

Chapitre 3

Définir un périmètre de tests : la stratégie

1. Périmètre fonctionnel

Le présent chapitre s'attachera à définir les notions de périmètre de tests et de stratégie de recette, que celle-ci se déroule directement en sortie de développement lors d'une recette technique ou plus en aval au niveau d'une recette fonctionnelle, toujours dans le cadre d'un chantier de release.

Nous aborderons ces notions en partant d'un sujet plus large que celui du test : l'analyse de risques. Cette démarche doit normalement être menée par l'équipe MOA en charge des études avant-projet et cela, bien avant le démarrage de celui-ci, comme nous l'avions précisé dans le chapitre Généralités qui exposait des généralités sur le processus de production logicielle.

Mais cette tâche n'est pas obligatoire : l'analyse de risques n'est pas toujours réalisée. Malheureusement...

1.1 Parties prenantes, risques et exigences

1.1.1 Lister les parties prenantes

Par partie prenante d'un projet, nous entendons une liste des types d'acteurs du projet et/ou de la solution logicielle produite.

Nous rencontrons donc des parties prenantes spécifiques à un projet (les développeurs, les chefs de projets MOE et MOA, les équipes de recette...) et des parties prenantes concernées par le produit réalisé, les types d'utilisateurs (par exemple, un administrateur, un contributeur et un rédacteur en chef pour une application de gestion de contenus).

Une partie prenante peut d'ailleurs être concernée avant, pendant et après le projet suivant son niveau d'implication. Elle peut être dans l'entreprise qui réalise le projet, comme elle peut être un anonyme à l'autre bout de la planète.

Quoi qu'il en soit, cette étape permet d'identifier toutes les typologies de personnes interagissant à un moment ou un autre avec le projet et/ou le produit.

1.1.2 Lister les risques

Une fois les parties prenantes identifiées et réparties selon qu'elles sont impliquées avec le projet ou le produit, s'en suit une phase pendant laquelle le responsable de l'étude va identifier tous les risques dans la mesure du possible.

Deux familles de risques sont distinguées :

- Les risques sur projet, qui concernent les parties prenantes impliquées dans le processus de production logicielle.
- Les risques sur produit, qui concernent les parties prenantes qui interagiront avec le logiciel dès lors que celui-ci leur sera accessible en environnement de production. Ces risques-là n'existent donc pas pendant le projet, mais apparaissent à son issue.

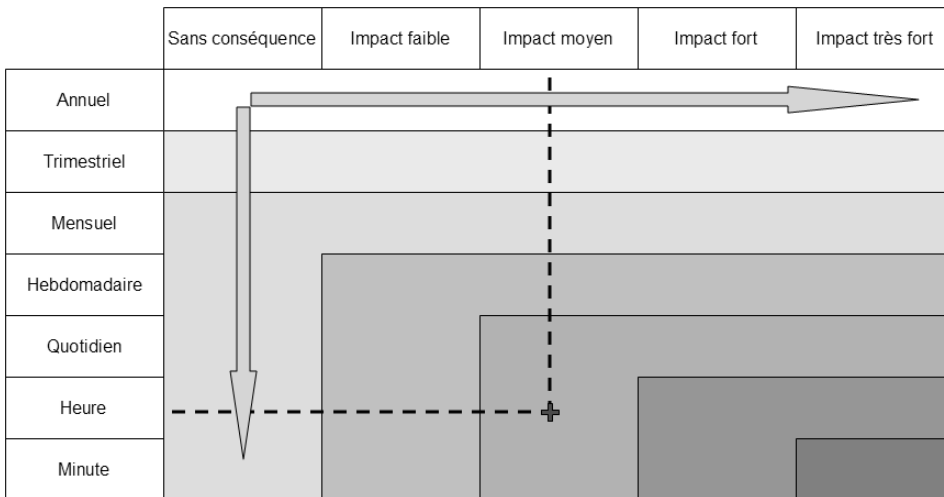
■ Remarque

Il est très important de ne pas confondre ces deux notions et en général d'éviter de faire une confusion entre le projet et le produit qui est son résultat.

Cette phase d'identification constitue la première étape d'une analyse de risques. Ces derniers sont ensuite catégorisés : ils peuvent être techniques, financiers, organisationnels, contextuels, juridiques, sociaux...

S'en suit alors une valorisation des risques amenant à les classer selon une matrice de risques :

- Selon un axe de probabilité de survenance (chaque minute, chaque heure, quotidienne, hebdomadaire, mensuelle...).
- Selon un axe de gravité d'impact (sans conséquence, impact faible, impact moyen, impact fort, très fort...).



L'analyse de risques consiste ensuite à définir les parades associées à chacun, c'est-à-dire des actions permettant d'éliminer ou réduire le risque, l'éviter en mettant en place un scénario de contournement... ou bien le prendre !

Le responsable de l'étude avant-projet identifiera aussi les points critiques, c'est-à-dire les moments du projet où on connaîtra un pic maximum de probabilité de survenance d'un risque.

Et une fois démarré, les chefs de projet MOA / MOE et Recette devront suivre ces risques dans leur domaine respectif et "faire vivre" cette matrice qui n'a rien de statique.

Nous ne détaillerons pas davantage ce processus d'analyse de risques et nous préférons renvoyer le lecteur sur l'abondante littérature relative à leur gestion, domaine qui est plus large que celui de l'organisation des tests.

Il est évident que les risques sur le projet peuvent impacter la recette qu'elle soit fonctionnelle ou technique puisque ces étapes en font partie.

En revanche, les risques sur le produit intéresseront davantage les équipes de tests comme nous allons le voir dans les sections qui suivent, car ils définissent très clairement les lieux de l'application à produire où la survenance d'une anomalie est susceptible d'avoir un impact.

Ainsi, nous parlerons de "matrice des risques projet" et "matrice des risques produit", ces deux matrices intervenant de manière différente dans le projet.

La matrice des risques projet sera un outil décisionnel utilisé par l'instance de pilotage du projet. À la fin du projet, cette matrice sera pour partie obsolète, tout du moins pour les risques qui disparaissent avec la fin du projet : seuls subsistent les risques "post-projet".

La matrice des risques produit sera elle aussi un outil décisionnel, mais elle aura un usage différent puisque c'est une fois le projet fini qu'elle trouve un intérêt plus important et doit impérativement être maintenue.

■ Remarque

La fréquence de survenance d'un risque produit varie dans le temps par définition puisque le logiciel montera progressivement en puissance entre le premier jour de son utilisation et le moment où sera atteinte une vitesse de croisière. En revanche, l'impact est le plus souvent constant.

C'est cette matrice des risques produit qui intéressera le chef de projet recette pour définir un périmètre de tests.

Un risque produit signifie en effet qu'il existe une probabilité de défaillance du logiciel. La parade à mettre en œuvre pour chaque risque produit est donc une batterie d'actions de vérification, ou tests logiciels, afin d'éprouver le produit dans son usage réel.

Définir la matrice des risques sur produit revient donc à définir un périmètre de tests dans une démarche que l'on appellera *Risk Based Testing* ou RBT.

1.1.3 Lister les exigences

Nous pouvons comprendre le terme exigence comme une contrainte ou un besoin selon le contexte – parfois les deux ensembles – une contrainte étant définie comme un besoin impératif, notion qui se traduit en anglais par le terme *requirement*. Par exemple, avoir besoin de 10 postes de travail est une exigence du projet pour répondre à une contrainte logistique.

Un autre exemple serait de dire que le logiciel futur permette de réaliser un virement bancaire – nous aurions alors une exigence fonctionnelle du produit – et simultanément d'être sécurisé comme le reste de l'application de banque en ligne où cette fonction est accessible – nous aurions alors une exigence technique de fiabilité du produit dans sa globalité.

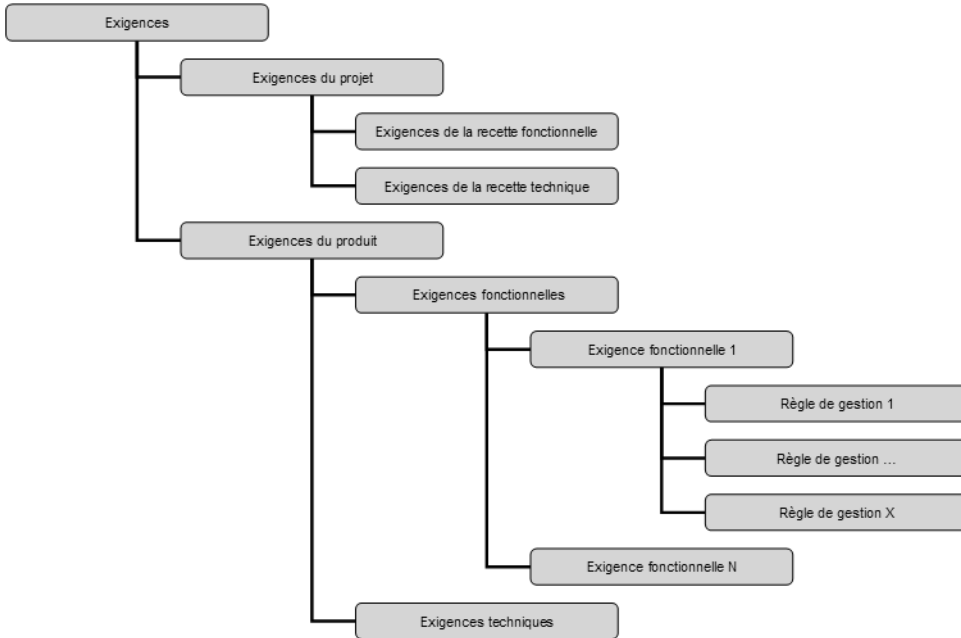
De ces exemples, nous comprenons que le mot exigence désigne de nombreuses choses au sein d'un même projet et que seul il ne veut rien dire : une hiérarchie des exigences est donc nécessaire dont la base est la distinction entre exigence projet et produit.

Les exigences du projet se décomposeront alors en fonction des différentes phases de son exécution : exigences organisationnelles, logistiques... dont celles qui concernent l'équipe en charge des tests techniques et celle en charge des tests fonctionnels.

Quant aux exigences du produit, elles répondent à une typologie différente mais nous pouvons déjà distinguer deux familles :

- Les exigences fonctionnelles et les règles de gestion à implémenter, c'est-à-dire une décomposition structurelle du produit à réaliser, une description du but.
- Les exigences techniques, c'est-à-dire les qualités spécifiques auxquelles doivent répondre toutes les parties du produit.

Le schéma ci-dessous reprend cette classification :



La démarche descriptive du projet que nous venons d'aborder ici se nomme "*requirement analysis*" en anglais ou "analyse des exigences". Conjointement à l'analyse des risques, elle constitue une autre manière d'aborder une étude avant-projet en recensant non pas une probabilité de survenance d'un événement désagréable, mais au contraire ce qui est souhaité et souhaitable.

Autre différence notable, autant une analyse des risques n'est pas toujours menée (ou bien très succinctement), autant une analyse des exigences est incontournable.

Le chef de projet recette s'intéressera donc par nécessité au résultat de cette étude qui d'ailleurs définit le contour des spécifications fonctionnelles – du moins la sous-arborescence des exigences fonctionnelles.