

Analyse : Le UP (Unified Process)

Introduction

L'analyse se consacre à l'étude des besoins décrits dans l'expression des besoins en les affinant et en les structurant.

L'objectif de l'analyse est d'accéder à une compréhension plus aigues des besoins et des exigences et d'en livrer une description facile à entretenir favorisant la structuration de l'ensemble du système y compris de son architecture.

La règle numéro1 dans l'expression des besoins (au niveau de l'étude des besoins) est d'utiliser le langage du CLIENT. **Les cas d'utilisation** constituent un excellent moyen d'exprimer les besoins.

Pour communiquer efficacement les fonctions du système au client, les cas d'utilisation doivent présenter les caractéristiques suivantes:

1. les cas d'utilisation doivent rester autant que possible indépendants les uns des autres. Il faut éviter de se perdre dans les détails quant aux interférences, à la concurrence et aux conflits entre cas d'utilisation.
2. les cas d'utilisation doivent être décrits dans le langage du client.
3. chaque cas d'utilisation doit être structuré de façon à constituer une spécification complète et intuitive des fonctionnalités.

L'objectif de l'analyse est de résoudre les problèmes restés flous lors de l'étude des besoins en faisant une analyse approfondie des besoins et des exigences. On peut utiliser le langage des développeurs pour décrire le système.

L'analyse permet de structurer les besoins et les exigences de telle sorte que leur compréhension, leur préparation, leur modification, leur réutilisation en soient facilité. Cette structure fondée sur les classes et la packages d'analyse est orthogonale à structure fournie par les besoins, fondée sur les cas d'utilisation.

Comparaison des modèles des cas d'utilisation et d'analyse.

Modèle des cas d'utilisation	Modèle d'analyse
Formulé dans le langage du client	Formulé dans le langage du développeur
Vue externe du système	Vue interne du système
Structuré par les cas d'utilisation: donne une structure à la vue externe	Structuré par les classes et les paquetages stéréotypés: donne une structure à la vue interne
Sert principalement de contrat entre le client et les développeurs. Sur ce que le système doit faire et ne doit pas faire	Sert principalement aux développeurs pour comprendre la forme que doit revêtir le système: la conception et l'implémentation
Susceptible de contenir des redondances voir des incohérences	Ne doit contenir aucune redondance ni incohérence
Exprime les caractéristiques du système, y compris les caractéristiques significatives sur le plan architectural	Esquisse la réalisation des caractéristiques dans le système y compris les caractéristiques significatives sur le plan architectural. Sert de premier brouillon du système.
Définit les cas d'utilisation qui seront ensuite analysés dans le modèle d'analyse.	Définit les réalisations des cas d'utilisation, chacune représentant l'analyse d'un cas d'utilisation du modèle des cas d'utilisation.

L'analyse en bref

L'analyse permet de structurer les besoins et les exigences de telle sorte que leur compréhension, leur préparation, leur modification, leur réutilisation en soient facilités. Cette structure fondée sur les classes et les packages d'analyse est orthogonale à la structure fournie par les besoins, fondée sur les cas d'utilisation.

Objectifs de l'analyse

- Un modèle d'analyse livre une spécification plus précise que celle des besoins y compris du modèle des cas d'utilisation
- Un modèle d'analyse s'écrit dans le langage des développeurs, et peut introduire un plus grand nombre de formalisme et servir de base à une réflexion sur le mécanisme interne du système.
- Un modèle d'analyse structure les besoins sous une forme qui en facilite la compréhension
- un modèle d'analyse peut-être envisagé comme une première ébauche du modèle de conception.
- L'analyse offre une vision d'ensemble du système, qu'il serait peut-être plus difficile d'obtenir en étudiant les résultats de la conception.

L'analyse n'est ni la conception ni l'implémentation.

La question que l'on pourrait se poser est la suivante: *«pourquoi on n'analyse pas directement les besoins et les exigences au moment de la conception et de l'implémentation du système.»*

Pendant la conception il faut modeler le système et en définir la forme. La conception répond à la question COMMENT? (Comment le système doit-il être réalisé.

Le modèle de conception décrit les objets, leurs relations, leurs interactions réciproques, l'allocation et la répartition sur les ressources physiques de traitement ainsi que les activités concurrentes, leur ordonnancement et leur synchronisation dans le système.

La conception va refléter la manière dont l'application (logiciel, système) va être réalisée, elle met en évidence la réflexion du concepteur (informaticien) par rapport à la réalisation. Dans la phase de conception, on ne peut pas faire abstraction des différents logiciels et matériels qui seront utilisés pour la réalisation.

Il est donc indispensable, avant de débiter la conception, d'avoir une vision précise et détaillée des besoins et des exigences dont le client ne se préoccupe pas un seul instant. Par ailleurs si l'on dispose d'une structure des besoins pouvant servir de façonnage (début de conception) du système c'est encore **MIEUX**

Rôle de l'analyse dans le cycle de vie du logiciel

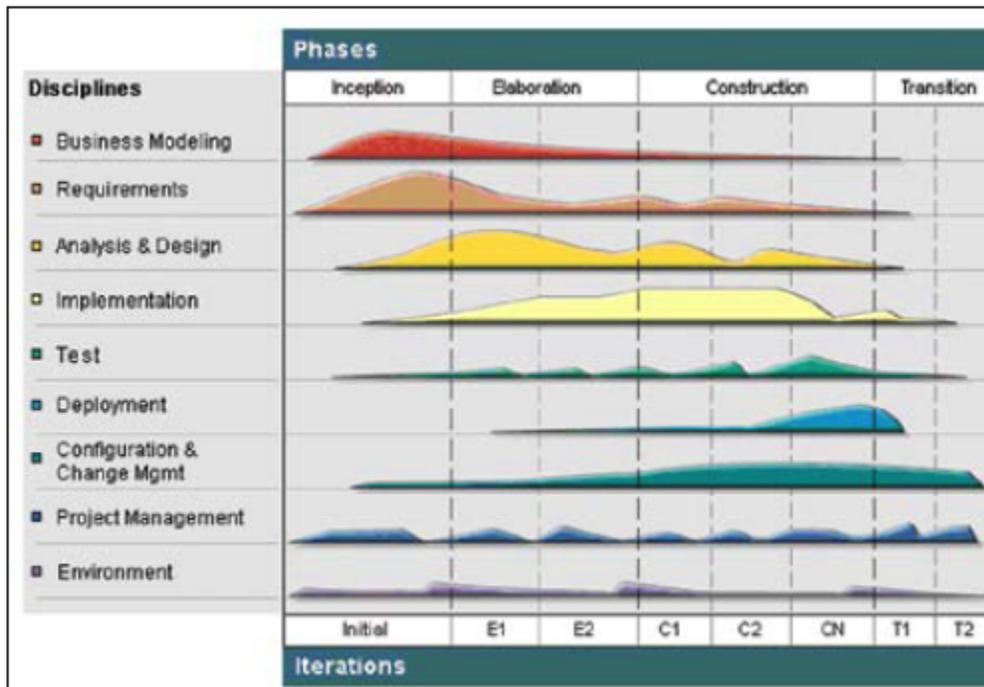


Figure 2: With the Rational Unified Process solution, you do some requirements planning, designing, implementing and testing of your applications at each phase of the solution delivery lifecycle, enabling you to address risk early and continuously.

Figure 1¹

L'analyse constitue le point de convergence des premières itérations de la phase d'élaboration. Elle contribue à la mise en place d'une architecture stable et favorise une compréhension approfondie des exigences.

Définition : le modèle d'analyse

Le modèle d'analyse est un ensemble de paquetage d'analyse. Chaque paquetage est composé lui-même de classe d'analyse et des réalisations-analyse des cas d'utilisation.

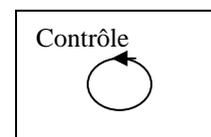
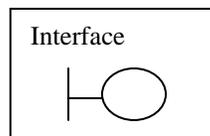
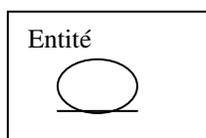
ftp://ftp.software.ibm.com/software/rational/web/datasheets/RUP_DS.pdf

Définition : classe d'analyse

1. Une classe d'analyse définit ou fournit rarement une interface en termes d'opérations et de signature. Son comportement est défini par des responsabilités à un niveau plus élevé, moins formel.
2. Une classe d'analyse définit des attributs bien que ceux-ci demeurent encore à un niveau plus élevé. Ces attributs sont reconnaissables à partir du domaine du problème.
3. Une classe d'analyse appartient à l'un des trois stéréotypes de base:
 - Classe frontière : on utilise une classe frontière pour modéliser l'interaction entre le système et ses acteurs. L'interaction implique souvent la réception et la présentation d'information de la part et en direction des utilisateurs.

Les classes frontières sont souvent une abstraction de fenêtres, de formulaires, d'interfaces de communication, de capteur, d'API etc....

- Classe entité : une classe entité sert à modéliser les informations de longue durée (données persistantes du système). Les classes entités font souvent apparaître une structure de données logique et contribuent à une meilleure compréhension des informations dont le système dépend.
- Classe de contrôle : elles représentent la coordination, le séquençement, les transactions et le contrôle d'autres objets. Elles servent souvent à encapsuler le contrôle associé à un cas d'utilisation spécifique. La dynamique du système est représentée par les classes contrôle.



Définition : réalisation de cas d'utilisation.

Une réalisation-analyse de cas d'utilisation est une collaboration au sein du modèle d'analyse décrivant la façon dont un cas d'utilisation donné est réalisé et exécuté en terme de classes d'analyse et d'interaction entre les objets qu'elles contiennent.

Une réalisation-analyse de cas d'utilisation offre une traçabilité directe vers un cas d'utilisation spécifique dans le modèle des cas d'utilisation. Une **réalisation de cas d'utilisation** présente une description textuelle des flots d'événements, des **diagrammes de classes d'analyse, et des diagrammes d'interaction illustrant la réalisation d'un scénario particulier du cas d'utilisation** en termes d'interaction entre les objets d'analyse.

Définition : package d'analyse

Les packages d'analyse offrent un moyen d'organiser les éléments du modèle d'analyse en **parties gérables**. «*Un package d'analyse peut représenter un sous-système*». Un package d'analyse peut se composer de classes d'analyse, de réalisation de cas d'utilisation et d'autres packages d'analyse.

Les packages d'analyse doivent être cohérents, (c'est-à-dire que leur contenu doit être fortement lié) , et faiblement couplés (c'est-à-dire leur dépendances mutuelles doivent être réduites le plus possible.)

Les packages peuvent présenter les caractéristiques suivantes:

- ☞ Les packages d'analyse peuvent marquer une séparation de questions d'analyse. Par exemple, dans un système il est possible que deux packages soient analysés séparément ou en parallèle par deux équipes ayant des compétences différentes.
- ☞ Les packages d'analyses doivent être issus des besoins fonctionnels du système.

Définition : description de l'architecture.

La description de l'architecture contient une vue architecturale du modèle d'analyse. On considère comme significatif sur le plan architectural les éléments d'informations du modèle d'analyse suivants:

- ☞ La décomposition du modèle d'analyse en package d'analyse avec leurs dépendances. Cette décomposition influence souvent sur les sous-systèmes des couches supérieures (conception et implémentation) et présente donc un intérêt pour l'architecture générale
- ☞ Les classes d'analyse:
 - les classes entités qui encapsulent un phénomène majeure du domaine du problème

- les classes frontières, qui encapsulent des interfaces de communication et les mécanismes d'interfaces utilisateur essentielles.
 - Les classes de contrôle qui encapsulent qui coordonnent les réalisations de cas d'utilisation significatives.
- ☞ Les réalisations de cas d'utilisation qui réalisent certaines fonctionnalités importantes; celles qui impliquent un grand nombre de classes d'analyse.