

lab-sticc.univ-brest.fr/~babau/

Méthodes de conception pour les Systèmes d'Information (UP)

Jean-Philippe Babau

Département Informatique, UFR Sciences, UBO
Laboratoire Lab-STICC

UBO

jean-philippe.babau@univ-brest.fr

2

UBO

Plan

- " Introduction à UP
- " Phases
- " Activités
- " Processus
- " Principes généraux
- " Autres approches

jean-philippe.babau@univ-brest.fr 3

UBO

Unified Software Development Process / Unified Process (UP)

- " **Beaucoup de méthodes**
 - . Liées à des outils
 - . A l'adaptation à UML (comme langage de notation) de méthodes existantes
 - . Liées au domaine (SI, Web, embarqué, critique, δ)
 - . Finalement, autant de méthodes que de concepteurs / projets δ
- " USDP : Rumbaugh, Booch, Jacobson (les concepteurs de UML)
 - . Purement objet
 - . Méthode / processus
 - . Regroupement de bonnes (meilleures?) pratiques de développement

jean-philippe.babau@univ-brest.fr

UBO

UP

- " Un processus pour gérer un projet logiciel de bout en bout
- " Un processus pour
 - . Spécifier les artefacts à produire
 - . Définir l'organisation des activités de l'équipe
 - . Diriger les tâches de chaque individu et de l'équipe dans son ensemble
 - . Proposer des critères pour le contrôle de produits et des activités de l'équipe
- " Regroupement de bonnes pratiques
 - . Non figé et générique (hautement adaptable : individus, cultures, ...)
 - . Gestion du changement
 - . Vérification en continu de la qualité
 - . Modélisation visuelle
 - . Approche itérative
 - . Centré sur les besoins et sur l'architecture
 - . Orienté vers la diminution des risques

jean-philippe.babau@univ-brest.fr

UBO

UP

- " UP définit un enchaînement d'activités
 - . Réalisées par un ensemble de travailleurs (rôle, métier) ayant pour objectif la réalisation d'artefacts à partir de besoins du client pour favoriser la réutilisation lors d'évolutions dans les besoins
- " UP est un cadre général de processus
 - . Mise en œuvre dans RUP, TTUP, UP agile, etc.
 - . Adaptation du processus au contexte de l'entreprise et du projet
 - " Connaissances et taille de l'entreprise
 - " Taille et type de projet
 - . Choisir une implémentation de UP correspond aux besoins du projet
 - " Expertise projet / processus

jean-philippe.babau@univ-brest.fr

UBO

Plan

- " Introduction
 - . Pourquoi une méthode ?
 - . Objectifs d'une méthode
- " UP
 - . Introduction
 - . **Phases**
 - . Activités
 - . Processus
 - . Principes généraux
 - . Autres approches

jean-philippe.babau@univ-brest.fr 7

UBO

UP : phases

- " **Un cycle de vie produit une version**
- " **Un cycle de vie = 4 phases**
 - . Lancement
 - . Elaboration
 - . Construction (50% du cycle de vie : développement principal et mise au point)
 - . Transition
- " **Une phase se découpe en itérations successives**
 - . Approche incrémentale
- " **Une phase et une itération se termine par un jalon**
 - . Evaluation
 - . Prise de décision

jean-philippe.babau@univ-brest.fr 8

UBO

La méthode UP : phases et activités

" 4 phases, plusieurs itération par phase
 " Dans chaque phase /itération : toutes les activités de développement

jean-philippe.babau@univ-brest.fr 9

UBO

Les phases UP

" **Lancement : étude prospective**

- . Quel est le domaine d'application?
- . Que doit faire le système ?
- . Quels sont les risques ?
- . Quel sont les coûts, les délais, les ressources et les moyens pour le projet ?
- . Comment le planifier ?
- . Jalon : « objectifs du projet »
- . Prise de décision: Est-ce qu'on accepte le projet ?

" **Elaboration : mise en place d'un prototype**

- . Spécification de la plupart des cas d'utilisation
- . Conception de l'architecture de référence (squelette du système)
- . Mise en %uvre de cette architecture (CU critiques, <10 % des besoins)
- . Planification complète
- . Les besoins sont-ils considérés? L'architecture stable? Les risques sont contrôlés ?
- . Jalon : « architecture »
- . Prise de décision: Est-ce qu'on peut passer à la réalisation?

jean-philippe.babau@univ-brest.fr 10

UBO

Les phases UP

- " **Construction : développement**
 - . Développement par itérations / incréments
 - " Pour aboutir à une architecture stable
 - . Le produit contient tout ce qui avait été planifié
 - " Il peut rester quelques erreurs non détectées
 - . **Jalon : «opérationnel»**
 - . **Prise de décision: le produit est-il suffisamment correct pour être installé chez un client ?**
- " **Transition : mise en place et suivi**
 - . Produit livré (version bêta)
 - . Correction du reliquat d'erreurs
 - . Essais et améliorations du produit
 - . Formation des utilisateurs
 - . Installation de l'assistance en ligne
 - . Le produit est-il satisfaisant ? Les manuels sont-ils prêts ?
 - . **Jalon : « release du produit »**
 - . **Prise de décision: le projet est fini ?**

jean-philippe.babau@univ-brest.fr 11

UBO

Plan

- " **Introduction**
 - . Pourquoi une méthode ?
 - . Objectifs d'une méthode
- " **UP**
 - . Introduction
 - . Phases
 - . **Activités**
 - . Processus
 - . Principes généraux
 - . Autres approches

jean-philippe.babau@univ-brest.fr 12

UBO

UP : activités

- “ Le développement est découpé en activités
 - . Pour chaque phase
 - . Pour chaque itération (au sein d'une phase)
- “ Activités complémentaires
 - . Gestion de projet : suivi du projet (personnes, budget, contrats, planning)
 - . Environnement : fournir un environnement technique à l'équipe pour le suivi des processus et pour le développement (modélisation et codage)
 - . Gestion de configuration et demandes de changement : gestion des versions (SVN, CVS)
- “ Activités principales
 - . Modélisation métier (*Business Modeling*)
 - . Expression des besoins (*Requirement Analysis*)
 - . Analyse et conception (*Analysis and Design*)
 - . Réalisation et tests (*Implementation and Tests*)
 - . Déploiement (*Deployment*)

jean-philippe.babau@univ-brest.fr 13

UBO

UP : activités

- “ Modélisation métier
- “ Expression des besoins
- “ Analyse
- “ Conception
- “ Réalisation
- “ Tests
- “ Déploiement

The diagram shows a flow from 'Besoins' (Requirements) to 'Code'. The process involves several modeling stages: 'Diagramme de cas d'utilisation' (Use Case Diagram), 'Diagrammes de séquence système' (System Sequence Diagrams), 'Diagrammes d'interaction' (Interaction Diagrams), 'Modèle du domaine' (Domain Model), 'Diagramme de classes participantes' (Participant Class Diagram), 'Diagramme de classes de conception' (Design Class Diagram), 'Maquette de l'UIH' (UIH Prototype), and 'Diagramme d'activité de navigation' (Navigation Activity Diagram). Arrows indicate the progression from these models towards the final 'Code' output.

jean-philippe.babau@univ-brest.fr 14

UBO

UP : modélisation métier

- “ Appréhender le domaine d'application
 - . Connaître les acteurs qui vont intégrer/manipuler le logiciel
 - “ Structure et fonctionnement de l'entreprise
 - “ Connaissances techniques, habitudes, etc.
 - . Compréhension du vocabulaire
 - . Compréhension des frontières du logiciel
- “ Capturer les spécificités du domaine
 - . Données métier
 - . Processus métier
- “ Modèle de l'entreprise ou du domaine
 - . Diagramme de classe UML
 - . Diagramme EMF
- “ Modèles des processus de l'entreprise
 - . Diagrammes d'activité UML
 - . Profil UML SPEM

jean-philippe.babau@univ-brest.fr 15

UBO

UP : modélisation métier

- “ Positionner le projet
 - . Définir les frontières du projet
 - “ Les éléments du métier concernés par le projet -> modèle du domaine
 - . Préciser l'intégration du projet dans un processus global
 - “ Un projet : informatiser UNE activité au sein de l'entreprise
- “ Définir l'impact du projet
 - . Changement d'habitudes
 - . Modification du processus général de l'entreprise
 - . Identifier les acteurs concernés
 - . Interactions avec les autres systèmes d'information

Modèle du domaine

jean-philippe.babau@univ-brest.fr 16

UBO

UP : expression des besoins

- “ Le modèle des CU représente le système vu de l'extérieur, son insertion dans l'organisation, ses frontières fonctionnelles
- “ Identification et représentation des besoins
 - . Description des acteurs et des cas d'utilisation
 - “ Diagramme de cas d'utilisation (CU)
- “ Spécification détaillée des besoins
 - . Détail des cas d'utilisation
 - “ Diagrammes de séquence
- “ Maquette de l'IHM de l'application (non couvert par UML)

The diagram illustrates the relationship between requirements and their implementation. A central cloud labeled 'Besoins' (Requirements) has three arrows pointing outwards: one to a 'Diagramme de cas d'utilisation' (Use Case Diagram), one to a 'Diagramme de séquence système' (System Sequence Diagram), and one to a 'Maquette de l'IHM' (UI Prototype).

jean-philippe.babau@univ-brest.fr

17

UBO

UP : expression des besoins

- “ Appréhender les besoins fonctionnels
 - . En décrivant les récits d'utilisation
 - . En précisant les objectifs du projet
- “ Les acteurs
 - . Acteur principal : celui qui déclenche le cas d'utilisation
 - . Personnes ou autres systèmes d'information qui interagissent avec le système
 - . Limites du système
- “ CU
 - . Créer de bons CU (cf. cours CU)
 - . Limiter le niveau de décomposition fonctionnel
 - “ Danger d'une décomposition fonctionnelle des CU qui reviendrait à une conception fonctionnelle « à l'ancienne »
 - . Ordonner les CU par priorité et niveau de risque

jean-philippe.babau@univ-brest.fr

18

UBO

UP : expression des besoins

- " Des rôles et des activités
 - . Analyste du domaine
 - " Modèle du domaine / du métier
 - " Modèle des CU / acteurs
 - " Glossaire
 - . Spécifieur de CU
 - " CU détaillés
 - . Concepteur d'interface utilisateur
 - " Maquette/prototype
 - . Architecte CU
 - " Structuration du modèle des CU

jean-philippe.babau@univ-brest.fr 19

UBO

UP : analyse

- " Elaboration d'un modèle « idéal »
- " Conception système
 - . Diagramme de classe
 - . Modèle du domaine : -> classes métier
 - . IHM : -> classes dialogue
 - . Modèle d'interaction métier /IHM : -> classes de contrôle
- " Modèles des interactions (GUI)
 - . Diagramme d'activité
 - . Enchaînement des menus, fenêtres, ⏪

jean-philippe.babau@univ-brest.fr 20

UBO

UP : analyse

- “ Identifier les packages d'analyse
 - . Regroupement logique indépendant de la réalisation
 - . Point de départ du découpage en sous-systèmes
- “ Identifier les classes de référence
 - . Architecture de référence
 - . Modèle de classes constituant l'essentiel du domaine
 - “ à partir du modèle du domaine / métier
 - . 3 stéréotypes de classe : frontière, contrôle, entité
- “ Identifier les exigences particulières communes
 - . Distribution, sécurité, persistance, tolérance aux fautes
 - . Les rattacher aux classes et cas d'utilisation
- “ Réalisation des CU par les objets d'analyse
 - . Projection des CU sur le diagramme de classe via des diagrammes de séquence

jean-philippe.babau@univ-brest.fr

21

UBO

UP : analyse

- “ Construction du modèle d'analyse pour préparer la conception
 - . Forme générale stable du système
 - “ Ne doit pas être remis en question fortement au cours de la vie du projet
 - . Haut-niveau d'abstraction
 - . Vision plus précise et formelle des CU
 - “ Dérivation sous forme de diagrammes de séquence
 - . Passage du langage du client à celui du développeur

jean-philippe.babau@univ-brest.fr

22

UBO

UP : analyse

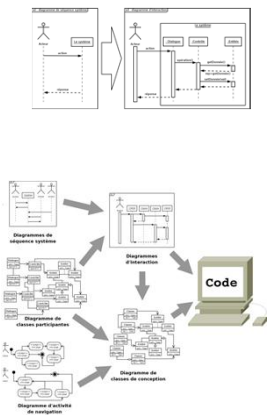
- “ Des rôles et des activités
 - . Architecte
 - “ Description de l'architecture
 - . Packages d'analyse
 - . Classes d'analyse
 - . Ingénieur CU
 - “ Réalisation/analyse des CU
 - . Ingénieur qualité
 - “ Evaluation de l'architecture
 - . Utilisation de design pattern, respect du guide de style, δ

jean-philippe.babau@univ-brest.fr 23

UBO

UP : conception

- “ Passage du monde « idéal » au monde « réel »
- “ Diagramme d'interaction
 - . Diagramme de séquence
 - . Dérivation du diagramme de séquence sur le diagramme de classe système
- “ Conception détaillée
 - . Diagramme de classe
 - . Intégration des spécificités du langage d'implémentation
 - . Intégration des spécificités des outils
 - . Utilisation de design-pattern



jean-philippe.babau@univ-brest.fr 24

UBO

UP : conception

- " Proposer une réalisation de l'analyse et des cas d'utilisation en prenant en compte *toutes les exigences*
- " **Conception architecturale**
 - . Les nœuds et la configuration du réseau pour le déploiement
 - . Les composants de communication et les sous-systèmes (modèle en couche, modèle Client/Serveur, etc.)
 - " Communications distantes, etc.
- " **Concevoir les classes**
 - . Décrire les méthodes, les états
 - . Prendre en compte les besoins extra-fonctionnels
- " **Concevoir les sous-systèmes**
 - . Mettre à jour les dépendances, les interfaces...
 - . Composants de service liés à l'application, de middleware etc.

jean-philippe.babau@univ-brest.fr

25

UBO

UP : conception

- " **Analyse des cas d'utilisation**
 - . Raffinement des scénarios des cas d'utilisation
 - " Diagramme de séquence ou de collaboration
 - . Si le scénario est trop complexe
 - " Décomposer en plusieurs scénarios enchaînés
- " **Le modèle structurel est construit pour supporter l'ensemble des scénarios**

jean-philippe.babau@univ-brest.fr

26

UBO

UP : conception

“ Des rôles et des activités

- . Architecte
 - “ Décrire l'architecture en package et/ou classes
- . Ingénieur de CU
 - “ Décrire les CU au travers des diagrammes de séquence
- . Chef de projet
 - “ Affecter les activités de réalisation au vu des sous-systèmes

jean-philippe.babau@univ-brest.fr 27

UBO

Remarques : différentes sortes de classes

“ **Classe conceptuelle**

- . Objets issus du Business Model

“ **Classe d'analyse**

- . 3 stéréotypes de Jacobson : frontière, contrôle, entité

“ **Classe logicielle**

- . Composants logiciels
- . Classes de conception

“ **Classe d'implémentation**

- . Classe implémentée dans un langage OO

“ Bien séparer par activité les niveaux de descriptions / raffinement

jean-philippe.babau@univ-brest.fr 28

UBO

UP : réalisation, tests, déploiement

- ~ Réalisation et tests
 - . Mise au point de logiciel
 - . Cf. cours Algorithmique et programmation
- ~ Déploiement
 - . Packaging et installation de logiciel
 - . Tests en utilisation réelle, in situ
 - . Formation
 - . Mise en place d'une assistance et gestion de bug
- . Phase cruciale pour la réussite du projet ...

jean-philippe.babau@univ-brest.fr 29

UBO

UP : réalisation, tests, déploiement

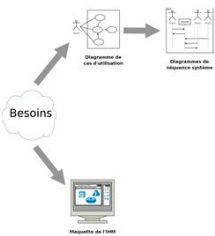
- ~ Des rôles et des activités
 - . Architecte
 - ~ Modèles de déploiement
 - . Développeur
 - ~ Codage d'un package
 - . Concepteur de tests
 - ~ Modèle de tests, cas de test, procédures de test, évaluation des tests, plan de tests
 - . Testeur d'intégration
 - ~ Tests d'intégration
 - . Formateur
 - ~ Plan de formation, transparents, exercices types

jean-philippe.babau@univ-brest.fr 30

UBO

Rappels sur l'analyse des besoins

- " Identification et représentation des besoins
 - . Diagramme de cas d'utilisation
 - . Description des acteurs et des cas d'utilisation
- " Spécification détaillée des besoins
 - . Diagrammes de séquence
 - . Détail des cas d'utilisation
- " Maquette de l'IHM de l'application (non couvert par UML)

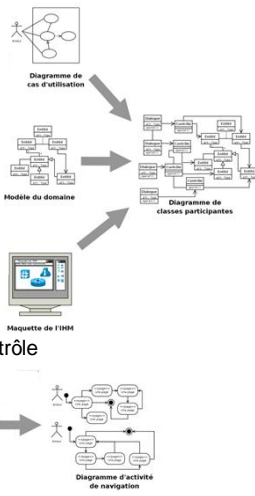


jean-philippe.babau@univ-brest.fr 31

UBO

Rappels sur l'analyse du problème

- " Elaboration d'un modèle « idéal »
- " Modèle du domaine
 - . Diagramme de classe
- " Conception système
 - . Diagramme de classe
 - . Modèle du domaine : -> classes métier
 - . IHM : -> classes dialogue
 - . Modèle d'interaction métier /IHM : -> classes de contrôle
- " Modèles des interactions
 - . Diagramme d'activité
 - . Enchaînement des menus, fenêtres, ⏪

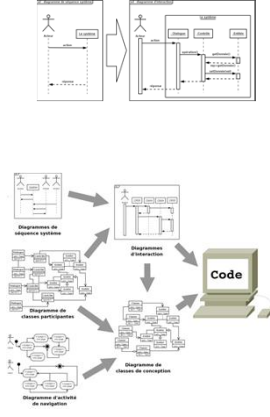


jean-philippe.babau@univ-brest.fr 32

UBO

Rappels sur la conception

- " Passage du monde « idéal » au monde « réel »
- " Diagramme d'interaction
 - . Diagramme de séquence
 - . Dérivation du diagramme de séquence sur le diagramme de classe système
- " Conception détaillée
 - . Diagramme de classe
 - . Intégration des spécificités du langage d'implémentation
 - . Intégration des spécificités des outils
 - . Utilisation de design-pattern



jean-philippe.babau@univ-brest.fr 33

UBO

Plan

- " Introduction
 - . Pourquoi une méthode ?
 - . Objectifs d'une méthode
- " UP
 - . Introduction
 - . Phases
 - . Activités
 - . Processus : Phases Vs Activités
 - . Principes généraux
 - . Autres approches

jean-philippe.babau@univ-brest.fr 34

UBO

UP : processus

- “ Objectifs d'UP : définir qui fait quoi, comment et quand.
 - . Pour un projet de développement logiciel
- “ Travailleurs (**qui fait**)
 - . Rôle
 - “ Un rôle définit le comportement et les responsabilités d'un individu ou d'un groupe.
 - . Exemple : architecte, chef de projet, formateur
- “ Artefact (**quoi,**)
 - . Produit concret d'une activité
 - . Exemples : modèle architectural, code source, exécutable, modèle des CU, manuel utilisateur, ⚪

jean-philippe.babau@univ-brest.fr

35

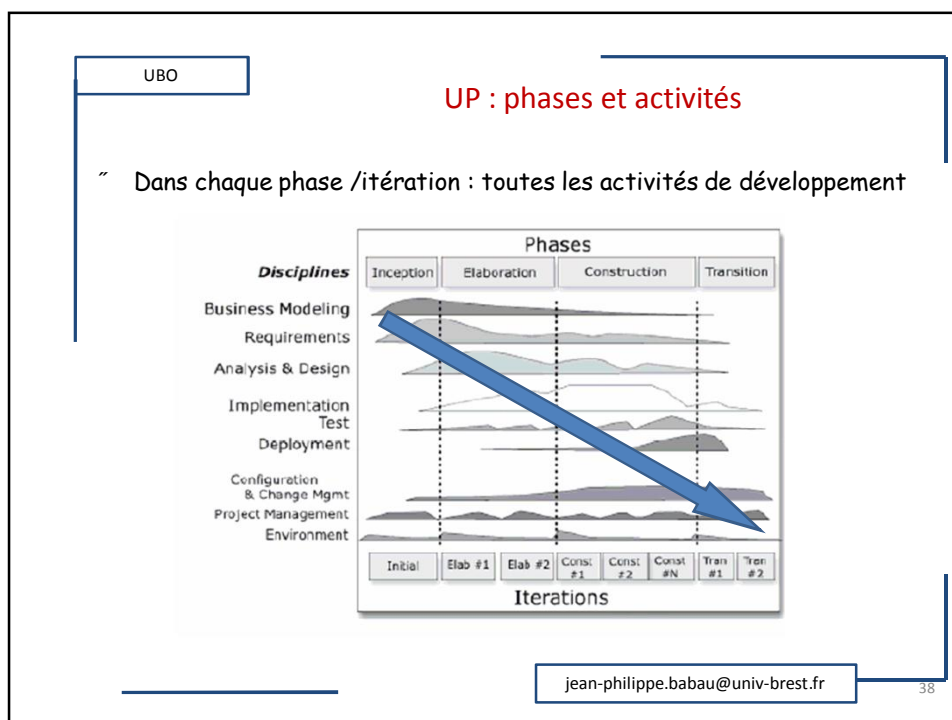
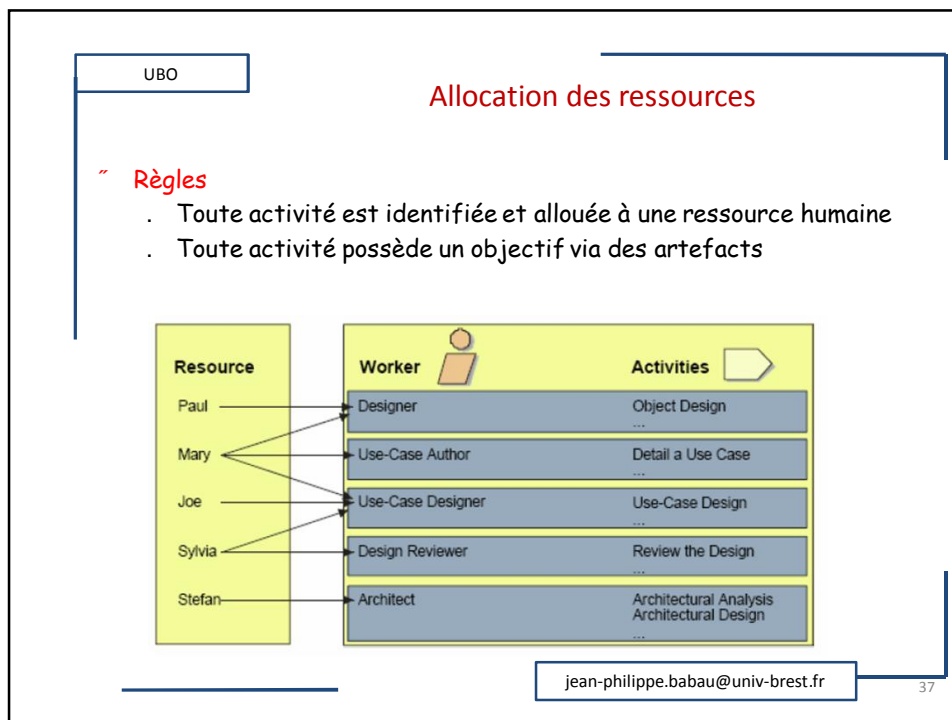
UBO

UP : processus

- “ Activités (**comment**)
 - . Activités principales et complémentaires
 - “ Découpées en multiples sous-activités ou unité de travail
 - . Une unité de travail possède un objectif exprimé en terme de création ou de mise à jour d'artefacts
 - . Exemples d'unité de travail
 - “ concevoir une classe, corriger un document, détailler un CU, planifier une itération (par le chef de projet), recherche et modélisation des cas d'utilisation (par l'analyste système), révision de la conception (par le design reviewer), exécution des tests de performance (par le testeur de performance)
- “ Workflow (**et quand.**)
 - . Planning des activités
 - . Modélisable en UML par un diagramme d'activité

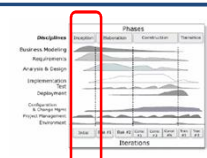
jean-philippe.babau@univ-brest.fr

36



UBO

Lancement Vs activités



- " **Modélisation métier**
 - . Comprendre le contexte du système (50-70% du contexte)
- " **Expression des besoins**
 - . Etablir les besoins fonctionnels et non fonctionnels (80%)
 - . Traduire les besoins fonctionnels en cas d'utilisation (50%)
 - . Détailler les premiers cas par ordre de priorité (10%)
- " **Analyse**
 - . Analyse des cas d'utilisation (10% considérés)
 - . Pour mieux comprendre le système à réaliser, guider le choix de l'architecture
- " **Conception**
 - . Première ébauche de l'architecture
 - " Nœuds, réseau, sous-systèmes, couches logicielles
 - . Examen des aspects importants et à plus haut risque
- " **Environnement**
 - . Inventaire et choix des outils de développement
- " On fait une première « passe » sur toutes les activités
 - . Comprendre le projet et identifier tous les enjeux

jean-philippe.babau@univ-brest.fr

39

UBO

Livrables de la phase de lancement

- " Première version du modèle du domaine ou de contexte de l'entreprise
- " Liste des besoins fonctionnels et non fonctionnels
- " Ébauche des modèles de cas, d'analyse et de conception
- " Esquisse d'une architecture
- " Liste ordonnée de risques et liste ordonnée de cas
- " Grandes lignes d'un planning pour un projet complet
- " Première évaluation du projet, estimation grossière des coûts
- " Glossaire

jean-philippe.babau@univ-brest.fr

40

UBO

Elaboration Vs activités

- “ **Modélisation métier**
 - . Définir le contexte du système (100% du contexte)
- “ **Expression des besoins**
 - . Terminer la capture des besoins (CU) et en détailler de 40 à 80%
 - . Faire un prototype de l'interface utilisateur
- “ **Analyse**
 - . Analyse architecturale complète du domaine (architecture de référence)
 - . Raffinement des cas d'utilisation (>10%)
- “ **Conception**
 - . Terminer la conception architecturale
 - . Effectuer la conception correspondant aux cas sélectionnés
- “ **Réalisation**
 - . IHM
 - . Prototype
 - “ Faire en sorte de pouvoir éprouver les choix
- “ **Test**
 - . Des prototypes et maquettes réalisés (édition et navigation)

jean-philippe.babau@univ-brest.fr

41

UBO

Livrables de la phase d'élaboration

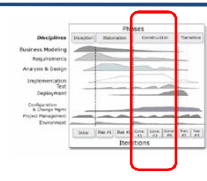
- “ Un modèle de l'entreprise (processus, glossaire) complet
- “ Les cas d'utilisation et l'expression des besoins (80%)
- “ La description des architectures
 - . Technique, composants, classes
 - . Une version des modèles
- “ Conception (<10%), implémentation (<10%) et déploiement
 - . Une architecture de base exécutable
 - . Un manuel utilisateur préliminaire
- “ Une liste des risques mise à jour
- “ Un projet de planning pour les phases suivantes
 - . Itérations
- “ Évaluation du coût du projet

jean-philippe.babau@univ-brest.fr

42

UBO

Construction Vs activités



- “ Plusieurs itérations
 - . Selon le risque et les priorités
- “ Expression des besoins
 - . Spécifier l'interface utilisateur
- “ Analyse
 - . Terminer l'analyse de tous les cas d'utilisation, la construction du modèle structurel d'analyse, le découpage en packages...
- “ Conception
 - . L'architecture est fixée et il faut concevoir les sous-systèmes
 - . Concevoir les classes
- “ Réalisation
 - . Réaliser, faire des tests unitaires, intégrer les incréments
- “ Test
 - . Toutes les activités de test : plan de test, conception de test, évaluation...

jean-philippe.babau@univ-brest.fr

43

UBO

Livrables de la phase de construction

- “ Un plan du projet pour la phase de transition
- “ L'exécutable
- “ Tous les documents et les modèles du système
- “ Une description à jour de l'architecture
- “ Un manuel utilisateur suffisamment détaillé pour les tests

jean-philippe.babau@univ-brest.fr

44

UBO

Transition Vs activités

- " Préparer la version beta à tester
- " Installer la version sur le site, convertir et faire migrer les données nécessaires...
- " Gérer le retour des sites
- " Le système fait-il ce qui était attendu ? Erreurs découvertes ?
- " Adapter le produit corrigé aux contextes utilisateurs (installation...)
- " Terminer les livrables du projet (modèles, documents...)
- " Déterminer la fin du projet
- " Reporter la correction des erreurs trop importantes (nouvelle version)
- " Organiser une revue de fin de projet (pour apprendre)
- ...
- " Planifier le prochain cycle de développement

jean-philippe.babau@univ-brest.fr

45

UBO

Livrables de la phase de transition

- " L'exécutable et son programme d'installation
 - . Selon les cibles
- " Les documents légaux
 - . Contrat, licences, garanties
- " Les documents de développement
- " Les manuels utilisateur, administrateur et opérateur
- " Le matériel de formation
 - . Cours, exercices
- " Les références pour le support utilisateur
 - . Suivi de bug, assistance
 - . Site Web

jean-philippe.babau@univ-brest.fr

46

UBO

Plan

- “ Introduction
 - . Pourquoi une méthode ?
 - . Objectifs d'une méthode
- “ UP
 - . Introduction
 - . Phases
 - . Activités
 - . Processus
 - . **Principes généraux**
 - . Autres approches

jean-philippe.babau@univ-brest.fr

47

UBO

UP : itératif et incrémental

- “ Des itérations
 - . Chaque phase comprend des itérations
 - . Une itération a pour but de maîtriser une partie des risques et apporte une preuve de faisabilité
 - . Une itération produit un système partiel opérationnel (exécutable, testé et intégré) avec une qualité égale à celle d'un produit fini
 - . Une itération peut être évaluée
 - “ Produire des informations pour savoir si on va dans une bonne direction
- “ Un incrément par itération
 - . Le logiciel et le modèle évoluent suivant des incréments
 - . Les incréments : série de prototypes qui vont en s'améliorant
 - “ De plus en plus complet et précis
 - “ Intégration de retours utilisateurs
- “ Processus incrémental
 - . Gestion de configuration nécessaire

jean-philippe.babau@univ-brest.fr

48

UBO

UP : itératif et incrémental

- “ Une itération est un mini-projet
 - . Objectifs pour le prototype, critères d'évaluation
 - . Comporte toutes les activités
 - . Est terminée par un point de contrôle
 - . Conduit à une version montrable implémentant un certain nombre de CU
 - . Dure entre de 2 à 4 semaines
 - “ Butée temporelle qui oblige à prendre des décisions

- “ Organisation des itérations selon les priorités établies pour les cas d'utilisation et l'étude de risque
 - . Le plus important et le plus risqué au début
 - . Planification des itérations
 - “ Modifiable
 - “ Gestion de projet plus complexe : suivi et adaptation des itérations

jean-philippe.babau@univ-brest.fr 49

UBO

UP : piloté par les besoins

- “ Objectif du processus
 - . Construction d'un système qui réponde aux besoins par construction de modèles

- “ Utilisation des Cas d'Utilisation (CU)
 - . Tout au long du cycle
 - . Validation des besoins / utilisateurs
 - . Point de départ pour l'analyse (découverte des objets, de leurs relations, de leur comportement) et la conception (sous-systèmes)
 - . Guide pour la construction des interfaces
 - . Guide pour la mise au point des plans de tests

jean-philippe.babau@univ-brest.fr 50

UBO

Avantage des CU

- “ **Centré utilisateur**
 - . Support de communication en langue naturelle entre utilisateurs et concepteurs basé sur les scénarios (et non liste de fonctions)
 - . Egalement adapté aux informaticiens
- “ **Assurent la traçabilité par rapports aux besoins de toute décision de conception sur l'ensemble du projet**
 - . Tout modèle peut se référer *in fine* à un CU
- “ **Assurent le lien pour une vision commune des membres du projet sur l'architecture**

jean-philippe.babau@univ-brest.fr 51

UBO

UP : orienté vers la diminution des risques

- “ **Différentes natures de risques**
 - . Besoins
 - “ Mauvaise interprétation des besoins, du domaine
 - . Technique
 - “ Architecture technique inadaptée (performances insuffisantes)
 - . Autres
 - “ Outil inutilisable (formation, administration)
 - “ Risques non techniques (personnel de développement insuffisant, problèmes commerciaux ou financiers)
- “ **Gestion des risques**
 - . Identifier et classer les risques par importance
 - . Organiser les itérations en fonction des risques
 - . Anticiper les difficultés avant qu'elles ne se produisent
- “ **Tout risque fatal pour le projet est à découvrir au plus tôt**

jean-philippe.babau@univ-brest.fr 52

UBO

UP : centré sur l'architecture

- " Objectif de l'analyse/conception : construire une architecture
 - . Qui permette de promouvoir la réutilisation
 - . Qui permette de séparer les préoccupations
 - " Gérer la complexité
 - " Travail en équipe
 - " Découpage en packages
 - . Stable
 - " Architecture de référence (10% des classes)
 - . La réalisation des CU doit s'appuyer sur l'architecture
 - " Architecture de référence : intégration des CU les plus importants

jean-philippe.babau@univ-brest.fr 53

UBO

Plan

- " Introduction
 - . Pourquoi une méthode ?
 - . Objectifs d'une méthode
- " UP
 - . Introduction
 - . Phases
 - . Activités
 - . Processus
 - . Principes généraux
 - . Autres approches
 - " Agile UP, Basic UP, Enterprise UP, Essential UP, Open UP, Oracle Unified Method, TTUP

jean-philippe.babau@univ-brest.fr 54

UBO

Agile UP

- “ Proposé par Craig Larman (2005)
- “ UP peut être considéré comme complexe, formel et lourd dans sa description générale,
 - . UP essaye de prendre en compte toutes les options possibles, pour toutes les entreprises, et toutes les tailles de projets
 - “ Nombreux artefacts et activités
 - . Doit être adapté à chaque projet
 - “ Suivi strict des activités : rigidité
 - . Tendance à la cascade

jean-philippe.babau@univ-brest.fr

55

UBO

Agile UP

- “ Itérations courtes (3 semaines maximum)
- “ Mode organisationnel léger
 - . Petit ensemble d'activités et d'artefacts
- “ Fusion analyse / conception
- “ Utilisation d'UML pour comprendre et concevoir
 - . Plus que pour générer du code
- “ Planification adaptative
- “ Bref, application de UP dans l' « esprit Agile »
 - . En considérant que UP est agile naturellement dans sa conception (et pour ses concepteurs), mais ne l'est pas dans ses applications

jean-philippe.babau@univ-brest.fr

56

UBO

TTUP : Two Tracks Unified Process

- " Proposé par Valtech (consulting) en 2004
- " Objectif
 - . Prendre en compte les contraintes de changement continu
- " Création d'un nouveau SI
 - . Deux grandes sortes de risques
 - " Imprécision fonctionnelle : inadéquation aux besoins
 - " Incapacité à intégrer les technologies : inadaptation technique
- " Evolution du SI
 - . Deux grandes sortes d'évolutions
 - " Evolution fonctionnelle
 - " Evolution technique

jean-philippe.babau@univ-brest.fr 57

UBO

TTUP

- " Principe général : développement en Y
 - . Besoins techniques : réalisation d'une architecture technique
 - . Besoins fonctionnels : modèle fonctionnel
 - . Réalisation du système
 - " Fusionner les résultats des deux branches du processus

Inception	Elaboration		Construction	
Incrément 1	Incr.2	Incr.3	Incr.4	Incr.5 ...

jean-philippe.babau@univ-brest.fr 58

UBO

TTUP

- “ **Côté fonctionnel**
 - . Modèle d'analyse réutilisable
 - . Indépendant des technologies (PIM / CIM du MDA)
- “ **Côté technique**
 - . Architecture technique indépendante des besoins fonctionnels (PDM du MDA)
 - . Notion de CU techniques
 - . Dépend de l'existant
 - “ Culture d'entreprise, solutions éprouvées
 - . Architecture à base de composant
 - . Architecture technique réutilisable
- “ **Branche commune**
 - . Conception préliminaire : le plus délicat
 - . PSM du MDA

jean-philippe.babau@univ-brest.fr

59

UBO

Conclusion

- “ **Une méthode**
 - . Des règles pragmatiques
 - “ L'organisation doit être au service de la production
 - . Des règles communes
 - “ Des conventions de nommage, de présentation, d'organisation
 - . Des règles éprouvées et reconnues
 - “ Expérience, standards
 - . Des adaptations selon le contexte
 - “ Pas de rigidité
- “ **Expliciter et formaliser toutes les activités liées au développement**
 - . Développer n'est pas simplement coder
 - . Pas d'activités « invisibles »
 - “ Associée à un exécutant, avec des objectifs, des artefacts
 - . Pas d'activité « isolée »
 - “ Liens avec les autres activités au sein d'un processus global
 - . Une activité est intégrée dans un processus

jean-philippe.babau@univ-brest.fr

60

UBO	Conclusion
<ul style="list-style-type: none"> ” La méthode UP <ul style="list-style-type: none"> . Pilotage par les besoins <ul style="list-style-type: none"> ” Etre au service du client . Diminution des risques <ul style="list-style-type: none"> ” Identifier au plus tôt ” Commencer par le plus difficile . Centré sur l'architecture <ul style="list-style-type: none"> ” Structurer tous les modèles (UC, déploiement et composants, classes) . Des modèles pour <ul style="list-style-type: none"> ” Préciser, formaliser : lever les ambiguïtés, éviter les confusions ” Expliciter, commenter, noter, tracer : communiquer, suivre, se référer ” Faire des modèles : partager, interpréter 	<p>jean-philippe.babau@univ-brest.fr</p> <p>61</p>

UBO	Conclusion
<ul style="list-style-type: none"> ” Qualités humaines <ul style="list-style-type: none"> . Être tourné vers le client <ul style="list-style-type: none"> ” Comprendre le client (métier et domaine) ” Comprendre les besoins du client (besoins) ” Comprendre les capacités d'intégration de suivi du client (déploiement) . Savoir travailler en équipe <ul style="list-style-type: none"> ” Savoir communiquer <ul style="list-style-type: none"> . Utiliser des modèles . Respecter des conventions ” Produire des documents réutilisables ” Qualités projet <ul style="list-style-type: none"> . Travailler par itération . Planifier et respecter les délais . Être guidé par des objectifs . Faire valider chaque production 	<p>jean-philippe.babau@univ-brest.fr</p> <p>62</p>

UBO

Bibliographie

- " *OMG et UML*
 - . <http://www.omg.org/>
 - . <http://www.uml.org/>

- " *Cours de Yannick Prié*
 - . <http://liris.cnrs.fr/~yprie/>

- " *Rational Unified Process Best Practices for Software Development Teams*
 - . [IBM Rational Unified Process Web Site](#)

jean-philippe.babau@univ-brest.fr

63