



Langage de modélisation UML : Introduction (FAD)

3^{ème} année A

**Année universitaire :
2020-2021**

henda.sfaxi@esprit.tn



Plan

- Présentation du module
- Etat des lieux
- Remise en context
- Élaboration d'un logiciel
- Cycle de vie d'un logiciel
- Modélisation
- UML
- Plan du module



Présentation du module

Titre

Nombre d'heures

Evaluation

Déroulement

Outils

► Présentation du module (1/2)

- Titre du module : Langage de modélisation UML
- Nombre d'heures : 42h (s'étale sur un semestre)
- Évaluation :
 - **Mode d'évaluation : écrit**
 - Calcul de la moyenne du module :
 - **Moyenne** = Note **CC** * 40% + Note **Examen** * 60%
 - **Note CC** = Moyenne (note des quizz + note du test)
- Déroulement :
 - Séances de formation à distance (FAD)
 - Séances de formation présentielle à l'école (FAE)

Présentation du module (2/2)



- Outils pour les TDs :
 - StarUML 
 - PowerDesigner 
 - ou tout autre outil de votre choix



► Objectifs et Prérequis

- Objectifs
 - Maîtriser la modélisation d'un système d'information
 - Savoir mettre en œuvre les diagrammes UML
 - Comprendre les niveaux d'abstraction selon les étapes d'analyse, de conception et d'implémentation dans le cadre d'un projet informatique
- Prérequis
 - Concepts fondamentaux de l'orientée objet

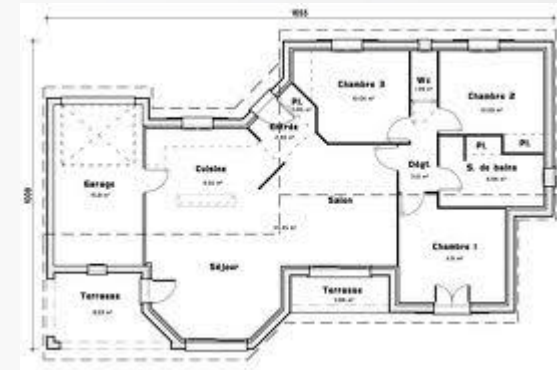
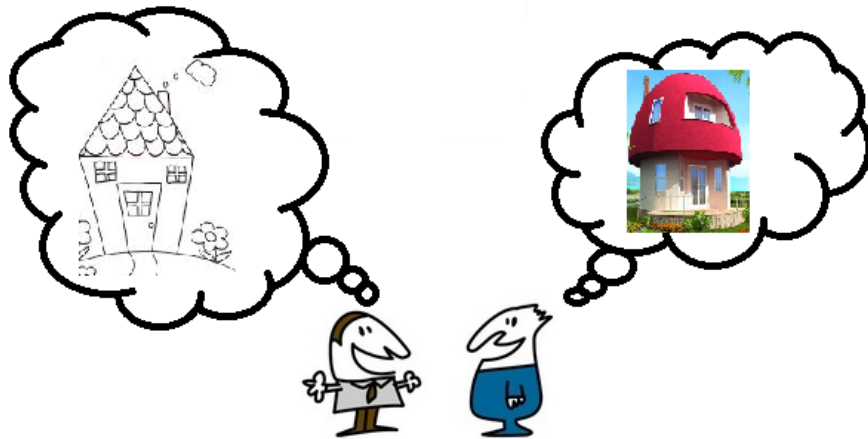


Etat des lieux

► Scénario 1

Analogie avec la construction d'un bâtiment :

Comment éviter ça?



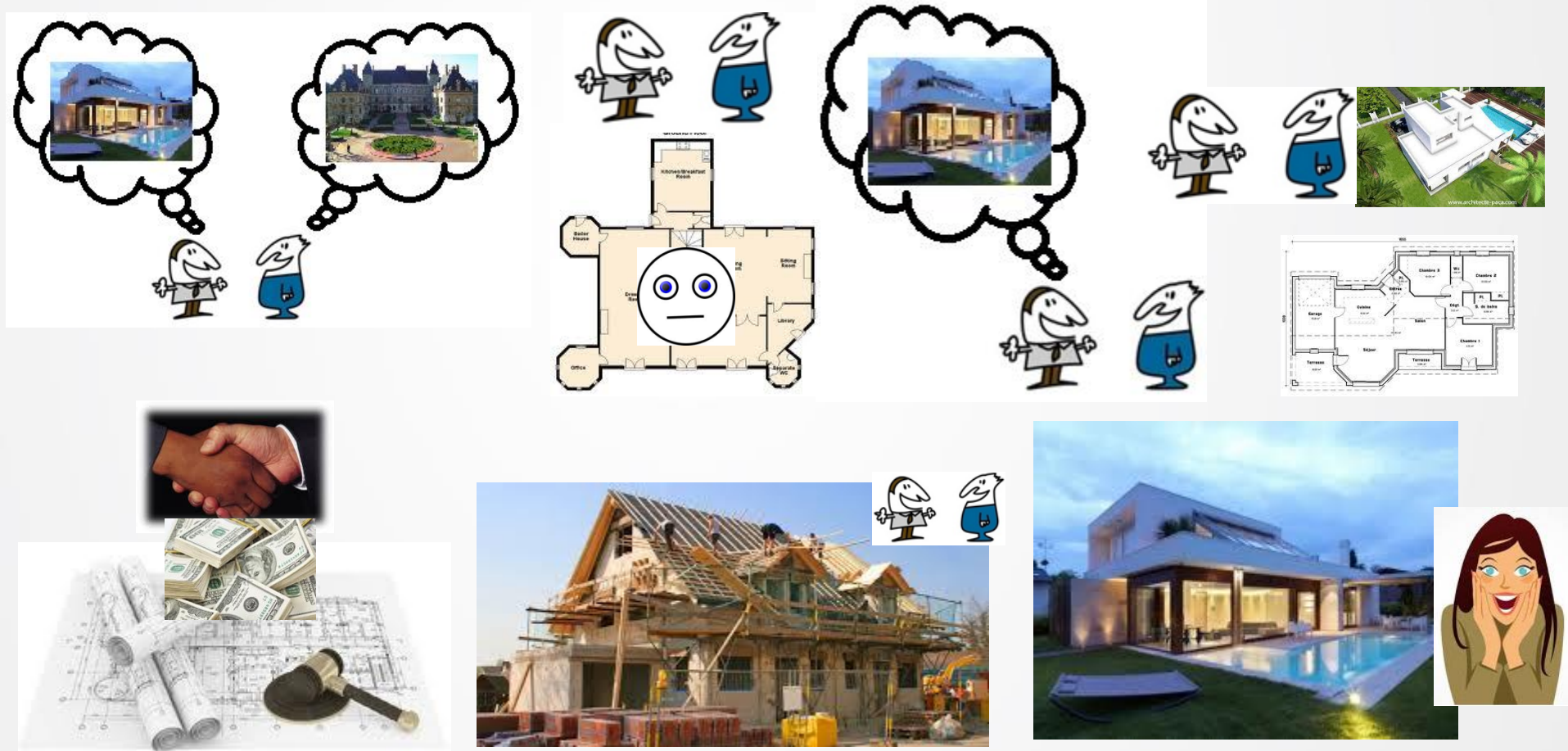
► Causes / Solutions ?



► Scénario 2

Analogie avec la construction d'un bâtiment

Pour aboutir à ça...





Remise en contexte

Système d'information (SI)

► Contexte



- Demande de

- Réalisation d'un nouveau logiciel
- Mise à jour corrective d'un logiciel
- Mise à jour évolutive d'un logiciel



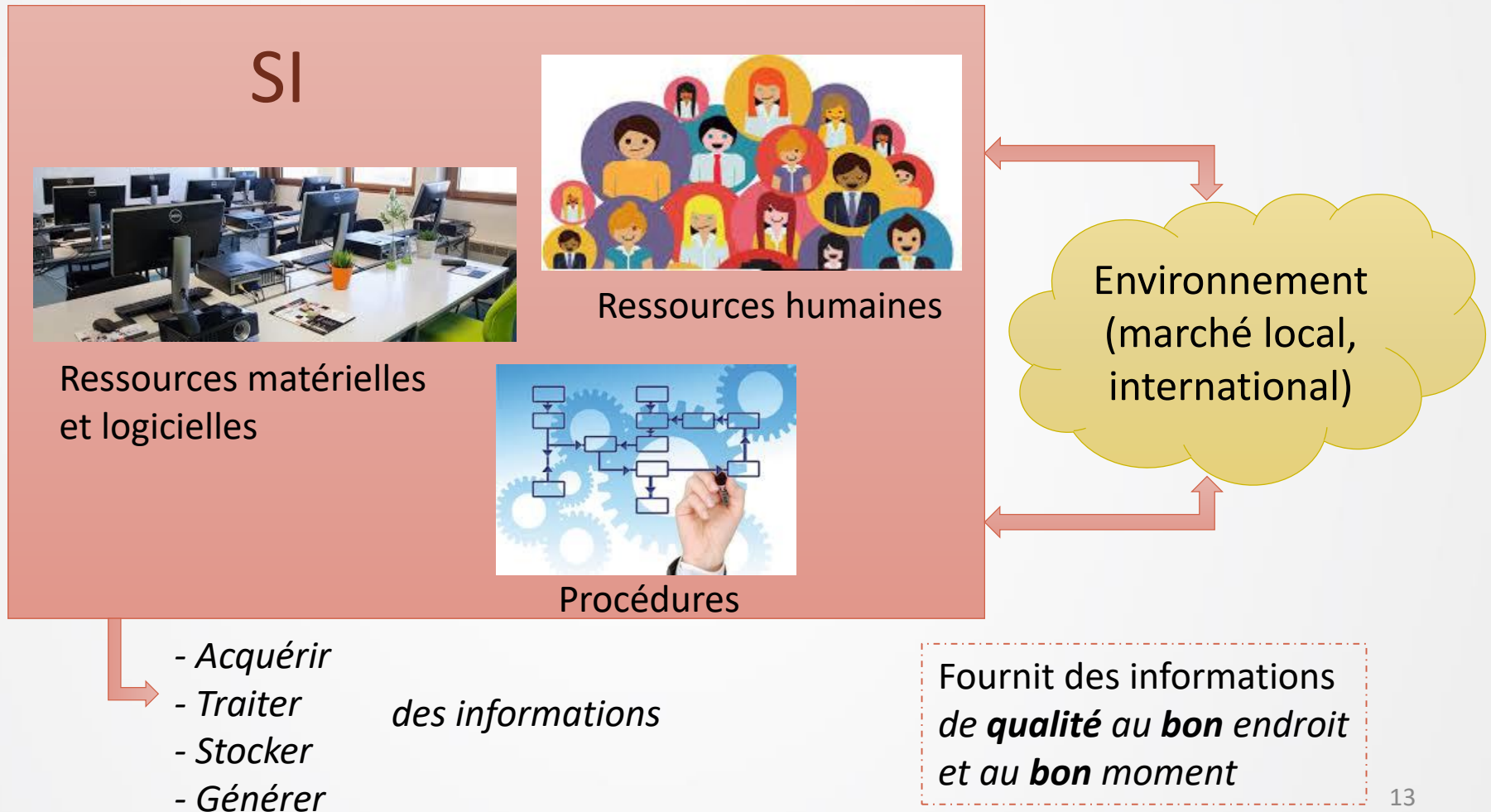
- Quelques interrogations :

- Logiciel dédié à qui?
- Où va-t-il être déployé?
- Quelles études faut-il mener?

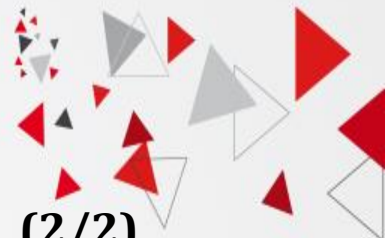


➡ Étudier le système d'information (SI) du client

► Système d'information (SI) ^(1/2)



► Système d'information (SI) (2/2)



Identifier les
besoins

Comprendre ses
besoins

Étudier un SI?

Etudier la
faisabilité

Identifier les
ressources
nécessaires



Élaboration d'un logiciel



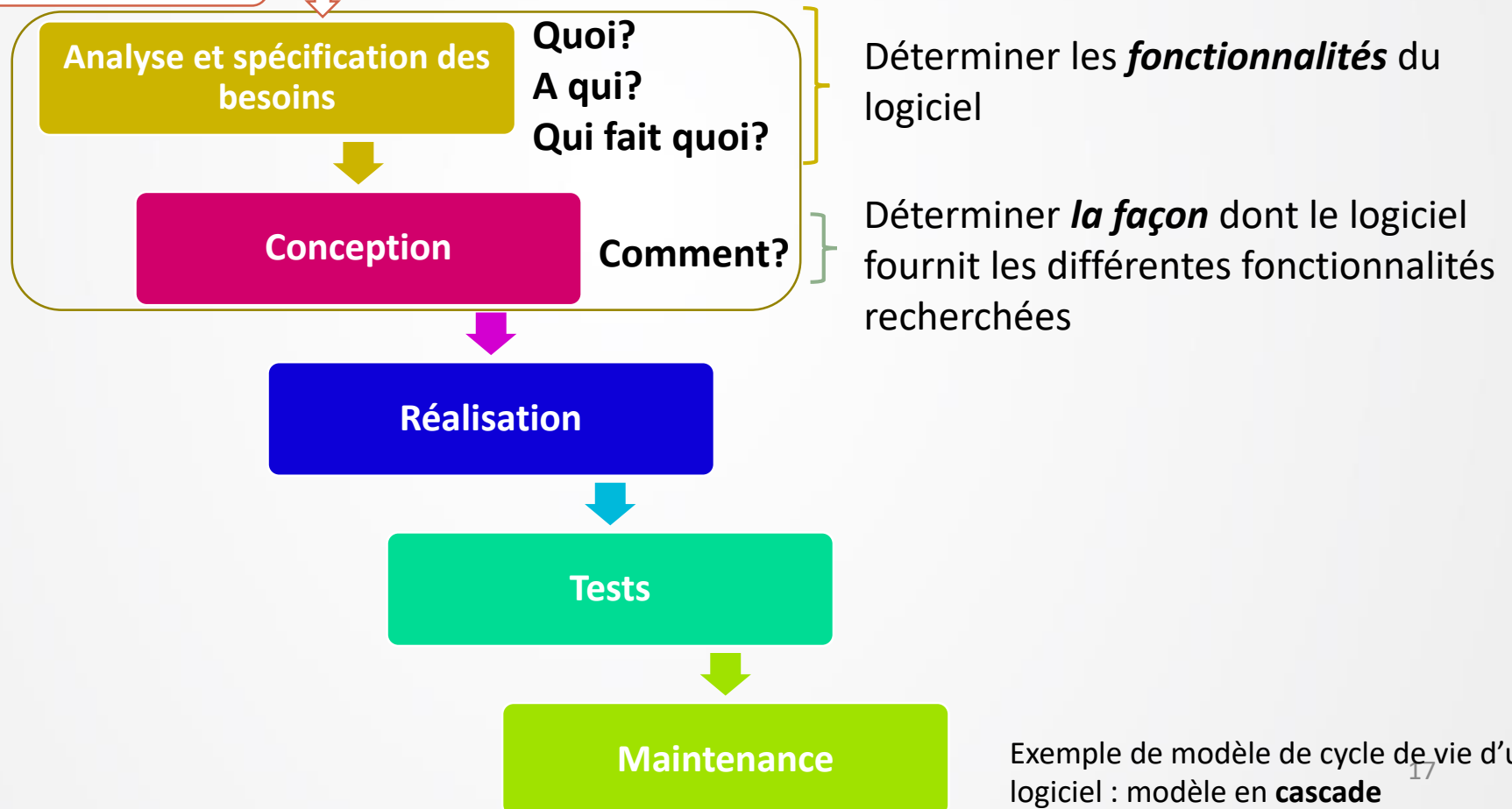
Cycle de vie d'un logiciel

► Elaborer un logiciel

(1/5)

- Etapes du cycle de vie d'un logiciel

Étude de faisabilité



Exemple de modèle de cycle de vie d'un logiciel : modèle en **cascade**

► Elaborer un logiciel

(2/5)

- Analyse et spécification des besoins du système
 - Dégager les besoins fonctionnels et non fonctionnels du futur système exprimés par le client
 - Identifier les profils des futurs utilisateurs



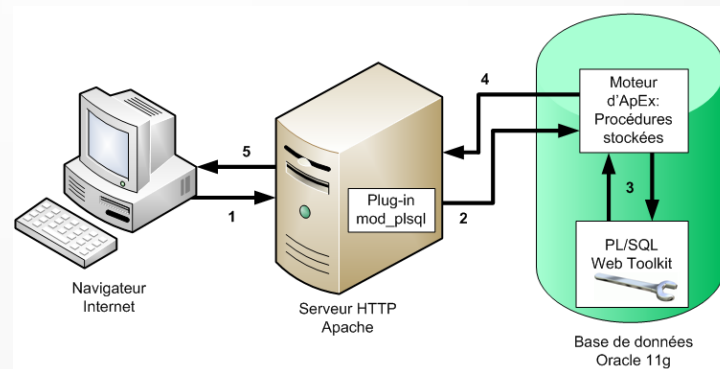
- Identifier le qui fait quoi : les besoins de chaque utilisateur
- Déterminer les éléments qui composent le futur système

► Elaborer un logiciel

(3/5)

- Conception

- Déterminer les détails des éléments du système
 - Déterminer les communications entre ces éléments
 - Déterminer le fonctionnement interne de chaque élément
 - Préparer l'architecture physique et logique du système
- Selon les technologies utilisées



Source : http://igm.univ-mlv.fr/~dr/XPOSE2007/cmyle_apex/fonctionnement.php

► Elaborer un logiciel

(4/5)



Quelle méthode
utiliser?

Que faire au niveau
de chaque étape?

Comment organiser les
étapes du cycle de vie?



(5/5)

Le Génie Logiciel
(Software engineering)
a les réponses à tout

Désigne l'ensemble
des méthodes,
des techniques
et des outils
nécessaires à la
production de
logiciels



► Pourquoi c'est si important



Comment le Client a exprimé son besoin



Comment le Chef de Projet l'a compris



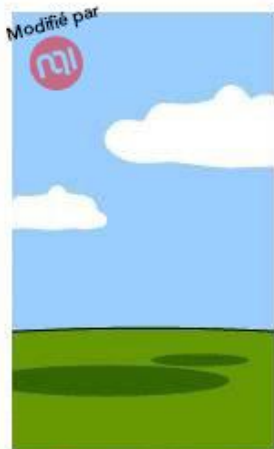
Comment l'Ingénieur l'a conçu



Comment le Programmeur l'a écrit



Comment le Commercial l'a décrit



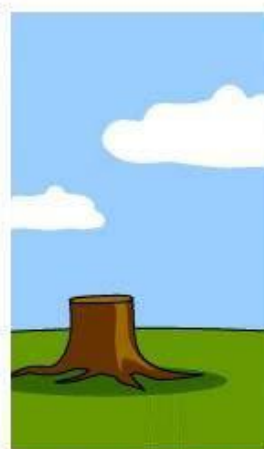
Comment le projet a été documenté



Ce qui a été finalement installé



Comment le Client a été facturé



Comment la Hotline répond aux demandes



Ce dont le Client avait réellement besoin

Source : <https://fr.planisware.com/hub/blog/gestion-de-projet-son-importance-en-une-image>

► Communiquons!



La communication est la clé!



Et c'est toujours mieux
avec des schémas





Modélisation



Besoin d'un formalisme

- Un formalisme
 - Aidant à la compréhension du fonctionnement du système
 - Compréhensible et par les clients et par l'équipe de développement
 - Donnant une vision ***abstraite*** du fonctionnement du nouveau système
 - Indépendant de la technologie utilisée
 - En cohérence avec l'approche adoptée

 Besoin de **modèles**

► Modélisation

(1/3)

Créer des
plans

Schématiser

Modélisons!

Avoir une
référence



► Modélisation

(2/3)



Vérifier si ce
qu'on a compris
correspond à ce
qui est demandé

Organiser/
diviser le
travail

Dans quel(s) but(s)?

Pour quel(s) objectif(s)?

Avoir une visibilité
détaillée sur le
travail à faire

Avoir une visibilité
globale sur le
travail à faire

► Modélisation

(3/3)



But ultime
Objectif final



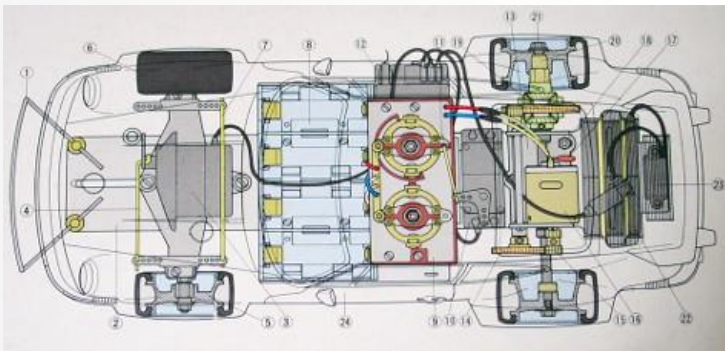
- Un logiciel :
- Qui répond aux besoins
 - De qualité
- Un **bon** logiciel

► Modèle

(1/2)

- Chaque domaine requiert des modèles
 - Exemples :

Construction automobile



Dessin technique du châssis

Construction d'une maison



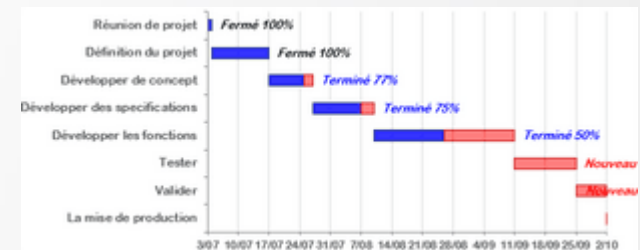
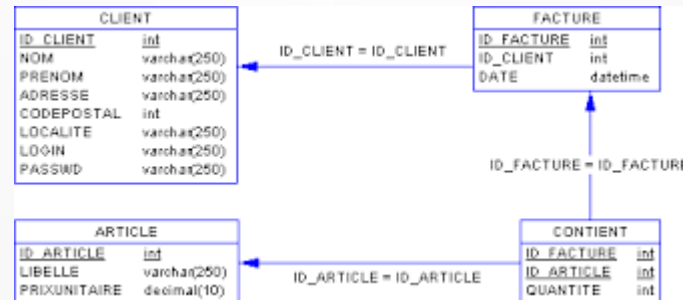
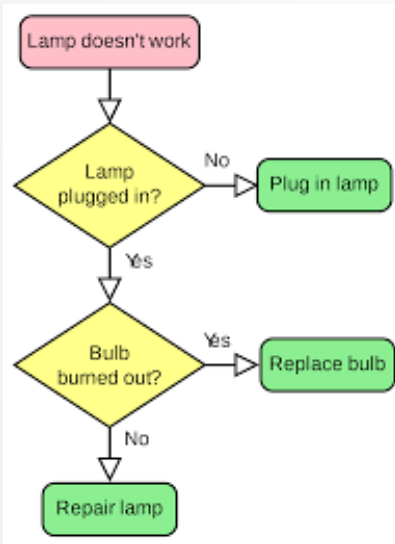
Vue simplifiée de la disposition des chambres d'un étage

► Modèle

(2/2)

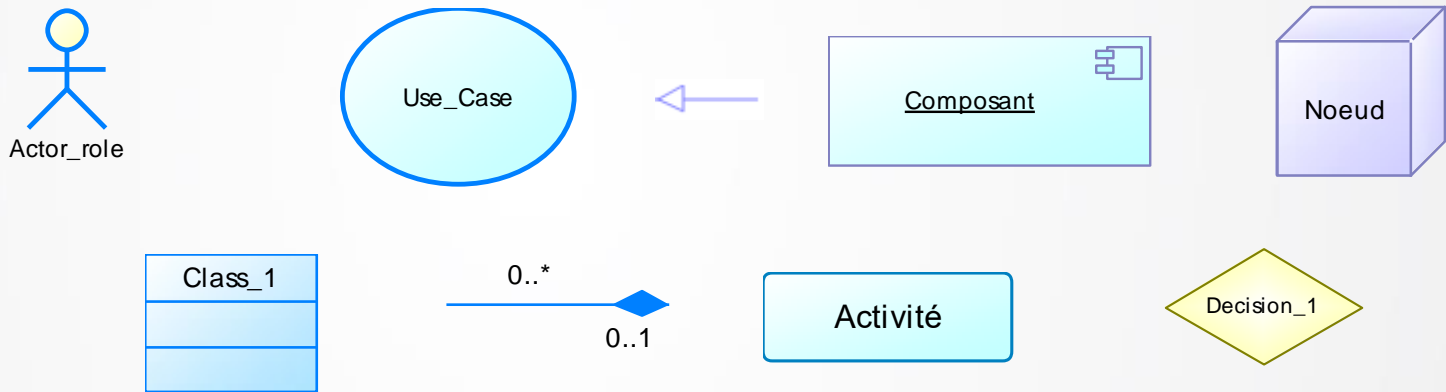
- Chaque modèle est élaboré :
 - selon des règles
 - en utilisant des symboles et des notations

→ Selon un *langage de modélisation*

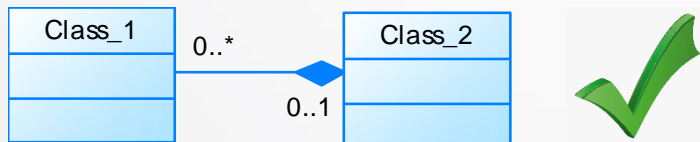


► Langage de modélisation

- Offre une notation pour la représentation **abstraite** de concepts (syntaxe)



- En suivant des règles de construction et d'utilisation des concepts (sémantique)





Exemples de langage de modélisation

- Merise : langage et méthode de conception
→ Systèmes procéduraux
- UML : langage de modélisation
→ Systèmes Orientés Objets (OO)
- SysML, SADT : langage de modélisation
→ Systèmes temps réel



UML



Pour l'Orientée Objet

► UML : acronyme



Unified

Unifié

Modeling (de) **Modélisation**

Language

Langage

Un peu d'histoire

(1/2)



- Les premières méthodes d'analyse (années 70)
 - Découpe cartésienne (fonctionnelle et hiérarchique) d'un système.
- L'approche systémique (années 80)
 - Modélisation des données + modélisation des traitements (Merise, Axial, etc.).
- L'émergence des méthodes objet (1990-1995)
 - Plus de 50 méthodes objet sont apparues durant cette période (Booch, Classe-Relation, Fusion, HOOD, OMT, OOA, OOD, OOM, OOSE, etc.) !
 - Aucune méthode ne s'est réellement imposée.

Un peu d'histoire

(2/2)



- Les premiers consensus (1995)
 - OMT (James Rumbaugh) : vues statiques, dynamiques et fonctionnelles d'un système
 - Notation graphique riche et lisible
 - OOD (Grady Booch) : vues logiques et physiques du système
 - Introduit le concept de package (élément d'organisation des modèles).
 - OOSE (Ivar Jacobson) : couvre tout le cycle de développement.
 - La méthodologie repose sur l'analyse des besoins des utilisateurs.

UML



- UML : Unified Modeling Language



- Naît de l'unification et de la normalisation des méthodes dominantes BOOCH, OMT et OOSE (1995)
- Standard de modélisation OO
- Version actuelle : 2.5.1 (décembre 2017)
- 13 diagrammes
- Langage de modélisation et non pas une méthode



► Diagrammes UML



Légende :

--- : vue fonctionnelle

--- : vue statique

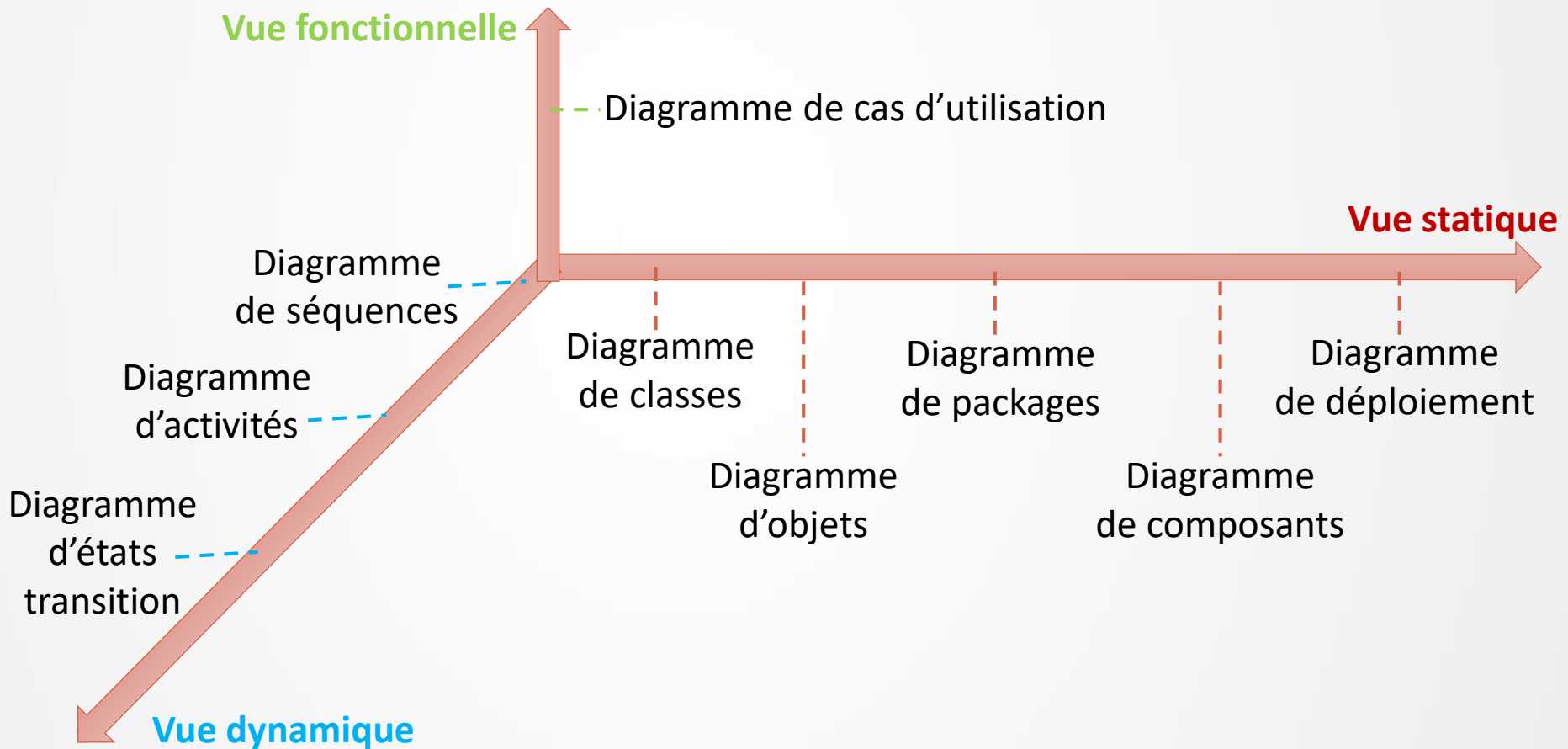
--- : vue dynamique/d'interaction



Les vues

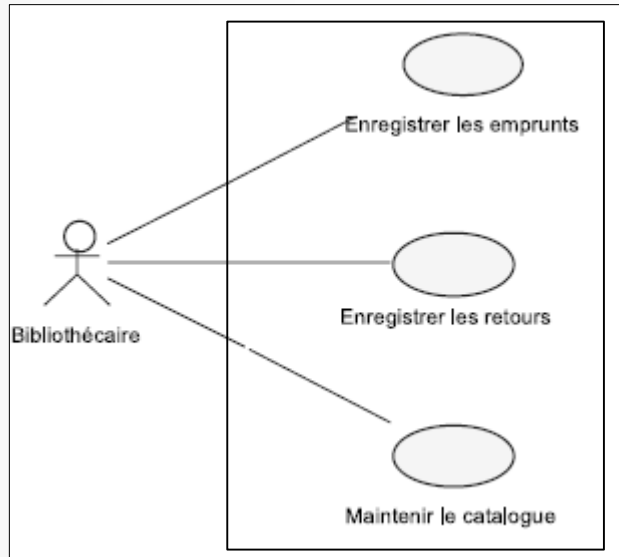
- Décomposer le système selon une
 - Vue fonctionnelle
 - Exprimer les besoins des utilisateurs et les activités qu'ils veulent réaliser
 - Vue statique
 - Caractériser les éléments d'un système et les relations entre ces éléments
 - Déterminer la structure d'un objet : structure des données et organisation des opérations qui s'y appliquent
 - Vue dynamique
 - Distinguer les interactions entre les éléments du système

► Les diagrammes du module

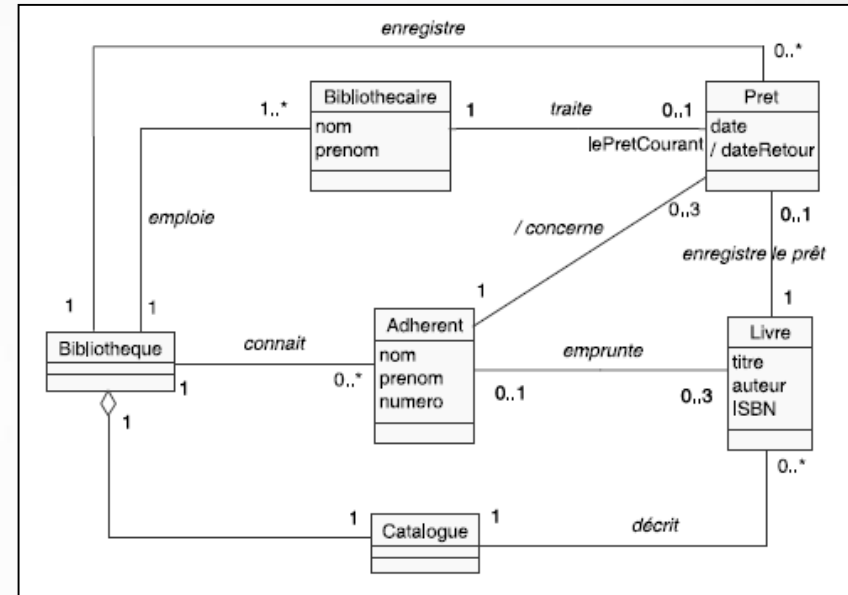


Exemples

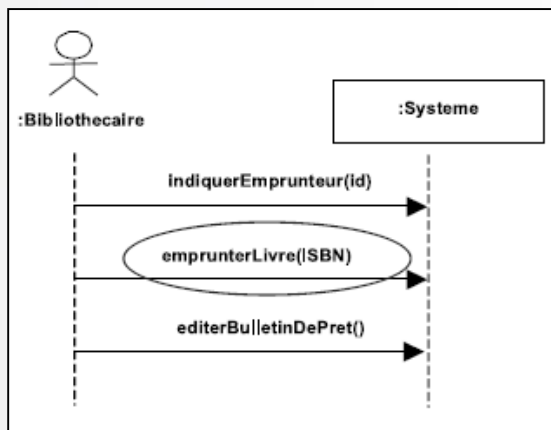
Vue fonctionnelle



Vue statique



Vue dynamique



**SYSTÈME D'INFORMATION
D'UNE BIBLIOTHÈQUE**

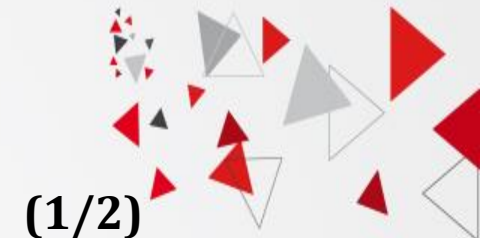


Plan du module



Plan module

(1/2)



- Introduction
- **Partie 1 : Analyse**
- **Chapitre 1** : Analyse fonctionnelle
 - Diagramme de contexte statique
 - Diagramme de cas d'utilisation
- **Chapitre 2** : Analyse dynamique
 - Diagramme de séquences système
 - Diagramme d'activités
- **Chapitre 3** : Analyse statique
 - Diagramme de classes d'analyse
 - Diagramme d'objets
 - Diagramme de package



Plan module

(2/2)

- **Partie 2 : Conception**
- **Chapitre 4 : Conception dynamique**
 - Diagramme de séquences objet
 - Diagramme d'états-transition
- **Chapitre 5 : Conception statique**
 - Diagramme de classes de conception
 - Diagramme de composants
 - Diagramme de déploiement

► Des questions?

