

Dynamic Systems Development Method



1 Introduction

1.1 Définition de RAD

Les sociétés informatiques se voient contraintes de livrer de plus en plus vite des systèmes informatiques opérationnels à leurs clients. De ce fait, il est indispensable de repenser les processus de développement des systèmes informatiques.

La technique du développement rapide d'applications (RAD - Rapid Application Development) permet de réaliser et de maintenir des systèmes informatiques en alliant utilisation performante d'outils et de techniques, prototypage et courts délais de livraisons. Un projet géré avec une approche RAD permet, en général, de fournir un système opérationnel en six mois.

Idées fausses concernant RAD

Parmi les idées fausses fréquentes existant au sujet de RAD, il y a celles-ci :

- RAD, JAD (Joint Application Development) et Ateliers Facilités sont synonymes, alors qu'en fait JAD n'est que l'une des techniques pouvant être utilisée dans des projets RAD, une technique très puissante somme toute.
- Le développement client/serveur est de type RAD car il implique habituellement l'utilisation d'interfaces graphiques utilisateurs, etc. Pourtant, bien que les outils de construction d'interfaces graphiques utilisateurs soient très utiles dans les projets RAD, ils n'apportent en aucun cas une solution globale à la gestion de ces projets. Les développements client/serveur peuvent être de très longs projets.
- L'utilisation de la technique RAD ne peut être envisagée que pour de petits projets de développement très ciblés et non pas pour l'extension de systèmes existants ou pour des projets très importants. En fait, la technique RAD peut être employée pour tout projet de développement répondant aux principes DSDM décrits dans le chapitre 2 'principes de base' et présentant les critères énoncés au chapitre 2 et dans l'annexe A.
- RAD est « la porte ouverte au bricolage ». L'utilisation de la méthode DSDM permet de réfuter cette idée.

1.2 Raison d'être de DSDM

DSDM (Dynamic Systems Development Method) est une méthode RAD domaine publique qui part du principe que pour toute entreprise ayant besoin d'un outil RAD pour améliorer son aptitude de développement informatique, il y a une autre entreprise qui a besoin de reconstruire entièrement son processus de développement de produits et de systèmes. DSDM offre un canevas permettant d'élaborer et de maintenir des systèmes dans des délais très courts et constitue une « recette » pour assurer de manière continue le succès des projets RAD. Cette méthode ne s'adresse pas uniquement aux développeurs RAD mais concerne également toutes les personnes intéressées par une méthode de développement performante, y compris les utilisateurs, les chefs de projets et le personnel en charge de l'assurance qualité.

De nombreux projets de développement de systèmes ne répondent pas de manière appropriée aux besoins de l'utilisateur final. On peut classer ces échecs dans une des cinq catégories suivantes :

- 1)Le système ne répond pas aux besoins de gestion qui ont justifié son développement. Il est alors abandonné ou fait l'objet de modifications coûteuses.
- 2)Le système n'est pas assez performant et ne peut donc pas répondre aux besoins des utilisateurs. De nouveau, il est abandonné ou des tâches de maintenance coûteuses sont entreprises.

- 3)Le système comporte des erreurs causant des problèmes inattendus. Des corrections doivent être effectuées, ce qui représente un coût supplémentaire.
- 4)Les utilisateurs n'acceptent pas d'utiliser le nouveau système, soit pour des raisons politiques, soit parce qu'ils n'ont pas été assez impliqués dans le développement ou qu'ils ne sont pas convaincus par l'intérêt de son utilisation.
- 5)Les systèmes sont acceptés au début, puis leur maintenance devenant impossible au fil du temps, ils sont finalement abandonnés.

DSDM s'attache à empêcher ces cinq types d'échecs.

L'approche DSDM est basée sur l'hypothèse fondamentale suivante : **on ne construit rien qui soit parfait du premier coup**, mais il est possible de réaliser 80% d'une solution en utilisant seulement 20% du temps nécessaire à la réalisation de la solution totale. Un des problèmes majeurs que posent les approches plus traditionnelles du développement de logiciels est le fait qu'elles supposent que les utilisateurs présumés du système informatique sont capables de déterminer par avance tous leurs besoins futurs. Le problème est d'autant plus complexe que l'existence même d'un nouveau système informatique modifie les besoins des utilisateurs, dans la mesure où il remet en question leurs méthodes de travail.

Avec l'approche classique, séquentielle (ou « en cascade »), il n'est pas possible de démarrer une étape avant que la précédente ne soit terminée et entièrement testée. En pratique, ce qui prend le plus de temps, c'est de passer du projet « réalisé à 80% » à la solution finalisée, en supposant qu'aucune étape n'aura jamais besoin d'être révisée. Cela signifie que l'on passe énormément de temps à revenir sur des étapes soit disant « terminées » et à défaire des éléments précédemment validés. Il en résulte soit des projets livrés en retard et dépassant le budget alloué, soit des projets qui ne répondent pas aux besoins puisqu'on a pas pris le temps de revoir les spécifications.

Avec DSDM, il est possible de revenir sur toutes les étapes précédemment effectuées selon l'approche itérative qui caractérise cette méthode. Ainsi, **il suffit que l'étape courante soit suffisamment avancée pour pouvoir passer à l'étape suivante**, puisqu'elle pourra être terminée au cours d'une prochaine itération. On part du principe que les besoins risquent de changer à mesure que la compréhension s'affine et qu'à ce stade, un travail plus approfondi constituerait une perte de temps !

Les applications réalisées selon l'approche DSDM permettent de répondre aux besoins actuels et immédiats du business alors que l'approche traditionnelle prend en compte toutes les possibilités envisagées. On peut donc s'attendre à ce que le système informatique résultant reflète mieux les besoins réels, soit plus facile à tester et à intégrer aux méthodes de travail des utilisateurs. Le coût de développement de la plupart des applications ne représentant qu'une petite partie des coûts supportés durant le cycle de vie total, il semble logique de réaliser des systèmes informatiques plus simples, qui répondent à leurs objectifs et qui, une fois développés, sont plus faciles à maintenir et à modifier. Cette démarche est possible dans la mesure où la partie maintenance peut être considérée comme un élément incrémental de la solution totale.

1.3 **Avantages de l'utilisation de DSDM**

Grâce à l'utilisation d'un processus itératif basé sur le prototypage, DSDM implique les utilisateurs tout au long du cycle de vie du projet. Ceci présente de nombreux avantages, dont les suivants :

- les utilisateurs s'approprient plus facilement le système informatique,
- le risque de réaliser un système informatique qui ne convienne pas est moindre,
- le système final est plus susceptible de répondre aux besoins réels des utilisateurs,
- les utilisateurs seront mieux formés puisque ce sont leurs représentants qui définiront et coordonneront cette formation,

- la mise en œuvre sera probablement plus facile grâce à la coopération de toutes les parties concernées tout au long de la phase de développement.

1.4 Contenu de DSDM

La méthode DSDM propose une approche globale du développement de logiciel dans un environnement de développement rapide (RAD). De nombreuses méthodes de développement de logiciels se concentrent uniquement sur une activité, telle que l'analyse et la conception ou la gestion de projet. DSDM fournit un canevas couvrant l'ensemble du cycle de développement afin d'assurer le succès du projet. Le schéma 1.1 illustre tous les domaines abordés par DSDM. La couche supérieure représente le cycle de développement propre à la méthode DSDM. Les couches inférieures correspondent à toutes les parties du projet de développement qui sont prises en considération par la méthode.

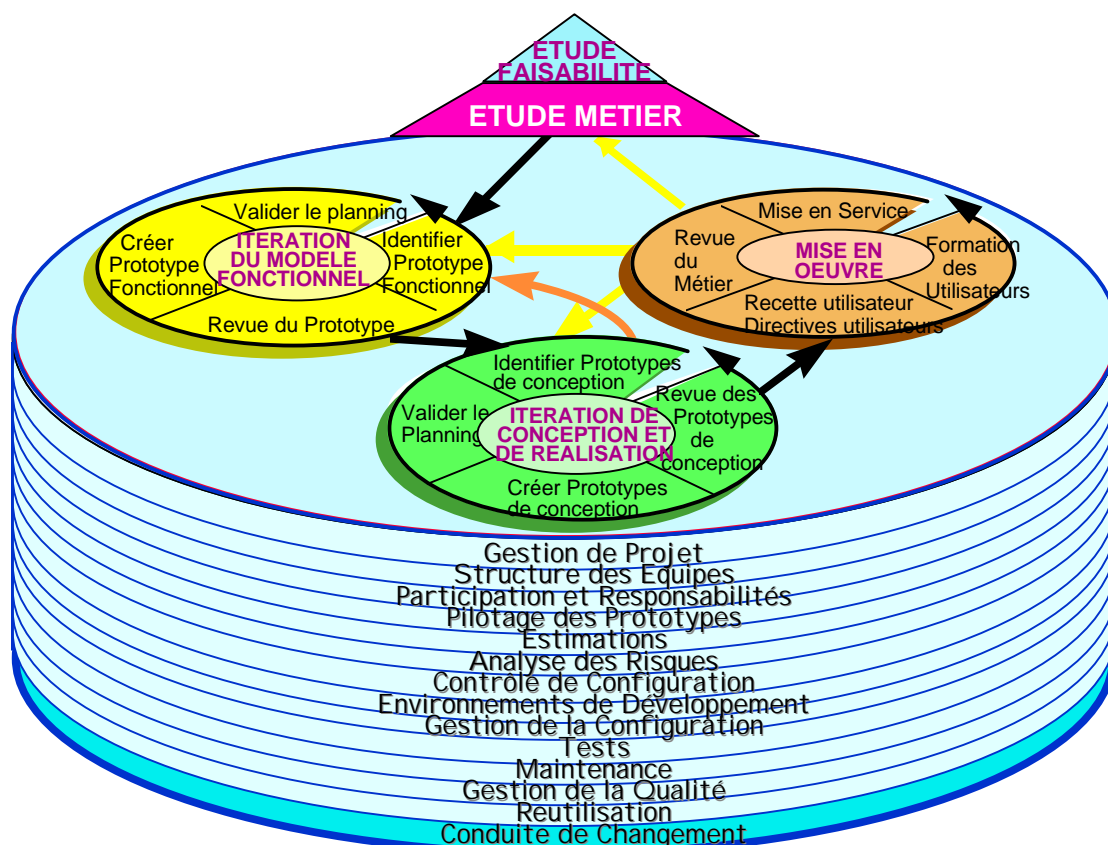


Schéma 1.1: Synthèse de DSDM en trois dimensions

Tout au long du développement de la méthode DSDM, le Consortium des membres (*chapitre 1.6*) a surtout porté son attention sur les points spécifiques qui doivent être pris en compte au cours d'un projet RAD. Par exemple, DSDM ne spécifie aucune technique particulière pour la gestion de projet ou la gestion de configuration mais traite de ces activités spécifiques dans un projet RAD.

1.5 Contenu du manuel

La **partie 1** du manuel aborde les fondements de la méthode DSDM, décrit sa philosophie générale et propose une vue très globale de son fonctionnement. Cette partie s'adresse à un large groupe de lecteurs (et pas uniquement au personnel informatique). Elle doit être lue entièrement par toute personne qui, commençant à s'intéresser à DSDM, a besoin de comprendre les grandes lignes de la méthode et quand elle doit être utilisée. La partie 1 comporte les chapitres suivants .

Le chapitre 2 présente les principes sur lesquels est basée la méthode DSDM. Il est essentiel de le lire pour comprendre la spécificité de cette approche.

Le chapitre 3 décrit le processus et en quoi il diffère des approches de projet informatique traditionnelles.

Le chapitre 4 décrit les types de projets pour lesquels il convient d'utiliser la méthode DSDM. Les caractéristiques des projets couvrent les domaines d'organisations, culturels et technologiques.

Le chapitre 5 s'intéresse à la culture de projet qui doit être en place pour que la méthode DSDM puisse être efficace.

Le chapitre 6 présente les différents rôles des intervenants dans un projet de réalisation de système informatique faisant appel à DSDM.

Les chapitres 7 à 9 exposent certaines techniques clés permettant aux projets DSDM d'avancer plus rapidement vers la solution adéquate pour l'entreprise. Il s'agit des « Timeboxes », de la hiérarchisation des objectifs, des « Ateliers Facilités » et du prototypage.

Le chapitre 10 constitue un guide pour les entreprises qui souhaitent intégrer la méthode DSDM à leur organisation.

La **partie 2** s'adresse au personnel informatique qualifié qui souhaite comprendre les différences entre DSDM et les approches plus traditionnelles. Elle peut faire l'objet d'une lecture intégrale la première fois mais elle est plus spécialement destinée à être utilisée comme une source de références. Cette partie comporte les chapitres suivants :

Le chapitre 11 définit le cycle de développement DSDM de façon plus formelle que dans le chapitre 3. Ces deux chapitres sont complémentaires.

Le chapitre 12 décrit les catégories de prototype DSDM, le moment où ils doivent être produits au cours du cycle de vie DSDM et comment ils sont contrôlés.

Le chapitre 13 présente les définitions formelles des éléments produits durant un projet DSDM.

Les chapitres 14 et 15 définissent les rôles au sein d'un projet DSDM et présentent les structures qui permettront aux équipes DSDM de travailler efficacement.

Le chapitre 16 traite de la planification, de la surveillance et du contrôle de projet. Il propose également quelques repères aux chefs de projet qui utilisent DSDM pour la première fois.

Le chapitre 17 indique ce qui doit être mesuré dans les projets DSDM pour évaluer la productivité ou la perte de cadence. Il indique quand et comment faire cette estimation durant un projet DSDM.

Le chapitre 18 présente certains des problèmes spécifiques au développement DSDM et apporte quelques éléments de solution.

Le chapitre 19 montre comment l'assurance et le contrôle qualité sont intégrés à DSDM, et la façon de réaliser un audit de projet DSDM.

Le chapitre 20 décrit les activités liées à la gestion de la configuration dans un projet DSDM. Une bonne gestion de la configuration est essentielle pour pouvoir contrôler des produits évoluant rapidement au cours d'un projet DSDM.

Le chapitre 21 décrit le processus de test inhérent à DSDM et donne des conseils quant aux outils à utiliser.

Le chapitre 22 présente l'ensemble des techniques de modélisation recommandées par le Consortium aussi bien pour le développement structuré et que pour le développement orienté objet. Certaines entreprises souhaitent cependant utiliser leurs propres techniques. Ce chapitre permet d'évaluer les techniques courantes.

Le chapitre 23 s'intéresse plus particulièrement aux « Ateliers Facilités », (travail collectif animé par un médiateur), une technique puissante mise en œuvre par la méthode DSDM. Ce chapitre complète la description de son utilisation logique qui est en est faite au chapitre 8.

Le chapitre 24 définit les besoins de l'environnement de développement idéal selon l'approche DSDM ainsi qu'un processus susceptible d'aider les entreprises à se procurer les outils adéquats.

La **partie 3** aborde des problèmes plus vastes que ceux directement liés à un projet DSDM particulier. Elle contient des résumés de livres blancs rédigés par des groupes de travail au sein du Consortium et abordant des sujets aussi variés que l'utilisation de DSDM pour des projets de grande envergure ou pour des développements Internet. Ces résumés ne représentent que la partie émergée de l'iceberg, mais il donnent un aperçu du contenu des documents disponibles pour les membres à part entière. De plus, la partie 3 évalue la position de DSDM dans l'environnement professionnel et informatique global.

1.6 Le consortium DSDM

Etant donné le manque de concertation autour de RAD et l'incapacité des entreprises à appliquer cette démarche sans utiliser de méthodes propriétaires, un Consortium des plus importantes sociétés du Royaume-Uni a été créé en janvier 1994. Au sein de ce Consortium, on trouve de grands comptes des secteurs public et privé, de grandes sociétés informatiques, de petits éditeurs d'outils, ainsi que des cabinets de consultants. Avec le succès de la méthode DSDM, ce Consortium a pris une dimension internationale.

L'objectif initial du Consortium était de définir une méthode du domaine public pour le développement rapide d'applications (RAD) qui soit susceptible de devenir la norme *de fait*. Cet objectif a été réalisé au Royaume Uni, même si le succès de la méthode s'étend aujourd'hui bien au-delà de ces frontières.

Vous trouverez, décrite ci-dessous, la structure organisationnelle du Consortium telle qu'elle se présente au Royaume Uni. Les consortium qui se forment dans les autres pays utilisent la même organisation. Celle-ci comporte cinq groupes de travail qui supervisent les différentes activités. Ces groupes de travail sont les suivants :

- Le groupe de travail technique, « moteur » du Consortium. Ce groupe a créé la méthode et continue de la maintenir. Toutes les propositions relatives à l'orientation générale de la méthode (basées sur les idées et les retours d'informations émanant des membres du Consortium) doivent être adoptées par tous les membres du Consortium.
- Le groupe de travail Promotion qui propose et met en œuvre des stratégies pour promouvoir la commercialisation du Consortium et de la méthode.
- Le groupe de travail Formation et accréditation qui homologue tous les cours de formation et examens DSDM, et procure des examinateurs certifiés.
- Le groupe de travail International qui encourage le développement de consortium DSDM dans d'autres pays.
- Les services membres destinés aux membres qui garantissent que les membres bénéficient pleinement des avantages offerts par le Consortium. Ils coordonnent les Groupes d'utilisateurs qui se réunissent régulièrement pour partager leur expérience de la méthode.

Les équipes projet, qui dépendent du groupe Technique, vont et viennent selon les besoins propres au développement de la méthode. Les membres du Consortium qui travaillent dans ces équipes ont fourni tout le contenu de la méthode, de manière libre et volontaire. Ce qui fait la force de DSDM, c'est que cette méthode est uniquement fondée sur l'expérience et la connaissance des meilleures pratiques que possèdent les membres.

1.7 Retours d'information et commentaires

DSDM repose sur les personnes qui utilisent cette méthode afin de garantir que celle-ci constitue une bonne approche de la technique RAD. Si, durant la mise en œuvre de la

méthode, vous avez des idées à exprimer sur un domaine particulier, ne manquez pas de contacter le Consortium DSDM.

Pour de plus amples informations sur le Consortium, ses activités, ses membres et la procédure d'adhésion, veuillez nous contacter à l'adresse suivante :

« J'ai été très dubitatif au début concernant toute l'idée du RAD, mais maintenant je suis convaincu que le RAD a besoin d'un canevas, et que DSDM fonctionne. »

Secrétariat DSDM

Consortium DSDM
Kent House
81 Station Road
Ashford
Kent TN23 1PP, UK

Tél. : +44 (0)1233 661003
Fax : +44 (0)1233 661004
Site Web : <http://www.dsdm.org>

Secrétariat DSDM au Benelux

DSDM Benelux
Groenezoom 1
1171 JA Badhoevedorp, Netherlands

Tel: +31 (0)20 4499900
Fax: +31 (0)20 4499999
Président : Ron van Toledo
Secrétariat: Jolanda Teske

Secrétariat DSDM en France

DSDM France
6 rue des deux Communes
94300 Vincennes, France

Tel: +33 (0)1 40 68 46 77
Fax: +33 (0)1 40 68 40 20
Président : Ian Stokes
Trésorier : Benjamin Schwarz
Vice Président (Technique) : Jean-Yves Reynaud
Vice Président (Promotion) : David Sapiro
Vice Président (Certification) : Jacqueline Lautmann
Secrétariat : Ian Stokes
Email : dsdm@dsdm.asso.fr
Internet : www.dsdm.asso.fr

Partie 1 Les fondements de la méthode

2 Principes de base

Ce chapitre décrit les principes de base sur lesquels repose la méthode DSDM. Chacun de ces principes est appliqué, de façon appropriée, au cours des différentes étapes de cette méthode.

Principes

Commentaires

- | | | |
|------|--|--|
| I. | L'implication active des utilisateurs est indispensable. | L'approche DSDM est centrée sur l'utilisateur. En effet, si les utilisateurs ne sont pas fortement impliqués tout au long du cycle de vie du développement, un fossé peut se creuser au fur et à mesure que les décisions sont prises. Les utilisateurs peuvent alors ressentir que la solution finale leur est imposée par les développeurs et/ou leur encadrement. La place des utilisateurs n'est pas en dehors de l'équipe de développement, fournissant des informations et contrôlant les résultats mais au contraire directement au sein du processus de développement et en y participant activement. |
| II. | Pouvoir de décision des équipes DSDM. | Les équipes DSDM sont formées à la fois de développeurs et d'utilisateurs. Elles doivent être capables de prendre des décisions lorsque les besoins évoluent ou sont modifiés. Ces équipes doivent être en mesure de statuer et d'accepter certains niveaux de fonctionnalité, d'utilisation, etc. sans avoir recours trop fréquemment à leurs supérieurs hiérarchiques. |
| III. | Livraison fréquente de produits. | <p>Une approche basée sur les produits est plus flexible qu'une approche basée sur les activités.</p> <p>Une équipe DSDM travaille essentiellement sur des produits pouvant être livrés dans un délai négocié. Ceci permet à l'équipe de choisir la meilleure façon de réaliser les produits demandés dans le laps de temps alloué. En utilisant des délais courts, l'équipe peut facilement décider quelles activités sont nécessaires et suffisantes pour réaliser de bons produits.</p> <p>Remarque : le terme "produits" comprend les produits de développement intermédiaires, et pas uniquement les systèmes livrés.</p> |
| IV. | L'adéquation aux besoins est le critère | L'objectif principal de DSDM est de livrer |

essentiel pour que les produits livrés soient acceptés.

la fonctionnalité nécessaire en temps voulu. Si ce principe est acceptable, le système pourra être amélioré ultérieurement. Les pratiques habituelles privilégient plutôt l'application d'un cahier des charges et la conformité aux éléments déjà livrés, même si les besoins sont souvent imprécis, que les éléments précédemment livrés peuvent être imparfaits et que les besoins de gestion peuvent avoir changé depuis le début du projet.

- V. Il est nécessaire d'utiliser un développement itératif et incrémental pour obtenir une solution adaptée aux besoins.

La méthode DSDM permet de développer des systèmes de façon incrémentale. Ainsi, les développeurs peuvent tirer pleinement profit des retours utilisateurs. De plus, il est possible de livrer des solutions partielles pour répondre à des besoins de gestion urgents.

L'itération est propre à tous les développements de logiciels. DSDM en tient compte de façon explicite et utilise donc l'itération pour améliorer en permanence le système en cours de développement.

Lorsque le travail de reprise n'est pas pris en compte de façon explicite dans le cycle de vie du développement, il est nécessaire de revenir à une étape antérieure "terminée" d'un travail et les procédures de contrôle requises alors retardent le développement. Comme le travail de reprise est intégré dans le processus DSDM, le développement peut continuer plus rapidement pendant le processus d'itération.

- VI. Toutes les modifications effectuées au cours du développement sont réversibles.

Pour contrôler l'évolution de tous les produits (documents, logiciels, produits de test, etc.), l'état de chaque élément doit être connu à tout instant. Ce qui signifie que la gestion de la configuration doit être omniprésente.

Le "backtracking" (retour en arrière) est une caractéristique de DSDM. Cependant, il est parfois préférable de recommencer plutôt que de revenir en arrière. Ceci dépend de la nature de la modification et du contexte dans lequel elle a été effectuée. L'annulation de modifications n'est possible qu'au sein d'une étape incrémentale de développement.

- VII. Les besoins sont définis à un niveau global.
- VIII. Les tests sont intégrés à toutes les étapes du cycle de vie.
- IX. Une approche basée sur la collaboration et la coopération entre toutes les personnes intéressées par le projet est essentielle.
- La spécification et l'initialisation d'une référence des besoins à un niveau global sert à déterminer et à « geler » les objectifs et le périmètre du système, permettant ainsi une étude approfondie des implications de ces besoins. D'autres actualisations de la référence peuvent être effectuées en cours du développement mais le périmètre ne doit pas être modifié sauf de façon limitée. Les tests ne sont pas traités comme des activités indépendantes. Développé de façon incrémentale, le système est également testé et révisé de manière incrémentale par les développeurs ainsi que les utilisateurs. Grâce à cette approche, non seulement le développement évolue dans la bonne direction par rapport aux besoins industriels, mais il est techniquement adéquat. En amont dans la démarche DSDM, les tests permettent surtout de vérifier la conformité aux besoins du business et le respect des priorités. Vers la fin du projet, l'accent est mis plutôt sur le bon fonctionnement de tout le système.
- De par la nature même des projets DSDM, les spécifications détaillées n'ont pas nécessairement été définies lorsque les développeurs sont initialement sollicités pour effectuer le travail. La direction à court terme que doit prendre le projet devra donc être décidée rapidement, sans recourir à des procédures de contrôle, des modifications restrictives. Les personnes concernées par le projet incluent non seulement le métier et l'équipe de développement directement concernés, mais aussi d'autres équipes telles que celles de support et de gestion des ressources. Lorsque le développement est assuré par une société externe, le fournisseur et le sous-traitant doivent avoir pour objectif la mise en œuvre d'un processus aussi efficace que possible et offrant toute la souplesse voulue durant la phase d'avant-contrat et durant l'exécution des travaux faisant l'objet de ce contrat.

3 Présentation du processus DSDM

3.1 Le processus

Conformément au principe numéro cinq, le cycle de vie de développement mis en œuvre par la méthode DSDM est itératif et incrémental. Le système informatique n'est donc pas nécessairement livré en une seule fois mais de manière progressive, chaque nouvel élément fourni représentant un enrichissement fonctionnel du système. Ceci permet de traiter les besoins urgents du business en amont du processus et les fonctionnalités moins importantes durant les phases ultérieures. La nature itérative de la méthode DSDM permet aux utilisateurs de visualiser le travail en cours, de le commenter et de suggérer des modifications à chaque étape incrémentale. Cette approche nouvelle, par rapport à la méthode traditionnelle dite « en cascade », est rendue possible par les nouvelles technologies qui permettent aux développeurs de démontrer le bien fondé de leurs idées en visualisant les produits intermédiaires réalisés durant le cycle de développement du système.

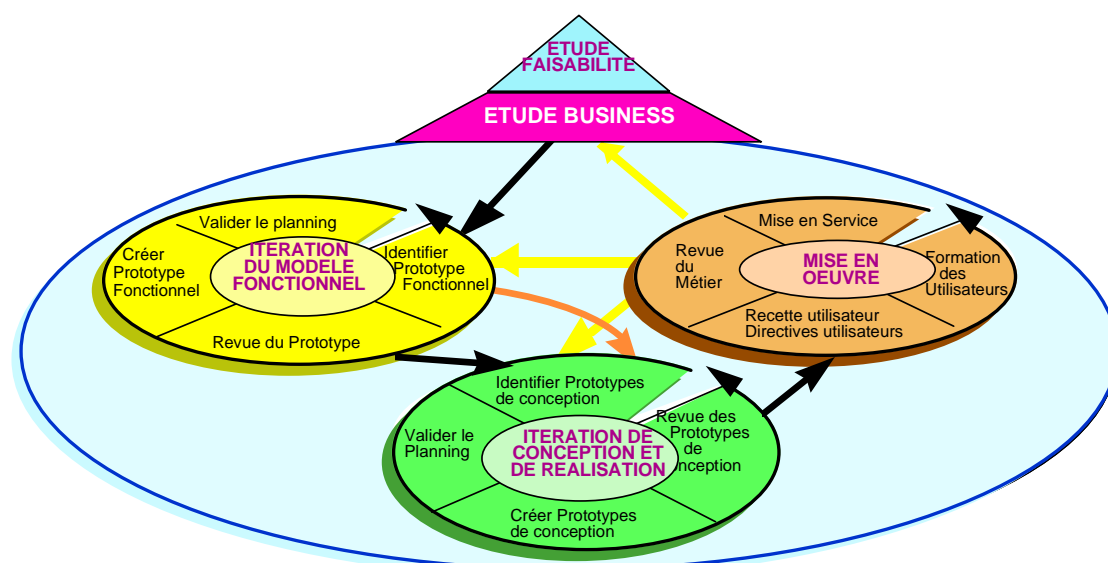


Figure 0.1 : Le processus de développement DSDM

DSDM est plus un canevas qu'une méthode. Elle n'indique pas comment les choses doivent être effectuées dans le détail mais fournit « l'ossature » du processus global ainsi qu'une description des processus et produits qui devront être adaptés à un projet ou à une entreprise particulière. Comme l'indique la figure 0.1, le processus de développement comporte cinq phases : L'Etude de Faisabilité, L'Etude du « Business », le Modèle Fonctionnel itératif, la conception et le développement itératifs et, pour finir, la Mise en œuvre dans l'environnement de travail. Ces cinq phases sont définies au chapitre 0.

L'Etude de Faisabilité et l'Etude du « Business » sont réalisées de manière séquentielle. Elles établissent les règles de base pour le reste du développement et, en raison de la nature itérative incrémentale de ce dernier, doivent être achevées avant que toute autre tâche soit accomplie pour un projet donné. La façon dont les trois phases suivantes peuvent converger ou se chevaucher est spécifique à chaque projet et dépend de plusieurs facteurs dont les principaux sont la nature du système développé et les outils utilisés à cet effet.

Chaque phase du processus de développement génère un minimum de produits. Chaque produit sert un objectif bien défini et fait intervenir un ensemble de critères de qualité au moyen desquels la réalisation de cet objectif peut être évaluée. La façon dont le produit sera développé et la nature de son contenu sont laissées à l'appréciation de l'entreprise ou de l'équipe projet. Les produits sont décrits au chapitre 13.

3.1.1 L'Etude de Faisabilité

Un aspect important de l'Etude de Faisabilité consiste à décider si DSDM est l'approche qui convient pour le projet considéré. Y sont également abordées les considérations habituelles telles que la définition du problème à résoudre et l'évaluation des coûts, ainsi que l'aptitude du système envisagé à résoudre le problème du métier, examinée d'un point de vue technique.

Dans la mesure où la méthode DSDM est généralement utilisée pour développer des systèmes dont le business a un besoin urgent, L'Etude de Faisabilité sera nécessairement courte, ne pouvant s'étendre au-delà de quelques semaines. Ceci peut avoir un impact sur l'établissement des normes permettant à l'entreprise de déterminer si le rapport de faisabilité est ou non acceptable. Durant les projets DSDM, il n'y a pas suffisamment de temps pour produire des documents volumineux, à moins que ceux-ci soient absolument nécessaires. Le Rapport de Faisabilité couvrira donc les sujets habituels mais de manière peu détaillée. La philosophie DSDM consiste à faire ce qu'il faut mais pas davantage.

Outre le Rapport de Faisabilité, L'Etude de Faisabilité engendre deux autres produits, l'un et l'autre destinés à étayer les conclusions de ce même rapport. Le premier est un Plan Global de développement, qui vient renforcer l'hypothèse que l'objectif fixé est réalisable et le second, un Prototype de Faisabilité. Le prototype est un produit facultatif qui permet de démontrer qu'une solution est possible. Ce peut être un produit logiciel, mais aussi un simple document.

3.1.2 L'Etude du Business

Une fois que DSDM a été choisie comme étant la méthode appropriée, L'Etude du Business (l'étude des besoins industriels) permet d'établir les bases de tout le travail ultérieur. Comme l'Etude de Faisabilité, celle-ci doit être aussi courte que possible, tout en autorisant une compréhension des besoins et des contraintes techniques de l'activité suffisamment approfondie pour permettre la poursuite en toute sécurité du projet.

Comme son nom le suggère, son principal objet est l'étude des processus à automatiser et des besoins en informations. Les contraintes de temps qui caractérisent un projet DSDM exigent que cette activité s'appuie sur une collaboration étroite et sur l'organisation d'Ateliers Facilités (travail collectif animé par un médiateur) regroupant un personnel compétent. Ainsi, ce dernier pourra rapidement mettre en commun ses connaissances et atteindre un consensus quant aux priorités de développement. De ces Ateliers Facilités résulte la Définition du Domaine Industriel qui identifie non seulement les processus de l'entreprise et informations associées mais aussi les classes (ou types) d'utilisateur qui seront concernées d'une manière ou d'une autre par la mise en place du système. A partir de ces classes, les individus qui participeront au développement seront identifiés et une décision sera prise avec leur encadrement quant à leur niveau d'implication. Les utilisateurs clés d'un projet DSDM sont les utilisateurs « ambassadeurs », ainsi appelé parce que leur rôle s'apparente à celui de diplomates. Ils appartiennent à l'équipe de projet informatique et constituent un canal de communication à deux voies entre l'entreprise et les équipes informatiques. Leur présence est indispensable pour assurer le succès d'un projet DSDM. En effet, leur responsabilité consiste non seulement à fournir les informations nécessaires au projet mais aussi de tenir l'ensemble de la communauté informatique au courant de son évolution. Les responsabilités qui incombent aux divers rôles DSDM sont récapitulées au chapitre 15.

Toutes les fonctionnalités identifiées dans la définition du profil de l'activité doivent être classées par ordre de priorité de façon à ce que les plus importantes soient développées en premier, les autres pouvant être ajoutées plus tard, si nécessaire. L'ordre de priorité reposera principalement sur les besoins du métier, mais il prendra également en compte les contraintes techniques. C'est ainsi qu'au vu de ces contraintes, l'équipe de développement pourra être amenée à traiter en priorité un besoin pourtant moins essentiel du point de vue du business. Certains besoins non fonctionnels, tels que la sécurité peuvent aussi avoir un impact sur l'ordre de priorité.

Comme certains modules logiciels commenceront à être développés dès la phase suivante (le Modèle Fonctionnel Itératif), il importe de bien comprendre non seulement la fonctionnalité développée mais aussi l'architecture système qui sera adoptée. C'est pourquoi la Définition de l'Architecture Système, qui décrit aussi bien les plates-formes opérationnelles et de

développement que l'architecture du logiciel développé (c'est à dire les principaux composants et interfaces associées), constitue un autre élément essentiel de L'Etude du Business. Comme tous les autres éléments produits durant l'Etude du Business, la Définition de l'Architecture Système est ajustée au cours des étapes ultérieures et peut évoluer au fil du projet.

Enfin, autre opération importante, le Plan Global produit durant L'Etude de Faisabilité est détaillé de manière à créer le plan Global de Prototypage. Ce plan décrit le mode de développement durant les phases « Modèle Fonctionnel Itératif » et « Conception et Réalisation Itératives » de manière plus précise. Il indique en outre comment les produits seront vérifiés et contrôlés au fur et à mesure de leur développement. Pour de plus amples informations sur ces vérifications et contrôles, reportez-vous aux chapitres 18 à 21.

3.1.3 Modèle Fonctionnel Itératif

Le Modèle Fonctionnel Itératif a pour objet d'affiner la définition des aspects du système informatique liés au business. Il consiste à décrire plus en détail les besoins de haut niveau, en termes de traitement et d'information, qui ont été identifiés durant l'Etude du Business. A cette fin, sont produits à la fois des modèles d'analyse standard et des modules logiciels.

Les phases « Modèle Fonctionnel Itératif » et « Conception et Réalisation Itératives » consistent l'une et l'autre en un cycle de quatre activités :

1. Identifier ce qui doit être produit.
2. Décider comment et à quel moment le faire.
3. Créer le produit.
4. Vérifier qu'il a été développé de manière appropriée (en réexaminant les documents à l'aide d'un prototype de démonstration, ou en testant une partie du logiciel).

Le Modèle Fonctionnel élaboré au cours de ces cycles intègre à la fois des modèles d'analyse et des modules logiciels comportant les principales fonctionnalités, et répondant à certains des besoins non fonctionnels, en particulier ceux liés à la facilité d'utilisation.

Les modules logiciels du Modèle Fonctionnel sont testés au fur et à mesure de leur production. Ceci comprend les tests techniques unitaires mais doit aussi inclure autant d'autres types de test que possible, y compris les mini-tests de recette réalisés par les utilisateurs ambassadeurs dès que de nouveaux composants sont disponibles. Les tests associés à la phase Modèle Fonctionnel auront nécessairement pour objet de déterminer ce que font les composants (même si leurs fonctions sont limitées) et si ces derniers peuvent ou non servir de base à un ensemble de fonctionnalités utilisable. Les aspects non fonctionnels tels que les performances sont testés durant la phase Conception et Réalisation Itératives. Sur le diagramme des processus, ceci est représentée par une flèche vers l'arrière qui va de la phase Conception et Réalisation Itératives à la phase modèle fonctionnel itératif.

Il est souvent plus facile, et en fait plus logique, de traiter en une seule fois les détails d'un domaine fonctionnel et ses aspects non fonctionnels avant de passer aux détails d'un autre domaine. Le degré de convergence des phases modèle fonctionnel itératif et Conception et Réalisation Itératives dépendra principalement de la manière dont l'application aura été « éclatée » en divers modules et des possibilités offertes par l'environnement de développement.

3.1.4 Conception et Réalisation Itératives

Les conditions préalables dont dépend le passage du modèle fonctionnel à la conception et au développement incluent l'accord sur un Prototype Fonctionnel qui peut être ou ne pas être automatisé. Le Prototype Fonctionnel peut ne représenter qu'une partie du modèle fonctionnel, ce qui veut dire que les activités de conception et d'élaboration ainsi que celles de prototypage fonctionnel peuvent être menées de front. De même, dans un grand projet DSDM, la mise en œuvre peut être échelonnée et une partie de cette dernière menée de front avec les activités de conception et d'élaboration.

La phase Conception et Réalisation Itératives est l'étape durant laquelle le système informatique est amené à un niveau de réalisation conforme à des standards suffisamment élevés pour qu'il puisse être placé en toute sécurité entre les mains des utilisateurs. Le produit le plus important est ici le Système Testé. Le diagramme des processus DSDM ne représente pas la phase de test comme une activité distincte dans la mesure où ces tests interviennent aussi bien durant la phase modèle fonctionnel itératif que durant la phase Conception et Réalisation Itératives. Certains environnements ou dispositions contractuelles exigent que des phases de test distinctes soient ajoutées à la fin de l'étape incrémentale de développement, mais celles-ci ne doivent en aucun cas être l'activité majeure qu'elle constituent dans les méthodes de développement plus traditionnelles. Les tests préconisés selon l'approche DSDM revêtent la même importance et nécessitent autant d'efforts, mais ils sont répartis tout au long du développement.

Le Système Testé ne répondra pas nécessairement à tous les besoins *identifiés*, mais il satisfera ceux qui ont été définis pour l'étape incrémentale en cours. Eu égard aux contraintes de temps, il aura été préalablement décidé de remettre à une date ultérieure le développement des parties moins essentielles du système informatique. Cependant, toutes les spécifications de base (ce que DSDM appelle « l'ensemble minimum utilisable ») devront être intégrées au système testé, avec autant d'autres éléments que le temps le permettra.

3.1.5 Mise en Œuvre

La phase de Mise en Œuvre couvre le passage de l'environnement de développement à l'environnement opérationnel. Elle inclut la formation des utilisateurs qui n'ont pas pris part au projet. Lorsque le système est livré sur une période étendue à une communauté d'utilisateurs dispersée, la phase de Mise en Œuvre peut être répétée.

Les produits de cette phase comprennent le Système Livré avec toute la documentation convenue, y compris la documentation utilisateur. La Documentation Utilisateur est finalisée durant cette phase mais doit avoir été commencée durant la phase Conception et Réalisation Itératives. L'une des responsabilités des utilisateurs ambassadeurs consiste à s'assurer de la qualité de la documentation et de la formation utilisateur, et que l'esprit de la méthode est fidèlement transmis.

L'autre produit de cette phase est le Document de Capitalisation de Projet qui est généré dès que le système informatique (ou l'étape incrémentale) est considéré comme achevé. Le Document de Capitalisation de Projet permet de résumer ce que le projet a accompli en ce qui concerne les objectifs à court terme. Tout particulièrement, il passe en revue tous les besoins qui ont été identifiés durant le développement et évalue la position du système par rapport à ces besoins. Il y a quatre résultats possibles (dont trois sont représentées par des flèches en boucle sur le diagramme des processus DSDM de la figure 0.1). Ces résultats sont les suivants :

1. Tous les besoins ont été satisfaits. Aucune autre tâche n'est donc nécessaire.
2. Un aspect fonctionnel important a été provisoirement ignoré durant le développement afin de pouvoir livrer à la date convenue. Il est donc nécessaire de revenir à la phase Etude du business et de dérouler le processus à partir de là.
3. Une fonctionnalité à faible priorité a été délibérément ignorée pour des raisons de délais. Celle-ci doit à présent être ajoutée, ce qui implique un retour à la phase Modèle fonctionnel itératif du processus.
4. Un aspect technique mineur a également été délaissé en raison de l'urgence mais peut à présent être traité en revenant à la phase Conception et Réalisation Itératives du processus.

3.2 Assurer le bon fonctionnement du processus

Une définition souple des besoins

La quatre résultats ci-dessus illustrent parfaitement les différences entre DSDM et les méthodes de développement de logiciel traditionnelles. Selon ces dernières, la spécification des besoins est fixe et le logiciel livré répond à l'intégralité de ces besoins. Les ressources et le temps nécessaires pour parvenir à ce résultat varient tout au long du développement. Avec DSDM, il se passe exactement le contraire : le temps de développement est fixe pour la durée de vie du projet et les ressources sont, autant que possible, fixes également mais, la spécification des besoins peut évoluer. D'où la nécessité de hiérarchiser les objectifs à mesure qu'il sont mis en évidence durant l'Etude du Business et affinés durant le cycle de vie.

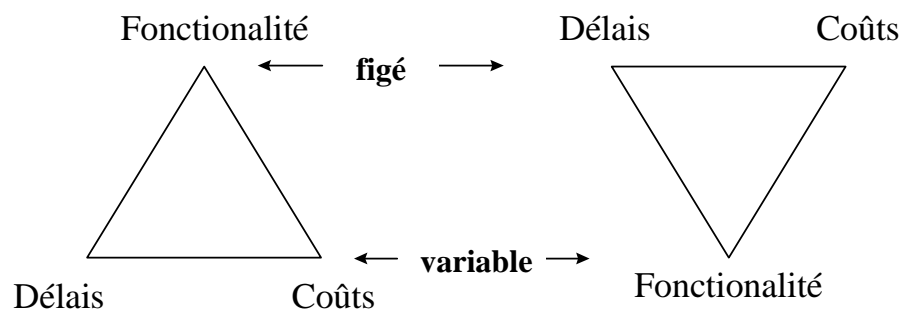


Figure 0.2 : Différences entre les approches traditionnelles et DSDM

Le niveau de flexibilité en matière de spécifications des besoins influe considérablement sur les processus et contrôles du développement, ainsi que sur l'acceptabilité du système informatique. L'un des neuf principes de base de DSDM stipule que la satisfaction des besoins industriels est le principal critère d'acceptabilité des produits livrés. Cette conception du développement s'oppose à celle qui consiste à satisfaire toutes les demandes secondaires d'une spécification de besoins, tout en adoptant une approche orientée vers la qualité.

Le mécanisme DSDM de base qui permet de gérer la flexibilité des besoins est le « timeboxing », décrit plus en détail au chapitre 7. Dans tout projet RAD, il y a une date d'achèvement fixe qui détermine le délai de réalisation ou « timeboxe ». DSDM affine le concept de timeboxing en subdivisant la timeboxe globale en timeboxes intermédiaires de deux à six semaines. Ce sont ces timeboxes qui témoignent de l'avancement du projet et qui lui confèrent sa dynamique.

Canaux de communication

DSDM permet de réduire considérablement les délais de livraison en raccourcissant les canaux de communication entre les utilisateurs et le personnel informatique, entre les analystes et les concepteurs, entre tous les membres de l'équipe, ainsi qu'entre les différents niveaux hiérarchiques. Les mécanismes qui permettent d'y parvenir varient d'un canal à l'autre. Par exemple, la communication entre les développeurs et les utilisateurs est facilitée par l'utilisation du prototypage qui remplace avantageusement la création de documents volumineux. C'est la raison pour laquelle DSDM définit un Modèle fonctionnel plutôt qu'une spécification fonctionnelle.

Les documents dans la méthode DSDM

DSDM utilise une approche particulière pour déterminer quel type de documentation est nécessaire pour un projet donné. Dans les approches traditionnelles, la documentation produite sert presque exclusivement à transmettre des idées d'un développeur vers un autre, ou des développeurs vers les utilisateurs. En maintenant, tout au long du développement, des équipes réduites composées à la fois d'utilisateurs et de développeurs, DSDM rend de tels documents largement superflus.

Pour ce qui concerne la documentation technique, il est conseillé de se conformer aux usages de la société, mais en revoyant, si possible, ces dernières à la baisse afin de minimiser les

obstacles superflus de nature à ralentir le projet. D'un point de vue technique, DSDM fournit des lignes directrices pour déterminer quels types de documentation il est nécessaire de contrôler et pour quelle raison. Le critère de base est le suivant : un document doit être contrôlé s'il est indispensable aux développeurs pour assurer la mise en œuvre ou s'il est nécessaire à des fins de maintenance. L'écart entre la documentation jugée indispensable et celle qui s'avère effectivement utile une fois le produit livré est souvent considérable. Ne réaliser que le minimum nécessaire pour assurer la sécurité, tel est l'un des moyens préconisés par DSDM pour livrer les systèmes plus rapidement tout en respectant les exigences de qualité. Certains documents peuvent être nécessaires à des fins d'audit.

Une question souvent posée est celle-ci : « Quelle est la quantité de documentation réellement suffisante et indispensable ? ». La documentation peut être considérée comme suffisante lorsque qu'un document supplémentaire n'ajouterait rien à la compréhension de ce qui a été élaboré. Par exemple, il est de coutume dans les projets traditionnels d'investir beaucoup d'efforts dans l'écriture, la révision, la validation et la mise à jour de spécifications de besoins détaillées qui, bien souvent, ne sont d'aucune utilité long terme.

4 Quand utiliser DSDM

Vous pouvez utiliser la méthode DSDM intégralement la plupart du temps et partiellement tout le temps.

4.1 Introduction

La décision d'utiliser ou de ne pas utiliser DSDM pour un projet de développement n'est pas à prendre à la légère. Certains projets semblent, de toute évidence, faits pour cette méthode. D'autres requièrent plus de réflexion quant à déterminer si l'utilisation de DSDM représenterait un avantage pour cette activité.

Afin d'aider le meilleur choix, DSDM fournit un « Filtre d'Adaptation », décrit à l'annexe A. Ce filtre prend en compte les critères de succès essentiels de DSDM et les caractéristiques de projet qui sont particulièrement propices à l'utilisation de cette méthode. Chaque projet potentiel doit être évalué individuellement au moyen du filtre. Si le projet satisfait à la plupart des critères fournis par ce dernier, DSDM peut être considérée comme l'approche de développement appropriée. L'impossibilité de répondre à tous les critères ne disqualifie pas systématiquement DSDM mais il sera alors nécessaire de développer des stratégies de gestion des risques pour tenir compte de ces divergences.

Les critères inclus dans le Filtre d'Adaptation ne sont fournis **qu'à titre indicatif** et ne doivent en aucun cas être considérés comme exclusifs. La plupart des entreprises voudront enrichir ce filtre de leurs propres critères, reflétant leur expérience de la méthode dans leur environnement spécifique.

L'objet de DSDM n'est pas uniquement de développer de nouveaux systèmes. Elle permet également d'améliorer des systèmes existants dans la mesure où ceux-ci passent avec succès le test du filtre d'adaptation.

La légitimité de DSDM comme méthode de développement doit être soigneusement contrôlée durant les premières phases du projet. S'il apparaît que le système ne présente pas toutes les caractéristiques voulues mais que l'utilisation de DSDM constitue tout de même un avantage, il est, comme nous l'avons vu plus haut, impératif de développer des stratégies de gestion des risques qui prennent en compte les aspects plus problématiques du projet.

4.2 Principaux critères de succès d'un projet de développement DSDM

Les principaux critères de succès de la méthode DSDM sont les suivants :

1. **Acceptation de la philosophie DSDM avant de commencer le travail.** Il est important que les responsables comprennent et acceptent la philosophie DSDM. Ceci inclut l'acceptation par toutes les parties que le produit livré ne constituera pas l'intégralité du produit spécifié. Une adhésion continue à la philosophie DSDM tout au long du projet est également primordiale.

Ceci découle des principes de base de DSDM.

2. **Pouvoir de décision des utilisateurs et des développeurs de l'équipe de développement.** L'encadrement doit aussi accepter de responsabiliser les utilisateurs confirmés de l'équipe de développement ou de participer eux-mêmes au travail de cette équipe, afin d'éviter que le travail ne soit ralenti par l'attente de décisions prises à l'extérieur de l'équipe. Le personnel utilisateur doit se sentir libre de prendre des initiatives sans en référer à des supérieurs hiérarchiques extérieurs à l'équipe. De même, les développeurs de l'équipe doivent pouvoir prendre des décisions, notamment en ce qui concerne la faisabilité technique.

Ceci découle directement du principe II.

3. **L'encadrement doit tout mettre en œuvre pour assurer une réelle implication de l'utilisateur final.** La participation volontaire et l'implication des utilisateurs sont absolument essentielles. Si cette implication n'est pas acquise à l'issue de L'étude de faisabilité, il est préférable d'abandonner la méthode DSDM au bénéfice d'une approche plus traditionnelle.

Ceci découle du principe I.

4. **Livraison incrémentale.** L'entreprise doit être à même de livrer les systèmes de manière incrémentale.

Ceci découle des principes III et V.

5. **Communication aisée entre les développeurs et les utilisateurs.** Idéalement, les développeurs et les utilisateurs seront rassemblés en un même lieu, dans leur propre environnement dédié, à l'abri des interruptions quotidiennes. Mais cette configuration optimale n'est pas essentielle tant que des contacts fréquents entre les uns et les autres sont maintenus tout au long du projet.

Ceci découle des principes I et IX.

6. **Stabilité de l'équipe.** Compte tenu du chevauchement de compétences requis (dû aux interactions constantes entre les analystes du business, les concepteurs de système et les programmeurs) et de la rapidité du développement, des changements de personnel fréquents compromettraient gravement le succès du projet. Bien entendu, il est toujours possible d'appeler des spécialistes en renfort durant le projet mais le noyau de base de l'équipe doit rester le même tout du long.

Ceci découle en partie du principe IX.

7. **Compétence de l'équipe de développement.** L'équipe doit être constituée de personnes compétentes aussi bien sur le plan du business qu'en termes d'environnement technique. Tous les membres de l'équipe doivent également faire preuve de qualités relationnelles.

Ceci découle en partie du principe IX.

8. **Taille de l'équipe de développement.** L'équipe DSDM doit être restreinte de manière à minimiser les problèmes d'organisation et de communication tout en développant le sentiment de propriété. Voir le chapitre 14.

9. **Des relations commerciales constructives.** Lorsque les développeurs et les utilisateurs appartiennent à des entreprises différentes et que le développement est couvert par un contrat formel, ou lorsque les développeurs font partie de la même entreprise mais opèrent dans le cadre d'un contrat de service, les relations entre les intervenants doivent s'adapter à l'évolution des besoins sans que cela se traduise par des changements organisationnels lourds et coûteux.

Ceci découle du principe IX.

10. **Technologie de développement.** La technologie de développement doit être compatible avec l'approche DSDM. Voir le chapitre 24.

4.3 Caractéristiques des systèmes pour lesquels la méthode DSDM est préconisée.

Outre les facteurs de succès décrits plus haut, DSDM se montrera particulièrement efficace pour les systèmes offrant les caractéristiques suivantes :

- **Interactivité, fonctionnalités clairement démontrables au niveau de l'interface utilisateur.** La méthode DSDM s'appuie principalement sur le prototypage incrémental et une forte implication des utilisateurs. Il est donc important que ces derniers puissent évaluer facilement les fonctionnalités en visualisant et en manipulant des prototypes en état de fonctionnement. D'où l'importance de développer des fonctionnalités démontrables. L'interface utilisateur doit comporter des écrans ainsi que des états et offrir la possibilité d'imprimer des fichiers (l'impression des fichiers peut être fournie de manière provisoire uniquement, à des fins de démonstration et de vérification des fonctionnalités).

Ceci découle en partie des principes I et V.

- **Le groupe utilisateur est clairement défini.** En l'absence d'un groupe utilisateur clairement défini, le développement risque d'être mal orienté ou, pire encore, de négliger entièrement un aspect important du projet.

Ceci découle du principe I.

- **Il est possible de réduire la complexité en la décomposant ou en l'isolant du système.** Le niveau de complexité est souvent difficile à déterminer à l'avance et peut varier d'un projet à un autre. De plus, certaines interactions entre différents composants peuvent être difficiles à identifier dans un premier temps.

DSDM peut être utilisé pour un système complexe à condition que l'application puisse être décomposée de manière à réduire cette complexité.

Par exemple, lorsqu'une application requiert une modélisation statistique complexe, le projet peut être envisagé selon deux approches distinctes. L'une consiste à utiliser les éléments de modélisation statistique existants et dûment testés, et l'autre à développer les modèles à partir de zéro. Si la première de ces démarches est tout à fait acceptable dans un projet DSDM, la seconde représente un risque, sauf s'il est possible de réduire la complexité en décomposant le système concerné en plusieurs modules ou de la rendre transparente pour l'utilisateur via l'interface.

Si la complexité existe mais qu'elle peut être isolée du reste du système, il sera toujours possible par la suite de la gérer d'une manière plus formelle et de rétablir l'intégration du système.

Ceci découle en partie du principe V.

- **Si l'application est volumineuse, elle peut être subdivisée en plusieurs modules fonctionnels.** Si le système proposé est très volumineux, il doit être possible de le subdiviser en plusieurs entités plus faciles à gérer, chacune correspondant à une fonctionnalité distincte. Ces modules pourront être livrés simultanément ou de façon incrémentale. Certaines des fonctionnalités pourront d'ailleurs être fournies selon les méthodes traditionnelles dites "en cascade". Durant le développement de chaque sous-projet, il est toutefois essentiel de ne jamais perdre de vue l'architecture globale du système. En d'autres termes, tous les aspects du projet doivent être contrôlés et coordonnés.

Remarque : le nombre d'équipes DSDM travaillant simultanément sur un même projet doit être réduit au maximum, sans dépasser le nombre maximal de membres dans une équipe que recommande DSDM. Cette démarche facilite le contrôle global du projet et minimise le travail d'intégration. Les tâches liées à la gestion et à la coordination des timeboxes ne doivent en aucun cas être sous-estimées.

Ceci découle en partie des principes III et IV.

- **Le projet est circonscrit dans le temps.** Une date butoir doit être fixée pour la réalisation du projet. S'il n'y a aucune raison particulière de définir une date de fin, les risques de dépassement de délai sont plus grands et les avantages fondamentaux de DSDM risquent d'être perdus. Les contraintes de temps sont d'ordinaire dictées par les objectifs de l'entreprise mais elles peuvent aussi être liées à la disponibilité du personnel pendant une période donnée.

Ceci découle en partie du principe III.

- **Les besoins peuvent être « priorisés ».** Il convient de hiérarchiser les besoins au moyen des règles MoSCoW (voir le chapitre 7) (nous avons déterminés *FISPE* en français)

Remarque : Si les besoins étaient déjà définis lorsque la décision d'utiliser la méthode DSDM a été prise, il sera nécessaire d'obtenir l'accord du « propriétaire du projet » pour les rendre modifiables, puis les reconsidérer au moyen des règles MoSCoW (*FISPE*)

Ceci découle en partie du principe VII.

- **Les besoins ne sont pas clairs ou sujets à des modifications fréquentes.** Dans les périodes de changements rapides, la spécification des besoins en début de projet peut s'avérer difficile, ce qui rend les approches traditionnelles inadaptées. L'approche DSDM a été spécifiquement conçue pour faire face à des besoins susceptibles de changer et d'évoluer en cours de projet.

Bon nombre d'applications se prêtent mal à des spécifications anticipées car les utilisateurs ne savent pas toujours exactement ce qui sera nécessaire en début de projet. C'est, par exemple, ce qui se produit lorsque la conception d'un processus du business est modifiée et que de nouveaux produits et services sont développés. La méthode DSDM se prête parfaitement à l'élaboration de telles applications dans la mesure où elle permet aux utilisateurs de changer d'avis au cours du développement.

Si les détails des besoins propres aux systèmes sont trop clairement définis, il sera plus difficile de gagner du temps avec les utilisateurs en utilisant le prototypage comme mode d'expression des besoins. Si tous les besoins sont fixes, il sera moins aisé d'en laisser certains de côté à titre temporaire afin de pouvoir livrer dans les délais prévus.

Ceci découle des principes V et VI.

4.4 Caractéristiques des systèmes qui requièrent des précautions particulières quant à l'application de la méthode DSDM

Bon nombre de systèmes ne semblent ne pas offrir les caractéristiques et gages de succès requis par l'approche DSDM. Toutefois, si l'environnement de support de développement est suffisamment robuste et l'équipe de développement suffisamment expérimentée dans la mise en œuvre de cette méthode, l'application des techniques DSDM peut parfois s'avérer bénéfique.

Avec certains systèmes, l'utilisation de DSDM devra s'accompagner de précautions particulières. Ces systèmes possèdent généralement une ou plusieurs des caractéristiques suivantes :

- **Applications temps réel et/ou de contrôle des processus.** Le traitement de ce type d'application n'est pas visible au niveau de l'interface utilisateur, est souvent très complexe et ne peut pas être décomposé en modules plus petits. De plus, un important travail de vérification et de validation est souvent indispensable. L'approche DSDM est donc normalement considérée comme inadaptée pour ces applications.
- **Applications nécessitant que les besoins soient entièrement définis avant d'écrire la première ligne programme.** Pour certains projets, il est demandé ou considéré comme essentiel que les besoins soient entièrement spécifiés et approuvés avant que ne démarre le développement proprement dit. Dans de telles conditions, l'utilisation de DSDM peut s'avérer désastreuse. DSDM étant une approche itérative, l'obligation de définir à l'avance tous les besoins mettrait sérieusement en péril la dynamique et la synergie de l'équipe de projet DSDM. Le temps requis pour la définition des besoins entraînerait des retards dans l'exécution du projet, le rôle des utilisateurs au sein de l'équipe projet deviendrait mineur et l'encadrement pourrait sérieusement remettre en question la crédibilité de DSDM en tant que méthode de développement rapide. Des techniques telles que les « ateliers facilités » auraient toujours leur utilité pour la spécification des besoins mais la méthode DSDM ne pourrait pas être appliquée dans son ensemble.

- **Applications pour lesquelles la sécurité est primordiale.** Tout comme la spécification détaillée des besoins, les validations et les vérifications approfondies qu'exigent la sécurité des systèmes stratégiques s'inscrivent mal dans la logique de développement itérative de DSDM.

Notez cependant qu'un membre du Consortium a réussi à mettre en œuvre un tel système. Dans des conditions semblables, les compétences et l'expérience du coordinateur technique sont particulièrement mises à contribution en amont du projet, lorsqu'il s'agit d'en déterminer la faisabilité selon la méthode DSDM. Pour de plus amples informations sur le rôle du coordinateur technique, reportez-vous au chapitre 15.

- **Nécessité de livrer des composants réutilisables.** Certains projets exigent que les composants produits soient réutilisables. Ceux-ci doivent alors être exactement conformes. DSDM ne semble pas la méthode appropriée lorsqu'il s'agit de développer des composants volumineux réutilisables et absolument conformes aux spécifications. L'approche DSDM est cependant recommandée si les composants réutilisables offrent une grande modularité et peuvent être élaborés de manière incrémentale sans dommage pour le reste de l'application.

5 Culture de projet

5.1 Gestion de l'approche DSDM

La gestion des projets traditionnels est surtout une affaire de contrôle : contrôler que le projet ne s'écarte pas des spécifications, contrôler les ressources, etc. La gestion selon DSDM consiste à autoriser en permanence les changements, tout en rectifiant continuellement le cours du projet de manière à maintenir le cap sur l'objectif ciblé : une date de livraison fixe pour la livraison d'un système utilisable. Pour appliquer la méthode DSDM avec succès, l'entreprise sera peut-être contrainte d'effectuer des modifications aux niveaux organisationnel, social et technique, tous ces éléments ayant une incidence sur la menée du projet.

Avec DSDM, les utilisateurs et les développeurs travaillent conjointement pour produire un système à la fois conforme aux besoins de l'entreprise et facile à maintenir. Ceci implique une démarche entièrement nouvelle de la part des chefs de projet qui ont l'habitude de contrôler étroitement leurs développeurs, ceux-ci risquant de se sentir mal à l'aise face à une approche qui repose sur un consensus utilisateur-développeur. L'organisation des activités journalières d'un projet DSDM peut, il est vrai, constituer un challenge pour tout chef de projet.

L'approche de collaboration des projets DSDM permet de développer une culture selon laquelle « personne n'est à blâmer » qui, en cela, se distingue des méthodes de gestion de projet traditionnelles. Ces dernières supposent généralement que les besoins sont fixes, au moins pour la durée du projet ; il en résulte que les chefs de projet passent souvent beaucoup de temps à prévenir tout écart par rapport aux spécifications et à prévenir les dépassements en termes de temps ou de budget. Si le système informatique élaboré ne répond pas aux besoins spécifiés, les développeurs en sont rendus responsables. Si, en revanche, il répond aux besoins définis mais se montre d'une utilité restreinte pour le business, ce sont alors les utilisateurs qui sont blâmés. Ces attitudes, propres à engendrer les conflits, sont à proscrire dans les projets DSDM. Le chef de projet DSDM a pour responsabilité de renforcer la collaboration de manière à inhiber ce type de réaction.

5.2 Travail par rapport à des objectifs

L'équipe DSDM est constituée à la fois de développeurs et d'utilisateurs qui travaillent en étroite collaboration. DSDM recommande de définir des objectifs clairs pour cette équipe et de responsabiliser ses membres pour leur permettre de les réaliser à leur manière, plutôt que de planifier chaque tâche de façon détaillée. L'approche basée sur les tâches ne fonctionne pas dans un projet DSDM. En effet, la planification et le contrôle de tâches séquentielles cadrent mal dans un environnement où l'accent est mis sur le changement et l'inévitabilité du processus itératif. Pour que l'approche DSDM puisse être appliquée avec succès, il faut que les objectifs soient mesurables d'une façon ou d'une autre et que les équipes demeurent petites, avec des utilisateurs et des développeurs qui œuvrent ensemble vers les mêmes objectifs. La façon dont l'équipe s'organise pour atteindre un objectif est sans importance. Le seul critère de succès est résumé par la question suivante : « les objectifs du business ont-ils été atteints ? » (voir les chapitres 19 et 21). Exécuter une tâche (par exemple, l'écriture d'un programme) dans le temps imparti ne présente aucun intérêt si cette tâche ne répond pas aux besoins du business.

5.3 Equipes auto-dirigées

Les projets DSDM sont extrêmement dynamiques, génèrent des prototypes qui évoluent rapidement et sont ponctués de fréquentes démonstrations aux utilisateurs. Le temps est trop mesuré pour gérer cette activité quotidienne selon une approche bureaucratique. Les développeurs et les utilisateurs sont informés en permanence de l'état du projet car ils travaillent en équipe réduite et à proximité les uns des autres (voire, idéalement, dans la même pièce). Une réunion quotidienne d'une demi-heure pour faire le point sur les réalisations de la

journée constitue un bon moyen de tenir chaque membre de l'équipe au courant de l'évolution du projet.

Les équipes DSDM sont auto-dirigées. La table 5.1 montre les différences entre ce type d'équipe et les équipes gérées de façon plus étroite :

Equipes contrôlées	Equipes autonomes
Application des directives	Prise d'initiative
Recherche de récompenses individuelles	Esprit d'équipe
Priorité accordée aux objectifs de bas niveau	Priorité accordée aux solutions
Compétition	Coopération
Fin d'activité une fois le but atteint	Amélioration continue
Gestion des situations d'urgence	Prévention des situations d'urgence

Table 5. 1: Equipes contrôlées et équipes autonomes

5.4 Collaborer de manière efficace

La majorité des problèmes rencontrés durant le développement d'un système est imputable à un défaut de communication entre l'utilisateur final et les développeurs. Dans de nombreuses entreprises, les processus de développement de système sont conçus de manière à réduire le risque que les uns et les autres soient tenus responsables de ces mêmes problèmes.

Avec l'approche de collaboration préconisée par DSDM, les interactions entre les utilisateurs et les développeurs durant l'exécution des tâches sont multipliées. Des mécanismes de retour d'information sont mis en place afin de réagir rapidement aux modifications de l'environnement qui ont une incidence sur la nature et l'objet des tâches de développement effectuées. Ceci présente notamment l'avantage de réduire les besoins de documentation formelle pour chaque problème traité et, partant, la nécessité de placer la documentation détaillée sous contrôle de modification. De plus, la poursuite individuelle d'une perfection souvent illusoire est inhibée par la culture d'équipe.

Pour assurer le développement rapide des applications, il est essentiel que la communication entre les intervenants soit claire et concise. Malheureusement, les développeurs et les utilisateurs ne possèdent généralement aucune compétence particulière en matière de communication. La plupart des gens pensent que c'est quelque chose que tout le monde maîtrise naturellement, ce qui est une erreur. Les membres d'une équipe de développement dynamique doivent être d'excellent communicateurs. Les « bidouilleurs » d'arrière-salle qui sont incapables ou peu désireux d'expliquer les problèmes techniques d'une manière compréhensible pour l'utilisateur de base, ne sont pas qualifiés pour collaborer avec des utilisateurs au sein d'une équipe DSDM (le recours à des techniciens experts est important mais ceux-ci ne sont pas nécessairement des membres à part entière de l'équipe de développement). De même, les utilisateurs doivent être capables d'exprimer leurs besoins d'une manière intelligible pour les personnes étrangères à leur domaine.

Les projets DSDM doivent mettre en œuvre un processus de communication informel mais faisant l'objet d'une planification dont les réunions quotidiennes mentionnées plus haut sont un exemple. Le choix d'une petite équipe opérant dans un même lieu (idéalement, à proximité des autres utilisateurs) permet de minimiser les problèmes de communication qui caractérisent les groupes de travail plus étoffés. De plus, avec des équipes physiquement dispersées, il est impossible de développer l'esprit d'équipe prôné par cette approche. Pour résumer, dans le projet DSDM idéal, tous les membres des équipes opèrent dans l'environnement de l'utilisateur et la taille de ces équipes demeure réduite. Si possible, les projets doivent être subdivisés et répartis entre de petites équipes autonomes.

Pour assurer le succès d'un projet, il doit être également possible, si nécessaire, de retirer l'équipe de développement de son environnement de travail ; par exemple, à l'occasion de « ateliers facilités » (voir le chapitre 8).

« Nous ne pouvions pas accueillir les développeurs chez nous, donc nous nous sommes rencontrés, mais entre ces visites, la gestion du contact était difficile. Il n'est pas en effet toujours facile de visualiser ce que l'informaticien veut nous expliquer au moyen du téléphone. »

6 La composante « business » des équipes DSDM

6.1 Introduction

Les systèmes informatiques appartiennent au personnel du business et non aux développeurs. L'approche DSDM ne peut fonctionner sans l'implication de l'utilisateur car c'est lui qui détient le verdict final quant au succès du projet et à l'acceptabilité du système. C'est également lui qui décide si le système développé est bien celui qui a été demandé. Les responsables d'entreprise doivent par conséquent accepter de décharger le personnel choisi de ses responsabilités pendant des périodes prolongées (voir à plein temps si le projet l'exige) pour lui permettre de collaborer avec les équipes de développement DSDM.

Ce partenariat entre les développeurs et les utilisateurs implique que l'encadrement donne aux utilisateurs dotés de l'autorité, de la volonté, du sens des responsabilités et des compétences voulues le pouvoir de prendre des décisions au nom de leur communauté. L'exécutif sponsor (le parrain) joue un rôle essentiel en permettant au personnel clé de se libérer de ses occupations pour venir rejoindre les équipes DSDM en tant qu'utilisateur ambassadeur (le rôle d'exécutif sponsor - parrain - revient à la personne qui détient les plus hautes responsabilités dans l'entreprise pour tout ce qui touche les domaines concernés par le projet).

Une fois impliqués dans un projet DSDM, les utilisateurs sont plus désireux de coopérer avec le personnel informatique parce qu'ils ont une meilleure compréhension de ce qui est accompli et de l'importance de leur contribution au projet. Cependant, les attentes des utilisateurs vont parfois bien au-delà de ce que permet techniquement de réaliser l'approche de collaboration. Il est donc essentiel que les développeurs maintiennent ces aspirations dans des limites raisonnables et que pour leur part, les utilisateurs fassent en sorte que les développeurs ne perdent jamais de vue les besoins du business.

Outre l'exécutif sponsor (parrain), trois types d'utilisateur jouent un rôle clé dans l'approche DSDM. Il s'agit de l'utilisateur ambassadeur qui travaille au sein de l'équipe, des Utilisateurs conseillers qui fournissent une autre perspective sur des aspects particuliers du business et du visionnaire qui fait en sorte que la vision globale (en d'autres termes les objectifs) de l'entreprise soit connue et maintenue tout au long du projet. Les responsabilités correspondant à ces différents rôles sont détaillées au chapitre 15.

6.2 Utilisateurs ambassadeurs

Les utilisateurs appelés à devenir des membres de l'équipe de développement doivent être représentatifs de toute la communauté des utilisateurs et perçus comme tels par cette dernière (et pas seulement par l'encadrement). Ils doivent être investis de l'autorité nécessaire pour prendre des décisions et guider le travail des développeurs. Ceci est d'autant plus important que les contraintes de temps peuvent entraîner des modifications dans la stratégie de développement. Les utilisateurs choisis devront avoir une attitude positive vis-à-vis du développement. Ils seront en effet exposés à un flot continu d'idées nouvelles et de technologies pouvant s'avérer intimidantes.

Les développeurs ne doivent pas décider de manière arbitraire de ce qui est requis à une quelconque étape du projet. La présence d'utilisateurs au sein même de l'équipe devrait permettre d'éviter cet écueil mais parfois, l'attitude arrogante de certains informaticiens vis-à-vis des utilisateurs peut avoir des répercussions négatives sur la mise en œuvre de DSDM. Ceci est particulièrement dommageable car, alors, les utilisateurs peuvent avoir l'impression que leurs opinions ne sont pas suffisamment prises en considération et se désolidariser du projet de développement.

Le point de vue d'un utilisateur ambassadeur peut être continuellement ignoré. Par exemple un autre utilisateur plus opiniâtre peut réduire tous les autres au silence ou l'utilisateur ambassadeur peut avoir peur d'émettre une opinion. Si l'une de ces situations se présente, les développeurs doivent donner à l'utilisateur ainsi mis de côté la possibilité de s'exprimer et convaincre les vociférateurs qu'il est important de tenir compte de toutes les opinions.

L'essentiel étant de faire en sorte que les utilisateurs et les développeurs puissent travailler ensemble de manière efficace et fonctionner en tant qu'équipe.

Si les décisions prises par les utilisateurs travaillant au sein de l'équipe sont annulées par un autre utilisateur extérieur à cette équipe, il convient soit d'intégrer ce dernier (ou une personne le représentant) à l'équipe projet, soit d'obtenir de l'encadrement des directives claires concernant les prises de décision.

Si les utilisateurs ont du mal à s'entendre rapidement sur la marche à suivre ou si certains utilisateurs ambassadeurs semblent ne pas participer activement aux délibérations, c'est sans doute que les participants sont trop nombreux. Il est alors conseillé de réduire leur nombre tout en s'efforçant de maintenir un capital de connaissance des processus du business suffisant au sein de l'équipe.

6.3 Le visionnaire

Le « visionnaire » possède une « vision » globale des objectifs qui sont à l'origine du projet et dont ce dernier ne doit en aucune manière dévier durant le développement (bien que ce soit malheureusement le cas pour bon nombre de projets). Une fois le développement lancé, il peut être tentant de n'impliquer que le personnel qui sera appelé à utiliser le système sur une base quotidienne mais qui ne possède pas toujours une compréhension suffisante des objectifs du business. Ceci peut engendrer une « dilution » des objectifs initiaux et une tendance à la re-spécification du système à remplacer. Il est primordial que le Visionnaire soit présent à chaque événement important faisant intervenir les utilisateurs ou chaque fois que les responsables prennent des décisions concernant l'évolution du projet.

Il peut arriver au cours du développement que le système s'oriente vers quelque chose qui ne semble plus répondre aux besoins du business ou qui paraisse aller au-delà ou en deçà de ces mêmes besoins. Il est alors essentiel de consulter le Visionnaire afin de s'assurer de la validité de la direction prise.

Bien souvent les rôles de visionnaire et l'exécutif sponsor sont, tout naturellement, assumés par une seule et même personne.

6.4 Utilisateurs conseillers

Les utilisateurs conseillers ne participent pas au projet à temps complet mais il est fait appel à eux lors des sessions de prototypage et de test importantes. Il est essentiel d'obtenir des informations de première main de la part des utilisateurs et de développer chez la vaste communauté de ces utilisateurs un sentiment d'appropriation et de participation qui, par la suite, facilitera le déroulement des phases de mise en œuvre et d'acceptation. Par exemple, une bonne utilisation des utilisateurs conseillers consisterait à permettre à un panel de personnes investis de ce rôle d'assister à des démonstrations de prototypes organisées sous forme d'ateliers. Ainsi, toute la gamme de compétences et d'expérience de ces utilisateurs peut être mise à profit dans un environnement dynamique.

En dehors des événements programmés, les utilisateurs conseillers communiquent leurs informations et leur point de vue par l'intermédiaire des utilisateurs ambassadeurs.

Visionnaire : « Vraiment, j'apprécie mon implication. Visualiser des choses très tôt nous fait déborder d'idées. »

Utilisateur ambassadeur : « Ecrire nos propres écrans d'aide assurent que nous allons recevoir quelque chose que nous pouvons comprendre pour la première fois de notre vie ! »

7 Timeboxing et hiérarchisation des besoins

7.1 Timeboxing

Le concept de timeboxing (gestion d'une cartouche temporelle) est un aspect primordial des projets DSDM. Sans lui, les équipes de développement risqueraient de perdre de vue leurs objectifs et de ne plus contrôler l'évolution du projet.

Parmi les différentes définitions du terme « timeboxe », il en est une qui s'impose, c'est « le temps écoulé entre la date de début et la date de fin du projet ». La date de fin est fixe ; c'est la date de livraison du système (ou d'une partie de celui-ci). En jargon DSDM, on parle de **timeboxe globale**. DSDM a étendu le concept de timeboxe en créant des **timeboxes de bas niveau** pour chaque développement incrémental, fixant ainsi une série de dates de livraison (intermédiaire ou finale) pour le produit.

Le « timeboxing » consiste à définir la date limite à laquelle un objectif doit être atteint, plutôt que la date à laquelle une tâche doit être effectuée. La définition des timeboxes doit faire l'objet d'un consensus et les objectifs de ces dernières doivent être clairement définis conformément aux besoins de haut niveau établis dans le document Définition du domaine d'activité. L'un des aspects importants des timeboxes est que le contrôle qu'ils mettent en place n'est pas basé sur l'activité. L'objectif d'une timeboxe est de réaliser quelque chose. La manière dont ce quelque chose sera réalisé relève entièrement de la décision des personnes en charge du développement, dans la mesure où les normes et les procédures de contrôles du projet sont respectées.

Les timeboxes de bas niveau produisent des indices d'évaluation de l'avancement et de la qualité du projet visibles. Parmi les exemples de ce qu'une timeboxe peut inclure figurent une partie du Modèle fonctionnel, un module ou une version anticipée du système. Toutes les vérifications et tests nécessaires sont intégrés à la timeboxe (il serait en effet difficile de les faire tenir dans les timeboxes ultérieures). En d'autres termes, une timeboxe contient au moins un cycle complet des phases Modèle fonctionnel itératif ou Conception et Réalisation Itératives, pour au moins une partie du système global.

Les timeboxes de bas niveau sont le plus souvent d'une durée de deux à six semaines, l'idéal étant qu'elles soient aussi courtes que possible. Mais ces limites ne revêtent aucun caractère d'obligation. L'un des arguments majeurs en faveur des timeboxes courtes est le fait qu'elles facilitent l'évaluation de ce qui peut être accompli dans ce délai. Non seulement le timeboxing permet de contrôler le projet mais il facilite l'estimation des ressources nécessaires pour élaborer un système opérationnel.

C'est l'équipe qui a le pouvoir de décision quant au contenu des timeboxes. Il revient aux développeurs et aux utilisateurs de définir conjointement les objectifs de la timeboxe et aux développeurs d'évaluer le temps nécessaire pour les réaliser. A la date fixée, les utilisateurs doivent être en mesure d'approuver la livraison des produits couverts par la timeboxe. S'il apparaît que les dates limites ne pourront pas être respectées, il sera nécessaire de revoir les produits à la baisse de manière à pouvoir livrer dans les temps impartis. En d'autres termes, les besoins peuvent évoluer mais les délais jamais. Il appartient aux utilisateurs de décider si la satisfaction de tels ou tels besoins doit être intégrée à la timeboxe en cours ou reportée à une timeboxe ultérieure. Le seul paramètre immuable est la date limite de la timeboxe. Cette approche permet d'effectuer des livraisons de produits fréquentes et facilite le contrôle du projet mais elle alourdit quelque peu la charge administrative.

Toute timeboxe ultérieure intégrant des tâches dépendantes ne peut être démarrée qu'une fois la précédente menée à son terme. Si ce n'était le cas, la timeboxe précédente risquerait en effet de produire un résultat incomplet, ce qui pourrait influencer sur les objectifs et les activités de la timeboxe suivante.

La timeboxe (et DSDM en général) atteint son maximum d'efficacité lorsque l'équipe de développement s'appuie sur des outils permettant une livraison rapide des produits, qu'il

s'agisse de modèles, de documents ou de logiciels. L'équipe doit être capable d'élaborer et de faire évoluer les produits rapidement sans être ralentie par les technologies qu'elles utilisent.

Plusieurs exemples de contrôles basés sur les timeboxes sont fournis dans la brochure « DSDM : la méthode en pratique » (se reporter à la bibliographie).

7.2 Les règles MoSCoW (FISPE)

Dans la mesure où les timeboxes doivent demeurer fixes, il arrive que le nombre de fonctions livrées soit inférieur à celui initialement prévu. Il est toutefois indispensable que l'essentiel du travail soit réalisé et que seules les tâches les moins importantes soient laissées de côté. La méthode qui permet d'y parvenir est la priorisation des besoins.

Les Fonctions Indispensables, Souhaitables, Possibles, Éliminées (FISPE) : Les règles MoSCoW (FISPE) permettent d'établir une hiérarchie claire des besoins. Les « o » du terme MoSCoW en anglais n'ont d'autre objet que d'ajouter une pointe d'humour. Quant aux lettres M, S, C, elles ont la signification suivante :

- « **Must have** » (**Indispensable**) désigne les besoins fondamentaux du système. Si ces besoins ne sont pas satisfaits, le système sera inexploitable et donc inutile. Les « Must Have » définissent le **sous-ensemble minimum utilisable**. Un projet DSDM garantit la réalisation intégrale du sous-ensemble minimum utilisable.
- « **Should have** » (**Souhaitable**) désigne les besoins importants qui peuvent être contournés dans le court terme et qui seraient considérés comme obligatoires dans des conditions de délai moins contraignantes. Le système demeure cependant exploitable et utile même si ces besoins ne sont pas satisfaits.
- « **Could have** » (**Possible**) désigne les besoins dont la satisfaction peut être aisément omise durant le développement d'un incrément.
- « **Want to have but Won't have this time round** » (**souhaité mais non réalisable dans l'immédiat, donc Élimine**) désigne des besoins dont la satisfaction peut être remise à plus tard, lors de développements futurs.

« De temps en temps, il y avait un problème avec la technologie qui a fait tomber un besoin 'souhaitable' de la timeboxe, mais les utilisateurs étaient confortables avec la décision. Quoi, aucun problème de caractère contentieux ? Le seul problème c'est la précipitation des utilisateurs. »

Tout ceci constitue les besoins définis pour le système complet. La « liste des fonctions souhaitées » n'apparaît pas dans ce classement. Les règles MoSCoW (FISPE) servent de base pour toutes les décisions concernant ce qui sera réalisé par l'équipe de développement tout au long du projet et durant chaque timeboxe.

A mesure que de nouveaux besoins apparaissent ou que les besoins existants sont définis plus en détail, il importe d'évaluer au moyen des règles MoSCoW (FISPE) dans quelle mesure ceux-ci sont essentiels au succès du travail en cours. D'un bout à l'autre du projet, toutes les priorités doivent être ainsi passées en revue afin de déterminer si elles sont toujours valides.

La non obligation de réaliser l'intégralité d'un produit durant un projet ou une timeboxe est un principe essentiel. C'est l'existence de besoins de bas niveau qui permet aux équipes de livrer dans les délais impartis lorsque des problèmes surviennent, en abandonnant les développements dont le niveau de priorité est le plus faible.

« Nous avons eu moins que nous avons demandé, mais plus que nous avons anticipé, parce que nous avons reçu précisément ce dont nous avions besoin. »

8 Ateliers facilités

8.1 Définition

On appelle Atelier Facilité une technique de collecte des informations et de prise de décision en équipe dont l'objet est d'accélérer la planification et le développement de l'entreprise. C'est une méthode de communication interactive qui met en présence un personnel expérimenté et investi d'un pouvoir de décision durant une ou plusieurs sessions de travail dirigées par un « facilitateur » indépendant. L'atelier est un processus utile lorsqu'il est nécessaire de prendre des décisions, d'explorer des idées et d'échanger des connaissances en vue de résoudre un problème professionnel.

L'intérêt de la technique d'Atelier Facilité est qu'elle permet aux membres d'un groupe d'individus clés d'échanger des informations et de prendre des décisions mutuellement acceptables. L'atelier constitue un forum d'échange de points de vue et de prise de décisions consensuelles se déroulant dans un cadre structuré et traitant d'un domaine spécifique du business. Des résultats clairs sont obtenus durant ces ateliers, permettant à tous les participants de passer en revue les décisions prises par le groupe. Les ateliers facilités sont décrits plus en détail au chapitre 23.

8.2 Qu'est-ce que permet de réaliser un Atelier Facilité ?

Particulièrement utiles dans un contexte DSDM, les Ateliers Facilités offrent les avantages suivants :

- **Vitesse** - en réunissant dans un atelier les principales personnes concernées par le projet, il est possible d'arriver à un consensus sur la définition des besoins des utilisateurs beaucoup plus rapidement (par exemple, un ou deux jours au lieu de deux semaines ou deux mois) que ne le permettent les méthodes traditionnelles.
- **Appropriation** - les personnes concernées par le projet qui participent au processus de décision se sentent généralement plus concernées par les décisions prises et plus impliquées dans les tâches de développement ultérieures.
- **Productivité** – l'atelier instaure un climat d'émulation dans lequel les idées des uns engendrent les idées des autres et qui est propice à une meilleure compréhension du point de vue d'autrui. Les ateliers ont ainsi permis de clarifier bien des malentendus.
- **Consensus** - l'atelier offre aux participants l'occasion de discuter de divers sujets appropriés, tels que les principales questions et problèmes qui les préoccupent, avec l'idée d'arriver à un consensus sur toutes les décisions importantes.
- **Qualité des décisions** – les personnes concernées par le projet prenant une part active dans le processus de décision, le niveau de confiance dans les résultats d'un atelier et de compréhension de ces derniers est généralement élevé.
- **Perspective globale** – les participants à un atelier peuvent apprécier l'importance des domaines du business couverts par le projet mais qui ne les concernent pas directement.

8.3 Quand recourir aux Ateliers Facilités

Certaines étapes du cycle de vie DSDM sont plus particulièrement propices à l'utilisation des Ateliers Facilités. La première est l'Etude de Faisabilité, durant laquelle divers participants au projet déterminent dans quelle mesure l'approche DSDM est adaptée à un problème particulier. La seconde est la phase d'Etude du Business qui permet d'analyser le périmètre, l'approche et les différents rôles envisagés, et de produire un modèle de haut niveau du domaine de business concerné. Dans la mesure où une grande variété de personnes sont susceptibles d'être concernées par cette phase, le recours à un Atelier Facilité présente un avantage certain. De plus, durant l'Etude du Business, les ateliers constituent un outil efficace

pour concevoir les nouveaux processus qui seront supportés par le système. Mais là où les ateliers atteignent leur maximum d'efficacité, c'est dans la collecte et la hiérarchisation des besoins. Le processus de atelier peut aussi être utilisé durant la révision des prototypes, ainsi que pour la révision finale, une fois le système mis en œuvre. Mais l'équipe DSDM peut recourir à cette technique en bien d'autres occasions : par exemple, l'utilisateur ambassadeur peut réunir d'autres utilisateurs en atelier afin de déterminer quel type d'aide il sera nécessaire d'apporter aux utilisateurs pour le système en cours de développement.

« Suite aux ateliers facilités sur la définition du domaine industriel, nous avons compris que le périmètre était loin au delà de nos moyens. Un nouvel atelier permettait de prioriser en fonction de nos estimations pour respecter les contraintes. »

9 Prototypes

9.1 Pourquoi les prototypes sont-ils nécessaires ?

Tandis que les ateliers définissent les besoins et la stratégie de niveau global, les prototypes fournissent le mécanisme permettant aux utilisateurs de s'assurer de la validité du système de manière détaillée. La démonstration d'un prototype confère aux utilisateurs une meilleure compréhension des possibilités du nouveau système et améliore la qualité des informations que ceux-ci renvoient aux développeurs. Le processus de développement s'en trouve accéléré et la confiance de l'utilisateur dans le système développé accrue.

Toutefois, il est important de noter qu'un prototype n'est rien de plus qu'un prototype. Il n'a pas besoin d'être complet ou testé au regard des caractéristiques fonctionnelles ou non fonctionnelles requises. Cependant, les prototypes DSDM sont conçus dans une perspective **incrémentale**, c'est à dire qu'ils sont appelés à évoluer pour s'intégrer au système final. Pour plus de détails sur les catégories de contrôles et de prototypes, reportez-vous au chapitre 12.

9.2 Gestion de l'itération

Le succès des diverses activités de prototypage repose sur la bonne gestion des aspects suivants :

- timeboxing ;
- contrôle des modifications ;
- gestion de la configuration ;
- implication de l'utilisateur ;
- assurance qualité des produits ;
- éléments appris des itérations précédentes.

La stricte adhésion aux principes du timeboxing offre deux garanties. La première est que les cycles de prototypage ne perdent jamais de vue leur objectif (à savoir, la satisfaction des besoins prioritaires définis) et la seconde, que la qualité des produits est évaluée au niveau de chaque timeboxe.

Les procédures de contrôle des modifications doivent permettre d'introduire les changements de manière à la fois rapide et efficace. Les procédures s'appuyant sur des spécifications trop précises sont de nature à ralentir le développement.

Pour pouvoir faire marche arrière, c'est à dire revenir à un prototype antérieur, il convient de mettre en œuvre une gestion de la configuration efficace. Par exemple, les utilisateurs peuvent décider que l'interface précédente était plus satisfaisante que celle en cours ou que les fonctionnalités évoluent dans la mauvaise direction eu égard à leurs besoins immédiats. Dans ces conditions, il doit être possible de revenir en arrière sans heurt et de s'assurer que tous les développeurs savent de manière certaine quelle est la version ou variante en cours.

Bien que les étapes de prototypage soient irréalisables sans l'aide des utilisateurs, il convient de s'assurer que ces derniers n'entraînent pas le processus de développement dans une mauvaise direction. Il peut arriver qu'un groupe d'utilisateurs imprime au développement une direction qui ne tienne pas suffisamment compte des besoins du business étrangers à leurs propres préoccupations. C'est à l'encadrement de faire en sorte qu'un panel d'utilisateurs représentatifs soit associé aux activités de prototypage.

Autre point important concernant la participation des utilisateurs, les représentants des utilisateurs sélectionnés doivent être disponibles à tout moment.

Dans la mesure où la stabilité de l'équipe est assurée tout au long du développement, les membres de cette équipe pourront tirer des enseignements des activités de prototypage

précédentes et ainsi travailler de façon plus efficace tout en créant des produits de meilleure qualité.

9.3 *Tirer le meilleur profit des démonstrations de prototype*

Avant chaque session de démonstration, les démonstrateurs sont tenus de préparer leur audience. Les objectifs de la session mais aussi les limites de ce qui sera montré doivent être clairement définis. Ce n'est pas, comme on pourrait le penser, parce qu'un ou deux utilisateurs ont été impliqués dans le développement d'un prototype que tous les autres auront le même niveau de compréhension de ce qui se passe. Les utilisateurs ambassadeurs pourront, il est vrai, discuter avec leurs collègues mais ce serait une erreur de compter entièrement sur ce type d'échange tant que l'ensemble des utilisateurs n'aura pas vu une démonstration d'un logiciel en état de fonctionnement. En effet, il est souvent difficile d'expliquer une chose que l'on est pas en mesure de montrer - c'est l'une des raisons d'être de la méthode DSDM.

Durant la session, les discussions doivent être encouragées. Une démonstration trop rapide du prototype serait parfaitement inutile car les utilisateurs n'auraient alors pas le temps de voir ce que l'on attend d'eux. L'équipe de développement doit quitter la session rassurée, avec le sentiment qu'elle est sur la bonne voie, et non frustrée de n'avoir pu s'exprimer comme elle l'aurait souhaité.

10 Faire adopter DSDM dans une entreprise

10.1 *Changement de culture*

Une entreprise qui souhaite faire adopter l'approche RAD a déjà une culture et des pratiques de travail acceptées dont certaines peuvent aller à l'encontre de l'introduction efficace de DSDM, et ce même avant que la question se pose de savoir si un projet particulier est approprié à DSDM ou si le personnel a les compétences nécessaires et les outils adéquats pour le mener à bien.

Les questions à prendre en considération sont les suivantes :

- Comment, à l'heure actuelle, le personnel est-il affecté aux projets ? Les tâches ne sont-elles dévolues aux développeurs qu'en fonction de leur position/ rang/ grade dans l'entreprise ? (Les jeunes membres du personnel n'ont pas accès aux activités d'analyse et les personnels de haut niveau ont depuis longtemps oubliés par manque de pratique leurs déjà piètres compétences techniques.)
- Les chefs de projet de l'entreprise ont-ils réellement du «mordant» ? Ou bien se sentent-ils toujours handicapés par des règlements incontournables ?
- L'environnement de travail actuel est-il contrôlé par consensus ou par des règlements ?
- Dans quelle mesure les développeurs sont-ils favorables au changement des méthodes de travail ?
- Le personnel peut-il être redéployé librement dans une autre partie de l'entreprise sur la base de projets individuels ou bien l'environnement technique et d'entreprise requiert-il que chaque employé soit affecté de façon permanente à une fonction ou un bureau ?
- Est-il possible de créer un espace pour les « ateliers facilités », par exemple, des salles adéquates peuvent-elles être réquisitionnées par le projet pour plusieurs jours de travaux ininterrompus ?
- Le personnel d'exploitation pourra-t-il répondre rapidement à des demandes de transfert ou de support système lorsque nécessaire. ?
- L'environnement de développement convient-il au prototypage avec les utilisateurs ?

Si la réponse à l'une de ces questions n'est pas satisfaisante, la direction informatique et le comité de direction devraient étudier comment infléchir ou contourner cette situation. Par-dessus tout, ils doivent être convaincus qu'aucuns des principes DSDM ne constitue un obstacle insurmontable. Si l'entreprise ne s'est jamais auparavant engagée dans un projet RAD, elle devrait s'assurer des services d'une personne ayant cette expérience qui puisse l'assister dans le transfert vers DSDM.

10.2 *Plan d'action*

Afin d'être couronnée de succès, la mise en œuvre de DSDM dans une entreprise doit être planifiée et gérée avec beaucoup de soin.

Tout d'abord, les raisons de la mise en œuvre de DSDM doivent être comprises. Des pressions extérieures à l'entreprise telles que celles du marché ou de changements législatifs peuvent la rendre indispensable. En ce cas l'entreprise souhaitera vivement appliquer DSDM pour satisfaire à ses besoins et fera pression sur son département informatique. Dans d'autres situations, ce sont les services informatiques qui prendront l'initiative de passer au DSDM, souvent parce qu'il faut que l'entreprise s'engage davantage afin de fournir les systèmes adéquats.

Une fois la raison de l'introduction de DSDM définie, il faut établir un plan pour sa mise en œuvre. Le plan typique est le suivant :

1. Identifier un champion pour DSDM

Le succès de la mise en œuvre de DSDM dépend souvent de la motivation d'une personne qui croit fermement en DSDM et est déterminée à remettre en question et à changer la culture existante de l'organisation et les pratiques en vigueur là où c'est pertinent. Ce champion sera extrêmement utile pour convaincre les sceptiques, surtout lorsqu'ils occupent des postes élevés dans l'entreprise.

2. Communiquer la philosophie et les concepts de DSDM à tous ceux qui sont concernés

Normalement, cet objectif est atteint par la participation à une série de stages de formation DSDM « sensibilisation » ciblant, le plus souvent, le niveau des cadres supérieurs et le conseil d'administration (tant du business que de la L'informatique), toutes les catégories affectées d'utilisateurs du business et le personnel affecté au développement et au soutien en L'informatique. Dans le cadre d'organisations plus importantes, ces stages sont conçus pour le groupe identifié comme participant au premier projet DSDM et à sa mise en œuvre planifiée.

3. Examiner les pratiques et les procédures de développement en vigueur et les comparer à l'approche DSDM

De nombreuses entreprises constatent qu'elles utilisent déjà des éléments de DSDM tels que la participation de l'utilisateur, la délégation du pouvoir, le prototypage, les «Ateliers Facilités» et les «timeboxes». En recourant à des exemples de pratiques couronnées de succès déjà mises en œuvre, DSDM peut être considéré comme s'établissant sur la base des meilleures pratiques en vigueur.

4. Identifier les domaines qui devront changer en recourant aux même comparaisons.

Dans les business où il existe un écart important entre la culture présente et celle requise par DSDM, un programme complet de changement de gestion devrait être défini. Ce qui peut exiger le recours à des consultants extérieurs spécialisés. Même si l'écart est relativement faible, il demeure nécessaire de définir un plan. Au nombre des questions qu'il faut aborder figurent la délégation de pouvoir, la polyvalence des membres d'une équipe et les compétences techniques. L'utilisation des règles MoSCoW (FISPE) peut contribuer à établir des priorités pour le plan.

Là où existent des écarts entre la pratique existante et les procédures, la méthode pour les supprimer doit être définie soit pour une action immédiate, soit pour un projet pilote qui puisse informer en retour et agir sur les procédures de l'entreprise. Cette dernière approche ne doit être utilisée que dans le cas de changements ou de vides relativement mineurs des méthodes de travail existantes.

5. Obtenir le soutien et la participation à toutes les activités du plan

Une fois le soutien au plan obtenu, il devient possible de réaliser l'ensemble du travail préparatoire nécessaire pour lancer le premier projet sur une base saine.

« Alors, il n'y a rien de nouveau dans DSDM. En effet, nous le faisons depuis longtemps sans le connaître, mais désormais nous disposons d'un ensemble cohérent et nous sommes plus crédibles devant nos clients. Oui, nous avons été obligés de reprendre nos standards, mais tant mieux. »

6. Identifier le premier projet

Le filtre d'adaptation et les critères essentiels de succès de DSDM servent à identifier un projet adéquat. Certains business choisiront des projets destinés à satisfaire des besoins urgents pour lesquels il n'y a aucune autre alternative que DSDM si l'entreprise a des délais très courts à respecter. D'autres business choisiront un projet pour un secteur d'activité où elles ont déjà établi des relations de travail étroites et fructueuses avec les cadres supérieurs et les principaux utilisateurs. Quelle que soit la situation, il faut évaluer les risques et convenir de leur gestion (voir Chapitre 18).

Il est particulièrement important de choisir, en se référant aux définitions de rôles pour DSDM, des équipiers qui conviennent aux activités projetées. (Voir Chapitre 15).

7. Formation de l'équipe chargée du projet

Tous les membres de l'équipe, à plein temps ou à temps partiel, doivent être formés par les stages de formation appropriés et accrédités pour DSDM. Il peut également s'avérer nécessaire de fournir une formation technique en ce qui concerne les outils, la configuration de gestion, les tests, etc.

8. Mise sur pied de l'environnement de développement

Il est préférable de consacrer une salle à l'équipe DSDM dans le service approprié du business. Tout l'équipement technique nécessaire au travail de l'équipe devrait être installé dans cette salle.

9. Lancement du projet

Lorsque tout est prêt, il est de coutume d'organiser sous une forme ou une autre une réunion spéciale pour lancer le projet. Cette réunion peut prendre la forme d'une présentation de la vision du business par le «visionnaire», d'activités renforçant la cohésion de l'équipe, une réception, une journée portes ouvertes ou l'association de plusieurs de ces éléments.

10. Gestion du premier projet.

Il est fréquent qu'une organisation fasse appel à un Mentor DSDM pour prêter assistance lors du premier projet, afin de s'assurer que l'approche DSDM est suivie correctement et que l'équipe garde le cap sur les bénéfices devant résulter du projet. En fait, la présence d'un Mentor DSDM peut s'avérer très utile avant le début d'un projet pour aider à identifier les changements dans les méthodes de travail qui doivent être pris en compte.

11. Révision du projet

Après le succès de la mise en œuvre du premier projet, il est recommandé de prendre le temps d'analyser le projet, afin d'en tirer les enseignements. Que faudra-t-il faire différemment la prochaine fois ? Qu'est-ce qu'il faudra garder ? Les documents, les procédures et tout conflit avec les méthodes de travail existantes devront faire l'objet d'une attention particulière. Ces résultats devront être enregistrés et transmis au projet suivant.

12. Faites connaître votre succès

Les articles de présentation, les vidéos, les bulletins d'information, les articles de presse, etc., contribuent à faire connaître la réussite du projet à tous ceux qui travaillent dans l'organisation. Ceci conduit souvent d'autres services à demander des projets DSDM

13. Se présenter à l'examen de praticien DSDM.

Les membres de l'équipe peuvent, s'ils le souhaitent, se présenter à l'examen de praticien DSDM acquérant ainsi une qualification professionnelle en DSDM. On peut se procurer le programme de l'examen auprès du Consortium DSDM.

14. Se préparer pour les prochains projets DSDM.

Lors des premières étapes de la propagation de DSDM dans une organisation, de nombreuses compagnies séparent les membres de l'équipe DSDM pour former de nouvelles équipes avec des équipiers qui eux n'ont pas encore une expérience en DSDM. Il s'agit de former davantage d'équipes tout en incluant des développeurs DSDM confirmés dans chaque équipe. D'autres organisations considèrent que la division d'une bonne équipe crée trop de perturbations et la garde telle qu'elle est constituée aussi longtemps que possible.

15. Dernier point : quantifier les bénéfices pour le business.

Les bénéfices anticipés ont-ils été réalisés ? Est-ce que le bien fondé de cette approche pour le cas soumis par le business a été justifiée ? Il est sage de considérer à la fois les bénéfices matériels et ceux qui sont moins apparents. L'un des bénéfices typiques découlant du premier projet DSDM est une meilleure compréhension mutuelle entre le service du business et celui de la l'informatique et de leurs points de vue différents. Dans certaines compagnies, la relation entre les services business et l'informatique en a été considérablement améliorée.

Deuxième partie – Caractéristiques

11 Le squelette de développement du processus DSDM

11.1 Introduction

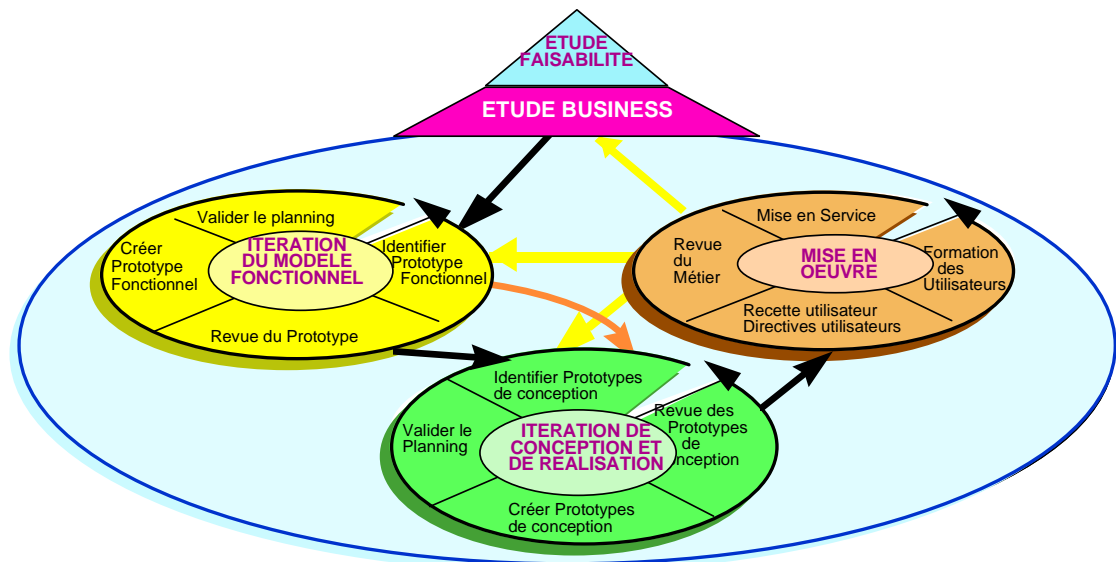


Schéma 0.1. Le squelette du processus de développement de DSDM

Ce chapitre définit le cycle de vie par défaut à adopter pendant un projet DSDM. Il n'est pas obligatoire et devrait être adapté pour satisfaire aux exigences d'un projet spécifique, c'est pourquoi il est appelé «cadre du processus de développement». Ce chapitre en donne la définition formelle et le chapitre 3 contient une description du processus.

Le cadre du processus de développement est basé essentiellement sur les Principes I, III, V et VIII. Dans DSDM, il y a cinq phases de développement :

- 0 Etude de Faisabilité ;
- 1 Etude du Business ;
- 2 Modèle Fonctionnel Itératif ;
- 3 Conception et Réalisation Itératives
- 4 Mise en Œuvre

Le schéma 11.1 donne une idée du squelette du processus de développement du DSDM. Le diagramme montre une approche de prototypage incrémentale allant dans le sens contraire des aiguilles d'une montre à partir du haut. Les flèches foncées indiquent les points de transfert d'une phase du cycle de vie à la suivante. Les flèches claires indiquent les points où le développement peut facilement revenir à une phase précédente. Chaque phase est définie en termes de :

- 0 d'objectifs de la phase ;
- 1 des rôles du business et de la l'informatique impliqués ;
- 2 des conditions préalables à l'entrée dans une phase ;
- 3 des produits livrés par la phase.

Veuillez noter que les noms de rôles utilisés dans la section 11.2 ne sont pas les rôles DSDM décrits dans le chapitre 15. Il s'agit de noms de rôles utilisés en développement de l'informatique pour indiquer les catégories de compétences requises pour chaque phase. Le rôle du chef de projet n'est pas indiqué ici dans la mesure où il est présent dans l'ensemble du processus. Le chapitre 15 contient une matrice dressant la liste des rôles en informatique cités ci-dessous par rapport aux rôles DSDM.

Tous les produits ne sont pas livrés à la fin d'une phase. Certains sont bien évidemment des produits intérimaires qui permettent à un projet donné de progresser. Les points auxquels ils sont fournis sont décidés de projet à projet.

11.2 Définition du cadre du processus de développement

11.2.1 Etude de Faisabilité

Objectifs

Etablir si une proposition de développement peut satisfaire aux exigences du business de l'organisation

Evaluer si l'application est adaptée au développement DSDM

Esquisser des solutions techniques possibles pour le problème du business

Parvenir à une première évaluation de la durée et des coûts

Rôles

Analystes du marché, utilisateurs concernés par le système informatique, spécialistes techniques, encadrement supérieur utilisateur.

Conditions préalables

Accord sur l'étendue de l'enquête

Accord initial sur la définition du problème de business à traiter

Produits

Rapport de Faisabilité

Faisabilité du Prototype (optionnel)

Plan Global

11.2.2 Etude du Business

Objectifs

Etudier le périmètre des processus de l'organisation qui doivent être pris en charge.

Définir les grandes lignes du développement futur en termes de produits livrables par prototypage (définir ceux qui sont susceptibles d'un incrément et ceux, s'il y en a, qui sont jetables) et les contrôles du prototypage

Identifier les représentants des catégories d'utilisateurs pour les activités de prototypage

Etablir les priorités des besoins du système envisagé

Réévaluer que l'application convient au développement DSDM

Fournir une base solide au déroulement du développement technique

Evaluer la portée des besoins non fonctionnels

Rôles

Analystes, utilisateurs concernés du système informatique, encadrement supérieur utilisateur, architecte de système

Conditions préalables

Accord du Rapport de Faisabilité, y compris l'accord sur la faisabilité à la fois du développement et de l'applicabilité de l'approche DSDM

Produits

Définition du Domaine d'Activité Industrielle

Liste par ordre de priorité des besoins

Plan Global de Prototypage

Définition de l'Architecture de Système

11.2.3 Modèle Fonctionnel Itératif**Objectifs**

Démontrer les fonctionnalités requises grâce à un modèle fonctionnel composé de modèles prototypes fonctionnels et statiques

Enregistrer les besoins non fonctionnels que le prototype fonctionnel pourrait ne pas démontrer, par exemple, maintenabilité, performance, capacité, capacité de traitement, disponibilité, récupération en cas d'échec, rattrapage, sécurité, traces de contrôle, accès de multiples utilisateurs, gestion de réseau, et opération sur une plate-forme différente.

Rôles

Des analystes du marché pour vérifier que les prototypes supportent les processus requis par le business, des concepteurs/ programmeurs ayant une expérience technique de l'environnement de prototypage, utilisateurs de système informatique concernés, cadres supérieurs utilisateurs.

Conditions préalables

Accord de la Définition du Domaine d'Activité Industrielle et le Plan Global de Prototypage

Présence de l'environnement de prototypage

Engagement des cadres supérieurs utilisateurs à laisser le temps aux usagers destinataires de développer les prototypes

Produits

Modèle Fonctionnel

Prototypes Fonctionnels

Listes des Besoins non Fonctionnels

Enregistrement de révision du Modèle Fonctionnel

Stratégie de Mise en Œuvre

Rapport d'Analyse du Risque de Développement

11.2.4 Conception et Réalisation Itératives

Objectifs

Raffiner les Prototypes Fonctionnels pour satisfaire aux besoins non fonctionnels

Concevoir l'application pour qu'elle réponde de façon démontrable aux besoins de l'utilisateur

Rôles

Des analystes du business pour s'assurer que la conception du système supporte les processus requis du business, des concepteurs/ programmeurs ayant une expertise technique de l'environnement de conception, des utilisateurs concernés du système informatique, des cadres supérieurs utilisateurs.

Conditions préalables

Accord sur un Prototype Fonctionnel et les besoins non fonctionnels qui lui sont associés

Accord sur tous les résultats et les changements de portée figurant dans l'Enregistrement de la révision du prototype fonctionnel

L'environnement de Conception et de Réalisation est en place

Produits

Conception de Prototypes

Enregistrements des modifications aux Prototypes

Systèmes Testés

Enregistrement des Ttests

11.2.5 Mise en œuvre**Objectifs**

Installer le Système Testé dans l'environnement de travail de l'utilisateur

Former les utilisateurs du nouveau système

Déterminer les besoins futurs de développement

Former les opérateurs et le personnel de soutien

Rôles

Analystes du business pour s'assurer que le système peut supporter les processus requis par le business, concepteurs /programmeurs ayant une expertise technique de l'environnement du système, formateurs, auteurs techniques, utilisateurs, cadres supérieurs utilisateurs, personnel d'opérations et de support

Conditions préalables

Accord de toutes les parties concernées sur le Système Testé, par exemple, les cadres supérieurs utilisateurs et le support technique

Période de formation offerte aux utilisateurs

L'environnement cible est en place

Produits

Documentation pour les utilisateurs

Population d'utilisateurs formés

Fourniture du système

Document de Capitalisation du Projet

11.3 Maintenance

La nature itérative et incrémentale de DSDM fait que la maintenance peut être considérée comme la poursuite du développement. La maintenance est un aspect normal du cycle de vie d'un système informatique qui peut être effectuée selon une approche similaire au développement initial. Les réparations non urgentes et les améliorations peuvent être groupées et effectuées en se servant de techniques DSDM. Ce travail devrait suivre les principes DSDM avec une implication importante de l'utilisateur et devrait être traité comme une autre itération du système, reprenant de nouveau la méthode DSDM en commençant par L'étude du business.

L'une des approches possible des changements apportés au système est de réécrire plutôt que d'amender ce qui doit être changé. Produire un système maintenable n'est pas incompatible avec cette approche. Chaque changement du système devra être évalué en fonction de ses mérites en ce qui concerne le coût et l'échelle des temps de l'amendement et de la réécriture. Il est essentiel d'être pragmatique à ce propos quand il s'agit de décider de la meilleure approche pour les changements

Pendant la maintenance, la présence d'un «sponsor exécutif» est importante : il devrait y avoir un «sponsor exécutif» (parrain) à la fois pour le développement et pour la maintenance. Il devrait également y avoir un sponsor en L'informatique pour la maintenabilité.

Toutes les tâches de maintenances doivent être classées par ordre de priorité avec les utilisateurs pour que seuls les besoins de haute priorité et d'un bon rapport qualité/prix du business soient satisfaits. Les économies ainsi réalisées sur le budget de maintenance et en capacité de ressources peuvent être alors consacrées aux exigences réelles de remplir les objectifs du business.

Les questions de maintenabilité doivent être étudiées très tôt

La maintenance est incontournable dans la mesure où les besoins du business changent, de sorte que même si la maintenance intervient à la fin de la définition du Cadre du processus de développement, elle doit être prise en compte dès le début du projet. Les systèmes informatiques qui ont un faible degré de maintenabilité :

- 0 absorbent davantage de ressources en maintenance ;
- 1 prennent plus longtemps à changer ;
- 2 risquent davantage d'introduire des erreurs supplémentaires lors de changements et d'être peu fiables ;
- 3 et leur maintenance sera plus onéreuse.

Les systèmes informatiques qui ont un faible degré de maintenabilité constituent un risque réel pour le business. Au pire, un nouveau système peut se révéler rapidement problématique, un héritage dont la maintenance est impossible, et dont les utilisateurs vont demander le remplacement. Les systèmes futurs développés en utilisant DSDM ne doivent pas ajouter au fardeau déjà existant et considérable qui pèse sur toute l'industrie.

L'un des principes de DSDM est particulièrement pertinent en ce qui concerne la maintenance et la maintenabilité : à savoir que le critère essentiel de l'acceptation de produits livrables est l'adaptation aux besoins du business. En d'autres termes, il faut servir les objectifs du business. En conséquence, si l'un des objectifs du business est de disposer d'un système informatique maintenable fournissant un bon rapport coûts/ bénéfices alors il n'y a rien à perdre en y intégrant la maintenabilité.

Les composants ayant une faible maintenabilité peuvent ralentir le développement d'incrément futurs. La maintenabilité et la «RADabilité» sont donc indissociables.

La maintenabilité dans DSDM

DSDM, en soi, n'assure pas la maintenabilité. Celle-ci est rendue possible par l'association de quatre facteurs : les outils, les personnels, la documentation et les lignes directrices de bon usage.

Le manuel de DSDM fournit, dans d'autres chapitres une assistance en ce qui concerne les trois premiers points. Les grandes orientations de bon usage couvrent des thèmes tels que les normes, les guides de style etc. : en fait, tout ce qui serait vraisemblablement fait automatiquement pour un projet «en cascade» et qui ne devrait pas être oublié pour les projets DSDM. Si tous ces aspects sont pris en compte au début du cycle de vie du projet, la maintenabilité devient un attribut naturel de tous les systèmes informatiques fournis et ne constitue pas une charge supplémentaire. En fait, ils devraient être pris en compte pour l'ensemble de l'installation avant que le projet soit lancé afin que le coût et le temps de mise en place des grandes orientations ne pèsent pas sur les projets.

Objectifs de Maintenabilité

Un projet DSDM offre trois possibilités de choix, qui sont indiquées ci-dessous, en ce qui concerne les objectifs (exigences) de maintenabilité. Les décisions quant à ceux qui sont pertinents pour un projet donné figurent obligatoirement dans L'étude du business. Ces trois niveaux sont les suivants :

- **La maintenabilité est un attribut requis du système informatique livré initial.** Ici, le critère de déploiement n'est pas simplement de fournir la fonctionnalité requise de façon robuste, mais que la conception et le code satisfassent au standard de maintenabilité avant que le système soit accepté et utilisé dans le business.
- 0 **Livrer d'abord, revoir la technique de système plus tard.** La priorité du business est d'obtenir et de mettre en œuvre rapidement la fonctionnalité nécessaire. Le système doit avoir une longue durée de vie et être maintenable, mais, dans ce cas, le business est prêt à investir dans la technique système après la mise en œuvre

Ceci implique un coût de développement plus élevé que de concevoir la technique système en vue de la maintenabilité dès le début, néanmoins cela permet livraison initiale plus rapide et peut permettre des coûts réduits de durée de vie pour le détenteur, plutôt que d'être confronté pendant des années à des problèmes de maintenance. (Ce qui est souvent le cas quand le temps d'accès au marché est critique – qu'il s'agisse de vendre des logiciels sur un marché en évolution rapide ou de logiciels destinés à satisfaire un business qui se développe rapidement.)

- **Solution tactique à court terme.** L'objectif est la livraison dans les plus brefs délais. L'acceptation ne tiendra pas compte de la maintenabilité. Il est convenu que le système informatique sera remplacé ou réécrit avant que les coûts de maintenance ne deviennent un problème.

Le système ou le composant du système sera développé avec une durée de vie limitée en service. Sa succession sera planifiée et le système qui remplacera la solution temporaire sera bien conçu et maintenable. Ceci ne peut être considéré que comme un «pis-aller» qui jouera également le rôle de spécification exécutable de fonctionnalité pour le remplacement maintenable.

Il est particulièrement important que toute décision relative à la construction d'une solution technique soit documentée, dans la mesure où ces systèmes ont tendance à devenir, à long terme, les systèmes d'entreprise.

Comme pour ce qui concerne toutes les exigences non fonctionnelles clefs, à la suite de la décision sur la maintenabilité dans le cadre des objectifs du business, les risques de l'approche choisie doivent être définis et une stratégie de gestion du risque doit être adoptée.

La décision prise au début du projet doit être réaffirmée à toutes les étapes marquantes du développement. L'objectif de maintenabilité peut être changé si, par exemple, l'échelle des temps

devient plus critique. Cependant, aucun changement ne doit être pris à la légère et la stratégie de risque doit être revue.

12 Le contrôle du prototypage

12.1 Catégories de prototypes

DSDM recommande quatre catégories de prototypes :

- **Business (Métier)**
démontrant l'automatisation des processus du business ;
- **Utilisation (Convivialité)**
étudiant les aspects de l'interface utilisateur qui n'affectent pas la fonctionnalité ;
- **Performance et Capacité**
assurer que le système pourra traiter avec succès l'ensemble des charges de travail ;
- **Technique**
essai d'une approche spécifique de conception ou prouver un concept

En pratique, la plupart des prototypes reposent sur un certain degré d'association des catégories mentionnées, cependant, pour plus de clarté ils sont ici traités séparément.

12.1.1 Prototypes Business

Objet

Un Prototype Business illustre la compréhension qu'a le développeur des exigences fonctionnelles. Les développeurs peuvent se servir de ce prototype pour montrer aux utilisateurs comment le système final pourrait fonctionner. Ceci permet aux utilisateurs de mieux formuler les besoins réels de leur business.

Description

Un Prototype Business est conçu uniquement pour montrer comment le système informatique supporte les processus du business. Il n'est pas conçu pour être attrayant ou particulièrement facile à utiliser ; les fonctionnalités secondaires telles que la vérification des erreurs ne sera pas non plus nécessairement applicable. Il est très important que les utilisateurs comprennent le but de la démonstration d'un prototype business.

Utilisation

Un Prototype Business permet aux développeurs de démontrer aux utilisateurs comment ils comprennent les principaux besoins du système au début du projet. Ce premier prototype peut ensuite évoluer pour couvrir un domaine plus étendu de fonctionnalité et incorporer des aspects non fonctionnels.

Pour que la fonctionnalité soit démontrée le mieux possible le développeur s'appuiera également sur un document de démonstration écrit.

Tout écart entre la compréhension du développeur et les besoins du business sont notés.

Position dans le cycle de vie du projet

Le premier Prototype Business sera en démonstration le plus tôt possible dans le cycle de vie du projet (éventuellement dès L'étude de faisabilité, mais pas plus tard que le Modèle fonctionnel itératif). Ceci permettra de confirmer que le système en construction est le bon. Dans un premier temps, la fonctionnalité fondamentale du système sera démontrée, puis, plus tard, il sera possible de faire une démonstration de besoins fonctionnels plus détaillés et/ou d'autres domaines pourront faire l'objet d'un prototype business.

Le Prototype Business final montrera clairement aux utilisateurs comment le système va fonctionner. Il est possible qu'il ne soit pas esthétiquement satisfaisant et qu'il lui manque beaucoup de fonctionnalité, mais il permettra d'être certain que le système en construction est le bon.

12.1.2 Prototypes d'Utilisation

Objet

Un Prototype d'Utilisation sert à garantir une utilisation aussi simple et intuitive que possible du système informatique. Les utilisateurs devraient pouvoir trouver du plaisir à se servir du système et son utilisation devrait être évidente. Si tel est le cas, la formation des utilisateurs à une utilisation efficace du système sera plus rapide et ce tout assurant toujours une bonne compréhension des possibilités complètes du système.**

Description

Les systèmes informatiques bien conçus sont simples et directs, l'apprentissage de leur utilisation et leur compréhension est facile et plutôt distrayante. Il est impossible de parvenir à la convivialité sans tester le système sur les utilisateurs. Un prototype d'utilisation montre l'interaction de l'utilisateur avec le système. Il peut ne pas réellement automatiser la tâche, par exemple, un formulaire apparaît, l'utilisateur y ajoute des données mais rien n'est enregistré sur le disque. L'utilisateur peut comprendre comment se déplacer dans le système et comment l'interface fonctionne.

Utilisation

Il est demandé à un utilisateur de se servir du prototype pour réaliser un certain nombre de tâches. Toutes les difficultés rencontrées par l'utilisateur dans cet exercice sont notées afin d'améliorer la convivialité du système.

Ceci contraste avec le prototype business par lequel le développeur fait la démonstration du système pour n'en montrer que la fonctionnalité. Un prototype business peut être plutôt rébarbatif pour l'utilisateur, c'est pourquoi il est préférable que le développeur élimine pour l'utilisateur la nécessité d'apprendre l'interface.

Le danger existe de construire un prototype convivial qui ne puisse pas être développé dans le système final. Les développeurs doivent faire attention à ne pas susciter des attentes irréalistes.

Position dans le cycle de vie du projet

Un Prototype d'Utilisation peut être développé soit pendant le «Modèle Fonctionnel Itératif» soit pendant la «Conception et la Réalisation Itératives», cependant le développer le plus tôt possible présente de nombreux avantages. Pour qu'un système informatique soit simple à utiliser, il doit avoir un modèle conceptuel simple que les utilisateurs puissent comprendre aisément et qui les aide à se déplacer dans le système. Un prototype de convivialité confirme qu'un bon modèle conceptuel a été choisi. Si ceci n'est pas réalisé très tôt dans le projet et que le modèle ne soit pas le bon, la conception du système pourrait en souffrir. Afin de réduire ce risque, un prototype de convivialité de haut niveau devrait être développé au stade du «Modèle Fonctionnel Itératif» ou très tôt dans la «Conception et Réalisation Itératives». Des fonctions utilisables tels que les boutons dans une fenêtre peuvent être conçus plus tard. Si les normes de l'interface utilisateur sont choisies tôt dans le projet, alors tous les programmes peuvent être développés selon cette norme convenue. Ce qui réduit la nécessité de retravailler ce point. Le fait de disposer d'un guide de style d'installation stable avant de commencer le prototypage serait encore plus utile.

L'association des Prototypes Business et d'Utilisation présente dans la pratique de nombreux avantages.

12.1.3 Prototypes de Performance et de Capacité

Objet

Un Prototype de Performance et/ou de capacité permet de s'assurer que le système informatique final pourra assumer l'intégralité de la charge de travail aux heures de pointe. Bien que les utilisateurs puissent être impliqués, cette catégorie de prototype est normalement destinée aux développeurs.

Description

Ce prototype traite des aspects non fonctionnels du système tels que le chargement de données, les états séquentiels nocturnes et le traitement par lot en fin de mois ainsi que la performance en ligne sur écran. Des vérifications devront être faites pour savoir si le système cible a assez de ressources (y compris la puissance de l'unité centrale (CPU), le débit des communications, etc.) pour fonctionner de façon adéquate dans l'environnement réel même lorsque d'autres systèmes sont en concurrence pour les ressources machine.

Utilisation

Les Prototypes de Performance et de capacité sont, en général, développés pour servir aux développeurs qui vérifient que le système informatique peut satisfaire aux exigences de performances requises. Un scénario test est monté et répété alors que différents aspects du système sont changés pour voir comment celui-ci se comporte. Il vaut mieux automatiser ce processus afin de le répéter facilement, mais il peut être effectué manuellement lorsqu'un groupe d'utilisateur suit un document écrit. Le système est surveillé pour repérer où se produisent les «encombrements».

Position dans le cycle de vie du projet

Cette catégorie de prototype est normalement utilisée plus tard dans le cycle de vie du projet dans la «Conception et Réalisation Itératives», après que la fonctionnalité requise ait été déterminée. Les Prototypes Business existants pourront souvent être utilisés pour tester la performance. Il peut arriver qu'au début de la conception du projet, les développeurs soient préoccupés de savoir si un certain élément de fonctionnalité peut être fourni dans les limites de l'environnement de la machine/ du réseau. En ce cas, un prototype de performance et de capacité peut être conçu spécialement pour en vérifier les limites. Un tel prototype peut avoir une apparence très différente du système final.

12.1.4 Prototypes Techniques

Objet

Les développeurs disposent souvent de diverses options de conception et parfois d'un choix d'outils. Un Prototype Technique fait l'essai d'une approche de conception spécifique ou d'un outil permettant de choisir entre ces options.

Description

Cette catégorie de prototype a normalement une fonctionnalité limitée et est destinée aux seuls développeurs. Le prototype fait la démonstration aux développeurs des possibilités et des limites d'une approche, d'une technique ou d'un outil.

Utilisation

Le développeur conçoit plusieurs prototypes et évalue les avantages de chaque approche technique.

Position dans le cycle de vie du projet

Bien que la sélection d'un outil se fasse habituellement sans référence à un projet particulier, un prototype de capacité peut être développé pendant L'Etude de Faisabilité pour s'assurer que l'approche technique potentielle aura des bases saines. Les prototypes de conception technique qui suivent sont créés pendant la phase de Conception et Réalisation d'un projet.

Des conceptions ou outils différents servent à développer les prototypes qui représentent les options ouvertes au développeur. Chaque prototype peut alors être évalué et la meilleure approche, la meilleure technique ou le meilleur outil peut être sélectionné.

12.2 Cycles de prototypage

Chaque phase du développement (le Modèle Fonctionnel, la Conception et la Réalisation Itératives) contient l'itération par le prototypage. Il s'agit de contrôles de base qui doivent être mis en œuvre afin d'assurer le succès de ces activités. Ces contrôles font partie intégrante d'un cycle de prototypage.

Un cycle de prototypage passe par quatre étapes :

- identifier le prototype ;
- convenir d'un plan ;
- créer le prototype ;
- réviser le prototype.

Identifier le prototype

Avant de se lancer dans le développement d'un prototype, ce qui doit en faire l'objet doit être clairement identifié. Cette décision sera fondée sur les priorités relatives des besoins fonctionnels et non fonctionnels. Après avoir segmenté l'application en prototypes possibles, il faut, en appliquant les règles MoSCoW (*FISPE*), définir clairement quelles sont les parties essentielles de chaque prototype et quelles sont celles qui sont des «extra». Ceci va limiter l'étendue d'activité de chaque cycle de prototypage et déterminer les priorités des développeurs et des usagers qui conçoivent le prototype.

Divers critères peuvent permettre de sélectionner les prototypes : domaine de l'activité, traitement de base requis, les groupes utilisateurs, l'information obtenue et l'importance critique du traitement pour le système final.

Les résultats des examens de cycles de prototypages précédents fournissent des informations précieuses pour l'identification des prototypes à développer.

Les critères d'acceptation du prototype devraient être définis dans leurs grandes lignes avant tout développement afin d'orienter l'activité de prototypage dans la direction la plus utile.

Convenir d'un plan

Le Plan Global de Prototypage fixe une limite au temps à consacrer à chaque prototype. L'équipe doit convenir du plan détaillé du cycle de prototypage en cours. Ceci comprend le prototypage prioritaire des «must have» (indispensables). Les parties moins importantes seront traitées en fonction du temps disponible.

Il ne faut pas laisser le plan dériver à moins que surgissent des problèmes significatifs, par exemple, un changement imprévu et dramatique du terrain d'application. Toutefois, ce type de problème entraîne généralement l'arrêt temporaire de toute activité pendant que la direction du projet est repensée.

Il est important que les utilisateurs comprennent clairement les raisons des limites de temps. Ce sont leurs priorités qui vont déterminer les composants essentiels d'un prototype. Ils doivent être pleinement conscients du fait que s'ils demandent une enquête approfondie pour un secteur donné, ils pourraient être amenés à décider de quels autres secteurs ils peuvent se passer.

Créer un prototype

Les prototypes sont, en général, développés en collaboration avec les utilisateurs. Cependant, lorsque des questions telles que la performance seront traitées, plus tard dans le développement, les utilisateurs se placeront en retrait.

Les prototypes ne sont pas nécessairement automatisés. Au début du développement, par exemple, un plan sur papier du type «story board » peut s'avérer plus économique et plus flexible qu'une interface utilisateurs entièrement automatisée pour explorer la fonctionnalité. Ce sont les impératifs du business et techniques qui motiveront le choix du médium de prototypage.

Révision du prototype

Chaque prototype doit être revu et examiné par l'équipe de prototypage (développeurs et utilisateurs) ainsi que par les autres parties concernées, lorsque approprié (ex : analystes du business et l'encadrement utilisateur) pour vérifier :

- quels objectifs il satisfait ;
- quels domaines devront être inclus dans le développement subséquent ;
- quels domaines omis peuvent être reportés en toute sécurité (ou éventuellement incorporés à un autre projet) afin de respecter l'échelle des temps impartie au projet ;
- l'acceptabilité de ce qui a été produit.

Cette révision, en plus de vérifier que les critères de qualités pertinents ont été respectés, a deux objectifs majeurs :

- s'assurer que l'équipe de développement est sur la bonne voie ;
- amener les utilisateurs à tous les niveaux à s'impliquer dans le travail accompli et à venir

12.3 Gestion du processus de prototypage

Un certain nombre de questions clefs apparaissent dans la gestion du processus de prototypage.

Le nombre d'itérations

En règle générale, il est recommandé de permettre trois itérations :

- **investigation**, pour savoir si l'approche choisie est la bonne ;
- **affiner**, pour construire sur les commentaires et l'information en retour après le prototype d'investigation ;
- **consolidation**, pour satisfaire pleinement les objectifs du prototype.

Ceci implique trois prototypes conduisant à un composant Modèle fonctionnel et trois prototypes conduisant à une partie du Système livré. Toutefois, ce nombre de prototypes sera manifestement réduit si la nature de l'application et/ou l'environnement technique entraînent la fusion de la modélisation fonctionnelle et de la conception et du développement. Il est généralement conseillé d'éviter plus de trois itérations dans la mesure où cela pourrait conforter l'impression que les choses peuvent attendre.

Collaboration et consensus

Le prototypage requiert une collaboration et un consensus entre les développeurs et les utilisateurs, ce qui est à distinguer très clairement du contrat négocié au début du processus car ceci peut fréquemment être la source de confrontations entre les développeurs et les utilisateurs.

Choisir ce qui doit faire l'objet d'un prototype

Le chef de projet et l'équipe décident, lorsqu'ils produisent le Plan global de prototypage où les catégories de prototypes (et toutes combinaisons de prototypes) sont les plus appropriées étant donné l'environnement du business et l'environnement technique.

L'approche horizontale ou l'approche verticale

Lorsque l'équipe DSDM a choisi le domaine à prototyper, elle doit, en premier lieu, déterminer si les prototypes seront construits sur un modèle horizontal ou vertical. Il s'agit simplement d'un choix de découpage de l'application pour que toutes les catégories de prototypes apparaissent dans une tranche d'application horizontale ou verticale.

Dans l'approche horizontale, c'est d'abord l'ensemble du système informatique qui est construit à un haut niveau pour vérifier qu'il pourra couvrir l'intégralité du projet. Le détail est intégré plus tard. Avec l'approche horizontale, une section du système informatique est construite de façon incrémentielle jusqu'à ce qu'elle soit parfaitement comprise puis la section suivante est entamée.

Ces deux approches ont chacune leurs avantages et désavantages. Si tout le concept du système/ secteur du business est neuf une approche horizontale peut être plus sûre. Si la structure d'ensemble du système est bien comprise mais que le détail est vague, l'approche verticale pourrait être meilleure. L'un des avantages de l'approche verticale est qu'elle permet le déploiement incrémentiel de sous-systèmes qui apportent un bénéfice immédiat pour l'entreprise. En général, les systèmes seront construits en associant les approches horizontale et verticale. Le chef de projet en collaboration avec l'équipe devrait sélectionner la catégorie de prototype appropriée et l'approche à mettre en œuvre

13 Descriptions des produits DSDM

13.1 Introduction

Ce chapitre contient des descriptions globales de produits pour chacun des produits définis dans le cadre de le cadre du processus de développement DSDM. Chaque description de produit comporte les indications suivantes :

- le moment de production ;
- qui pourrait être chargé d'accepter le produit ;
- le but du produit ;
- une série de questions qui peuvent être posées pour assurer l'adéquation entre le produit et son objectif.

Etant donné la grande diversité des environnements dans lesquels DSDM peut être utilisé, les détails relatifs à la construction spécifique des produits ou à ce à quoi ils devraient ressembler ne sont pas fournis dans les descriptions afin qu'elles puissent s'appliquer à tout environnement.

Chaque phase du squelette du processus de développement DSDM génère un certain nombre de produits livrables. Dans certains cas, ces produits livrables sont des produits essentiels mais d'autres produits « associés » peuvent être nécessaires à leur support. Par exemple, un prototype fonctionnel peut être le produit principal d'une phase spécifique et d'autres produits associés peuvent inclure enregistrements de révision et/ou de tests dont la fonction première est de montrer que le principal produit a été validé correctement. (Dans ce cas les produits associés peuvent avoir une certaine valeur pendant la maintenance). Les produits associés peuvent avoir une plus grande importance dans certaines entreprises que dans d'autres. Les organisations sous Assurance Qualité certifiée, par exemple, vont certainement avoir besoin de conserver certains enregistrements de qualité.

De nombreux autres produits peuvent dériver d'un projet de développement. Particulièrement, ce qui est fréquent, des produits qui gouvernent la relation entre l'organisation de développement et l'organisation utilisatrice ainsi que des produits générés ou utilisés par la direction du projet pour contrôler et /ou surveiller l'activité. Ces produits ne sont pas traités plus avant dans cette section du manuel dont l'objectif principal est de traiter des produits qui sont spécifiquement pertinents pour la communauté utilisatrice.

Là où le développement est sous-traité par une organisation de développement séparée, certains produits peuvent devoir être cloisonnés pour protéger les sensibilités commerciales des parties concernées.

Le Tableau 13.1 identifie quels produits sont générés par chaque phase DSDM. L'Annexe B contient également certains produits tests à prendre en considération.

Phase DSDM	Produits
Etude de Faisabilité	Rapport de faisabilité, avec le support éventuel d'un Plan global de faisabilité de prototype
Etude du Business	Définition du domaine d'activité du business (y compris la Liste des besoins hiérarchisés) Définition de l'architecture de système Plan global de prototypage
Modèle Fonctionnel Itératif	Modèle fonctionnel (y compris les prototypes fonctionnels et la Liste des besoins non fonctionnels), avec les documents justificatifs de révision et /ou des tests enregistrés Stratégie de mise en œuvre (y compris la stratégie de chargement de données si

	approprié) Rapport d'analyse de risque de développement
Conception et Réalisation Itératives	Conception de prototypes (produits intermédiaires), accompagnée des documents justificatifs de révision Système testé, avec les documents justificatifs de tests.
Mise en Œuvre	Documentation utilisateur, indiquant comment le système informatique doit être utilisé, avec l'identification de toutes procédures de soutien nécessaires Système livré, avec les documents justificatifs de construction, de livraison et d'acceptation Population d'utilisateurs formés (y compris les opérateurs et le personnel d'accompagnement) Document de révision du projet

Tableau 13.1: Les phases DSDM et leurs produits associés

13.2 Développer les descriptions de produits

- 13.3 Les descriptions de produits sont génériques et doivent être affinées en fonction des business spécifiques et des environnements de développement de chaque organisation utilisant DSDM. Cet affinage peut être réalisé de façon incrémentielle avec le développement de l'expérience dans l'utilisation de DSDM. Une attention particulière doit être portée à l'élaboration des descriptions de produits. Elles ne doivent pas être si contraignantes que la flexibilité incorporée à DSDM aille se perdre dans les méandres d'une bureaucratie inutile. Voir au Chapitre 22 les techniques de modélisation suggérées.

Il faut également noter que beaucoup de produits, bien que générés à une étape spécifique du cycle de vie, devront faire l'objet de maintenance au cours des étapes subséquentes.

13.3 Descriptions de produits

La numérotation des descriptions de produits indique si les produits sont «indépendants» ou s'ils font partie d'un ensemble de produits. La Définition du domaine d'activité du business, par exemple, porte le numéro PD.3 qui indique qu'elle est complète en soi. La Liste hiérarchisée des besoins porte le numéro PD.3a, ce qui indique qu'elle fait partie de la Définition du domaine d'activité du business, mais est suffisamment importante pour être traitée comme un produit distinct.

PD.1 Rapport de Faisabilité

Réalisé dans :	L'Etude de Faisabilité
Accepté par (suggestion) :	Visionnaire

Objet

Donner les grandes lignes du problème à traiter par le nouveau système.

Définir la portée du projet ou de la série de projets.

Donner une indication préliminaire de tout secteur qui se trouve inclus dans la portée du projet qui peut être souhaitable mais qui n'est pas essentiel.

Enoncer, du moins dans les grandes lignes, L'étude de cas du Business pour le ou les projet(s) – y compris lorsque c'est possible, les coûts, les bénéfices, les attentes et les risques (quantifiables ou non)

Indiquer les solutions alternatives qui ont été ou pourraient être envisagées

Définir les principaux produits que devra livrer le projet.

Rapport sur l'adéquation de l'utilisation de DSDM pour le projet, ce qui peut varier avec chaque solution.

Documenter les objectifs du projet y compris les critères de performance du processus.

Documenter les contraintes de haut niveau technique et du business, ex, échelle des temps, plates-formes matérielles et logicielles.

Identifier si le système est lié à la gestion de sécurité ou s'il peut y avoir des problèmes de gestion de sécurité liés aux produits.

Décrire à un niveau élevé les processus du business et de données qui devront être automatisés.

Identifier à un niveau élevé l'interface nécessaire pour les données et applications existantes.

Identifier les processus et/ou système du business (automatisés ou non) qui pourraient être affectés par le nouveau système et qui devront peut-être être changés pour sa prise en charge.

Définir le temps de vie attendu du système informatique et donc les exigences de maintenabilité telles que définies dans la Section 11.3.

Critères de qualité

1. La définition du problème s'accorde-t-elle aux besoins de l'encadrement utilisateur ?
2. La portée du projet est-elle suffisamment claire pour être définie dans l'Etude du Business ?
3. Les objectifs du business à atteindre par le développement sont-ils clairement définis ?
4. La solution apportée au problème, telle qu'énoncée dans les principaux produits à livrer et dans les objectifs du projet, est-elle réalisable tant en termes techniques qu'en termes du business ?
5. S'agit-il d'un cas pour lequel l'utilisation de DSDM est valide, à savoir, l'application satisfait-elle aux critères d'utilisation de DSDM tels qu'énoncés dans le Chapitre 4
6. L'encadrement accepte-t-il ce qui a été inclus et ce qui a été exclu de la portée du projet ?
7. Les systèmes associés et leurs interfaces ont-ils tous été identifiés ? Est-ce qu'un impact sur ces systèmes est acceptable ?

PD.1a

Réalisé dans :	Etude de Faisabilité
Accepté par (suggestion) :	Visionnaire, Coordinateur technique

Objet

Etayer les résultats du Rapport de faisabilité

Fournir une visualisation du nouveau système informatique envisagé

Critères de qualité

1. Le prototype ajoute-t-il de la valeur au Rapport de faisabilité ?
2. Si le prototype est automatisé, aide-t-il à évaluer l'adéquation de DSDM à l'environnement de l'application et du développement

PD.2 Plan Global

Réalisé par:	L'Etude de Faisabilité
Accepté par (suggestion) :	Sponsor exécutif (parrain), Visionnaire, Coordinateur technique

Objet

Donner à la direction une idée des implications financières et en matière de ressources (développeur et utilisateur) du projet envisagé.

Fournir une base d'accord sur l'échelle des temps pour les activités de développement proposées.

Définir les critères d'acceptation pour les produits livrables considérés, à savoir, le système sera conforme à tous les besoins convenus.

Assurer que la Direction est consciente de la nécessité pour l'équipe de développement (y compris les utilisateurs) d'avoir un pouvoir de décision.

Lister les installations et les locaux spécifiques dont l'équipe ou les équipes de développement auront besoin (par exemple, des salles propres dans un même lieu)

Définir l'adaptation requise de l'approche DSDM pour ce projet

Confirmer les approches concernant la gestion de configuration, le changement de contrôle, les tests et la gestion du risque.

Identifier toutes les questions relatives à la mise en œuvre du système, en particulier pour ce qui concerne, entre autres, le chargement de données et le passage à l'utilisateur.

Identifier les standards et les lignes directrices requises par le projet, par exemple, pour le codage et pour le style d'interface utilisateur ainsi que celles qui existent déjà sous une forme qui convient.

Critères de qualité

1. Les estimations de l'effort sont-elles réalistes au vu des détails contenus dans le Rapport de faisabilité ?
2. L'estimation de l'échelle des temps est-elle cohérente avec les besoins du business pour le projet ? Réciproquement, les besoins du business ont-ils été abordés en termes de ce qui sera livré et du moment où cela sera fait ?
3. L'encadrement utilisateur est-il capable d'engager le niveau de ressources requis de chaque type d'utilisateur aux moments précisés dans le plan du projet ?
4. La gestion du développement est-elle en mesure d'engager le niveau et le type de ressources requises pour satisfaire à l'échelle des temps du Plan Global ?
5. Les approches de la gestion de configuration, du changement du contrôle et des tests sont-elles alignées sur les procédures existantes et dans le cadre des lignes directrices pour DSDM ?
6. Tous les équipements et installations nécessaires seront-ils disponibles comme requis ?

7. Les critères d'acceptation sont-ils assez clairs et sont-ils suffisamment rigoureux pour définir la qualité des « livrables » tout en permettant une flexibilité des besoins pendant le développement ?
8. Toutes les normes et lignes directrices requises sont-elles identifiées et disponibles ; pour celles qui ne seraient pas encore disponibles, le plan prévoit-il des ressources suffisantes pour leur développement ou leur obtention ?

PD.3 Définition du Domaine d'Activité Industrielle

Réalisé dans :	L'Etude du Business
Accepté par (suggestion) :	Visionnaire, Utilisateur(s) ambassadeur(s), Chef de projet

Objet

Identifier les besoins du business qui doivent être supportés par le système informatique envisagé.

Affiner le cas global du business (documenté dans le Rapport de faisabilité) pour inclure les analyses concernant les bénéfices, les risques, les coûts et l'impact.

Donner les grandes lignes des besoins informatiques des processus du business qui seront supportés.

Identifier les catégories d'utilisateurs sur lesquels le développement et l'introduction du système proposé aura un impact.

Identifier les processus et les scénarios du business qui doivent changer.

Clarifier toutes les interfaces avec les autres systèmes (manuels ou automatisés)

Vérifier que le système proposé est toujours en adéquation au développement avec DSDM (adapté si nécessaire).

Critères de qualité

1. Le contexte, les processus et les objectifs du business sont-ils définis et ont-ils fait l'objet d'un accord ?
2. Tous les besoins identifiés à ce point ont-ils été hiérarchisés (y compris les besoins non fonctionnels) ?
3. Toutes les priorités ont-elles été assignées en collaboration avec les utilisateurs ?
4. Les critères de haut niveau pour l'acceptation du Système livré ont-ils été définis ?
5. Les secteurs du business sont-ils clairement documentés, y compris les besoins informatiques de haut niveau qui sont affectés par le système ?
6. Est-ce que la frontière envisagée pour le projet de nouveau système est réaliste dans l'échelle des temps ?
7. Toutes les catégories d'utilisateurs affectés par le nouveau système sont-elles identifiées ?
8. Les besoins informatiques et de traitement du système projeté sont-ils définis au moins dans leurs grandes lignes ?
9. Est-il toujours évident que les besoins du business sont traités par le projet de nouveau système ?
10. Pour chaque processus du business, la personne qui en est responsable est-elle identifiée ?
Peuvent-elles consacrer les ressources et le temps nécessaire au projet ?

11 Tous les grands événements du business ont-ils été identifiés ?

PD.3a Liste hiérarchisée des besoins

Objet

Fournir au Modèle fonctionnel itératif une liste hiérarchisée des besoins (fonctionnels et non fonctionnels) à examiner.

Identifier le sous-ensemble utilisable minimum de fonctionnalité de support pour le business. (la livraison de ce sous-ensemble est garantie.)

Critères de qualité

1. Tous les besoins identifiés à ce point ont-ils été hiérarchisés ?
2. Toutes les priorités ont-elles été assignées en collaboration avec les utilisateurs ?
3. Si le développement porte sur un projet, est-ce que l'encadrement utilisateur accepte l'éventualité que les solutions basses à des besoins de faibles priorités ne soient pas nécessairement livrées (du moins dans un premier temps) ?
4. Si le développement porte sur un ensemble de projets, est-ce que l'encadrement utilisateur accepte la possibilité que les solutions aux besoins de faible priorité ne soient livrées qu'à une date ultérieure ?

PD.4 Définition de l'Architecture de Système

Réalisé dans :	Etude du Business
Accepté par (suggestion) :	Coordinateur technique, Chef de projet, Coordinateurs des opérations

Objet

Fournir une compréhension commune des architectures techniques qui seront utilisées pendant le développement et la mise en œuvre.

Décrire la plate-forme cible et (si différente) la plate-forme développement.

Donner une description globale de l'architecture logicielle (à savoir, les principaux modules ou composants logiciels – traitement et données – et leurs interactions).

Critères de qualité

1. L'architecture est-elle adaptée aux besoins ?
2. Les risques de l'architecture considérée ont-ils été correctement étudiés – en particulier, tous ses composants sont-ils disponibles et mutuellement compatibles ?
3. La migration de la plate-forme de développement vers la plate-forme cible pourra-t-elle se produire sans heurts ? Si non, tous les problèmes prévisibles ont-ils été identifiés ?
4. L'architecture logicielle globale est-elle suffisamment bien définie pour donner aux développeurs une idée pointue du projet de système informatique ?
5. L'architecture est-elle définie à un niveau approprié pour qu'elle ne soit pas trop vulnérable aux changements à mesure que le projet progresse ?
6. A-t-on tiré parti de toutes les possibilités de réutilisations des composants existants ?

7. Peut-on compter sur l'architecture pour faire face aux exigences de performance, de capacité et de résilience ?

PD.5 Plan Global de Prototypage

Réalisé dans :	Etude du Business
Accepté par (suggestion) :	Sponsor exécutif (parrain), Visionnaire, Coordinateur technique

Objet

Affiner le Plan global pour le Modèle fonctionnel itératif et la Conception et le Développement itératifs

Fournir une stratégie de développement aux analystes du business et aux concepteurs.

Définir quelles catégories de prototypes vont être développées et quand.

Hiérarchiser les activités de prototypage.

Définir les mécanismes pour décider du moment de l'arrêt d'une activité de prototypage spécifique.

Identifier les personnes qui vont assumer les divers rôles et responsabilités des phases suivantes du projet.

Identifier quels articles doivent être soumis à la gestion de configuration et donner les grandes lignes de l'application du contrôle de configuration.

Définir l'approche qui sera adoptée pour tester : quels types de tests seront conduits, comment ils seront prescrits et consignés par écrit.

Critères de qualité

1. L'échelle des temps est-elle cohérente avec les objectifs du business énoncés dans le Rapport de faisabilité et la Définition du domaine d'activité du business ?
2. L'ordre des activités du plan de prototypage correspond-il à la liste hiérarchisée des besoins ?
3. Le plan reflète-t-il la nécessité de traiter les zones à risque au moment approprié ?
4. Est-ce que toutes les catégories d'utilisateurs affectés figurent dans le plan de prototypage ?
5. L'effort utilisateur envisagé est-il consistant avec les besoins des processus existants du business et le développement ?
6. L'ensemble du personnel sera-t-il disponible pour fournir l'effort nécessaire au moment voulu ?
7. La sélection des catégories de prototypes est-elle réalisable dans l'environnement de développement attendu ?
8. Les articles de configuration sont-ils appropriés à la fonctionnalité de la Définition du domaine d'activité du business et aux produits livrables du Plan global ?
9. La méthode de contrôle de configuration est-elle adaptée à l'environnement ?
10. L'étendue, la profondeur et la formalisation des tests est-elle appropriée ?

PD.6 Modèle Fonctionnel

Réalisé dans :	Modèle Fonctionnel Itératif
----------------	-----------------------------

Accepté par : (suggestion) :	Visionnaire, Utilisateur(s) ambassadeur(s), Utilisateur(s) conseiller(s), Coordinateur technique
---------------------------------	--

Objet

Fournir une démonstration cohésive des besoins de fonctionnalité et de données à satisfaire, y compris de toutes les contraintes connues au moment.

Démontrer la faisabilité de réalisation des besoins non fonctionnels.

Critères de qualité

1. Le Modèle fonctionnel correspond-il aux besoins des utilisateurs tels que définis au cours des discussions et des sessions de prototypage ?
2. Est-il dans le cadre du développement tels que défini dans la Définition du domaine d'activité du business ?
3. Toutes les parties du Modèle fonctionnel sont-elles compatibles entre elles ?
4. Le modèle contient-il un sous-ensemble minimal utilisable ?
5. Les aspects essentiels de l'intégrité et de la sécurité sont-ils intégrés au Modèle fonctionnel ?
6. Les besoins pour l'administration du système sont-ils visibles ?
7. Tous les modèles statiques (ex : modèles d'information) sont-ils cohérents avec le prototype fonctionnel et réciproquement ?
8. Le modèle permet-il d'être confiant dans la réalisation des niveaux corrects de performance, de capacité et de maintenabilité ?
9. La documentation de soutien nécessaire et à un niveau adéquat est-elle disponible ?

PD.6a Prototype Fonctionnel

Réalisé dans :	Modèle Fonctionnel Itératif
Accepté par (suggestion) :	Visionnaire, Utilisateur(s) ambassadeur(s)

Objet

Fournir une première version de composant de système qui contienne la quasi-totalité de la fonctionnalité requise pour supporter les processus du business. (Elle ne doit pas nécessairement satisfaire aux besoins non fonctionnels, bien que cela permette d'être certain que ce sera réalisable.)

Permettre aux utilisateurs de comprendre les possibilités offertes par le système.

Permettre aux utilisateurs de comprendre facilement à quoi on s'attend qu'ils réagissent afin qu'ils soient en mesure d'opérer le nouveau système informatique efficacement.

Fournir la base d'un accord avec les utilisateurs à tous les niveaux en ce qui concerne la direction prise par le projet.

Critères de qualité

1. Les composants écrits du Prototype fonctionnel, quels qu'ils soient, sont-ils réalisables dans l'environnement de développement ?
2. Toutes les interfaces système importantes sont-elles apparentes, au moins dans leurs grandes lignes ? Leur mise en œuvre par la suite semble-t-elle possible ?

3. Tous les besoins essentiels du processus du business sont-ils identifiables dans le Prototype fonctionnel ? S'ils ne le sont pas tous, la documentation de support est-elle disponible ?
4. Tous les besoins essentiels en information sont-ils identifiables dans le Prototype fonctionnel ? S'ils ne le sont pas tous, la documentation de support est-elle disponible ?
5. Là où les besoins non fonctionnels ont été traités par le Prototype fonctionnel sont-ils clairement décrits ?

PD.6b Liste des Besoins non Fonctionnels

Réalisé dans :	Modèle Fonctionnel Itératif
Accepté par (suggestion) :	Utilisateur(s) ambassadeur(s), Coordinateur technique

Objet

Affiner et développer les besoins non fonctionnels pour la Conception et le développement itératif (même s'ils ont été satisfaits dans un Prototype fonctionnel).

Critères de qualité

1. Tous les besoins non fonctionnels sont-ils suffisamment quantifiés ?
2. Là où les besoins non fonctionnels ont été traités par le Prototype fonctionnel sont-ils notés comme tels dans la liste des besoins non fonctionnels ?
3. Toutes les zones identifiées dans les contraintes de haut niveau du Rapport de faisabilité ont-elles été examinées ?
4. L'ensemble constitué par les besoins non fonctionnels est-il complet et cohérent en lui-même et avec le reste du Modèle fonctionnel ?
5. Est-ce que tous les besoins non fonctionnels ajoutent de la valeur aux processus du business ?
6. Les besoins non fonctionnels sont-ils réalistes et réalisables ?

PD.6c Enregistrements de révision du Modèle Fonctionnel

Réalisé dans :	Modèle Fonctionnel Itératif
Accepté par (suggestion) :	Chef de projet, Chef de la qualité

Objet

Enregistrer les informations en retour des utilisateurs pour tous les modèles fonctionnels et les prototypes.

Contribuer à la planification et à l'exécution des activités de conception.

Souligner quels secteurs peuvent être mis en œuvre dans un travail de développement futur.

Contribuer à tout développement futur en permettant d'éviter des pièges similaires à ceux qui peuvent avoir été rencontrés à ce point dans ce projet.

Critères de qualité

1. Les enregistrements de révision couvrent-ils tous les prototypes fonctionnels ?

2. Tous les commentaires des utilisateurs sont-ils enregistrés à leur satisfaction ?
3. L'encadrement utilisateur a-t-il convenu des secteurs pour lesquels les utilisateurs ont demandé un développement plus avancé ?
4. Si des conflits non résolus concernant les besoins des utilisateurs se sont produits, ont-ils été soulignés pour des considérations de gestion et/ou techniques ?
5. Les enregistrements de révision fournissent-ils assez d'informations pour montrer où les prototypes ne répondent pas aux attentes du projet à ce point ?
6. Si certains secteurs doivent être "gelés" tels qu'ils sont (ex : une partie de l'interface utilisateur), sont-ils notés dans les enregistrements de révision ?

PD.7 Stratégie de mise en œuvre

Réalisé par :	Modèle Fonctionnel Itératif
Accepté par (suggestion) :	Coordinateur technique, Coordinateur des opérations

Objet

Affiner le plan du projet pour les étapes suivantes du développement.

Définir de façon plus détaillée les coûts et l'effort pour permettre à l'encadrement de réévaluer les bénéfices du développement.

Critères de qualité

1. Le calendrier de la conception et du développement s'accorde-t-il toujours aux besoins du business ?
2. L'estimation des coûts et de l'effort (développeur et utilisateur) semble-t-elle réaliste ?
3. Les ressources nécessaires (développeurs et utilisateur) pour satisfaire à ce plan sont-elles disponibles ?
4. Si pertinent, est-ce que les procédures de transmission au personnel de maintenance et de support sont claires ?
5. Si pertinent, est-ce que les exigences pour le chargement des données et/ou de transfert système ont été étudiées de façon appropriée ?

PD.8 Rapport d'Analyse de Risque du Développement

Réalisé dans :	Modèle Fonctionnel Itératif
Accepté par (suggestion) :	Sponsor exécutif (parrain), Visionnaire, Utilisateur(s) ambassadeur(s)

Objet

Aider la direction à décider de l'avenir du projet.

(Note : L'analyse du risque commence au début du projet, mais étant donné la brièveté de l'échelle des temps des projets DSDM jusqu'à la livraison d'un incrément, ce point est le plus tardif auquel le développement de stratégies pour éviter ou contenir le risque sont

envisageables. A partir de ce point, les risques qui apparaîtront seront vraisemblablement traités de manière plus réactive.)

Critères de qualité

1. Est-ce que tous les facteurs affectant potentiellement le succès du projet ont fait l'objet de débats ?
2. Les risques sont-ils suffisamment quantifiés pour qu'une décision soit prise ?
3. Est-ce qu'une contre-mesure, au moins, est associée à chaque risque identifié ?

PD.9 Prototype de Conception

Réalisé dans :	Conception et Réalisation Itératives
Accepté par (suggestion) :	Coordinateur technique, Utilisateur(s) ambassadeur(s), Utilisateur(s) conseiller(s)

Objet

Donner aux développeurs l'assurance qu'une stratégie spécifique de conception est possible.

Donner aux utilisateurs (particulièrement aux utilisateurs cadres) la preuve que le développement se fait dans la bonne direction.

Donner aux utilisateurs l'occasion d'aider à améliorer le système en fournissant des informations en retour aux développeurs.

Critères de qualité

1. Le prototype traite-t-il de façon satisfaisante les questions voulues ?
2. Est-ce que tous les risques, si l'on va de l'avant avec ce prototype, sont clairement identifiés ?
3. Le prototype se conforme-t-il à tous les besoins applicables de l'utilisateur ?
4. Le prototype est-il conforme à tous les standards et lignes directrices applicables du développement ?

PD.9a Enregistrement de révision de Prototype de Conception

Réalisé dans :	Conception et Réalisation Itératives
Accepté par (suggestion) :	Chef du projet, Chef de la qualité

Objet

Enregistrer les informations en retour des utilisateurs pour toutes les conceptions de prototypes.

Contribuer à éviter aux développements futurs de tomber dans aucuns des pièges qui auraient pu être rencontré.

Signaler et planifier tout secteur qui devrait être mis en œuvre ou ajusté soit pendant la phase de Conception/ Réalisation, soit qui pourrait être traité après la livraison du système.

Critères de qualité

1. Les documents de révision couvrent-ils toutes les conceptions de prototypes ?

2. Est-ce que toutes les remarques émanant des utilisateurs et des développeurs sont enregistrées à leur satisfaction ?

PD.10 Système testé

Réalisé dans :	Conception et Réalisation Itératives
Accepté par (suggestion) :	«Visionnaire», Utilisateur(s) ambassadeur(s), Coordinateur technique, coordinateur des opérations

Objet

Fournir un système qui exécute toutes les fonctionnalités convenues et qui satisfait à toutes les exigences non fonctionnelles convenues. (En raison des contraintes d'échelle des temps du business, le Système testé peut ne couvrir qu'un sous-ensemble convenu de tous les besoins fonctionnels et non fonctionnels.)

Fournir un système opérationnel qui peut en toute sécurité être placé dans l'environnement de l'utilisateur.

Fournir au personnel de maintenance et de support suffisamment d'informations pour réaliser des améliorations, aider les utilisateurs, accomplir des tâches de gestion de système, etc.

Critères de qualité

1. Le système satisfait-il à tous les critères d'acceptation définis par l'utilisateur ?
2. Les développeurs sont-ils convaincus que le système est suffisamment robuste pour être mis en pleine opération ?
3. Le système a-t-il été testé à un niveau approprié pour l'utilisation qui va en être faite ?
4. Peut-on constater que tous les besoins essentiels (fonctionnels et non fonctionnels) ont été testés et là où nécessaire démontrés aux utilisateurs ?
5. Est-ce que tout et chaque aspect concernant la sécurité et l'aspect sécurité des produits ont été correctement validés ?
6. Toutes les fonctionnalités fournies pour supporter la mise en œuvre ont-elles été testées de façon adéquate (en particulier a-t-on tenu compte de besoins relatifs à la conversion de données / au chargement logiciel) ?
7. Peut-on trouver l'origine de tous les composants du Système testé dans le Modèle fonctionnel ?
8. Tous les composants rejetés par les documents de révision de conception ont-ils été omis du Système testé ?
9. La documentation du système est-elle cohérente avec le logiciel ?

PD.10a Enregistrement des Tests

Réalisé dans :	Conception et Réalisation Itératives
Accepté par (suggestion):	Chef de projet, Chef de la qualité

Objet

Montrer que les tests ont été réalisés en conformité au plan de test défini dans le Plan global de prototypage.

Critères de qualité

1. Les tests sont-ils documentés de façon appropriée (par exemple, chaque test identifie-t-il les besoins et les règles du business traités par le test),
2. Si approprié, les fiches techniques des tests ont-elles été révisées ?
3. Existe-t-il des archives montrant que tous les tests requis ont été réalisés et que l'implication de l'utilisateur dans ces tests a été ce qu'elle devait être ?
4. Tous les problèmes relevés pendant les tests ont-ils été correctement identifiés et enregistrés ?
5. Les tests de régression ont-ils été conduits comme approprié ?
6. Les rapports de test contiennent-ils suffisamment de détails pour permettre de reproduire les tests à l'avenir ?

PD.11 Documentation Utilisateur

Réalisé dans :	Mise en Œuvre
Accepté par (suggestion) :	Utilisateur(s) ambassadeur(s)

Objet

Expliquer aux utilisateurs comment se servir du Système livré.

Critères de qualité

1. Est-ce que les utilisateurs disposent d'une aide sous une forme appropriée (ex : documents électroniques, documents papiers et installations d'aide) ?
2. Offre-t-elle un guide pas à pas complet et sans ambiguïté pour l'utilisation du Système livré ?
3. Couvre-t-elle toute la fonctionnalité dans le système tel qu'il est livré ?
4. Explique-t-elle l'interaction du système avec d'autres systèmes, manuels ou autres ?
5. Lorsqu'il y a plusieurs catégories d'utilisateurs, explique-t-elle qui doit lire quoi ?
6. Peut-on en faire facilement un classement par tâches dans l'entreprise ?
7. Est-elle rédigée dans la langue des utilisateurs ?
8. Si nécessaire, offre-t-elle des explications détaillées de toute procédure manuelle associée au système informatique ?
9. Est-ce qu'elle contient une aide sur ce qu'il faut faire en cas d'erreur, par exemple, qui appeler, comment récupérer et comment résoudre les problèmes ?

PD.12 Formation des groupes d'utilisateurs**Objet**

Permettre à tous les utilisateurs d'utiliser et d'opérer efficacement le système informatique ainsi que tout nouveau processus du business.

Critères de qualité

1. Les utilisateurs formés ont-ils suffisamment de connaissances et de compétences pour gérer et opérer le système ?
2. Une formation adéquate a-t-elle été donnée à tous les utilisateurs pour lesquels c'est nécessaire ?

3. En cas de besoin, un matériel de formation continue destiné aux futurs utilisateurs est-il disponible ?
4. La Documentation Utilisateur est-elle facilement accessible à tous les utilisateurs ?
5. Existe-t-il une stratégie de formation de nouveaux arrivants dans le groupe des utilisateurs ?
6. Existe-t-il une stratégie destinée à former les utilisateurs actuels aux développements futurs du système ?

PD.13 Système livré

Réalisé dans :	Mise en Œuvre
Accepté par (suggestion) :	Sponsor exécutif (parrain), Visionnaire, Utilisateur(s) ambassadeur(s), Conseiller(s) utilisateur(s), Coordinateur technique, Coordinateur des opérations

Objet

Exécuter la fonctionnalité décrite dans le Modèle fonctionnel conformément à toutes les contraintes définies par les Besoins non fonctionnels.

Critères de qualité

1. Les changements, s'il y en a, opérés sur le Système testé (voir PD.10) ont-ils été correctement autorisés, mis en œuvre et testés ?
2. Le système fonctionne-t-il comme requis dans son environnement cible ?
3. Semble-t-il opérer aux niveaux de service requis ?
4. Y a-t-il des problèmes imprévus concernant le placement du système dans l'environnement cible qui ne sont toujours pas résolus ?
5. Toutes les activités de chargement et de conversion de données ont-elles été exécutées avec succès ?
6. Tous les articles de configuration ont-ils été correctement archivés ?
7. Tous les articles de configuration sont-ils identifiés ?
8. La version correcte de chaque article de configuration est-elle enregistrée ?
9. Est-ce que tous les problèmes connus non encore résolus ont été répertoriés ?

PD.14 Document de révision du projet

Réalisé dans :	Mise en Œuvre
Accepté par (suggestion) :	Sponsor exécutif (parrain), Visionnaire, Utilisateur(s) ambassadeur(s)

Objet

Evaluer le succès du travail de développement.

Permettre la prise de décisions concernant un travail de développement à réaliser à l'avenir.

Décider des omissions délibérées qui pourraient maintenant être traitées.

Critères de qualité

1. Est-ce que tous les secteurs omis dans cet incrément ont été identifiés ?
2. Les informations suffisantes sont-elles disponibles pour permettre à l'encadrement de décider d'aller de l'avant ou non dans le travail de développement ?
3. Est-ce que toutes les leçons tirées pendant l'incrément sont répertoriées ?

14 Projet DSDM et Structures d'équipes

Dynamique d'équipe

Les bénéfices les plus importants de l'approche DSDM proviennent de l'implication des utilisateurs dans le développement de l'application. Alors, quelle est la meilleure structure d'équipe pour maximiser le bénéfice de l'implication de l'utilisateur ? Il est important que l'équipe ne soit pas trop importante, idéalement pas plus de six personnes y compris les utilisateurs. Il faut que les utilisateurs puissent disposer du temps nécessaire pour être réellement impliqués et se sentir impliqués dans le projet afin de prendre possession de l'application qui est en cours de construction.

Etant donné qu'une équipe ne devrait pas dépasser plus de six personnes, elle pourrait n'être constituée que de deux ou de trois personnes, le nombre optimum étant de trois ou quatre. Le plus important est que l'équipe «travaille» dans le contexte. Les exemples suivants ont tous été mis en pratique avec succès :

Le duo : un utilisateur et un développeur travaillant ensemble. Bien que simple, ceci peut s'avérer une méthode très efficace pour créer des partenariats forts d'utilisateur/développeur.

Deux utilisateurs, un ou deux développeurs. Les utilisateurs se stimulent mutuellement pour créer une solution «qui dépasse la somme de leurs compétences combinées » et le développeur en profite.

Un utilisateur, un développeur DSDM confirmé et un développeur sans expérience de l'approche DSDM mais qui l'acquiert par sa participation.

La décision relative à la composition de l'équipe dépend des personnalités et des circonstances pratiques. Il est important de bien peser quelle composition d'équipe sera celle qui, selon toute probabilité, produira, le plus rapidement le meilleur résultat. Le directeur de projet doit être prêt à changer cette structure si elle ne marche pas. Il est cependant nécessaire de faire attention à ce que le partenariat interrompu ne mette pas en danger le progrès des travaux à venir.

La philosophie de tolérance à l'erreur

Traditionnellement la spécification fonctionnelle est l'arbitre final de ce qui est inclus ou ne l'est pas dans la portée du business ou souhaité par lui. Lorsque les exigences sont vagues ou ambiguës, la porte est ouverte aux conflits et aux récriminations quant à savoir à qui incombe la faute, qui doit payer et à la mise en cause éventuelle du développeur qui «aurait dû savoir parce que c'était évident ». La responsabilité et le développement commun évitent tout ceci, cependant, si les rôles et les idées traditionnelles persistent, les dangers sont plus grands. Si les développeurs et les utilisateurs ne travaillent pas ensemble en équipe et s'ils restent sur leurs positions et dans leurs rôles traditionnels, alors il n'y a pas de spécification détaillée à laquelle se référer.

Il est essentiel que la responsabilité individuelle soit bien comprise et assumée, mais comme dans toute équipe si des problèmes surviennent, ils doivent être considérés comme un échec de l'équipe plutôt que celui d'une personne, et l'équipe doit poursuivre son travail pour résoudre la difficulté. Les utilisateurs et les développeurs partagent la responsabilité du développement.

Les caractéristiques du développeur

Afin que l'approche DSDM puisse être mise en œuvre, les experts en développement et les développeurs moins qualifiés doivent travailler ensemble : tous les développeurs vont travailler avec les utilisateurs. Les concepts d' «analyste» et de «programmeur » dans le sens où l'un

défini le travail en détail pour que l'autre l'encode n'existent plus ici. Les rôles traditionnels de l'analyste et du programmeur se fondent dans l'équipe. Parce qu'aucune barrière ne s'élève entre l'analyse, la conception et la programmation, les développeurs moins expérimentés apprennent rapidement une approche du développement plus centrée sur le métier et, le personnel plus expérimenté met en œuvre des compétences techniques plus pointues. Chaque développeur doit être capable de comprendre les besoins des utilisateurs et de les interpréter sous une forme informatisée. La méthode ne fonctionne que si les développeurs ont les compétences requises et que le pouvoir décisionnel en la matière leur soit délégué.

Il en découle que les compétences requises des développeurs DSDM sont plus étendues que celles des programmeurs traditionnels. Ils sont non seulement capables d'écouter ce qui est dit et de transformer les mots en une conception fonctionnelle et en modèles efficaces, mais font preuve de solides compétences relationnelles. Ils doivent être modestes, bons communicateurs et bons diplomates. Ils doivent être capables d'admettre qu'ils ont mal interprété les besoins et être prêts à changer, même si cela entraîne le rejet de leurs travaux précédents.

14.4 Le cadre du projet DSDM

Le schéma 14.1 montre les relations qui peuvent exister entre les rôles DSDM et l'organisation ainsi que l'infrastructure du projet. Les rôles sont définis au Chapitre 15.

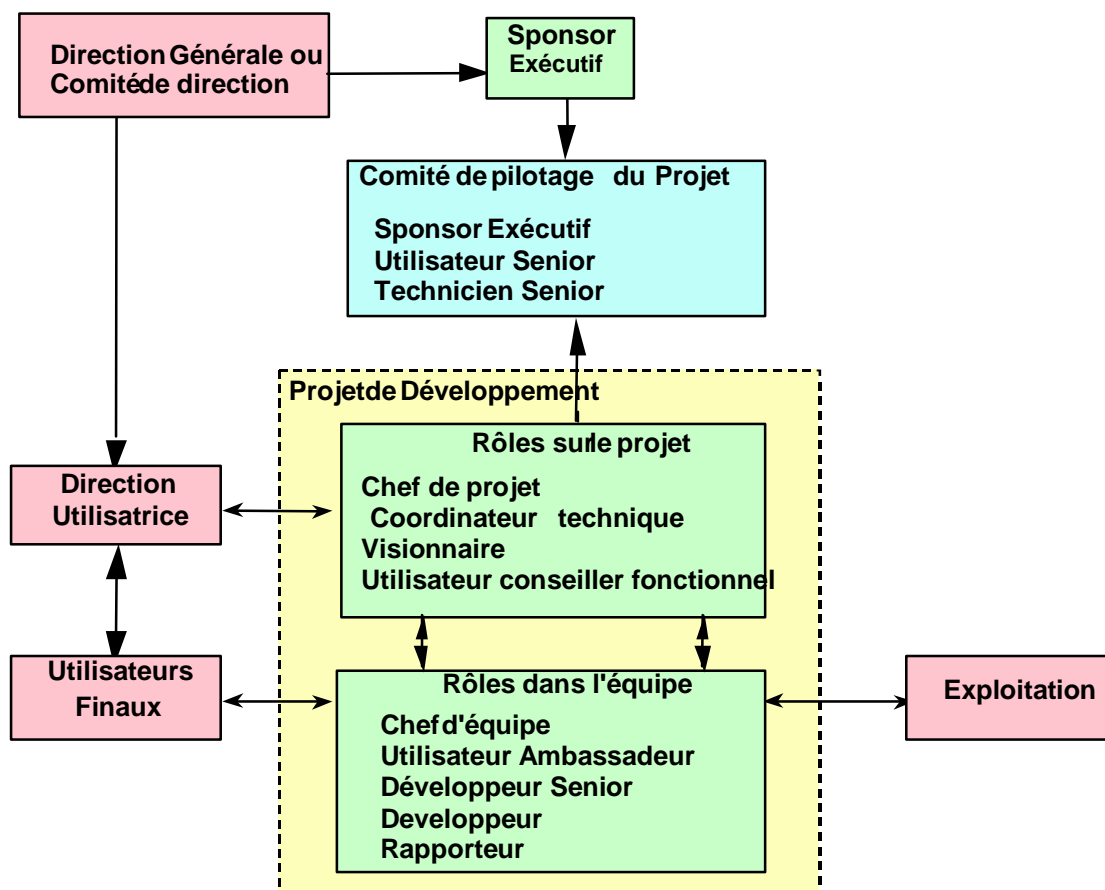


Schéma 14.1 : L'organisation du projet DSDM

Les schémas 14.1 et 14.3 présentent l'exemple de deux structures de projets possibles montrant comment une personne peut remplir plusieurs rôles. Les traits représentent les lignes habituelles de communication plutôt que la structure de dépendance. Les traits pleins indiquent une participation totale, les traits en pointillé indiquent que la participation n'est pas

à plein temps. Pour simplifier le réseau de communication dans l'équipe de développement, les équipes sont représentées comme un anneau avec des points de raccordement.

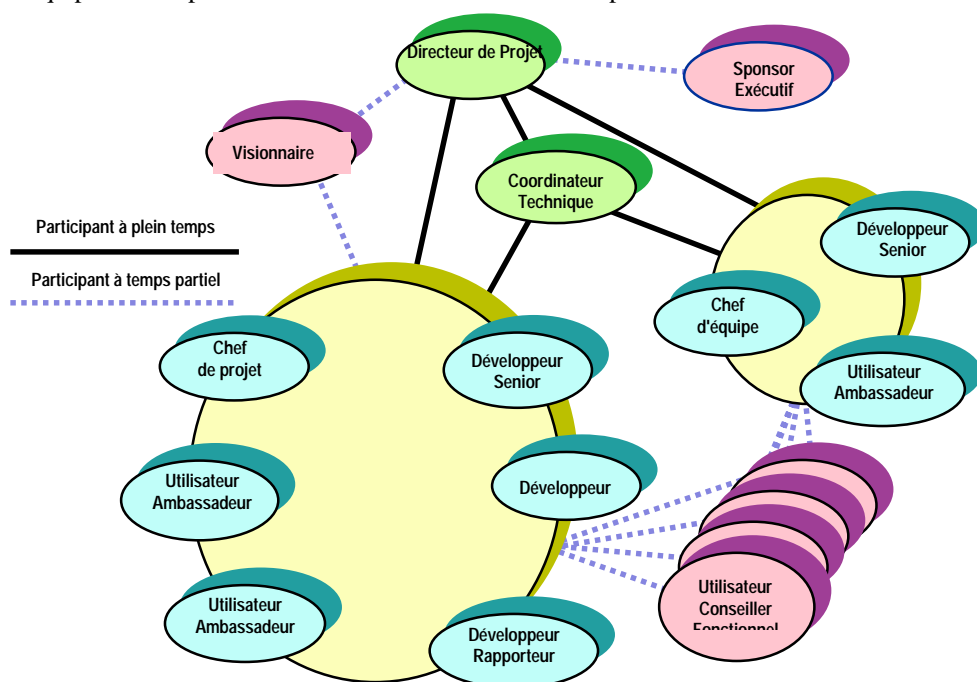


Schéma 14.2 : *Projet à plusieurs équipes incluant plus d'un sous-ensemble de développement par équipe*

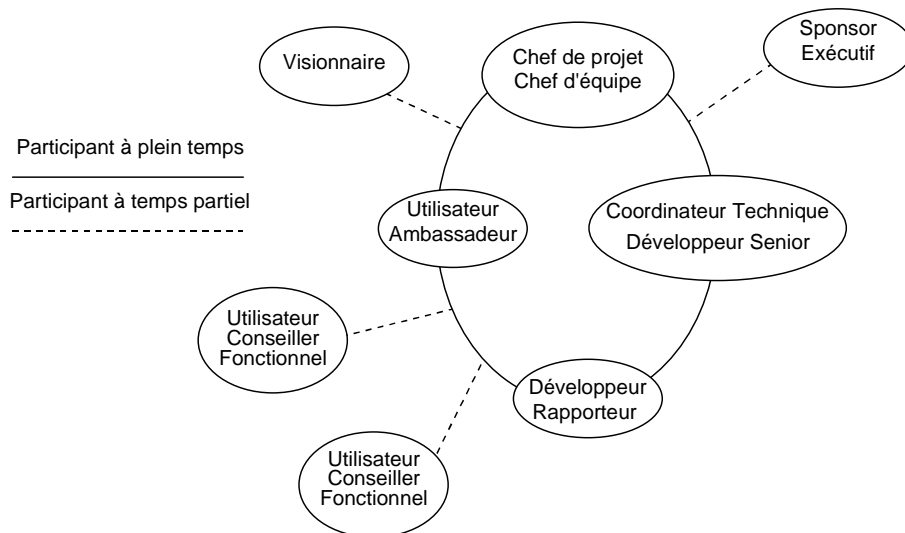


Schéma 14.3 : *Projet de plus petite envergure avec seulement une équipe, un regroupement de rôles plus important mais des responsabilités relationnelles similaires.*

14.5 Gestion de l'équipe

La question se pose dans une équipe DSDM de savoir comment organiser la structure et la gestion de l'équipe. Dans une équipe d'égaux, les rôles sont examinés et attribués par les membres de l'équipe (y compris le rôle de chef d'équipe) sur la base d'une évaluation des aptitudes inhérentes, des caractéristiques personnelles et de l'expérience de chaque personne.

En général, les structures classiques hiérarchisées des équipes n'encouragent pas des pratiques de travail flexibles et des communications ouvertes qui sont essentielles à l'approche de travail DSDM, cependant, les membres de l'équipe peuvent décider d'adopter la structure, quelle qu'elle soit, qui leur convient le mieux. Les équipes DSDM étant des structures sociales temporaires qui doivent être efficaces dès leur mise en place, il est essentiel de fournir des efforts supplémentaires pour instaurer l'esprit d'équipe dès sa formation. Les activités destinées à renforcer la cohésion de l'équipe sont fortement recommandées et ce le plus tôt possible et chaque fois qu'il est nécessaire de faire appel à un nouveau membre.

Le chef de projet peut venir du business ou du service informatique. Toutefois, en ce qui concerne le chef de projet les règles de base sont impérativement les suivantes, il doit ;

comprendre les questions relatives au business ;

pouvoir comprendre le point de vue des utilisateurs ;

comprendre les questions techniques.

Cette dernière considération implique qu'habituellement les chefs de projets viennent du milieu de la l'informatique et que leurs compétences dans le développement de systèmes informatiques ont été acquises au cours de nombreuses années d'expérience. Il n'est pas réaliste d'attendre du personnel utilisateur qu'il comprenne tous les points liés à la l'informatique qui doivent être traités sans, sous une forme ou une autre, le soutien direct de l'informatique. Pour les grands projets, le chef de projet ne ferait pas partie d'une équipe et pourrait gérer plus d'un projet simultanément.

L'arbitration est l'un des aspects importants du rôle de chef de projet. Lorsqu'il y a des discussions sur les priorités ou les mérites relatifs de certains aspects du système en développement, le chef de projet, sans toutefois prendre de décision, aide les parties concernées à décider de la meilleure approche pour le projet. Attitude très difficile à assumer pour des chefs de projets qui ont établi leur réputation en menant à bien des projets, en dépit des difficultés, grâce à une approche autoritaire de la gestion. Ces compétences sont toujours pertinentes au niveau d'un programme ; c'est le contact quotidien avec les équipes au niveau du projet qui doivent être abordées différemment.

Au cours des premières étapes du projet, motiver l'équipe ne devrait pas présenter de problèmes. Néanmoins, l'aiguillon de la productivité peut disparaître si les choses ne se passent pas bien. Les membres utilisateurs de l'équipe sont particulièrement aptes à perdre leur enthousiasme initial car ils ne sont pas habitués aux nombreux revers qu'un projet informatique peut rencontrer. Afin d'éviter que cette situation se produise, l'équipe devrait organiser des activités sociales permettant à l'équipe de se réunir et de parler d'autre chose que du travail.

14.6 Entre les projets de développement

Une période de détente doit être accordée entre chaque projet à tous les développeurs de projets RAD afin d'éviter leur «épuisement ». Ce qui ne signifie pas qu'ils n'accomplissent pas des tâches utiles mais simplement qu'ils doivent bénéficier d'une période pendant laquelle ils sont moins soumis aux pressions de produire dans de très brefs délais.

On peut envisager plusieurs types d'activités. L'une des plus évidente est la formation, une autre est l'amélioration et la reconstruction de composants identifiés pendant le projet comme étant potentiellement réutilisables. Ces deux activités sont moins stressantes que le travail sur un projet RAD mais améliorent la capacité de l'organisation à livrer plus rapidement des systèmes informatiques de qualité.

14.7 Développement de carrière et formation

La structure organisationnelle d'une équipe de projet DSDM est essentiellement horizontale. Quelqu'un est bien désigné comme chef de projet, ce qui représente certaines catégories de responsabilités plutôt qu'une autorité selon la hiérarchie traditionnelle. Ce qui peut causer une

certaine confusion par rapport aux emplois, aux titres et aux échelons occupés dans un service. Cela n'a pas lieu d'être.

En raison d'un possible «épuisement», la participation à des projets DSDM constituera rarement le seul emploi d'une personne donnée. La nature fondamentale d'une équipe DSDM est de rassembler temporairement un groupe de personnes dans le but de mener à bien une tâche précise. Par conséquent, une distinction très claire doit être faite entre les rôles assumés au cours d'un projet DSDM et les fonctions ou grades dans l'entreprise.

Le développement de carrière devrait se poursuivre normalement selon les lignes traditionnelles, cependant, avec l'expérience acquise (à condition que la personne concernée fasse preuve des capacités personnelles nécessaires pour traiter des tâches DSDM) des rôles comportant davantage de responsabilités peuvent être assumés dans les projets DSDM.

Une grande expérience en développement de l'informatique et une connaissance approfondie du métier sont nécessaires avant qu'un développeur puisse assumer la responsabilité du développement d'un secteur d'activité. Il est recommandé qu'une personne ne soit nommée développeur en chef ou chef d'équipe DSDM qu'après, non seulement, avoir suivi une formation officielle adéquate mais, également après avoir acquis une expérience significative dans un projet général ainsi qu'une expérience de rôles moins élevés dans des projets DSDM. L'idéal serait aussi que cette personne ait passé un certain temps dans le service utilisateur à comprendre le travail du point de vue de l'utilisateur.

« Mon équipe DSDM refuse de travailler autrement. »

15 Les rôles DSDM

15.1 Le rôle prévaut sur «l'individuel »

Les grandes lignes des rôles qu'il est recommandé d'attribuer sont définies dans ce chapitre. Soulignons que ces rôles ne correspondent pas nécessairement chacun à une et une seule personne. En effet, une personne peut couvrir un ou plusieurs rôles et un rôle peut être réparti entre deux ou plusieurs personnes. Pour un grand projet, par exemple, les responsabilités du coordinateur technique seront réparties entre plusieurs personnes, à savoir, le concepteur/l'architecte de système, Le chef de la qualité du projet et le chef de la configuration du projet.

Un grand projet est facilité par une répartition des rôles sur plusieurs équipes de développement travaillant simultanément ou se chevauchant en termes du cadre temporel de développement. Pour des projets plus simples, certains rôles comme celui de Chef de projet et de Chef d'équipe seraient vraisemblablement fusionnés.

Il est évident que, par exemple, des contraintes géographiques et de disponibilité du personnel peuvent avoir de conséquences sur la mise sur pied de l'équipe idéale, mais il est fortement recommandé que tous les rôles soient pris en considération et attribués comme approprié. De plus, il est impératif que les personnes concernées prennent pleinement conscience des responsabilités qui leur incombent.

15.2 Définition des rôles DSDM

Sponsor exécutif (parrain)

Attributs

Capacité d'engager les fonds et les ressources nécessaires

Capacité de mettre en question

Capacité de prendre des décisions

Connaissance des politiques de l'organisation

Connaissance du business

Responsabilités

Il s'agit d'un rôle de cadre utilisateur de haut niveau. Le sponsor exécutif est le «champion RAD» qui s'est engagé en ce qui concerne la méthode et veut que le système soit mis en œuvre. Ce rôle «détient» le système et en a la responsabilité en dernier ressort. Le Sponsor exécutif doit occuper une position suffisamment élevée dans l'organisation pour pouvoir résoudre des questions relatives au business (par exemple, ouvrir des portes autrement fermées) et prendre des décisions financières. La responsabilité de ce rôle est cruciale pour assurer et permettre des progrès rapides pendant toute la durée du projet, passant outre la bureaucratie et les politiques qui entravent le développement. Ses responsabilités spécifiques sont les suivantes :

Assurer que le processus décisionnel pour la progression de questions relatives au projet est efficace et rapide ;

Réagir aux questions de progression ;

Assurer que les fonds et autres ressources soient disponibles quand nécessaires ;

Contrôler le maintien de l'adéquation du projet au business ;

Engagement et disponibilité pendant tout le cycle de développement.

Visionnaire

Compétences

Excellent communicateur

Connaissance excellente des objectifs du business

Conscience très développée des possibilités technologiques.

Responsabilités

Le visionnaire est un rôle utilisateur. C'est lui qui est habituellement responsable du démarrage d'un projet par son enthousiasme et son engagement envers les idées et les objectifs du business. Le visionnaire devrait être impliqué du début à la fin du processus de conception et de livraison pour assurer que les objectifs originels sont atteints. Si au cours du projet des questions se posent requérant la considération d'un niveau supérieur de l'encadrement, le visionnaire soit prendra la décision, soit fournira l'encadrement supérieur du point de vue du business. Ses responsabilités spécifiques sont :

Promouvoir la traduction de la vision en pratique de travail ;

Avoir une vision plus large de bout en bout du processus du business ;

Contribuer aux sessions relatives aux besoins clefs ;

Contribuer aux sessions importantes de conception ;

Contribuer aux sessions importantes de révision ;

Résoudre les conflits dans l'ensemble du domaine du business détenu par le visionnaire ;

Faire en sorte que les ressources utilisateurs soient disponibles au fur et à mesure qu'elles sont requises ;

Contrôler les progrès en relation à la version originelle ;

Engagement et disponibilité pendant toute la durée du cycle de développement.

Utilisateur ambassadeur

Compétences

Connaissance des secteurs pertinents du business et des politiques de l'organisation

Perception globale de haut niveau de la façon dont le système informatique devrait fonctionner pour supporter le business

Bonnes aptitudes à la communication

Capacité de disséminer les connaissances et les idées.

Responsabilités

L'utilisateur ambassadeur vient généralement du domaine du business traité et fait partie intégrante de l'équipe de développement. Le rôle fournit le point focal des décisions relatives à la conception. Il est le moteur de la conception du système car il transmet les contributions et idées des autres utilisateurs pendant les diverses sessions et travaille en étroite collaboration avec le chef d'équipe et le coordinateur technique. Le détenteur du rôle d'utilisateur ambassadeur doit faire preuve de volonté de réussite du projet, d'autorité, de responsabilité et de connaissance afin d'être capable d'assurer la construction du système qui convient au business. Ce qui n'implique pas obligatoirement qu'il occupe un poste élevé dans l'organisation mais qu'il soit investi du pouvoir et du temps nécessaire pour assumer ce rôle et participer pleinement au projet comme requis. Ses responsabilités spécifiques sont les suivantes :

Fournir les données essentielles aux sessions concernant les besoins du business et la conception ;

Fournir le détail des scénarios du business ;
Communiquer avec les autres utilisateurs et obtenir leur aval ;
Fournir des données aux sessions de prototypage ;
Réviser la documentation ;
Evaluer et accepter le logiciel livré ;
Fournir la documentation pour les utilisateurs ;
Assurer que la formation se fait de façon adéquate ;
Organiser et contrôler les tests utilisateurs.

Utilisateur conseiller

Compétences

Connaissance pratique du domaine du business.
Aptitude à communiquer les connaissances et les idées.

Responsabilités

L'Utilisateur conseiller fournit la connaissance au quotidien du travail/de la tâche à automatiser. Le chargé de ce rôle sera probablement l'une des personnes qui utiliseront le système lorsqu'il sera réalisé. Ses responsabilités spécifiques sont :

- Fournir l'information à la demande ;
- Participation au processus de prototypage et de révision par ses conseils et son assistance sur des questions pratiques importantes ;
- Approbation des conceptions et des prototypes dont l'utilisation est acceptable ;
- Contribuer aux tests business et d'ergonomie

Chef de projet

Compétences

Bon communicateur
Compétences en matière de planification et de gestion de projet
Connaissance des politiques de l'organisation
Connaissance du business
Aptitude à résoudre les problèmes – qu'ils soient techniques ou relatifs au personnel

Responsabilités

Le Chef de projet vient de la communauté utilisatrice ou de la L'informatique. Le rôle opère hors du cadre de chaque équipe de développement. Il est d'abord et avant tout responsable de la livraison du produit et également de la coordination et doit rendre compte à l'encadrement. Ses responsabilités spécifiques sont les suivantes :

- Rendre compte aux cadres supérieurs et au Comité directeur ;
- Planification et calendrier du projet ;
- Contrôler les progrès ;
- Gestion du risque ;
- Orienter et motiver les équipes ;
- Définir les objectifs des équipes ;

Présider les réunions du projet ;

Gestion de la participation utilisateur dans les équipes de développement (par exemple, assurer la disponibilité) ;

Traitement des exceptions ;

Identifier et faire appel aux rôles spécialistes lorsque requis ;

Traiter les problèmes de progression des équipes du projet.

Coordinateur technique

Compétences

Expérience des outils utilisés

Expérience des normes utilisées

Technicien supérieur ayant déjà rencontré ce type de problèmes auparavant

Bonne vision technique

Capable de communiquer efficacement la vision technique.

Responsabilités

Rendre compte au Chef de projet : le coordinateur technique se situe au-dessus des équipes de développement mais participe à chacune d'elles sur une base de temps partiel. Le rôle assure la consistance du travail des équipes et que le projet est dans l'ensemble cohérent et sain. Ce rôle peut incorporer celui d'Administrateur de base de données ou fournir l'interface avec l'administrateur de base de données pendant le processus de développement. Ce rôle est donc le « ciment » du projet tout en fournissant les conseils nécessaires aux décisions techniques et à l'innovation. Ses responsabilités spécifiques sont :

Définir l'environnement technique ;

Contrôler les procédures de gestion de configuration ;

Assurer l'adhésion aux standards de bonnes pratiques ;

Conseiller et coordonner les activités techniques de chaque équipe, en particulier en ce qui concerne les interfaces entre les équipes et l'utilisation de base de données ;

Assister aux sessions de prototypage pour conseiller sur l'application des standards ;

Assurer que les objectifs de maintenabilité sont satisfaits ;

Convenir et contrôler l'architecture logiciel ;

Identifier les possibilités de réutilisation ;

Gérer le contrôle de mise en pratique.

Chef d'équipe

Compétences

Bon communicateur

Savoir écouter

Technicien compétent

Connaissance du business

Aptitudes analytiques

Formé dans les techniques à mettre en œuvre (ex, les « ateliers facilités », le prototypage).

Responsabilités

Le chef d'équipe s'assure que l'équipe fonctionne comme un tout et satisfait aux objectifs qu'elle s'est fixée. Il est responsable de la livraison du système associé au domaine du business et du contrôle du changement ainsi que de la documentation du projet. Ses responsabilités spécifiques sont les suivantes :

Développer et maintenir la conformité à la définition du domaine d'activité du business ;

Assurer le traitement des besoins du business formulés par les utilisateurs ;

Organiser les sessions de prototypage et de révision entre les utilisateurs et les développeurs ;

Encourager la participation complète des membres de l'équipe dans le cadre des rôles et des responsabilités définis. ;

Conduire les sessions de prototypage à atteindre les objectifs fixés dans la «timeboxe » et dans le cadre des contraintes de l'ordonnancement ;

Assurer la documentation et le contrôle des changements ;

Promouvoir le bien-être et la motivation de l'équipe.

Développeur et développeur confirmé

Compétences (développeur confirmé)

Expérience des techniques de prototypage

Expérience des outils et des diverses techniques utilisées

Savoir écouter/communiquer

Bonne connaissance du domaine du business.

Compétence (développeur)

Connaissance des techniques de prototypage

Connaissance des outils et des diverses techniques utilisées

Savoir écouter/communiquer

Connaissance du business.

Responsabilités

Un développeur confirmé modélise et interprète les besoins de l'utilisateur et, en se servant du document d'Etude du Business et en y incorporant les remarques additionnelles de l'utilisateur pour les convertir en prototypes et en code livrable. Ce rôle répertorie et développe tout élément non prototypable du système. Il est aidé dans ces tâches par un développeur dans le cadre du développement des compétences DSDM. Ses responsabilités spécifiques sont les suivantes :

Création d'une documentation détaillée lorsque nécessaire ;

Travailler avec les utilisateurs pour définir les besoins du business, créer les prototypes et les programmes finalisés ;

Création des autres composants comme, par exemple, un modèle de logique de données ;

Créer la fiche technique des tests ;

Réviser et tester son propre travail et celui des autres.

Facilitateur

Compétences

Excellentes aptitudes relationnelles

Excellent communicateur

Savoir présenter

Impartialité

Connaissance des techniques d'atelier

Connaissance du business.

Responsabilités

Le facilitateur est indépendant de l'équipe du projet. Il est responsable de la gestion du processus d'atelier et le catalyseur de la préparation et de la communication. Il est responsable du contexte et non du contenu de l'atelier. A cette responsabilité sont associées les responsabilités spécifiques suivantes :

- Convenir de la portée de l'atelier avec le chef de projet ;
- Planifier l'atelier ;
- Etre familiarisé avec le domaine du business ;
- Interviewer les participants pour s'assurer qu'ils conviennent ainsi que de la réalisation de tout travail préparatoire ;
- Faciliter la tâche de l'atelier pour qu'il satisfasse à ses objectifs ;
- Faire une revue de l'atelier par rapport à ses objectifs.

Rapporteur

Compétences

Savoir écouter/communiquer

Connaissances techniques et du business

Bonne aptitude à la communication écrite.

Responsabilités

Le rapporteur est présent et rédige les actes de toutes les réunions de l'équipe, des «ateliers facilités», et sessions de prototypage, accords convenus et décisions prises. Ceux-ci sont ensuite soit relus et adoptés par l'équipe sur le moment, soit diffusés pour révision ou accord. Ses responsabilités spécifiques sont :

Enregistrer tous les points formulés qui sont pertinents pour le système dans le média convenu (les documents formatés peuvent s'avérer très utiles) ;

Aider à l'interprétation subséquente ;

Gérer la distribution de la documentation du projet.

Rôles	Visionnaire	Utilisateur ambassadeur	Utilisateur conseiller	Chef de projet	Coordinateur technique	Chef d'équipe	Dev. confirmé	Développeur	Facilitateur	Rapporteur
Compétences										
Savoir communiquer	X	X	X	X	X	X	X	X	X	X
Ecouter/interpréter	X					X	X	X	X	X
Compétences gestion				X		X				
Savoir motiver				X		X			X	
Au fait des politiques du business		X		X						
Au fait des exigences du business	X	X	X	X		X	X	X	X	X
Connaissance pratique du business	X	X	X							
Connaissance technique				X	X	X	X	X	X	X
Connaissance des outils					X		X	X		
Vision technique					X					
Compétences techniques						X	X			
Compétences analytiques						X	X		X	
Connaissance des standards					X	X	X	X		
Expert en Standards					X					
Compétences rédactionnelles										X

Schéma 15.1 : La matrice DSDM rôle/Compétences

Rôle DSDM	Sponsor exécutif	Visionnaire	Utilisateur ambassadeur	Utilisateur conseiller	Chef de projet	Coordinateur techn.	Chef d'équipe	Dev. confirmé	Dev	Facilitateur	Rapporteur
Rôle informatique conventionnel											
Analyste/Programmeur						X	X	X	X		X
Analyste business					X		X	X			X
Concepteur						X	X	X	X		X
Encadrement utilisateur	X	X									
Architecte système						X					
Expert technique						X		X	X		
Utilisateurs			X	X	X		X				X

Schéma15.2: Matrice de conversion des rôles conventionnels de l'informatique en rôles d'équipe DSDM

15.3 Rôles d'experts

Le chef de projet ou le chef d'équipe peut impliquer ou faire appel à des rôles d'experts sur une base *ad hoc* pour remplir des fonctions spécifiques dans les équipes centrales DSDM. Ces rôles ne seront pas tous requis pour tous les projets. Les rôles requis vont dépendre de la dimension et de la nature du projet. Certains de ces rôles ont pour but de faciliter le transfert de compétences à l'équipe centrale DSDM

Ces rôles doivent être intégrés correctement comme approprié à l'équipe du projet. Les informations à fournir par l'expert à l'équipe centrale doivent être planifiées officiellement, les personnes concernées doivent être identifiées et leur disponibilité vérifiée afin qu'elles soient en mesure d'assister aux réunions où leur présence est requise, telles que les «ateliers facilités», etc. D'autres rôles peuvent également être requis pour des fonctions spéciales de DSDM, par exemple, un concepteur Web pour les développements Internet. Ceux-ci sont spécifiés dans le livre blanc approprié.

Rôle expert	Responsabilité
Consultant Business	Expert dans un domaine spécifique pouvant conseiller et guider

	ou produire un document
Modélisateur Business	Fournir une expertise de modélisation dans une technique spécifique, ex : OO ou modélisation structurée.
Architecte Business	Entreprendre la planification du business, conseiller sur la structure et le flux du business, et analyser/évaluer l'impact sur le modèle business global.
Coordinateur des processus business	Promouvoir, développer et coordonner les processus nouveaux ou changés du business
Consultants Techniques	Fournir une expertise dans des domaines spécifiques tels que les réseaux, la conception de bases de données, des considérations opérationnelles, l'utilisation des outils et les évaluations techniques
Spécialiste d'adéquation du système à l'homme	Conseiller en ce qui concerne la conception de l'interface utilisateur pour un maximum de productivité, de confort, d'ergonomie, etc.
Chargé de planification capacité/ performance	Conseiller sur les besoins en capacité du système informatique final ; Conseiller pour toutes exigences de performance.
Assesseur de réutilisation	Conseiller sur les composants de la bibliothèque existante dont la réutilisation conviendrait à ce développement. En fin de projet, examiner s'il y a des composants qui puissent être intégrés à la bibliothèque réutilisable.
Expert en matière de sécurité	Conseiller en matière d'exigences de sécurité
Architecte données	Conseiller sur les standards communs de données, fournir une analyse de l'impact sur les autres systèmes par l'utilisation des données communes et évaluer le modèle de données.
Contrôleur de Configuration	Si l'organisation a identifié le contrôle de configuration comme une fonction nécessaire indépendante : rôle de bibliothécaire de configuration devrait également être inclus dans l'équipe.
Responsable de la qualité	Comme requis par le système de gestion de la qualité de l'organisation.
Intégrateur de systèmes.	Selon le volume de l'ensemble du projet, il peut être nécessaire qu'un intégrateur de systèmes prenne charge de l'assemblage final des composants du projet. Il s'agit en général de quelqu'un qui est redéployé à partir d'une des équipes DSDM ou du Coordinateur technique.
Equipiers pour la gestion du support et de la maintenance.	Si les développeurs ne sont pas ceux qui vont assumer la gestion de support, l'équipe de gestion de support doit participer au développement afin d'acquérir les connaissances qui lui permettront de fournir un soutien satisfaisant à l'utilisateur final.
Coordinateur des Opérations	Le service des opérations doit trouver le système informatique satisfaisant. La participation des personnes qui seront responsables des aspects opérationnels est donc nécessaire pendant la conception et pendant la mise en œuvre. Et ce, également pour garantir que le système sera inclus dans le Plan de sauvetage en cas de catastrophe sur le site, si pertinent
Fournisseur d'infrastructure	Fournir et installer le matériel, le système logiciel et l'environnement communications requis pour le développement et la production.
Responsable du service/bureau de support	Conseiller et négocier, si nécessaire, l'Accord sur le niveau de service pour le système informatique final. Convenir, si pertinent, de la façon dont le Bureau de support traitera le système.
Responsable des tests	Produire une stratégie de tests, assurer l'installation correcte de

	l'environnement du test et coordonner les tests et les activités relatives aux tests.
Responsable des mesures	Aider à l'estimation et au comptage des points de fonction pour le projet. Collecter et analyser les mesures lorsque le projet est réalisé pour information en retour dans le processus d'estimation.
Spécialiste de mise en conformité	Conseiller sur les aspects légaux, de protection de données, de trace de contrôle, de l'an 2000 et toute autre question de mise en conformité.

Tableau 15.3 : expert : rôles et responsabilités

16 Planification et contrôle du projet

16.1 Introduction

Il est important de comprendre que DSDM fournit un cadre global de gestion de projets. Par conséquent, les chefs de projet doivent se familiariser avec le contenu de tous les chapitres de ce manuel. Ce chapitre n'a pas pour but d'être un guide exhaustif des activités de gestion de projet, ni non plus d'être complet en lui-même. Il a un double objet, l'un est d'attirer l'attention sur les sections spécifiques du manuel qui fournissent des conseils concernant la planification, la surveillance et le contrôle et, le second est de conseiller les Chefs de projet en L'informatiques confirmés mais n'ayant qu'une expérience limitée de DSDM. Veuillez vous référer à la section 1.5 pour une bibliographie utile si DSDM est nouveau pour vous. A la fin de ce chapitre se trouve une section distincte destinée aux Chefs de projet ayant des formations et des expériences différentes. Ceux dont c'est le cas trouveront également que la plus grande partie des conseils contenus dans les chapitres cités dans la section 1.5 leur sont utiles.

L'objectif de toute gestion de projet est de livrer la production désirée, à temps et dans les limites du budget imparti, grâce à un usage judicieux des ressources disponibles. Ce qui implique un certain nombre d'activités qui peuvent être groupées, au sens large, sous les titres de planification, surveillance et contrôle. Cela met en œuvre, également, une grande diversité de compétences comprenant la gestion de personnel ainsi que la gestion de ressources techniques. La gestion d'un projet DSDM n'est donc pas dissemblable de ce point de vue de la gestion de tout autre projet. Ce sont les activités détaillées de la gestion DSDM de projet qui sont souvent très différentes de celles menées dans une gestion de projet traditionnelle.

Les projets DSDM prennent des formes très diverses allant du développement des systèmes informatiques à des conceptions de processus business. Le développement d'un système informatique peut être un système taillé sur mesure ou porter sur la mise en œuvre d'un ensemble existant. Le projet peut être réalisé dans l'entreprise ou par un fournisseur de logiciels. Le projet DSDM peut être unique ou être inclus dans un développement plus vaste lequel peut être géré par des méthodes traditionnelles ou par DSDM.

Le chef de projet peut venir de l'informatique ou de la communauté des utilisateurs du système. Il ou elle peut avoir de nombreuses expériences DSDM ou aucune, être très expérimenté(e) dans la gestion de projet ou pas du tout. Dans certaines organisations, le chef de projet ne sera nommé que lorsque les termes et conditions du projet auront été négociés et convenus. Dans d'autres organisations, la même personne peut être chargée de cette tâche de bout en bout du projet.

Etant donné la grande diversité de situations des projets, les conseils suivants devront être adaptés aux circonstances particulières.

16.2 Planification du Project

Planification initiale

Le chef de projet doit produire un plan global dans le cadre de l'étude de faisabilité. Ce plan a pour but de définir correctement les termes et conditions pour que le projet DSDM réussisse. Par la suite, un plan détaillant l'accord avec les utilisateurs relatif à la répartition du projet en « timeboxes » et en produits livrables initiaux (normalement un certain nombre de prototypes) est établi. Les plans sont traités dans le Chapitre 13, les « timeboxes » dans le Chapitre 7 et les prototypes dans les Chapitres 9 et 12.

Les étapes détaillées de la planification d'un projet DSDM varieront en fonction des critères mentionnés dans leurs grandes lignes dans l'introduction de ce chapitre. Habituellement, les étapes suivies sont :

- un minimum de compréhension des besoins pour évaluer si l'approche DSDM convient au projet. (Voir Chapitre 4 pour savoir quand recourir au DSDM et l'Annexe A pour les filtres d'adaptation)
- établir avec l'encadrement utilisateur les meilleures conditions pour le projet. L'idéal serait que l'équipe comprenne des développeurs et des utilisateurs confirmés en informatique, qu'elle soit localisée sur le site des utilisateurs et que lui soit donné le pouvoir de prendre une grande diversité de décisions concernant le développement du système. Tous ces aspects doivent faire l'objet de négociations au commencement du projet. Les rôles dans l'équipe du projet et des autres parties intéressées doivent être définis. Les chapitres suivants fournissent tous un matériel de référence en la matière : Chapitre 5 sur la culture du projet, Chapitre 6 sur la composante « business », Chapitre 14 sur les structures d'équipe et le Chapitre 15 sur les rôles DSDM.
- s'assurer que les gestionnaires utilisateurs ont convenu de libérer leur personnel pour l'équipe de développement pendant les périodes de temps nécessaires (y compris le détachement à plein temps lorsque le projet l'exige). L'encadrement utilisateur doit libérer ceux des membres du personnel qui ont l'autorité, la volonté, la responsabilité et les connaissances nécessaires pour prendre des décisions au nom de l'ensemble de la communauté. Ceci exige que le chef de projet ait une bonne relation avec le sponsor exécutif.

Il est possible que le chef de projet ne soit nommé qu'après la réalisation de ces activités initiales. Par ailleurs, il n'est pas toujours possible de mettre en place un environnement entièrement satisfaisant. Dans ces situations, le chef de projet devra évaluer les risques avec une attention particulière et entamer les actions appropriées pour réduire les risques. Le Chapitre 18 sur l'évaluation du risque fournit un guide utile.

La préparation des plans DSDM

Il est particulièrement important, au moment où les plans commencent à être produits, que le chef de projet définisse avec les utilisateurs ce qui constituera un système conforme aux objectifs du business. Une des critiques fréquemment adressées au «RAD » est que le chef de projet prend pour excuse l'expression «conformité avec les objectifs du business » pour produire rapidement des systèmes qui contiennent des bogues, dont la maintenance est difficile ou qui sont incomplets. Ces critiques ne peuvent pas être adressées aux projets DSDM bien gérés pour lesquels le chef a obtenu une bonne entente sur ce qui constitue la conformité aux objectifs du business.

Si le business a besoin d'un système complet avant qu'une partie puisse être utilisée, le processus DSDM sera alors géré en fonction de ce besoin. Si le business requiert un système ne comportant aucune erreur, le processus DSDM produira ce type de système. Si le business exige un système pour lequel il y a peu de maintenance et qui facilite des développements ultérieurs alors le DSDM convient parfaitement ! Le sens donné à l'expression de «conformité aux objectifs du business» constituera une partie importante de l'aspect qualité des plans produits par le chef de projet.

Une bonne compréhension de ce que les utilisateurs considèrent être conforme aux objectifs de l'entreprise est nécessaire pour être en mesure de décider comment les activités clefs du développement des plans doivent être entreprises. Celles-ci comprennent :

- définir quels besoins doivent être traités en priorité ;
- développer un plan de prototypage valide ;
- établir et convenir du calendrier « timeboxe » et du contenu initial.

Mener à bien ces activités va dépendre d'une bonne compréhension du cadre de développement du processus DSDM (voir Chapitres 3 et 11) et des outils et techniques appropriées pour DSDM (voir Chapitres 20 et 24).

Plans du projet

Plutôt que d'établir dès le début des plans détaillés pour l'ensemble du projet, les plans DSDM évoluent de façon plus ou moins détaillée à mesure que le projet avance, que les besoins sont progressivement définis et que les enseignements sont tirés. Toutefois, une organisation qui n'aurait pas de standards ou de besoins pour la production de plans d'ensemble pour ses projets serait bien étrange.

Dans certaines organisations (particulièrement celles qui développent des systèmes informatiques similaires sur des plates-formes similaires pour des clients similaires), il peut être possible de définir un plan qui s'applique à tous les projets. Pour d'autres organisations, au contraire, chaque projet est suffisamment différent des autres pour qu'un plan spécifique par projet soit nécessaire, dans la mesure où les projets qui se suivent ont peu de choses en commun. Cependant, dans la majorité des cas, il s'agit d'une situation intermédiaire, où une partie substantielle des aspects du plan liés à la qualité peut être réalisée en se référant à une documentation existante. Cette particularité s'applique aux procédures et aux standards à utiliser.

Tous les produits générés et toutes les activités entreprises dans le cadre du projet doivent figurer dans le plan. Ceci comprend les produits livrables (prototypes, documentation, etc.), mais également, soit directement dans le document ou en référence à d'autres documents :

- la procédure de lancement du projet ;
- le contrôle de configuration ;
- le détail de la structure du produit ;
- les descriptions des produits ;
- la gestion du risque ;
- les consignes de travail.

La rigueur et le formalisme appliqués à la production des plans seront différents pour chaque situation. Dans certains cas, un compte rendu de réunion ou une note brève pourra suffire, alors que dans les situations contractuelles, le client peut souhaiter donner son approbation à un plan de qualité distinct au début du projet.

Il est important que l'effort investi dans la planification soit approprié et en rapport avec la dimension du projet. Il faut également signaler qu'aucune raison fondamentale ne justifie la rédaction complète du plan du projet au début de ce projet. Ce qui est important, c'est que les parties requises du projet soient prêtes quand on aura besoin d'elles. Ce qui n'implique absolument pas que la qualité puisse être un objectif mouvant, susceptible de changer à mesure que le projet progresse.

Déplacement de l'accent par rapport à la planification traditionnelle

Pour un chef de projet immergé dans les méthodes traditionnelles, la phase de planification et les activités initiales impliquées par un projet DSDM peuvent s'avérer particulièrement exigeantes.

Un chef de projet traditionnel centrera normalement ses activités sur l'obtention d'un contrat détaillé avec le client, contrat portant sur la totalité du système à livrer accompagné des coûts et de l'échelle des temps. Dans un projet DSDM, le chef de projet se concentre sur la mise en place d'une relation de collaboration avec les clients, les incluant pleinement dans le tissu de l'équipe.

Dans le cadre d'un projet traditionnel, la préoccupation majeure du chef de projet est de comprendre les besoins en détail et complètement afin d'obtenir le niveau correct de ressources et estimer les délais de réalisation. Dans un projet DSDM, le chef se préoccupe de convenir avec les utilisateurs du processus par lequel les besoins du business seront satisfaits.

De plus, le plan d'un projet traditionnel est extrêmement détaillé et, en théorie, exécuté avec le moins de changements possibles. Dans un projet DSDM, les plans initiaux sont conçus pour

comporter suffisamment de détails pour établir les principaux paramètres du projet et dans la ferme attente que les clients changeront le plan au cours du projet.

La transition à partir de la planification traditionnelle

Il serait sage pour un chef de projet confirmé mais sans expérience personnelle DSDM de faire en sorte que l'équipe, soit comporte des membres ayant une expérience DSDM, soit ait un accès libre et facile à cette expérience. De plus, l'acceptabilité de DSDM et une expérience DSDM précédente au sein de l'organisation sont des facteurs importants. Si « RAD » et DSDM sont bien connus et compris, les risques seront faibles. Par contre, si l'organisation repose exclusivement sur des méthodes traditionnelles et si son approche vers de nouvelles méthodes est conservatrice, alors les risques seront proportionnellement plus élevés. Dans ce cas, le Chapitre 10 sur l'introduction de DSDM dans l'organisation fournit des conseils importants.

L'expérience pratique de DSDM a montré qu'il existe un processus intermédiaire situé entre l'application complète de DSDM et celle des méthodes traditionnelles, processus qui convient aux chefs de projets confirmés en L'informatique entreprenant leur premier projet DSDM. De nombreuses organisations vont établir le détail des besoins comme si elles entreprenaient un projet traditionnel ; ceci leur donne confiance en ce qui va être livré. Ils conviennent ensuite avec les clients d'une approche DSDM au projet se servant de prototypage et de «timeboxes ». Si cette approche est adoptée, il devient vital que le Chef de projet réalise que le plan détaillé initial n'est pas immuable comme c'est le cas dans le développement traditionnel. A mesure que chaque «timeboxe » est réalisée, il faut absolument obtenir des informations en retour des utilisateurs sur les progrès à ce point pour ajuster, en fonction de ces informations, les étapes suivantes du plan. Si chaque « timeboxe » est simplement réalisée et la suivante exécutée sans aucun ajustement on obtient simplement qu'un développement traditionnel de systèmes livrés par phases. Cette approche n'est pas recommandée mais l'expérience de membres de DSDM montre qu'elle est viable. Toutefois, le processus de retour de l'information et d'ajustement est un composant nécessaire pour qu'un projet soit conforme à DSDM.

Surveillance et contrôle du projet

Activités

Dans un projet géré traditionnellement, le chef de projet dispose d'un plan détaillé qui lui permet de surveiller et de contrôler les activités. Dans un projet DSDM, il y a normalement plus d'activités qui se déroulent parallèlement. Par conséquent, le chef de projet DSDM a plusieurs responsabilités spécifiques pour faire en sorte que le projet soit contrôlé dans le Modèle fonctionnel et la conception et le développement itératifs. Elles comprennent :

- maintenir une cadence rapide de progrès ;
- obtenir l'accord de hiérarchisation des besoins ;
- obtenir un accord sur le contenu détaillé des «timeboxes » suivantes ;
- gérer les relations avec le client ;
- maintenir la culture et la motivation de l'équipe ;
- communiquer les progrès ;
- vérifier que les enseignements ont été tirés et les plans mis à jour.

Cette liste n'est pas exhaustive mais les points mentionnés sont développés ci-dessous. D'autres points importants sont couverts dans divers chapitres, par exemple, s'assurer que les besoins non fonctionnels sont satisfaits et que les tests sont intégrés dans l'ensemble du cycle de vie.

Avec une petite équipe DSDM qui travaille avec des objectifs clairs dans des «timeboxes» de courte durée, les systèmes de contrôle complexes et bureaucratiques sont superflus. Il est néanmoins important de capter l'information relative au temps passé sur les produits livrables afin d'alléger la tâche de réalisation d'estimations supplémentaires.

Maintenir une cadence rapide de progrès

La rapidité du développement peut poser problème aux chefs venant d'une formation traditionnelle de développement de L'informatique. Si des problèmes surgissent pendant une «timebox», le Chef traditionnel est souvent tenté de renégocier la date finale, dans la mesure où c'est ce qu'il ferait en temps normal. Dans un projet DSDM, la date butoir fixée pour une «timebox» est sacro-sainte et ce habituellement parce qu'elle est dictée par les besoins du business. Il en découle que la pratique reconnue est de renégocier le contenu de la «timebox» plutôt que sa durée. Ceci est grandement facilité si les produits livrables ont été classés selon les règles MoSCow. (voir le Chapitre 7).

Obtenir un accord sur les priorités et le contenu détaillé des «timeboxes»

Il incombe au chef de projet, dès qu'une «timebox» est réalisée, de s'assurer que ce qui va être livré par les «timeboxes» suivantes est clairement compris et que les besoins pertinents sont très précisément établis. Il est très vraisemblable que les utilisateurs changeront d'avis en ce qui concerne les priorités et les besoins. Le chef de projet doit être ouvert à de tels changements tout en vérifiant que les utilisateurs en comprennent parfaitement les conséquences.

Gestion des relations avec les clients

Dans un projet DSDM, les utilisateurs sont normalement membres à part entière de l'équipe du projet ou du moins ont des relations de travail étroites avec cette équipe. Cette relation d'étroite collaboration est habituellement un atout pour accélérer le développement et renforcer l'implication dans une l'utilisation réussie du système.

Cependant, si les rôles et les relations n'ont pas été clairement établis dès le début du projet, il peut arriver que parfois l'inclusion d'utilisateurs dans une équipe de développement puisse être une source de stress pour le chef de projet. Il est également important de traiter des différences administratives que chaque groupe représenté dans l'équipe apporte au projet. L'enregistrement et l'évaluation du coût du temps de l'utilisateur, par exemple, peut devenir une cause de friction si les utilisateurs ne le font pas généralement mais que le service informatique de l'information le fait.

Les activités spécifiques de renforcement de la cohésion de l'équipe doivent être incorporées très tôt dans le projet afin que toute tendance à adopter une attitude «eux d'un côté, nous de l'autre» soit rapidement brisée.

Préserver la culture et le moral de l'équipe

Le chef de projet assume également la prime responsabilité d'entretenir et de soutenir la motivation et le moral de l'équipe. Ce qui n'est pas un problème normalement, dans la mesure où le moral de l'équipe DSDM est ravivé constamment par le rythme rapide des progrès et les réactions très positives des utilisateurs quant à la qualité et à la fréquence des produits livrables. Pour éviter une baisse de moral s'il se produisait des contretemps, il est important de surveiller la santé de l'équipe et d'incorporer des récompenses fréquentes au travail. Chaque épreuve surmontée avec succès peut être fêtée selon la culture de l'organisation, par exemple par une session pizza ou gâteaux ou bien en sortant l'équipe du lieu de travail pour une heure.

Les chefs de projets doivent également être attentifs à leur propre épuisement tout autant qu'à celui des membres de leur équipe. Il aurait été rapporté que les chefs de projets DSDM vieillissent prématurément !

Les chefs de projets qui ont une forte propension à exercer un contrôle personnel risquent également de rencontrer des difficultés dans l'exécution des projets DSDM. Pour que la rapidité de la cadence de développement et de livraison soit maintenue, il faut que l'équipe

DSDM ait le pouvoir de prendre des décisions. Ce qui tend à conduire à des méthodes démocratiques impliquant une charge importante de communication pour le Chef et les membres de l'équipe. Là encore, le processus est facilité si le chef d'équipe a convenu clairement des rôles et responsabilités de chaque membre de l'équipe et des autres parties concernées. Le rôle de chef de projet doit être plutôt celui d'un arbitre que celui de l'autorité décisionnelle suprême.

Communiquer les progrès

On ne soulignera jamais assez que la responsabilité pour l'ensemble des communications incombe au chef de projet. La communication entre les membres de l'équipe sera vraisemblablement très efficace particulièrement s'ils sont installés sur le site de leurs utilisateurs. Dans ces circonstances, le chef de projet se préoccupera surtout d'informer régulièrement l'encadrement et les parties intéressées de l'évolution du projet. Le rythme des progrès est tel qu'il est facile de ne plus être à jour en ce qui concerne l'évolution du projet. Le chef de projet n'a pas le temps d'attendre que des décisions soient prises par les comités directeurs et autres, il est donc particulièrement important que ceux qui tiennent les cordons de la bourse soient informés en permanence.

Vérifier que les enseignements sont tirés et mis à profit et que les plans sont mis à jour

Un chef de projet pour qui DSDM est une nouvelle expérience doit également s'assurer qu'il ou elle comprend pleinement l'importance du retour d'informations provenant des utilisateurs concernant les produits livrables et les ajustements des plans du projet qui en résultent. Dès qu'une «timeboxe» est réalisée, il faut convenir d'une réunion ou d'une présentation formelle des produits livrables démontrant leur qualité aux parties concernées. Ces réunions permettent également de collecter des informations en retour, de confirmer les attentes des utilisateurs et de convenir des objectifs pour l'itération suivante. Cette activité est cruciale pour que les produits livrés finaux conviennent véritablement aux objectifs du business.

16.4 Gestion des projets DSDM : situations diverses

La plus grande partie de ce chapitre a été consacrée aux chefs de projets venant de l'informatique ayant une grande expérience des méthodes traditionnelles. Cette section est destinée aux chefs de projets venant d'horizons divers.

Un chef de projet nouvellement promu ayant une grande expérience DSDM en tant que programmeur ou analyste doit s'appliquer à développer des compétences gestionnaires plus générales plutôt que des aspects spécifiques à DSDM. Ceci portera habituellement sur la gestion des relations interpersonnelles et les compétences liées à la connaissance des politiques de l'organisation. Il peut être utile qu'un chef de projet plus expérimenté serve de modèle ou de guide pour le rôle et qu'une relation de soutien se développe avec cette personne.

Un chef de projet nouvellement promu sans expérience DSDM ne devrait envisager d'adopter l'approche DSDM que si dans l'environnement où il se trouve, il puisse bénéficier d'un bon niveau de support et d'une grande expérience DSDM qui puisse nourrir son approche. Si ce n'est pas le cas, un soutien DSDM extérieur doit être intégré.

Un chef de projet issu de la communauté des utilisateurs et qui s'apprête à gérer son premier projet DSDM doit s'assurer que le personnel de l'informatique recruté pour le projet a l'expérience nécessaire. Ces personnels doivent avoir l'expérience DSDM et disposer pour leur travail d'outils de l'informatique extrêmement performants.

Les conseils donnés dans ce chapitre s'appliquent en grande partie à toute personne gérant un projet qui sera réalisé par une organisation extérieure telle qu'une entreprise de logiciels. Ces organisations ont habituellement des chefs de projets confirmés en DSDM. Il est néanmoins particulièrement important que les rôles et les responsabilités concernés soient très clairement définis. Ceux-ci vont dépendre de la nature du contrat conclu. Bien qu'il soit plus facile de gérer des projets DSDM sur une base de temps et de matériel, la norme demeure d'établir ces

contrats sur une base de coûts fixés. Quel que soit le cas, il est important que les deux parties développent une relation de collaboration. Les membres du Consortium DSDM ont une grande capacité d'assistance et d'expérience pour traiter ces types de relations et de contrats.

Enfin, en ce qui concerne la gestion de projet, le projet DSDM peut être inclus dans un projet plus vaste géré par d'autres méthodes plus traditionnelles. Ce qui peut entraîner des risques spéciaux dans la mesure où le projet DSDM peut dépendre d'une livraison venant d'une autre partie du «grand» projet. Ce produit peut être en retard ou ne pas correspondre aux besoins réels. Il est donc important d'avoir un plan d'urgence pour faire face aux problèmes qui pourraient provenir du reste du projet. Il est également important que les communications soient ouvertes, franches, et confiantes. L'autre projet peut être méfiant de toute dépendance qu'il pourrait avoir des livrables du projet DSDM. La qualité, par exemple, peut être une source d'inquiétude pour cet autre projet. Le rôle du Chef de projet sera donc, en ce cas, de dissiper ces craintes. S'il s'agit de «grands» projets, on pourrait envisager de détacher du personnel du projet DSDM, après le développement d'un incrément, dans le reste du projet et réciproquement. Il faut néanmoins garder présent à l'esprit que la stabilité de l'équipe doit être préservée et qu'elle doit être, dans la mesure du possible, dégagée d'autres responsabilités. Cependant, l'interaction permettra d'améliorer les communications et de mieux comprendre les méthodes des uns et des autres. La même chose peut être réalisée pour des projets plus restreints en établissant des canaux de communications efficaces ou en installant les équipes sur le même site.

17 Mesures et estimations des projets DSDM

17.1 Mesurer

Pourquoi mesurer ?

Les mesures sont destinées à examiner et à souligner certains aspects d'un projet. Mesurer est nécessaire pour :

- établir un profil de prédiction de développement futur ;
- fournir la preuve que le processus est bon et qu'il fonctionne ;
- examiner le processus lui-même afin de faire ressortir et quantifier les problèmes

La seule façon de bien estimer une tâche est de se référer à des tâches similaires antérieures. Les enregistrements de mesures de projets précédents sont essentiels pour produire une bonne estimation du projet actuel.

L'analyse des mesures enregistrées pendant un projet fournit la base du suivi des progrès et peut mettre en lumière des points exigeant l'attention de l'encadrement.

Les mesures peuvent fournir les munitions nécessaires pour convaincre l'encadrement des bénéfices tangibles pour l'organisation de l'introduction de DSDM.

Quelles mesures ?

Dès le début du projet, il faut décider des mesures à collecter. Tout d'abord, il est important d'établir la finalité de toute mesure enregistrée. Des organisations et des projets différents auront des objectifs eux aussi différents ; c'est pourquoi il n'est pas possible de prescrire une série standard de mesures à appliquer à tous les projets.

Toutefois, la plupart des organisations seront intéressées par l'estimation de leur(s) projet(s) suivant(s), la planification du contenu des «timeboxes», la réalisation de produits de qualité, le suivi des progrès d'un projet et la satisfaction de leurs objectifs. Par conséquent, les mesures de base de productivité, de livraison des produits et de qualité seront normalement requises.

Deux sortes de mesures peuvent être prises, elles sont classées ici en mesures « matérielles » («hard ») et «de pondération» («soft »). Les mesures «matérielles » sont réalisées par le comptage d'attributs empiriques du projet ou logiciel (par exemple, le nombre d'écrans dans le système). Les mesures «de pondération » sont réalisées à partir de questionnaires destinés à concrétiser les avis et les perceptions des personnes concernées par le projet ou le processus.

Mesure de volume

Les mesures incluront pratiquement toujours le concept de «volume » du logiciel ou de «volume » de la tâche, puisque les mesures les plus utiles sont liées au volume. Le volume du logiciel n'est pas quelque chose qui s'exprime facilement en termes de dimensions physiques. Par contre, de nombreuses mesures peuvent être employées pour donner une idée du volume, par exemple :

- le nombre de lignes exécutables de code source ;
- le nombre et la complexité des fonctions business ;
- le nombre et la complexité des entrées et sorties ;
- le nombre d'entités dans la structure de données ou d'objets dans l'article module ;
- le nombre de points de fonction.

Lorsque les mesures de volumes sont destinées à l'estimation et à la planification, les points de fonction, basés sur les fonctions du business et les exigences en données d'une application,

sont maintenant généralement utilisés comme standard. Ceci exige, toutefois, que des praticiens formés et confirmés soient disponibles.

Mesurer l'effort

La plupart des projets mesurent l'effort engagé dans les activités du projet. Il devrait être enregistré au niveau «moléculaire» le plus bas, puis totalisés pour couvrir des activités plus vastes. Contrairement aux projets traditionnels qui enregistrent les heures ouvrées pour chaque tâche figurant dans le plan du projet, les plans de projet DSDM sont basés sur les produits et les "timeboxes" plutôt que sur les tâches au plus bas niveau.

Il est donc utile de collecter les informations relatives à l'effort engagé dans chaque cycle de prototypage. Chaque prototype doit pouvoir être classé dans l'une des quatre catégories ou comme une de leur combinaison ou sous-ensemble.

Lorsque l'effort est mesuré, il est important de s'assurer que tous les efforts pertinents engagés sont enregistrés, les heures supplémentaires incluses mais à l'exclusion du temps non projet. De plus, les projets DSDM demandent plus d'effort de la part de l'utilisateur que les projets conventionnels, par conséquent, un enregistrement et l'analyse subséquente du succès de projet montreront si l'effort fourni était suffisant. Il est évident que la direction voudra une estimation du temps utilisateur requis pour les projets futurs.

L'effort étant enregistré au plus bas niveau «moléculaire», il est possible d'en faire un ensemble. Il est recommandé que l'effort soit enregistré par cycle de prototypage et totalisé :

- par prototype
- par phase DSDM ;
 - par incrément ;
 - par projet.

Il est important d'examiner le pourcentage d'effort engagé dans chacune des phases DSDM.

Mesurer les défauts

Les défauts des projets devraient être enregistrés scrupuleusement. Pour pouvoir effectuer une analyse constructive, ils devraient être classés selon leur degré de gravité et leur type. Une classification selon trois niveaux de gravité est vraisemblablement adéquate pour distinguer : 1) ceux qui entraînent l'échec de l'application, 2) ceux qui font que l'application ne satisfait pas à un ou deux de ses objectifs business, enfin 3) ceux qui sont relativement triviaux et cosmétiques. Les types de défauts suggérés sont :

- fonctionnel
- données ;
- interface interne (ex : utilisation inconsistante d'un paramètre);
- interface externe (vers d'autres systèmes, manuels ou automatisés) ;
- non fonctionnel (ex : performance ou sécurité).

Les défauts doivent être enregistrés accompagnés de l'information indiquant le moment du cycle de vie où ils ont été découverts et leur cause probable. Un projet DSDM devrait découvrir les plus importants problèmes liés au business pendant les tests de prototypes dans le modèle fonctionnel itératif. Si de graves problèmes concernant le business sont découverts après cette phase, cela peut indiquer que le processus de prototypage est défectueux.

Les indicateurs comparatifs généraux de qualité de l'application finale livrée sont basés sur les défauts enregistrés, normalement après la mise en route de l'application. Les indicateurs typiques sont :

- moyenne des temps de bon fonctionnement ;

le nombre de défauts de gravité 1 et 2 rapportés pour 1000 points de fonction pendant le premier mois (ou toute autre période) de traitement réel.

effort requis en maintenance corrective.

Le processus DSDM est spécialement conçu pour fournir des applications bien adaptées aux besoins du business et de grande qualité. Si le processus ne donne pas satisfaction, cela vient probablement d'une application défectueuse des procédures de gestion de la qualité.

Mesurer le succès

Le succès d'un projet est déterminé selon la réalisation ou non des objectifs fixés au départ. Il est donc nécessaire de décrire les objets dès son amorce en termes précis et mesurables. La méthode DSDM met l'accent sur la satisfaction de tout ce qui est «must have» (indispensable) dans un cadre de temps fixe ; par conséquent toute mesure relative au succès inclura tous ces éléments.

Ce concept s'applique également aux «timeboxes» individuelles, lorsque les objectifs de chaque «timeboxe» ont été établis et hiérarchisés à son début. Ces objectifs devraient également être exprimés en termes mesurables pour que l'examen en fin de «timeboxe» puisse en évaluer le résultat facilement.

Les projets DSDM s'efforcent de livrer les «must have» (indispensable) dans de courtes périodes de temps prédéfinies en ne réalisant pas les besoins de basse priorité. Si les «could have» (facultatif) et les «should have» (souhaitable) sont également satisfaits cela implique que les estimations ne sont probablement pas assez serrées et devraient être resserrées dans les projets futurs.

DSDM a également pour objectif de parvenir au rachat et à la détention de l'application livrée par l'utilisateur. Les mesures «de pondération» sont utiles pour savoir jusqu'à quel point l'application livrée est acceptable par la communauté des utilisateurs.

L'utilisation des mesures de productivité

La productivité est probablement la mesure de base pour l'estimation. C'est également celle qui aura la plus grande valeur pour «vendre» les avantages de DSDM à l'encadrement. Ceci requiert les mesures de volume et d'effort provenant de projets précédents. En général, la productivité peut être exprimée en termes du nombre d'heures ouvrées requises par point de fonction.

Cependant, la productivité n'est pas constante. Les grands projets sont de façon significative moins productifs que ceux de taille plus modeste. Plus il y a de personnes impliquées dans un projet, plus les frais généraux augmentent en raison des questions de communication. Le recours à des mesures de productivité exige que des projets de taille similaire soient comparés. Selon Symons (voir la bibliographie), les projets à 1000 points de fonction sont environ moitié moins productifs que les projets à 300 points de fonction.

Trop comprimer le laps de l'échelle des temps peut également affecter négativement la productivité dans la mesure où davantage de personnes seront impliquées dans le projet et par conséquent les surcoûts de communications suivront.

De plus, la productivité varie considérablement en fonction de nombreux autres facteurs dont les principaux sont :

le type et la complexité de l'application ;

les techniques et les outils utilisés ;

l'expérience et les compétences des membres de l'équipe ;

le degré d'enthousiasme et d'implication des utilisateurs dans le projet ;

le niveau de qualité (ou autre attribut non fonctionnel) requis du système livré ;

la taille de la base utilisateur et par conséquent du nombre d'utilisateurs différents qui devront participer au projet ;

le degré de respect des critères propices et des facteurs essentiels de succès de DSDM.

Il est important d'enregistrer ces caractéristiques du projet avec toutes les mesures standard collectées. Elles contribuent à expliquer les anomalies dans l'analyse des mesures venant de différents projets ainsi qu'à ajuster les estimations pour le projet courant.

La productivité peut être exprimée pour l'ensemble du projet et également fragmenté au niveau d'unités minimales «moléculaires» pour lesquelles des mesures d'effort sont disponibles. Par conséquent, il peut être exprimé pour chaque étape du cycle de vie DSDM, chaque «timebox», chaque cycle de prototypage, pour les activités de tests de conception et de développement, pour chaque cycle de mise en œuvre, etc., tant que des mesures concernant toutes ces étapes ont été collectées pour des projets antérieurs.

17.2 Niveaux de ressource

L'équilibre de l'effort pour l'ensemble du cycle de vie diffère dans DSDM des méthodes traditionnelles car :

l'utilisation du prototypage incrémentiel va faire entrer non seulement l'effort de conception dans les activités d'analyses mais également la mise en œuvre et l'effort test dans la conception ;

les activités habituellement concentrées en fin de développement telles que les tests d'acceptation constituent ici des blocs minimaux d'activités puisqu'ils sont répartis judicieusement dans l'ensemble du cycle de vie et normalement complétés par une étape composite de tests système et acceptation avant la mise en œuvre.

La courbe des niveaux de ressource pour l'ensemble du projet sera donc beaucoup plus plate qu'avec les approches traditionnelles. La courbe classique du niveau de ressource ignore la participation peut-être sporadique mais néanmoins importante des utilisateurs aux points clés du cycle de vie. Avec DSDM, la participation de l'utilisateur est relativement uniforme dans l'ensemble des phases de développement. Le nombre des développeurs dans l'équipe peut progresser après L'étude de faisabilité et L'étude du business, mais à partir de ce point il devrait demeurer raisonnablement statique.

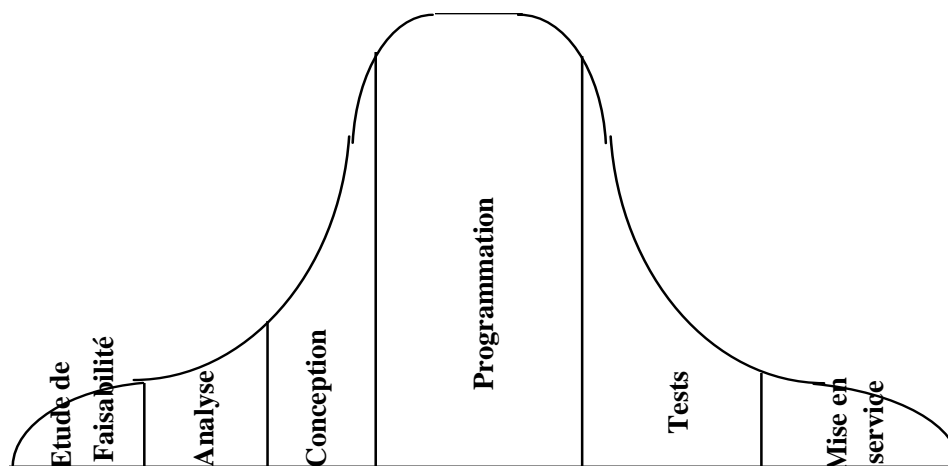


Schéma 17.1 : Chargement de ressource possible pendant un projet traditionnel

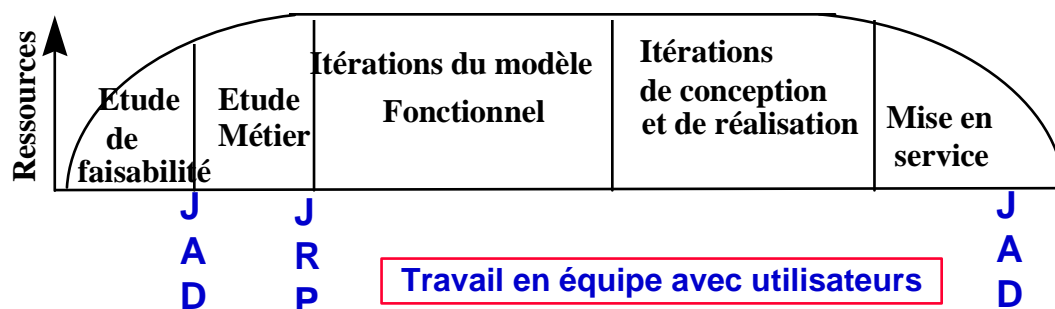


Schéma 17.2 : Chargement de ressource pendant un projet DSDM

Il est impératif d'inclure les niveaux de ressource utilisateur dans toutes les estimations DSDM. Ils devraient être clairement identifiables afin de pouvoir contrôler et enregistrer la participation réelle de l'utilisateur et la comparer aux estimations. Sans le niveau attendu et approprié de participation utilisateur, la poursuite du projet en utilisant DSDM devrait être reconsidérée très attentivement.

17.3 Estimation : lignes directrices

Les principes de l'estimation

Le principe fondamental qui privilégie la livraison fréquente des produits plutôt que les activités inclut deux principes sous-jacents. Les principes d'estimation affectent la façon dont l'estimation est réalisée dans les projets DSDM, à savoir :

Les estimations doivent être serrées dès le début avec des livraisons fréquentes ;

Les estimations basées sur les fonctions globales du business sont les plus proches du processus de développement DSDM.

Le premier principe d'estimation implique qu'il est inacceptable dans DSDM que les activités dépassent le temps qui leur est imparti et d'avoir de longues échelles des temps pour convenir de la qualité des produits. L'objectif est de livrer le système central à la date stipulée. En maintenant les estimations d'effort serrées et à court terme, l'équipe de développement ne peut pas perdre de temps dans des activités qui ne soient pas consacrées à satisfaire les objectifs du business. Il exclut également toute justification visant à prendre des dispositions de contingence dans les estimations.

Le second principe d'estimation implique que le point de départ de l'estimation devrait être la fonctionnalité attendue des produits finaux plutôt que les activités mises en œuvre pour livrer ces produits. Ce qui maintient la concentration sur la livraison/production de bénéfice pour le business.

L'utilisation, dans les projets DSDM, de «timeboxes» alignées sur la livraison de composants spécifiques du business conforte ces deux principes.

Quand faire les estimations

De même que pour toutes les estimations, celles-ci sont susceptibles de mise au point à mesure que les travaux progressent. Les estimations doivent être révisées au début de chaque «timebox». Cependant, dans DSDM, les estimations doivent être fournies à deux moments du projet.

Une estimation initiale approximative de l'effort est produite lorsque la plus grande partie de L'étude de faisabilité a été réalisée. Cette estimation va permettre à la direction d'évaluer la validité de la poursuite du projet en se servant de DSDM. Elle est basée sur les processus, les données et les interfaces du business tels que développés dans le Rapport de faisabilité et sert à développer le plan global.

La deuxième estimation est produite après la réalisation de L'étude du business. A ce point, l'étendue du projet aura été décidée, les fonctionnalités nécessaires du business auront été définies et le Plan global de prototypage préparé. La taille finale de l'équipe est décidée à ce point. La deuxième estimation peut être basée sur les composants principaux qui seront vraisemblablement inclus dans le logiciel livré, ce qui associé à la taille de l'équipe, permet de définir la durée des «timeboxes» qui seront définies pour le développement des divers composants.

Dans la mesure où les projets DSDM progressent rapidement, les estimations devraient être contrôlées à intervalles réguliers. Là où apparaît une déviation significative de l'estimation, les originaux devraient être examinés avec attention. Si cette déviation est considérée raisonnable sur la base des informations qui étaient disponibles lors de la production des estimations premières, l'étendue et la complexité du projet devraient être révisées afin de vérifier que le projet n'a pas accidentellement dévié de sa trajectoire d'origine. Là où les priorités ont changé ou si de nouvelles fonctions nécessaires ont été révélées par le prototypage, des échanges doivent avoir lieu pour les inclure.

A la fin d'un incrément, le travail doit faire l'objet de mesures qui sont incorporées dans la révision du projet et la base de donnée métrique de l'organisation.

Qui fait les estimations

Avec DSDM, les estimations devraient être réalisées par l'équipe qui va effectuer le travail et ce afin de convaincre l'équipe de la nécessité d'établir des échelles des temps serrées. Si pour des raisons commerciales, les estimations sont réalisées à l'extérieur, l'équipe doit vérifier et être convaincue que ces estimations peuvent être satisfaites.

Le chef de projet et le développeur confirmé de l'équipe réaliseront vraisemblablement les estimations pendant L'étude de faisabilité et L'étude du business. Pendant les phases de prototypage, les développeurs auront besoin d'estimer le volume de fonctionnalité qu'ils peuvent produire dans une «timebox» donnée et réévaluer les estimations à mesure qu'ils progressent.

Techniques d'estimation

Les techniques d'estimation peuvent être divisées, pour faire simple, en deux catégories : du haut vers le bas, basée sur les besoins du système ; et du bas vers le haut, basée sur la connaissance de la conception et de la taille des composants. L'approche du haut vers le bas, plus familière, réalise l'estimation par analogie, là où le projet envisagé peut être comparé à des projets similaires déjà réalisés. L'approche du bas vers le haut implique de compter les programmes, les écrans, les rapports et toute information liée à la mise en œuvre indiquée par une conception, et d'estimer l'effort pour chacun de ces éléments.

« Function Point Analysis (FPA) » (l'analyse de points de fonction) est une mesure technique utile pour estimer les projets DSDM dans la mesure où les points de fonction peuvent être obtenus très tôt dans le cycle de vie. Ils sont basés sur les besoins du business et sont indépendants de la technologie utilisée pour la mise en œuvre du système. FPA est, en fait, une technique permettant de quantifier le volume «du point de vue utilisateur» de la fonctionnalité fournie par le système.

Afin de compter les points de fonction correctement il faut comprendre dans le détail les besoins du business pour l'application. Les projets DSDM ne génèrent pas de spécifications détaillées des besoins pendant le processus de développement. Par conséquent, les points de fonction ne peuvent pas être comptés avant le modèle fonctionnel itératif. D'où, pour utiliser les points de fonction lors de l'étape de L'étude de faisabilité et L'étude du business, il est toujours nécessaire d'estimer combien de points de fonction seront inclus dans l'application finale.

Les praticiens expérimentés en points de fonction pourront faire une estimation avec une marge de 20% du compte éventuel de la Définition du domaine du business. Une estimation ayant ce degré d'exactitude est normalement suffisante pour servir de base au Plan global, étant donné que les projets DSDM fonctionnent avec des «timebox» dont la durée est fixe et qu'ils visent à éliminer les fonctions de basse priorité.

Les points de fonction peuvent également contribuer à vérifier la répartition d'un projet en «blocs» de taille convenable. Ce qui peut se faire à différents niveaux :

la répartition en blocs d'une taille convenable pour qu'une seule équipe de projet puisse les traiter ;

la répartition en blocs d'une taille convenable pour une seule «timebox».

Dans la mesure où une équipe de projet DSDM ne comportera pas normalement plus de 6 personnes et devra livrer un système ou un incrément en, au maximum, six mois, la taille de l'application qui peut être traitée par une seule équipe a donc une limite supérieure définitive. Avec ces contraintes, une équipe devrait être capable de traiter un volume maximum de 1000 points de fonction. Ce qui fait que les «grands» projets doivent être répartis en blocs de 1000 points de fonction ou moins, et traités comme un programme de développements parallèles ou séquentiels.

De fait, les points de fonction sont une macro mesure à appliquer des projets entiers et non à des fonctions individuelles. Cependant, le nombre total de points de fonction dans une application est obtenu par la somme des parties élémentaires. Par conséquent, les points de fonction peuvent contribuer à indiquer la taille relative de chaque partie élémentaire quand les prototypes sont répartis en «timeboxes» de durée convenable tant que des estimations très précises ne sont pas requises. La façon dont le contrôle de «timebox» opère implique que l'imprécision inhérente points de fonction à ce niveau détaillé n'est pas cruciale.

Il faut également se souvenir que les points de fonction sont une mesure des besoins fonctionnels du business. Ils ne doivent pas être utilisés pour estimer l'activité de maintenance pour les changements correctifs.

Le calcul de points de fonction, exprimé comme un compte des points de fonction, n'est que la première partie du processus d'estimation. Il est ensuite nécessaire de connaître la quantité d'effort nécessaire pour produire un point de fonction et de combien cet effort augmente l'échelle des articles de grandes tailles et complexes. Cette information provient des mesures prises au cours de projets antérieurs et éventuellement de sources dans l'industrie ou de vendeurs d'outils.

Une fois l'effort total calculé, l'étape suivante est souvent l'allocation de l'effort aux différentes phases, en se fondant sur des pourcentages typiques, par exemple, «30% de l'effort total sera consacré à la conception ». Cette approche ne convient pas dans DSDM. Le profil ressource est habituellement plat et le projet n'est pas divisé en phases d'activités différentes, mais plutôt en prototypes liés aux fonctions du business sur lesquelles les estimations de points de fonction sont basées. Il en découle que l'estimation d'un prototype ou d'une «timebox» peut être déduite des estimations pour les fonctions allouées.

L'étude du business produit également la première information de haut niveau pour commencer à réaliser les estimations de bas vers le haut, basées sur une première version de la conception d'écrans, de rapports, de programmes, etc. L'estimation est toujours très satisfaisante et la plus exacte possible lorsque plusieurs méthodes sont utilisées.

A mesure qu'un projet progresse, les estimations doivent être recalculées. Les développeurs doivent estimer le niveau de fonctionnalité totale produite. S'ils réalisent ces estimations par des méthodes de bas vers le haut, ils doivent établir un lien entre les articles qu'ils programment et les fonctions hiérarchisées du business. Le produit de la ré-estimation ne précise pas la durée de réalisation d'une quantité fixe de fonctionnalité, mais précise plutôt la fonctionnalité qui sera réalisée à la fin d'une «timebox». Si l'estimation indique que tous les «must have» (indispensables) ne peuvent pas être fournis dans le cadre des «timeboxes», l'équipe du projet doit revoir les priorités et procéder aux changements nécessaires.

17.4 Comparaisons avec les standards de l'industrie

Les mesures collectées au sein d'une organisation lui seront vraisemblablement beaucoup plus profitables que toute mesure importée de l'extérieur car elles reflètent l'éthique de cette

organisation particulière. Il est néanmoins utile de savoir ce que les autres parviennent à mesurer, comment se porte l'organisation, ce qui pourrait être possible et contribuer à «vendre» DSDM dans l'organisation. Il peut être utile de connaître les «standards» lorsque que l'on commence à collecter des mesures.

Plusieurs sources fournissent des données comparatives de mesures. Certaines compagnies se spécialisent dans la fourniture d'expertise de mesure sur une base commerciale et disposent de vastes banques de données de mesures prises lors de projets antérieurs qui peuvent être analysées en fonction des méthodes de développement, des outils, des secteurs de l'industrie, etc. Plusieurs outils d'estimation incorporant les résultats de cette analyse sont disponibles. Toutefois, ils doivent être calibrés pour satisfaire à un environnement spécifique.

Le Consortium DSDM construit, à l'heure actuelle, une base de données d'expériences issues de projets DSDM qui servira de guide et contribuera à la promotion de la méthode. L'annexe C contient un formulaire simple de collecte de mesures de base. Le Consortium vous serait reconnaissant de prendre le temps de remplir ce formulaire pour autant de projets que possible et de le présenter au bureau du Consortium. Le Consortium s'engage à garder confidentielle toute information relative à votre organisation si vous l'exigez.

18 Evaluation des risques

18.1 Introduction

Le présent chapitre comporte trois sections. La première section énonce les principes généraux de gestion des risques inhérents à tout projet de développement. Les deux dernières sections décrivent les techniques de gestion des risques spécifiques aux projets DSDM.

18.2 Généralités

La plupart des fournisseurs de systèmes informatiques s'intéressent à deux risques principaux. Le premier est la non conformité du système aux exigences commerciales et le second, le dépassement de coût ou de délai. Le concept DSDM a été conçu pour réduire ces deux risques. Les systèmes qui répondent aux exigences commerciales s'appuient sur une approche incrémentielle et itérative et une remontée d'informations permanente de la part des utilisateurs. Les dépassements de coût et de délai sont évités grâce à la mise en place de cadres de temps limités pour chaque phase du projet.

Certains risques inhérents aux projets de développement traditionnels peuvent également survenir dans les projets DSDM. Par exemple, le recours à des technologies de pointe permet d'accroître considérablement les capacités et les performances mais leur instabilité génère des risques supplémentaires. Par sa nature, l'approche DSDM permet de limiter les risques inhérents à l'utilisation de nouvelles technologies. C'est pourquoi il est essentiel d'élaborer des prototypes pour tester les performance et la stabilité du système. L'approche DSDM incrémentielle permet d'évaluer à petite échelle les nouvelles technologies avant de les déployer totalement et de les appliquer à l'ensemble du système.

Certains prétendent que les approches de développement rapide présentent des risques spécifiques non présents dans les méthodes traditionnelles, tels que la non maintenabilité du code, ou encore la présence d'un grand nombre de défauts non identifiés. Ces deux risques peuvent émerger dans les projets de développement rapide mal gérés. Par contre, les projets DSDM comportent des garde-fous permettant de garantir une qualité irréprochable du produit.

Néanmoins, cela ne signifie pas que la gestion d'un projet DSDM soit une entreprise dénuée de risque. En général, les risques surviennent lorsqu'un ou plusieurs principes de base ne sont pas observés ou lorsque les activités de limitation des risques inhérents à la non conformité ne sont pas mises en oeuvre. La suite de ce chapitre est exclusivement consacrée aux risques liés aux principes de base des projets DSDM.

18.3 Risques liés au non respect des principes DSDM

La participation de l'utilisateur est insuffisante

Le premier principe DSDM est la constitution d'une équipe comprenant les développeurs ainsi que des personnels de l'entreprise, à l'endroit même où le système sera utilisé. Le non respect de ce principe peut compromettre gravement le succès du projet. Différentes approches peuvent être mises en oeuvre afin d'éviter ou de minimiser ce risque.

Un facteur clé permettant d'obtenir le juste niveau de participation des utilisateurs consiste à identifier non seulement les différentes catégories d'utilisateurs concernées mais également le nombre d'utilisateur potentiels. En effet, il est facile d'impliquer des utilisateurs à temps plein lorsque leur nombre atteint plusieurs centaines. Le problème qui se pose alors est de sélectionner un échantillon représentatif puis de faire accepter les résultats obtenus aux autres utilisateurs. Ce problème doit être résolu très tôt, de préférence avant le début du projet.

Si le système est destiné à une petite entreprise, tous les employés peuvent être impliqués mais le temps qu'ils pourront consacrer au projet sera bien évidemment limité. Il est alors recommandé de déterminer la participation des utilisateurs ambassadeurs dans leur emploi du

temps, et ce dès le début du projet. Des sessions improvisées pourront par ailleurs être ajoutées en fonction des besoins.

Les sessions moins fréquentes mais non moins essentielles, telles que les ateliers de prise de décisions doivent être définies aussi tôt que possible, afin d'avertir la direction des modifications intervenant dans son planning et celui de son personnel. Pour s'assurer de la présence des utilisateurs à la date convenue, le chef de projet doit confirmer le rendez-vous et clarifier le rôle des utilisateurs, peu de temps avant la date prévue.

Cette méthode requiert l'adhésion des utilisateurs et de la direction, qui doivent prendre une part active au projet de développement. C'est pourquoi il est conseillé d'organiser une journée de sensibilisation à l'attention des employés n'ayant jamais participé à un projet DSDM.

Si l'implication des utilisateurs n'est pas totale dès le début du projet, il est encore possible de limiter les risques. Par exemple, si l'entreprise est membre du Consortium, l'équipe de développement RAD peut entreprendre le projet avec la participation exclusive des utilisateurs principaux une demie journée par semaine. A chaque session, un accord est passé sur ce qui doit être accompli d'ici la prochaine session.

Lorsque de graves difficultés empêchent la participation des utilisateurs, les risques peuvent être limités grâce à l'utilisation d'un outil de productivité de groupe et la création d'une équipe virtuelle, communiquant par le biais du courrier électronique et des téléconférences. Néanmoins, cette solution est loin d'être idéale et ne doit être utilisée qu'en dernier recours.

Lorsque les utilisateurs finaux sont issus d'un grand nombre de services différents, il est indispensable que toutes les classes d'utilisateurs soit représentés au sein de l'équipe de développement ou qu'elles soient régulièrement consultées. Si tel n'est pas le cas, le système risque de ne pas être accepté par les catégories d'utilisateurs non représentées dans l'équipe de développement.

Un risque majeur survient lorsque le niveau de participation des utilisateurs prévu au début du projet n'est pas atteint. Dans ce cas, le problème doit être communiqué d'urgence à la hiérarchie de l'organisation et résolu à la satisfaction générale.

Le temps consacré à la prise de décisions met en danger le calendrier du projet

Un problème peut se poser lorsque la responsabilité de chaque membre de l'équipe n'a pas été clairement définie. La crainte de certains développeurs d'outrepasser leurs attributions peut en effet limiter leurs initiatives et par conséquent, ralentir le projet. Ce risque peut être enrayé de plusieurs manières :

- Le processus de prise de décisions doit être indiqué clairement dans les termes de référence du projet.
- Les attributions de chacun des membres de l'équipe doivent être clairement stipulées dans leur contrat.
- Les décisions stratégiques doivent être séparées des décisions opérationnelles. En effet, ces dernières sont prises par l'équipe de développement alors que les décisions concernant la stratégie à adopter incombent à la direction. Un processus d'approbation rapide doit être décidé avant que le travail de recherche soit entrepris.

Un autre facteur peut ralentir le projet lorsqu'une décision à prendre n'est pas du ressort de l'équipe de développement et que la personne investie du pouvoir de décision met trop longtemps à définir la stratégie à adopter. Pour éviter ce risque, le chef de projet peut soumettre à la personne responsable une série de recommandations parmi lesquelles elle pourra faire un choix. Il est souhaitable que ces recommandations soient définies en collaboration avec les développeurs de sorte qu'ils soient en mesure d'assumer la décision.

Les différentes phases d'un projet DSDM étant limitées dans le temps, les conséquences d'un retard seront beaucoup plus graves que dans un projet à long terme. Par exemple, si l'équipe doit attendre cinq jours la décision d'un cadre, ce délai représente un pourcentage nettement plus élevé dans un projet de 60 jours que dans un projet de 12 mois. Les conséquences seront

d'autant plus graves que le calendrier ne sera pas modifié dans un projet DSDM. Ce seront en effet les caractéristiques fonctionnelles du produit qui en pâtiront.

Le chef de projet doit faire preuve de fermeté et refuser de débiter le projet tant que les facteurs extérieurs à l'équipe de développement ne sont pas définis. Le choix des ressources, de la plate-forme d'implémentation et des principaux composants doit être arrêté.

Les membres de l'équipe se concentrent sur les activités plutôt que sur le produit

Les risques se multiplient lorsque la durée des cadres temporels est décidée arbitrairement ou lorsque les développeurs sont plus préoccupés par leurs activités que par la livraison d'un produit fonctionnel.

Le meilleur moyen de réduire ces risques est de sélectionner soigneusement les membres de l'équipe, d'instaurer un système de primes liées au mérite et aux résultats, et de définir des cadres temporels de courte durée. Les professionnels du développement rapide ont tendance à se passionner pour leur travail mais perdent souvent de vue les contraintes liées à la livraison du produit. Le principal critère de sélection des développeurs doit s'appuyer sur leur niveau de sensibilisation aux exigences de livraison du produit. Lorsque la durée des cadres temporels n'est pas dictée par les besoins de l'entreprise, celle-ci doit néanmoins être la plus courte possible. Un délai de deux mois est largement suffisant pour l'implémentation d'un module complexe, surtout si l'on a recours aux outils et techniques de productivité.

Les caractéristiques fonctionnelles ne correspondent pas aux attentes de l'entreprise

Les deux principaux symptômes de ce risque sont la surcharge de fonctionnalités et le manque de recherche de solutions techniques.

La meilleure approche de gestion des risques consiste à respecter les principes de base de la méthode DSDM. La mise en œuvre d'un prototypage contrôlé et orienté vers les objectifs, le développement itératif ainsi que la forte participation des utilisateurs permettent aux développeurs à garder à l'esprit la première priorité, qui est de répondre aux besoins de l'entreprise.

En effet, les développeurs, assistés par les utilisateurs ambassadeurs, ont tendance à consacrer trop de temps à l'amélioration de l'interface utilisateur. L'élaboration d'un guide de style bien défini permet de limiter les excès de créativité dans ce domaine.

Les développeurs doivent rendre compte de leur progression par rapport aux objectifs plutôt que d'indiquer l'état d'avancement d'une tâche particulière. Cette approche permet de créer un bon état d'esprit au sein de l'équipe.

Il convient de réaliser des bilans réguliers et sérieux de chaque module, en fonction du jeu minimal de fonctions et des exigences fonctionnelles de chaque module. Les besoins de l'entreprise doivent être réexaminés afin de s'assurer que le projet avance dans la bonne direction. Par exemple, si les développeurs ne gardent pas constamment à l'esprit la maintenabilité du code, le système n'atteindra pas ses objectifs à long terme.

Le développement itératif et incrémentiel n'est pas bien maîtrisé

La surcharge de fonctionnalités constitue un risque spécifique au développement itératif et incrémentiel. La possibilité de changer d'avis est un principe fondamental dans un projet DSDM, bien qu'il représente une faiblesse lorsque l'équipe ne parvient pas à arrêter son choix sur les caractéristiques fonctionnelles du produit. Les délais très courts, le principe des primes et un personnel sensibilisé aux exigences sont les meilleurs remèdes contre ce piège. De plus, la consultation fréquente des utilisateurs clés aide à sélectionner les fonctionnalités indispensables.

Un autre risque consiste à ne pas adopter pleinement l'approche itérative et incrémentielle. Le cycle de vie en cascade, qui consiste à développer une itération puis un incrément, présente un certain nombre de difficultés et de dangers. Il faut éviter de découper le système en trop peu

d'incrément ou de réduire le nombre d'itérations car cela entraîne une limitation de la participation des utilisateurs.

Note : La conception modulaire du système facilite considérablement le développement incrémentiel. Si, au stade de l'analyse des besoins, la conception du système n'est pas modulaire, il est préférable de ne pas adopter la méthode DSDM comme processus de livraison, ou de remettre en question la conception du système.

Le retour en arrière est difficile voire impossible

Dans un projet de développement rapide, on est parfois tenté de prendre des raccourcis, en implémentant un incrément sans la possibilité de revenir à la version antérieure, si celui-ci doit être supprimé par la suite. Cette pratique constitue un risque majeur, c'est pourquoi la gestion de la configuration est primordiale dans un projet DSDM. Des outils fiables et un contrôle approprié de la gestion sont nécessaires pour éviter ce problème. Malheureusement, les outils de gestion de la configuration sont peu répandus. Il est donc nécessaire de consulter des manuels de méthodologie.

La gestion de la configuration requiert beaucoup de travail et de temps. C'est donc pour cette raison qu'il est recommandé de contrôler régulièrement le processus de développement, afin de vérifier qu'il est conforme aux objectifs.

Les besoins de haut niveau ne sont pas définis

Lorsque les besoins de haut niveau ne sont pas définis, les risques de surcharge de fonctionnalités et des dépassements de délais sont majeurs. D'une certaine manière, ce risque est aggravé lorsque l'équipe bénéficie d'une grande liberté d'action et que les utilisateurs participent pleinement. La tendance à étudier les besoins dans les moindres détails avant de geler le domaine d'application du système est humaine. Le meilleur moyen de geler le domaine d'application est de découper le système en modules, et d'attribuer à chacun d'eux un cadre de temps limité.

Une autre manière de limiter ce risque consiste à fixer avec les utilisateurs principaux une date limite pour la définition des besoins.

Les tests ne sont pas intégrés dans le cycle de vie

Si le système est développé selon une méthode itérative et incrémentielle, avec un niveau élevé de participation des utilisateurs, il sera testé tout au long du cycle de vie du projet. Néanmoins, il convient d'élaborer un système de mesure permettant de vérifier que les tests seront bien effectués.

L'évolution d'un projet DSDM est tellement rapide qu'il est essentiel d'adopter une approche de test du système claire et précise. Une approche de gestion des risques consiste à désigner parmi les membres de l'équipe, un responsable chargé de vérifier que les tests nécessaires ont été effectués et suffisamment documentés.

“Nous avons acheté une masse de technologies nouvelle. Par conséquent, il y avait trop de choses à apprendre, même avec le soutien des fournisseurs. La formation à DSDM a été malheureusement ignorée parmi toutes les autres choses que nous devons assimiler.”

Toutes les parties prenantes n'adhèrent pas une approche de collaboration et de coopération

Les systèmes informatiques de grande envergure impliquent souvent de nombreux participants, comprenant des fournisseurs et des revendeurs. Dans ce cas, l'équipe de développement doit se soumettre à un grand nombre de conditions et de termes contractuels.

Notamment, lorsque des contrats extérieurs sont requis, l'approche DSDM recommande une rémunération basée sur les fournitures et le temps passé plutôt qu'une rémunération fixe.

Lorsque cette conditions ne peut être appliquée, les activités de gestion des risques doivent inclure de fréquentes vérifications ainsi que la localisation du personnel sur site.

Les modifications relatives au domaine d'application et aux priorités doivent être évitées. Si néanmoins elles surviennent, les procédures de communication et de décisions deviennent rigides et le partenariat étroit requis dans un projet DSDM en pâtit. Si un manque de confiance s'insinue entre les développeurs et les utilisateurs, la méthode DSDM devient une obligation et n'est plus un atout.

18.4 Autres risques

L'approche DSDM n'est pas totalement applicable

Il est fortement recommandé d'effectuer l'évaluation des risques au début du projet, en utilisant le filtre d'adéquation (voir annexe A) ainsi que d'autres critères définis au chapitre 4. L'évaluation des risques doit être effectuée et mise à jour après chaque itération du modèle fonctionnel.

L'équipe de développement ne comprend pas le processus de développement

Si les développeurs ne sont pas familiarisés avec le processus de développement, une formation approfondie doit leur être dispensée avant le début du projet. De même, si les utilisateurs ne connaissent pas la méthode DSDM, il est souhaitable d'organiser à leur attention une journée de sensibilisation au début du projet. Afin d'éviter un problème de gestion d'emploi du temps propre aux projets de développement rapide, la formation des utilisateurs doit également s'adresser à la direction. En effet, il n'est pas rare que celle-ci continue d'exiger de ses employés qu'ils fournissent 50% de leur travail habituel, alors qu'ils consacrent 70% de leur temps au projet.

L'introduction du système va bouleverser l'organisation des utilisateurs

Si les utilisateurs participant au projet ne sont pas ceux qui utiliseront le produit fini, il convient alors de s'assurer de leur bonne volonté et de leur capacité à s'intégrer au processus de développement. Il est probable que les utilisateurs qui redoutent les changements ne participeront pas de manière satisfaisante au projet.

L'approche initiale du projet se révèle inappropriée au cours du développement

Au fur et à mesure que le projet avance, la nature du développement devient plus évidente et il peut s'avérer nécessaire de modifier ou d'abandonner l'approche DSDM.

Lorsque les besoins sont mis à jour et mieux compris, le jeu minimal de fonctions peut se révéler trop important et trop complexe pour être géré efficacement par une seule équipe de développement, comme il était prévu au départ. Il faut alors constituer une seconde équipe. Néanmoins, ce facteur est susceptible d'imposer une pression excessive aux utilisateurs impliqués dans le projet. C'est pourquoi il est conseillé de réduire leur niveau de participation.

Au cours des phases de conception et de construction, des problèmes techniques peuvent ralentir le projet, malgré que l'équipe s'efforce d'atteindre le niveau de fonctionnalité jugé essentiel dans le produit fini. Si une telle situation se produit, le chef de projet peut négocier avec les utilisateurs une réduction du niveau de fonctionnalité requis. Si cette première démarche s'avère infructueuse, il doit opter pour une approche de développement plus lente et centrée sur la technologie.

Les outils de développement ne prennent pas en charge la livraison par étapes

La sélection d'outils et de techniques conçus pour le cycle de vie en cascade peut retarder le projet. C'est notamment le cas des techniques de gestion de configuration ou des outils de GLAO (génie logiciel assisté par ordinateur) qui s'appliquent à l'intégralité du système et ne permettent pas la construction de modèles partiels.

Les développeurs ne sont pas familiarisés avec l'environnement de développement

Les longues courbes d'apprentissage ne sont pas acceptables dans un projet DSDM. Les développeurs doivent être familiarisés avec la majorité des composants de l'environnement de développement. Ils doivent maîtriser tous éléments liés aux activités de développement et de contrôle tels que le matériel, les outils, les techniques, les normes ainsi que les procédures de gestion de configuration. Lorsqu'un domaine de compétence spécifique est nécessaire, il convient de solliciter les spécialistes au début du projet afin de s'assurer qu'ils seront disponibles le moment venu. Si cette condition n'est pas respectée, la faillite du projet est assurée.

19 Qualité

19.1 Philosophie de la qualité

Du point de vue de l'utilisateur, la qualité d'un système informatique se définit souvent en termes de capacité et d'adéquation à ses besoins. Cet aspect peut être défini comme la «conformité aux objectifs» du système. Sachant qu'un des principes de base de la méthode DSDM stipule que la conformité du système aux exigences de l'entreprise est une condition requise pour l'acceptation de la livraison, l'approche DSDM répond parfaitement aux critères de qualité des utilisateurs. En d'autres termes, la qualité est inhérente au système et favorisée par le processus DSDM.

Différentes techniques DSSM permettent d'assurer la qualité du produit.

- Les ateliers de prise de décision garantissent la définition correcte et la prise en compte des spécifications du système dès le début du projet.
- La participation active et permanente des utilisateurs garantit la compréhension mutuelle de tous les intéressés.
- Les examens et bilans (des prototypes et de la documentation) permettent de vérifier la conformité du système aux exigences de l'entreprise. (Les critères de qualité sont décrits au chapitre 13.)
- Des tests approfondis permettent de contrôler la conformité des spécifications du système.
- La gestion de la configuration et des modifications permet de préserver la qualité déjà intégrée dans le système.

Dans certains environnements, les activités liées à la qualité sont perçues comme étant très onéreuses (parfois même comme un mal nécessaire), car elles accroissent les tâches administratives et engendrent des frais supplémentaires. Dans un projet DSDM, toutes les activités doivent concourir à l'élaboration du produit final. C'est pourquoi la conception de produits intermédiaires, qui augmente considérablement le nombre d'inspections nécessaires, n'est pas recommandée.

Le processus de contrôle de la qualité doit être flexible dans un projet DSDM. En effet, une trop grande rigidité des activités liées au contrôle de la qualité risque d'éloigner le projet de son objectif principal de conformité aux exigences de l'entreprise. L'assurance qualité des projets DSDM diffère de celle des projets traditionnels pour les raisons énumérées ci-après.

- Le nombre de produits étant limité, les activités de vérification et d'inspection sont réduites.
- L'objectif principal d'un projet DSDM étant la conformité du produit aux exigences de l'entreprise, les spécifications du système sont régulièrement contrôlées.
- Un projet DSDM comporte moins de modifications contractuelles car elles sont prévues dans le processus de développement.
- Les spécifications sont mieux définies et le processus de développement s'adapte mieux à l'évolution des besoins de l'entreprise.

19.2 Planification de la qualité

Bien que la mise en œuvre correcte de l'approche DSDM permette à elle seule de d'assurer la qualité du produit, il convient néanmoins de sélectionner les techniques d'assurance qualité les mieux appropriées, de désigner les membres de l'équipe chargés de les appliquer et de décider à quel moment elles seront déployées. En d'autres termes, le contrôle qualité doit être planifié. Le recours à un système de gestion de la qualité permet de réduire la planification du projet et d'identifier la meilleure pratique pour les projets futurs.

La planification de la qualité fait partie intégrante de la planification du projet, qui tient compte des délais, des ressources, des rapports, des contrôles de gestion, etc. Certaines entreprises élaborent un plan d'assurance qualité indépendant. D'autres ne souhaitent pas que l'assurance qualité soit perçue comme une activité à part et préfèrent intégrer leur plan d'assurance qualité dans la planification globale du projet.

Quelle que soit la planification, différents aspects de l'assurance qualité doivent être considérés :

- Identification des produits à livrer et de ceux qui requièrent des activités de contrôle qualité spécifiques.
- Choix de la méthode de contrôle qualité pour chaque type de produit (analyse ou test).
- A quel moment les contrôles qualité seront-ils effectués ? Ces contrôles sont-ils facultatifs ou obligatoires ? Faut-il contrôler tous les produits de même type ou seulement un échantillon ? Les contrôles doivent-ils être réalisés au cours du développement ou lorsque le produit est achevé ?
- Désignation des personnes responsables des inspections et des tests pour chaque type de produit ainsi que des personnes habilitées à valider le produit.
- Définition des critères de qualité utilisés pour évaluer chaque type de produit. (Les critères de qualité définis au chapitre 13 peuvent servir de référence.)
- Quelles seront les procédures d'assurance qualité utilisées ?
- Quels éléments seront retenus pour étayer les décisions à prendre ?
- Quelles seront les normes appliquées aux produits (normes de codification et guides de style) ?

19.3 Contrôle de la qualité

De nombreuses entreprises exigent un contrôle régulier des projets de développement afin de vérifier qu'ils sont conformes aux normes et aux pratiques de l'entreprise. De tels contrôles sont à proscrire dans les projets DSDM car ils génèrent un travail redondant et un déploiement d'effort inutile.

Il est prouvé que la méthode de contrôle la plus efficace consiste à comparer les normes, les procédures et les pratiques à l'expérience passée de l'entreprise. Cette approche est encore plus efficace lorsque des outils de développement rapide sont utilisés et que l'expérience de l'utilisation de nouveaux outils est recueillie pour les projets à venir. De même, l'expérience acquise de la méthode DSDM peut être affinée pour les futurs projets.

Les principales questions à traiter lors du contrôle d'un projet DSDM sont les suivantes :

- La participation des utilisateurs est-elle efficace ?
- Les utilisateurs ont-ils un pouvoir de décision ?
- Le cycle de vie est-il respecté ?
- Les conclusions des rapports d'inspection des prototypes sont-elles prises en compte ?
- Est-il possible de revenir en arrière si cela est nécessaire ?

- Les priorités sont-elles suivies ?
- Les cadres temporels sont-ils respectés ?

19.4 Exigences non fonctionnelles

Les exigences non fonctionnelles ont un impact important sur la manière dont le contrôle qualité est appliqué. Toutes les exigences non fonctionnelles doivent être soigneusement examinées afin d'évaluer l'impact qu'elles auront sur le processus de développement et sur le niveau de précision des tests statiques et dynamiques. Les exigences liées à la performance, la fiabilité, la sécurité et la maintenabilité sont d'une importance capitale dans les projets ayant pour objectif de livrer un système informatif dans un délai le plus court possible. L'entreprise doit décider le plus tôt possible de ce qui doit être fait tout de suite et de ce qui peut être remis à une date ultérieure.

La section 11.3 décrit les différents types de décisions pouvant être prises en fonction des exigences non fonctionnelles ainsi que les niveaux de maintenabilité acceptables dans un projet DSDM. Le même processus s'applique à toutes les exigences non fonctionnelles.

19.5 Ouvrages de référence

La British Standards Institution et le Consortium DSDM ont élaboré un guide à l'attention des entreprises qui utilisent la certification TickIT. TickIT est une procédure de certification spécifique en fonction du secteur d'activité, décrite dans la norme ISO 9000. Bien qu'elle ait été conçue pour les entreprises britanniques, elle peut également servir de référence aux entreprises localisées hors du Royaume Uni.

«The Dynamic Systems Development Method and TickIT» clarifie clause par clause les sections du guide TickIT, relatives à la méthode DSDM. Reportez-vous à la bibliographie pour obtenir de plus amples informations.

20 Gestion de la configuration

20.1 Introduction

Ce chapitre décrit l'importance de la gestion de configuration (GC) dans un projet DSDM et en présente les grandes lignes.

Les éléments suivants (dont certains sont des principes de base) sont des aspects fondamentaux de la gestion de configuration d'un projet DSDM :

- Les membres de l'équipe DSDM doivent être investis d'un pouvoir de décision sans surcharge administrative.
- Les outils et techniques de gestion de configuration ne doivent pas ralentir le processus de développement.
- Les modifications sont inévitables dans un processus de développement itératif.
- Tous les changements doivent être réversibles.
- Tous les modules et prototypes doivent être reproductibles (y compris les scripts de test et les données de construction).

Le respect de ces principes implique que la gestion de configuration doit être prise très au sérieux dans un projet DSDM.

20.2 Importance de la gestion de configuration

La gestion de configuration est rendue indispensable par la nature dynamique du processus DSDM. En raison du nombre important d'activités concurrentes et de la fréquence de livraison des produits, ces derniers doivent être rigoureusement contrôlés lorsqu'ils atteignent leur stade d'achèvement.

Au cours des phases de prototypage, les développeurs travaillent souvent sur les mêmes produits. C'est notamment le cas pour la structure des données, qui est commune à tous les modules. Il est par conséquent indispensable que les développeurs sachent que les données qu'ils utilisent sont parfaitement connues et contrôlées et qu'ils soient capables de prévenir ou de résoudre les éventuelles situations d'incompatibilité ou de conflit.

Il arrive qu'un prototype se révèle être sans avenir. Dans ce cas, il est indispensable de pouvoir revenir en arrière à un état antérieur connu afin de redéfinir le travail.

De même, il est essentiel d'être en mesure d'effectuer des tests de régression à chaque étape de construction des modules et d'identifier les versions des différents composants.

20.3 Ressources humaines

Les membres de l'équipe de développement doivent être formés aux outils et aux procédures de gestion de configuration utilisés. L'élaboration d'un système de gestion de configuration efficace exige non seulement que les développeurs maîtrisent les outils techniques mais également qu'ils sachent recueillir les informations relatives à la conception et aux modifications du produit, qui doivent être enregistrées.

Si tel n'est pas le cas, des informations capitales, qui devraient être accessibles à tous les développeurs, risquent d'être occultées. La gestion de configuration est essentiellement basée sur la communication de l'état du système. Les développeurs doivent avoir l'assurance qu'ils travaillent sur des bases saines à chaque fois qu'ils développent un composant ou qu'ils reviennent à une version antérieure.

Le contrôle des produits doit faire partie intégrante des activités de l'équipe de développement. La méthode traditionnelle qui consiste à confier la gestion de configuration à

une personne extérieure à l'équipe de développement peut mettre en péril le projet DSDM. Les membres de l'équipe de développement doivent garder le contrôle des éléments de configuration.

C'est aux développeurs qu'incombe la responsabilité de déterminer l'impact d'une modification. C'est également à eux de juger si le produit qu'ils mettent en œuvre doit faire partie des éléments de configuration.

Néanmoins, les développeurs omettent bien souvent de référencer le produit sur lequel ils travaillent. C'est pourquoi il est nécessaire de désigner un responsable parmi les membres de l'équipe, qui sera chargé de gérer la configuration et de résoudre les divergences d'opinions. Ce responsable est généralement le coordinateur technique. S'il quitte l'équipe de développement, il doit être remplacé par un membre de l'équipe.

20.4 Stratégie de gestion de configuration

La stratégie de gestion de configuration doit être définie avant la fin de la phase d'analyse des besoins. Elle doit être visible pour tous les développeurs et doit au moins indiquer à quelle fréquence les modifications seront identifiées. Voici quelques exemples :

- Identification de chaque prototype avant la démonstration.
- Identification quotidienne, qui permet une plus grande flexibilité et reflète la vitesse d'évolution d'un produit.
- Identification à la fin d'un cadre temporel, qui constitue le minimum absolu et doit être réservée aux cadres temporels de très courte durée. Cette stratégie est déconseillée lorsque des cadres temporels d'une durée supérieure à trois semaines sont fréquemment utilisés.

Les autres aspects de la stratégie de gestion de configuration comprennent la définition des éléments de configuration, de leur modularité ainsi que de la méthode de contrôle des modifications. Cette dernière diffère selon que le fournisseur du système est interne ou externe à l'entreprise. L'objectif principal est la livraison du produit à la date prévue. Pour atteindre cet objectif, il est indispensable de définir très tôt la procédure de validation des modifications, qui requiert l'analyse de l'impact de ces modifications.

20.5 Caractéristiques de la gestion de configuration dans un projet DSDM

Normalisation

La norme ANSI 1042:1987 décrit comme suit l'aspect administratif de la gestion de configuration :

La plupart des activités courantes de gestion de configuration peuvent être facilement automatisées en raison de leur caractère répétitif et administratif. L'efficacité de la méthode de gestion de configuration repose sur une planification rigoureuse permettant de savoir comment les tâches seront effectuées. Les principales activités de gestion de configuration consistent à définir une procédure d'identification des éléments de configuration, des composants et des modules ainsi qu'à planifier l'examen systématique des modifications avant d'autoriser leur implémentation dans un programme. Ces activités font appel à des connaissances techniques. Le rôle des personnes chargées de la gestion de configuration est d'allier ces considérations techniques aux décisions de gestion tout en fournissant l'assistance administrative nécessaire, sans pour cela ralentir le processus de décision.

La suite de cette section est consacrée aux aspects essentiels de la gestion de configuration, qui doivent être pris en considération dans un projet DSDM.

Procédures administratives

La norme ANSI 1042:1987 indique que les procédures administratives peuvent être facilement automatisées. Néanmoins, le recours à un processus d'automatisation n'est pas envisageable lorsque la gestion de la configuration comprend un facteur humain.

L'aspect administratif peut être éliminé grâce à l'utilisation d'outils de gestion de configuration. Certains outils de développement de haut niveau intègrent des modules de gestion de configuration. Si la possibilité de choisir leur est donnée, les membres de l'équipe de développement doivent adopter ces outils.

Quels que soient les outils sélectionnés, certaines restrictions s'imposent quant à l'utilisation des outils de gestion de configuration. En effet, dans un processus de développement parallèle, tous les développeurs ont accès à l'ensemble des éléments de configuration. Il est alors nécessaire de verrouiller le développement simultané d'un même élément de configuration, ou de résoudre les conflits qui surviennent lors du développement parallèle.

Éléments de configuration DSDM

Le principal aspect de la gestion de configuration consiste à sélectionner les éléments de configuration, c'est-à-dire les éléments qui doivent être contrôlés pour que tous les membres de l'équipe de développement soient informés de l'état des différents produits. En raison de la nature dynamique du processus DSDM, ces informations doivent être les plus complètes possible, en évitant toutefois de surcharger les développeurs de tâches administratives. Le choix judicieux des outils et des procédures permet d'éviter ce problème.

La gestion de configuration est souvent appliquée à la fin du développement des modules qui constitueront le produit final. Dans ce cas, la documentation associée aux produits développés est traitée séparément. Dans les projets DSDM, il est conseillé d'introduire la gestion de configuration aussi tôt que possible, et de traiter tous les documents de la même manière que les produits. Les éléments de configuration doivent être identifiés au début du projet. A l'issue de chaque phase, tous les éléments de configuration doivent être identifiés, avant de passer à la phase suivante.

Les modèles dynamiques doivent bien évidemment faire partie des éléments de configuration. A l'issue d'un cycle de prototypage, chaque prototype doit être identifié comme élément de configuration. Par exemple, si le modèle fonctionnel inclut trois itérations de prototypage, chacun des trois prototypes doit faire l'objet de la gestion de configuration. Le prototype validé peut alors passer à la phase suivante. L'élément de configuration du prototype est constitué du prototype, des tests qui lui ont été appliqués et d'un compte-rendu des observations des utilisateurs. Les commentaires des utilisateurs permettent de déterminer l'orientation de la phase suivante.

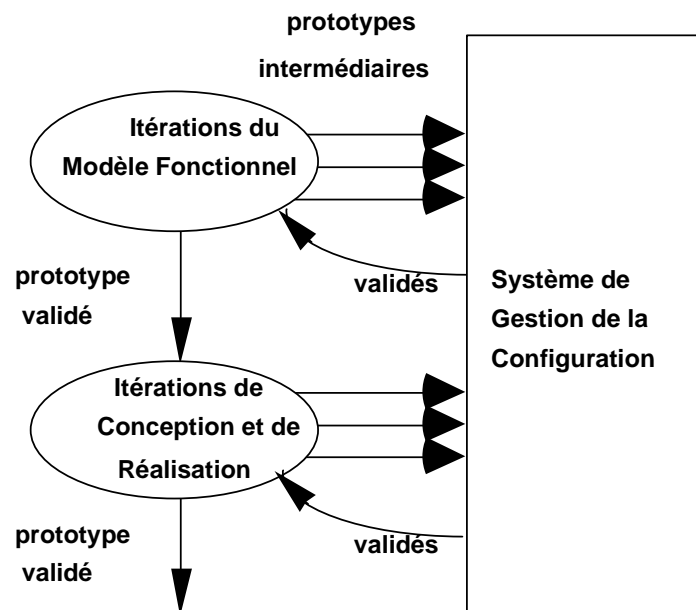


Figure 20.1 : Contrôle de configuration des prototypes

Les spécifications du système sont mises à jours et affinées tout au long du processus de développement. Il est indispensable que tous les membres de l'équipe soient informés de l'état le plus récent de chaque spécification. Par conséquent, toutes les spécifications, y compris celles définies lors de L'étude de faisabilité, doivent être considérées comme des éléments de configuration.

Les spécifications de haut niveau doivent être définies le plus tôt possible afin de s'assurer que le travail de tous les développeurs converge vers le même objectif. Les spécifications principales sont définies au début du projet, lors de la phase d'analyse des besoins.

Les démonstrations ne servent pas uniquement à vérifier que toutes les erreurs ont été corrigées. Elles permettent également de s'assurer que les spécifications définies par l'entreprise sont toujours intégrées dans le système et qu'elles n'ont pas été supprimées au cours du processus de développement. En effet, toutes les spécifications définies lors de la phase d'analyse des besoins doivent constituer des éléments de configuration, même si certaines paraissent futiles aux développeurs.

Il est indispensable d'être en mesure d'analyser l'impact d'une modification, surtout lorsqu'elle est commune à tous les développeurs. Par exemple, si la taille d'un champ est modifiée dans la base de données, les développeurs doivent connaître les programmes, les rapports et les écrans qui en seront affectés. Cela signifie que chaque champ de la base de données constitue un élément de configuration potentiel. Néanmoins, l'analyse de l'impact de cette modification peut être effectuée au niveau de la table. C'est pourquoi le niveau de modularité du système est un élément essentiel pour la gestion de configuration.

Tous les produits testés ainsi que leur version antérieure doivent faire partie des éléments de configuration. De même, tous les tests effectués doivent être disponibles dans le cas où des tests de régression seraient requis. Ainsi, s'il s'avère nécessaire de revenir à une version antérieure, les tests correspondants pourront être réutilisés.

Il est conseillé de conserver des copies de sauvegarde de tous les états connus des différents modules et de préciser leur existence dans le système de gestion de la configuration.

Le personnel de gestion de la configuration

La norme ANSI précitée indique clairement que le personnel de gestion de la configuration doit être exclusivement constitué par les membres de l'équipe de développement.

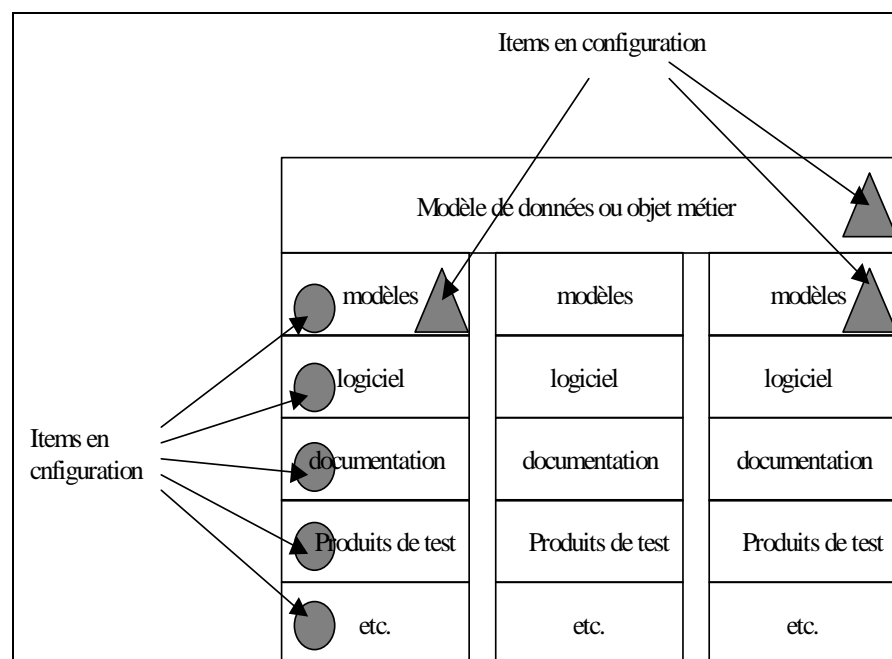
20.6 Mise en œuvre de la gestion de la configuration

Dans les projets DSDM, il est recommandé d'adopter une approche générique fondée sur les procédures courantes, tout en gardant à l'esprit les principes de gestion de configuration énoncés précédemment. Enfin, les procédures de gestion de configuration ne doivent pas entraver les habitudes de travail des développeurs.

Bien qu'il ne faille pas sous-estimer l'importance de cette tâche, l'élaboration d'un système de gestion de configuration rigide risquerait de retarder la livraison du système.

20.7 Outils

Les outils de gestion de configuration peuvent être utiles. Néanmoins, ils ne sont pas indispensables. Une gestion efficace peut être réalisée à l'aide des répertoires inclus dans les outils de documentation et de développement. Il est alors facile de contrôler la progression verticale d'un produit. Par contre, toute la difficulté consiste à le relier horizontalement aux produits qui ont été développés dans un autre module.



Erreur ! Source du renvoi introuvable. **Figure 20.2 : Relations entre les éléments de configuration**

Les outils doivent être capables de gérer tous les éléments de configuration tels que les objets, les fonctions, les documents, le plan de configuration, les bibliothèques, les groupes de composants logiciels, les structures de données, etc.

Ils doivent également permettre l'enregistrement des données et des scripts de tests, afin que des tests de régression puissent être effectués en cas de besoin.

« Nous n'avons jamais été très forts en gestion de configuration, et nous avons donc essayé de l'ignorer pendant notre projet pilote – mais nous avons perdu la maîtrise même avec une équipe de trois développeurs. Néanmoins, les utilisateurs adorent le système ! »

21 Tests

« Je ne connais pas le nombre total de points de fonctions : il y en a certainement des tonnes ! Mais nous avons eu beaucoup moins d'erreurs qu'habituellement avec un tel système. Tester au fur et à mesure: ça marche ! »

21.1 Introduction

Dans un projet DSDM, les tests doivent menés tout au long du cycle de vie et étayés par des techniques éprouvées de génie logiciel.

Le recours à la méthode DSDM est justifié lorsque le produit doit être développé dans un délai très court. C'est pourquoi le respect des principes DSDM est souvent primordial pour l'entreprise. Les tests, dont le but est d'améliorer la qualité et la fiabilité du système, n'en sont pas moins importants. L'annexe B expose les grandes lignes de la planification des tests.

21.2 Principes de test

Dans un projet DSDM, les tests reposent sur les six principes suivants :

PT.1 Validation

Principe

Dans un projet DSDM, l'objectif prioritaire des tests est de contrôler à tous les niveaux la conformité du produit aux attentes de l'entreprise (voir principe IV au chapitre 2).

Commentaires

Le résultat des tests doit être examiné à l'issue de chaque phase du développement, afin de s'assurer que les fonctionnalités du système correspondent bien aux besoins de l'entreprise, et ce même s'ils n'ont pas été clairement spécifiés par écrit.

Le niveau d'adéquation du processus de développement aux activités informatisées de l'entreprise permet de mesurer le degré de «conformité aux objectifs» du système.

La conformité aux besoins de l'entreprise doit être vérifiée lorsque de nouvelles spécifications ou des modifications sont ajoutées ou lorsque certaines tâches sont automatisées.

PT.2 Tests basés sur les besoins de l'entreprise

Principe

Le test des composants du système destinés à répondre aux besoins de l'entreprise constitue la priorité principale.

Commentaires

Les tests sont toujours soumis à des contraintes de temps et de ressources et le temps consacré aux tests n'est jamais suffisant pour localiser toutes les erreurs. C'est pourquoi les tests doivent se concentrer essentiellement sur la conformité des fonctionnalités du système aux besoins de l'entreprise.

PT.3 Tests basés sur les erreurs

Principe

Le but d'un test est d'identifier les erreurs.

Commentaires

Les tests ne permettent pas de prouver qu'un système informatique fonctionne. Ils aident simplement à créer un sentiment de confiance, lorsque des erreurs sont identifiées puis corrigées.

PT.4 Tests effectués tout au long du cycle de vie**Principe**

Les tests doivent être effectués sur tous les produits et à tous les stades du processus de développement DSDM.

Commentaires

Il n'y a pas à proprement parler de phase de test dans un projet DSDM. Les produits et la documentation qui leur est associée sont générés tout au long du processus de développement. C'est pourquoi la planification des tests doit faire partie intégrante du cycle de vie itératif.

PT.5 Tests indépendants**Principe**

Les produits ne doivent pas être testés par leurs créateurs.

Commentaires

Les tests sont plus efficaces lorsqu'ils sont effectués par une personne qui n'a pas participé au développement du produit. Dans les projets DSDM, le fort niveau de participation des utilisateurs permet d'avoir constamment un point de vue extérieur.

PT.6 Réutilisation des tests**Principe**

Les tests doivent pouvoir être réutilisés.

Commentaires

Bien que les tests deviennent obsolètes à mesure que le prototype évolue, certains d'entre eux peuvent néanmoins être réutilisés. La réutilisation d'un ancien test est indispensable lors de l'amélioration d'un prototype. Il arrive également qu'une version antérieure d'un prototype soit réutilisée. Dans ce cas, il est nécessaire de récupérer les composants logiciels et la documentation utilisateur, mais également la batterie de tests associée à ce prototype.

Pour qu'il puisse être reproduit, un test doit être documenté. Il est nécessaire de garder une trace de l'exécution de chaque test et de conserver les résultats à des fins de contrôle et de validation. Les outils de test permettent d'élaborer rapidement des tests réutilisables et réduisent l'effort de création et de maintenance de la documentation.

21.3 Mise en œuvre des tests**Conformité aux objectifs**

Dans un projet DSDM, toutes les parties prenantes doivent s'entendre sur la stratégie à adopter pour atteindre les objectifs. Le système est conforme aux objectifs lorsqu'il répond à toutes les exigences émises par l'entreprise. Ces exigences peuvent être appréhendées comme une hiérarchie d'objectifs associée aux activités de l'entreprise réalisées au moyen du système développé.

Cette conception hiérarchique des objectifs remplace la méthode traditionnelle de définition des spécifications fonctionnelles du système. Une telle approche est plus efficace car elle met l'accent sur les résultats plutôt que sur les mécanismes mis en œuvre pour les atteindre.

Les exigences de l'entreprise sont établies lors de la phase d'analyse des besoins. À partir de ces exigences, une hiérarchie d'objectifs est établie, permettant de mettre à jour les conditions requises pour la réalisation d'un système de qualité. La figure 21.1 illustre les relations existant entre le système informatique et les objectifs.

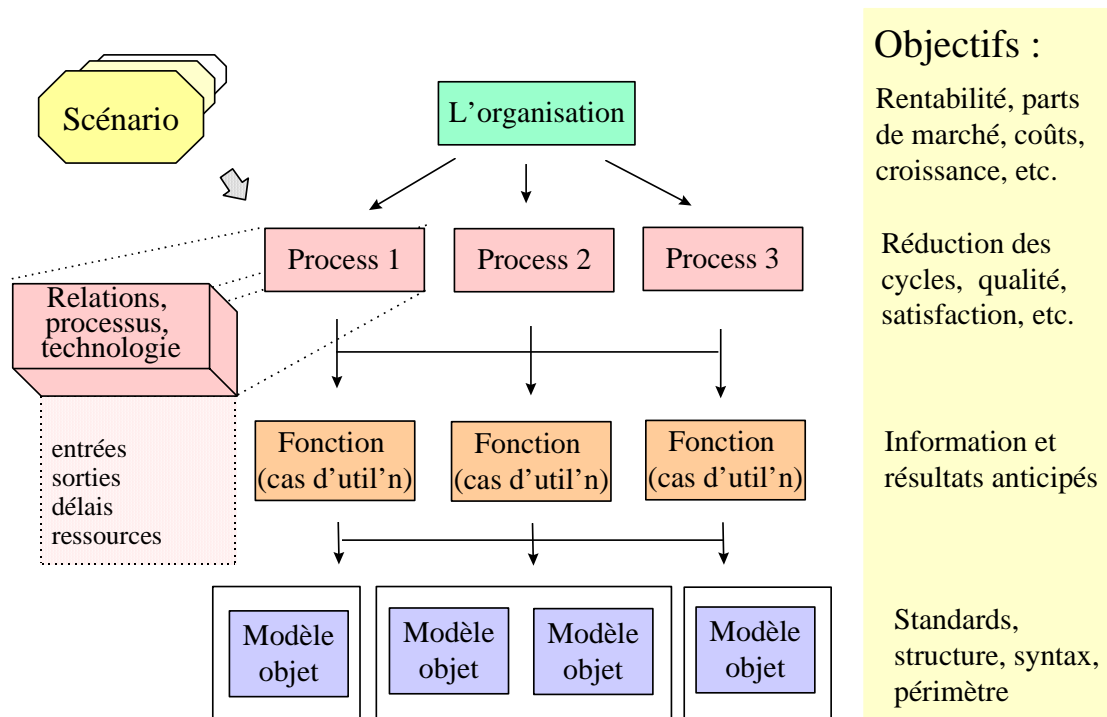


Figure 21.1 : Relation entre les objectifs et le produit fini

Attribution de priorités

Au même titre que toutes activités de développement DSDM, les tests doivent être classés en fonction de leur priorité. Les composants du système dotés des caractéristiques suivantes doivent être testés en priorité :

- Gros volumes de traitement.
- Activités complexes de l'entreprise, exigeant une charge de travail importante.
- Interface homme-machine, surtout si la clientèle de l'entreprise aura accès au système informatique (par exemple s'il s'agit d'une application de guichet, d'une application Internet ou d'une borne interactive).

Validation progressive du système

La première étape est la phase d'analyse des besoins, lors de laquelle les grandes lignes du système du définies. Celui-ci évolue ensuite au cours des itérations du modèle fonctionnel, de conception et de construction. Chaque itération doit être testée. Les tests peuvent être statiques ou dynamiques, selon la technique utilisée. Le choix de la technique dépend de la nature du produit et du type de test utilisé. Tous les produits documentaires doivent faire l'objet d'inspections, et tous les modèles (y compris les prototypes) peuvent être soumis à des tests dynamiques.

La phase finale de test comprend le test du système, le test d'intégration de haut niveau ainsi que les essais des utilisateurs.

La méthode DSDM étant fondée sur le développement incrémentiel, les procédures de test doivent permettre de s'assurer que le passage à un nouvel incrément n'affecte pas les données et les fonctions déjà intégrées dans le système. Cela implique la nécessité d'effectuer des tests de régression au cours du développement des prototypes verticaux et/ou de la livraison incrémentielle du système.

Phase DSDM	Tests
Etude de faisabilité	Test du prototype
Analyse des besoins	Aucun
Itération du modèle fonctionnel	Test du prototype Test de l'unité Test d'intégration de faible niveau Test d'acceptation de l'utilisateur de faible niveau Test de régression (pour les prototypes verticaux)
Itération de conception et de construction	Test du prototype Test de l'unité Test d'intégration de faible niveau Test du système Test d'intégration du système Test de régression (en cas de livraison évolutive de plusieurs prototypes) Test d'acceptation de l'utilisateur
Mise en œuvre	Test d'acceptation fonctionnelle Test déploiement et de répartition

Table 21.1 : Tests dynamiques au cours des différentes phases DSDM

21.4 Test du prototype

Un prototype DSDM peut avoir deux rôles différents :

- C'est un composant partiel du système.
- Il permet de réunir les informations nécessaires à la clarification des spécifications fonctionnelles et non fonctionnelles.

Différents critères de test s'appliquent en fonction du rôle du prototype :

- Lorsque le prototype représente un composant fonctionnel du système final, la réussite ou l'échec d'un test dépendent du niveau de concordance des résultats obtenus avec les **résultats escomptés**, basés sur les spécifications connues.
- Lorsque le rôle du prototype est de clarifier les spécifications, les critères de réussite du test sont basés sur la qualité des informations générées, par rapport aux **informations escomptées**. Néanmoins, il n'est pas possible de prédéfinir ces informations de la même manière que les **résultats escomptés**. Dans ce cas, un test est réussi lorsqu'il permet d'étendre la connaissance et la compréhension du testeur.

Un seul prototype peut jouer à la fois ces deux rôles. Dans ce cas, il doit être testé en fonction des deux types de critères.

Au cours d'un processus DSDM, le prototypage est généralement itéré trois fois. Les prototypes peuvent être divisés en quatre catégories. Toutes les catégories de prototypes peuvent être testées en même temps en fonction des «résultats escomptés» et des «informations escomptées».

Le tableau 21.2 présente les quatre catégories de prototypes ainsi que les différents types de test appliqués à chaque itération.

Catégorie de prototype	Itérations au cours des phases DSDM		
	Recherche	Amélioration	Consolidation
Objectifs	IE	IE/RE	RE
Exploitation	IE	IE/RE	RE
Performance	IE	IE/RE	RE
Fonctionnalité	IE	IE	IE

Tableau 21.2 : Critères de test des différentes catégories de prototype
(IE = Information Escomptée, RE = Résultats Escomptés)

21.5 Types de tests

Le tableau 21.3 présente les différents types de tests applicables. Chaque type de test s'applique à une catégorie de spécifications techniques, fonctionnelles ou non fonctionnelles.

Type de test	Applicabilité
Fonctionnalité	Principales activités de l'entreprise
Utilisation	Interface utilisateur, facilité d'utilisation, efficacité des tâches utilisateur
Performance	Temps de réponse, volume
Sécurité	Sécurité d'accès, sécurité des données et des communications
Exploitation	Installation, exploitation, sauvegarde/restauration, redémarrage/reprise, configuration
Conversion	Conversion des données issues de systèmes existants

Tableau 21.3 : Les différents types de tests et leur applicabilité

Les tests relatifs aux informations utilisateur (documentation, aide, support de formation) sont intégrés dans une des catégorie ci-dessus.

Tous les systèmes doivent faire l'objet de tests de fonctionnalité. Pour les petits systèmes informatiques, les différents types de tests relatifs aux spécifications non fonctionnelles (performance, exploitation, etc.) sont uniquement nécessaires lorsque le système est implémenté sur une nouvelle plate-forme dont les caractéristiques techniques ne sont pas encore connues. Ces tests peuvent également être limités ou supprimés si les caractéristiques à analyser ont déjà été testées dans un prototype

Techniques de test

D'une manière générale, les tests se divisent en deux principales catégories : les tests statiques et les tests dynamiques.

Les tests statiques regroupent toutes les formes de tests «humains» (inspections, analyses, revues) ainsi que l'analyse du code à l'aide d'un programme utilitaire appelé «analyseur statique». Les tests statiques doivent être réservés aux documents importants.

Les tests dynamiques s'adressent essentiellement aux prototypes dynamiques, au système testé et au système livré. Les tests dynamiques appliqués au système testé et au système livré doivent être sélectionnés en fonction de l'histoire des prototypes. Lorsque le système livré a été élaboré à partir de plusieurs prototypes, les tests de l'unité, d'intégration et d'acceptation doivent avoir été exécutés au cours du développement de ces prototypes.

Un projet DSDM ne requiert pas de techniques spécifiques quant à la mise en œuvre des tests. L'association classique des techniques structurelles et fonctionnelles (boîte blanche et boîte noire) s'applique parfaitement. Le choix de la technique, de la profondeur et du champ d'application du test dépend de la technologie utilisée pour élaborer le système, ainsi que des caractéristiques du projet. Néanmoins, en raison des contraintes de temps, les tests sont moins rigoureux dans un projet DSDM. Les principaux aspects de l'élaboration d'un test sont les suivants :

- documentation des tests,
- techniques d'élaboration ou de sélection des tests,
- contrôle de l'environnement de test et l'exécution des tests.

Dans un projet DSDM, les procédures de test requièrent une documentation moins importante. Une simple liste des conditions à tester est suffisante. Cette liste, établie en fonction des objectifs techniques et des besoins de l'entreprise, permet de déterminer le champ d'application et le niveau de priorité des tests. Cette même liste peut servir à consigner le résultat des tests (échec ou réussite). Dans la majorité des cas, il n'est pas nécessaire de documenter les résultats escomptés, qui sont définis par les testeurs lors de l'exécution du test. Les testeurs sont généralement les utilisateurs. Enfin, le recours à certains outils de capture et de réexécution permettent d'accroître le niveau de documentation.

Toutes les techniques d'élaboration de tests, telles que la «division d'équivalence», peuvent être utilisées dans un projet DSDM. Toutefois, en raison des contraintes de temps associées à ce type projet, ces techniques ne seront efficaces que si elles sont parfaitement maîtrisées par les développeurs, ce qui requiert une solide formation et une bonne expérience pratique de leur utilisation.

Dans la plupart des projets DSDM, les tests sont conçus en fonction du point de vue de l'utilisateur final, via l'interface utilisateur. Ces tests seront élaborés à partir des scénarios de l'entreprise. Les développeurs devront néanmoins tester les fonctionnalités non visibles telles que la mise à jour des bases de données, ainsi que les caractéristiques non fonctionnelles telles que la performance du système.

Les techniques conventionnelles utilisées pour tester la performance, l'accès simultané ou la sécurité s'appliquent parfaitement aux projets DSDM.

Les systèmes informatiques élaborés au moyen du processus DSDM sont généralement dotés d'une interface graphique utilisateur et s'appuient sur le modèle client-serveur. Il n'est pas rare non plus qu'ils incluent des programmes orientés objet. Le recours à ces techniques soulève des problèmes en matière de test, qu'aucune méthode de développement ne parvient à résoudre parfaitement. Voici quelques solutions aux problèmes engendrés par l'utilisation de ces nouvelles technologies :

- En ce qui concerne l'interface graphique utilisateur et le modèle client-serveur, les tests peuvent être segmentés afin que chaque élément puisse être traité indépendamment des autres (interface graphique utilisateur, communications, codification, etc.). Pour des raisons évidentes de commodité, les segments correspondent en générale à des prototypes.
- Les tests d'intégration ne sont plus verticaux mais horizontaux. Les tests des fonctionnalités non visibles sont effectués via l'interface graphique utilisateur, ce qui déclenche une chaîne d'événements qui testent l'intégration du serveur.
- En ce qui concerne les programmes orientés objet, il existe de nombreuses techniques spécifiques permettant d'effectuer des tests d'intégration sur les objets. La plupart de ces méthodes sont basées sur l'historique des messages transmis d'un objet à l'autre.
- La méthode conventionnelle des «tests de transition d'état» a prouvé son efficacité en ce qui concerne l'interface graphique utilisateur, le modèle client-serveur et la programmation orientée objet.

La manière dont ces techniques sont mises en œuvre dépend essentiellement de la compétence des développeurs. Avant de sélectionner ces techniques dans un projet DSDM, une entreprise doit avoir la certitude que ses développeurs sont capables de mener à bien les tests dans un environnement moins dynamique.

21.6 Outils de test

Le recours aux outils de test permet d'accélérer le processus et facilite la tâche des développeurs qui savent que leur programme tourne bien. Cette section décrit les principaux outils de test.

Les analyseurs statiques de code sont les outils les mieux appropriés aux projets DSDM car ils fournissent un bon niveau d'inspection du code. Ils accroissent la vitesse de progression du projet car il n'est plus nécessaire d'attendre la vérification d'une troisième personne.

Afin de palier au manque de rigueur des tests, des outils d'analyse dynamique tels que les vérificateurs de limites de tableau, les détecteurs de perte de mémoire ou les profileurs permettent de détecter les erreurs lors des démonstrations. Sans ces outils, ces erreurs ne seraient pas détectées.

Une autre catégorie d'outils de test est utilisée pour la capture et la réexécution. Ils s'avèrent être d'une grande utilité pour l'élaboration et la réexécution des tests. A mesure que le système évolue, d'anciens tests peuvent être répétés afin de s'assurer que les fonctionnalités stables du système n'ont pas été affectées par les modifications. En outre, les tests automatisés peuvent être exécutés toutes les nuits.

Toutefois, il faut savoir que ces outils sont très sensibles aux modifications de l'interface utilisateur, ce qui les rend parfois inutiles. Les deux recommandations ci-après vous permettront d'éviter cet écueil.

- Assurez-vous que l'outil sélectionné est capable d'ignorer les modifications «esthétiques» de l'interface utilisateur (il doit prendre en compte les objets et non les pixels), et qu'il est doté d'un excellent système de mise à jour des scripts.
- Adoptez une stratégie d'élaboration de prototypes qui permet de stabiliser le plus tôt possible l'interface utilisateur. Cela ne signifie aucunement que l'interface doit être gelée très tôt au cours du processus de développement. Néanmoins, lorsque l'interface utilisateur subit d'importantes modifications, un outil de capture et de réexécution, aussi perfectionné soit-il, ne produira pas tous les bénéfices escomptés. Plus tôt les prototypes passeront cette étape, plus vite les outils seront efficaces.

Les outils de capture et de réexécution peuvent avoir une autre fonction que celle de réexécuter les tests. Ils permettent également de documenter les tests. En l'absence de script de test, la manière la plus rapide de documenter les tests est de les enregistrer lors de leur exécution, grâce à la fonction d'enregistrement des outils de capture et de réexécution. Le script ainsi enregistré pourra être réexécuté et archivé.

Type d'outil	Test du prototype	Test d'intégration	Test du système	Test d'acceptation	Test de régression
Générateurs de données de test	X	X	X	X	X
Analyseurs statiques	X				
Analyseurs dynamiques	X	X	X		
Pilotes, simulateurs, émulateurs	X	X	X	X	X
Capture/ réexécution, comparateur	X	X	X	X	X
Analyseurs de performance	X		X	X	X
Gestion des tests	X	X	X	X	X

Tableau 21.4 : Domaines d'application des outils de test

21.7 Tests des projets de grande envergure

Les projets de grande envergure, qui requièrent des ressources nettement plus importantes que la moyenne, soulèvent un certain nombre de problèmes :

- Toutes les équipes du projet n'utilisent pas la méthode DSDM, ce qui induit des conflits concernant le niveau de documentation et la coordination des livraisons.
- Les équipes qui ont recours à des cadres temporels en guise de méthode de gestion de projet génèrent des produits «flexibles», ce qui peut entraîner des problèmes d'intégration.

Les mécanismes de test des projets DSDM de grande envergure ne diffèrent pas de ceux décrits précédemment. Toutefois, l'échelle du projet, définie essentiellement par le nombre de personnes par équipe et le nombre de composants à tester, met en évidence l'importance des éléments décrits ci-après.

Tests d'intégration

Les tests d'intégration ont une importance beaucoup plus grande dans les projets qui comportent de nombreux sous-systèmes ainsi que des livraisons multiples et fréquentes. La multiplicité des équipes de développement participant à un même projet justifie la création d'une équipe indépendante responsable des tests d'intégration. Le rôle de cette équipe est de recueillir le travail des équipes de développement, de le tester puis de l'intégrer afin de constituer un incrément du système.

La stratégie de test doit refléter l'importance des tests d'intégration et définir comment les modules développés seront testés et intégrés dans le système global, sans affecter ni déstabiliser les composants déjà intégrés. Dans les projets de grande envergure, le domaine d'application des tests d'intégration inclut potentiellement des éléments non logiciels, tels que le matériel, la formation ou de nouvelles structures de l'entreprise. Il est essentiel de débiter très tôt les tests d'intégration et de préserver la cohérence du projet. Les personnes chargées de la planification des tests sont les responsables de l'infrastructure du projet, les membres du support technique ainsi que les développeurs principaux.

Interfaces

Afin d'obtenir une solution intégrée conforme aux objectifs, il est parfois nécessaire de classer les interfaces en fonction de leur priorité. Ces dernières doivent être identifiées lors de la phase d'analyse des besoins (incluse dans la phase de définition de l'architecture du système), et définies de manière détaillée à l'attention de tous les intervenants qui participeront à son développement.

Architectures

Les tests d'intégration mettent en évidence l'importance des architectures. Ils permettent de s'assurer que tous les modules et sous-systèmes assemblés prennent en charge les processus de l'entreprise.

Les tests de l'architecture ont pour objet de déceler les éventuels problèmes relatifs à la définition de l'architecture du système. Toutefois, il est impossible de prévoir la capacité et la performance finales. C'est pourquoi les tests de l'architecture doivent évoluer avec le système.

Outils de test

Dans les projets de grande envergure, il peut s'avérer utile de créer un «bureau de la modélisation», chargé de tester les prototypes. Le volume important de tests requis dans un projet de grande envergure (essentiellement les tests d'intégration) étend le domaine d'application des outils d'automatisation de tests. Le bénéfice de ces outils est particulièrement appréciable lorsqu'ils sont appliqués aux tests de régression. La stratégie et la planification des tests doivent tenir compte de l'utilisation potentielle de ces outils.

Niveau de rigueur des tests

Une approche simple et informelle, qui est tout à fait appropriée dans un projet DSDM de petite taille, se révélera infructueuse dans le cadre d'un projet de plus grande envergure. En effet, l'accroissement de l'échelle du projet, essentiellement en termes d'effectifs, exige des procédures de test plus strictes et rigoureuses. Toute la difficulté consiste à doser le niveau de formalisme des tests pour s'assurer de leur efficacité, sans pour cela compromettre les principes de la méthode DSDM ni générer une surcharge de tâches administratives.

22 Techniques de modélisation

22.1 Introduction

Le présent chapitre décrit les techniques de modélisation pouvant être appliquées aux processus de développement DSDM. Il tente d'identifier les techniques appropriées aux projets DSDM et donne un aperçu de leurs objectifs au cours de chaque phase du cycle de vie. L'étude détaillée de ces techniques ne peut pas être traitée dans ce manuel en raison de leur grande diversité. Néanmoins, la bibliographie incluse à la fin de cet ouvrage contient des références de publications qui décrivent de manière plus approfondie les techniques de modélisation.

Le processus de développement DSDM peut être divisé en plusieurs phases, caractérisées par un objectif différent. Les principales phases comprennent L'étude de faisabilité, la phase d'analyse des besoins et la phase d'itération du modèle fonctionnel.

Selon ce concept, une méthode de développement doit être fondée sur :

- un cadre structuré permettant de définir le processus de développement,
- un ensemble de techniques permettant de modéliser le système,
- une série de procédures et de techniques permettant la déduction des modèles,
- des heuristiques permettant d'améliorer le processus de développement,
- un langage commun permettant d'échanger les idées et de transmettre les décisions.

Ce chapitre a pour objet l'identification et la description des différentes perspectives à partir desquelles un système peut être modélisé.

22.2 Définitions terminologiques

Les termes suivants sont employés tout au long de ce chapitre :

- **Modèle** : abstraction de certaines caractéristiques du système ou de l'entreprise, considérées d'un point de vue particulier.
- **Technique de modélisation** : méthode de représentation schématique d'un aspect spécifique du système ou de l'entreprise
- **Produit DSDM** : ensemble d'un ou plusieurs modèles, accompagné d'un certain nombre d'informations sur le projet.
- **Outil DSDM** : programme d'assistance utilisé lors de la mise en œuvre d'une ou plusieurs techniques et de l'élaboration du système final.

22.3 Problèmes de sélection des techniques

L'identification des techniques de modélisation adaptées au développement rapide doit tenir compte d'un certain nombre d'éléments importants.

Développement rapide

L'objectif de la méthode DSDM est d'obtenir le plus tôt possible des systèmes opérationnels. C'est pourquoi les techniques utilisées ne doivent imposer une surcharge administrative qui risquerait d'alourdir inutilement le projet. Celui-ci peut en effet être ralenti lors du passage à nouvelle phase, notamment entre la phase d'analyse des besoins et la conception logique du système.

De nombreuses techniques influencent la manière dont est perçu le système au cours des différentes phases. C'est pourquoi le passage en douceur de la phase d'analyse des besoins à

la phase de conception est parfois difficile. Le développement itératif et incrémentiel propre aux projets DSDM exige que les techniques soient facilement comprises par les utilisateurs. Ce type de développement permet l'amélioration progressive du système. Toutefois, les techniques de modélisation doivent fournir une représentation suffisamment détaillée du système, pour qu'elles soient efficaces lors de la phase de construction.

Communication

L'insuffisance de communication constitue la principale source d'erreurs et d'incompréhension entre les différentes personnes impliquées dans un projet de développement. La communication est parfois difficile entre les différents participants (utilisateurs, développeurs, experts ou directeurs), chacun ayant son propre jargon. C'est pourquoi toute technique doit favoriser la communication, en incitant les développeurs à poser les bonnes questions et à fournir aux utilisateurs les moyens de contrôler si le système est conforme à leurs besoins. La technique et le modèle généré doivent être facilement compris par les utilisateurs, au moins dans les grandes lignes.

Fossé sémantique

Le fossé sémantique résulte de la différence bien naturelle de point de vue entre les utilisateurs et les développeurs. La principale préoccupation des utilisateurs est de savoir si le système proposé permet d'effectuer les tâches de l'entreprise. Quant aux membres de l'équipe de développement, ils appréhendent le système d'un point de vue plus technique et le perçoivent en termes d'applications spécifiques, de bases de données et de communications (voir figure 22.1).

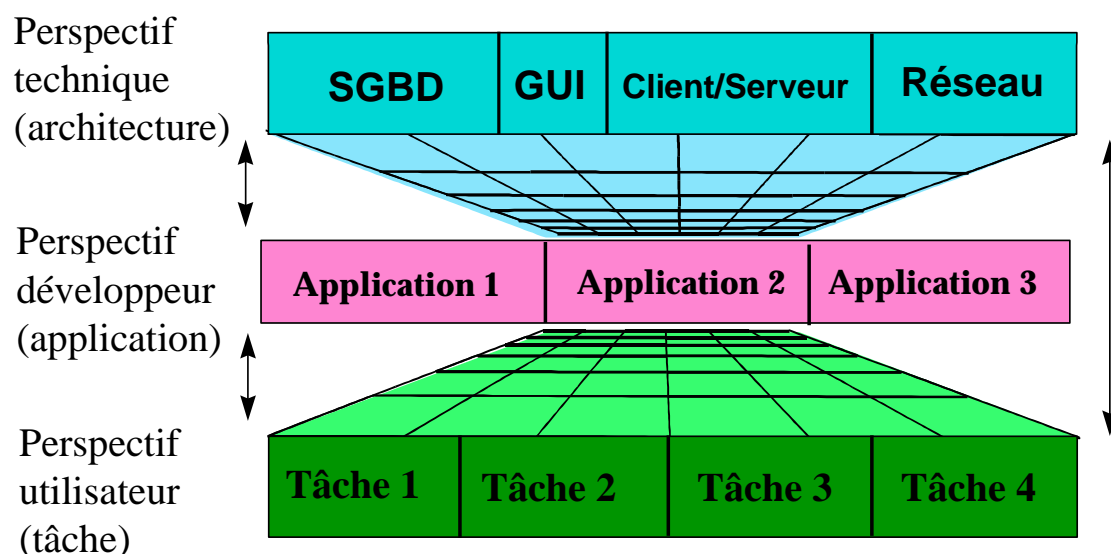


Figure 22.1 : Problèmes de communication lors du développement d'un système informatique

Cette différence de perception du système donne souvent lieu à une incompréhension et aboutit parfois à l'élaboration d'un système non conforme aux attentes de l'entreprise. L'équipe d'un projet DSDM peut combler ce fossé sémantique en adoptant une approche de développement centrée sur les utilisateurs, dont le principe est l'appréhension du système en fonction de la perspective des utilisateurs.

Une telle approche s'appuie sur les principes suivants :

- Production d'un sous-ensemble de modèles se limitant à la perspective d'une seule classe d'utilisateurs. Ces modèles peuvent ensuite être utilisés lors de la phase d'itération du modèle fonctionnel.

- Lorsque chaque perspective a été affinée au moyen d'une série de prototypes fonctionnels, un modèle fonctionnel tenant compte de toutes les perspectives individuelles peut être élaboré afin d'accroître la performance et de réduire les redondances. Cette approche technique plus détaillée doit être déployée au cours de l'itération de conception et de construction, parfois même après que certaines fonctionnalités aient été mises en œuvre.

Avant d'élaborer le modèle fonctionnel, il convient de définir les normes de développement et les grandes lignes de l'architecture, afin d'éviter le développement de composants difficiles à intégrer. Cette tâche n'est pas très longue et ses grands principes peuvent être définis au cours d'un atelier de prise de décision.

Modélisation du système

La modélisation permet à l'équipe de développement DSDM d'améliorer sa compréhension des activités de l'entreprise. Une meilleure connaissance des problèmes de l'entreprise aide à générer des modèles plus précis qui reflètent les réalités du monde de l'entreprise. L'analyse des différents points de vue contribue à une meilleure compréhension du problème. Quatre perspectives principales permettent l'élaboration de modèles de différents niveaux :

- La **perspective du traitement** modélise le système comme un ensemble de processus qui traitent les données d'entrée et génèrent des données de sortie. Les différents processus peuvent être associés pour former des processus de plus haut niveau, ou décomposés en sous-processus. Cette approche correspond aux questions «Pourquoi, Quoi et Comment» que l'on se pose lors de la phase d'analyse des besoins.
- La **perspective des données** considère les informations de l'entreprise comme un ensemble d'objets ou d'entités et modélise les relations existant entre ces objets.
- La **perspective du comportement** modélise le comportement du système en termes d'événements et d'états, chaque événement modifiant l'état du système. Les événements peuvent être générés à l'intérieur ou à l'extérieur du système.
- La **perspective de l'interface utilisateur** modélise les interfaces et les interactions entre l'utilisateur et le système.

22.4 Sélection des techniques

Les techniques les mieux appropriées au développement DSDM sont celles qui sont comprises à la fois par les utilisateurs et par les développeurs et qui favorisent la communication entre toutes les parties prenantes. D'autres techniques de modélisation peuvent être utilisées par les développeurs pour vérifier l'intégrité et la cohérence des données du système. Ces techniques ne sont pas toujours bien comprises par le personnel de l'entreprise. Par conséquent, lorsqu'elles sont requises, les membres de l'équipe de développement peuvent choisir de ne pas solliciter la participation des utilisateurs.

La meilleure approche consiste à considérer le processus de développement comme une suite d'améliorations des modèles représentant les différents aspects du système. Cette approche de «construction de modèles» est inhérente à la plupart des méthodes traditionnelles de développement. Cependant, les développeurs doivent sélectionner les modèles en fonction de leur capacité à élaborer un système doté d'un niveau satisfaisant d'intégrité et de flexibilité.

22.5 Développement orienté utilisateur

Les techniques sélectionnées dans un projet DSDM doivent se concentrer sur la modélisation du système à partir de la perspective de l'utilisateur. Cette approche, appelée «développement orienté utilisateur», modélise des concepts qui sont familiers à l'utilisateur et lui permet de mieux comprendre les techniques utilisées. Le développement orienté utilisateur s'appuie non seulement sur des techniques de description des processus internes du développement, mais également sur les techniques de développement des interfaces utilisateur, qui tiennent compte des besoins et des contraintes. Les techniques de développement des interfaces sont orientées utilisateur et basées sur des modèles plus formels, décrits ci-après :

- L'**analyse utilisateur** permet d'identifier les différentes classes d'utilisateurs finaux ainsi que leurs responsabilités, leurs attributions, leurs compétences et leurs besoins d'accès au système informatique.
- L'**analyse d'exploitation** permet de déterminer les caractéristiques de l'interface qui répondent aux besoins non fonctionnels des utilisateurs.
- La **modélisation des tâches** consiste à modéliser les différentes tâches mises en oeuvre par les utilisateurs du système.
- La **définition du scénario des tâches** décrit l'exécution d'une tâche par un utilisateur donné.
- La **modélisation conceptuelle utilisateur (Modélisation Objet Utilisateur)** génère un modèle du système à partir de la perspective de l'utilisateur. Ce modèle peut être facilement compris par les utilisateurs. Il peut être comparé à un plan de métro, qui fournit les informations nécessaires pour planifier un trajet, sans divulguer les détails techniques.
- La **conception de l'interface graphique utilisateur** permet l'exécution des tâches de l'entreprise.
- Le **prototypage de l'interface utilisateur** permet de visualiser la conception de l'interface utilisateur proposée.

Les modèles orientés utilisateur sont ensuite affinés au moyen de techniques de conception informatique plus précises pour donner les modèles de **traitement, de comportement et de données**. Ces modèles doivent être ensuite confrontés à la perspective de l'utilisateur afin de vérifier leur cohérence. Un grand nombre d'améliorations sont apportées aux modèles d'un système informatique, jusqu'à l'obtention du modèle le plus détaillé, c'est-à-dire le code. Le choix des techniques de modélisation de l'interface est indépendant de l'approche utilisée pour modéliser les données et les fonctions de base.

La liste des techniques énumérées précédemment n'est pas exhaustive, mais représente une sélection des techniques appropriées aux projets DSDM. D'autres techniques peuvent être utilisées. L'approche DSDM consiste à tirer parti de l'expérience et des connaissances acquises par les utilisateurs. En outre, seuls les éléments essentiels doivent être documentés à un niveau de détail permettant la compréhension. Enfin, le système doit être documenté au niveau du code. Les plus longs projets bénéficiant d'un faible niveau de participation des utilisateurs requièrent un volume de documentation intermédiaire plus important que les projets DSDM.

22.6 Techniques de modélisation DSDM

La modélisation du système permet aux membres de l'équipe DSDM d'acquérir une bonne connaissance de l'entreprise et du système à élaborer. En outre, elle améliore nettement la communication. La compréhension des problèmes permet l'élaboration de modèles précis qui reflètent les réalités du monde de l'entreprise. L'appréhension du système en fonction de différents points de vue permet d'accroître le niveau de compréhension.

La réponse aux questions Quoi, Comment, Où, Qui, Quand et Pourquoi permet d'avoir une vue d'ensemble du système et des besoins de l'entreprise.

Ce concept, développé par John Zachman, peut être décrit de la manière suivante :

- **QUOI** La **perspective des données** modélise les entités et les relations au sein de l'entreprise (données et relations).
- **COMMENT** La **perspective du traitement** modélise les processus de l'entreprise qui transforment les données d'entrée en données de sortie (processus et E/S), ainsi que leurs interfaces.
- **OÙ** Modélisation des sites de l'entreprise, considérés comme des noeuds et des lignes (sites et communication de réseau).

- **QUI** La **perspective de l'interface utilisateur** modélise le personnel de l'entreprise et ses attributions (utilisateurs et tâches), ainsi que les interactions entre les questions Qui, Quoi, Comment et Quand.
- **QUAND** La **perspective du comportement** modélise les événements importants de l'entreprise (délais et planification).
- **POURQUOI** La **perspective de l'entreprise** modélise les objectifs et la stratégie de l'entreprise, en fonction des différentes manières de les réaliser (raisonnement, fins et moyens).

Les interactions entre les six perspectives doivent également être modélisées. Par exemple, la perspective de l'entreprise doit correspondre à celle du traitement, afin que chaque processus soit justifié par un objectif de l'entreprise.

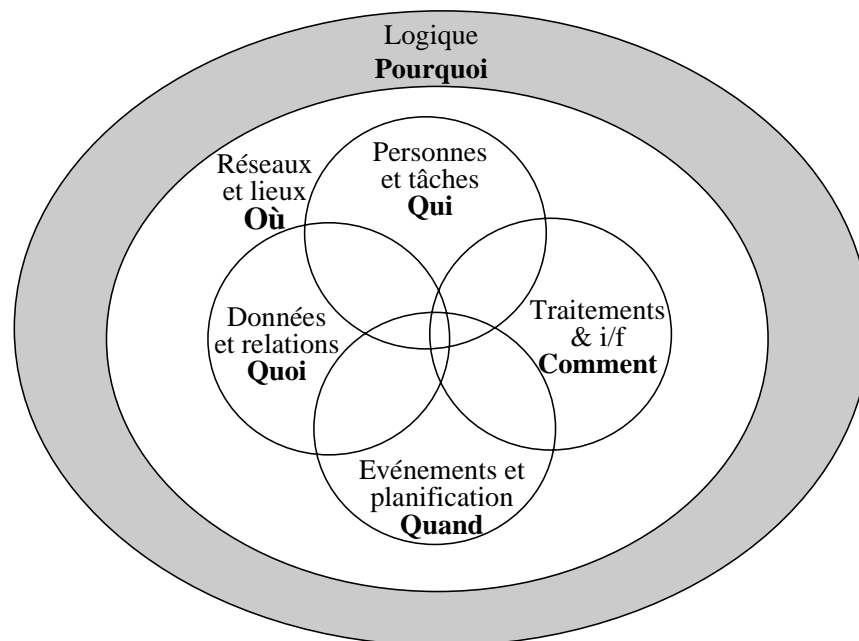


Figure 22.2 : Interactions entre les différentes perspectives de modélisation

Modélisation à partir de la perspective d'un seul utilisateur

Au lieu de modéliser l'entreprise dans son ensemble, il est préférable de modéliser les aspects dont un utilisateur a besoin pour exécuter une **tâche spécifique**. Une telle approche sera mieux perçue par les utilisateurs. En outre, elle permet à chaque utilisateur de contribuer au domaine dans lequel il a le plus d'expérience et de connaissances.

La perspective de l'utilisateur (basée sur les tâches) se superpose aux autres perspectives en répondant aux six questions pour un utilisateur donné réagissant à un événement donné. Les techniques qui tiennent compte de la perspective de l'utilisateur doivent être confortées par des techniques et des modèles plus précis, afin de refléter également les perspectives plus détaillées des données, du comportement et du traitement, nécessaires à la construction du système.

Afin d'éviter la duplication de composants identiques, il est conseillé de regrouper les perspectives individuelles des utilisateurs en un modèle unique prenant en compte l'ensemble du système. Lorsqu'un tel modèle a été créé, il doit être réutilisé le plus souvent possible au cours du développement. Il est conseillé de faire l'inventaire des composants de ce modèle (données élémentaires, processus, interfaces ou objets intégrés) afin d'encourager une approche de développement centrée sur les composants. Pour cela, les outils sont d'une aide précieuse.

Modélisation et abstraction

La modélisation implique un certain niveau d'**abstraction**. C'est pourquoi il est souvent nécessaire d'omettre un certain nombre de détails d'un modèle donné, afin de mettre en évidence un aspect particulier. Les modèles physiques représentent les questions *comment, quand, où, pourquoi, qui, quoi*, alors que les modèles logiques se concentrent uniquement sur la question *quoi* (quelles données, quels processus, quelles interactions entre eux).

Dans le cas de systèmes complexes dont les exigences non fonctionnelles constituent un risque majeur, il est nécessaire d'élaborer des modèles supplémentaires, permettant de modéliser certaines caractéristiques comme la sécurité ou la performance.

Dans le cadre d'un processus de développement DSDM, certains modèles peuvent être animés, sous la forme de divers prototypes. Certains outils permettent de générer des prototypes fonctionnels à partir de modèles, d'autres permettent de déduire les modèles à partir des prototypes. Les outils doivent être capables de maintenir la synchronie entre les modèles et les prototypes.

Modélisation du système à partir de différentes perspectives

Différents modèles peuvent être élaborés à partir de la perspective de différents «agents» ou rôles, à divers stades du cycle de vie DSDM (Tableau 22.1).

Phase du cycle de vie DSDM	Agent (rôle DSDM)	Description de la perspective
Etude de faisabilité	Planificateur (Commanditaire exécutif/Visionnaire)	Résumé exécutif destiné au planificateur et à l'investisseur, précisant la taille et la forme du système, ainsi que les conclusions de l'analyse financière.
Analyse des besoins	Propriétaire (Utilisateur Ambassadeur)	Description des interactions entre les entités et les processus de l'entreprise, en fonction de la perspective du propriétaire qui devra utiliser le système.
Itération du modèle fonctionnel	Concepteur (Développeur)	Description logique détaillée en fonction de la perspective du concepteur .
Itération de conception et de construction	Créateur (Développeur) (Coordinateur technique)	Description physique détaillée en fonction de la perspective du créateur qui doit élaborer le système. Cette description tient compte des contraintes liées aux outils, à la technologie et aux ressources.
Mise en œuvre	Utilisateur (Utilisateur ambassadeur/conseiller)	Documentation et composants fonctionnels du système final.

Tableau 22.1 : Perspectives de modélisation DSDM

Le tableau 22.1 présente des exemples de techniques de modélisation appropriées aux différentes phases du cycle de vie DSDM. Toutefois, il faut savoir que les phases d'itération du modèle fonctionnel et d'itération de conception et de construction peuvent se chevaucher ou même fusionner, selon le type d'outils utilisés pour élaborer le système.

22.7 Utilisation de modèles pour les produits DSDM

Le processus de développement DSDM permet d'identifier un certain nombre de produits devant être générés à l'issue de chaque phase (voir chapitres 11 et 13). Ces produits tiennent

compte des objectifs, des exigences techniques et des contraintes du projet. Les modèles fournissent une partie des informations techniques requises.

L'application de chaque technique de modélisation génère un modèle qui constituera une partie du produit final. Plusieurs techniques différentes peuvent être utilisées au cours d'une même phase de développement. Par conséquent, un produit DSDM peut être constitué de plusieurs modèles.

Décisions relatives à l'élaboration des modèles

Les modèles peuvent être divisés en deux principales catégories :

- **Modèles de développement**, élaborés lors du développement du système ;
- **Modèles de maintenance**, requis par les responsables de la maintenance du système ;

Par conséquent deux types de techniques de modélisation DSDM peuvent être définies :

- **Techniques de base**, qui génèrent les modèles clés nécessaires au développement et à la mise à jour du système ;
- **Techniques complémentaires**, contenant des informations d'aide nécessaires uniquement lors des phases du développement.

Les modèles générés au moyen des techniques de base doivent être inclus dans la documentation. Le tableau 22.2 présente des exemples de modèles de base.

Le choix des techniques à déployer incombe au responsable technique du projet.

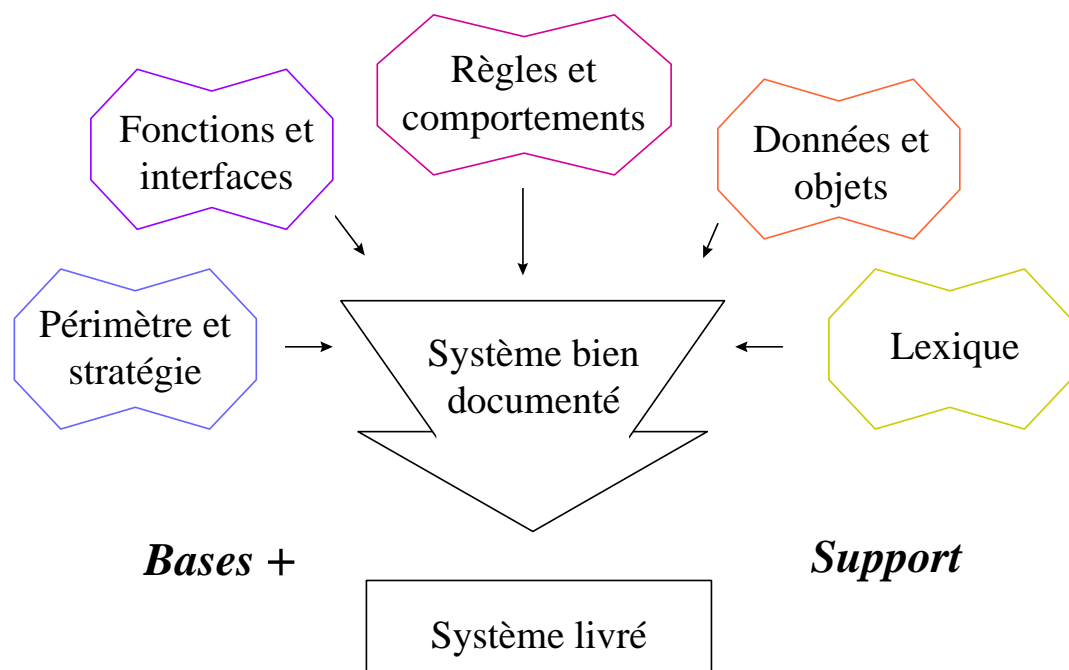


Figure 22.4 : Modèles clés requis lors du développement et de la maintenance

Il est recommandé de confronter les différents modèles de base. Par exemple, si les éléments suivants ont été identifiés,

- Facteurs majeurs de réussite (Pourquoi)
- Fonctions ou tâches (Comment)
- Données (Quoi)

Il est conseillé de confronter les tâches aux facteurs majeurs de réussite afin de planifier les cadres temporels et classer les fonctionnalités en fonction de leur niveau de priorité. La confrontation des tâches et des données permettra de mettre en évidence les dépendances entre les différentes tâches.

22.8 Développement d'applications

Le tableau 22.2 présente les produits DSDM et les modèles utilisés lors de chaque phase. Ces informations devraient permettre aux développeurs d'identifier les modèles appropriés, que l'approche soit structurée, orientée objet ou autre. En fait, de nombreux modèles peuvent être créés lors d'une phase antérieure à celle indiquée dans le tableau. Cela ne pose aucun problème, tant que le modèle atteint le niveau d'utilité escompté avant la fin de la phase mentionnée dans le tableau.

De nombreux modèles mentionnés dans le tableau ont un rôle identique. Il est donc inutile de les développer tous. Il convient de sélectionner soigneusement les techniques, afin de générer les produits nécessaires au projet et d'atteindre les objectifs définis pour chaque phase.

Les techniques spécifiques indiquées dans le tableau sont sélectionnées en fonction des facteurs suivants :

- type de système (système traditionnel, développement Internet/Intranet),
- environnement de développement,
- compétence et expérience de l'équipe de développement.

Par exemple, si vous optez pour une approche orientée objet (O-O), vous devrez sélectionner des outils et des techniques de modélisation O-O. Quelle que soit l'approche adoptée, toutes les rubriques du tableau doivent être associées à une technique de modélisation. Ces techniques jouent un rôle fondamental dans l'élaboration d'un système conforme aux exigences de l'entreprise.

Phase DSDM	Produits DSDM	Objectifs de modélisation (Quoi, Comment, Où, Qui, Quand, Pourquoi)	Exemples de modèles	Exemples d'interactions appropriées
Etude de faisabilité	Rapport de faisabilité Prototype de faisabilité Plan préliminaire	Domaine d'activité et modèle de l'entreprise Données clés de l'entreprise Activités clés de l'entreprise Sites clés de l'entreprise Personnes/Utilisateurs clés Événements clés Optique/Domaine d'activité/Objectifs de l'entreprise Interfaces clés	Modèles de base : Facteurs majeurs de réussite Diagramme du contexte Modèles complémentaires : Riche illustration Hiérarchie des fonctions Plan de l'architecture réseau Diagramme du flux de travail Organigramme hiérarchique	Entité/Organisation Entité/Processus Processus/Site Processus/Organisation Événement important/Site Site/Rôle Objectif/Responsabilité

Tableau 22.2 (première partie) : Modèles et produits DSDM

Phase DSDM	Produits DSDM	Objectifs de modélisation (Quoi, Comment, Où, Qui, Quand, Pourquoi)	Exemples de modèles	Exemples d'interactions appropriées
Analyse des besoins	Définition du domaine d'application Plan de prototypage Définition de l'architecture du système	Modèle de l'entreprise Modèle du système de haut niveau Fonctionnement de l'entreprise : Données/Relations/Règles Événements de l'entreprise Scénarios de l'entreprise Architecture de l'entreprise Localisation du système Utilisateurs du système	Modèles de base : Modèle des relations entre entités (haut niveau) Modèle des processus de l'entreprise Diagrammes de flux de données de haut niveau Facteurs majeurs de réussite Modèle objet de l'entreprise Cas d'utilisation Modèle de l'architecture technique Modèles complémentaires : Fonctions dépendantes Scénarios de l'entreprise Modèles de tâches Modèle des événements de l'entreprise	Processus/Entité Processus/Site Lieu des événements Rôle des personnes/Rôle du système CSF/Processus Tâche/Objet

Tableau 22.2 (seconde partie) : Modèles et produits DSDM

Phase DSDM	Produits DSDM	Objectifs de modélisation (Quoi, Comment, Où, Qui, Quand, Pourquoi)	Exemples de modèles	Exemples d'interactions appropriées
Itération du modèle fonctionnel	Modèle fonctionnel Stratégie de mise en oeuvre Rapport d'analyse des risques	Modèle du système Prototypes fonctionnels Spécifications (fonctionnelles et non fonctionnelles)	Modèles de base : Modèle logique des données Diagrammes du flux de données Cas d'utilisation Ecrans/Menus Modèle objet Modèles complémentaires : Processus dépendants Analyse des scénarios Interaction entre les objets/Diagrammes de collaboration Modèle conceptuel utilisateur Modèle objet de l'interface utilisateur Conception du système de navigation dans les écrans Modèle de la topologie du réseau Evénements du système Diagramme des états du système Modèles dynamiques objets	Processus/Entité Rôle de l'utilisateur/Fonction

Tableau 22.2 (troisième partie) : Modèles et produits DSDM

Phase DSDM	Produits DSDM	Objectifs de modélisation (Quoi, Comment, Où, Qui, Quand, Pourquoi)	Exemples de modèles	Exemples d'interactions appropriées
Itération de conception et de construction	Système testé intégrant toutes les spécifications fonctionnelles et non fonctionnelles Prototypes	Modèle de la technologie Modèle des composants Système testé : Ecrans/Menus/Rapports Utilisateurs et sites du système Stratégie de la technologie	Modèles de base : Modèle physique des données Modèle physique des processus Plan de répartition objet Modèle de la topologie du réseau Code Modèles complémentaires : Modèles détaillés des processus utilisateur	Rôles du système/utilisateurs Événements du système/Événements de l'entreprise Application/objet Plate-forme/Application Utilisateur/Données d'application Structure des données/Application Structure des données/Stockage Entité/ Structure des données

Tableau 22.2 (quatrième partie) : Modèles et produits DSDM

Phase DSDM	Produits DSDM	Objectifs de modélisation (Quoi, Comment, Où, Qui, Quand, Pourquoi)	Exemples de modèles	Exemples d'interactions appropriées
Mise en oeuvre	Documentation utilisateur Système livré par les utilisateurs formés Rapport de revue de projet	Système opérationnel, testé et documenté Documentation/Information d'aide)	Modèles de base : Structure des composants physiques Information d'aide Utilisateur/Description des tâches opérationnelles Modèles complémentaires : Définition des composants physiques Architecture de la sécurité	Fonction/Contrôle d'accès Utilisateur/Fonction

Tableau 22.2 (cinquième partie) : Modèles et produits DSDM

22.9 Résumé

Il existe un large choix de techniques pouvant être appliquées dans un projet DSDM. Chaque phase du projet requiert une technique spécifique. De plus, certaines techniques sont mieux appropriées à certains processus de développement. Les techniques adoptées dans un projet DSDM doivent être suffisamment flexibles pour satisfaire la diversité des intérêts en jeu.

Il est recommandé d'adopter une approche orientée utilisateur, basée sur les tâches qui doivent être effectuées. Une telle approche doit être étayée par une bonne coordination des équipes de développement, le prototypage et les ateliers de prise de décisions.

L'approche orientée utilisateur permet l'élaboration des modèles suivants :

- moyens de communication efficaces entre tous les participants ;
- manière de comprendre l'essence d'un problème, afin de le transposer sous une forme différente sans perte d'information ;
- moyen d'obtenir une représentation abstraite en occultant les détails inutiles ;
- mécanisme efficace permettant de définir le domaine d'activité de l'entreprise.

23 Ateliers de prise de décisions

23.1 Utilisation des ateliers de prise de décisions

Les ateliers de prise de décisions constituent une technique flexible pouvant s'appliquer à divers secteurs de l'entreprise. Ils sont organisés lorsque des décisions techniques ou stratégiques doivent être prises et réunissent essentiellement des membres de l'entreprise. Toutefois, les partenaires et les fournisseurs sont parfois conviés à ces réunions. Les ateliers de prise de décisions favorisent la livraison de produits de qualité dans les meilleurs délais.

Ils peuvent être organisés pour débattre des questions suivantes :

- L'**analyse de l'optique de l'entreprise** consiste à déterminer et développer l'optique et la mission de l'entreprise, afin de définir les grandes lignes de sa stratégie future. Les objectifs des différents secteurs de l'entreprise peuvent ensuite être définis.
- La **planification du système informatique** permet d'envisager et de planifier le développement du système d'information de l'entreprise, en fonction de ses besoins et de ses objectifs.
- La **conception des processus de l'entreprise** consiste à identifier, concevoir et prototyper les processus qui permettent de réaliser les objectifs de l'entreprise.
- L'**analyse des bénéfices du système d'information** permet d'envisager la meilleure manière d'utiliser les ressources financières de l'entreprise au profit du système d'information.
- La **définition des spécifications du système d'information** consiste à déterminer et valider la liste des fonctions dont le système d'information de l'entreprise sera doté.
- La **conception du système d'information** permet de définir les spécifications du système.
- Les **options techniques du système** permettent :
 1. de déterminer l'option technique qui répond le mieux à certains besoins de l'entreprise,
 2. d'effectuer rapidement la revue d'un projet qui nécessite un redressement.
- La **planification du test d'acceptation** permet de définir les critères d'acceptation du système. Ce type d'atelier doit être organisé tout au long du projet DSDM, à l'issue de chaque incrémentation.

Toutes les décisions importantes d'un projet DSDM doivent être prises lors d'un atelier de prise de décisions. Cette technique est plus connue dans le milieu informatique sous le terme de Développement Applicatif Joint (DAJ). Certaines méthodes de développement rapide différencient, au sein du DAJ, «l'analyse des besoins jointe» et la «conception applicative jointe». Les techniques utilisées sont identiques quelle que soit la phase du cycle de vie.

Le tableau 23.1 indique les différents types d'ateliers de prise de décisions pouvant être appliqués à chaque phase DSDM.

Phase DSDM	Type d'atelier de prise de décisions
Etude de faisabilité	Définition des spécifications du système d'information Bénéfices du système d'information de l'entreprise Options techniques du système
Analyse des besoins	Définition des spécifications du système d'information Bénéfices du système d'information de l'entreprise Conception des processus de l'entreprise Classement des spécifications en fonction de leur priorité
Itération du modèle fonctionnel	Définition des spécifications du système d'information (comprenant l'attribution de priorités) Planification du test d'acceptation
Itération de conception et de construction	Conception du système d'information Options techniques du système
Mise en œuvre	

Tableau 23.1 : Utilisation des ateliers de prise de décisions dans un projet DSDM

23.2 Processus de l'atelier de prise de décisions

La définition, la préparation, le déroulement et les conclusions d'un atelier de prise de décisions peuvent être considérés comme un sous-processus au sein du projet global. Les objectifs relatifs à la durée et au budget sont fixés par le responsable de l'atelier (voir ci-après). La figure 23.1 illustre les phases d'activité et les objectifs spécifiques d'un atelier de prise de décisions.

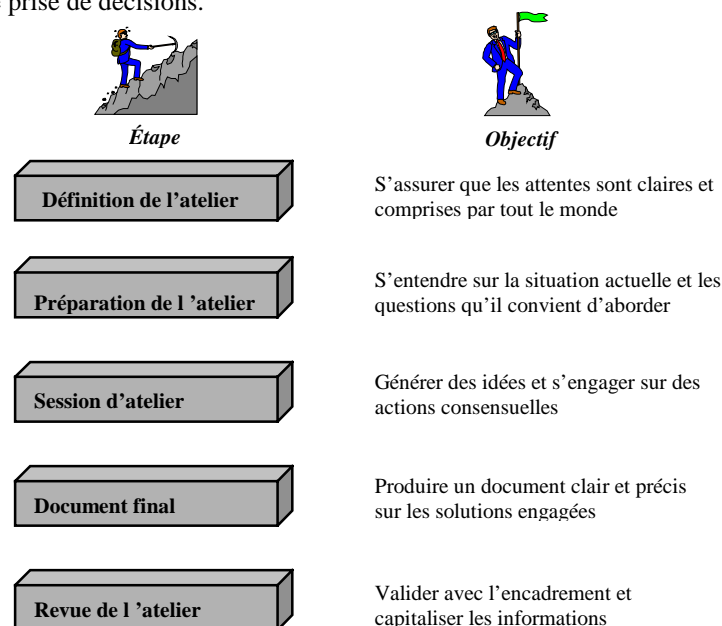


Figure 23.1 : Processus d'un atelier de prise de décisions

23.3 Rôles et responsabilités des intervenants

Responsable de l'atelier

Le responsable de l'atelier est souvent le commanditaire exécutif qui finance le projet. Il doit être recruté parmi les directeurs ou les cadres supérieurs de l'entreprise et doit représenter tous les utilisateurs ayant accès au système d'information. Il ou elle doit être investi d'un pouvoir de décision et doit être habilité à approuver les dépenses financières tout au long du projet. Le responsable de l'atelier est également chargé de résoudre les conflits entre les différents participants.

Le responsable de l'atelier de prise de décisions doit au moins assister au début et à la fin des sessions clés, afin de lancer le débat et recueillir les conclusions. S'il ne peut pas être présent, il doit désigner une personne chargée de le représenter.

Organisateur de l'atelier

En l'absence du commanditaire exécutif, l'organisateur de l'atelier est chargé de définir les objectifs de la séance. Il travaille en collaboration avec le médiateur pour recentrer le débat lorsque certaines questions sont plus longuement discutées. L'organisateur de l'atelier doit identifier les participants afin de s'assurer que tous les intéressés sont représentés. Le visionnaire ou le commanditaire exécutif doit arbitrer les conflits et définir les priorités. Si aucun d'entre eux n'est présent, c'est à l'organisateur de l'atelier qu'incombe ces responsabilités. Ce dernier, dont la présence est indispensable tout au long de l'atelier, est souvent le chef de projet.

Médiateur

L'expérience prouve qu'un médiateur compétent représente la clé du succès d'un atelier de prise de décisions. Il doit appartenir à un secteur de l'entreprise différent de celui du projet, afin de garantir son impartialité. Son rôle est d'assurer le bon déroulement de l'atelier, et de vérifier que tous les participants ont les mêmes chances de s'exprimer. Il doit gérer les aspects humains de la session et faire en sorte qu'un travail d'équipe soit réalisé. Le médiateur est chargé de consigner les résultats et les décisions sur un tableau à feuilles mobiles, qui matérialise la «mémoire du groupe». Comme son nom l'indique, il a un rôle de médiation et ne contribue pas au contenu des débats. Les responsabilités des différents intervenants sont décrites au chapitre 15.

Participants

Les participants ont une excellente connaissance du projet et représentent les différentes catégories de personnel impliquées (personnel de l'entreprise et membres de l'équipe de développement). Ils gèrent et exploitent le système. Ils comprennent des membres de la direction, des représentants du personnel administratif ainsi que les membres de l'équipe de développement.

Les participants doivent être animés par une motivation commune et faire preuve d'autorité et de responsabilité. Ils doivent être habilités à accepter une décision sans avoir besoin d'en référer à un supérieur hiérarchique.

Enfin, tous les participants doivent adhérer au concept de l'atelier de prise de décisions et s'y préparer.

Rapporteurs

Un rapporteur *technique* est chargé de saisir les résultats produits lors de la séance (par exemple, les spécifications du système). Pour cela, il peut avoir recours à un outil informatique approprié. Le médiateur est chargé de lui expliquer les modèles techniques qu'il doit produire. Le rapporteur *technique* doit être capable d'imprimer les différentes parties du modèle afin qu'elles puissent être révisées à tout moment.

Un rapporteur *administratif* produit un compte-rendu des questions discutées lors de l'atelier. Il doit avoir une bonne connaissance du projet, afin d'être familiarisé avec la terminologie, les problèmes et l'environnement du projet.

Le type et le nombre de rapporteurs nécessaires sont décidés en fonction de la nature de l'atelier.

Observateurs

La présence d'observateurs doit être évitée. Etant donné qu'ils ne participent pas aux débats, leur présence peut causer des perturbations et entraver le travail d'équipe.

Si néanmoins la présence d'observateurs est inévitable, il est conseillé de les placer hors du champ de vision des participants et de bien leur faire comprendre que leur rôle est d'observer et non de participer. Toutefois, la présence d'un observateur peut se justifier lorsqu'il s'agit d'un médiateur stagiaire qui étudie la dynamique de groupe de l'atelier.

Prototypeurs (si nécessaire)

Ce rôle est réservé à des techniciens hautement qualifiés, capables de créer des écrans très rapidement. Un ordinateur et des logiciels appropriés doivent être mis à leur disposition. La présence d'un prototypeur est justifiée lorsqu'un prototypage d'écrans est prévu au cours de l'atelier.

Le rôle des prototypeurs doit être défini avec le médiateur. Si les écrans à produire sont nombreux et complexes, il est conseillé de les créer pendant la nuit. Dans ce cas, le prototypeur assiste à l'atelier afin d'identifier la conception des écrans, développe les prototypes durant la nuit puis les présente le lendemain. Il est alors nécessaire de recourir aux services de plusieurs prototypeurs.

24 Environnements de soutien DSDM

24.1 Introduction

Au cours du développement, le volume et la complexité des informations générées augmentent rapidement. Les outils de développement permettent d'automatiser totalement ou partiellement certaines tâches, quelle que soit la technique utilisée. Par exemple, il existe un grand nombre d'outils qui facilitent les activités de modélisation décrites au chapitre 22. Toutefois, les outils d'aide ne se limitent pas aux programmes d'analyse et de conception, plus connus sous le nom d'outils de GLAO (Génie Logiciel assisté par Ordinateur). Ils comprennent également des «débugueurs», des générateurs de code et des programmes de test.

Le développement rapide d'applications (RAD) requiert un excellent environnement d'aide. Un tel environnement permet de tirer le meilleur profit de la méthode DSDM. Certains vont même jusqu'à dire qu'aucune approche de développement rapide n'est efficace sans un solide environnement d'aide.

Le premier problème d'un acheteur potentiel est le choix des outils informatiques d'aide les mieux appropriés. Les outils requis dans le cadre d'un développement rapide sont à peu près similaires à ceux qui sont utilisés dans les projets traditionnels. Toutefois, un projet DSDM a des besoins d'automatisation plus spécifiques.

L'objet de ce chapitre est d'identifier les principales caractéristiques des environnements d'aide des projets DSDM. Un environnement d'aide unique ne satisfait pas à lui seul tous les besoins. Il est donc nécessaire de recourir à plusieurs outils au sein d'un même projet. Il convient néanmoins de dresser la liste des besoins, dans l'espoir qu'un environnement d'aide unique soit un jour disponible.

La figure 24.1 représente un environnement d'aide DSDM idéal. Les outils «verticaux» sont spécifiques à chaque phase du projet DSDM et permettent, dans certains cas, la reprise de la codification existante. Les outils «horizontaux» sont déployés tout au long du processus de développement et certains d'entre eux intègrent des fonctions de gestion de configuration et de contrôle de processus.

Un tel environnement permet l'intégration à différents niveaux :

- **Données**, lorsque plusieurs outils partagent le même répertoire de données ;
- **Contrôle**, lorsqu'un outil peut notifier et/ou déclencher des actions dans d'autres outils ;
- **Plate-forme**, lorsque les outils peuvent être portés sur un certain nombre de plates-formes différentes.

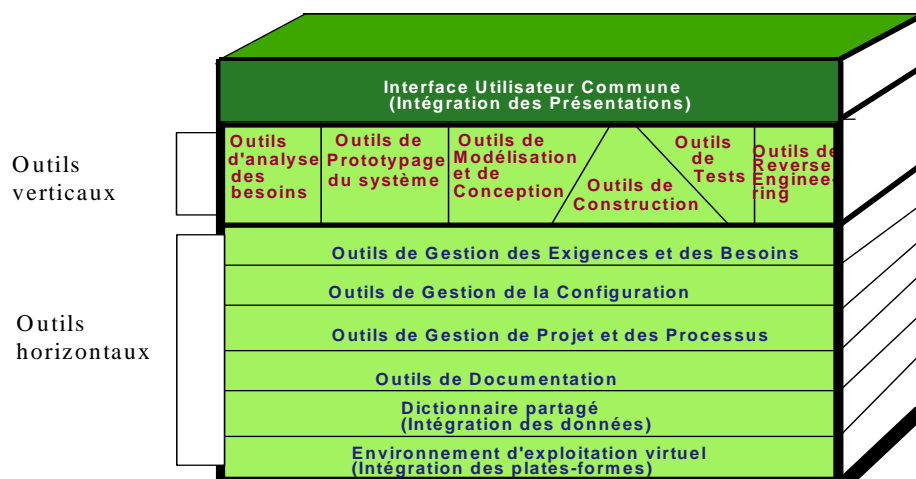


Figure 24.1 : Environnement d'aide DSDM idéal

Enfin, il convient de mettre en garde les acheteurs novices contre les revendeurs imposteurs qui mettent leur étiquette sur des outils existants, et les présentent comme de nouveaux outils de développement rapide. Les fonctionnalités de ces outils sont très limitées. Bien souvent ils permettent uniquement le «développement rapide de programmes», qui n'est qu'un aspect du «développement rapide d'applications». Ces revendeurs sans scrupule qui assimilent ces produits aux outils de développement rapide portent non seulement atteinte à ceux qui leur font confiance mais également à la réputation des méthodes de développement rapide.

Avant de sélectionner un environnement d'aide pour un projet DSDM, il convient de choisir un processus de développement puis d'identifier les outils d'aide appropriés à l'approche adoptée. De nombreuses entreprises font l'erreur d'acheter les outils d'aide avant d'avoir identifié leur processus de développement.

L'expérience des projets DSDM montre que l'utilisation de certains outils bon marché permet d'accroître considérablement la qualité et la productivité. Ces outils permettent d'automatiser les tâches suivantes :

- Génération du code ;
- Génération de prototypes ;
- Animation ;
- Génération de documents automatisée.

24.2 Sélection des outils de développement rapide

Avant d'acquérir un outil d'aide pour un projet DSDM, il convient de définir clairement les besoins, afin de sélectionner les outils les mieux appropriés.

L'approche recommandée de sélection des outils RAD s'appuie sur les mêmes principes que la méthode DSDM.

Les différentes étapes du processus de sélection des outils RAD sont les suivantes :

- **Etude de faisabilité** : identification des étapes du processus de développement et des techniques qui peuvent être automatisées ;
- **Analyse des besoins** : identification des risques majeurs liés à l'utilisation d'outils RAD (par exemple, les membres de l'équipe de développement possèdent-ils les compétences requises pour l'utilisation de ces outils ?) ;
- **Itération du modèle fonctionnel** : analyse détaillée des besoins afin de sélectionner les outils RAD ; mise en œuvre de prototypes à l'aide de copies d'évaluation des outils ;
- **Itération de conception et de construction** : évaluation détaillée et sélection des fournisseurs ;
- **Mise en œuvre** : déploiement des outils sélectionnés, formation des utilisateurs des outils et contrôle de leur utilisation ;

La suite du chapitre décrit les questions à traiter au cours de ces différentes étapes.

24.2.1 Etude de faisabilité - Choix des tâches à automatiser

Un certain nombre de tâches peuvent être automatisées lors de chaque phase du projet DSDM. Le principal critère de sélection des outils d'aide consiste à évaluer les bénéfices produits par cet outil, en fonction de l'investissement financier. En d'autres termes, il s'agit de savoir si la dépense occasionnée par l'achat de cet outil sera rentabilisée.

L'identification et la sélection des outils «rentables» doivent se faire au début du projet. Le tableau 24.1 dresse la liste des outils qui peuvent être utilisés dans un projet DSDM.

- | | |
|---|---------------------------------|
| • Outil d'estimation | • GLAO |
| • Outil de planification | • Générateur de rapports |
| • Traitement de texte | • Générateur d'écrans |
| • Dictionnaire de données | • Générateur de programmes |
| • Analyseur de performance | • Compilateur, Editeur de liens |
| • Outil d'aide au prototypage | • Batterie de tests |
| • Utilitaires de reprise et de sécurité | • Générateur de données de test |
| • Bibliothèque de développement | • Testeur/Débogueur interactifs |

Tableau 24.1 : Outils appropriés au développement rapide d'applications

24.2.2 Analyse des besoins - Identification des risques et des contraintes

L'utilisation d'un nouvel outil modifie inévitablement les pratiques et les procédures de travail de l'entreprise et introduit un certain nombre de risques.

Ces derniers doivent être identifiés dès le début et gérés tout au long du projet.

Les principaux risques résultant de l'utilisation d'outils RAD sont les suivants :

- **Motivation** : l'entreprise est-elle prête à relever le défi et accepter les changements inhérents à l'introduction d'un nouvel outil ?
- **Investissement** : l'entreprise est-elle prête à assumer la dépense financière liée à l'acquisition d'un nouvel outil, sachant que le coût réel ne se limite pas à la dépense initiale mais inclut également le coût des mises au point et des adaptations nécessaires ?
- **Définition des processus de développement** : l'entreprise a-t-elle défini un processus de développement, qui justifie l'acquisition d'un nouvel outil ?
- **Compétences** : l'entreprise possède-t-elle le niveau de compétence requis pour s'adapter à de nouveaux outils ?
- **Barrières culturelles** : Y a-t-il des désaccords au sein de l'entreprise pouvant entraver l'introduction d'un nouvel outil ?
- **Plate-forme technique** : l'environnement technique de l'entreprise est-il compatible avec le nouvel outil ?
- **Support technique** : le service d'assistance technique de l'entreprise peut-il gérer l'utilisation d'un nouvel outil ?
- **Architecture technique** : l'infrastructure technique actuelle peut-elle prendre en charge le nouvel outil ?

24.2.3 Itération du modèle fonctionnel - Définition des exigences de l'environnement d'aide DSDM

Le premier obstacle rencontré lors de la sélection d'un environnement d'aide est la grande diversité des outils. En raison des contraintes évidentes de temps, il est impossible d'effectuer une évaluation détaillée de tous les outils, afin de sélectionner les meilleurs.

Cette section dresse la liste des exigences d'un environnement DSDM idéal. Etant donné qu'aucun outil ou ensemble d'outil ne peut satisfaire à lui seul toutes les exigences, cette liste représente les exigences optimales, parmi lesquelles l'entreprise doit sélectionner celles qui lui sont indispensables.

SE.1 Facilitation du processus de développement

Exigence

L'environnement d'aide doit prendre en charge le processus de développement DSDM.

Commentaire

Tout environnement d'aide doit apporter de la valeur au processus de développement. Par exemple, les informations générées au cours de chaque phase doivent être facilement accessibles lors des phases suivantes. Beaucoup d'énergie peut être déployée inutilement lorsque l'environnement d'aide entrave le processus de développement.

SE.2 Prise en charge des techniques de développement DSDM**Exigence**

L'environnement d'aide doit prendre en charge les techniques de modélisation sélectionnées pour le projet DSDM.

Commentaire

Les techniques de modélisation appliquées dans un projet DSDM sont nombreuses et dépendent de la nature de l'approche de développement adoptée (développement structuré ou orienté objet).

Les divers modèles développés sont interdépendants. L'environnement d'aide doit prendre en charge ces interrelations.

SE.3 Facilité d'utilisation**Exigence**

L'interface utilisateur de l'environnement d'aide doit satisfaire les besoins des développeurs.

Commentaire

La qualité de l'interface utilisateur d'un outil d'aide facilite l'utilisation de cet outil et crée un environnement de productivité. Une bonne interface donne envie d'utiliser l'outil. En revanche, il est prouvé que les outils dotés d'une interface utilisateur médiocre sont rapidement délaissés.

La facilité d'utilisation d'un outil présente plusieurs aspects :

- facilité d'apprentissage ;
- facilité de mémorisation du fonctionnement de l'outil ;
- satisfaction de l'utilisateur ;
- gain de productivité ;
- capacité de l'outil à réduire le taux d'erreurs induites par l'utilisateur.

SE.4 Compatibilité**Exigence**

Un environnement d'aide DSDM doit être capable d'importer et d'exporter des données vers d'autres outils d'aide.

Commentaire

Un environnement d'aide unique ne peut à lui seul satisfaire toutes les exigences DSDM. C'est pourquoi il est nécessaire de recourir à plusieurs outils adaptés aux différentes phases du processus de développement.

Les informations stockées dans le répertoire d'un outil doivent être accessibles à partir des autres outils. C'est pourquoi les outils d'un environnement d'aide DSDM doivent être compatibles.

Un certain nombre de techniques permettent désormais le transfert des données entre les différents outils. La technologie OLE (Object Linking and Embedding) est la norme la plus utilisée dans l'environnement PC. Le format CDIF (CASE Data Interchange Format) est utilisé dans les autres environnements.

SE.5 Participation des utilisateurs

Exigence

Tout environnement d'aide DSDM doit permettre la participation des utilisateurs.

Commentaire

L'implication des utilisateurs est un principe fondamental de l'approche DSDM. L'environnement d'aide DSDM ne doit pas entraver la participation active des utilisateurs au processus de développement ni leur intégration dans l'équipe.

SE.6 Développement itératif

Exigence

L'environnement d'aide DSDM doit prendre en charge le développement itératif.

Commentaire

Le développement itératif ou incrémentiel est inhérent au processus de développement DSDM. Chaque incrément subit un certain nombre d'itérations avant qu'il ne soit finalement validé. Une itération permet l'ajout d'une fonction ou l'amélioration d'une fonction existante.

L'environnement d'aide DSDM doit prendre en charge le développement itératif. Si, à l'issue de la dernière itération, la nouvelle version est inacceptable, un retour à l'état précédent doit être possible.

SE.7 Contrôle de version et gestion de configuration

Exigence

La gestion de configuration et le contrôle de version sont des aspects fondamentaux de l'environnement d'aide DSDM.

Commentaire

Un projet de développement DSDM est un processus très complexe, au cours duquel plusieurs domaines de fonctionnalité sont développés. Chacun de ces domaines comprend plusieurs cycles d'itération. Enfin, plusieurs versions du système final sont parfois installées sur des systèmes différents, dans des lieux éloignés géographiquement.

Le processus de développement DSDM génère de nouveaux produits ou de nouvelles versions de produits existants beaucoup plus rapidement que les projets traditionnels. C'est pourquoi les produits doivent être fréquemment contrôlés.

En raison de la nature dynamique du processus de développement DSDM, une bonne gestion de configuration est indispensable. Lorsque cela est possible, les tâches de gestion de configuration doivent être automatisées.

SE.8 Contrôle des données**Exigence**

Les données générées dans un environnement d'aide DSDM doivent être régulièrement contrôlées.

Commentaire

Le nombre des produits et de leurs différentes versions est considérable dans un projet de développement DSDM. Tous les développeurs n'ont pas les mêmes droits d'accès aux informations. Par conséquent, la gestion et le contrôle des données doivent être très stricts et de préférence automatisés.

L'utilisation d'une base de données ouverte facilite considérablement l'importation et l'exportation des données entre les différents environnements d'aide DSDM.

SE.9 Réutilisation**Exigence**

Les composants développés lors d'un projet DSDM doivent pouvoir être réutilisés lors des projets suivants.

Commentaire

Le principe de base de la méthode DSDM est le développement, dans un cadre de temps limité, de produits conformes aux exigences de l'entreprise. La réutilisation de composants développés lors de précédents projets ou lors d'une phase antérieure du projet en cours permet de réaliser plus facilement cet objectif.

La réutilisation de composants existants peut réduire considérablement la durée du processus de développement.

SE.10 Navigation**Exigence**

L'environnement d'aide DSDM doit être doté d'un système de navigation.

Commentaire

Les développeurs qui utilisent l'environnement d'aide DSDM ont des responsabilités différentes. Les utilisateurs occasionnels doivent pouvoir consulter facilement la base de données et de naviguer d'une zone de conception à l'autre. Les utilisateurs confirmés doivent avoir les moyens de rechercher et d'identifier les composants réutilisables.

SE.11 Documentation**Exigence**

L'environnement d'aide DSDM doit permettre de produire la documentation nécessaire.

Commentaire

La documentation d'un projet DSDM est toute aussi importante que celle d'un projet traditionnel. Néanmoins, le processus DSDM étant soumis à des contraintes de temps très strictes, la documentation du système doit être générée plus rapidement.

L'environnement d'aide DSDM idéal génère automatiquement la documentation de toutes les phases DSDM, de L'étude de faisabilité à la codification.

SE.12 Mise en service du système final

Exigence

L'environnement d'aide DSDM doit être capable de produire le système final.

Commentaire

La livraison évolutive ou incrémentielle est inhérente au processus de développement DSDM. A mesure que le nombre d'incréments augmente, les problèmes d'intégration des nouveaux produits se multiplient.

SE.13 Environnement multi-utilisateur**Exigence**

Plusieurs développeurs doivent pouvoir travailler simultanément dans l'environnement d'aide DSDM.

Commentaire

La taille de l'équipe de développement augmente avec la taille du système en cours de développement. En outre, plusieurs équipes de développement peuvent travailler en même temps sur le projet. Il est par conséquent indispensable que l'environnement d'aide DSDM soit multi-utilisateur.

SE.14 Normalisation**Exigence**

L'environnement d'aide DSDM doit être conforme aux normes en vigueur.

Commentaire

La normalisation est fondamentale pour l'utilisation, la compatibilité et la portabilité de l'environnement d'aide DSDM. Un certain nombre de normes sont désormais reconnues.

WindowsTM et MotifTM sont des normes d'interface utilisateur pour les plates-formes UNIX et compatibles IBM. Les deux principales normes d'environnement d'aide sont PCTE (Portable Common Tools Environment) et CDIF (CASE Data Interchange Format).

SE.15 Tests**Exigence**

L'environnement d'aide DSDM doit permettre de tester tous les produits.

Commentaire

Des tests doivent être effectués tout au long du cycle de vie du projet. Les tests statiques et dynamiques ainsi que les tests de régression peuvent être automatisés. Les contrôles de cohérence entre les différents produits incombent aux utilisateurs.

24.2.4 Itération de conception et de construction - Evaluation et sélection des outils

Les besoins en automatisation varient d'un projet à l'autre. Néanmoins, cette section présente les principales questions à traiter lors de la sélection d'un environnement d'aide.

Sélection de l'environnement d'aide DSDM

L'environnement d'aide doit être personnalisé en fonction des besoins spécifiques du projet et de l'entreprise. A titre indicatif, voici les principaux critères de sélection pouvant être retenus :

- Le **fournisseur** doit être bien établi et ses représentants doivent être facilement accessibles. Afin de faire le meilleur choix, il convient de se poser les questions suivantes :
 - Où l'outil a-t-il été développé ?
 - Où se trouve le distributeur de l'outil ?
 - Quels sont les risques de retard de la livraison ?
 - Une copie d'évaluation de l'outil est-elle disponible ?
 - Le fournisseur propose-t-il un service d'assistance technique ?
 - Existe-t-il des références d'autres installations ?
- L'**environnement hôte** dans lequel l'outil est intégré doit être considéré attentivement :
 - L'outil fonctionne-t-il sur différentes plates-formes ?
 - L'outil peut-il être configuré pour fonctionner en réseau ?
 - L'outil est-il multi-utilisateur et permet-il l'accès concurrent aux données ?
 - La plate-forme hôte est-elle compatible avec la stratégie informatique de l'entreprise ?
 - L'outil requiert-il un environnement matériel spécifique ? Si ce n'est pas le cas, sera-t-il nécessaire d'acquérir du matériel supplémentaire ?
- Quelle **aide** l'outil apporte-t-il à l'utilisateur ?
 - L'interface utilisateur de l'outil est-elle compatible avec le style de l'interface «maison» ?
 - L'outil est-il facile à utiliser et présente-t-il les informations clairement ?
 - Quels sont les temps de réponse dans des conditions de charge importante ?
 - Quelle est la qualité du support de formation ? L'outil intègre-t-il un système d'aide en ligne ?
 - Quelle sera la courbe d'apprentissage de cet outil ? La formation du fournisseur sera-t-elle suffisante ?
 - Quelle est la qualité du manuel d'utilisation ? La documentation est-elle suffisante ?
 - Existe-t-il un compte-rendu des erreurs décelées dans cet outil ?
- Le **coût global** de l'outil, comprenant la dépense initiale ainsi que les coûts de formation et de maintenance, entre-t-il dans le budget alloué aux outils d'aide ?
- **Couverture du cycle de vie.** L'outil permet-il de traiter toutes les phases du cycle de vie DSDM ? S'il n'offre pas une couverture totale, l'achat d'outils complémentaires doit être envisagé.
- **Techniques couvertes.** De quelles manières l'outil prend-il en charge les techniques de développement DSDM ?
 - Diagrammes de flux de données
 - Diagrammes relationnels des entités
 - Diagrammes de transition d'états
 - Diagrammes de la structure de Jackson
 - Dialogues utilisateur

- Orientation objet
- Autres techniques sélectionnées pour le projet
- **Intégration des techniques :** les techniques sont-elles intégrées ? C'est-à-dire, est-il nécessaire de saisir à nouveau les données générées au moyen d'une technique afin qu'elles puissent être utilisées par une autre technique ? Si des modifications sont apportées à un modèle, les autres modèles sont-ils mis à jour automatiquement ?
- **Navigation**
 - L'outil permet-il à l'utilisateur de naviguer entre plusieurs modèles de développement ?
 - L'outil permet-il de consulter la base de données des composants réutilisables ?
- **Contrôle de cohérence**
 - L'outil assure-t-il la cohérence des données ou bien est-il doté d'un module de vérification de la cohérence des données ?
 - Quels contrôles de cohérence l'outil prend-il en charge ?
- **Architecture ouverte**
 - Les développeurs ont-ils accès à la structure de la base de données, afin de préparer les documents et d'effectuer des contrôles supplémentaires ?
 - L'outil est-il ouvert ? Permet-il l'importation et l'exportation des données ?
 - L'outil peut-il être interfacé avec d'autres outils ?
 - La base de données comporte-t-elle des restrictions quant au nombre et au type de données élémentaires pouvant être saisies ?
- **Documentation.** L'outil est-il doté de fonctions d'impression de la documentation ? Il convient de se poser les questions suivantes :
 - Existe-t-il une documentation spécifique pour chaque technique comme par exemple, la description des entités ou des processus ?
 - Les informations générées peuvent-elles être importées dans un système de traitement de texte ?
 - L'outil est-il capable de générer automatiquement la documentation ?
 - L'outil prend-il en charge différents formats de documents ?
 - L'outil génère-t-il automatiquement le code ?
- **Contrôle des modifications.** L'outil peut-il prendre en charge les différentes versions du système et permet-il de garder la trace des modifications ?
- **Tests.** L'outil intègre-t-il des fonctions permettant de tester les produits DSDM ? Il convient de se poser les questions suivantes :
 - L'outil prend-il en charge le résultat des tests statiques ?
 - L'outil prend-il en charge les tests dynamiques ?
 - L'outil prend-il en charge les tests de régression ?

24.2.5 Mise en œuvre - Mise en service

L'achat initial d'un nouvel outil de développement rapide ne représente qu'une petite partie de son coût global. Le coût des activités suivantes s'ajoute à la dépense initiale :

- **Formation des utilisateurs.** Qu'il s'agisse d'un cours de formation au sens strict ou d'une séance de familiarisation, tous les utilisateurs de l'outil doivent recevoir une formation.
- **Déploiement de l'outil.** Une stratégie de déploiement du nouvel outil doit être définie, selon qu'il est destiné à l'ensemble de l'entreprise ou uniquement au projet en cours.
- **Assistance technique.** Une équipe d'assistance technique doit être constituée afin d'aider les utilisateurs à se familiariser avec l'outil.
- **Mises au point.** Après que l'outil ait été introduit dans l'environnement réel du projet, il doit subir un certain nombre de mises au point et d'adaptations afin qu'il soit compatible avec les processus de l'entreprise, les procédures de contrôle qualité et les autres environnements d'aide.
- **Contrôle de l'utilisation de l'outil.** Après avoir introduit l'outil dans l'environnement de l'entreprise, il convient de contrôler régulièrement son utilisation, en étudiant les erreurs détectées et en déterminant si l'outil a produit les bénéfices escomptés.

Partie 3 - Evolution de la méthode DSDM

Depuis sa création, la méthode DSDM a évolué. Le processus de développement et les principes DSDM constituent désormais une base solide, à partir de laquelle la méthode peut se développer. Les deux premières parties de ce manuel ont énoncé les fondements de la méthode ainsi que les points essentiels qui la différencient des autres méthodes de développement. Néanmoins, toute méthode doit s'enrichir afin de répondre aux nouvelles exigences de ses adeptes. C'est pourquoi il convient de développer la méthode DSDM et d'identifier les domaines dans lesquels elle ne peut pas apporter de plus-value car les méthodes existantes sont parfaitement efficaces.

Le champ d'application de la méthode DSDM ne se limite pas au domaine informatique. Un «environnement professionnel DSDM» a été défini afin de situer la méthode DSDM et d'étudier les relations existant entre l'équipe de développement DSDM et les autres secteurs de l'entreprise. Il permet de définir le domaine d'application et l'orientation du projet, dans un contexte professionnel plus étendu.

A mesure que le champ d'application de la méthode DSDM s'élargit, le Consortium publie pour ses membres un certain nombre de documents consacrés à la méthode DSDM. Cette documentation est essentiellement constituée de différents manuels et des Livres Blancs DSDM. A l'occasion du développement de la version 3 de la méthode, le Consortium a défini la structure de la bibliothèque DSDM, regroupant toutes les publications actuelles et futures, consacrées à la méthode DSDM.

La troisième partie du manuel traite les questions suivantes :

- Description de la structure de la bibliothèque DSDM.
- Identification de l'environnement professionnel dans lequel la méthode DSDM est applicable.
- Définition d'un cadre permettant d'intégrer les futures applications de la méthode.

La troisième partie du manuel est consacrée aux dernières évolutions de la méthode et de ce fait, garantit l'actualité du manuel.

La troisième partie de l'ouvrage contient les chapitres suivants :

- Le chapitre 25 présente la structure de la bibliothèque DSDM et décrit le contenu des Livres Blancs DSDM.
- Le chapitre 26 décrit l'environnement professionnel dans lequel la méthode DSDM est mise en œuvre et identifie les éléments de l'entreprise qui contribuent au succès de la méthode.
- Enfin, le chapitre 27 présente les autres domaines dans lesquels la méthode DSDM est appliquée avec succès.

25 Bibliothèque DSDM

Le Consortium DSDM publie un grand nombre d'ouvrages, destinés aussi bien aux membres du Consortium qu'aux entreprises n'en faisant pas partie. L'ensemble de ces publications a été regroupé pour constituer la bibliothèque DSDM. Sa structure est prévue pour englober les publications existantes et futures. La figure 25.1 illustre la structure de la bibliothèque DSDM.

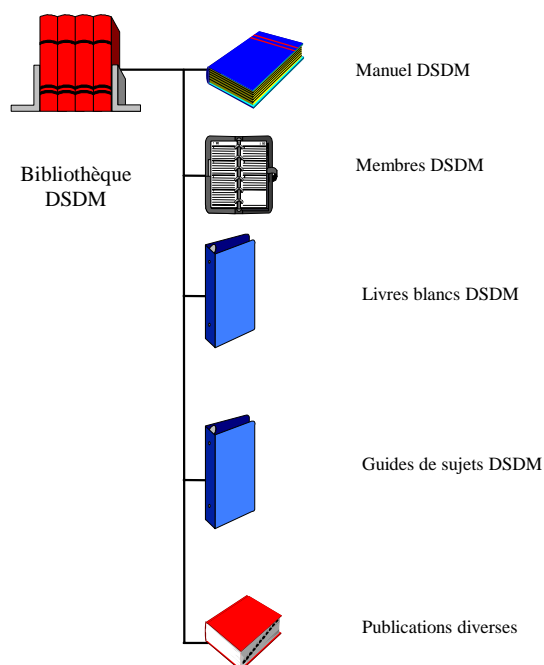


Figure 25.1 : La bibliothèque DSDM

La plupart de ces publications ont été produites par des membres du Consortium, à l'attention des autres membres. Ces ouvrages sont uniquement accessibles aux entreprises membres du Consortium. Toutefois, celles qui ont été produites en collaboration avec des entreprises extérieures au Consortium s'adressent à un public plus large.

La bibliothèque DSDM comprend les publications suivantes :

- **le manuel DSDM** (le présent ouvrage) ;
- **le manuel des membres DSDM**, qui contient des informations utiles sur le Consortium et prodigue des conseils aux adeptes de la méthode ;
- **les livres Blancs DSDM** destinés aux membres du Consortium et qui contiennent des informations pratiques et des conseils relatifs à l'application de la méthode dans un grand nombre de secteurs.
- **les guides DSDM**, qui fournissent des conseils pratiques et développent de manière plus détaillée les questions traitées dans les Livres Blancs DSDM ;
- **les publications diverses DSDM**, qui traitent un certain nombre de questions relatives à la méthode DSDM et qui s'adressent à un public plus large que le Consortium. Le guide TickIT ainsi que l'ouvrage «DSDM - the Method in Practice» sont deux exemples de ces publications ouvertes.

25.1 Résumé du Livre Blanc DSDM

Cette section contient les synopsis des Livres Blancs.

Note : le synopsis du Livre Blanc de la maintenance n'est pas mentionné dans ce chapitre car son contenu est inclus dans la section 11.3.

25.1.1 La méthode DSDM appliquée aux grands projets

Qu'est-ce qu'un grand projet ?

Avant de répondre à cette question, il convient de préciser que L'étude du Groupe de travail sur les grands projets (Large Projects Task Group) concerne les grands projets et non les programmes. Toutefois, il n'est pas exclu que ses principes puissent un jour être appliqués aux programmes.

Il existe diverses manières de définir la taille d'un projet. Selon le Groupe de travail, un grand projet est un projet qui dépasse l'enveloppe d'un projet DSDM. Les limites d'un projet DSDM peuvent être définies comme suit :

- plus de 5 développeurs par équipe ;
- plus de 5 équipes de développement ;
- plus de 1000 points de fonction par équipe de développement ;
- délai supérieur à 6 mois.

Ces chiffres ne représentent pas des limites absolues. En pratique, ce sont le nombre de seuils dépassés et l'amplitude des dépassements qui déterminent si le projet est en dehors de l'enveloppe DSDM et si les recommandations du Livre Blanc sont applicables.

Compétence et expérience

Avant d'entreprendre un projet de développement, toute entreprise doit confronter ses compétences et son expérience aux besoins du projet. Cette règle s'applique également aux projets DSDM. Par conséquent, la question qui se pose n'est pas «quelle est la taille du projet ?» mais plutôt «le projet entre-t-il dans le cadre de nos compétences et de notre expérience ?».

Le Livre Blanc identifie un certain nombre de questions que toute entreprise devrait se poser afin d'évaluer ses compétences et son expérience. Le but de cette démarche est de minimiser les risques en identifiant l'unique zone d'incertitude du projet, qui consiste à adapter la méthode DSDM à l'échelle du projet.

Application des principes DSDM

Le Livre Blanc est consacré à l'application des neuf principes DSDM aux projets de grande envergure et traite essentiellement des problèmes liés à l'augmentation des effectifs (développeurs, utilisateurs, etc.).

L'analyse détaillée contenue dans le Livre Blanc met en évidence les caractéristiques essentielles d'un projet de grande envergure :

- documentation rigoureuse des procédures et des normes ;
- organisation et coordination au sein des équipes, entre les équipes et entre les équipes et les membres de l'entreprise ;
- excellente communication permettant la visibilité des problèmes, des décisions et de leur impact sur le projet ;
- architecture étendue ;
- approche centrée sur les résultats permettant de maintenir l'intérêt et la motivation des développeurs et des membres de l'entreprise tout au long du projet.

Tous ces aspects ont un point commun, le formalisme. Une approche relativement informelle convient tout à fait à un petit projet DSDM et peut même être un facteur de réussite. En revanche, le manque de rigueur et d'organisation peut faire échouer un projet de plus grande envergure. Toute la difficulté consiste à adopter un niveau de formalisme permettant de gérer les problèmes relatifs à la taille du projet tout en respectant les principes DSDM, afin de ne pas alourdir les tâches administratives.

Découpage des grands projets

La meilleure façon de gérer un projet de grande envergure consiste à le diviser en sous-projets. Le Livre Blanc décrit les différentes manières de découper un projet en sous-projets concurrents ou séquentiels.

L'accroissement de la taille d'un projet induit inévitablement une augmentation des risques. Le découpage d'un projet en sous-projets permet de réduire ces risques.

Conclusion

La principale conclusion du Groupe de travail est qu'il n'existe pas de contre-indication majeure à l'application de la méthode DSDM aux grands projets. Le Consortium a déjà validé l'application partielle de la méthode à ce type de projets. Par conséquent, lorsqu'il a été décidé que l'intégralité de la méthode DSDM n'était pas appropriée à un grand projet, il est toujours possible de l'appliquer partiellement, en recourant à ses principes de base tels que le prototypage, les cadres temporels et les ateliers de prise de décisions.

En résumé, lorsque la méthode DSDM est appliquée à un projet de grande envergure, il est nécessaire d'allier la rigueur, en matière d'organisation, de coordination et de communication, aux neuf principes de la méthode.

25.1.2 La méthode DSDM appliquée aux solutions Internet

Principe

En raison de l'évolution rapide de la technologie Internet, très peu d'entreprises ont une expérience solide en matière de développement Internet. Les applications Internet sont très variées et vont du projet pilote au projet de grande envergure. C'est pourquoi il est indispensable d'identifier les meilleures pratiques.

La demande croissante des entreprises en matière de développement rapide souligne la nécessité de la méthode DSDM. Plusieurs entreprises membres du Consortium ont déjà eu recours à la méthode DSDM pour développer leurs applications Internet. Toutefois, il est nécessaire de résoudre un certain nombre de problèmes spécifiques aux applications Internet (utilisateur inconnu et éloigné, passage des frontières de l'entreprise, architecture client «apatride», rôle des spécialistes, etc.).

Le Groupe de travail Internet prodigue des conseils quant à l'utilisation de la méthode DSDM pour le développement des applications Web, Intranet, Extranet. Cette pratique implique l'extension et la spécialisation de la méthode.

Contenu

Le Livre Blanc décrit la nature des systèmes Internet ainsi que les différentes caractéristiques des applications publiques Internet et Intranet. Après avoir énoncé les règles de base de ce type de développement, le Livre Blanc donne des indications sur la manière d'adapter la méthode DSDM au développement Internet :

- interprétation des neuf principes ;
- risques liés au projet et à l'entreprise ;
- modifications de la structure du processus de développement ;
- contrôle des produits et critères de qualité ;

- structure et rôle des équipes de développement ;
- techniques complémentaires ;
- problèmes liés aux outils et aux langages.

Le Livre Blanc décrit également les problèmes spécifiques aux applications Internet et donne des conseils pour les résoudre :

- contexte professionnel des systèmes Internet ;
- gestion du «contenu», qui peut être indépendant du système informatique ;
- architectures Internet.

Conclusion

Lorsqu'elle est étayée par des techniques permettant de résoudre les problèmes spécifiques aux applications Internet, la méthode DSDM est parfaitement adaptée au développement Internet, Extranet et Intranet. Des produits complémentaires permettent de gérer les problèmes d'utilisation et de sécurité spécifiques aux systèmes Internet.

25.1.3 Sous-traitance

De nombreuses raisons peuvent amener une entreprise à sous-traiter un projet de développement. Dans ce cas, elle peut choisir parmi plusieurs types de contrat pour définir ses relations avec son fournisseur.

Le Livre Blanc étudie l'application des neuf principes DSDM en fonction de plusieurs contrats de rémunération, notamment ceux à rémunération fixe et ceux à rémunération journalière. Il apparaît clairement que les contrats à rémunération fixe, préconisés dans les projets de développement traditionnels appliquant le cycle de vie en cascade, n'est pas approprié au développement rapide d'applications (RAD). En effet, les contrats à rémunération fixe requièrent la définition précise des spécifications avant la conception du système, ce qui est impossible dans un projet de développement rapide.

Un contrat à rémunération journalière offre beaucoup plus de flexibilité qu'un contrat à rémunération fixe et convient mieux à un cycle de vie de développement itératif, qui ne permet pas une estimation fiable de l'effort dès le début du projet. Malheureusement, la plupart des clients n'accueillent pas favorablement le principe de la rémunération à la journée, qui apparaît comme un gouffre financier sans fin et qui leur laisse la responsabilité de gérer tous les aspects du projet. C'est pourquoi les contrats de rémunération avec gestion du temps sont plus séduisants car ils laissent au fournisseur le soin de gérer et de planifier tous les aspects du projet. Ce type de contrat est particulièrement adapté aux projets DSDM car il allie la flexibilité chère au fournisseur tout en déchargeant le client des tâches fastidieuses de gestion. En outre, ce type de contrat permet de dissiper la crainte des clients de ne jamais voir le projet se terminer car les livraisons fréquentes de produits associées au développement DSDM donne une meilleure visibilité de l'état d'avancement du projet.

Le Livre Blanc met en évidence un autre type de contrat particulièrement adapté aux projets DSDM. Il s'agit du contrat structuré, qui établit une relation de confiance entre le client et le fournisseur. Comme les contrats de rémunération à la journée, les contrats structurés ne nécessitent pas la définition détaillée des spécifications du produit, et permettent de faire appel aux ressources au fur et à mesure des besoins.

Le résumé du Livre Blanc stipule que :

- Les contrats structurés ou les contrats de rémunération à la journée avec gestion du temps sont les mieux appropriés aux projets DSDM.
- La définition détaillée des spécifications doit être évitée.
- La participation des utilisateurs doit être spécifiée par contrat, surtout pour les projets à rémunération fixe.

Note : Un certain nombre de relations contractuelles ont prouvé leur efficacité dans les projet DSDM :

- Le fournisseur présente une estimation du coût total du système avant de débiter le projet. En outre, il calcule un prix estimatif pour chaque phase ou cadre temporel. Les estimations doivent être régulièrement confrontées aux résultats, afin de mettre en évidence les dépassements éventuels. Le client peut décider à tout moment d'arrêter le projet. Dans ce cas, il n'aura pas dépensé inutilement son argent puisque dans ce type de projet, les livraisons de produits sont fréquentes.
- Le nombre et la durée des itérations doivent être fixés au début du projet. Bien que cette pratique puisse compromettre certains principes DSDM, elle permet néanmoins d'évaluer le coût de chaque itération et d'effectuer des livraisons fréquentes.
- Le client et le fournisseur peuvent signer un contrat de rémunération flexible, basé sur l'évaluation du «coût le plus probable». Dans ce cas, si le coût réel est inférieur au coût estimé, la différence est répartie entre le client et le fournisseur. Par contre, lorsque le coût réel excède l'évaluation, le coût supplémentaire est à la charge du client, mais le prix de la journée est revu à la baisse. Cette pratique permet de freiner l'ajout de spécifications inutiles, tout en préservant la flexibilité du processus de développement.
- Une autre approche efficace consiste à rémunérer les activités de développement en fonction d'un tarif journalier, jusqu'à un certain stade d'itération du modèle fonctionnel. A ce stade, les parties prenantes ont une meilleure visibilité du domaine d'application et peuvent se mettre d'accord sur un prix fixe pour l'achèvement du système.
- Une variante de la technique précédente consiste à rémunérer le développement du premier incrément sur une base journalière puis à déterminer un prix fixe pour les incréments suivants. Là encore, cette technique est rendue possible grâce à une meilleure appréhension du domaine d'application, après le développement du premier incrément.

25.1.4 Prototypage du processus de l'entreprise

Les techniques de prototypage du processus de l'entreprise sont appliquées dans le cadre des ateliers de prise de décisions. Le Livre Blanc considère le prototypage du processus de l'entreprise comme un moyen d'élargir le domaine d'application d'un projet DSDM. Le domaine d'application doit englober les éléments du processus de l'entreprise et aider les utilisateurs à s'adapter aux changements. Le prototypage du processus de l'entreprise révèle les différences entre les applications informatiques qui fonctionnent grâce à un processus manuel et celles dont le fonctionnement est automatisé.

Le Livre Blanc présente les avantages de la méthode DSDM appliquée au prototypage du processus de l'entreprise. Il décrit l'application de la méthode, indique où et quand elle doit être utilisée ainsi que les personnes qui doivent être impliquées. L'annexe du Livre Blanc présente quelques techniques de modélisation permettant de prototyper le processus.

Les points importants sont les suivants :

L'application des principes DSDM lors de la conception du processus de l'entreprise peut apporter un grand soutien aux utilisateurs :

- Le prototypage du processus de l'entreprise requiert des compétences en génie logiciel.
- Le délai peut être considérablement écourté lorsque les besoins sont définis au cours d'un ou plusieurs ateliers de prise de décisions.
- Le prototypage du processus de l'entreprise aide à combler le «fossé sémantique» qui sépare les développeurs et les utilisateurs.
- L'ensemble des scénarios de l'entreprise, classés en fonction de leur priorité, permet de tester le système de bout en bout.
- Le prototypage du processus de l'entreprise favorise la compréhension et la coordination du projet DSDM dans son contexte professionnel.

Selon le Livre Blanc, le prototypage du processus de l'entreprise présente les avantages suivants :

- Il permet de rester concentré sur les objectifs de l'entreprise.
- Il complète efficacement les techniques DSDM.
- Il donne un aperçu global des activités informatiques de l'entreprise, et traite à la fois les problèmes de technologie, de processus et de personnel.
- Il représente une puissante technique holistique de recherche de solutions informatiques.

26 Environnement professionnel DSDM

26.1 Introduction

La plupart des commentaires des utilisateurs de la méthode DSDM ont été pris en compte dans la version 3 et inclus dans le présent manuel. Toutefois, un certain nombre de remarques ont été jugées en dehors du champ d'application de la méthode. En effet, la méthode DSDM ne couvre qu'une partie des activités de développement et elle ne peut pas ajouter de plus-value à certaines pratiques courantes de développement. Néanmoins, certains commentaires concernaient plus l'environnement professionnel dans lequel la méthode était appliquée que la méthode elle-même.

Pour qu'elle soit parfaitement efficace, une méthode doit satisfaire un certain nombre de critères de réussite. La méthode doit être :

- facilement comprise par tous les utilisateurs de la méthode ;
- modulable, afin de s'adapter aux petits comme aux grands projets ;
- suffisamment flexible pour s'adapter aux contraintes spécifiques du projet et de l'entreprise ;
- délimitée par un cadre architectural, dans lequel elle peut être appliquée avec succès.

Le cadre architectural identifie les principales relations de travail qui doivent s'établir entre l'équipe de développement et l'environnement de l'entreprise. En outre, il doit définir les normes et politiques applicables à tout projet de développement, que la méthode DSDM soit utilisée ou pas.

Un projet DSDM n'est pas une fin en soi mais plutôt un moyen de mettre en œuvre un programme de changement au sein d'une entreprise. Par exemple, la méthode DSDM peut s'appliquer à une société qui souhaite adopter une culture d'entreprise plus orientée vers le service client. Un tel changement d'orientation a des impacts considérables sur les valeurs et les croyances de l'entreprise. L'orientation de l'entreprise est déterminée par les pressions que l'environnement extérieur exerce sur elle. La figure 26.1 présente les éléments de l'environnement professionnel de l'entreprise.

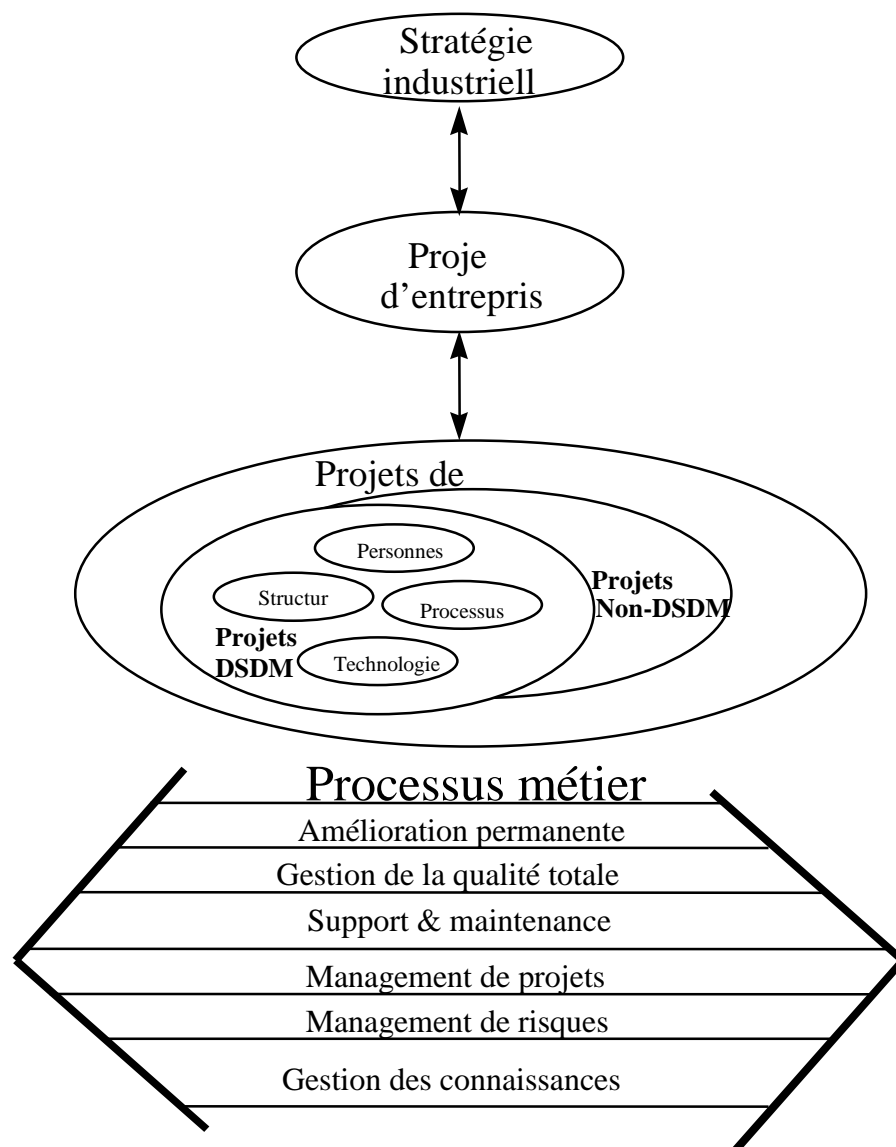


Figure 26.1 : Environnement professionnel DSDM

26.2 Orientation professionnelle de l'entreprise

L'orientation professionnelle de l'entreprise définit sa vision, sa mission et sa stratégie, ainsi que la direction vers laquelle elle évolue. La communication de cette vision à tous les employés est essentielle. Néanmoins, de nombreuses entreprises ne parviennent pas à obtenir l'adhésion de leur personnel à leur nouvelle politique. Les principes DSDM et les techniques permettant d'obtenir l'implication des utilisateurs, telles que les ateliers de prise de décisions, sont indispensables pour résoudre ces problèmes de communication.

26.3 Programme de changement

Un programme de changement est un projet permettant à l'entreprise de réussir sa réorientation, en tenant compte d'un certain nombre de contraintes budgétaires et temporelles. L'aspect fondamental d'un changement est l'impact qu'il aura sur la culture et les valeurs de l'entreprise. Tout changement doit faire l'objet d'une gestion et d'un contrôle rigoureux. Une entreprise doit être capable de s'adapter à l'évolution permanente de son environnement professionnel. C'est pourquoi toute entreprise moderne qui souhaite survivre et évoluer doit accepter des changements continuels.

Nombreuses sont les entreprises qui adoptent une politique de changement improvisée, mal conçue et mal planifiée. Dans ce cas, elles sont constamment confrontées à des changements inconnus dans leur environnement professionnel. Par contre, les entreprises qui réussissent élaborent un programme de changement dynamique, organisé et planifié.

Le cadre dans lequel le programme est mis en œuvre est un aspect essentiel. Il constitue l'architecture du programme et définit les grandes lignes de la planification des changements et de l'évolution de l'entreprise. Cette architecture détermine également les relations entre les projets DSDM et les projets non DSDM. Bien qu'elle varie d'une entreprise à l'autre, il est néanmoins possible de donner quelques indications sur sa structure, qui donnera à son tour quelques informations sur l'environnement dans lequel la méthode DSDM peut être appliquée.

Définition du cadre architectural DSDM

Il n'existe pas de définition universelle du cadre architectural. Dans le passé, les nombreuses tentatives de définition de ce terme ont donné des résultats très différents. C'est pourquoi les membres du Consortium DSDM ont tenté de rassembler toutes ces définitions afin d'en faire la synthèse et d'y intégrer les nouvelles tendances. Le résultat reflète parfaitement l'actuel courant de pensées en matière de développement de solutions informatiques et satisfait la plupart des besoins architecturaux.

La définition du cadre architectural tient compte des principes DSDM et d'un grand nombre d'objectifs. Bien qu'un peu complexe, la définition du Consortium DSDM résume tous ces aspects :

Le cadre architectural DSDM est un ensemble de disciplines techniques, de traitement, d'exploitation et de gestion permettant de spécifier la conception et l'élaboration de systèmes informatiques compatibles, qui satisfont les objectifs de l'entreprise. Il fournit un ensemble de politiques, de procédures et de normes grâce auxquelles des systèmes informatiques adaptés aux besoins de l'entreprise peuvent être définis, contrôlés, coordonnés, mise en œuvre et mis à jour.

Objectifs du cadre architectural

Le principal objectif du cadre architectural DSDM est d'instaurer un langage commun à tous les membres de l'entreprise et de l'équipe de développement, afin que tous les efforts convergent vers les objectifs de l'entreprise.

Le cadre architectural est constitué de trois principaux éléments :

- les critères de conformité s'appliquant à chacun des systèmes informatiques de l'entreprise ;
- le cadre technique et opérationnel qui régit ces systèmes ;
- la gestion de ce cadre (c'est-à-dire la gestion du programme de changement).

La définition du cadre architectural DSDM fournit un moyen pratique d'atteindre ces trois objectifs.

Le cadre architectural DSDM représente une architecture technique et professionnelle qui repose sur l'identification des normes et des applications. Les normes et les applications supposent l'existence et l'accès à une infrastructure de communication, qui constitue le pivot du réseau d'information. Cette infrastructure prend en charge l'intégration des applications courantes et leur interrelation.

De nouvelles normes continuent d'émerger et d'évoluer dans divers domaines d'application. Le cadre architectural DSDM ne cherche pas à remplacer ces nombreuses normes. Son objectif est d'identifier celles qui conviennent à la méthode DSDM.

Rôles et responsabilités

Le cadre architectural DSDM définit un certain nombre de rôles clés au sein de l'entreprise, permettant d'assurer la réussite du projet de changement. En plus des rôles DSDM définis ailleurs que dans ce manuel, des rôles extérieurs au domaine d'application de la méthode DSDM doivent être définis au sein de l'entreprise. Ces derniers varient d'une entreprise à l'autre et d'un projet à l'autre. Par exemple, une entreprise peut désigner une personne responsable de l'architecture des données, des besoins relatifs à la sécurité du système ou encore de la bibliothèque des composants réutilisables.

26.4 Projets de changement

Un projet de changement est un ensemble d'activités soumises à une gestion rigoureuse, qui permettent de modifier des comportements au sein d'une entreprise. Chaque projet de changement comprend quatre pôles principaux qui sont en relation les uns avec les autres :

- les **employés**, qui possèdent divers rôles et diverses responsabilités ;
- le **processus**, qui définit les relations de travail des employés ;
- la **structure**, qui définit l'organisation et les relations des employés ;
- la **technologie**, qui définit les ressources techniques nécessaires au processus.

Tout projet de changement est constitué de ces quatre pôles. Jusqu'à présent, de nombreux projets (informatiques ou non) ont ignoré les relations d'interdépendance existant entre ces quatre éléments. Un programme de changement contient plusieurs projets de changement, appartenant à l'une des catégories suivantes :

- projet DSDM, pour lequel la méthode DSDM a été jugée applicable ;
- projet non DSDM, pour lequel la méthode DSDM n'est pas applicable en raison de la réticence des employés, du manque de maturité de l'entreprise ou lorsque la technologie n'est pas adaptée au projet.

Quelle que soit la catégorie à laquelle il appartient, tout projet doit tenir compte des quatre pôles définis précédemment. Ces quatre éléments doivent être considérés dans leur ensemble car ils sont étroitement liés.

26.5 Projets de changement DSDM

Les projets de changement DSDM sont des projets pour lesquels la méthode DSDM a été jugée applicable. Jusqu'à présent, le filtre d'adéquation DSDM s'est concentré sur la

technologie. Il peut désormais être étendu, afin d'inclure des critères relatifs à la culture d'entreprise et à la capacité d'adaptation des employés à la méthode DSDM.

Par conséquent, les limites et les caractéristiques des projets de changement DSDM doivent être redéfinies, afin que la méthode DSDM ne se limite plus à l'aspect technologique.

Technologie

La méthode DSDM a été utilisée pour développer des projets basés sur la technologie :

- applications d'entreprise ;
- infrastructures de réseaux et Internet.

La technologie du processus DSDM a déjà été traitée précédemment dans ce manuel.

Processus

Il existe plusieurs définitions du « processus de l'entreprise» :

- série d'actions ayant pour but de réaliser les objectifs de l'entreprise, qui peuvent être modélisées chronologiquement en termes d'activités, d'événements et de flux de données et structurellement en termes de rôles et d'interactions ;
- activités au sein d'une entreprise où les employés travaillent ensemble pour réaliser un objectif, tel que la satisfaction de la clientèle.

Un processus d'entreprise décrit la manière dont les employés travaillent au sein de l'entreprise. Les processus d'entreprise peuvent être divisés en plusieurs catégories :

- **processus vitaux**, qui caractérisent l'entreprise (par exemple, la vente de polices d'assurance est un processus vital pour une compagnie d'assurance) ;
- **Processus de soutien**, qui assurent le fonctionnement de l'entreprise.

Employés

Les employés apportent les compétences nécessaires au bon fonctionnement de l'entreprise. Ils contribuent au développement de la culture d'entreprise. Les employés représentent une ressource flexible et peuvent par conséquent s'adapter aux contraintes du système. L'échec ou la réussite du système dépend de leur capacité d'adaptation et d'apprentissage.

Les êtres humains sont des êtres sociables par nature et s'adaptent facilement au travail d'équipe. La définition précise des rôles, des responsabilités et du pouvoir de décision au sein d'une équipe permet d'accroître son efficacité.

Les rôles et les responsabilités définis au sein des processus de l'entreprise permettent d'obtenir une description précise des postes des employés.

Structure

La structure définit la constitution des équipes et les relations entre les différents groupes. Elle détermine l'autorité et les rôles de direction au sein de l'entreprise. Lorsque les activités sont réparties sur plusieurs sites, la structure définit la topologie du système.

Taille du projet

Plus la taille du projet augmente, plus la nécessité de recourir à une méthode précise apparaît. La flexibilité et la modularité de la méthode DSDM lui permettent de satisfaire la plupart des exigences des grands projets.

26.6 Processus de l'entreprise

Dans le contexte de l'environnement DSDM décrit précédemment, un certain nombre de processus d'entreprise sont importants :

- programmes d'amélioration permanente ;

- systèmes de gestion de la qualité ;
- prise en charge et maintenance des systèmes opérationnels existants ;
- gestion du programme ;
- gestion des risques tels que l'impact sur la performance financière de l'entreprise ;
- gestion des connaissances permettant de réutiliser les acquis de l'entreprise.

Amélioration permanente

L'exploitation de l'expérience et des connaissances acquises est essentielle à la survie de l'entreprise. Les leçons tirées de projets antérieurs doivent être communiquées à l'ensemble de la communauté de l'entreprise. Le Software Engineering Institute (SEI) a défini le modèle CMM (Capability Maturity Model) qui permet d'évaluer la capacité d'adaptation et d'apprentissage de l'entreprise.

L'approche DSDM est un processus de développement. L'application de la méthode DSDM présente les mêmes caractéristiques que l'introduction d'un nouveau processus dans l'entreprise. Le modèle CMM décompose les principes de qualité et la gestion du processus en cinq niveaux, chacun d'eux comprenant un certain nombre de techniques fondamentales. Les différents niveaux de maturité du modèle CMM ainsi que son impact sont décrits ci-après :

1. **Le niveau initial**, dans lequel le processus de développement est improvisé voire chaotique. Dans ce cas, très peu de processus sont définis et la réussite du projet dépend de l'effort individuel.
2. **Le niveau de répétition**, dans lequel des processus de gestion de base sont mis en œuvre afin de contrôler le délai, le coût et la fonctionnalité du système. Les processus utilisés permettent la reproduction de projets antérieurs similaires ayant démontré leur efficacité.
3. **Le niveau de définition**, basé sur un processus logique et cohérent. A ce niveau, les processus de gestion et de développement sont documentés, normalisés et intégrés au sein du processus global de l'entreprise.
4. **Le niveau de gestion**, basé sur un processus détaillé et planifié. Le processus et les produits font l'objet de mesures et de contrôles précis.
5. **Le niveau d'optimisation**, dans lequel le processus est continuellement amélioré grâce à des technologies et des pratiques innovantes.

Le Consortium pense que l'introduction de la méthode DSDM dans une entreprise correspond au second niveau de maturité du processus. Les projets dont le modèle CMM est au niveau initial peuvent améliorer la discipline de leur processus de développement grâce à la méthode DSDM. L'entreprise peut ensuite affiner son processus, afin d'accroître la maturité du modèle CMM. Les activités cruciales permettant d'atteindre le niveau deux de maturité du modèle CMM sont les suivantes :

- Gestion des besoins - Rôle de l'utilisateur ambassadeur dans l'équipe DSDM (chapitre 15) ;
- Planification du projet (chapitre 16) ;
- Contrôle du projet (chapitres 16 et 17) ;
- Gestion de configuration (chapitre 20) ;
- Assurance qualité (chapitre 19) ;
- Gestion de la sous-traitance - Relations entre le client et le fournisseur (chapitre 26).

Système de gestion de la qualité

La gestion de la qualité consiste à exploiter le résultat des contrôles qualité, afin d'améliorer les processus de l'entreprise, y compris le processus d'assurance qualité. Un système de gestion de la qualité peut être élaboré afin de gérer les produits de contrôle qualité du projet DSDM.

Support technique et maintenance

Le support technique et la maintenance du système sont des activités cruciales qui doivent être gérées et contrôlées efficacement. La méthode DSDM peut être appliquée au cours de la phase de maintenance du cycle de vie d'un système. En outre, la maintenabilité d'un système est une exigence non fonctionnelle qui facilite des activités de maintenance. Il est indispensable que les besoins de maintenance soient définis par un personnel compétent et qualifié.

Gestion du programme

Le chef du programme de changement est responsable de la planification et du contrôle de plusieurs projets de changement, DSDM et non DSDM. Il est souvent nécessaire de faire appel à des partenaires et des fournisseurs extérieurs. Il est indispensable d'établir et d'entretenir de bonnes relations avec eux, afin de réussir l'intégration technique des produits DSDM et non DSDM.

Les projets de changement sont interdépendants et ont des impacts les uns sur les autres. C'est pourquoi la gestion du programme définit, grâce au cadre architectural, les limites et le domaine d'application de chaque projet, ainsi que les relations d'interdépendance existant entre les différents projets.

La gestion du programme de changement comprend les éléments suivants :

- la **structure de l'entreprise**, qui comprend les utilisateurs, la direction, les représentants des fournisseurs, les comités de direction, ainsi que leurs rôles et leurs responsabilités ;
- la **planification et le contrôle** qui consistent à gérer les ressources, les cadres temporels ainsi que les produits et leur qualité finale ;
- les **produits**, qui décrivent les résultats du projet de changement (Note : Les projets de changement ne génèrent pas tous des résultats tangibles. C'est le cas des projets qui visent à modifier la culture d'entreprise ou certains comportements au sein de l'entreprise.) ;
- les **activités** supplémentaires exécutées au sein d'un projet de changement, afin de réaliser les objectifs du programme de changement.

Gestion des risques

La mise en œuvre d'un changement s'accompagne d'un certain nombre de risques. Tout projet de changement doit recenser et gérer les risques qui lui sont propres. Dans un programme de changement, il est indispensable de gérer les risques liés à l'entreprise, afin qu'elle reste opérationnelle et viable financièrement tout au long des projets.

Par conséquent, les risques doivent être gérés au travers de plusieurs projets de changements parallèles, qui auront un impact sur les projets DSDM et leurs dépendances. Les principaux risques associés à un programme de changement sont les suivants :

- **Risques liés aux personnes** : maintenir le moral des participants et s'assurer qu'il se sentent concernés par le changement.
- **Risques liés à l'entreprise** : s'assurer que l'entreprise continue d'être viable pendant la mise en œuvre du changement.
- **Risques liés aux ressources** : s'assurer que les ressources nécessaires sont disponibles (budgets, outils, compétences, etc.).
- **Risques liés à la technologie** : s'assurer que l'infrastructure peut recevoir le changement envisagé.

Gestion des connaissances

De nombreuses entreprises se préoccupent désormais de gérer les connaissances afin d'accroître leur compétitivité. Le modèle de gestion des connaissances de l'entreprise comporte quatre aspects essentiels :

- L'**encadrement**, lorsque la gestion des connaissances est indispensable à la stratégie de l'entreprise ;
- la **culture** d'entreprise, qui encourage la diffusion de la connaissance ;
- la **technologie**, qui relie les membres de l'entreprise et permet le partage de la connaissance ;
- les **mesures**, lorsque l'entreprise a effectué une série de mesures, en fonction d'indicateurs financiers ou non financiers.

Le processus de gestion des connaissances de l'entreprise est essentiel dans le domaine de la réutilisation des composants informatiques d'un projet à l'autre. Dans un projet DSDM, il convient d'identifier les composants pouvant être réutilisés et de définir le processus de gestion des connaissances.

27 Avenir de la méthode DSDM

La méthode DSDM est en constante évolution et le Consortium s'efforce d'identifier les secteurs où elle peut être appliquée avec succès. Le champ d'application de la méthode ne se limite pas au domaine informatique. Elle a pour vocation d'apporter des solutions viables aux problèmes épineux qu'une entreprise peut rencontrer, dans un délai le plus court possible. Dans ce cas, le champ sémantique du terme «système» doit être élargi. Le concept «système d'entreprise» englobe les personnes, les processus, les structures et la technologie.

Depuis sa création, la méthode DSDM a été utilisée dans un grand nombre de domaines et s'est révélée fiable et polyvalente. Elle est désormais appliquée dans des secteurs différents de ceux pour lesquels elle a été initialement créée.

Voici les principaux domaines dans lesquels la méthode DSDM a prouvé son efficacité :

- prototypage du processus de l'entreprise ;
- gestion des changements au sein de l'entreprise ;
- nouvelle culture d'entreprise.

Le Consortium souhaite vivement recueillir les commentaires des membres qui ont appliqué la méthode DSDM dans des domaines non informatiques, afin d'inclure ces informations dans les futures mises à jour du Livre Blanc et des publications DSDM.

La survie de la méthode DSDM dépend des membres du Consortium. La méthode doit évoluer et refléter les derniers courants de pensée. C'est pourquoi il vous appartient, en tant que membre du Consortium, de transmettre votre expérience de la méthode, afin qu'elle puisse rester utile et crédible.

28 Bibliographie

- August, J., «Joint Application Design», Yourdon Press, 1991, ISBN 0-13-508235-8
- Beizer, B., «Software System Testing and Quality Assurance», Van Nostrand Reinhold, 1984
- De Marco, T. et Lister, T., «Peopleware: Productive Projects and Teams», Editions Dorset House, 1987, ISBN 0-932633-05-6
- Evernden, R., «The Information Framework», IBM Systems Journal, volume 35, numéro 1, pages 37 - 68, 1996
- Folkes, S. et Stubenvoll, S., «Accelerated Systems Development», Prentice Hall, 1992, ISBN 0-13-006073-9
- Jacobson, I., Ericsson, M., Jacobson, A., «The Object Advantage - Business Process Re-engineering with Object Technology», Addison-Wesley, 1995, ISBN 0-201-42289-1
- Kerr, J. et Hunter, R., «Inside RAD: How to build fully functional computer systems in 90 days or less», McGraw-Hill, 1994
- Martin, J., «Rapid Application Development», Editions MacMillan, 1991, ISBN 0-02-376775-8
- Myers, G., «The Art of Software Tests», Wiley, 1979
- Redmond-Pyle D. et Moore A., «Graphical User Interface Design and Evaluation - A Practical Process», Prentice Hall, ISBN 0-13-315193-X
- Stapleton, J., «DSDM: The Method in Practice», Addison Wesley Longman, 1997, ISBN 0-201-17889-3
- Symons, C., «Software Sizing and Estimating - Mk II Function Point Analysis», Wiley, 1991, ISBN 0-471-92985-9
- Sowa, J.F. et Zachman, J.A., «Extending and Formalising the Framework for Information Systems Architecture» IBM Systems Journal, volume 31, numéro 3, pages 590 - 616, 1992
- Tudor, D. et Tudor, I., «Systems Analysis and Design - A Comparison of Structured Methods», Editions Macmillan, 1997, ISBN 0-333-72139-X
- Turner, P. et Jenkins, T., «Euromethod and Beyond», International Thompson Computer Press, ISBN 1-85032-149-3
- Wood, J. et Silver, D., «Joint Application Development, 2nd Ed.» John Wiley & Sons, 1995
- Zachman, J. A., «A Framework for Information Systems Architecture», IBM Systems Journal, volume 26, numéro 3, pages 276 - 292, 1987
- ISO 9000-3, «Quality Management and Quality Assurance Standards, Partie 3 : Guidelines of ISO 9001 to the Development, Supply and Maintenance of Software», 1991
- «Dynamic Systems Development Method and TickIT», ISBN 0-580-27081-5 (disponible auprès du Consortium DSDM et de la British Standards Institution)
- «Mark II Function Point Analysis Counting Practices Manual» (disponible auprès de l'UKSMA)
- «IFPUG Function Point Counting Manual» (disponible auprès de la National Software Metrics Association ou de l'IFPUG)
- Vous trouverez également une mine d'informations sur Internet. Voici une liste de sites Web qui traitent un certain nombre des questions abordées dans ce manuel :
- Brad Appleton's Software Engineering Links (contient des liens vers plus de 700 sites de génie logiciel)

<http://www.enteract.com/~bradapp/links/swe-links.html>

- Software Engineering Institute (Capability Maturity Method)

<http://www.sei.cmu.edu/>

- The Configuration Management Yellow Pages

http://www.cs.colorado.edu/users/andre/configuration_management.html

- The Software Quality Page

<http://www.tiac.net/users/pustaver/>

- Transaction Processing Performance Council

<http://www.tpc.org/>

- United Kingdom Software Metrics Association

<http://www.exemplar.co.uk/UKSMA.html>

Malgré tout le soin apporté à la collecte de ces informations, le Consortium ne peut garantir l'exactitude des adresses ni la fiabilité des informations contenues dans ces pages Web.

29 Glossaire

Atelier Facilité	Forum réunissant des membres de l'entreprise et de l'équipe informatique, investis d'un pouvoir de décision. La séance, animée par un médiateur impartial, a pour but de générer des produits et des décisions dans un cadre consensuel.
Besoin industriel ou besoin métier	Caractéristique technique ou professionnelle du système indispensable pour qu'il soit conforme aux exigences de l'entreprise.
Timeboxe	Période de temps comprise généralement entre deux et six semaines au cours de laquelle une équipe élabore, teste et valide un produit (partiel ou complet) ou un ensemble de produits.
Classe d'utilisateurs	Groupe d'utilisateurs ayant le même type d'attributions au sein de l'entreprise.
Composant	Élément d'un système informatique.
Développeur	Membre de l'équipe informatique, qui construit des composants.
Equipe	Groupe d'utilisateurs et de développeurs participant à un projet DSDM afin de livrer un ou plusieurs produits permettant de réaliser les procédures de l'entreprise.
Fonctionnalité	Partie informatisée d'un système permettant d'automatiser les processus de l'entreprise.
Incrément	Élément d'un système informatique destiné à être utilisé dans l'environnement professionnel de l'entreprise.
Itération	Amélioration et extension de produits existants.
Jeu minimal de fonctions	Fonctionnalité minimale d'un composant permettant de satisfaire les exigences de l'entreprise.
Livraison incrémentielle (ou évolutive)	Livraison d'incréments dans une période de temps.
Principe	Élément de base de la philosophie DSDM.
Produit	Tout élément généré au cours du développement DSDM, qu'il s'agisse d'un document de gestion de projet, d'un modèle d'analyse, d'un rapport de test ou d'un composant logiciel.
Prototypage incrémentiel (ou évolutif)	Transformation d'un prototype en un composant qui constitue un incrément ou un système informatique complet.
Prototype	(nom) Composant qui n'est pas nécessairement automatisé et qui permet de se rendre compte si le système est conforme aux attentes de l'entreprise. (verbe) Elaborer un tel composant.
RAD	Développement Rapide d'Applications. Elaboration d'un système informatique dans une période de temps limitée, sans recourir aux méthodes traditionnelles de développement.
Règles de MoSCoW (FISPE)	Technique permettant de classer les besoins en fonction de leur priorité.
Système	Ensemble des procédures manuelles et automatiques qui permettent de générer les produits et les services d'une entreprise. Note : lorsque le contexte le précise sans ambiguïté, le terme «système» désigne un système informatique.
Système informatique	Ensemble des composants matériels et logiciels qui prennent en charge les processus de l'entreprise.
Utilisateur	Personne utilisant une partie ou l'intégralité du système informatique.

Annexe A - Filtre d'adéquation

A.1 Objectifs du filtre d'adéquation

Le filtre d'adéquation est un ensemble de critères permettant de déterminer si la méthode DSDM est appropriée à un projet donné. Les facteurs d'adéquation reposent sur les principaux critères de réussite de la méthode et sur les caractéristiques du projet, décrites au chapitre 4. La liste de ces critères est indicative et non exhaustive. En outre, le filtre d'adéquation doit être affiné et enrichi à mesure que l'expérience de la méthode DSDM de l'entreprise augmente.

En plus du filtre d'adéquation, une liste de facteurs supplémentaires permet de déterminer si le projet est adapté à l'approche DSDM. Il n'y a ni bonne ni mauvaise réponse mais leur impact sur le projet doit être clairement identifié.

A.2 Résultats du filtre d'adéquation

Le filtre d'adéquation permet de déterminer si la méthode DSDM est applicable partiellement ou intégralement. Lorsqu'elle ne peut pas être appliquée dans son intégralité, il convient d'identifier les techniques les mieux appropriées au projet. Si le responsable du projet souhaite appliquer la méthode DSDM bien qu'un ou plusieurs facteurs de réussite ne soient pas satisfaits, ces facteurs doivent être considérés comme des risques majeurs et faire l'objet d'un contrôle rigoureux.

Lorsque les membres de l'équipe de développement n'ont aucune expérience de la méthode DSDM, un très haut niveau d'adéquation du projet est recommandé.

	Facteur d'adéquation	adapté (O/N)	Commentaires
1.	Le commanditaire et les membres de la direction comprennent-ils et acceptent-ils la philosophie DSDM ?		L'adhésion à la méthode est fondamentale.
2.	Les membres de l'équipe seront-ils habilités à prendre des décisions au nom de leur communauté ?		Aspect essentiel de la méthode DSDM.
3.	Y-a-t-il un engagement de la part des responsables garantissant la participation des utilisateurs ?		S'assurer qu'un groupe d'utilisateurs habilités à prendre des décisions a été constitué et qu'ils se sont engagés à participer activement au processus de développement.
4.	L'entreprise peut-elle s'adapter aux livraisons fréquentes d'incréments ?		Nécessité des procédures de gestion de configuration et de livraison.
5.	Les développeurs auront-ils accès aux utilisateurs tout au long du projet ?		Leur cohabitation sera-t-elle nécessaire ou bien un niveau plus faible de participation sera-t-il suffisant ?
6.	La composition de l'équipe de développement subira-t-elle des modifications au cours du projet ?		La stabilité de l'équipe comprenant les représentants des utilisateurs est fondamentale.
7.	L'équipe de développement possède-t-elle les compétences requises ?		Outre leurs compétences techniques, les développeurs doivent avoir une bonne connaissance des activités de l'entreprise.
8.	Quel est l'effectif de chacune des équipes de développement ?		Chaque équipe ne doit pas comporter plus de six personnes, en comptant les utilisateurs.
9.	Existe-t-il une relation d'entraide au sein des équipes ?		Entre les développeurs et les utilisateurs.
10.	La technologie utilisée sera-t-elle adaptée au prototypage ?		La plate-forme de développement doit permettre le développement itératif et lorsque c'est nécessaire, le développement réversible.
11.	L'interface utilisateur offre-t-elle de nombreuses possibilités ?		Ecrans, rapports, impression de fichiers, etc.
12.	La personne qui contrôle les ressources du projet est-elle bien identifiée ?		Existe-t-il une personne capable de résoudre les problèmes politiques et de fournir l'accès aux ressources ? Y-a-t-il un groupe bien défini d'utilisateurs ?
13.	Le développement sera-t-il complexe, du point de vue des techniques informatiques ?		Plus le développement est complexe, plus les risques sont grands.
14.	La solution informatique peut-elle être développée sous forme d'incréments ?		Le système peut-il être découpé en composants plus petits ?
15.	Le projet est-il soumis à un délai fixe ?		La livraison du système est-elle urgente ?
16.	Les besoins peuvent-ils être classés par priorité ?		Les règles de MoSCoW sont-elles applicables ?
17.	Les spécifications sont-elles trop détaillées et trop figées ?		Les utilisateurs pourront-ils définir les besoins de manière interactive ?

Questions supplémentaires	Niveau du risque		
Evaluation du risque	Elevé	Moyen	Faible
Quel travail a déjà été fait pour définir les besoins ?			
Quel est l'état de l'entreprise ?			
Quelles ressources ont été allouées au projet ?			
Qui sont les fournisseurs des ressources de développement ?			
Quel est le coût estimatif du projet ?			
Quelles sont les attentes de l'entreprise ?			
L'entreprise a-t-elle l'expérience de ce type de projet ?			
Des composants logiciels ou des processus de l'entreprise issus de projets antérieurs pourront-ils être réutilisés ?			
Le projet est-il compatible avec l'architecture technique et celle de l'entreprise ?			
L'utilisation de nouvelles techniques sera-t-elle nécessaire ?			
Les interfaces devront-elles subir des modifications ?			
La conception de la base de données constituera-t-elle une partie importante du projet ?			
Le projet requiert-il une infrastructure importante (matériel, formation, etc.) ?			
Existe-t-il des problèmes avec les fournisseurs ?			
Y-a-t-il des problèmes de relations industrielles ?			
Le projet fait-il partie d'un programme plus large ?			
Quel sera le coût des exigences non fonctionnelles ?			

Annexe B - Planification des tests

La planification des tests doit débuter lors de L'Etude de Faisabilité. Les activités éventuelles de planification des tests sont résumées dans le tableau ci-après. La quantité d'effort requis lors de la planification des tests dépend de la taille et de la nature du projet. Les informations présentées dans le tableau concernent les grands projets soumis à un contrôle rigoureux et doivent être adaptées à la nature du projet à l'étude.

Phase DSDM	Produit de planification des tests		Activité de planification des tests
Etude de faisabilité	T.DP.1	<i>Evaluation des tests (partie de la DP.1)</i> DP = Description du Produit	La méthode d'évaluation des besoins de l'entreprise doit être identifiée. Les besoins, et par conséquent les critères d'acceptation doivent être étudiés afin de s'assurer de leur faisabilité. L'évaluation doit également couvrir : <ul style="list-style-type: none"> - les besoins relatifs à l'intégration du système - le nombre et le type des environnements de test - les outils qui seront utilisés - les systèmes existants à interfacer - les besoins en matière de compétences et de formation
	T.DP.2	<i>Modèle des tests de faisabilité (partie de la DP.1a)</i>	Ensemble de tests dynamiques pratiqués sur le prototype de faisabilité optionnel afin d'étudier le comportement de l'architecture du système (modularité, performances et robustesse).
Analyse des besoins	T.DP.3	<i>Plan général des tests (partie de la DP.5)</i>	Mise en place d'un environnement de test, planification des ressources de test et recensement des activités d'identification, de création et d'exécution des tests (produits de test T.DP.4-8)
	T.DP.4	<i>Modèle de test du système de l'entreprise (partie de la DP.3)</i>	Ensemble de tests permettant de contrôler la conformité du système aux besoins de l'entreprise. Ces tests reposent sur des scénarios du processus de l'entreprise. Ils ont pour but de diminuer le coût du système et d'améliorer la qualité, la vitesse et le service client. Le modèle de test permet également de tester les étapes du processus pris en charge par le système informatique.
	T.DP.5	<i>Modèle de test de l'architecture du système (partie de la DP.4)</i>	Plan de test du système informatique comprenant des tests statiques et des tests dynamiques : <ul style="list-style-type: none"> - tests d'intégration du système - tests de connectivité et de logiciel standard personnalisé - test du volume de la base de données - tests de conformité aux normes de l'entreprise - tests de coexistence
Itération du modèle fonctionnel	T.DP.6	<i>Plan d'inspection du système informatique (partie de la DP.6)</i>	Plan indiquant le calendrier (en termes de scénarios d'entreprise à tester) des tests d'inspection du prototype fonctionnel et des exigences non fonctionnels).
Itération de conception et de construction	T.DP.7	<i>Plan des tests du prototype de conception du système (partie de la DP.9)</i>	Liste des tests permettant de contrôler le fonctionnement du système ainsi que sa conformité aux exigences non fonctionnelles et aux spécifications d'implémentation.
Mise en œuvre	T.DP.8	<i>Plan des tests du système implémenté (partie de la DP.14)</i>	Calendrier des tests d'efficacité du système informatique, mis en œuvre après son installation. L'objectif de ce plan est de prévoir les contrôles permettant de déterminer si une action est requise pour maintenir les objectifs de l'entreprise lors de l'implémentation du système.

Annexe C - Fiche de mesures

Cette fiche permet l'enregistrement des mesures de projets DSDM. Complétez une ou plusieurs fiches par projet et adressez-les au secrétariat DSDM.

Fiche de résultat des mesures DSDM

Société	Projet	Contact	Date
Brève description du projet ou prototype et de la Technologie utilisée			
Domaine d'application des mesures (prototype, itération, phase, projet)			
Complexité (mise à jour par le prototypage)			

Type de mesure	Unité	Quantité	Qualifications/ Commentaires
Taille	Point de fonction ou ligne de code		
Effort (différencier si possible l'effort des développeurs de celui des utilisateurs)	heure		
Durée	mois ou semaine		
Qualité			
Aspects spécifiques au projet (voir chapitre 4)			