UML/RUP : Organisation des processus de développement

Des concepts à la réalité
Application et compléments pour construire une ingénierie du processus de développement

Les directions des entreprises marquent un intérêt renouvelé à la notion de processus.

Si on entend par là « ce que fait l'entreprise, avec quelles ressources, pour servir quels objectifs», c'est la réaffirmation de principes simples et anciens :

- *Pour maîtriser la performance de l'entreprise, il faut rapporter la consommation de ressources (financières, humaines ou matérielles) à l'intérêt de ce qu'elles produisent (article ou service à la vente, support d'un objectif stratégique).*
- *Pour organiser la réactivité de l'entreprise il faut mesurer l'impact de différents scénarios sur des ressources, des produits ou des critères particuliers. Ces scénarios peuvent relever de l'organisation (re engineering), du marché (variation des caractéristiques de la demande) ou du risque (défaillance d'une ressource ou d'un processus).*
- *Pour organiser la pérennité de l'entreprise il faut pouvoir reproduire les savoir-faire dans le temps. Il faut intégrer de nouvelles ressources ou élargir le périmètre de l'entreprise pose évidemment le besoin de faire savoir ce qui doit être fait pour réaliser le « programme » de l'entreprise.*

Une organisation qui se consacre au développement informatique est une véritable entreprise. La mise sous contrôle de ses processus de production (informatique) déterminera sa capacité à satisfaire ces trois vocations.

UML/RUP (Rational Unified Process) donne un cadre référentiel aux processus de développement informatique.

La liste des activités qu'il prévoit, et aussi bien le modèle de consommation des ressources qu'elles induisent, sont utiles au gestionnaire de projet. Ils lui fixent un cadre conceptuel de planification.

Cependant, la prise en charge de la notion de processus par RUP s'avère relativement trop sommaire pour fournir un outillage opérationnel dans l'organisation pratique d'un projet de développement.

C'est pourquoi il a fallu étayer son utilisation avec des concepts et des moyens plus opérationnels, apportés par un outil particulier d'analyse de processus.

Ceux ci ouvrent en outre la voie d'une capitalisation de l'expérience de l'équipe de développement, dans le cadre d'une gestion de la connaissance appliquée.

RUP : un modèle général de gestion des projets de développement informatique

Rational Unified Process (RUP) est un modèle type de processus de développement informatique. Il résulte de l'unification des travaux de MM. Booch, Rumbaugh, et Jacobson. Il est étroitement lié à la technologie Unified Modeling Language (UML)

Ce modèle propose de gérer les développements informatiques en utilisant un modèle d'activités réparties en 4 catégories

Phases	*Contenu des phases*
Inception	Establish business case for system
	High-level project definition (define actors and general use cases)
Elaboration Phase	Analyse problem domain
	Establish architectural foundation
	Develop project plan and eliminate high risk areas
Construction Phase	Develop, test, and integrate all project components
Transition Phase	Handover product to user community

Cycle de Développement

Taxonomie : Un modèle logique

Appliquer ce modèle confère à chaque développement des garanties de productivité et de fiabilité.

En établissant un référentiel d'activités, d'objectifs et de livrables, le Schéma Directeur que constitue RUP apporte une meilleure structure aux différentes parties d'un projet de développement, notamment dans un contexte distribué.

Il donne aux différents acteurs un vocabulaire identique, voire une représentation logique homogène, pour décrire et comprendre les différents travaux, livrables et points de contrôle qui construisent chaque phase d'un cycle de développement.

	Phases			
	Inception	***Elaboration***	***Construction***	***Transition***
Activities	• Define system scope • Identify Actors and high-level interactions (Use Cases) • Define project success criteria • Identify risk • Estimate required resources (people, tools, software, machines, etc.) • Develop phase plan showing major project milestones	• Analyse problem domain • Establish sound architectural foundation • Develop project plan • Eliminate highest risk elements of project • Look at every aspect of the system; but not very deep	• Implement and test all features • Integrate and test all components	• Focus on delivering the product to the user community • Preliminary user interaction ("beta-testing") • Bug fixes, enhancements • Parallel operation with previous legacy system • Database conversion • User and maintenance documentation • User and maintenance training • Product roll-out to marketing, sales and distribution
		Most critical phase of project – Most difficult technical decisions are tackled – Technical prototypes validate design and provide framework for Construction	Management transitions from developing intellectual property to producing quality deployable software	

Deliverables

Phases			
Inception	***Elaboration***	***Construction***	***Transition***
• Vision Document Core requirements, Key features, Main constraints • Business Case Business Context, Succes Criteria, Market Recognition, Financial Forecast • Initial Use Case model • Business Model • Project Glossary • Project Plan • Prototypes	• Nearly completed Use Case Model • Non-functional requirements • Software architecture description and executable prototype • Development plan for entire project –Define iterations • Update of any existing documents –Risk list –Business Case	Software • Complete • Integrated • Tested • Fully-functional • Deployable • User manual • Release notes	• Achieve user self-supportability • Stakeholder agreement that software is stable, complete and consistent with vision's evaluation criteria • Achieve final product baseline as quickly and cost-effectively as possible

Milestone

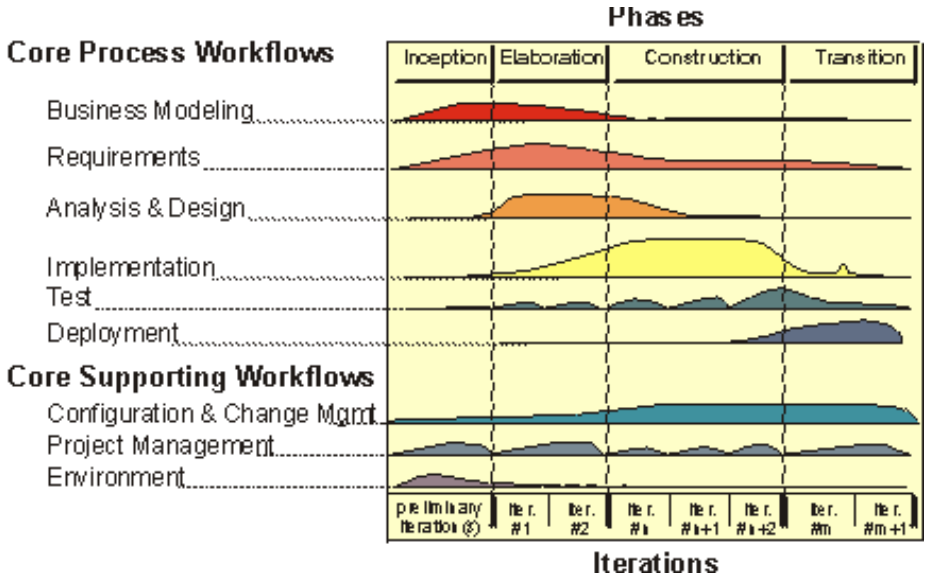
Phases			
Inception	***Elaboration***	***Construction***	***Transition***
• Stakeholder agreement/signoff on scope, cost, and schedule estimates • Understanding of requirements based on Use Cases • Credibility of estimates, priorities, risks, process • Actual costs vs. Planned costs • Is the depth and breadth of developed architectural prototypes sufficient?	• Lifecycle Architecture Milestone • Is project vision stable? • Is architecture stable? • Is Construction phase reasonably planned in sufficient detail? • Do stakeholders agree that current vision can be achieved? Actual costs vs. Planned costs Believable plan based actual time spent? Have major risks been addressed and resolved?	• Initial Operational Capability Milestone • Is the release stable and mature? • Are the stakeholders ready for the transition to the user community? • Actual costs vs. Planned costs	• Product Release Milestone • Have objectives been met? • Should another development cycle start? • Are the users satisfied? • Actual costs vs. Planned costs
• Project may be cancelled or re-analysed if milestone not met	Project may be cancelled or re-analysed if milestone not met	Project may be postponed by a release if milestone not met	

Consommation de ressources : Un modèle prédictif

Chacun des groupes d'activités utilisés par RUP peut être caractérisé par un comportement type.

Les types de ressource consommées, leur niveau de consommation et leur calendrier d'intervention est interpolé par le modèle, en fonction d'une expérience postulée a priori.

Un tel calendrier peut encore être amendé par des stratégies générales d'organisation.



Un développement en mode fontaine réinitialisera les phases de travail amont à l'occurrence d'un problème inopiné, ou à l'apparition d'une variation de contextes. De ce fait, il y a nécessairement une sur utilisation des phases, directement fonction de leur place dans l'enchaînement des taches.

Un développement en mode continu (Workflow) lisse les surconsommations dans la mesure où les mêmes perturbations (problèmes et contextes) ne réinitialisent les phases qu'après la fin de leur enchaînement, et donc également pour chacune.

Un référentiel conceptuel

Les assertions du modèle RUP restent générales. S'il donne la forme générale d'un discours, il n' affiche pas vraiment l'ambition de structurer un développement concret.

L'absence de formats standardisés de livrables laisse aux équipes informatiques une grande latitude de compréhension des termes employés, voire des taches opérationnelles impliquées dans les différentes rubriques du référentiel.

UML est le support naturel de la communication aux travers des projets RUP. Le pari que ce média puisse apporter a lui seul une garantie d'inter opérabilité des informations contenues dans les documents, et de leur inter interprétabilité dans des tâches particulières, ne suffit pas à garantir un résultat dans ce domaine.

La maîtrise des paramètres de performance du développement

RUP attire à juste titre l'attention sur l'importance de l'architecture générale du processus de développement dans la recherche de performance.

Les limites des approches techniques

L'évolution des pratiques informatiques a permis d'intégrer successivement différentes composantes de cette performance : en termes d'outils techniques (bases de données relationnelles…) en termes de méthodes de construction du code (Langage C, Modèle Objet…)

Cependant, le développement informatique reste complexe. Il nécessite l'intervention de plusieurs corps de métier ayant ou maîtrisant des cultures et des outils différents.

Il pose en permanence, comme un défi, le besoin de maîtriser l'application à des contextes mouvants (Aspect fonctionnel = besoins métier) de macro innovations (aspect Technologique : Concepts et offres technologiques nouvelles) ou de micro innovations (aspect organique = intégrations à des contextes chaque fois particuliers) dont la combinatoire est très importante.

La programmation de type Objet, par exemple, est une avancée marquante… dont les limites sont cependant maintes fois reconnues en ce que, pour garantir la réutilisabilité des composants, il faut les réduire à des dimensions extrêmement réduites.

Le choix de leur périmètre renvoie à la modélisation de leur contexte d'application (sur les trois plans technologique, organique et fonctionnels) qui s'avère rapidement un exercice inductif (a priori) impossible.

Les apports de la maîtrise du processus

Si tous les contextes d'application du développement informatique ne peuvent être cernés a priori dans un modèle technique général, il reste possible de modéliser la meilleure manière d'approcher, de définir, puis d'appréhender, ces mêmes contextes.

Cette meilleure manière de faire se définit comme un processus, c'est à dire un ensemble d'activités, rattachées en domaines homogènes de finalité, énoncées dans un plan d'enchaînements et de dépendances, dans un modèle contrôlable.

UML/RUP tente bien de définir un tel paradigme : former un référentiel sur les modes d'organisation des équipes de développement, sur la manière de développer la compétence de groupe, sur la manière d'intégrer la relation avec les clients des projets.

Cependant, il ne peut être considéré comme une technique de modélisation du processus de développement informatique. (à l'intérieur d'une plus large incompétence, puisqu'il ne peut non plus nous guider dans la modélisation des processus Métier des utilisateurs. Voir plus bas)

RUP propose un modèle postulé, un résultat donné, sans proposer de mécanisme d'adaptation de ce modèle, et a fortiori un mécanisme de gestion d'autres modèles.

La notion de savoir-faire au centre de la maîtrise de l'innovation.

Dans les développements informatiques, il y a bien des acteurs, des activités, des enchaînements de tâches, des produits, des consommations de temps, des itérations, des échanges …et, plus loin encore, des objectifs, des contraintes, des indicateurs d'évaluation de performance.

Toutes ces choses caractérisent classiquement les processus d'une entreprise, aussi bien qu'un projet de développement informatique.

Pourtant le terme de processus est d'emblée suspect aux développeurs informatiques. Avant peu, on obtient l'objection qu'un projet ne ressemble à aucun autre alors que la notion de processus renverrait à celle de re-production du même … et en grande série si possible.

Cette objection est légitime et pose à l'exercice de modélisation une contrainte majeure : ce qui doit être modélisé est le savoir faire du développement, qui est avec le processus (de développement) dans le rapport de la cause à l'effet :

Tout processus est l'instance d'un savoir faire théorique (Modèle).

Le processus est l'effet observable, dans le cadre d'une instance particulière, du savoir faire qui est à l'œuvre en dessous. Le second ne peut être reconnu que par l'intermédiaire du premier, mais seul le second permet de modéliser (a priori) le premier.

La voie de la modélisation du processus de développement informatique, si elle se veut opérationnelle (au regard de l'obligation d'innovation) doit prendre en charge, au travers de la notion de savoir faire, un problème quasiment épistémologique.

L'apport d'une technique évoluée d'analyse de processus

La confrontation des modèles proposés par RUP aux concepts manipulés par un outil d'analyse de processus (Kapi : Knowledge Acces & Process Intellligence) aide à reconnaître leurs apports, puis à les relativiser, et à les compléter.
L'analyse des processus est clairement une technique nécessaire à tout projet informatique. Par nature, un programme informatique est une fiction plaquée sur la réalité de l'entreprise.

Une connaissance exacte des processus (opérationnels) qui doivent être supportés permet de réduire l'écart entre produit (logiciel) et besoin (de l'activité à informatiser).

Mais la vision que permet d'obtenir l'analyse de processus sur le contexte externe du projet peut s'appliquer avec beaucoup d'avantages sur l'intérieur du développement lui même.

La fonction topologique

Le modèle d'un processus établit par nature une topologie de toutes les interactions entre besoins, stratégies, moyens, contraintes et savoir faire dans l'entreprise.
Pour les besoins du planificateur, cette topologie forme une grille où peuvent être positionnés des temps, des acteurs, des contraintes …. Elle prend en support la représentation du processus en arbres de décomposition et en flux transversaux.
Pour les besoins du gestionnaire du projet, elle forme une grille où peuvent être lues de nombreuses informations, qui peuvent alors être évaluées dans le contexte qui leur donne un sens. Elle prend en support la représentation multi dimensionnelle, qui utilise les contextes en niveaux de synthèse de l'information, et le choix des dimensions d'analyse comme autant de cadres logiques de contrôle (qualité des livrables, respect des charges, respect des délais, respect des coûts, divergences des activités prises en charge, contrôle des affectations de ressources…).

La fonction topologique s'alimente de l'effort de typologie (concepts et catégories particuliers) que postule UML/RUP.

Elle résulte en une urbanisation généralisée des éléments du processus de développement (acteurs, activités, consommations, outils, objectifs …).

La fonction de contextualisation

La notion de savoir-faire sur laquelle on a insisté précédemment prend corps avec la fonction de contextualisation présente dans la plate-forme d'analyse de processus prise en référence (Kapi), et qui met en œuvre l'observation suivante :

Dans la réalité, toute entreprise combine de manière différente ses ressources (qu'elle trouve dans ses modèles structurels) et ses compétences individuelles (activités) ou organisationnelles (savoir faire) pour répondre à des événements (besoins, demandes, contraintes) particuliers. De ces combinaisons ou scénarios naissent des processus différents … et en créent éventuellement de nouveaux.

L'utilisation de la notion de savoir-faire dans la gestion de projet inductive conditionne la faisabilité de toute démarche de contrôle de l'innovation (particulièrement présente dans le développement informatique).

Elle règle la vie de la cellule de développement sur le cycle suivant :

- en fonction des différents modules ou fonctionnalités à prendre en charge par le projet concerné ;
- Chaque projet spécifique instancie (en processus contrôlables) le modèle général d'activité proposé par RUP : des acteurs sont sélectionnés, des activités sont détaillées de façon opérationnelle, des temps sont fixés …
- l'instance ainsi réalisée est un enrichissement du modèle. Elle peut prendre directement statut de modèle dès lors que le niveau de performance observé le justifie.
- L'évaluation de cette performance est atteint en phase de suivi du processus attendu, sur des supports variés : feuille de temps, feuille d'évaluation de la performance.

L'élargissement du modèle de processus à un modèle de gestion de la connaissance

L'analyse de processus est sous ces éclairages le principal vecteur d'une Gestion de la connaissance appliquée. La cellule de développement y trouve un moyen de gérer chaque projet en référence à un modèle progressivement enrichi par son expérience directe de la performance observée.

L'ambition initiale de RUP se trouve ainsi singulièrement complétée.

De modèle postulé a priori, il devient l'expression justifiée, choisie, d'une gestion de la compétence disponible dans une organisation (entreprise dédiée au développement informatique) bien identifiée, au travers de ses ressources disponibles et des activités qu'elle (ou qu'elles – à titre individuel) maîtrisent.

L'utilisation de modèles de processus (de développement) qualifiés – définis à priori ou par réutilisation de processus déjà parcourus – éclaire et sécurise les travaux informatiques grâce à la normalisation de processus éprouvés.

L'analyse structurée des données quantitatives et qualitatives du processus

Si comprendre un processus nécessite de le décrire, le maîtriser nécessite de le mesurer.

Une approche documentaire (graphique ou textuelle) est limitée sur de nombreux points : mise à jour en cas d'évolution, fourniture de vues synthétiques, construction de vues sélectives (présence d'une ressource ou d'une contrainte, réponse à un événement), réalisation de calculs interactifs (somme des coûts, indicateurs de performance…).

Une approche mature des processus pose le besoin de gérer des données au delà des dessins, pour construire des tableaux de bord sur lesquels les coûts, la performance, les risques ou les règles de gestion peuvent être dynamiquement suivis.

La capacité à créer des tableaux d'analyse structurée de l'information représentative des éléments structurels, quantitatifs ou qualitatifs du processus est essentielle au cycle d'enrichissement du modèle de référence des développements.

Elle permet de coordonner les différentes visions qui peuvent être protées sur le projet (pour l'utilisateur : le niveau de satisfaction de spécifications, pour le gestionnaire: les coûts assumés, pour l'animateur : le niveau des compétences vérifiées…)

Elle permet également de contrôler les modes d'adaptation du référentiel. Le postulat d'un modèle a priori, tout particulièrement pour un projet de développement informatique, ne peut exister longtemps sans phase de contrôle de ses variations. Ce contrôle n'a de contenu que mesurable, et de sens qu'au travers de domaines de contrôle (structures, acteurs, quantités, qualités).

Le projet mis en œuvre au sein de BNP Paribas

Le projet d'évaluer la pertinence de RUP dans une approche prédictive et une démarche de capitalisation de l'expérience des équipes de développement nous a amené à parcourir les étapes suivantes.

La phase de modélisation de projet

Touts les points discutés dans les chapitres précédents résultent en une modélisation assez complexe des projets de développement. Chaque étape en est cependant assez clairement définie pour pouvoir être réalisée sans ambiguïté.

- Construction du référentiel général du processus de développement :
  - éclaircissement des termes adoptés dans les descriptions des processus de développement ;
  - déclinaison des processus généraux en activités opérationnelles ;
  - définition de l'architecture et des formats des livrables ;
  - choix des indicateurs d'évaluation de la performance des processus de développement (pérennité des produits – via leur documentation …-, atteinte des objectifs fonctionnels, respect des délais …) ;
  - définition des rôles intervenants dans les projets (MOA, MOE, Sponsor…).
- Définition de la structure du projet :
  - définition des modules du projet.
    Les modules sont des Groupes de fonctionnalités homogènes,
    - ◊ Soit parce qu'ils appellent des instances du référentiel qui partagent des caractéristiques communes
    - ◊ Soit parce qu'ils répondent à un même domaine fonctionnel Utilisateur

- Choix des modèles d'adaptation du référentiel
  Dans le cas où le développement d'une fonctionnalité particulière peut profiter d'un pattern de développement déjà expérimenté dans le passé (ce dernier étant lui même une modification du référentiel général de gestion de projet, et étant appelé en cela « pattern local ») alors la fonctionnalité concernée accédera au référentiel par l'intermédiaire du pattern local.
- Instanciation des structures du référentiel, selon les contextes pertinents.
  Un contexte pertinent consiste en la combinaison d'événements ou de cas qui modifie les propriétés, ou la structure du référentiel général.
- Adaptation directe du référentiel, au cas où la carence générale de ce dernier est révélée par le cas précis du projet en cours.
- Affectation de ressources aux rôles prévus par le référentiel. Le choix des ressources dépendant de chaque projet.

- Instanciation quantitative du projet :
  - Traduction des stratégies d'organisation retenues (modes « Fontain » ou « Workflow ») dans les profils de consommation des phases du référentiel.
  - Définition des temps et calendriers alloués aux différentes activités.

La phase de suivi de projet

L'instanciation du référentiel résulte en un schéma de processus qui peut être utilisé comme un pattern spécifique du projet en cours de développement.

Les annotations de ce pattern résultent en :

- un suivi des réalisations (temps, dates, indicateurs de performance) ;
- un aménagement du pattern prévisionnel (au cas où des activités prévues s'avèrent inutiles, ou inversement), sur ses différents objets manipulés (acteurs, activités, enchaînements, livrables, temps, calendrier).

Après analyse, et dès lors que le mode d'organisation d'un processus particulier a fait la preuve de sa performance, son mode d'instanciation du référentiel général peut devenir à son tour, selon le périmètre fonctionnel concerné :

- soit un pattern local ;
- soit une adaptation définitive (au niveau du pattern général) du référentiel de gestion de projet.

Les phases d'analyse de projet

À tout moment de la construction ou du suivi du projet, le recours à l'analyse du processus de développement est une nécessité.

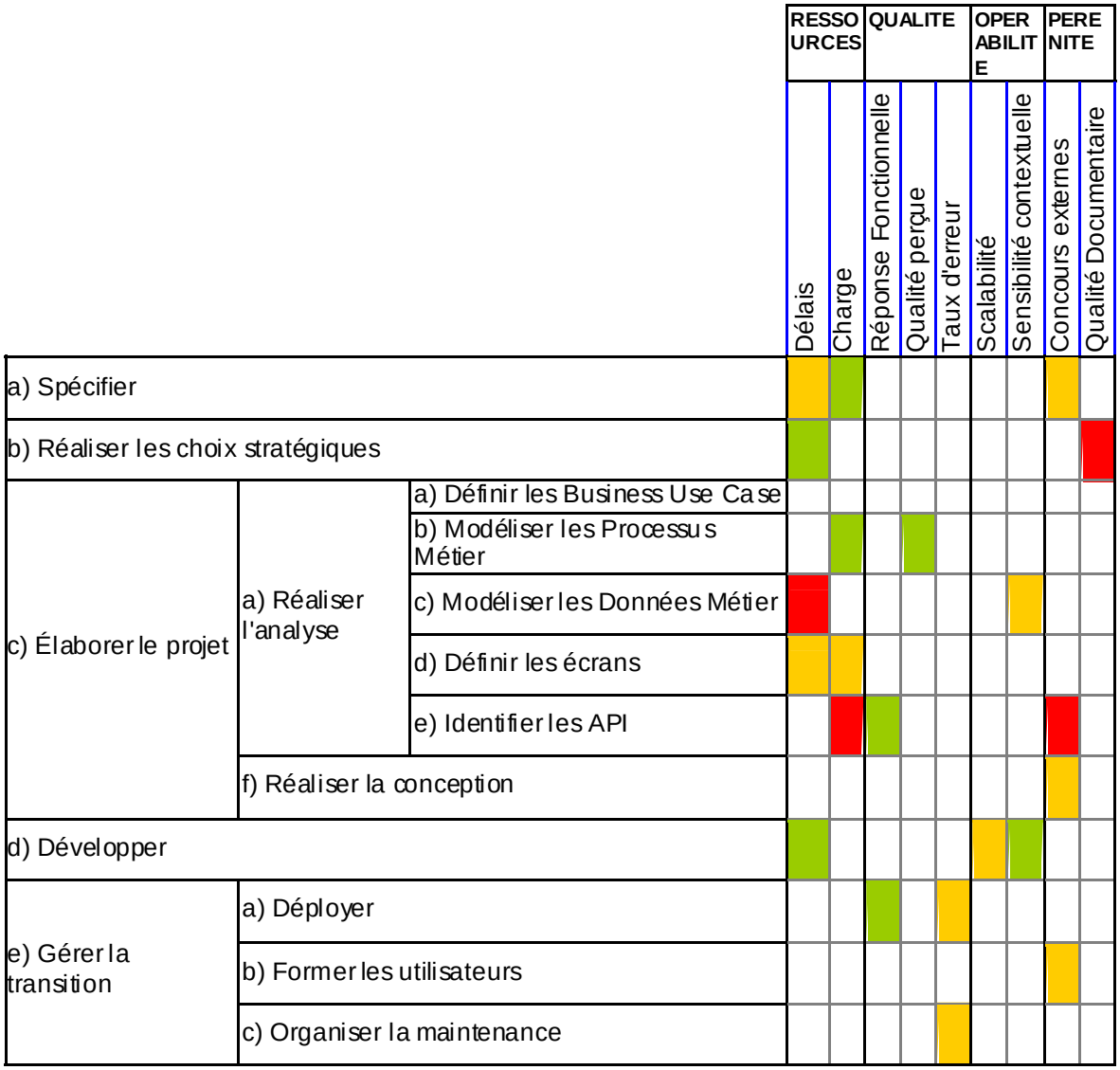
Sur le plan quantitatif, une telle analyse permet de repérer les incohérences de charge, ou d'attribution de responsabilités.

Sur le plan qualitatif, elle permet de repérer les points de dysfonction, ou les points de risque majeurs anticipés sur le déroulement du projet.

Cette analyse débouche sur des activités d'arbitrage ou de reporting.

Elle utilise l'apport des techniques multi-dimensionnelles, en termes de navigation comme en termes d'interactivité.

Le tableau suivant donne un exemple de tableau d'analyse multidimensionnelle, permettant de visualiser sous forme de tableau de bord l'état des différents indicateurs, synthétisés à des niveaux plus ou moins élevés du modèle RUP.



Philippe Denis
BNP PARIBAS
BP2S - Internet Reporting et Performance
Bruno Previtali
COTEBA Conseil