

# **Rapport projet TIW5**

Par Julien GIRAUD 11704709

# 1. Prise de décisions, arbitrage, gestion des conflits

## Qu'est-ce que la prise de décisions ?

La prise de décision a pour objectif de répondre à une question décisionnelle en se basant sur un ensemble de contraintes liées au problème. On distingue en particulier trois types de problèmes en entreprise, techniques, humains et stratégiques.

Les **problèmes techniques** sont liés à des contraintes pragmatiques comme le temps disponible, le coût, l'architecture ou les implémentations. Il faut prendre en compte des notions comme l'interopérabilité, le passage à l'échelle ou les qualifications des équipes. On résout généralement ce type de problème par le biais de recherches, d'expérimentations et d'outils de monitoring.

Les **problèmes humains** sont axés sur la gestion du personnel, du point de vue du recrutement ou des équipes. L'objectif est d'obtenir une meilleure productivité. Les méthodes doivent être adaptées en fonction des besoins et des effectifs, avec des méthodes agiles ou plus classiques comme les cycles en V. Les formations internes permettent également de résoudre des problèmes au niveau des individus.

Les **problèmes stratégiques** sont plus généraux, ils visent aussi pour but d'augmenter la productivité et la réactivité mais peuvent s'appliquer à d'autres domaines. On les traite surtout via des réunions et des méthodes de réflexion comme le brainstorming.

## Les contraintes

La prise de décision est fortement influencée par des contraintes. Il existe des contraintes internes sur lesquelles nous pouvons potentiellement agir, et des contraintes externes que nous sommes généralement obligé de subir. Dans tous les cas, ces contraintes ont un impact sur la prise de décision.

En contrainte interne il y a par exemple le temps de développement d'une fonctionnalité qui dépend des équipes, de leur répartition des tâches et des compétences des individus au sein de ces équipes. Il y a également les choix d'architecture qui dépendent des objectifs et de l'ampleur du projet.

Pour les contraintes externes on peut noter la difficulté à recruter des personnes qualifiées, les deadlines liées aux clients et à leurs contraintes, ou encore les changements d'avis des clients qui peuvent avoir un fort impact avec l'utilisation des méthodes agiles.

## Les méthodes d'arbitrage

Il y a trois axes principaux en ce qui concerne les méthodes d'arbitrage, les statistiques, les recherches et l'expérimentation.

Les statistiques passent généralement par le monitoring ou les indicateurs clés de performance appelés KPI, la recherche utilise plutôt les études ou l'expérience des individus et l'expérimentation se base sur les tests internes ou externes en prenant en compte l'expérience des individus.

## Conclusion

Il existe différents types de prise de décision avec des méthodes qui leur sont propres ou qui peuvent être transversales. Les décisions ont des impacts importants sur les équipes ainsi que sur l'entreprise et elles sont fortement influencées par les contraintes.

## **2. Gestion de la dette technique**

### **Qu'est-ce que la dette technique ?**

La dette technique est le nom donné aux défauts du code qui compliquent l'ajout d'une fonctionnalité ou la maintenance. Elle doit être remboursée ce qui implique des coûts, du temps et des bugs.

### **Sources de la dette**

Une conception logicielle négligée, le manque de temps, de connaissance ou une mauvaise priorisation des tâches sont des sources importantes de dette. L'utilisation d'éléments externes comme des API ou des bibliothèques qui évoluent en parallèle et impliquent aussi l'obsolescence de certaines parties du code.

### **Pourquoi s'en soucier**

La dette technique a des impacts à plusieurs échelles et peut avoir de graves conséquences, c'est pourquoi il est important de la réduire au maximum.

### **Impacts sur les développeurs**

Du point de vue du code la dette implique des difficultés de lecture. Il est alors plus difficile de développer des fonctionnalités, plus facile d'introduire ou de corriger des bugs et cela a un impact sur le moral des développeurs. Il devient plus difficile de se motiver à travailler et changer de poste peut devenir attrayant.

### **Impacts sur les équipes**

La dette pousse à augmenter les écarts entre les membres des équipes ce qui rend plus difficile l'attribution des tâches car seuls certaines personnes connaissent les spécificités du code liées à certaines parties de la dette. Des tensions peuvent apparaître entre certains membres et globalement le moral a tendance à baisser, et avec lui la productivité. Dans certains cas la réorganisation des équipes peut être envisagée pour limiter les effets de la dette.

### **Impacts sur l'entreprise**

Dans le cas des projets internes, la dette implique un ralentissement du développement des applications et la présence de plus de bugs qui se traduit par une perte de croissance et une perte d'argent. L'élaboration des plannings de développement et la réalisation des devis deviennent aussi plus difficile. Pour les projets externes les coûts augmentent pour les clients et les relations peuvent se dégrader.

### **Remboursement de la dette**

Le remboursement de la dette passe par du refactoring voire par la réécriture de toute l'application. Il faut se baser sur ces expériences précédentes, passer du temps lors de la conception et ne pas choisir la facilité si elle n'est pas bénéfique au long terme.

### **Conclusion**

La dette technique est néfaste à différentes échelles et il est important de tout faire pour la réduire au maximum via des méthodes de conception et du remboursement.

### **3. Qualité, adéquation du produit avec la demande**

En informatique, la qualité logicielle est une appréciation qui se base sur des indicateurs permettant de mesurer des éléments comme la performance, la fiabilité ou la tolérance aux pannes. Les deux principaux facteurs qui influent sur la qualité sont les coût et les délais.

#### **Analyse du besoin**

Il est important de savoir à qui s'adresse un logiciel pour comprendre les besoins et définir les critères de qualité. Les besoins sont exprimés par les clients ou les utilisateurs finaux. On peut les retrouver dans le cahier des charges et ils obéissent à des normes comme l'accessibilité, la sécurité ou la confidentialité.

#### **Adéquation du produit**

Afin de valider le produit, le client doit fournir un cahier des charges qui détaille ses attentes.

#### **Qualité fonctionnelle**

La qualité fonctionnelle a pour but de répondre aux besoins du client. Afin de réaliser un produit conforme, on sépare le produit en fonctionnalités afin de les traiter séparément. L'utilisation de diagrammes comme UML est préconisée afin d'exprimer un besoin technique tout en étant compréhensible par le client.

La méthode de gestion dépend du client. S'il est présent et réactif, les méthodes agiles peuvent maximiser le ciblage de ces besoins. S'il est peu présent, une méthode comme le cycle en V sera plus adaptée. Mettre en place un système de tickets est aussi un moyen efficace afin de suivre l'évolution des fonctionnalités, faire remonter des dysfonctionnements, définir les priorités et gérer les échéances.

L'utilisation d'environnements différents de la production, ainsi que la mise en place d'intégration continue et de tests automatisés permet d'assurer plusieurs critères de qualité en amont comme la fiabilité, la tolérance aux pannes et l'idempotence des systèmes.

#### **Qualité technique**

Un code de faible qualité possède une forte dette technique. Elle rend difficile l'ajout de nouvelles fonctionnalités et facilite l'ajout de bugs, c'est pourquoi il est important d'en créer le moins possible.

Dans un premier temps les outils de versionning comme GIT, la ré-utilisabilité des composants, les commentaires dans le code ou encore les revues de code sont des moyens efficaces pour lutter contre cette dette. Ensuite de nombreux problèmes en informatique sont connus et peuvent être résolus efficacement par les design patterns. Enfin il est possible d'utiliser des outils d'analyse de code qui couplés avec les tests unitaires qui permettent de mesurer la couverture en tests du projet. Ce type d'outil peut être utilisé pour empêcher des mises en production si l'application ne respecte pas suffisamment les critères de qualité.

#### **Conclusion**

Afin de garantir la qualité d'un produit il faut comprendre les besoins du client et définir les indicateurs à utiliser. La qualité du code ne doit pas être négligée et de nombreux outils permettent de l'améliorer. Enfin la mise en place de tests est incontournable afin d'assurer la conformité avec les indicateurs définis.

## **4. Conception, propositions commerciales, chiffrage**

Les développeurs n'ont pas forcément conscience de l'avant projet et du travail des commerciaux mais il s'agit des premières étapes qui mène au développement d'un produit.

### **Proposition commerciale**

Dans un premier temps un commercial prend contact avec un potentiel client, par exemple rencontré lors d'un salon commercial ou grâce à des prospects. Le but de cet échange est de comprendre les besoins et discuter des possibilités de produit que l'entreprise peut produire. L'objectif du commercial est de se démarquer des entreprises qui ne proposent qu'un devis basé sur une grille tarifaire, il personnalise l'offre selon les besoins et le budget du client. Cette démarche favorise les relations, en particulier lorsque les entreprises sont dans le même secteur d'activité.

### **Chiffrage**

Le montant de la proposition est l'information la plus importante de la proposition commerciale. Il s'agit d'une estimation datée du coup humain et matériel permettant de réaliser un produit. Ce chiffrage prend en compte un maximum d'informations sur le client ce qui permet d'augmenter la crédibilité de l'entreprise. Il est important d'être relativement honnête, d'indiquer ce que l'entreprise est capable de produire et de toujours utiliser la même méthode de chiffrage pour rester cohérent.

Plus l'entreprise a d'expérience plus ses estimations seront correctes. Il est possible d'effectuer des chiffrages tout au long du projet que ce soit en jour-homme ou en fonctionnalité et toute l'équipe peut y participer. Il est important de bien identifier les différents corps de métier qui entrent en jeu et il existe des méthodes qui permettent d'obtenir une estimation du chiffrage. Par exemple la méthode PERT qui est assez efficace se base sur l'estimation optimiste, la probable pondérée et la pessimiste.

### **Conception**

La conception est la première phase dans le cycle de vie d'une initiative. Son but est de définir les objectifs et l'organisation. Il faut identifier les risques et les contraintes pour éviter les erreurs de conception irréversibles, bien organiser la répartition des tâches, mettre en place un système de suivi comme GANT et prévoir des solutions en cas de problème. C'est lors de cette conception qu'on identifie les priorités et les modalités de retard de livraison. Dans le cas des méthodes agiles il y a une nouvelle phase de conception après chaque sprint ce qui peut mener à l'évolution du cahier des charges.

### **Conclusion**

La proposition commerciale et la phase de conception sont des étapes incontournables dans la réalisation d'un produit même si elles peuvent être invisibles aux yeux des développeurs. Ces étapes ont pour but d'exprimer les besoins du client et de s'adapter à lui avant de commencer un projet.

## **5. Gestion des coûts et des délais pendant la durée de vie du projet**

L'objectif de la gestion des coûts et des délais est d'évaluer le coût des technologies, des personnels, des sous-traitants et organiser la répartition du travail en fonction des tâches et des compétences du personnel en ce qui concerne les outils et technologies.

### **Gestion des coûts**

Les coûts sont les sommes engagées pour le projet, un budget global est défini mais tous les fonds ne sont généralement pas disponibles dès le début du projet. Il faut donc prévoir le système de paiement en plusieurs fois afin d'assurer la rémunération du service sans trop pénaliser la gestion du budget du client.

Le calcul du coût prend en compte le prix de chaque cœur de métier, ce qui a été déjà été payé ou qui va l'être prochainement et doit être ajusté au budget du client. Des systèmes de suivi de paiement existent afin de répartir au mieux le budget sur les périodes de développement. Dans le cas des méthodes agiles il est facile d'effectuer cette répartition en définissant une somme à verser pour chaque phases de développement, ainsi on paye pour chaque sprint jusqu'à ce que le produit soit terminé.

### **Gestion des délais**

La gestion des délais commence par une phase de planification. Il existe des outils qui facilitent cette phase comme Microsoft Project qui permet de réaliser des diagrammes de GANTT facilement. Il est important de savoir estimer le temps des différentes tâches, leur importance ainsi que l'ordre dans lequel elles doivent être réalisées. De cette façon il est plus facile d'anticiper la répartition des tâches en fonction des compétences du personnel et une bonne visibilité de la disponibilité des équipes permet d'être plus réactif en cas d'imprévus. Ainsi, une bonne analyse permet de minimiser les risques liés aux délais.

Les délais peuvent être plus ou moins souples en fonction des contraintes du projet mais il s'agit généralement de la plus forte contrainte une fois le budget défini. La mise en place de systèmes de tickets avec priorité et estimation du temps est un bon moyen pour gérer les tâches imprévus tout en respectant les délais.

### **Conclusion**

Il existe de nombreuses façon de calculer les coûts et planifier un projet, certaines se démarquent mais chaque entreprise à tendance à utiliser sa méthode personnelle. Il est bénéfique de passer du temps avec le client sur l'analyse et réfléchir à des points d'améliorations de ses méthodes mais les délais ne le permettent pas forcément. Différents outils existent pour faciliter ces étapes avec du monitoring et il est conseillé de les utiliser. Enfin la gestion des coûts et des délais évolue tout au long du projet et il est préférable d'itérer ce processus plusieurs fois au fur et à mesure de l'évolution du produit.

## 6. Agilité

Les méthodes agiles sont tirées du manifeste agile de 2001 qui estime que les méthodes traditionnelles ne répondent plus aux besoins actuels qui évoluent en permanence. Ce type de méthodes s'adaptent plus aux individus qu'aux contrats.

SCRUM est une implémentation de ce manifeste avec un système d'itérations de 2 à 4 semaines appelées Sprint où le client final, le Product Owner, est en étroite collaboration avec l'équipe de développement.

### Analyse des besoins et planifications

Dans le cadre des cycles en V tout est planifié en amont, à l'inverse en méthode agile la planification n'est effectuée que pour la prochaine itération. Les fonctionnalités sont listées sous forme de parcours client appelés User Stories et les étapes de développement sont définies en fonction de celles-ci. Tous les parties participent aux itérations, y compris le Product Owner qui attribue les User Stories à réaliser pour le prochain Sprint.

Il est possible d'effectuer des micro cycles en V de quelques mois pour se rapprocher des méthodes agiles tout en restant sur une méthode de gestion traditionnelle.

### Déroulement d'un Sprint

Les User Stories deviennent de plus en plus précises avec l'avancement du projet et elles se voient affecter des tâches. Tous les jours les membres de l'équipe se réunissent pour expliquer ce qu'ils ont fait la veille, ce qu'ils vont faire le jour même et parler des difficultés qu'ils ont rencontrées. Lors de ce moment les membres de l'équipe peuvent partager leur expérience et réfléchir ensemble à des solutions pour surmonter les difficultés.

En théorie l'équipe doit quotidiennement mettre à jour le Sprint Burndown, un outil de visualisation qui permet de connaître l'avancement de chacun par rapport au planning. Cet outil permet de voir si l'équipe est en avance ou en retard par rapport aux prévisions. En pratique il n'est pas souvent utilisé.

### Fin de Sprint

À la fin de chaque Sprint l'équipe met en production l'application et fait une réunion avec le Product Owner. Cette réunion permet d'indiquer ce qui n'a pas eu le temps d'être développé et de faire la démonstration de ce qui a été réalisé lors de la dernière itération. Ensuite l'équipe se réunit pour faire un point sur le dernier Sprint. Dans le cadre de cette réunion elle liste ce qui s'est bien passé, ce qui s'est mal passé et ce qui peut être amélioré. Le but est de faire une synthèse de ce qu'il faut continuer et de ce qu'il ne faut plus faire.

### Conclusion

Les méthodes agiles ont émergé pour combler les lacunes des méthodes traditionnelles qui ne répondent plus aux besoins actuels. Ces méthodes sont faites pour s'adapter à l'évolution permanente des besoins et impliquent beaucoup plus le client au sein des projets.