

INTRODUCTION UML

Écrit à partir des cours de Mireille BLAY
https://mbf-iut.i3s.unice.fr/doku.php?id=2014_2015:s2:start

CONTRAT

VOUS

- 5 min avant: entrez
- 8:00: écoutez
- Posez des questions
- Faites les exercices sérieusement
- Atteignez les objectifs
- Obtenez la meilleure note possible
- **RESPECTEZ LES REGLES**

Le prof

- Vous prépare le cours
- Répond à vos questions
- Ne vous prend pas par la main
- Est là pour vous AIDER, pas pour vous ASSISTER
- **FAIRE DE VOUS DES PROS**

- Description:

L'étudiant valide un module

- Pré-condition:

L'étudiant assiste à tous les cours et fait les exercices sérieusement en posant des questions. S'il y a un cours où il est absent, il rattrape le cours et pose des questions si nécessaire au cours suivant

- Flot basique:

1. L'étudiant arrive à l'heure au cours
2. L'étudiant prend des notes
3. L'étudiant fait les exercices
4. L'étudiant pose des questions au professeur

Déroulement

Flot basique:

1. L'étudiant arrive à l'heure au cours
2. L'étudiant prend des notes
3. L'étudiant fait les exercices
4. L'étudiant pose des questions au professeur

▪ Flot alternatif:

1a) l'étudiant n'arrive pas à l'heure

1. l'étudiant devra travailler pour rattraper...
2. L'étudiant va probablement ne pas comprendre ce qui est en relation avec le début
3. Aller à l'étape 4

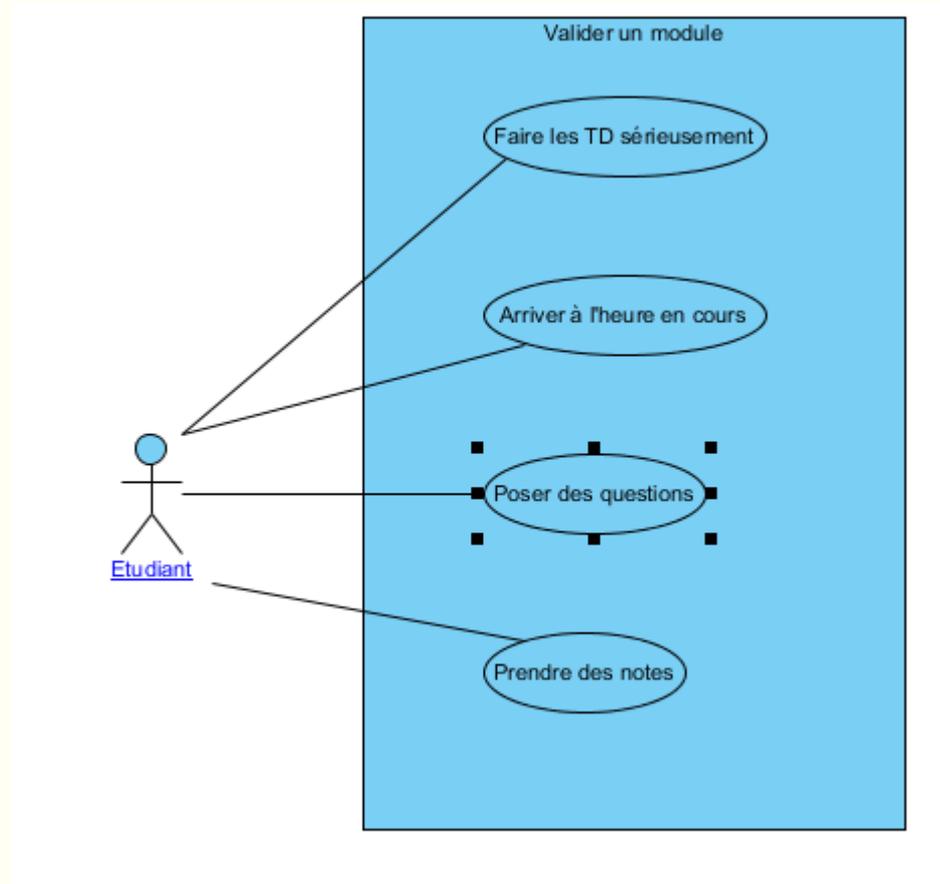
3a) l'étudiant ne fait pas ses exercices

1. le professeur met une mauvaise note à l'étudiant pour non rendu ou travail bâclé
2. L'étudiant va faire un EFFORT pour travailler le cours existant

- Postconditions:

A la fin de ce cas d'utilisation, soit l'étudiant valide le module (travail régulier et examen validé), soit... vous imaginez très bien ce qui arrive.

Use Case





CONTEXTE

Où se trouve le logiciel?

Où se trouve le logiciel ?

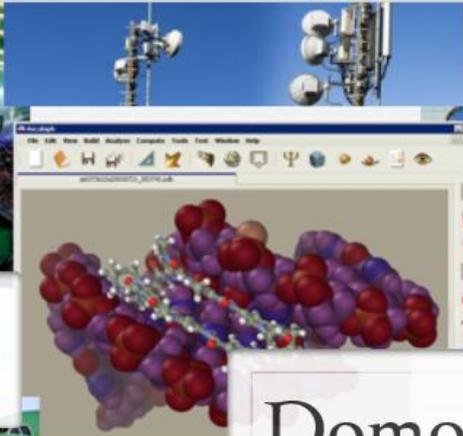
Finances



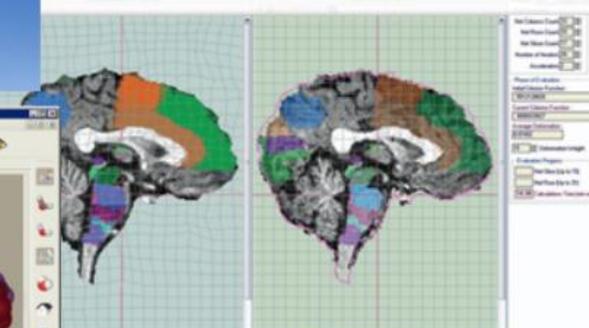
Arts, ..



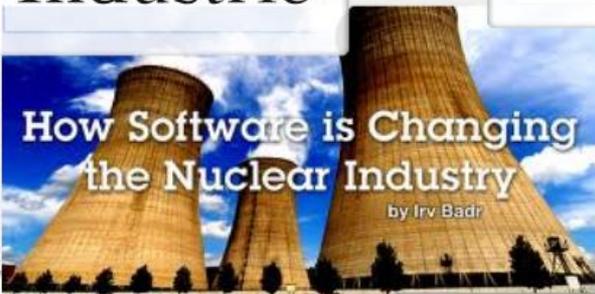
telecommunication



Medicine



Industrie



Transports



Domotique



Web



Fouille de données

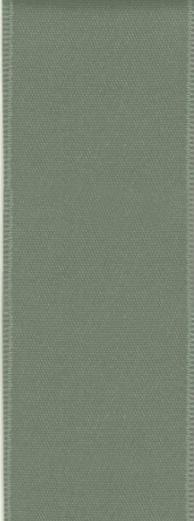


Nouvelles interactions



En musique, sport, ...

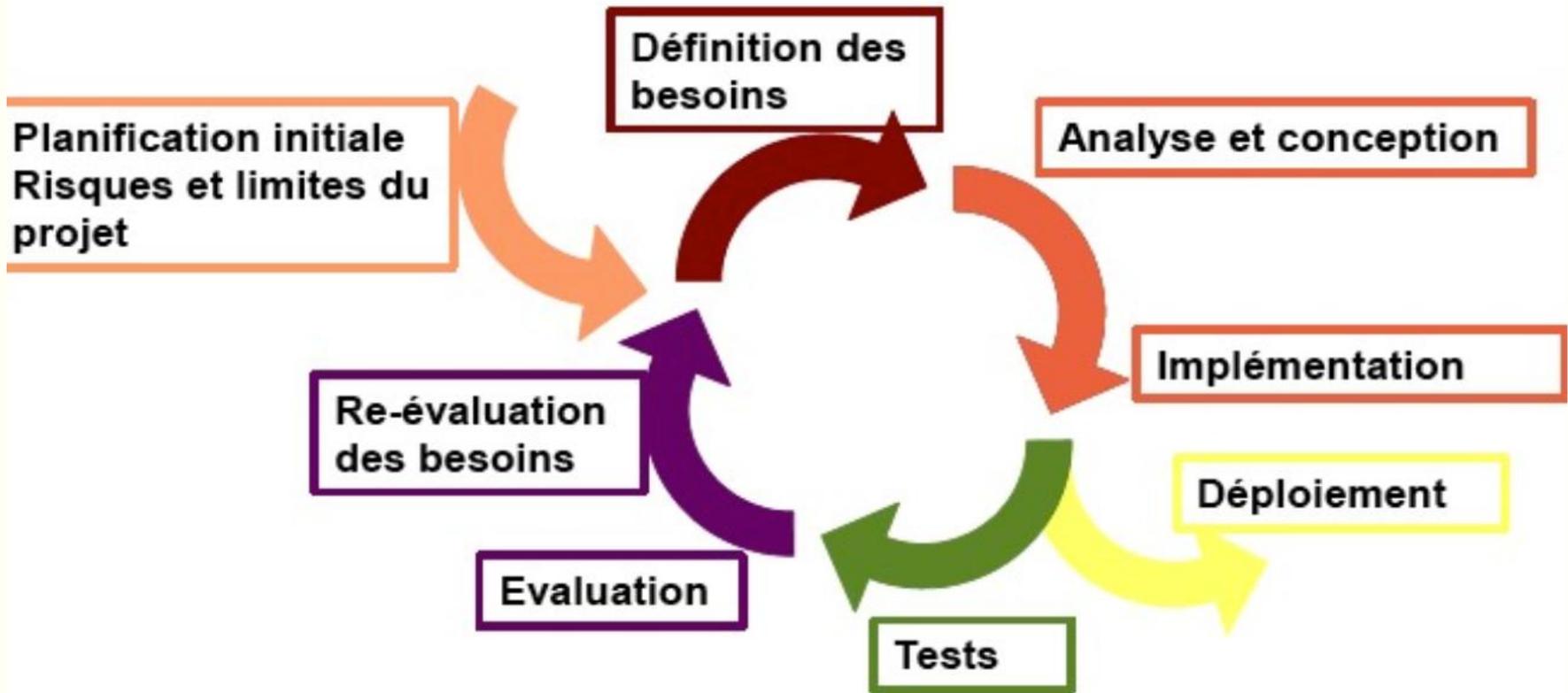
17/02/2015



VOUS DEVEZ RÉALISER UNE APPLI LOGICIELLE POUR VISUALISER LES DONNÉES MÉTÉO DANS UN COCKPIT

Que faites-vous?

Activités du développement logiciel



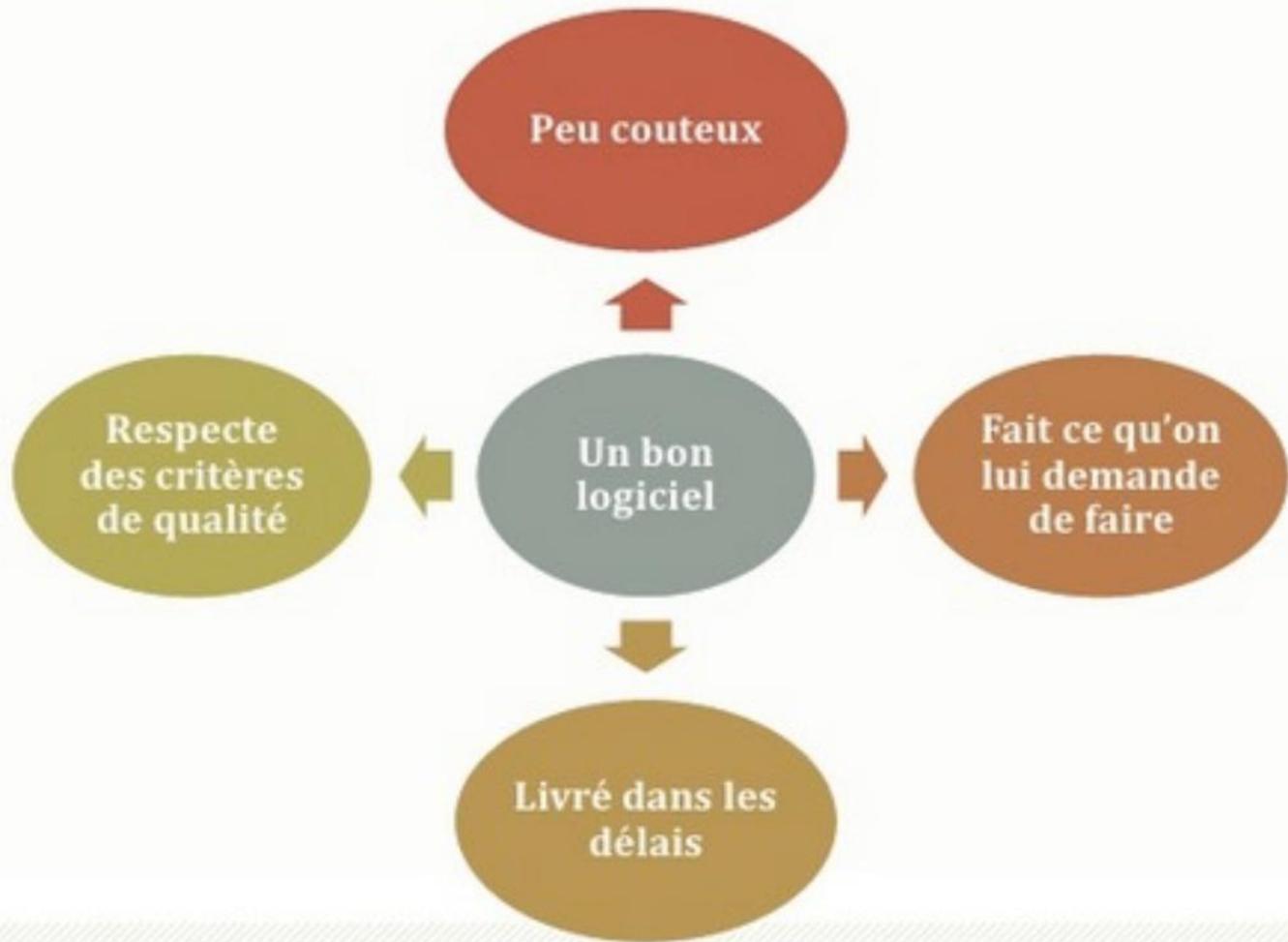
La programmation n'est qu'une étape du développement !!



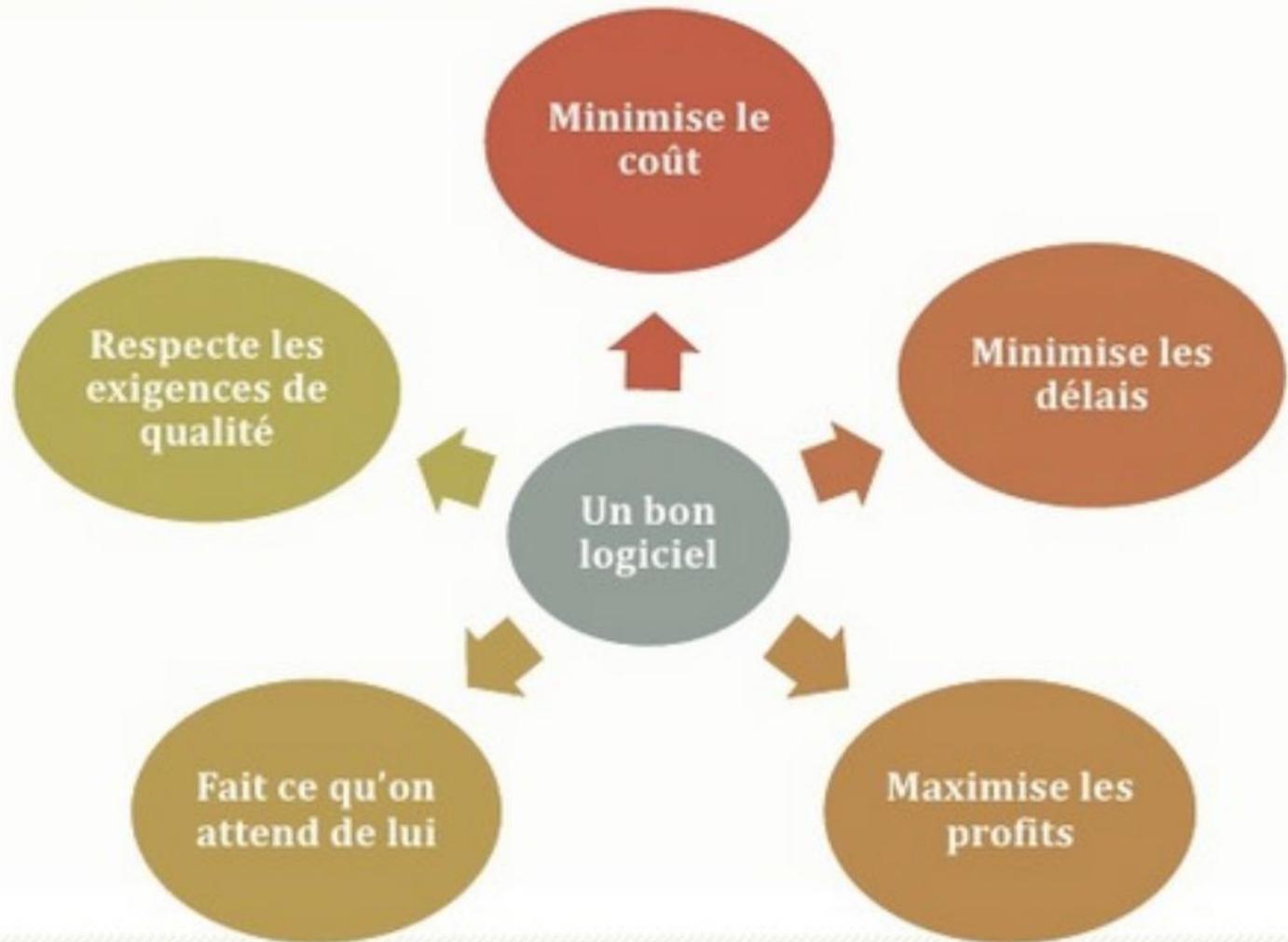
QUELLE QUALITÉ POUR LE LOGICIEL?

- Monteriez-vous dans l'avion pour lequel vous avez écrit :
 - le système de visualisation des données météo?
 - Le logiciel support aux commandes automatiques?
- Vous avez rôlé sur certains logiciels.... Pourquoi êtes-vous certains de faire mieux?
- Vos pages web sont-elles de qualité?

Un Bon Logiciel du Point de Vue d'un Client



Un Bon Logiciel du Point de Vue d'un Fournisseur



PLAN

1. Problèmes du développement logiciel
 1. Histoire brève jusqu'aux limites de la programmation structurée
 2. Du bidouillage au génie logiciel
2. Introduction à UML
 1. Un peu d'histoire
 2. Survol



PROBLÈMES DU DÉVELOPPEMENT LOGICIEL

Gestion progressive de la complexité

- Langage machine -> forte dépendance à l'architecture



- Langage évolué: séparer les concepts manipulés dans les langages de leur représentation dans la machine -> FORTRAN

Méthode d'analyse par décomposition

- Complexité des programmes -> mieux structurer les programmes (plus de goto ou programmation spaghetti)
- Découper les tâches en modules indépendants
- Programmation structurée =
- programmation dirigée par les traitements



Explosion des besoins

- Coût du matériel plus faible -> bien de consommation courant
- Montée en abstraction: objets, composants, services, frameworks, usines logicielles...
- Changement de méthodes de développements

Problématique du génie logiciel

- Voir cours de Mireille Blay

➔ Programming-in-the-Small •

- Problème de la **qualité** interne d'un composant

➔ Programming-in-the-Large ○

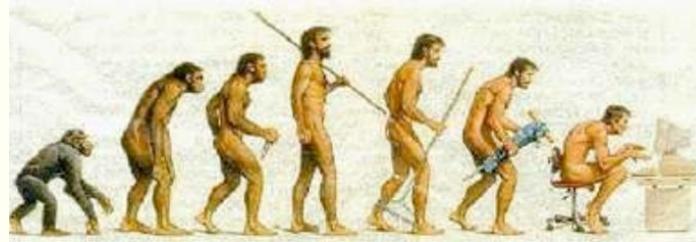
- Faire face à la construction de systèmes de plus en plus gros : non spécifique au logiciel, mais aggravé par sa “mollesse”.
- Problème de gestion de la complexité, de communication, etc.
- Maîtrise du *processus de développement*: délais, coûts, qualité.

● Programming-in-the-Duration ○□

- Problème de la maintenance corrective et évolutive.
- Notion de ligne de produits.



INTRODUCTION À UML



- Plusieurs méthodes -> un langage de modélisation standard
- ➔ **Modèle** : simplification de la réalité dont les buts sont



Visualiser
le système



Spécifier la
structure et le
comportement
du système



**Aider à la
construction**
du système



Documenter
les décisions

Qu'est-ce qu'un UML

➔ UML est un langage «visuel»

➔ Il supporte

- La visualisation
- La spécification
- La construction
- La documentation

Syntaxe et sémantique

Descriptions graphiques et textuelles

Architecture et comportement

Génération de code

On ne décrit pas l'univers tout entier...

Le logiciel joue un rôle majeur

■ UML **n'est pas une méthodologie** de développement, contrairement à RUP (*Rational Unified Process*).



Points forts

- Langage normalisé
 - Gain de précision
 - Gain de stabilité
 - Encourage l'utilisation d'outils
- Support de communication performant
 - Cadre l'analyse
 - Facilite la compréhension de représentations abstraites complexes
 - Polyvalent, souple -> langage universel
- Comprendre les concepts de poo

Points faibles

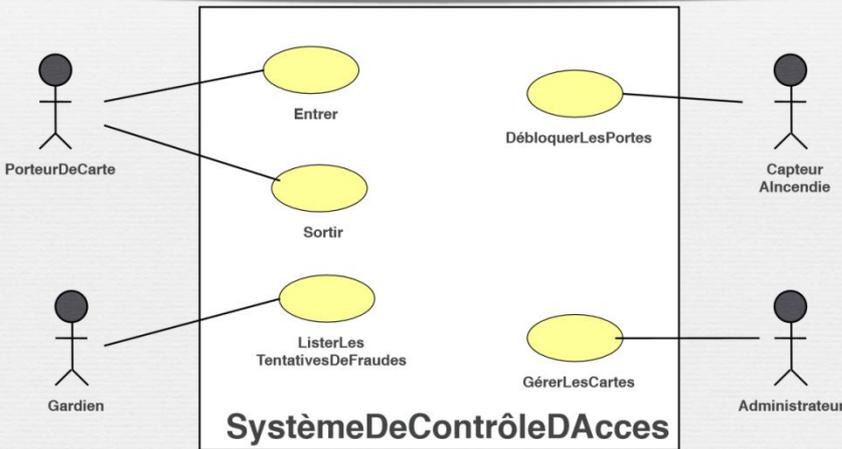
- Mise en pratique qui nécessite un apprentissage
 - Période d'adaptation
- Processus de développement (non imposé par UML) = autre clé de la réussite d'un projet

- **Vue fonctionnelle: interactions entre acteurs/utilisateurs et système**
 - Use Cases (objectifs à atteindre)
 - Diagrammes de séquences (scénarios d'interaction, forme chronologique)
 - Diagramme de collaboration
- **Vue structurelle: identifier les objets composants du prog**
 - Diagramme de classe (Liens/associations)
 - Diagramme de packages (Regroupement des classes liées)
 - Diagrammes d'objets
- **Vue dynamique: décrire l'évolution des objets complexes tout au long de leur cycle de vie**
 - Diagramme d'états (états guidés par interactions avec autres objets)
 - Diagramme d'activités (version simplifié)

Vue Fonctionnelle

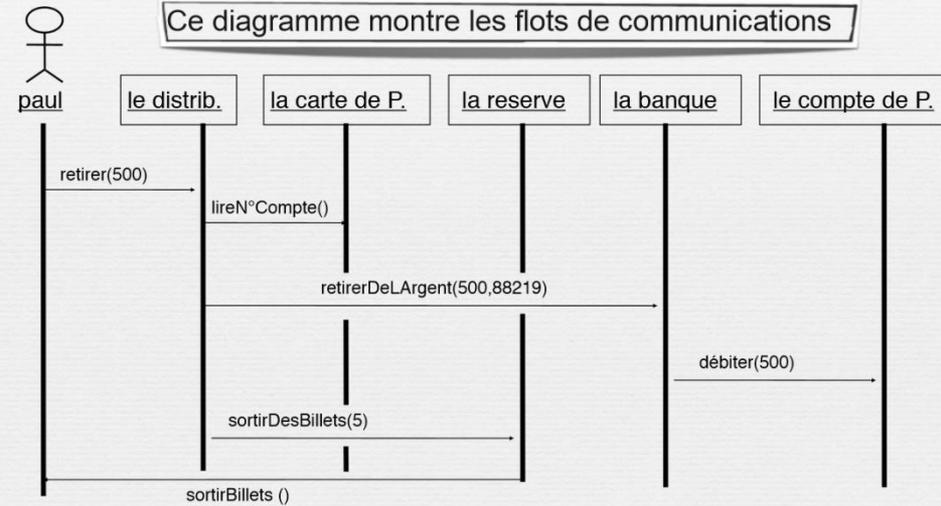
Diagrammes des cas d'utilisation

Ce diagramme montre ce que fait le système et qui l'utilise

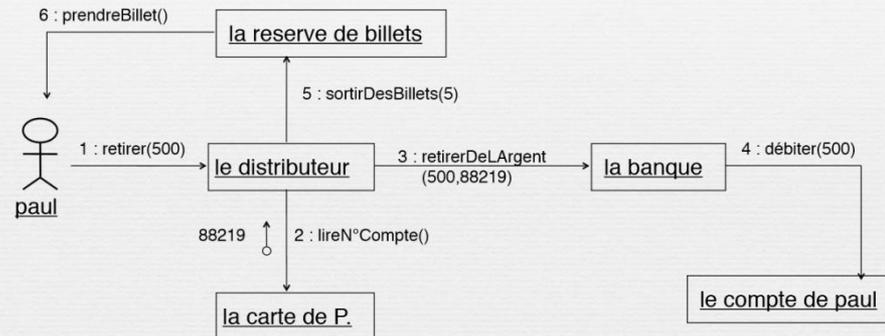


Diagrammes de séquence

Ce diagramme montre les flots de communications



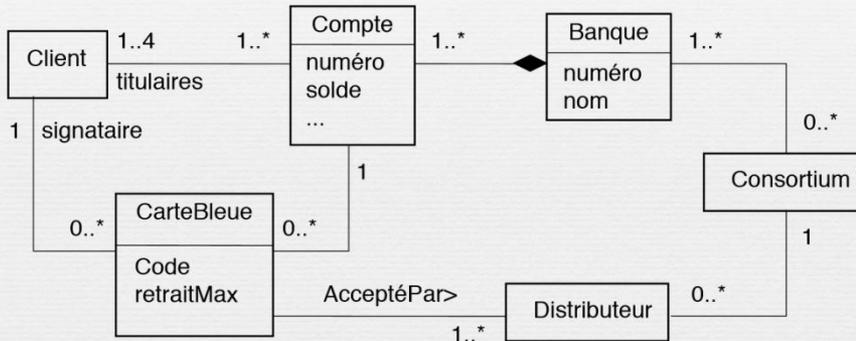
Diagrammes de collaboration



Non étudié cette année

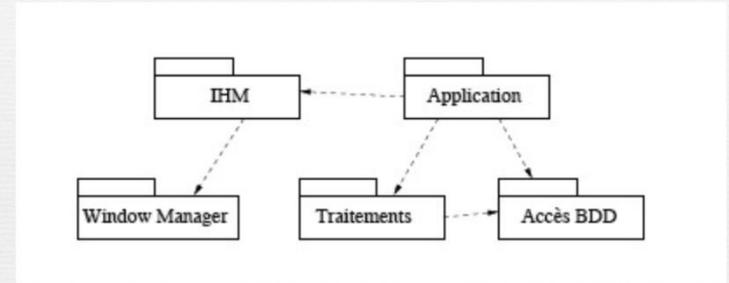
Vue Structurelle

Diagrammes de classes



Ce diagramme montre les classes et les relations entre elles

Diagrammes de packages

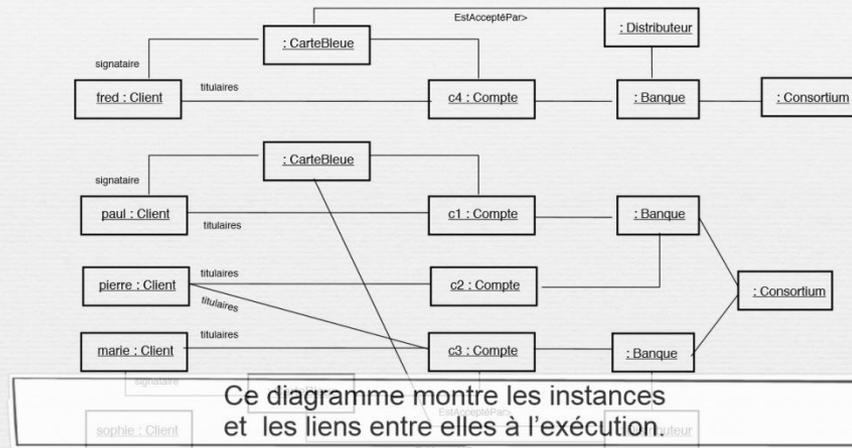


Regrouper entre elles des classes liées les unes aux autres de manière à faciliter la maintenance ou l'évolution du projet et de rendre aussi indépendantes que possible les différentes parties d'un logiciel.

53

<http://www.dfr.ensta.fr/Cours/docs/IN204/uml.pdf>

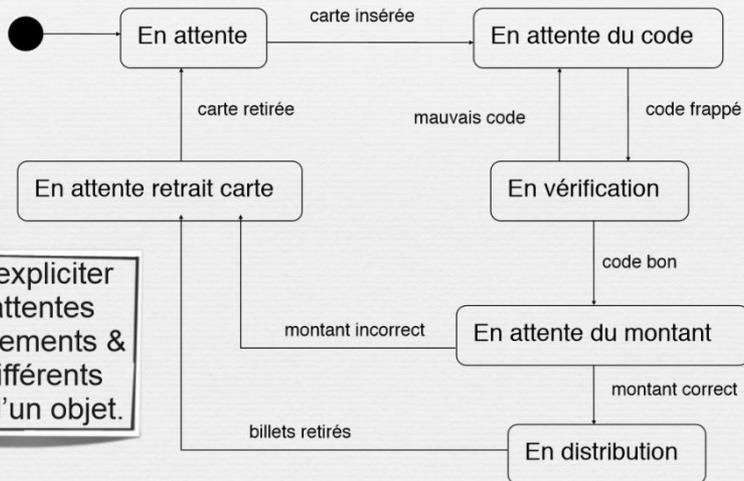
Diagrammes d'objets



Ce diagramme montre les instances et les liens entre elles à l'exécution.

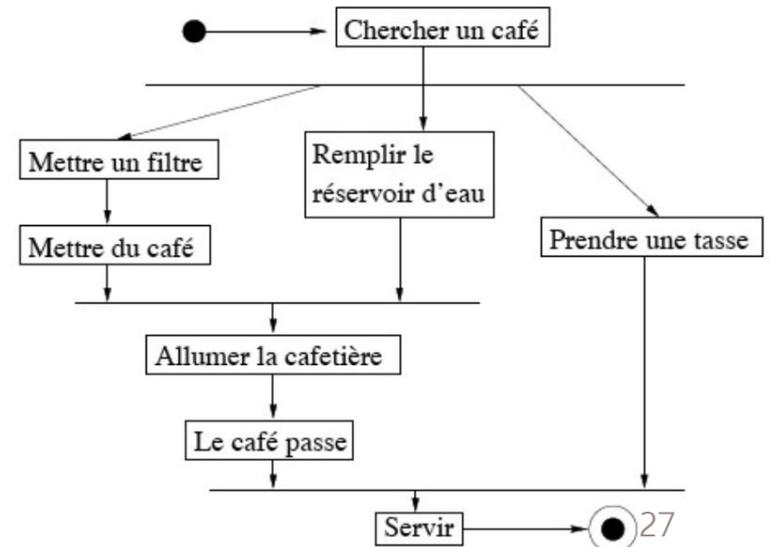
Vue Dynamique

Diagrammes d'états



Pour expliciter les attentes d'évènements & les différents états d'un objet.

Diagrammes d'activités



Bibliographie

- Voir le cours de Mireille BLAY