

Projet :	Plan Assurance Qualité
Document :	2UP_ARCHI_DEV1
VERSION	1.00
Objet	Ce document a pour objectif de définir la démarche TOW TRACK UNIFIED PROCESS.

Auteur	Vérfié 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 du document	1.00

SOMMAIRE

GESTION DES MODIFICATIONS APPORTÉES AU DOCUMENT	2
SOMMAIRE.....	3
INTRODUCTION AUX PROCESSUS UNIFIÉS.....	4
PROCESSUS DE DÉVELOPPEMENT LOGICIEL	4
PROCESSUS UNIFIÉ (UNIFIED PROCESS).....	5
LE PROCESSUS 2TUP.....	6
BRANCHE FONCTIONNELS (GAUCHE):	7
BRANCHE ARCHITECTURE TECHNIQUE (DROITE) :	7
BRANCHE CONCEPTION (MILIEU) :	7
ANALYSE :	8
CAPTURE DES BESOINS TECHNIQUES :	8
CONCEPTION GÉNÉRIQUE :	8
CONCEPTION PRÉLIMINAIRE :	8
CONCEPTION DES CLASSES :	8

Introduction aux processus unifiés

La complexité croissante des systèmes informatiques a conduit les concepteurs à s'intéresser aux méthodes.

On a comptabilisé en 1994 jusqu'à 50 méthodes.

Chaque méthode se définit par une notation et un processus spécifique.

UML a ouvert le terrain en fusionnant la notation.

Il reste cependant à définir le processus pour réellement capitaliser des règles dans le domaine du développement logiciel.

Les groupe de travail UML (les 3 amigos) ont donc travaillé à unifier non pas les processus, mais plus exactement les meilleures pratiques de développement objet.

Ces processus se distingueront par le générique < UNIFIED PROCESS >.

Nous développerons ici le < Two Track Unified Process >.

Processus de développement logiciel

Un processus définit une séquence d'étapes, en partie ordonné, qui concoure à l'obtention d'un système logiciel ou à l'évolution d'un système existant.

Pour produire des logiciels de qualité, qui répondent aux besoins des utilisateurs dans des temps et des coûts prévisibles.

On découpe le processus en deux axes :

- L'axe de développement technique, qui se concentre principalement sur la qualité de production.
- L'axe de gestion du développement, qui permet la mesure et la prévision des coûts et des délais.

Processus Unifié (Unified Process)

Un processus unifié est un processus de développement logiciel construit sur UML.

Il est :

- Itératif ;
- Centré sur l'architecture ;
- Conduit par les cas d'utilisation et piloté par les risques.

La gestion d'un tel processus est organisée suivant 4 phases :

- Pré – étude ;
- Elaboration ;
- Construction ;
- Transition.

Les activités de développement sont définies par 5 workflows fondamentaux qui décrivent :

- La capture des besoins ;
- L'analyse ;
- La conception ;
- L'implémentation ;
- Le test.

Tout processus UP répond aux caractéristiques ci à prés :

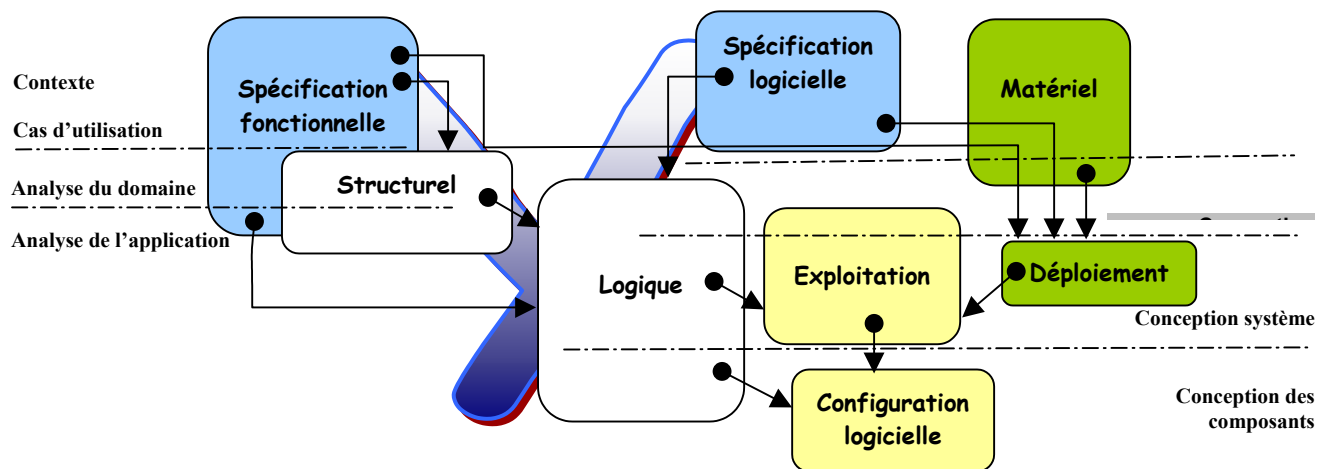
- Il est incrémental. La définition d'incrément de réalisation est en effet la meilleure pratique de gestion des risques techniques et fonctionnels.
- Il est piloté par les risques. Les causes majeures d'échec d'un projet logiciel doivent être écartées en priorités ; les deux principales causes sont l'incapacité de l'architecture technique à répondre aux contraintes opérationnelles et l'inéquation du développement aux besoins utilisateurs.
- Il est construit autour de la création et de la maintenance d'un modèle, plutôt que de la production de montage de documents.
- Il est itératif. Chaque itération porte sur un niveau d'abstraction de plus en plus précis.
- Il est orienté composant.
- Il est orienté utilisateur.

Le processus 2TUP

Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise.

Ce processus suit deux chemins.

- Architecture fonctionnel
- Architecture technique



Branche fonctionnels (gauche):

- Capture des besoins fonctionnels, qui produit le modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie, au plus tôt le risque de produire un système inadapté aux utilisateurs
- L'analyse, qui consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en terme de métier.

Branche architecture technique (droite) :

- La capture des besoins techniques, qui recense toutes les contraintes sur les choix de dimensionnant et la conception du système. Les outils et le matériels sélectionnés ainsi que la prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique).
- La conception générique, qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspect fonctionnel. Elle a pour objectif de d'uniformiser et de réutiliser les mêmes mécanismes pour tout un systèmes. L'architecture technique construit le squelette du système, son importance est telle qu'il est conseillé de réaliser un prototype.

Branche conception (milieu) :

- La conception préliminaire, qui représente une étape délicate, car elle intègre le modèle d'analyse fonctionnelle dans l'architecture technique de manière à tracer la cartographie des composants du système à développer.
- La conception détaillée, qui étudie ensuite comment réaliser chaque composant.
- L'étape de codage, qui produit ses composants et teste au fur et à mesure les unités de code réalisées.
- L'étape de recette, qui consiste enfin à valider les fonctionnalités du système développer.

Capture des besoins fonctionnels :

- Le niveau contexte a pour objet de définir la frontière fonctionnelle entre le système considéré comme une boîte noire et son environnement.
- Le niveau « cas d'utilisation » définit ensuite les activités attendues des différents utilisateurs par rapport au système toujours envisagé comme une boîte noire.

Analyse :

- Ouvre le système pour établir la structure des objets utilisés. Le modèle d'analyse du domaine définit la structure et le comportement des objets connus dans le métier des utilisateurs du système.
- Le modèle d'analyse de l'application y rajoute, suivant le même processus les objets qui sont connus des utilisateurs, dans le cadre de la mise en application de leurs besoins.

Capture des besoins techniques :

- Le modèle d'analyse technique établit des couches logicielles et y spécifie les activités techniques attendues.

Conception générique :

- Le modèle de conception technique définit les composants qui, délivrant des services techniques, assurent la réponse aux exigences opérationnelles du système.

Conception préliminaire :

- Le modèle de conception système organise le système en composants, délivrant les services techniques et fonctionnels. Ce modèle regroupe les informations de la branche de droite et de la branche de gauche. Elle peut être considérée comme la transformation du modèle d'analyse par projection des classes d'analyse sur les couches logicielles.

Conception des classes :

- Le modèle de conception des composants fournit l'image prête à fabriquer du système complet.