



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

**PSp4 – Les SFG (Spécifications
Fonctionnelles Générales)**



LES SFG (Spécifications Fonctionnelles Générales)

Les SFG sont la réponse du berger (en l'occurrence le fournisseur) à la bergère (en l'occurrence le client).

Le dossier d'expression de besoins et le cahier des charges avaient pour but d'exprimer des besoins : c'est la vue fonctionnelle, le point de vue de l'utilisateur. (Pour des raisons de simplification, le projet DELIRE se limite à l'élaboration d'un CdC)

Le dossier de spécifications (SFG + architecture fonctionnelle + architecture logique) à pour but de concevoir ou spécifier ce qui doit être réalisé pour atteindre l'objectif : c'est la vue technique, le point de vue de l'architecte

L'objectif est la recherche de plusieurs solutions, la comparaison desdites solutions et la détermination de la meilleure solution. Les critères de choix sont évidents :

1. Coût
2. Délais,
3. Technique nécessaires
4. Qualité escomptable,
5. Etude de risques

L'objectif est de converger vers une solution qui satisfasse tout à la fois le MOA et le MOE

Vos deux objectifs principaux, en tant que futur MOE, sont de remporter le marché d'une part, de pouvoir dégager un bénéfice sur l'opération d'autre part.

Il faut donc :

1. Définir suffisamment de fonctionnalités pour avoir une chance d'être retenu dans la short list
2. Ne pas proposer trop de fonctionnalités, ce qui risquerait de vous faire prendre un bouillon.

Dit autrement, ceci signifie que vous n'avez pas obligation à répondre à tous les besoins du client. Certaines demandes peuvent être irréalistes, ou contradictoire. Ce sera à vous de valoriser votre proposition en montrant au futur client que vous avez répondu par une autre approche et que votre produit va même au-delà de son attente.

En règle générale dans un appel d'offres personne ne répond formellement à l'ensemble des points. Chacun essaye de tirer la solution vers son domaine d'expertise : c'est là où il est sûr de gagner le plus d'argent et de se différencier des autres compétiteurs.

La rédaction des SGF

En règle générale les SFG sont rédigées par l'architecte fonctionnel, en relation avec l'architecte de composants: c'est leur métier, et qui plus est le choix des fonctionnalités est fortement corrélé au choix de l'architecture du système. Exprimé autrement, l'élaboration des SFG l'architecture fonctionnelle et l'architecture logique se font conjointement.



Dans le projet DELIRE, je vous propose une approche un peu différente :

1. Vous commencez par définir, à une ou deux personnes, un premier draft des SFG
2. Ensuite, chacun des membres va analyser ces SFG à l'aune de son activité principale
 - Architecture fonctionnelle
 - Architecture logique
 - Gestion des risques
 - Planification
 - Gestion des ressources humaines et du savoir faire

Et en fonction des retours de chacun, on converge vers une seconde (et probablement dernière) rédaction des SFG. Attention, dans cette phase où on élimine, de ne pas rendre le produit trop pauvre et sans intérêt.

Comment trouver la première liste des bonnes fonctionnalités ?

L'erreur serait de répondre par une fonctionnalité pour chaque demande du cahier des charges

Commencez par rédiger des scénarios d'utilisation. Ces scénarios devront bien représenter l'utilisation finale du produit. Une des solutions est de partir des storyboards rédigé par le MOA lors de l'élaboration du CdC.

La validation et la revue de ces scénarios vont permettre d'éviter 2 problèmes en termes de qualité :

1. La non-qualité : des fonctionnalités qui sont nécessaires et qui ne seront pas présentes dans le produit livré.
2. La sur-qualité : des fonctionnalités qui seront présentes dans le produit livré et qui ne seront pas nécessaires.

Lorsque vous rédigerez les scénarios, souvenez vous qu'il existe potentiellement plusieurs type d'utilisateurs : néophyte, utilisateur expérimenté, utilisateur avec un compte, administrateur de données... Il est donc nécessaire de créer au minimum un scénario par utilisateur.

Et la bonne nouvelle, et que lorsque vous aurez à exécuter les tests fonctionnels, vos scénarios seront déjà écrits.

Comment réduire le scope des fonctionnalités ?

La fonctionnalité la plus simple à développer, la plus facile à tester et celle dont on peut le mieux garantir la qualité est la fonctionnalité qui se révèle non nécessaire de développer. Une fois établi la liste des fonctionnalités générales, posez vous la question de savoir si chacune d'elles est vraiment nécessaire

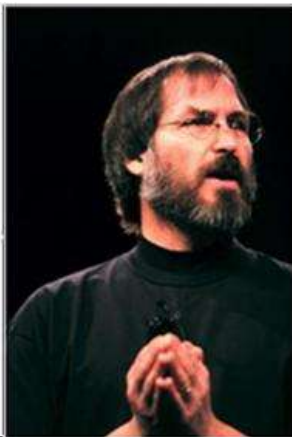
Steve Jobs, l'ancien patron d'APPLE, aujourd'hui décédé, avait plusieurs qualités. Il avait aussi un certain nombre de défauts, comme celui d'être tyrannique et colérique. Parmi ses qualités, j'en retiendrai deux, qui montrent combien pour lui le client était à la source du produit :

1. Au contraire de nombreux patrons, qui truffent leur présentation de références techniques, de tableaux illisibles et de listes indigestes, il avait l'habitude mettre très peu de mots sur chacun de ses slides, et chacun de ces mots était



choisi pour faire rêver le public. Mais je pense que Marie Hélène CAUSSE reviendra sur ce point

2. D'autre part, lorsque ses équipes lui présentaient les fonctionnalités d'un futur produit de la gamme APPLE, la première question qu'il se posait était :
« Quelle est la valeur ajoutée pour le client, est-il possible de la supprimer et en ce cas quel en est l'impact et le risque ? ». Et dans beaucoup de cas, des fonctionnalités se révélaient complexe, en doublon avec d'autres fonctionnalités, voire carrément inutiles. (Oui, Antoine, tu nous l'as déjà dit)



Comme vous le savez sans doute, les produits APPLE ont la réputation d'avoir une remarquable convivialité.

Comment classer par ordre d'importance les fonctionnalités

Souvenons-nous de vos deux objectifs principaux : remporter le marché et dégager un bénéfice. Il est donc nécessaire de prendre tant le point de vue du MOA que celui du MOE pour établir ces priorités.

1. Pour chaque fonctionnalité, estimez la valeur ajoutée pour le client
2. Pour chaque fonctionnalité, estimez également le coût de développement

A l'évidence, les fonctionnalités les plus basiques sont celles qui offrent une valeur ajoutée maximum à l'utilisateur tout en coûtant un minimum en termes de réalisation. Il faudra bien entendu valider que ce qui reste dans la première version du produit est utilisable (c'est à cela que pourront servir les scénarios).

Pour mémoire, vous aurez une démarche similaire à accomplir dans la gestion de risques : pour chaque risque identifié, il faudra vous poser la question de son impact et de sa probabilité.

D'autre part, il peut être judicieux d'estimer la fréquence d'utilisation d'une commande. A l'évidence ce sont les commandes les plus utilisées dont il sera nécessaire d'optimiser tant la convivialité (en gros le nombre de clics) que la performance.

A titre de comparaison, dans la gestion de données techniques, on a coutume de dire que chaque écriture d'une donnée dans la base implique plus de cent lectures : c'est donc la commande browse qu'il est nécessaire d'optimiser. Si besoin, la commande d'écriture est architecturée pour simplifier et booster la commande de lecture.



Prenons un exemple : une question qu'on se pose dans beaucoup de process lorsqu'on accède à une pièce mécanique dans une base de données de CAO est son bloc d'encombrement ($X_{min} - X_{max}$, $Y_{min} - Y_{max}$, $Z_{min} - Z_{max}$). Comme c'est un calcul un peu cher (il faut balayer toute les surfaces de la pièce) ce calcul sera lancé au moment de stocker la géométrie de la pièce en base, et le résultat en sera également stocké. De ce fait, l'accès au bloc d'encombrement sera une simple lecture d'attribut dans la base, et non le chargement de toute la géométrie pour lancer un calcul complexe.

Revenons sur mon ami Steve Jobs.

Depuis la fin des années 90, Les premiers Smartphones (Nokia, Sagem, PALM, RIM, Sony-Ericsson...) ne trouvaient pas leur clientèle

Pourtant Steve Jobs était persuadé qu'il y avait un marché. Mais il y est allé prudemment. Il a estimé qu'avant d'arriver sur le marché, il fallait disposer d'une interface utilisateur intuitive et d'un service d'accès à de la musique. Il a donc commencé par sortir l'iPod en 2001, puis iTunes en 2002. Le premier iPhone n'est arrivé sur le marché qu'en 2007. Avec le succès qu'on lui connaît.



Résultat : APPLE a cannibalisé le marché du Walkman de Sony, est devenu N°1 mondial de la vente de musique en ligne, et le n°1 mondial des Smartphones. Et la convivialité tant de l'iPod que de l'iPhone (Pour iTunes, c'est plus discutable) n'y est pas pour rien. En 10 ans, le marché d'APPLE a prodigieusement évolué, et cette entreprise est devenu la première capitalisation mondiale, à égalité avec EXXON. Bravo Steve !

La dimension contractuelle des SFG

C'est a priori vrai aussi du CdC (Cahier des Charges) : celui-ci est réputé exprimer l'ensemble des besoins du client ; pas de besoins cachés.

Mais dans la réalité, lorsqu'un client exprime ses besoins, il y a souvent des demandes formalisées (les besoins) et des demandes non formalisées (les attentes). C'est d'ailleurs pour cela qu'on fait appel à un MOA, qui est un professionnel du sujet et qui saura faire formaliser au client des points auxquels celui-ci ne pensera pas naturellement.

Sachez faire preuve de logique : aucun client ne vous spécifiera dans le CdC que la cuisine et la salle à manger sont proches et au même étage. Cet « oubli » ne vous autorise pas à les placer n'importe où.

Par contre il peut exister plusieurs choix techniques : n'imposez pas votre préférence, mais sachez présenter les intérêts et les inconvénients de chaque solution, et laissez le MOA faire un choix ou vous donnez carte blanche.

Et même si vous avez le sentiment d'être un peu trop verbeux, ne passez rien sous silence : le MOA pourrait interpréter à son profit ce qui n'a pas été écrit. Enfin, dans la mesure du possible soyez toujours factuel et chiffré.

L'élaboration des SFG, exercice délicat, est la recherche d'un équilibre contradictoire entre les demandes formalisées, les besoins, et les demandes non formalisées, les attentes :

1. Délai, Coût et Qualité : en règle générale, plus c'est bien, plus c'est cher.
2. Contraintes de pérennité : on achète un produit pour une certaine durée.
3. Evolution des technologies : attention à ne pas céder à la mode, mais attention à ne pas non plus s'appuyer sur un composant totalement dépassé.
4. Esthétique ou fonctionnel : ce n'est pas toujours contradictoire (à preuve les produits Apple), mais cela demande beaucoup de concertation entre le designer et l'architecte.
5. Obligations légales : il serait stupide de voir le produit être interdit en fin de projet.

Le MOA peut ne pas être au courant de certaines obligations légales : exemple distance entre deux maisons. Il est de votre responsabilité de MOE de l'en informer

Ce que vous aurez écrit dans les SFG (ainsi que ce que vous aurez défini dans le Plan Qualité produit) vous engage : vous aurez à le réaliser et à le démontrer.

Projetez-vous au jour de la recette : l'objectif est que le MOA signe l'acceptation du produit après avoir exécuté les tests de recette qui auront été définis et négocié entre le MOA et le MOE en début de projet. L'acceptation du produit impose le paiement définitif et démarre la période de garantie (par exemple la garantie décennale dans la construction d'une maison).

Or, vous avez investi dans le projet, et vous avez a priori besoin de rentrée d'argent, vous êtes donc en position de faiblesse pour négocier. Et vous avez toutes les chances, s'il vous manque une fonctionnalité ou si le niveau de qualité n'est pas au rendez-vous, de devoir concéder au MOA une remise de prix ou de prendre des engagements sur des livraisons supplémentaires.

C'est pourquoi j'insiste sur cette notion de 3 phases dans votre projet : basic, standard et advanced. Et dans les SFG, ne vous engagez que sur les fonctionnalités basic pour la première version du produit.



C'est la grande différence entre le développement du logiciel et un produit industriel : on peut le développer de façon incrémental. On peut livrer les fonctionnalités attendues par le client au fil des releases
On imagine mal livrer un avion par tronçons, ou une demi-voiture



Le lancement d'un produit

Que se passe-t-il dans le cas où vous ne répondez pas à un cahier des charges, mais vous décidez, après une étude de marché, de développer à votre propre compte un produit logiciel ?

C'est une situation classique dans l'industrie, en particulier pour les programmes (nouvel avion chez Boeing, nouvelle voiture chez Peugeot). Dans ce cadre, c'est souvent la même entreprise qui va jouer le rôle du MOA, et dans ce cadre piloter l'étude de marché et définir le CDC, et tenir le rôle de MOE, c'est-à-dire piloter le réseau de sous-traitants et assurer l'assemblage final du produit et sa commercialisation. Par contre, si vous disposez d'une expression de besoins, vous ne pourrez pas vous appuyer sur la réponse de plusieurs fournisseurs pour les comparer et sélectionner la meilleure

Un des pièges serait de se lancer bille en tête dans la première solution qui vous vient à l'esprit. C'est afin d'éviter cet écueil que le projet commence par une phase dite d'avant-projet dont le but est d'identifier toutes les solutions techniques acceptables afin de les comparer.

Les critères de comparaison seront on ne peut plus classiques :

1. Délai
2. Coût
3. Qualité
4. Contraintes de pérennité,
5. Evolution des technologies
6. Esthétique
7. Fonctionnel
8. Respect des obligations légales

Bien entendu, nous sommes dans une approche qualité, et ceci doit nécessairement passer par des chiffres : pas question de laisser les gens donner des notes au feeling, en fonction de la solution qu'ils préfèrent.

La probabilité pour qu'une solution se dégage en étant la meilleure sur tous les points est assez faible. Chaque solution sera souvent bonne sur certains aspects, moyennes sur d'autres et faibles pour les derniers.



Le choix de la « bonne » solution, en fait de la moins mauvaise, est de la responsabilité du MOA, en fonction de critères de Coût, Délai et Qualité : il peut, et c'est sans doute conseillé, impliquer le MOE dans le choix.

Une fois ce choix de solution retenu, on entre dans la phase dite de design préliminaire (Preliminary Design) : on travaille sur l'architecture du système (sous-systèmes, équipement et composants).

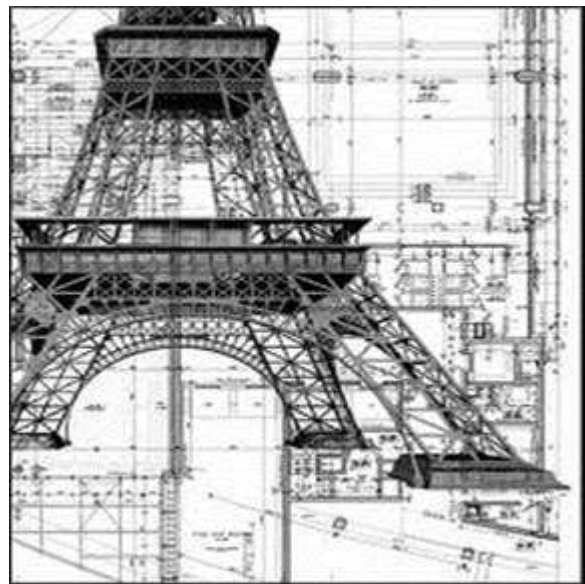
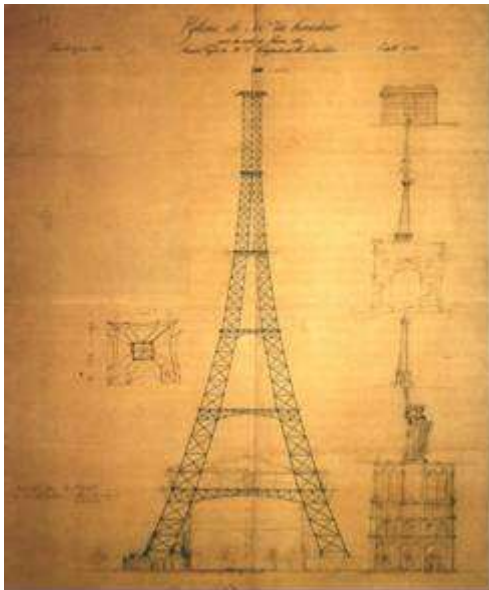
Dans le projet DELIRE, nous n'aurons ni le temps ni la compétence pour nous lancer dans une phase d'avant-projet. Les premières SFG proposées seront les bonnes.

Je l'ai dit, et je le répète, le choix des SFG, puis la définition de l'architecture logique sont les deux activités au travers desquelles se construit la réussite d'un projet. Il y a bien entendu beaucoup d'autres raisons d'échouer (mauvaise gestion des risques, ambiance exécrable dans l'équipe, erreur de planification...) mais de mauvais choix de SFG ou une architecture bricolée vous conduisent nécessairement dans le mur.

C'est dans ces deux activités que le génie de l'ingénieur est le plus sollicité.

Lorsqu'on passera à la phase de design détaillé (Detailed Design) c'est-à-dire quand on commencera à rédiger les STD (Spécifications Techniques Détaillées) , on sera plus dans des tâches d'exécution.

Le saviez-vous ? Avant de se lancer dans la construction de la Tour Eiffel (qui a duré 3 ans), les études et l'architecture ont également duré 3 ans.



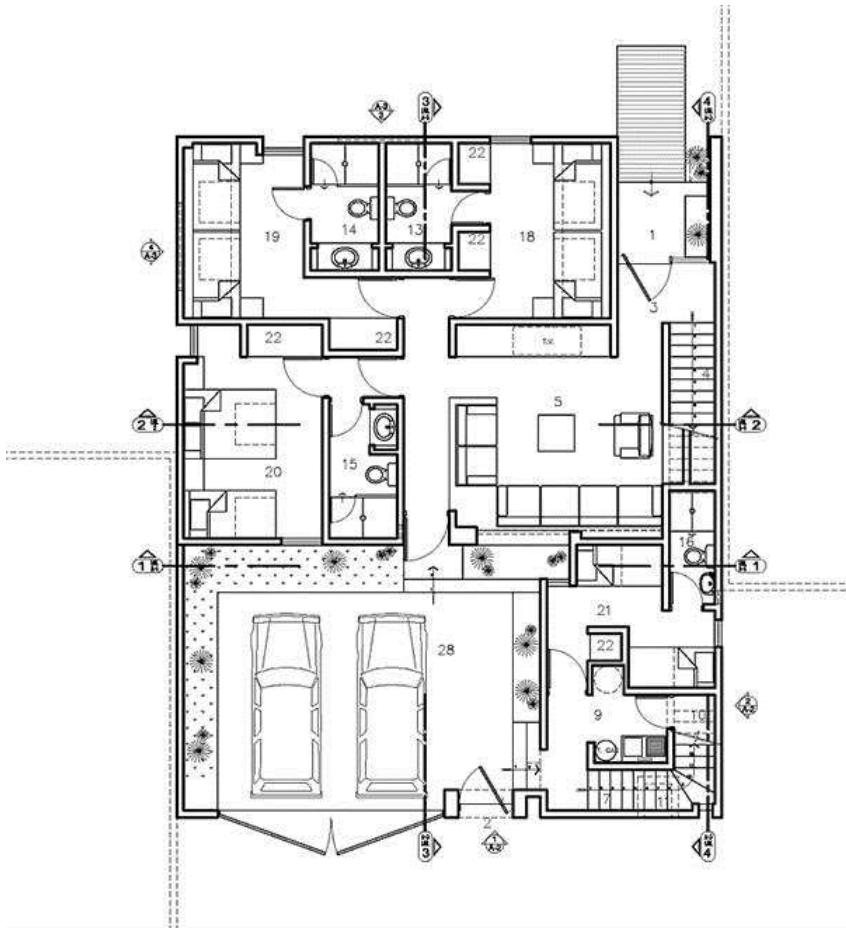
Un rapport de confiance

Je l'ai dit, et je le répète ici, une des clés du succès d'un projet est la relation de confiance qui existe entre le MOA et le MOE. De même que le CdC doit exprimer l'ensemble des besoins, les SFG doivent exprimer l'ensemble des fonctionnalités sans masquer certains problèmes sous-jacents.

Au contraire, en présentant au plus tôt les problèmes inhérents, vous permettez de converger vers une solution qui satisfasse les deux parties.

Faut-il adjoindre le dossier d'architecture aux SFG et le partager avec le MOA. Rappelons tout d'abord qu'on distingue 2 types d'architecture : l'architecture fonctionnelle et l'architecture logique.

Je vous conseille de partager l'architecture fonctionnelle avec le MOA, pour qu'il ait au plus tôt une bonne idée de l'interface du produit et qu'il puisse vous faire d'éventuels retours avant qu'il ne soit trop tard. A l'inverse, je ne vous conseille pas de partager votre architecture logique avec le MOA : elle reflète souvent le savoir-faire de votre entreprise, et ce savoir-faire est un de ses principaux capitaux intangibles.



C'est une des différences avec l'architecture dans la construction d'une maison : dans un même plan on a tant l'architecture fonctionnelle que logique



Mes conseils pour DELIRE

1. Commencez par identifier les rôles que vous allez supporter. Au minimum rédacteur, contrôleur, administrateur de base de données. Mais vous devriez voir apparaître également chef de livre, chef de tome, chef de chapitre.
 2. Pour chaque rôle, identifier clairement les actions à entreprendre.
 3. Une fois que vous avez identifié toutes les fonctionnalités à livrer, triez-les en leur affectant 3 critères :
 - a. La valeur ajoutée de la fonctionnalité pour l'utilisateur : il est prêt à payer très cher pour cela
 - b. La difficulté à la développer: autant commencer par el plus simple.
 - c. La généricité: il est plus intéressant de développer une fonctionnalité qui intéresse tous les rôles qu'une fonctionnalité qui ne concerne qu'un seul acteur.
- Cette démarche vous permettra de séparer les fonctionnalités en Basic, Standard et Advanced
4. Vérifier enfin que la version Basic :
 - a. Vous permet d'exécuter un scénario fonctionnel crédible.
 - b. N'est pas trop ambitieuse

Pour ce faire, je vous conseille de rédiger les scénarios fonctionnels en parallèle des SFG. Ils vous permettront :

1. De vérifier que vous n'avez pas oublié une fonctionnalité majeure
2. Que toutes les fonctionnalités proposées sont utiles

Et vous aurez ces scénarios prêts pour les tests fonctionnels.

Une fois que vous avez ce premier jet de SFG, vous commencez à élaborer Architecture Fonctionnelle et Architecture Logique de composants. Et vous itérez en fonction des difficultés rencontrés. Votre principe de base doit être de simplifier, simplifier et encore simplifier.

