



Les Livres Blancs de MARTE

Constitution d'un centre de test et Tierce Recette Applicative

Processus de qualification

Livre blanc rédigé
par Alain Sacquet

mars 2007

■ Constitution d'un centre de test

La création, au sein d'une DSI, d'un centre de test destiné à garantir la qualité des logiciels avant leur mise en production est un objectif largement partagé.

Comment en effet ne pas souhaiter organiser la recette des développements au sortir des études?

L'organisation concrète des tests au sein des DSI reste cependant une opération délicate.

MARTE présente ici sa vision de ces difficultés et décrit les points clés du déploiement réussi d'une activité de qualification.

Pourquoi est-il difficile de monter un centre de qualification ?

Allons droit au fait. Toutes les directions métier ou informatique connaissent l'importance des tests, mais rares sont celles qui sont réellement satisfaites de la place occupée effectivement par cette activité. On a le sentiment d'en faire trop ou pas assez, que personne ne se satisfait vraiment des résultats obtenus, que les intervenants sont peu motivés.

L'image des tests reste mitigée. Aucune solution n'apparaît réellement satisfaisante, définitivement acquise.

Pourquoi cette image, pourquoi ce sentiment ?

Une première salve de bonnes raisons vient tout de suite à l'esprit. Les tests sont l'activité qui fâche par excellence : spécifier avec les métiers est passionnant, concevoir la solution est créatif, développer techniquement ou faire évoluer le logiciel est constructif et l'exploiter en production c'est rendre service tous les jours.

Et puis, tester arrive toujours au mauvais moment, puisque la qualification devient l'activité dominante à la sortie du tunnel du développement, lorsque l'on est pressé de mettre en production.

Beaucoup d'interlocuteurs doivent se remettre autour d'une table : les métiers, les développeurs ou mainteneurs et les exploitants.

Pour finir, l'affaire n'est pas simple : on sait que la démarche de qualification fonctionnelle suppose une préparation importante, basée sur les exigences fonctionnelles initialement collectées, que des scénarios doivent être produits pour refléter le déroulement attendu des procédures informatisées, cas par cas. La fabrication technique des jeux d'essai ou des bases de test est une expertise en elle-même. Et tout cela précède l'exécution des tests qui dépend des livraisons partielles

de code, et de re-livraisons parfois imprévisibles.

Pour faire bon poids, ajoutons une querelle des anciens et des modernes, des pragmatiques et des dogmatiques, des « chefs de projets talentueux et énergiques » selon CMM et des tenants de la formalisation des procédures et de l'automatisation des exécutions.

Le malaise, le mal être des tests, viendrait alors d'une situation où face à toutes ces difficultés mal résolues, la nécessité de faire s'accroît.

Il faut en effet certifier le logiciel de mieux en mieux car les métiers, eux-mêmes sous pression, veulent mettre rapidement sur le marché des offres sans faille. Parce que les exploitants perçoivent comme une perturbation tout changement au logiciel et ses défauts comme un risque majeur pour leurs engagements de services.

Pour être authentique, cette situation n'est pas une fatalité. Les éléments pour construire des solutions satisfaisantes sont bien présents, mais ailleurs. Prenons deux exemples d'organisation de tests pour illustrer notre propos.

■ Deux exemples d'organisation de test

Tierce Recette Applicative du projet TTM « time to market »

Le projet TTM est un projet d'envergure dont la mise en production permet de déployer de nouvelles prestations proposées en clientèle à la rentrée de septembre et pour lesquelles le réseau de distribution a été formé. Ces nouvelles offres sont abondamment promues par une campagne marketing.

D'un point de vue informatique, ces nouvelles offres ont été installées dans le système existant, sous la forme d'un ensemble d'évolutions à l'applicatif, bien que la nouveauté de ces offres remette en cause la logique de fonctionnement du logiciel vieillissant et monolithique. Le caractère prioritaire du projet le fait déroger aux dates planifiées de mise en production. Le déploiement commercial de septembre impose une phase de test ultime en mai juin. Les retards pris ont amenés à contracter une TRA pour disposer de ressources en juin juillet.

Le centre de test est mis à contribution pour apprécier l'expertise dans les métiers du test de l'infogérant retenu. Il fournit l'outillage d'exécution des tests et les ressources en environnement physique. En effet, cette application étant extrêmement solidaire d'autres systèmes applicatifs pour

l'exécution du plan de production quotidien, il est pratiquement impossible de construire depuis un environnement vierge, les bonnes conditions de son exécution quotidienne. Le centre de test fournit les normes, standard, et outil, pour la documentation des scénarios et la procédure de déclaration des anomalies.

L'application étant complexe à appréhender, le sous traitant a essentiellement été retenu pour sa connaissance de ce domaine applicatif, le nombre des testeurs qu'il peut proposer en cette période estivale, et son pragmatisme dans l'arbitrage des tests à faire vraiment.

Le chef de projet encadrant la TMA de développement s'est engagé à produire la stratégie de test sous forme de scénario qu'il convenait de dérouler. Le prix de la prestation de recette a été convenu sur la base du nombre de scénario transmis, la complexité de l'application et l'hypothèse de deux itérations complètes de l'ensemble des scénarios en moyenne.

L'infogérant commence alors par prendre connaissance du projet, de l'application des scénarios et de l'environnement. Il consacre l'essentiel de ces efforts à comprendre les tests qu'il doit effectivement réussir à dérouler. Il regroupe les scénarios pour faire en sorte qu'ils s'enchaînent, car il

apparaît rapidement qu'il est extrêmement difficile d'initialiser des contextes de données dédiés à chaque scénario. Les tests seront donc peu parallélisés. D'un commun accord avec le chef de projet, les scénarios essentiels sont identifiés. Malheureusement, les tests prennent encore du retard du fait d'ultimes difficultés dans la mise au point des logiciels. Ainsi, la première séquence complète de déroulement d'un test de bout en bout n'a pas pu être faite avant fin juillet. L'interruption du déroulement empêchant la poursuite des vérifications. Un avenant a été conclu pour poursuivre les tests en Août. La mise en production s'est bien passée et les corrections ont pu être apportées rapidement en production lors des premières utilisations de la nouvelle version de l'applicatif. La documentation des scénarios de test sous word, a été archivée, mais elle n'est pas apparue très réutilisable car la valorisation des données est très spécifique à cette exécution de test et l'ensemble est peu lisible. Il apparaît que malgré ce qui avait été initialement convenu, l'infogérant n'a pas réussi à industrialiser l'exécution des tests et ne livre pas de script réutilisables. Aucune solution pérenne n'a été mise en œuvre pour gérer « chirurgicalement » les bases de test. Il n'en a pas été tenu rigueur à l'infogérant qui a

convaincu sans mal de sa bonne foi par les efforts fait pour exécuter concrètement les scénarios.

L'opération est globalement positive malgré un bug découvert dans une application connexe, mais déclenché par les nouvelles données qui lui ont été transmises, et qui aurait pu nuire à l'image de la société, s'il n'avait été promptement corrigé.

TRA du projet ARCHE

Le projet ARCHE est un projet qui a fait l'objet d'une feuille de route. Il donne ainsi lieu à une succession de mises en production échelonnées et déployables. Chacune de ces livraisons s'inscrit dans le planning des versions majeures de l'année. Le responsable de la sécurité des systèmes d'information a rappelé les scénarios critiques qu'il convenait d'exécuter pour qu'il puisse garantir qu'une mise en production concernant les processus métier impactés ne génère pas de risque juridique, financier ou d'image. Le chef de projet métier qui pilote la TMA et la TRA a ainsi complété les objectifs de test et les exigences de vérification. Il a transmis l'ensemble de ces éléments au responsable de la TRA pour cette partie du système d'information et les processus métier concernés. Le chef de projet TRA s'appuie sur le référentiel des scénarios de test associés aux processus métier pour indiquer l'impact des nouvelles exigences de test sur cette liste de scénarios. Le tarifateur convenu dans le cadre du contrat avec la TRA permet de calculer le coût spécifique de cette itération de certification.

La modularité de l'applicatif permet de segmenter les scénarios de test en autant de points de reprise. Des jeux de données sont alors refabriqués pour faciliter le déroulement en parallèle des tests.

Le plan de configuration et le planning de développement permettent de connaître la date au plus tôt d'exécution des tests de bout en bout des différents processus métier modifiés.

A l'issue de cette campagne de test, les nouveaux scénarios, et les nouveaux jeux de données sont ajoutés au référentiel de test par l'infogérant. Ce référentiel appartient au client.

Les indicateurs convenus permettent d'apprécier la performance de la TRA.

Que retenir de ces deux scénarios ?

Ces deux scénarios illustrent non pas une bonne et une mauvaise manière de faire, mais deux manières de faire cohérentes. On peut imaginer qu'ils se déroulent simultanément dans la même entreprise, sous le même management, dans le cadre du même système qualité.

En effet, le point important n'est pas d'industrialiser ou non les tests, ni même d'empêcher toute dérogation au planning annuel de mise en production. La « bonne pratique » selon le niveau 3 de CMM est de mettre en œuvre un processus de qualification tel que : « au cours d'un projet, le processus logiciel standard de l'organisation soit adapté de façon à développer un processus propre à ce projet et prenant en compte les

caractéristiques spécifiques de ce dernier ». Les deux projets décrits correspondent au même standard, mais largement adapté au projet, c'est-à-dire aussi à la qualité de l'architecture du logiciel et à sa testabilité. L'insatisfaction dans les projets de test naît essentiellement de l'inadéquation de la démarche retenue avec la situation à traiter.

Ne pas mettre en place une qualification performante, basée sur un référentiel de processus métier, de gestion des exigences, de tarification formelle, est une perte d'efficacité lorsque l'architecture du logiciel et la qualité de la gouvernance partagée avec les directions métier le permettent. Tout comme l'objectif de solliciter une sous-traitance pour industrialiser les tests d'un applicatif peu modulaire, sensible à la « date du jour », engoncé dans les autres applications, est une prise de risque et une dépense inutile.

Il y a donc bien une méthode et des éléments clés pour déterminer les adaptations efficaces d'un processus standard de qualification à la réalité d'une DSI.

Ce qui est fait, est alors fait en toute connaissance de cause, et un processus d'amélioration progressif peu être engagé, loin de la sinistrose du pauvre testeur isolé. Cette lucidité dissipe les malentendus et évite le sentiment de malaise.

Voyons donc, de façon un peu plus analytique et moins lyrique, les éléments clés de la méthode.

■ Méthode d'adaptation du processus de qualification

Le processus de qualification fait l'objet d'une description normalisée, tout au moins en ce qui concerne ce que l'on peut appeler « l'ingénierie de qualification ». Celle-ci décrit le cycle de vie des tests. Le test se compose de phases successives de préparation, d'exécution et de bilan des tests. L'activité de qualification, comme l'activité de développement donne lieu à des méthodes stables qui décrivent les tâches à mener à l'occasion de chacune de ces phases. Des objets propres au test sont créés à l'occasion de ce processus : scénario, jeux d'essai, base de tests, anomalies, etc.

Ces démarches sont des méthodologies. Ce livre blanc précise leur mode d'adaptation aux contextes. Les activités de qualification des systèmes sont ainsi rendues pertinentes et les insatisfactions chroniques se dissipent.

Les Tierces Recette Applicative (TRA) professionnalisent la pratique des tests. La réflexion sur l'adaptation de la méthode de test au contexte doit tout autant être menée à l'occasion d'une TRA. La satisfaction des partenaires dans le déroulement d'une TRA dépend, de la même manière, de la pertinence des activités concrètement conduites. Le contrat de TRA reprend ces modalités du processus de test, après leur adaptation au contexte et aux moyens disponibles dans le cadre de l'externalisation.

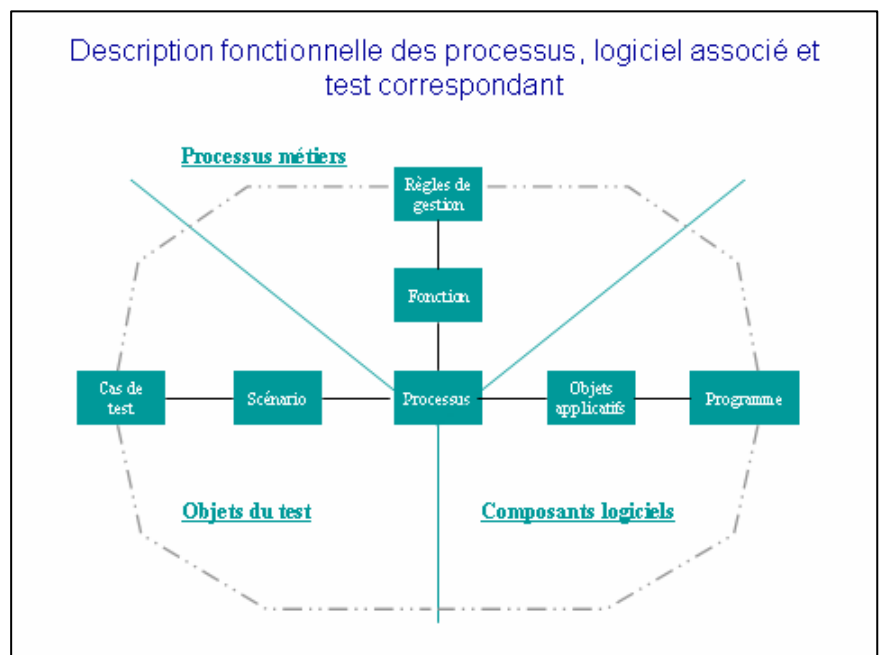
Nous verrons aussi en quoi les tests se prêtent bien à l'externalisation. La complexité de l'externalisation d'une recette se situe en effet à mi chemin entre l'externalisation de l'exploitation et de la maintenance. La dernière partie de ce document présente une série de points d'attention complémentaires, dont certains sont présents dans *Test Process Improvement*®.

Il s'agit d'éléments qui ne relèvent ni des invariants de l'ingénierie du test, ni des points clés d'adaptation à un contexte d'entreprise. Ce sont des aspects dont l'expérience montre tout simplement qu'ils font la différence entre des tests laborieux et des tests performants relativement au contexte bien sûr.

Les invariants : ingénierie du test

L'ingénierie du test est bien décrite dans de nombreux ouvrages et dans le standard TPI®. Rappelons rapidement que les tests découlent d'un certain nombre d'objectifs de vérification, qu'ils donnent lieu à la production de scénarios, que ces scénarios sont ensuite valorisés à l'aide de jeux d'essai. Viennent ensuite la phase d'exécution des tests et de recyclage des anomalies avant la décision de mise en production et d'activation.

Remarquons que ce processus est analogue à celui de développement du logiciel, puisque les objectifs du test sont spécifiés par les métiers, les scénarios sont ensuite conçus, puis les jeux d'essai fabriqués techniquement.

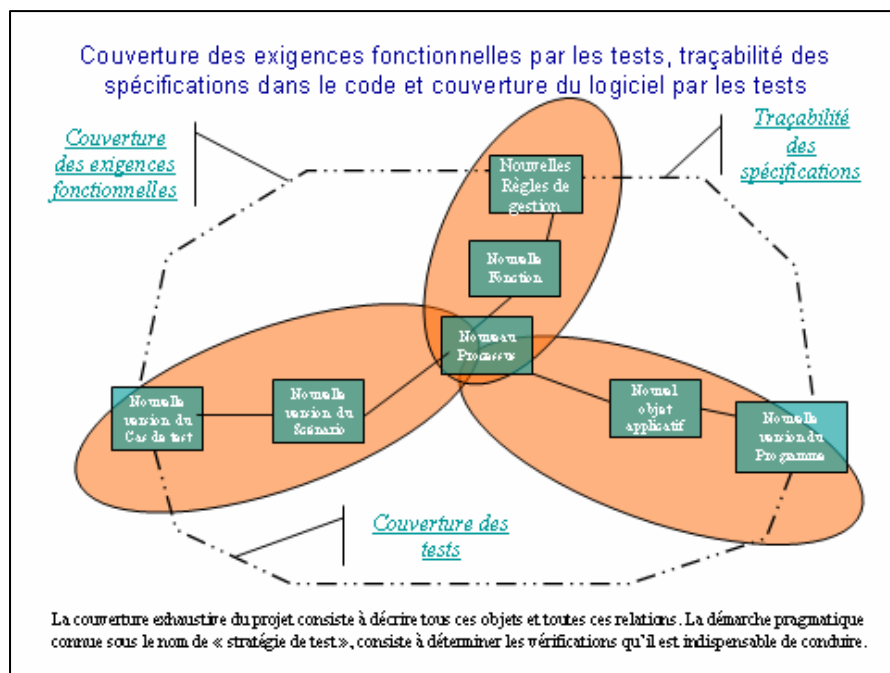


Remarquons également que ces objets propres au test sont en vis-à-vis, d'une part des spécifications fonctionnelles qui ont décrit le fonctionnement souhaité du logiciel, et du logiciel lui-même, d'autre part.

Il y a donc trois familles d'objets telles que décrites schématiquement dans le dessin ci-contre.

Le principe des tests consiste à vérifier que le comportement du logiciel correspond bien à ses spécifications. Les résultats des scénarios et des cas de test sont déterminés par rapport aux spécifications. Le principe des tests consiste à vérifier que le comportement du logiciel correspond bien à ses spécifications. Les résultats des scénarios et des cas de test sont déterminés par rapport aux spécifications. Faire les tests consiste à vérifier, par l'exécution du logiciel, que les résultats attendus sont obtenus comme prévus.

La couverture exhaustive du projet consisterait à décrire tous les tests pour toutes les spécifications du logiciel et d'exécuter tous ces scénarios. La démarche pragmatique connue sous le nom de « stratégie de test », consiste à déterminer les



vérifications qu'il est indispensable de conduire en considérant les fonctionnements les plus fréquents, ceux dont les défauts seraient les plus dommageables et les parties du logiciel où les défauts sont les plus probables du fait de sa complexité, ou d'une organisation de développement répartie aux quatre coins du monde par exemple.

Rappelons enfin que l'exécution du test a systématiquement lieu à un moment critique des projets, puisqu'il s'agit de la phase ultime du développement qui représente

toujours un instant de vérité pour les parties prenantes.

Cette description résumée de l'ingénierie de test est consensuelle. Ni les variations liées au pragmatisme des arbitrages de la « stratégie de test », ni les écarts de maturité atteints dans la gestion des environnements, la gestion des anomalies ou les autres secteurs clés du standard TPI n'expliquent la diversité des situations décrites à travers les deux exemples de la première partie de ce document. Pourquoi ?

Les points clés d'adaptation

Retour sur la stratégie de test

La stratégie de test vise un optimum entre l'effort à fournir et la diminution du risque associé. L'optimum en question n'obéit pas à un algorithme du simplex bien subtil. Le responsable des tests essaye d'atteindre un niveau de risque acceptable dans le temps imparti ! Il tente pour cela d'anticiper la fabrication des scénarios et des jeux d'essai.

Cette stratégie de test est essentiellement contingente : C'est une stratégie fortement contrainte. Elle fait au mieux. Elle subit et finit par ressembler davantage à une tactique d'exécution qu'à une stratégie.

Les aléas subis par la stratégie de test sont :

- Le temps imparti
- La définition du risque acceptable
- La productivité du test pour les phases de préparation et d'exécution

Il est possible de diminuer la sensibilité à ces trois aléas et de rendre la « stratégie » moins contingente en choisissant une organisation particulière du centre de test et en mettant l'accent sur quelques procédés. Ces choix fournissent différents contextes de mise en oeuvre de l'ingénierie de test standard, c'est-à-dire différentes stratégies possibles d'organisation des tests au sein d'une DSI.

Ces choix d'organisation sont de nature à rendre beaucoup plus efficace le processus de test qu'il soit conduit projet par projet, en interne ou en recourant à une (ou des) TRA.

Les aléas subis par la stratégie de test sont :

- *Le temps imparti*
- *La définition du risque acceptable*
- *La productivité du test pour les phases de préparation et d'exécution*

Quelles sont donc ces solutions ?

Le versioning et l'ancrage du processus de qualification

Un premier réglage consiste à améliorer la maîtrise du « temps imparti » au test. La préparation des scénarios ne peut être entamée sans que les fonctionnalités de la nouvelle version du logiciel ne soient connues. De même l'exécution du test suppose la disponibilité du logiciel, d'environnement de test, la préparation de base de test.

Il convient donc d'ancrer correctement ce processus qui est très sensible aux dérives calendaires de ses nombreuses « entrées ».

La solution consiste à définir a priori les périodes calendaires ouvertes aux tests pour les différents logiciels. Ces périodes sont calées elles-mêmes sur les quelques dates prévues annuellement de mise en production. Le projet se plie à ce calendrier et adapte sa feuille de route à ces contraintes. Cette organisation s'appelle le « versioning ». Elle concerne la DSI dans son ensemble ou certains

domaines seulement, pour la totalité du processus de développement ou les phases de qualification et de mise en production seulement. Le versioning affecte la gestion du portefeuille des projets et les feuilles de routes des différents programmes de développem

ent. Cette solution d'organisation dépasse donc le seul processus de qualification mais constitue cependant la clé de l'amélioration de sa performance. Il faut remarquer aussi que la portée du versioning n'est plus le projet, comme dans le cas de l'ingénierie du test, mais l'ensemble d'une nouvelle version du SI. Le

Le versioning donne du temps au test.

versioning donne du temps au test.

L'infogéreur

La répartition des rôles des chefs de projet pour le développement et la qualification constitue également une clé de la performance et de la gestion du temps. On sait qu'il est bon que les équipes de test se distinguent des équipes de développement de façon à éviter le conflit des priorités pour des ressources identiques. L'effet pervers d'une telle organisation est de mettre en face à face les chefs de projet de test et de

développement. MARTE préconise que les équipes de développement et de test soient distinctes, mais que la responsabilité du projet soit unique. Ce schéma est obtenu naturellement lorsqu'un unique chef de projet « infogéreur » fait travailler une TMA pour le développement et une TRA pour la recette. Dans tous les cas, l'unicité du chef de projet pour l'ensemble de ces phases permet de gérer un planning unique d'avancement des travaux dans le cadre de l'agenda du versionning. La réalisation des travaux de conception des scénarios de test peut se faire au plus tôt. De même, le suivi de l'avancement de la fabrication du logiciel et de son assemblage permet de connaître les ensembles logiciels intégrés disponibles pour des exécutions anticipées de test.

Certification ou qualification : le RSSI

Décider que les tests sont concluants est parfois délicat. L'ingénierie du test définit différentes catégories de vérification qui sont à conduire au fur et à mesure de l'intégration du logiciel. La validation de la nouvelle version du système doit bien sûr reposer sur des bases objectives. Différents acteurs sont parties prenantes à la définition de cet ensemble de conditions : la maîtrise d'ouvrage et la production notamment.

MARTE conseille que la définition du risque acceptable mette également à contribution le RSSI (responsable de la sécurité du SI). En effet, dans la plus grande partie des cas, les logiciels qui vont aller en production

ne sont qu'une nouvelle version d'un système d'information déjà existant. Les Processus Métier concernés par ces modifications sont déjà connus, ainsi que les risques financiers légaux ou d'image qui leur sont associés. Il est dès lors efficace de constituer un référentiel des vérifications que le RSSI veut

nicienne : le projet tourne autour du SI.

La mise à contribution du RSSI et la construction d'un référentiel des risques pour diminuer les aléas liés à la définition des conditions d'acceptation supposent naturel-

Les points clés de l'adaptation du processus de test résident dans :

- *La mise en place du versionning*
- *Les rôles des chefs de projets*
- *La mission de contrôle permanent du RSSI*
- *La bonne mesure de la testabilité du logiciel*

voir menées pour toutes les modifications de ces processus. Cette organisation stabilise le centre de certification et permet de construire un référentiel réutilisable des vérifications à mener.

D'une manière générale cette solution d'organisation permet de conférer une réelle mission à la structure permanente de certification du SI. Une part importante des tests dans les phases ultimes d'assemblage peut être construite efficacement à l'aide de cette organisation.

Là aussi, l'ingénierie de test n'est pas mise en cause. Au contraire, l'objectif présent dans TPI© de rendre réutilisable les « cahiers de test » devient concret. Cette solution naît d'une approche différente du projet qui n'apparaît plus comme le centre des préoccupations, mais comme une évolution du SI. La performance du processus de qualification résulte de cette révolution quasi coper-

lement d'avoir mis en œuvre le versionning.

La testabilité du logiciel

La productivité du test constitue le troisième aléa mais non le moindre. Nous énumérerons dans la dernière partie de ce document un ensemble de points d'attention qui contribuent à la performance du processus de qualification. La testabilité du logiciel est toutefois le facteur le plus important de la productivité du test. Certains logiciels sont extrêmement difficiles à tester, et n'ont manifestement pas été conçus pour voir leur comportement vérifier facilement.

Les éléments qui rendent les tests délicats ont été illustrés dans l'exemple de la première partie. Certains logiciels sont extrêmement liés à d'autres dans le plan de production quotidien ou hebdomadaire, rendant quasiment impossible une exécution déterministe et une interprétation certaine des résultats produits. D'autres ne

permettent pas de « contracter le temps », ou de gérer la date du jour, rendant très délicat la reproduction d'un cycle métier long, comprenant des calculs financiers sur plusieurs mois par exemple. Pour d'autres encore, c'est la gestion des bases de test qui pose problème. Impossible d'injecter une situation de référence, ou d'échantillonner un contrat !

Cette testabilité doit être appréciée au plus tôt, car elle conditionne la possibilité d'exécuter des tests en

parallèle, de réinitialiser des contextes, de constituer des points de reprise.

La séquence de test est alors un scénario très long, contenant de très nombreux cas à vérifier en une passe et pour lequel une interruption de l'exécution suppose une reprise au début.

Ce genre de situation doit être connue, et partagée car elle limite l'ambition du test, qu'il soit conduit par une TRA ou en interne. L'automatisation des tests n'a qu'un

ROI très faible dans ces circonstances et la réutilisation des scénarios est illusoire.

Seules des modifications du logiciel, le passage à des architectures de service ou, au moins, une amélioration de la modularité sont en mesure d'éclaircir l'horizon. Mais ces solutions ne sont pas disponibles à la veille du test.

La tierce recette applicative

La tierce recette applicative peut être organisée de différentes façons comme l'exemple initial l'a montré. L'objectif peut être de disposer forfaitairement de ressources externes disponibles par contrat, ou de contribuer au fonctionnement d'un centre de certification très structuré, dont l'activité est planifiée précisément dans le cadre du versionning.

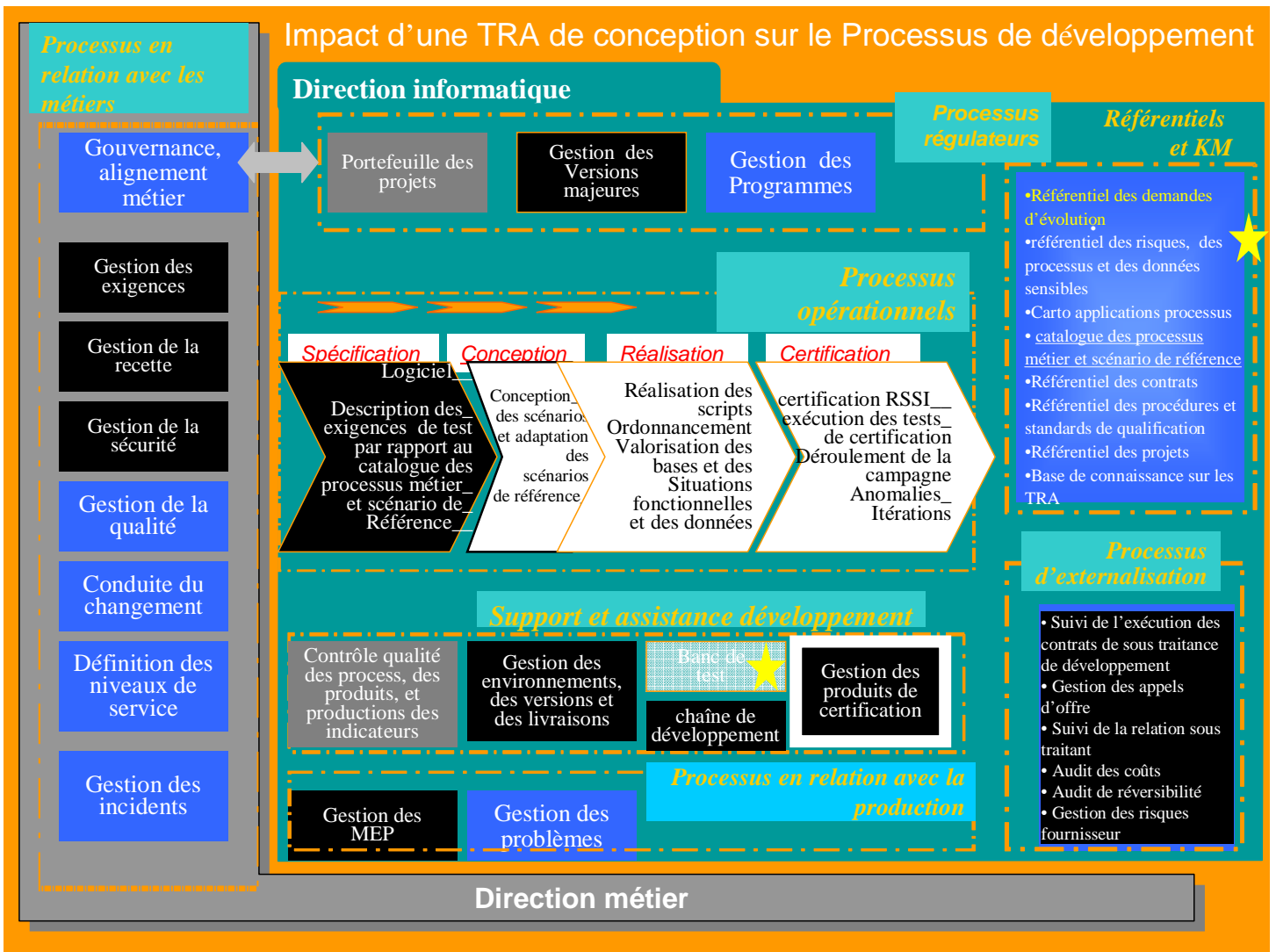
S'agissant d'une externalisation, il est intéressant de réfléchir aux

choix possibles en terme de :

- Mise à disposition par le fournisseur de ressources humaines,
- Mise à disposition par le fournisseur de procédés et d'outils ou utilisation des procédés maîtrisés en interne,
- Mise en commun de référentiel inventoriant les processus métiers critiques, les risques afférents et les scénarios de test associés,

- Portée de l'externalisation selon que le client fournit les spécifications fonctionnelles ou les scénarios de test.

Nous nous focaliserons ici sur deux points clés de la Tierce Recette : la définition des unités d'œuvre et la réversibilité du contrat.



Les unités d'œuvre de la tierce recette

Les tests sont une activité propice à l'externalisation. Moins parce les tests rebutent les équipes internes, ou parce que le risque d'altérer le logiciel est nul, que parce qu'il est possible de construire un contrat efficace. En effet, la définition d'un contrat de TMA est plus subtile que celle d'un contrat de TRA, notamment lorsque la TMA se voit confier la totalité des interventions sur un logiciel. Le point délicat concerne la tarification de la prestation. Dans le cas de la maintenance, le tarificateur n'est pas aisé à mettre au point lorsqu'il doit permettre de calculer le prix de maintenances évolutives décrites sous l'angle fonctionnel. Ce point est débattu dans le livre blanc que MARTE consacre à l'externalisation. Dans le cas des tests, il est en revanche possible de construire des unités d'œuvre non ambiguës, dès lors que le système est décrit à grande maille comme un ensemble de Processus Métier et de scénarios associés.

Cet ensemble de processus métier et de scénario constitue une trame du SI relativement stable. Les premiers processus qui ont été décrits sont les processus critiques, pour lesquels le RSSI a précisé les risques et les scénarios associés. Dès lors, les nouvelles spécifications (ou nouvelles exigences) ayant fait l'objet du projet peuvent être situées par rapport à l'un ou l'autre de ces scénarios. Cette première opération, conduite par le sous-traitant, permet d'identifier les

scénarios à exécuter, et de mesurer ainsi l'effort de test qu'il doit conduire. En final, le référentiel des scénarios et des processus métier peut être enrichi à l'issue de la nouvelle campagne de test.

Cette méthode conduit assurément

scénario, cas de test et le « banc de test ».

Le client aura tout intérêt à préciser dans son contrat qu'il est propriétaire des référentiels évoqués.

Qu'en est-il du « banc de test » ? En

La tierce recette est maîtrisable par le client car elle constitue une pure activité de service qui ne modifie pas le logiciel. Elle peut donner lieu à des contrats de service faisant l'objet de tarifs déterminables.

à intensifier la sous traitance autour de ces activités. La tierce recette est maîtrisable par le client car elle constitue une réelle activité de service, pouvant donc donner lieu à des contrats de service et des tarifs déterminables. Elle est en ce sens plus proche de l'infogérance de production qui est une pure activité de service récurrents que de la maintenance dont la complexité vient de ce qu'il s'agit d'une activité de fabrication modifiant l'objet sur lequel elle porte.

La réversibilité de la TRA et la propriété des référentiels

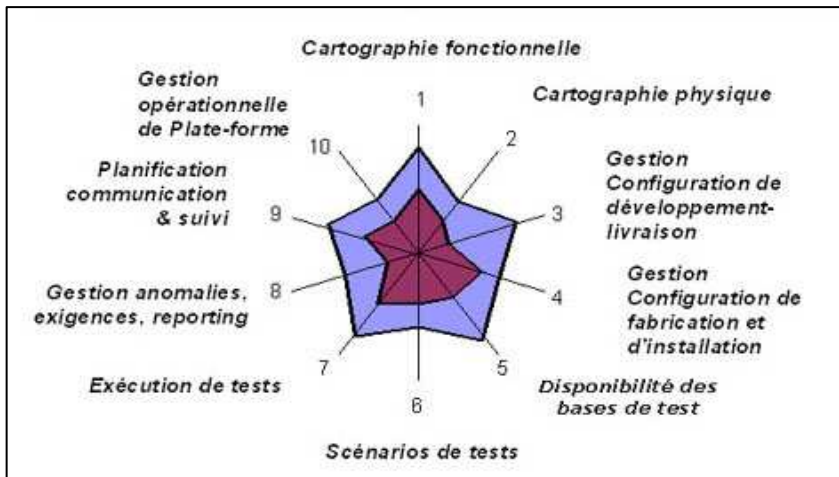
Comme pour tout contrat d'externalisation, il est important de veiller à la réversibilité et à la propriété des actifs. Dans le cas d'une TRA et dans la mesure où le logiciel est correctement testable, les deux catégories d'actifs auxquels il convient d'être vigilant sont les référentiels de processus métier,

fait, cette expression recouvre les procédés et les outils mis en place pour exécuter les tests. Il peut s'agir de script d'automatisation des lancements, des échantillonnages de base, de vieillissement des données, d'anonymisation. Il peut s'agir des robots d'exécution et de détection des écarts, ou de programmes nouveaux permettant de consulter des données.

Lorsque la TRA est l'occasion de mettre en place ce genre de moyens, leur statut doit être précisé au titre du plan de réversibilité (ou de transfert).

Les procédés de gestion des anomalies ou de planification entrent également dans cette catégorie, mais ils sont moins critiques. De plus, le client impose en général les outillages dont il dispose pour organiser contractuellement la relation avec son fournisseur.

Autres points d'attention



Ce chapitre récapitule quelques points d'attention qui contribuent à l'efficacité du test sans caractériser pour autant l'une ou l'autre stratégie.

Les anomalies d'exécution et les défauts du logiciel

Un point de discussion, et d'incompréhension parfois, concerne la différence entre les anomalies détectées par les outils d'exécution automatique des tests et les défauts du logiciel. Evidemment, tout écart lors d'un test n'est pas nécessairement imputable au logiciel à tester. Il peut s'agir d'un problème d'environnement ou de données de test. De ce fait, il est intéressant de distinguer l'usage qui doit être fait de l'outil de gestion des anomalies d'exécution souvent fournie avec l'automate de test, de l'usage de l'outil de gestion des défauts du logiciel qui accompagne les outils de gestion de configuration logiciel.

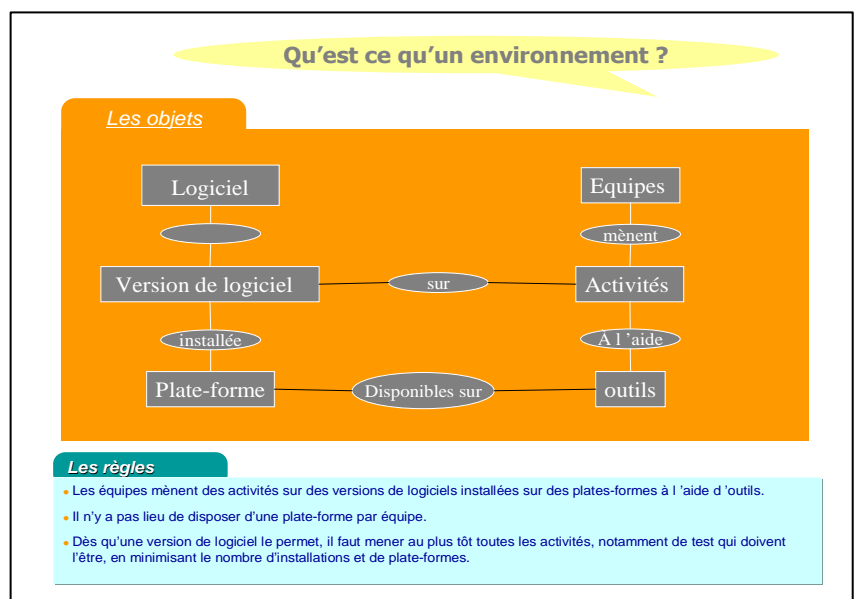
La gestion de configuration logiciel

Un point important de la qualité de la recette est la maîtrise des versions du logiciel sur lequel chaque test est fait. On veillera donc à disposer d'une Gestion de Configuration Logiciel qui assure ce service. Le plan de gestion de configuration précise par ailleurs la façon dont sera intégré le logiciel. Il indique les responsables de la fourniture des différents éléments. Un défaut est associé à une version d'un composant logiciel.

La gestion des environnements et les installations

De la gestion de configuration à la gestion des environnements il n'y a qu'un pas. La notion d'environnement est souvent très ambiguë. Il peut être pratique de convenir qu'un environnement est composé d'une plate-forme (matériel et logiciel d'infrastructure) sur laquelle est installée une version du logiciel applicatif et des données.

Pour être efficace, la solution consiste à ne pas multiplier les plates-formes pour éviter la multiplication des versions de logiciel en test à un même moment et les charges d'installation. Cette dérive apparaît lorsque chaque acteur entend disposer de ces propres moyens. Il faut au contraire définir les différentes activités de test qui peuvent être menées sur une version maîtrisée du logiciel et sur une même plate-forme par les différents acteurs.



En Conclusion

L'activité de test, l'organisation d'un centre de certification et les contrats de Tierce Recette Applicative sont souvent le parent pauvre de l'ingénierie logicielle. On constate pourtant que ces activités font appel à une grande technicité, un sens de la méthode et nécessitent un savoir faire notable dans la conduite des opérations. Autant de compétences à réunir qui font de la qualification un défi humain que doivent relever les

managers informatiques pour la mobilisation des « meilleurs », la définition des missions, et la valorisation justifiée des postes et des profils.

L'externalisation croissante des développements incite les directions informatiques à vérifier la qualité du logiciel dont les évolutions sont sous traitées. Quitte à externaliser à son tour cette vérification. Dans tous les cas, la gestion des « actifs

informatiques » de l'entreprise doit être assurée.

MARTE espère que ce document vous a conforté dans l'idée qu'une méthode existe pour construire en toute connaissance de cause le processus de qualification qui correspond à votre situation précise. Cette méthode, que les consultants de MARTE pourront dérouler avec vous, permet également de mettre en place les TRA et d'en négocier au mieux le contrat.