupe de 10,000 personnes, pour un projet dont la durée de vie est encore de 5 à 10 ans, que prendre du retard dans sa tâche peut obérer les chances du programme.



Gestion des ressources : nivellement et lissage des charges.

La planification par réseau (à partir de l'O.T.) apporte directement la solution au problème des délais, lorsqu'il n'existe pas de limitation de ressources pour exécuter le projet.

Ce contexte idéal ne se rencontre toutefois que rarement.

Dans la plupart des cas les ressources allouées au projet sont limitées et l'ordonnancement doit être réaménagé pour prendre cette contrainte en considération, quitte à consentir parfois un allongement de la durée du projet. C'est le nivellement des charges. C'est votre cas dans le projet DELIRE

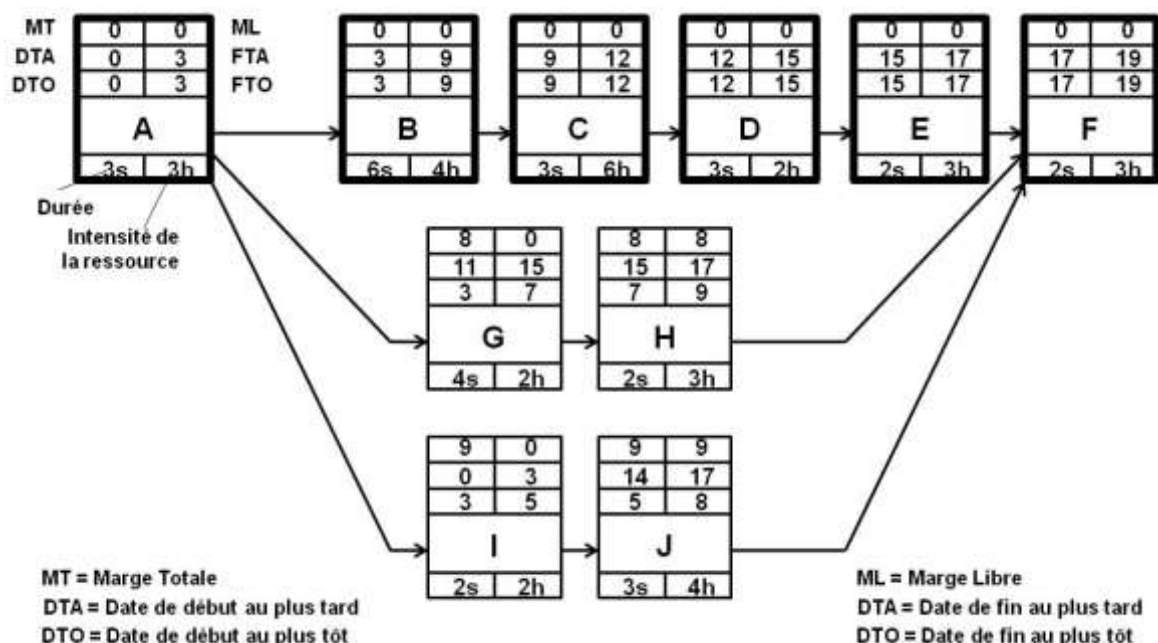
Dans d'autres cas, la contrainte prioritaire provient du délai, qui est fixé au projet avec un caractère impératif. Il faudra bien alors, fournir les ressources en quantité suffisante et arranger l'ordonnancement de telle sorte que cette quantité reste aussi constante que possible, pour éviter les temps morts et les allers et retours onéreux. C'est le lissage des charges. Pas de pot, c'est également votre cas dans le projet DELIRE. Dans la suite je ne traite que le nivellement des charges, mais la démarche serait similaire en lissage de charges.

Plusieurs modes de résolution peuvent être envisagés pour chaque activité

1. Soit déplacer l'activité sans changer ses caractéristiques
2. Soit lorsque la nature de l'activité le permet, changer l'intensité I de sa ressource et donc sa durée D .
 - a. La quantité de travail Q étant constante, on applique la règle de 3 de l'ingénieur $D = Q / I$
 - b. Ainsi une activité prévue avec 2 personnes pendant 3 jours pourrait se réaliser avec 3 personnes pendant 2 jours
3. Soit lorsque l'activité est sécable, la scinder en tronçons qui seront placés à des périodes de sous-emploi.

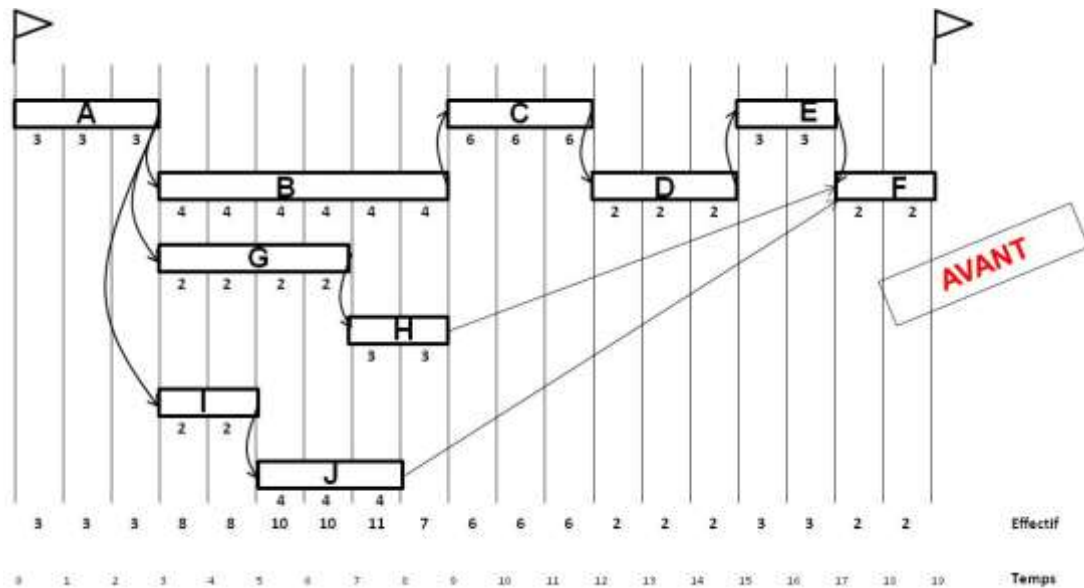
Nivellement des ressources

Une étude de nivellement s'impose lorsque les ressources attribuées sont limitées. Envisageons le cas d'un petit projet à réaliser avec un effectif de 5 personnes de même qualification.

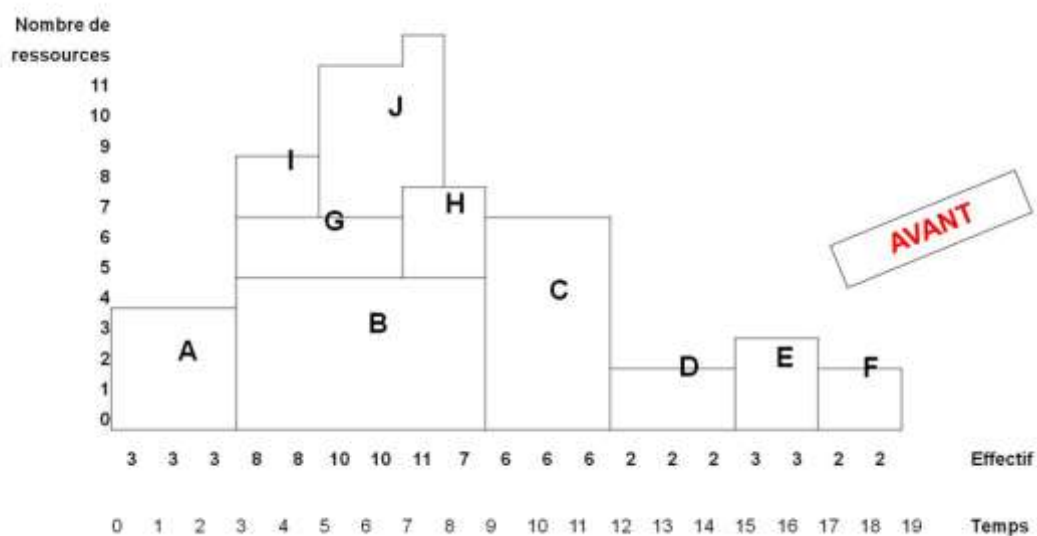


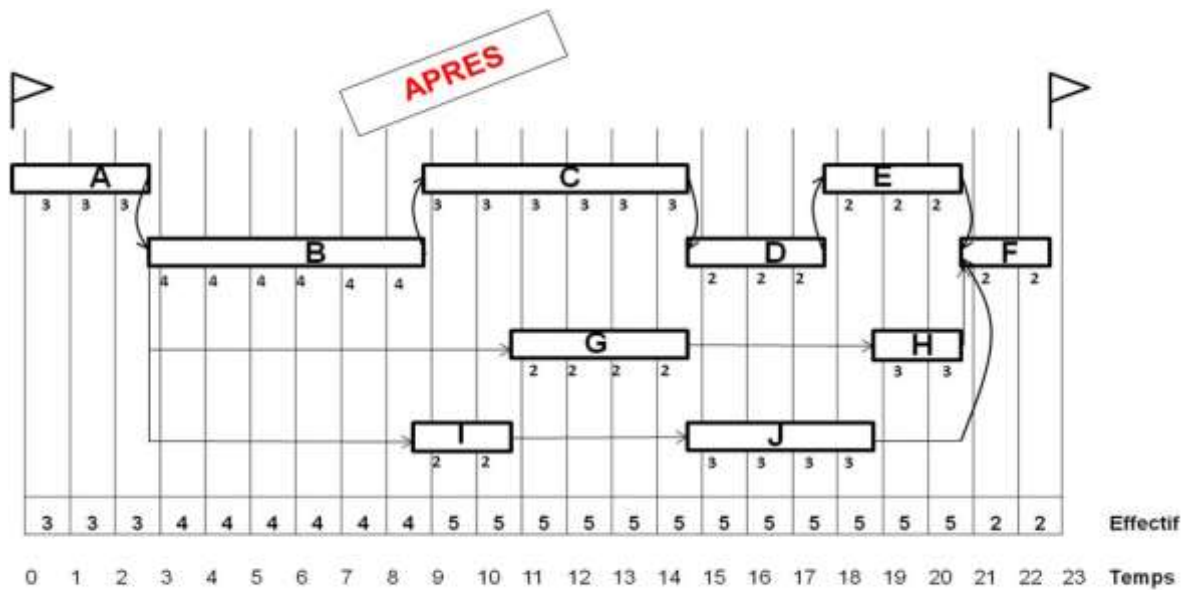
1. Il s'agit là d'un problème de nivellement que nous allons résoudre en appliquant une démarche PERT-CHARGE classique.
2. Le réseau suivant mentionne la logique retenue, la durée exprimée en semaine, les ressources de chaque activité, ainsi que les calculs de date.

A partir de ce réseau, nous établissons le diagramme de Gantt fléché, calé au plus tôt, et dressons les besoins en effectifs

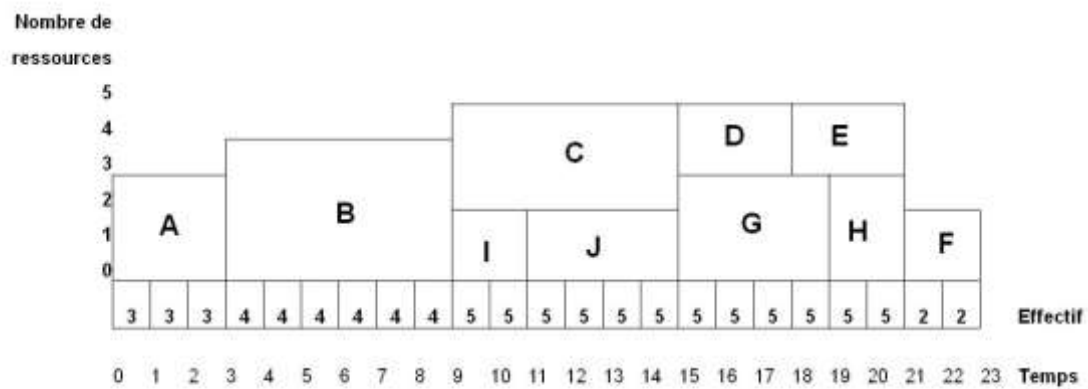


Puis nous établissons le plan de charge de la figure. Il apparaît qu'entre les dates 3 et 12, les effectifs nécessaires sont supérieurs à la disponibilité autorisée.





Planning après nivellement



Plan de charge après nivellement

APRES

Excellent !



A vous. Bonne chance !

