

CHAPITRE 1

INTRODUCTION A UML

2

PROBLEMATIQUE DES SI EN ENTREPRISE

Objectif des entreprises

3

- Produire de la richesse
- Vendre des produits et/ou des services,
- Gérer les différents acteurs de son environnement :
 - Personnel et actionnaires,
 - Clients,
 - Fournisseurs,
 - Concurrents,
 - institutions,...

Objectif des entreprises

4

- Pour chaque acteur, des informations les concernant seront gérés,
- Toute ces activités s'inscrivent dans une organisation caractérisée par autant de processus.

Objectif des entreprises

5

- Un processus décrit un ensemble d'interaction entre différents acteurs en vue de réaliser un objectif bien défini.
- Exemple:
 - Processus achat,
 - Processus vente,
 - Processus gestion personnel...

Importance des systèmes informatisés dans les entreprises

6

- Le système informatisé de l'entreprise est l'automatisation des processus métiers de l'entreprise (workflow);
- C'est un gage de compétitivité face à la mondialisation,
- Information stratégique (aide à la décision, survie).

Les défis actuels

7

- Investissement et usage des SI;
- Modification des stratégies et nouvelle organisation du travail;
- Tirer avantage des SI pour faire face à la mondialisation;
- Mettre en œuvre des infrastructures robustes, souples et évolutives;
- Contrôle responsabilité, éthique et sécurité.

importance des SI dans l'organisation de l'entreprise

8

- En entreprise il existe 3 niveaux de système d'information:
- 1 Niveau opérationnel (cadres opérationnel),
- 2 niveau gestion (cadres intermédiaires),
- 3 niveau stratégique (dirigeants)

Nouveaux défis

9

- En fonction de l'activité de l'entreprise différents SI seront gérés:
- Vente marketing,
- Fabrication et logistique,
- Finance comptabilité,
- Ressources humaines....,
- Existe différents SI pour chaque activité (notion processus, composantes si,...)

10

GENERALITE SUR UML

Introduction à UML

11

- L'objectif d'une application informatique : produire un **code** conforme **au besoin utilisateurs**.
- **Complexité** sans cesse croissante des **applications** informatiques modernes.
- Nécessité **d'encadrer** le développement d'une **démarche** rigoureuse pour **réduire les risques** (erreurs, incompréhension,...).

Introduction à UML

12

- **UML** est un outil **standard** permettant d'assister le(s) concepteur(s) de l'application dans les activités de développement.
- **UML fusion par OMG** (Object Management Group) des langages de de modélisation objet suivants:

Introduction à UML

13

- **Booch** de Grady Booch;
 - **OMT** de James Rumbaugh;
 - **OOSE** d'Ivar Jacobson.
- Nous verrons dans ce chapitre:
- Les **modèles et le processus de développement**;
 - le **processus unifié**;
 - Les différents **modèles d'UML**.

1 NOTION DE MODELE ET DE PROCESSUS DE DEVELOPPEMENT

14

- Les **modèles d'UML** représentés sous forme de pictogrammes sont de puissants **outils de communication**.
- Le **modèle d'application** informatique représente l'information nécessaire à la production et à l'évolution des applications.

1 NOTION DE MODELE ET DE PROCESSUS DE DEVELOPPEMENT

15

- Un modèle pourra spécifier:
 - Le **point de vue des utilisateurs** (fonctionnalités offertes aux utilisateurs) ;
 - Le **point de vue des données** (les données manipulées par l'application);
 - Le **point de vue technique** (architecture logicielle).

Liste de documents nécessaire pour réaliser une application (Source: UML 2 pour les développeurs)

16

Tableau 1.1 Liste non exhaustive des documents nécessaires à la réalisation d'une application

Document	Fonction
Documentation utilisateur	Précise la façon dont on peut utiliser l'application. Un tel document peut aussi contenir une section décrivant la façon d'installer l'application. Ce document est rédigé par l'équipe de développement et est destiné aux utilisateurs de l'application.
Documentation des services offerts par l'application	Présente une vision macroscopique de l'application et liste les fonctionnalités réalisées par l'application. Ce document est rédigé par l'équipe de développement et est destiné aux utilisateurs de l'application.
Documentation d'architecture de l'application	Précise la façon dont l'application est structurée en terme de gros composants. Ce document est rédigé par les architectes et est destiné à tous les membres de l'équipe de développement.
Documentation des logiciels nécessaires à l'utilisation de l'application	Dresse la liste des logiciels nécessaires à l'installation et à l'exécution de l'application. Ce document est rédigé par l'équipe de développement et est destiné aux administrateurs, afin qu'ils puissent mettre l'application en production.
Documentation des tests effectués	Liste l'ensemble des tests qui ont été effectués sur l'application. On peut de la sorte tester à nouveau l'application après l'avoir modifiée et vérifier ainsi qu'elle ne génère pas d'erreurs sur certains scénarios d'utilisation (scénarios couverts par les tests). Ce document est rédigé par l'équipe de développement et est destiné aux développeurs futurs.
Documentation de la conception de l'application	Précise la conception de l'application (en terme de classes, par exemple). Ce document est rédigé par l'équipe de développement et est destiné aux développeurs.
Spécification des besoins	Précise les besoins exprimés par le futur utilisateur de l'application, aussi appelé client. Ce document est rédigé par le client et est destiné à l'équipe de développement.

1 NOTION DE MODELE ET DE PROCESSUS DE DEVELOPPEMENT

17

- Les modèles UML sont souvent associés à **un processus de développement**.
- Ceci permet de garantir une meilleure **conduite de projet Informatique**.
- Le processus de développement permettra de gérer **le temps, les coûts et acteurs** du projet.

1 NOTION DE MODELE ET DE PROCESSUS DE DEVELOPPEMENT

18

- Dans le processus de conception de logiciel, plusieurs **activités** pourront être intégrées.
- Pour chaque **activité** des modèles seront produits.
- Parmi ces activités nous distinguerons:
 - les activités de **spécification des exigences**;
 - les activités **d'analyse**;
 - et les activités de **conception**.

1 NOTION DE MODELE ET DE PROCESSUS DE DEVELOPPEMENT

19

- **Les activités de spécification des exigences:**
- Dans cette activité, un modèle **de niveau contexte** est développé, afin de préciser les **frontières fonctionnelles du système.**

1 NOTION DE MODELE ET DE PROCESSUS DE DEVELOPPEMENT

20

les activités d'analyse

- Dans cette activité le modèle correspond à **une vue de l'intérieur du système**.
- Les objets ici représentent une **abstraction des concepts** manipulés par les utilisateurs.
- Pour une meilleure compréhension du système, il faudra :
 - envisager de décrire dans les **modèles une structure statique ;**
 - et le **comportement dynamique** de ces objets.

1 NOTION DE MODELE ET DE PROCESSUS DE DEVELOPPEMENT

21

les activités de conception

- dans cette activité le modèle correspond aux **concepts informatiques** qui sont utilisés par les outils, les langages ou les plates-formes de développement.
- Le modèle sert ici à **étudier**, **documenter**, **communiquer** et **anticiper** une solution.

2 DESCRIPTION DU PROCESSUS UNIFIE

22

- Le **Processus Unifié** (UP, pour Unified Process) est :
 - « itératif et incrémental »;
 - « centré sur l'architecture »;
 - « conduit par les cas d'utilisation »;
 - « et piloté par les risques » .

2 DESCRIPTION DU PROCESSUS UNIFIE

23

Un processus itératif et incrémental

- le projet est découpé en **itérations** de courte durée (environ 1 mois) ;
- À la fin de chaque itération, **une partie exécutable** du système final est produite.

Un processus Centré sur l'architecture

- tout système complexe doit être décomposé en parties **modulaires**.
- La modularité permet de garantir une **maintenance** et une **évolution** facilitées.

Un processus Conduit par les cas d'utilisation :

- Les **besoins** et des **exigences** des utilisateurs sont au centre du projet.
- Les **cas d'utilisation** du futur système sont **identifiés, décrits** avec précision et **priorisés**.

2 DESCRIPTION DU PROCESSUS UNIFIE

26

Un processus Piloté par les risques

- Identifier très tôt les **risques** majeurs du projet;
- **Lever** le plus rapidement possible ces risques identifiés.
- Les mesures à prendre dans ce cadre déterminent **l'ordre des itérations**.

2 1 LES PHASES ET LES DISCIPLINES DU PROCESSUS UNIFIE

27

- Le processus unifié comporte quatre phases:
 - Initialisation;
 - Élaboration;
 - Construction;
 - et transition.

La phase d'initialisation

- conduit à définir la « **vision** » du projet:
 - sa **portée**,
 - sa **faisabilité**,
 - son **business plan**,
- Ceci permettra de pouvoir décider au mieux de sa poursuite ou de son arrêt.

2 1LES PHASES ET LES DISCIPLINES DU PROCESSUS UNIFIE

29

La phase d'élaboration

- Trois objectifs principaux attendus en parallèle :
 - **identifier** et **décrire** la majeure partie des **besoins** des utilisateurs,
 - **construire** (et pas seulement décrire dans un document) **l'architecture** de base du système,
 - **lever** les **risques** majeurs du projet.

2 1 LES PHASES ET LES DISCIPLINES DU PROCESSUS UNIFIE

30

La phase de construction

- consiste surtout à **concevoir** et **implémenter** l'ensemble des éléments opérationnels (autres que ceux de l'architecture de base).
- C'est la phase la plus consommatrice en ressources et en effort.

2 1 LES PHASES ET LES DISCIPLINES DU PROCESSUS UNIFIE

31

La phase de transition

- permet de faire passer le système informatique des mains des **développeurs** à celles des **utilisateurs finals**.
- Les mots-clés sont :
 - conversion des données;
 - formation des utilisateurs;
 - Déploiement;
 - béta-tests.

3 LES MODELES UML

32

- UML 2.3 s'articule autour **de quatorze (14) types de diagramme.**
- Chaque diagramme permet de représenter des **concepts** particuliers d'un système logiciel.
- Ces types de diagrammes sont répartis en deux grands groupes :
 - Les diagrammes **comportementaux**;
 - Les diagrammes **structurels**.

3 1 LES DIAGRAMMES COMPORTEMENTAUX

33

- Diagramme de **cas d'utilisation** : Il montre les interactions fonctionnelles entre les acteurs et le système étudié.
- Diagramme de **séquence** : Il montre la séquence verticale des messages passés entre objets au sein d'une interaction.

3 1 LES DIAGRAMMES COMPORTEMENTAUX

34

- Diagramme **d'activité** : Il montre l'enchaînement des actions et décisions au sein d'une activité.
- Diagramme **de vue d'ensemble des interactions** : Il fusionne les diagrammes d'activité et de séquence.

3 1 LES DIAGRAMMES COMPORTEMENTAUX

35

- Diagramme de **communication** : Il montre la communication entre objets dans le plan au sein d'une interaction.
- Diagramme **d'états** : Il montre les différents états et transitions possibles des objets d'une classe.

3 1 LES DIAGRAMMES COMPORTEMENTAUX

36

- Diagramme de **temps** : Il fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps

3 2 LES DIAGRAMMES STRUCTURELS

37

- Le diagramme de **classes** : Il montre les éléments statiques de base (classes, associations, interfaces, attributs, opérations, généralisations, etc.)
- Le diagramme **d'objets** : Il montre les instances des éléments structurels et leurs liens à l'exécution.

3 2 LES DIAGRAMMES STRUCTURELS

38

- Le diagramme de **packages** : Il montre l'organisation logique du modèle et les relations entre packages.
- Le diagramme de **structure composite** : Il montre l'organisation interne d'un élément statique complexe.

3 2 LES DIAGRAMMES STRUCTURELS

39

- Le Diagramme de **composants** : Il montre des structures complexes, avec leurs interfaces fournies et requises.
- Diagramme de **déploiement** : Il montre le déploiement physique des modules sur les ressources matérielles.
- Le Diagramme de **profile**: permet de spécialiser, de personnaliser pour un domaine particulier un meta-modèle de référence d'UML