
Projet :	Plan Assurance Qualité
Document :	2UP_SPEC_DEV1
VERSION	1.00
Objet	Ce document a pour objectif de définir la démarche d'analyse et de conception objet ainsi les activités liées.

Auteur	Vérifié par:
Eric PAPET	Dominique MASSON

Validé par:	Guillaume FOURQUET			
Société:	IEP			
Fonction:	Gérant, ingénieur informaticien			
Date:				
Signature				

Liste de diffusion	

Gestion des modifications apportées au document

Auteur	Date Modification	Descriptions	Version document
Eric PAPET	12/12/2000	Création	1.00

SOMMAIRE

GESTION DES MODIFICATIONS APPORTÉES AU DOCUMENT	2
SOMMAIRE.....	3
INTRODUCTION.....	4
LES NIVEAUX D'ABSTRACTION :	4
<i>Le niveau contexte (activité de conceptualisation)</i>	4
<i>Le niveau domaine et application</i>	5
<i>Le niveau implémentation</i>	5
LA DEMARCHE	6
1. ETUDE PRÉLIMINAIRE	6
1.1 <i>Elaboration du cahier des charges</i>	6
1.2 <i>Identifier les acteurs</i>	6
1.3 <i>Identifier les messages</i>	6
1.4 <i>Modéliser le contexte</i>	6
2. CAPTURE DES BESOINS FONCTIONNELS.....	7
2.1 <i>Identifier les cas d'utilisation</i>	7
2.2 <i>Décrire les cas d'utilisation</i>	8
2.3 <i>Organiser les cas d'utilisation</i>	9
2.4 <i>Identifier les classes candidates</i>	9
2.5 <i>Valider et consolider</i>	10
3. CAPTURE DES BESOINS TECHNIQUES.....	11
3.1 <i>Spécification technique du point de vue matériel</i>	11
3.2 <i>Spécification d'architecture et influence sur le modèle de déploiement</i>	11
3.3 <i>Elaboration du modèle de spécification logicielle</i>	11
3.4 <i>Organisation du modèle de spécification logicielle</i>	11
3.5 <i>Développement des couches logicielles</i>	11
3.6 <i>Définition des concepts techniques</i>	11
3.7 <i>Description des cas d'utilisation technique</i>	11
4. DÉCOUPAGES EN CATÉGORIE.....	12
4.1 <i>Découpage en catégories</i>	12
5. DÉCOUPAGE EN INCRÉMENTS	12
6. DÉVELOPPEMENT DU MODÈLE STATIQUE (ANALYSE OBJET MÉTIER).....	13
6.1 <i>Affiner les classes</i>	13
6.2 <i>Affiner les associations</i>	13
6.3 <i>Ajouter les attributs</i>	13
6.4 <i>Ajouter les opérations</i>	13
7. DÉVELOPPEMENT DU MODÈLE DYNAMIQUE (ANALYSE OBJET MÉTIER).....	13
7.1 <i>Modélisation des cycles de vie des entités importantes</i>	13
7.2 <i>Diagramme de séquence d'appel entre les classes</i>	13
CONCLUSION	14

INTRODUCTION

Les Niveaux d'abstraction :

Le niveau contexte (activité de conceptualisation)

Correspond à une vision « boîte noire » du système, à l'exception de l'ébauche du modèle objet que représente les diagrammes d'objets participants.

- Comprendre le besoin du client :
 - Compléter, reformuler les besoins
 - Déterminer la frontière du système
- Préparer l'Analyse
 - Trouver les principales classes
 - Découper en catégorie ;
- Estimer les coûts et les délais ;

Entrées :

- cahiers des charges (analyse textuelle du cahiers des charges)

Sorties :

- Dossiers d'expression des besoins (étude préliminaire)
- Documents d'expression des besoins client..
- Diagramme de contexte, ensemble des Uses Cases
- Modèle d'objet initial (objets participants)
- Dictionnaire de données (permet de stocker des compléments textuels importants qui apportent des informations que les graphiques ne peuvent supporter.)

Objectif :

(Comprendre ce que veut le client, première découpe en package pour estimer des coûts et des délais)

- Description des acteurs
- Diagrammes de contexte
- Uses Cases :
 - Diagrammes de Uses Cases
 - Description textuelle
 - Diagrammes de d'Objets participant

Le niveau domaine et application

Permet de spécifier graduellement les caractéristiques des classes d'analyse (activité analyse objet).

Entrées :

- Dossiers d'expression des besoins
- Documents d'expression des besoins client..
- Diagramme de contexte, ensemble des Uses Cases
- Modèle d'objet initial
- Dictionnaire des données*

Sorties :

- Dossiers de spécification
- Diagrammes de classes
- Diagrammes dynamiques
- Dictionnaire complété

Objectif :

- Spécification graduelle des classes d'analyse
- Découpage en catégorie
- Diagrammes de classe.

Le niveau implémentation

Correspond au modèle de conception complet, et comprend toutes les classes techniques. C'est la solution d'implémentation, qui sera réalisée.

LA DEMARCHE

1. Etude préliminaire

Etude préliminaire ou (pré-étude) est la toute première étape de notre processus de développement. Elle survient à la suite d'une décision de démarrage de projet, et consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en considérant le système comme une boîte noire, afin d'étudier sa place dans le système métier plus global de l'entreprise. Après avoir identifié les acteurs qui interagissent avec le système, il sera développé un premier modèle UML de niveau contexte, pour pouvoir établir précisément les frontières fonctionnelles du système.

1.1 Elaboration du cahier des charges

Cette activité se est de la responsabilité de la maîtrise d'ouvrage.

1.2 Identifier les acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.

1.3 Identifier les messages

Un message représente la spécification d'une communication entre objets qui transporte de l'information avec intention de déclencher une activité chez le récepteur. La réception d'un message est normalement considérée comme un événement.

Cette notion de message est utilisé pour décrire les interactions de plus haut niveau entre les acteurs et le système.

1.4 Modéliser le contexte

Tous les messages (système <-> acteurs) identifiés précédemment peuvent être représentés de façon synthétique sur un diagramme de contexte dynamique.

On utilise un diagramme de collaboration UML.

- Le système utilisé est représenté par un objet central.
- Cet objet central est entouré par d'autres objets symbolisant les différents acteurs ;
- Des liens relient le système à chacun des acteurs.
- Sur chaque lien sont montrés les messages en entrée et en sortie du système, sans numérotation.

2. Capture des besoins fonctionnels

La technique des cas d'utilisation complète la capture des besoins fonctionnels de l'étude préliminaire en décrivant les différentes façon qu'auront les acteurs d'utiliser le système. La capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y. Elle formalise et détaille ce qui a été ébauché au cours de l'étude préliminaire.

Les éléments mis en jeu :

- Messages, acteurs, modèle de contexte dynamique.
- Acteurs principal, acteur secondaire.
- Cas d'utilisation.
- Fiche de description textuelle d'un cas d'utilisation.
- Scénario, enchaînement, diagramme d'activités.
- Inclusion, extension et généralisation de cas d'utilisation.
- Package de cas d'utilisation.
- Classes candidates, responsabilités, diagrammes de classes participantes.
- Traçabilité des cas d'utilisation avec les besoins fonctionnels.

2.1 Identifier les cas d'utilisation

Qu'est ce qu'un cas d'utilisation ?

Un cas d'utilisation modélise un service rendu par le système. Il exprime les interaction acteurs / système et apporte une valeur ajoutée notable à l'acteur concerné. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera. On exprimera donc des actions effectuées dans le cadre du métier de l'utilisateur, par opposition à des manipulations de l'application ou à des comportements technique.

Par exemple on ne développe pas la manipulation d'un IHM (écran), cela fera l'objet d'une étude d'ergonomie.

L'objectif est le suivant : l'ensemble des cas d'utilisations doit décrire exhaustivement les exigences fonctionnelles du système. Chaque cas d'utilisation correspond donc à une fonction métier du système, selon le point de vue d'un des acteurs.

Pour chaque acteur identifié durant l'étude préliminaire, il convient de :

- Rechercher les différentes intention métier avec lesquelles il utilise le système.
- Déterminer dans le cahier des charges les services fonctionnels attendus du système.
- On se servira des échanges de messages identifiés dans le modèle de contexte statique.
- Vérifier que chaque cas d'utilisation fournit une valeur ajoutée

REMARQUE :

- UN CAS D'UTILISATION N'EST NI UNE TRANSACTION, NI UNE FONCTION !
- DISTINGUEZ L'ACTEUR PRINCIPAL DES ACTEURS SECONDAIRES.

2.2 Décrire les cas d'utilisation

La technique recommandée pour identifier les cas d'utilisation à partir du modèle de contexte : considérez l'intention fonctionnelle de l'acteur par rapport au système dans le cadre de l'émission et de la réception de chaque message.

Exemple de tableau initiale :

CAS d'utilisation	Acteur principal, acteurs secondaires	Message(s) émis / reçus par les acteurs
Gestion des commandes	Réceptionniste	Emet : création, modification, annulation commande Reçoit : condition commande
	client	Emet : création client

Etablissez une première description textuelle succincte de chaque cas d'utilisation candidat. Chaque cas d'utilisation doit faire l'objet d'une définition a priori qui décrit l'intention de l'acteur lorsqu'il utilise le système et les quelques séquences d'actions qu'il est susceptible d'effectuer. Ces définitions servent à fixer les idées lors de l'identification des cas d'utilisation et n'ont aucun caractère exhaustif.

Conseil :

- Limiter à 20 le nombre de vos cas d'utilisation .
- Ne mélanger pas l'IHM et le fonctionnel

Comment structurer les fiches de cas d'utilisation ?

La fiche de description textuelle d'un cas d'utilisation n'est pas normalisée par UML. Nous préconisons pour notre part la structuration suivante :

Sommaire d'identification (obligatoire) : inclut titre, but, résumé, dates, version, responsables, acteurs.

Description des enchaînements (obligatoire) : Décrit les enchaînements nominaux, les enchaînements exceptionnels, les exceptions, mais aussi les pré conditions et les postpositions.

Besoins en IHM (optionnel) : Ajoute éventuellement les contraintes homme – machine. Contraintes non – fonctionnelles (optionnel) : fréquence, volumétrie, disponibilité, fiabilité, intégrité, confidentialité, performances, concurrence, ect ..

Complétez les description textuelles des cas d'utilisation par des diagrammes UML simples :

- Diagrammes d'activités
- Diagramme d'états
- Diagramme de séquences.

2.3 Organiser les cas d'utilisation

UML 1.3 définit trois types de relations standardisées entre cas d'utilisation :

- Une relation d'inclusion, formalisée par une dépendance stéréotypée : « include »
 - Exemple : le cas d'utilisation « authentification » est inclus par tous les autres cas d'utilisations.
- Une relation d'extension, formalisée par une dépendance stéréotypée « extend »
 - Exemple le cas d'utilisation « gestion des client » extend « gestion des commandes »
- Une relation de généralisation / spécialisation.

Qu'est ce qu'un packages :

Un packages en UML représente un espace de nomage qui peut contenir des éléments d'un modèle,

- Des diagrammes qui représentent les éléments du modèle,
- D'autres packages.

2.4 Identifier les classes candidates

Comme nous l'avons déjà mentionné, la définition des cas d'utilisation ne doit pas être une fin en soi !

La technique mise au point par « Ivar Jacobson » comporte deux objectifs principaux :

- Dialoguer avec le client sur son expression préliminaire des besoins, grâce à une description fonctionnelles de son niveau d'abstraction.
- Préparer la modélisation orientée objet en aidant à trouver les classes principales du futur modèle statique d'analyse.

Les premières classes candidates identifiées dans cette phase doivent être des concepts connus des utilisateurs du système, ce qu'on appelle couramment des objets métier. (Client, Commandes, Mission, Agence ect..).

L'analyse ajoutera dans un second temps des concepts « applicatifs », liés à l'informatisation(Profils utilisateurs, code barres ect..).

Qu'est – ce qu'une responsabilité : C'est une sorte de contrat ou d'obligation pour une classe, si une classes comporte plus de 5 responsabilités, elle doit être décomposée en plusieurs autres classes.

2.5 Valider et consolider

La révision des cas d'utilisation doit absolument inclure une phase de présentation aux futurs utilisateurs et poser les questions-clés ci-prés :

- Les frontières du système sont – elles bien définies ?
- Les acteurs sont-ils tous pris en compte ?
- Chaque cas d'utilisation est-il déclenché par un acteur ?
- Le niveau d'abstraction des cas d'utilisation est-il homogène ?
- Toutes les fonctionnalités du système sont – elles traitées ?

3. Capture des besoins techniques

3.1 Spécification technique du point de vue matériel

3.2 Spécification d'architecture et influence sur le modèle de déploiement

3.3 Elaboration du modèle de spécification logicielle

3.4 Organisation du modèle de spécification logicielle

3.5 Développement des couches logicielles

3.6 Définition des concepts techniques

3.7 Description des cas d'utilisation technique

REMARQUE : Cette étape débouche sur les activités de :

- conception générique ;
- conception préliminaire
- conception détaillée

La dernière activité est le codage et l'exécution des tests.

4. Découpages en catégorie

Ce chapitre traite du démarrage de l'analyse objet du système à réaliser.

En fin d'analyse des besoins, nous obtenons un découpage fonctionnel exprimé à travers des cas d'utilisation organisés dans le modèle de spécification fonctionnelle.

Pour passer à l'analyse, nous allons changer radicalement l'organisation du modèle et nous fonder sur les principes de l'approche orientée objet, notamment sur celui d'encapsulation. A cet effet, nous allons passer d'une structuration fonctionnelle via les cas d'utilisation et les packages de cas d'utilisation, à une structuration objet via les classes et les catégories.

4.1 Découpage en catégories

On regroupe les objets métiers par lien sémantique.

Dépendances entre catégories

On décrit les dépendances entre les catégories.

5. Découpage en incréments

L'analyse globale est finie. Nous allons maintenant découper le système à réaliser en incréments

Chaque incrément représente une unité du système à réaliser.

Chaque incrément va rentrer dans un cycle d'activité itératif :

- Analyse objet métier (spécifications détaillées) ;
- Conception générique ;
- Conception préliminaire ;
- Conception détaillée ;
- Réalisation ;
- Tests .

6. Développement du modèle statique (analyse objet métier)

6.1 Affiner les classes

6.2 Affiner les associations

6.3 Ajouter les attributs

6.4 Ajouter les opérations

7. Développement du modèle dynamique (analyse objet métier)

7.1 Modélisation des cycles de vie des entités importantes

7.2 Diagramme de séquence d'appel entre les classes

CONCLUSION

Dans ce document nous avons décrit la de réalisation de système informatique avec une démarche objet.

Nous avons abordé la partie fonctionnelle (analyse), la partie de conception n'est pas complètement décrit dans ce document.

Cette conception technique est importante dans la démarche objet, car elle permet de traiter la couche technique séparément du modèle métier, dans un objectif de la rendre réutilisable sur d'autre concept métier.

L'analyse globale permet de délimiter le système à réaliser, de le découper en incrément, qui permet de limiter les risques sur le projet.