



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

Rôles et Livrables



Le rôle du MOA

Vous êtes le représentant du client, et vous défendez ses intérêts

- Pour que ses besoins soient correctement satisfaits, il faut commencer par les formuler.
- Mais vous devez également vous projeter dans l'avenir.
 - Le produit que vous demandez est un composant d'un système d'information futur
 - Dans ce cadre, vous devez avoir une bonne vue de ce SI à terminaison, ce qui va pouvoir vous permettre de formuler vos contraintes.
 - Aurez-vous une solution propriétaire ou une solution hébergée dans un Data Center, AWS par exemple ?
 - Linux ou Windows pour le serveur ?
 - Scalabilité ?
 - Disponibilité ?
 - Sécurité ?

Vous devez également identifier quel sera le moment opportun pour déployer le produit, et combien vous êtes prêts à mettre sur la table

La réponse du MOE à votre C des C va être

- Les SFG (Spécifications Fonctionnelles Générales)
- Les critères de qualité du produit

Ce sera à vous

- De valider l'adéquation de sa réponse à votre demande.
- De choisir entre plusieurs propositions dans le cas de réponses multiples à un appel d'offres

Le MOE choisi, durant tout le projet, vous devrez être capables de faire des choix entre deux solutions proposées par le MOE.

En fin de projet c'est vous qui direz si le produit est validé ou non, à la suite de l'exécution des tests d'acceptance

Vous êtes responsables :

- De la définition du C des C
- De la définition d'un Business Plan
- De la validation des SFG
- Du suivi de projet, en défendant les intérêts du client
- De la définition et de l'exécution des tests d'acceptance

Quelle méthode comptez-vous déployer pour les réaliser

- Pas d'autre moyen que de rédiger très rapidement le C des C (il devrait d'ores et déjà être terminé)
 - En vous appuyant sur le cahier des charges de Denis Devictor qui a été présenté au premier cours.
 - En définissant un calendrier et un budget
 - En fixant d'éventuelles contraintes techniques. Par exemple, JEE.



- Vous devez ensuite élaborer le Business Plan MOA
 - Identifier le bénéfice espéré
 - Définissez quel pourcentage de ce bénéfice vous souhaitez affecter à la conception du produit.
- Vous devez (ou vous devriez) définir, en collaboration avec le MOE, les tests d'acceptance.

Quelles seront vos difficultés

- Devoir spécifier quelque chose que vous ne maîtrisez pas (très difficile)
 - Ma recommandation est de définir des Use cases
 - Définissez vos priorités : elles vous serviront pour choisir entre plusieurs réponses à votre appel d'offres.
- Définir vos contraintes (difficile)
 - Les contraintes à court terme sont relativement simples à définir
 - Ce qui est plus difficiles c'est de se projeter dans l'(avenir
 - Nombre d'utilisateurs
 - Capacité de stockage
 - Evolutions futures
 - Environnement informatique dans quelques années.
- Devoir définir un délai (facile)
 - La date de déploiement est en général connue un certain temps à l'avance
 - En règle générale, le MOA a été auparavant MOE de projets identiques. Il a donc une bonne idée du coût et du délai que représente ce type de projet
 - C'est donc à lui de lancer l'appel d'offres suffisamment tôt dans le temps.
- Devoir définir un budget (difficile)
 - La vraie difficulté est de bien estimer le bénéfice attendu suite au déploiement du produit
 - Pour un produit utilisé par 12,000 personnes
 - Si le gain de temps annuel est estimé entre 5 minutes et 3 heures,
 - Pour un coût horaire de 20 €, le bénéfice sera compris entre 20,000 € et 720,000 € : ce n'est pas la même chose.



Le rôle du Designer

Votre rôle est de faire un bon produit. Bon... pour l'utilisateur

Un bon produit est

- Complet : il couvre tous les besoins de l'utilisateur
- Simple à prendre en main
- Simple à utiliser

Complet

- Pas d'autres solutions que d'exécuter à la main vos tests fonctionnels, pour valider que :
 - Toutes les fonctionnalités nécessaires sont présentes
 - Seules les fonctionnalités nécessaires sont présentes

Simple à prendre en main

- L'idéal est que le produit ne nécessite ni formation à son usage ni documentation d'utilisation pour les cas nominaux
- Pour ce faire
 - Utiliser des standards de présentation, d'icônes
 - Faites en sorte que les mêmes actions se présentent de la même façon d'une commande à une autre.

Simple à utiliser

- Un seul critère : minimiser le travail de l'utilisateur
- Pour ce faire
 - Minimiser le nombre d'interactions
 - Privilégier les interactions simples et robustes : mieux vaut une sélection dans une liste que de cliquer sur un texte
 - Signaler en temps réel toute erreur de saisie : code SS, n° téléphone...,
 - Maximisez la place utilisée pour les informations utiles

Vous êtes responsable:

- De la définition/validation des SFG, en collaboration avec le MOE. Néanmoins, c'est le MOE qui signera les SFG et qui en assumera les erreurs éventuelles
- De l'architecture fonctionnelle et de la maquette de l'U.I.
- De la validation de l'architecture logique
- De la définition des tests fonctionnels

Quelle méthode comptez-vous déployer pour les réaliser

- Commencez par définir les typologies des utilisateurs (administrateur, médecin, infirmière...)
- Pour chaque profil, définissez vos Use cases. Ils vous permettront
 - D'écrire vos tests fonctionnels
 - D'identifier les commandes nécessaires
 - De valider votre UI et votre Architecture Fonctionnelle
 - Simplicité d'utilisation
 - Toutes les commandes nécessaires sont là
 - Aucune commande n'est inutile



- Faites simple
- Faites simple
- Faites simple
- Et définissez, ou appropriiez-vous, un standard.

Quelles seront vos difficultés

- Choisir entre générique et spécifique (moyennement difficile)
- Définir des priorités dans les SFG (difficile)
- Imaginer un Interface Utilisateur (très difficile)

Principe de base de tout ingénieur : réutilisez.

- Au moment où vous devez définir un UI (User Interface), commencez par identifier le ou les UIs que vous aimez bien.
- Posez-vous la question de pourquoi
- Mettez en évidence les points clé de ce qui vous plaît
 - Graphisme
 - Police de caractères
 - Simplicité des menus
 - Textes qui apparaissent quand on passe la souris sur une icône....
- Inspirez en vous largement pour votre propre UI

Je définirai une autre note de synthèse dans quelques semaines pour vous aider dans votre démarche



Le rôle de l'Architecte

Dans un projet, c'est généralement la phase de réalisation qui est la plus consommatrice d'énergie. Il est donc nécessaire de bien anticiper cette phase.

Un des objectifs de votre rôle est de permettre la distribution du travail.

- Il faut donc définir des composants en nombre suffisant pour partager le travail entre l'ensemble des membres de l'équipe de développement (L'ensemble de votre équipe)
- Il faut également que chacun puisse travailler de façon indépendante. Pour ce faire, chaque développeur doit savoir
 - Quelle composants il peut prérequiser
 - Quels services de ces composants il peut utiliser
 - Comment il peut les utiliser

Un second objectif de votre rôle est de garantir la maintenabilité et l'évolutivité de votre architecture

Les évolutions de votre architecture peuvent être

- Nouvelles fonctionnalités
- Evolution du modèle de données
- Modification de l'implémentation technique : passage d'une première architecture mono VM à une architecture multi-tiers multi instances.

Le résultat de votre Architecture Logique est

- La liste des composants logiciels, avec leur finalité et le développeur en charge de le réaliser
- La topologie des composants : le graphe acyclique (appelés / appelants) des composants
- La signature de chaque service public (au sens de : pouvant être appelé par un autre composant)

Une fois ceci posé, chaque développeur connaît

- La finalité de son composant
- Les services sur lesquels il peut s'appuyer dans les composants appelés
- Les services qu'il doit offrir aux composants appelants.

Un composant c'est

- Fortement cohérent (il en traite pas de domaines disjoints)
- Faiblement couple (si deux composants sont fortement couples, c'est probablement qu'ils auraient du former un même composant).

Question : comment savoir si une architecture est bonne ou non ?

- Difficile, l'architecture de composants est un métier qui s'apprend en se pratiquant.
- Ceci étant posé :
- savoir comment améliorer une architecture est difficile à transmettre et à appréhender,
- Identifier qu'une architecture est enfin aboutie se sent assez bien



Je ne résiste pas à vous présenter un des slides majeurs de la communication utilisés chez Boeing, qui montre que les choix d'architecture (de Design) influence pour 70% le coût final d'un produit.



Il en est de même en informatique.

Vos choix déterminent souvent la réussite ou l'échec du projet

- En termes de budget
 - Si vos interfaces sont incomplètes, chacun passera du temps à rechercher ce qui n'existe pas, et le coût de redesign des composants peut s'avérer coûteux
 - Si vos interfaces sont insuffisamment formalisées, la période d'intégration risque de se révéler délicate. :
- En termes de délai
 - C'est souvent l'incomplétude de l'architecture logique qui est responsable de l'impossibilité de démarrer la phase de Réalisation
- En termes de qualité
 - Certaines architectures sont complexes à réaliser (technologies non maîtrisées par l'équipe)
 - D'autres sont peu robustes : une modification d'un composant entraîne un effondrement en cascade des autres composants.



Vous êtes responsable:

- De l'évaluation de complexité du développement de chaque fonctionnalité, et par ce faire de la validation des SFG et des budgets affichés par le MOE
- De la définition du modèle de données du produit
- De la définition de l'architecture logique : liste des composants logiciels, finalité de chacun d'entre eux, Interfaces des services, liens entre les composants
- De la validation de la complétude de l'architecture logique (en exécutant à la main les scénarios fonctionnels
- Aucun composant et aucun service ne manque
- Chaque composant et chaque service est appelé, à moins que sa présence ne soit justifié par une évolution futur de la solution
- Du choix des composants externes utilisés
- De la validation des versions du produit en vous appuyant sur els priorités définies par le Designer, et en prenant en compte la difficulté de réalisation des composants
- De l'affectation des composants aux membres de l'équipe, en prenant en compte les savoir faire
- De la définition des tests d'intégration

Quelle méthode comptez-vous déployer pour les réaliser

- Pas d'autre moyen que d'attendre la première rédaction des SFG par le Designer et le MOE
- Puis commencer
- Et surtout itérer

Faites simple

Faites simple

Faites simple

Faites preuve de flemme et de procrastination

- Utilisez au maximum des patterns reconnus
- Utilisez au maximum des composants externes
- Identifiez des composants réutilisables au sein de votre architecture

L'architecture est un des métiers les plus complexes de l'informatique

Je définirai une autre note de synthèse dans quelques semaines pour vous aider dans votre démarche



Le rôle du Responsable Planning et Budget

Le planning, c'est le positionnement dans le temps des différentes tâches du projet
Le budget, c'est la gestion des charges (nombre d'heures de travail)

Votre travail dans le projet permet de structurer méthodiquement et progressivement une réalité à venir

Il y a deux périodes dans le projet DELIRE

- Les phases de spécification et de Structuration, où vous devez faire des hypothèses en termes de durées et de charges
- Les phases de Réalisation et de Convergence, où ce sont d'autres personnes qui ont fait les hypothèses en termes de durées et de charges

Dans la suite de cette note, je ne traite que les phases de spécification et de Structuration

Comment structurer méthodiquement et progressivement ? La technique consiste à détailler en découpant chaque budget en sous budget

- Commencez par avoir le budget global du projet
- Tartinez-le sur chaque phase
- Au sein de chaque phase, identifiez les tâches à accomplir
- Tartinez le budget de la phase sur ces différentes tâches
- Pour chacune des tâches
 - Définissez une date de début : c'est au plus tôt la date de fin d'une tâche prérequisite
 - Ainsi
 - La première version des SFG ne peut commencer avant la fin de la première version du C des C
 - La première version de l'A.F. ne peut commencer avant la fin de la première version des SFG.
 - Estimez une date de fin de la tâche. Pour ce faire
 - Estimez le nombre d'heures de travail par semaine pour cette tâche
 - Si il y a une seule personne pour cette tâche
 - Si la personne a annoncé un budget de 5 heures par semaine
 - Si la tâche représente 60 heures de travail
 - Alors la durée est de 12 semaines
 - Et la date de fin est égale à la date de début plus 12 semaines
- Construisez votre Gantt
- Construisez le budget de chaque personne
 - Identifiez les trous
 - Réaffectez la personne en support à une des tâches sous staffées
- Et itérez jusqu'à ce que le Gantt et les différents budgets soient OK.

Pensez à ne pas oublier 2 activités

- La Fièvre du Samedi Soir
- Les réunions



Vous êtes responsable:

- De la définition du planning de la phase de spécification et de structuration
- De la construction de l'O.T.
- De la définition du planning et du budget de la phase de Réalisation
- De la définition du planning et du budget de la phase de Convergence

Quelles seront vos difficultés

- Ne pas avoir oublié une tâche
- Faire le dimensionnement des tâches
- Ordonnancer les tâches
- Répartir les tâches en fonction des savoir-faire, en veillant à ce que tout le monde travaille de façon équitable, et de façon régulière



Rôle du Responsable Qualité

Il y a deux types de qualité dans un projet

- La qualité du produit réalisé : c'est ce qu'on appelle l'efficacité
- La qualité du process suivi : c'est ce qu'on appelle l'efficience

On définit souvent la qualité comme l'adéquation du produit au besoin du client. Dans la réalité, tous les besoins du client ne sont pas formalisés dans le Cahier des Charges. Les fonctionnalités nécessaires sont souvent explicitées, mais certains besoins comme la fiabilité, la convivialité ou les performances sont implicitement considérées comme acquises.

Et le jour où on remet le produit au MOA pour évaluation, il peut y avoir une très mauvaise surprise de sa part.

La solution : exprimer les objectifs de qualité au préalable.

Dans l'industrie, un objectif est toujours chiffré (une valeur) et mesurable (une méthode pour obtenir la valeur)

Ok, mais quelles sont les domaines de la qualité produit.

Pourquoi réinventer ce que d'autres ont créé pour vous ?

Le modèle CUPRIMD d'IBM:

- C : Capability
- U : Usability
- P : Performance
- R : Reliability
- I : Installability
- M : Maintainability
- D : Documentation

Se contenter d'afficher les valeurs et de définir comment on va les mesurer ne suffit pas :

- Il va falloir suivre que lors de l'intégration des différents composants il n'y aura pas de mauvaises surprises.
- Pour ce faire, il va falloir anticiper des mesures par composant durant la phase de réalisation

En tant que Responsable Qualité Produit, vous avez à :

- Travailler avec le designer et l'Architecte Logique pour identifier comment anticiper les risques de non atteinte des objectifs
- Travailler avec le Responsable des Tests pour identifier les mesures à suivre dans le cadre des Unit Tests.
- Et valider que tout cela est intégré au Planning

L'efficience du groupe se mesure à l'aune de deux critères

- La minimisation des tâches à faibles valeur ajoutée
- La capacité des membres du groupe à pouvoir suppléer à la non disponibilité ou à l'overbooking d'un des membres du groupe



Dans les deux cas, la solution passe par la définition de templates qui seront partagés par l'ensemble des membres du groupes

Ces templates ont la finalité

- Soit de simplifier le travail d'intégration. C'est le cas des templates
 - De présentation
 - De présentation d'un membre de l'équipe dans la FSS
 - De livrables
 - De fiche de livraison d ecomposant en phasqe de Réalisation
- Soit de simplifier le travail d'appréhension. C'est le cas des templates
 - De fiche de rapport de tests en phase de Convergence
 - De coding

Quelles seront vos difficultés

- Définir des chiffres réalistes pour les objectifs qualité produit
- Définir des standards qui soient facile à utiliser et à valider pour vos process

Des objectifs qualité produit réalistes. 3 critères sont importants

- Usability (Convivialité)
 - Vous pouvez par eemple définir un nombre d'interactions maximal pour réaliser les tâches les plus courantes
- Performance
 - Avoir un temps de réponse de 1/é seconde ou de 1/10 de seconde ne chane pas la perception de l'utilisateur
 - Par contre passer de 1/2 seconde à une seconde se sent pour l'utilisateur
 - Définissez un temps de réponse standard de 1/2 seconde pour els tâches les plus courantes : afficher un dossier, sauvegarder un document....
- Fiabilité
 - Critère compliqué : un oroduit iIndustriel a un MTBF (Main Time Between Failures) de 200 heures (un mois de travail)
 - Si vous vouliez démontrer que vous atteignez les 200 heures, il vous faudrait à peu près 1000 heures de tests. Pas réaliste
 - Vous pouvez par exemple définir que votre produit a atteint un bon niveau de fiabilité lorsque
 - Tous les tests fonctionnels ont été réalisés sans problème
 - Les tests d'acceptance ont été réalisés sans problème
 - Le scénario de la démonstration finale a été exécutés 3 fois de suite sans problème.



Définir des standards qui soient facile à utiliser et à valider pour vos process

- Il n'est pas possible que vous valideiez vous-mêmes que les standards ont été suivis
- Il est nécessaire que chacun fasse de l'autoévaluation
- Si le fait de certifier que vous avez respecté un standard prend trop d etemps, les membres de l'équipe ne le feront pas .
- Le standrd le plus simple est le dstanddard intégré à l'outil qiu valide en temps réel que vous le repsectez.

Dans le cadre de DELIRE ce n'est pas possible

- Faites des templates simples à utiliser
- Définissez des regles de programmation classiques : il existe des outils qui permettent ensuite de les valider



Rôle du Responsable Gestion des Risques

Comme je le dis régulièrement, « un projet, c'est une suite d'emmerdes ».
On est plus élégant dans l'industrie, on parle de risques

Dans un projet classique, les causes principales d'emmerdes sont engendrées par les innovations.

On pourrait envisager que la solution est : ne pas innover.

Oui mais... dans un monde de mondialisation, vous devez faire mieux et moins cher que la concurrence, et la financiarisation vous conduit à faire ceci avec moins de moyens... Une seule solution... Innover



Dans un projet, on segmente souvent les risques en 4 catégories

- Economiques : évaluer et tenir les prévisions budgétaires
- Organisationnels : équilibrer délais et ressources, actualiser en permanence les estimations de charges
- Fonctionnels : travailler en commun, valider progressivement
- Technologiques : maîtriser la nouveauté

Dans le projet DELIRE, comme beaucoup de tâches que vous devez réaliser sont pour vous nouvelles, vous êtes en permanence en train d'innover.

Vos risques sont

- Economiques : avoir une bonne idée du nombre d'heures nécessaires
- Organisationnels : faire en sorte que le travail soit régulier et équilibré, et que personne ne soit bloqué par la non disponibilité d'un résultat,
- Fonctionnels : être certain que la phase de Réalisation sera courte et productive
- Technologiques : maîtriser la nouveauté (Technique Informatique, méthodologie de Gestion de Projet)



La technique de la Gestion des Risques consiste à :

- Identifier les risques
- Les classer : probabilité, impact
- Choisir ceux qu'on veut traiter
- Définir le plan de levée : préventif ou curatif (on n'applique que si le risque se produit) ‘

Vous pouvez identifier de nombreux risques dans le projet DELIRE. Mon expérience me conduit à vous recommander de privilégier 3 risques majeurs

- Technologie non maîtrisée
- Non disponibilité d'un membre du groupe
- Architecture Logique incomplète et incohérente
- Budget et planning trop optimistes

Technologie non maîtrisée.

- Mon objectif dans DELIRE est de vous faire comprendre les principes de la Gestion de Projet.
- Comme je l'ai déjà dit, cela s'acquiert sur le tas, en faisant de nombreuses erreurs et en analysant le pourquoi de ces erreurs, leur impact sur le projet, et la façon dont il faudra prévenir le risque lors d'un prochain projet
- Mais vous serez néanmoins amenés à faire des développements, et ceux –ci pourront vous conduire à devoir utiliser des techniques que vous ne maîtrisez pas à ce jour.
- La bonne solution consiste à profiter des phases de Specification et de Structuration pour démarrer un prototype et se faire la main.

Non disponibilité d'un membre du groupe

- Un accident de transport ou de ski, une maladie un peu complexe peuvent arriver durant votre projet
- Bien entendu il va falloir travailler avec un groupe restreint pendant un temps plus ou moins long
- Par contre, ce qu'il ne faut pas, c'est que cette non disponibilité entraîne une perte d'information ou de savoir faire
- La non perte d'information se gère en ayant une base de données communes pour tous les travaux et les livrables
- La non perte de savoir-faire consiste à ce que chaque savoir-faire dans le groupe soit doublonné (partagé par 2 personnes)

Architecture Logique incomplète et incohérente

- Comme je l'ai déjà dit, l'Architecture Logique est une des activités les plus délicates dans un projet informatique, comme dans un projet non informatique
- L'architecture Logique est ce qui va permettre de distribuer le travail
- Remettre en cause l'Architecture Logique conduit souvent à redéfinir tout le projet
- La rigueur dans la démarche d'Architecture est de la responsabilité de l'architecte et sera traitée dans une prochaine note de synthèse
- Deux mécanismes vont vous conduire à valider la complétude et la cohérence de votre Architecture Logique



- L'exécution à la main des tests fonctionnels : ce qui suppose d'avoir une première version de l'A.L. ; et les tests fonctionnels rédigés
- La rédaction, en phase de Structuration, des Spécification Techniques Détaillées de chaque composant pour valider que les services nécessaires sont bien présents
- Bref
 - Une première version de l'A.L. dès la fin octobre
 - Et itération sur celle-ci (pouvant d'ailleurs remettre en cause l'A.F.), durant la phase de Structuration.
- Votre objectif est que la phase de Réalisation se passe sans problème.

Budget et Planning trop optimiste

- C'est dans la nature humaine que d'être optimiste
- Bref, votre phase de réalisation va être certainement plus chronophage et plus longue que ce que vous aviez prévu.
- Déraiper vous conduit à
 - Exploder votre budget et ne pas atteindre votre point d'équilibre
 - Décaler votre phase de convergence
 - Tester insuffisamment votre produit et livrer un produit de mauvaise qualité
 - Ne pas investir suffisamment sur la présentation et la démonstration et être pénalisé à la soutenance.
- La solution consiste à
 - Prévoir plusieurs versions pour votre produit
 - Etre modeste dans les objectifs de la première version
 - Et même, dans cette première version, à identifier des tâches qui pourront être abandonnées en cas de dérive du planning.
 - Ce qui suppose qu'elles soient positionnées en fin de phase de réalisation

Je rappelle que le Plan de Gestion des Risques n'est pas un Plan Management.

- Si une personne vient à quitter le groupe suite à un accident de ski, c'est à traiter dans le PGR
- Si une personne vient à quitter le groupe parce qu'elle pète les plombs, c'est à traiter dans le PM.



Rôle du Responsable des Tests

Le rôle du responsable des tests n'est pas d'écrire ou d'exécuter les tests

- Les Unit tests sont définis par les responsables de composants
- Les Tests Fonctionnels sont de la responsabilité du Designer
- Les Tests d'Intégration sont de la responsabilité de l'Architecte
- Les Tests d'acceptance sont de la responsabilité du MOA

Le rôle du responsable des tests est de faire en sorte que ce travail de rédaction et d'exécution soit fait et productif

Il est un peu tôt pour aborder ce point qui ne pourra vraiment démarrer qu'en phase de structuration

Néanmoins, les responsables de Tests peuvent avantageusement commencer à lire le document **PC1 - L'exécution des tests** mis en ligne sur le site de l'UPE

Je mettrai en ligne durant la phase de Structuration une autre note de synthèse pour vous aider dans votre démarche



Rôle du Chef de Projet

Le rôle principal du chef de Projet est de permettre à l'équipe de bien fonctionner

- Ambiance de travail agréable
- Productivité

Le document sur lequel il va s'appuyer est le Plan management

- Il est un peu tôt pour aborder ce document qui suppose une bonne connaissance des particularités de l'équipe.
- Je définirai une autre note de synthèse dans quelques semaines pour vous aider dans votre démarche de rédaction di Plan Management

Par contre, le MOE est responsable d'un autre livrable : les SFG (Spécification Fonctionnelles Générales)

- Bien sûr, vous serez aidés par le designer pour les rédiger
- Mais ultimement c'est vous qui vous engagerez dans ce qui est la réponse de votre équipe au Cahier des Charges

Quelles seront vos difficultés

- La difficulté n'est pas tellement dans l'identification des fonctionnalités : vous pouvez vous inspirer des points du Cahier des Charges ((Pas d'autre moyen que d'attendre la rédaction par le MOA du C des C)
- La vraie difficulté est de tartiner ces différentes fonctionnalités en fonction des versions successives du produit et d'associer à chasque version une date de disponibilité et un budget

La méthodologie pour y arriver

- Demander au Designer de définir des priorités sur les fonctionnalités à offrir : forte valeur ajoutée, faible valeur ajoutée
- Demandez à l'Architecte de définir des niveaux de difficulté pour les fonctionnalités à développer : facile à développer, difficile à développer.
- Commencez par définir 3 versions :
 - 1^{ère} version : les fonctionnalités simples à développer et à forte valeur ajoutée
 - 2^{nde} version : les fonctionnalités compliquées à développer et à forte valeur ajoutée plus, s'il reste de la disponibilité, les fonctionnalités simples à développer mais à faible valeur ajoutée
 - 3^{ème} version : les fonctionnalités compliquées à développer mais à faible valeur ajoutée
- Validez que chaque version a du sens pour l'utilisateur
- Validez également que la version est complète. Ainsi, dans la première version
 - Création de l'arborescence de l'hôpital
 - Création du personnel médical et rattachement à un noud de l'arborescence de l'hôpital.
- Itérez si la version n'a pas de sens, n'est pas complète, ou est visiblement surdimensionnée compte rendu du budget alloué.

Attention, une fois validées par le MOA, les SFG sont contractuelles

