



UNIVERSITÉ
Abdelhamid MEHRI
CONSTANTINE 2

Université Constantine 2 – Abdelhamid Mehri
Faculté des NTIC
Département MI
Semestre : 4

Introduction au Génie Logiciel (GL) GL1

Dr Djamel Benmerzoug
djamel.benmerzoug@univ-constantine2.dz

Matière GL1

Cours/TD par semaine: 1 cours + 1 TD

Modalités de contrôle des connaissances :

Evaluation continue :

- Assiduité, préparation des TDs
- Interrogation écrite

Contrôle fin de semestre

Note finale = $(2 * \text{contrôle} + \text{Évaluation continue}) / 3$

Quelques références:

- Aïcha CHOUTRI: **Génie Logiciel 1**. Support de cours destiné aux étudiants L2 Informatique. Université Constantine 2
- Delphine LONGUET: **Introduction au génie logiciel et à la modélisation**. Support de cours destiné aux étudiants L3 Informatique. Polytech Paris-Sud
- Laurent AUDIBERT: **UML 2: De l'apprentissage à la pratique**.

Plan de cours

- Introduction
 - Définition
 - Logiciel vs Programme
 - Logiciel : caractéristiques
- Crise du logiciel
 - Constat du développement logiciel fin années 60
 - Raisons de la faible qualité des logiciels
- Génie Logiciel
 - Facteurs ou attributs de qualité de logiciel
 - Concepts du GL
 - Processus de développement logiciel
- Modèle et modélisation
 - Définition
 - Conception orientée objet
 - Diagrammes UML
 - Exemple d'utilisation des diagrammes UML

Introduction

Définition : GL = Ensemble des méthodes, des techniques et des outils dédiés à la **conception**, au **développement** et à la **maintenance** des systèmes informatiques

Objectif : Avoir **des procédures systématiques** pour des logiciels de **grande taille** afin que :

- la spécification corresponde aux **besoins réels** du client
- le logiciel respecte sa **spécification**
- les **délais** et les **coûts** alloués à la réalisation soient respectés

Introduction

Logiciel vs Programme

- Ensemble d'entités nécessaires au fonctionnement d'un processus de traitement automatique de l'information
 - Programmes, données, documentation...
- Ensemble de programmes qui permet à un système informatique d'assurer une tâche ou une fonction en particulier

Introduction

Logiciel : caractéristiques

Environnement

- **utilisateurs** : grand public (traitement de texte),
spécialistes (calcul météorologique),
développeurs (compilateur)
- **autres logiciels** : librairie, composant
- **matériel** : capteurs (système d'alarme),
réseau physique (protocole),
machine ou composant matériel contrôlé

Spécification : ce que doit faire le logiciel, **ensemble de critères** que doivent **satisfaire son fonctionnement interne** et ses **interactions** avec son environnement

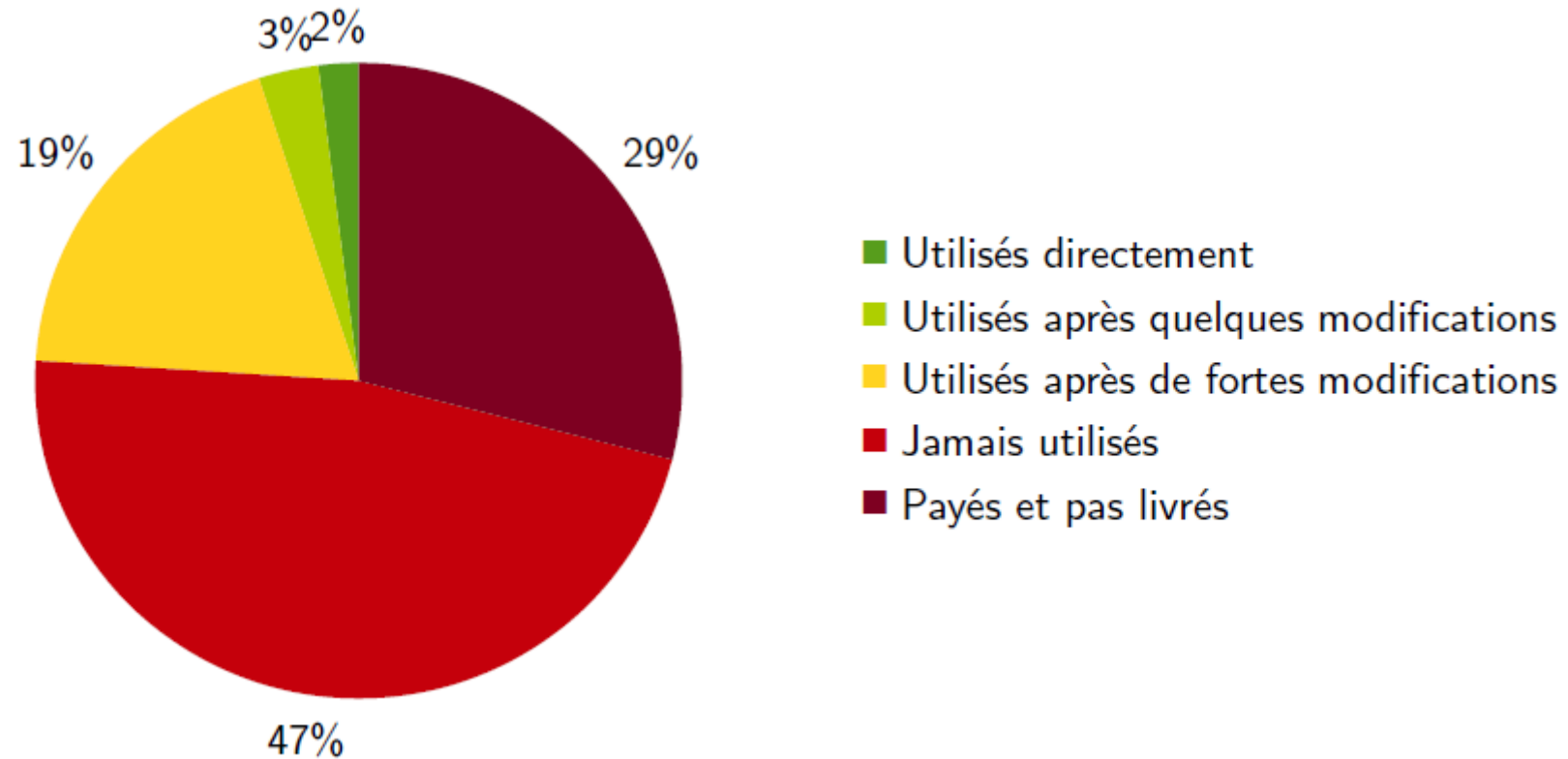
Crise du logiciel

Constat du développement logiciel fin années 60 :

- délais de livraison **non respectés**
- budgets **non respectés**
- **ne répond pas aux besoins** de l'utilisateur ou du client
- **difficile** à utiliser, maintenir, et faire évoluer

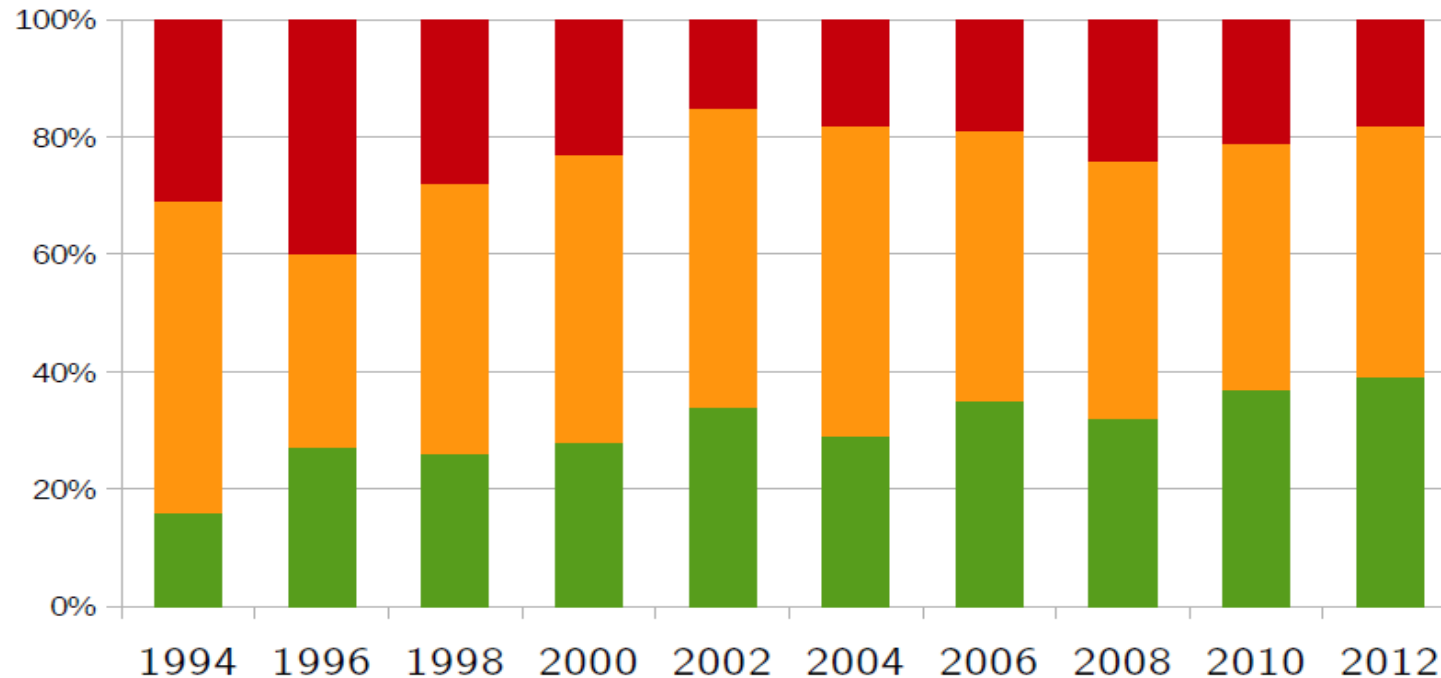
Crise du logiciel

Étude du *Department of Defense* des États-Unis sur les logiciels produits dans le cadre de 9 gros projets militaires



Crise du logiciel

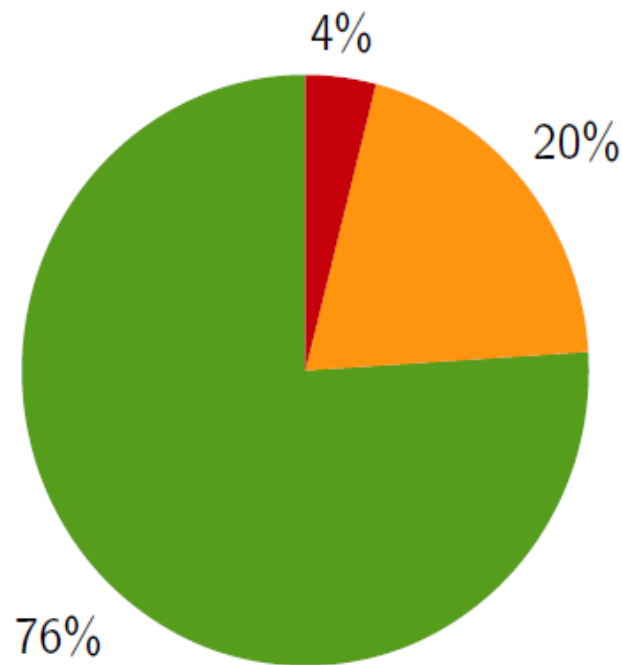
Enquête sur des milliers de projets, de toutes tailles et de tous secteurs



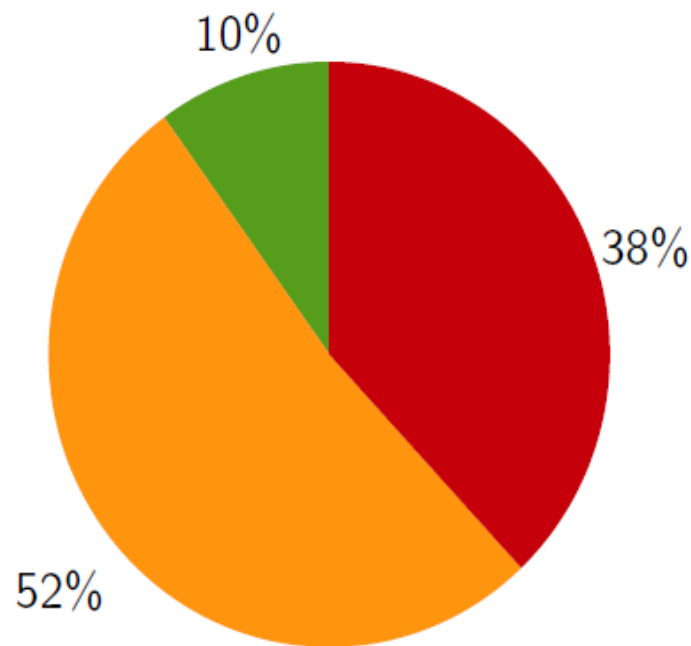
Standish group, *Chaos Manifesto 2013 - Think Big, Act Small*, 2013

- **Projets réussis** : achevés dans les délais et pour le budget impartis, avec toutes les fonctionnalités demandées
- **Projets mitigés** : achevés et opérationnels, mais livrés hors délais, hors budget ou sans toutes les fonctionnalités demandées
- **Projets ratés** : abandonnés avant la fin ou livrés mais jamais utilisés

Crise du logiciel



Petits projets
budget \leq \$1 million



Grands projets
budget \geq \$10 millions

- Projets réussis
- Projets mitigés
- Projets ratés

Standish group, *Chaos Manifesto 2013 - Think Big, Act Small*, 2013

Crise du logiciel

Raisons de la faible qualité des logiciels

Tâche complexe :

- Taille et complexité des logiciels
- Taille des équipes de conception/développement

Manque de méthodes et de rigueur :

- Manque de méthodes de conception
- Négligence et manque de méthodes et d'outils des phases de validation/vérification

Mauvaise compréhension des besoins :

- Négligence de la phase d'analyse des besoins du client
- Manque d'implication du client dans le processus

Crise du logiciel



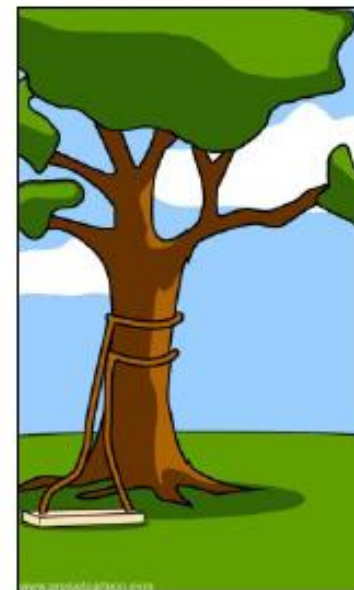
Ce que le client a expliqué



Ce que le chef de projet a compris



Ce que l'analyste a proposé



Ce que le programmeur a écrit



Ce dont le client avait vraiment besoin

Génie logiciel

Idée : appliquer les méthodes classiques d'ingénierie au domaine du logiciel

Ingénierie (ou **génie**) : Ensemble des fonctions allant de la conception et des études à la responsabilité de la construction et au contrôle des équipements d'une installation technique ou industrielle

Génie civil, naval, aéronautique, mécanique, chimique...

Génie logiciel

Facteurs ou attributs de qualité de logiciel:

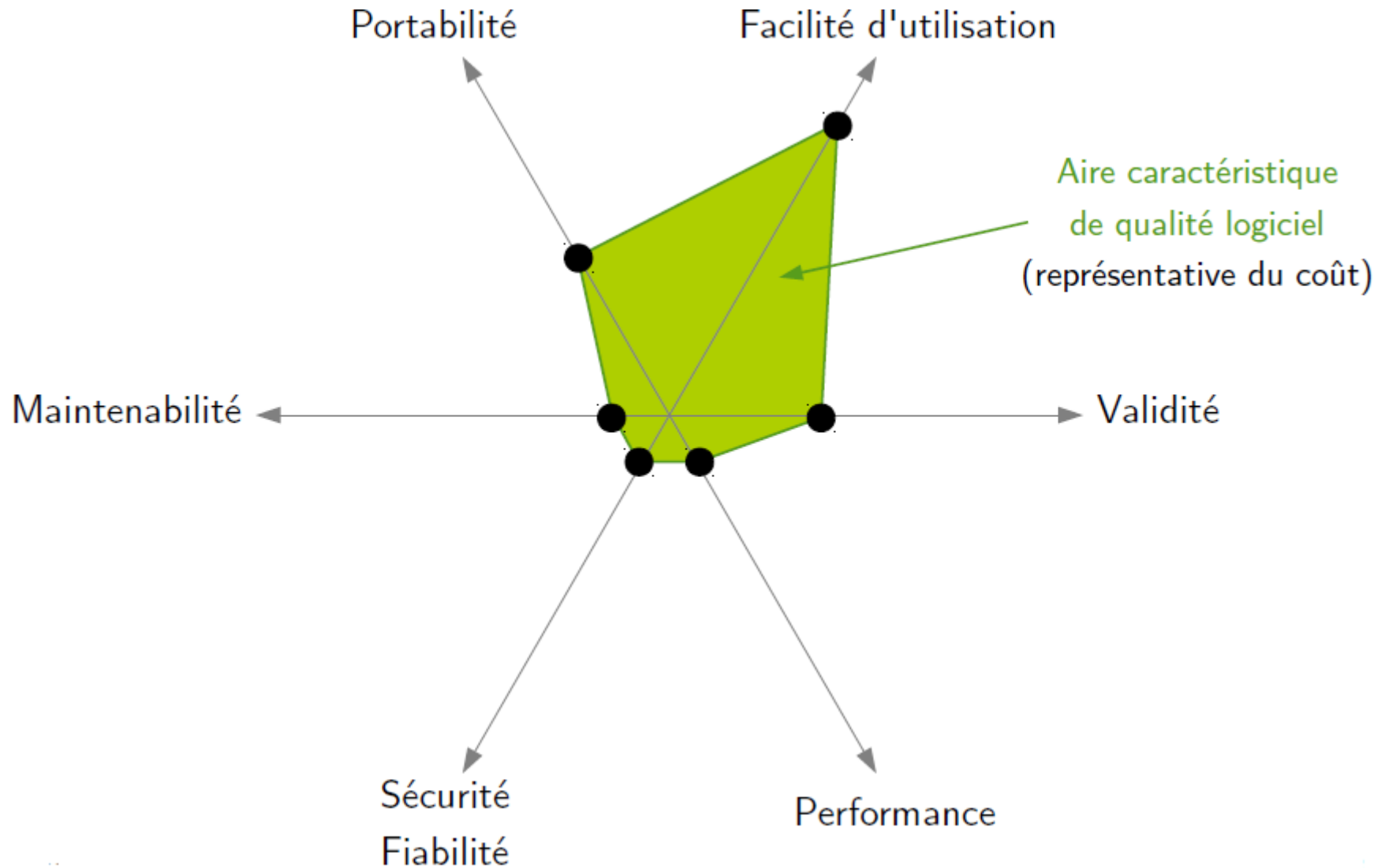
- **Validité** : réponse aux besoins des utilisateurs
- **Fiabilité** : tolérance aux pannes
- **Robustesse** : aptitude d'un logiciel à fonctionner dans des conditions anormales.
- **Maintenabilité**: facilité à corriger ou transformer le logiciel
- **Extensibilité** : facilité avec laquelle un logiciel se prête à une modification ou à une extension des fonctions qui lui sont demandées.
- **Réutilisabilité** : aptitude d'un logiciel à être réutilisé, en tout ou en partie, dans de nouvelles applications.

Génie logiciel

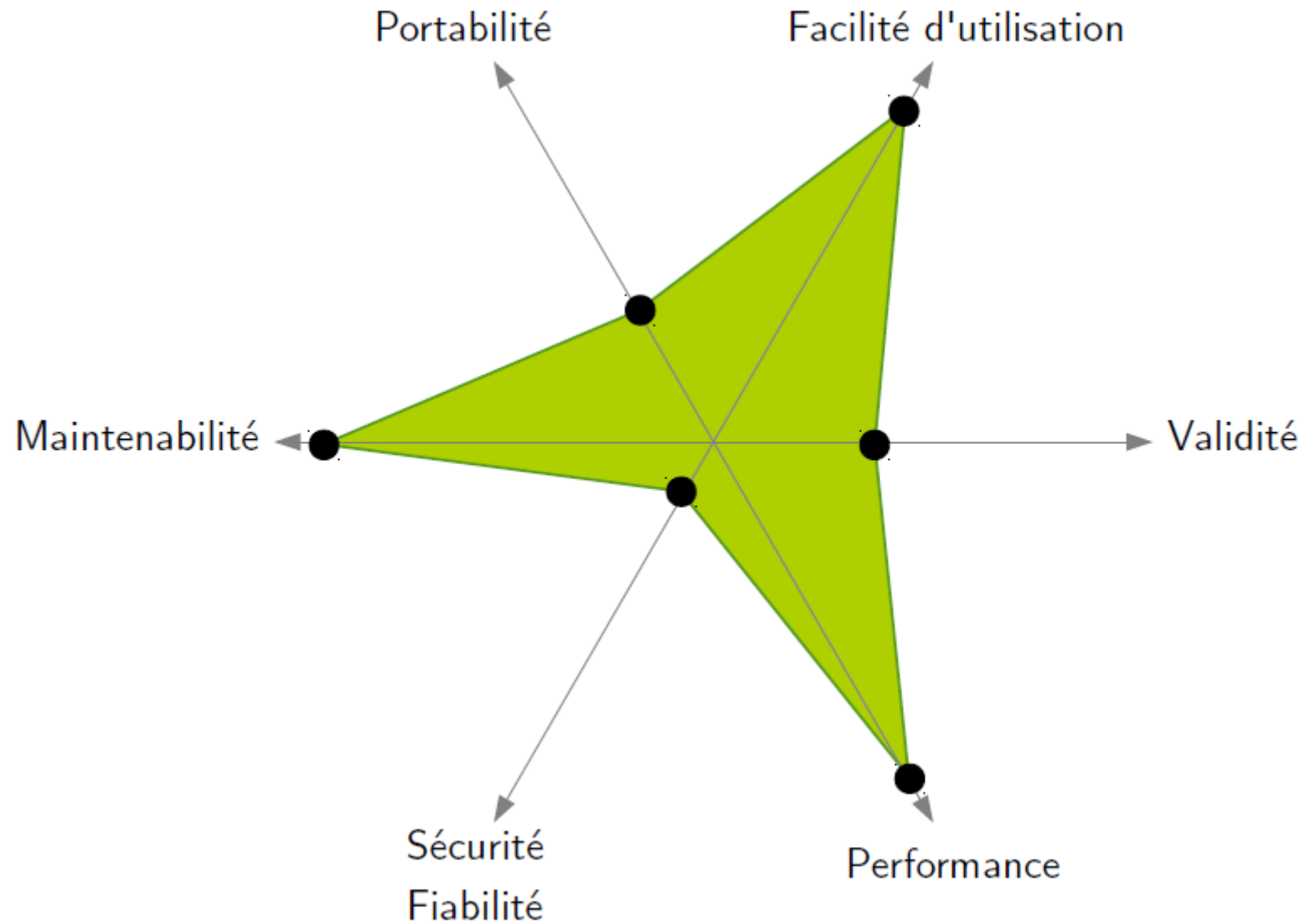
- **Efficacité** : aptitude d'un logiciel à bien utiliser les ressources matérielles telles que la mémoire, la puissance de l'U.C., etc.
- **Portabilité** : Facilité d'un logiciel à être porté sur de nouveaux environnements matériels et/ou logiciels.
- **Intégrité** : aptitude d'un logiciel à protéger ses différents composants contre des accès ou des modifications non autorisés.
- **Facilité d'utilisation**, d'entretien, etc.

Ces facteurs sont parfois contradictoires, le choix des compromis doit s'effectuer en fonction du contexte.

Exemple 1: Contrôleur de télécommande



Exemple 1: Jeu Vidéo



Génie logiciel

Concepts du GL

- Participants et Rôles
- Système et modèle de système
- Activités, tâches et Ressources
- Besoins fonctionnels et besoins non fonctionnels
- Notations, méthodes et méthodologies

Génie Logiciel

Processus de développement logiciel

Ensemble d'activités successives, organisées en vue de la production d'un logiciel

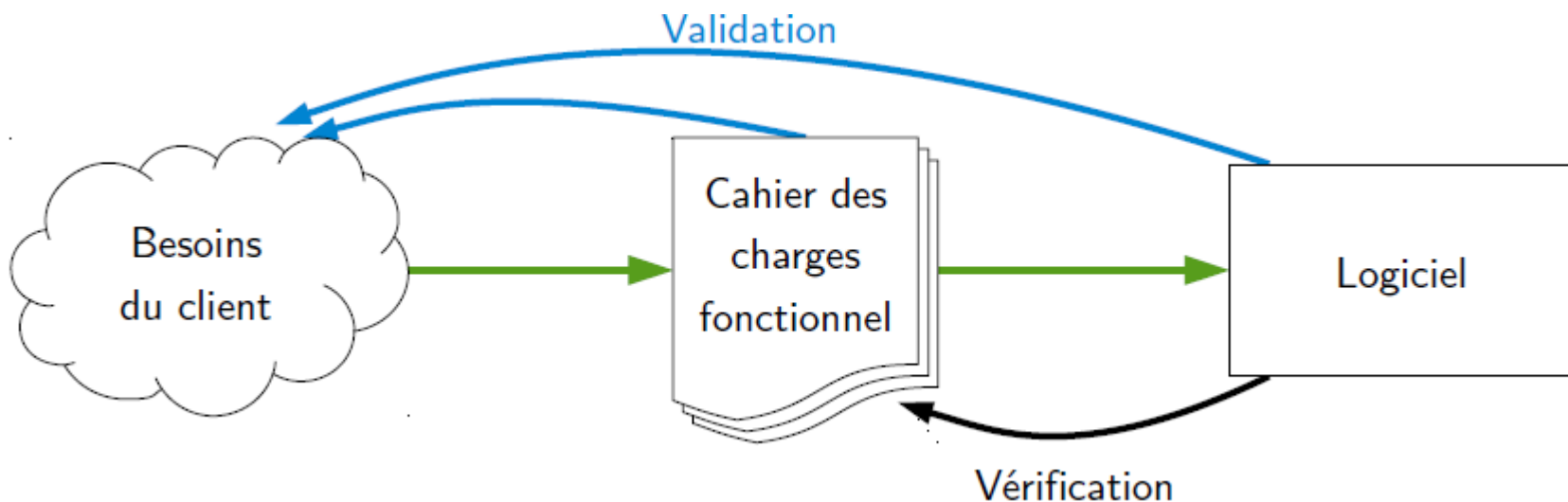
Activités du développement logiciel

- **Analyse des besoins** : Comprendre les besoins du client
- **Spécification**: Établir une description claire de ce que doit faire le logiciel
- **Conception**: Élaborer une solution concrète réalisant la spécification
- **Programmation**: Implantation de la solution conçue

Génie Logiciel

Activités du développement logiciel (suite)

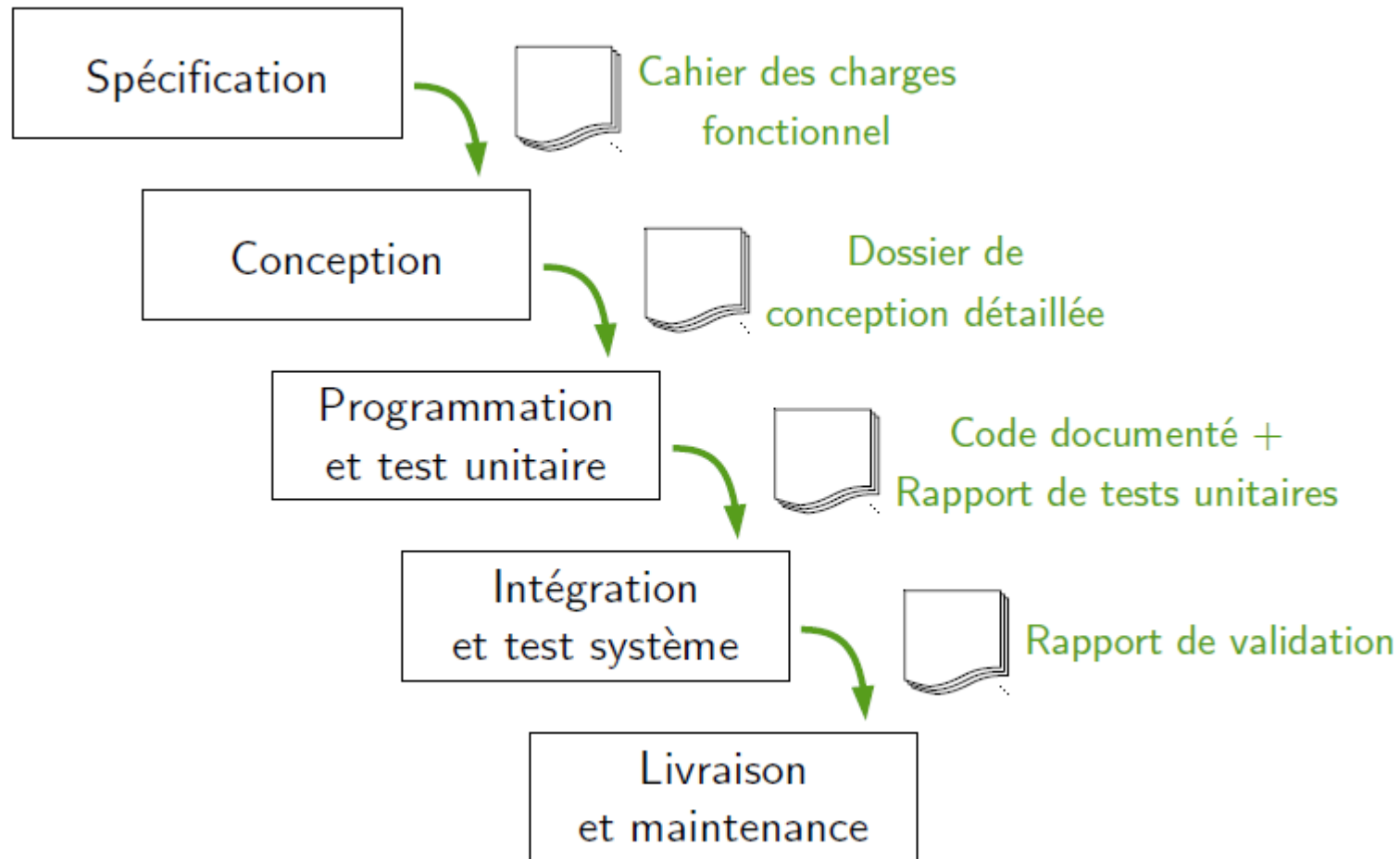
- **Validation et vérification**
 - **Validation** : assurer que les besoins du client sont satisfaits (au niveau de la spécification, du produit fini...)
 - **Vérification** : assurer que le logiciel satisfait sa spécification
- **Livraison**
- **Maintenance**



Génie Logiciel

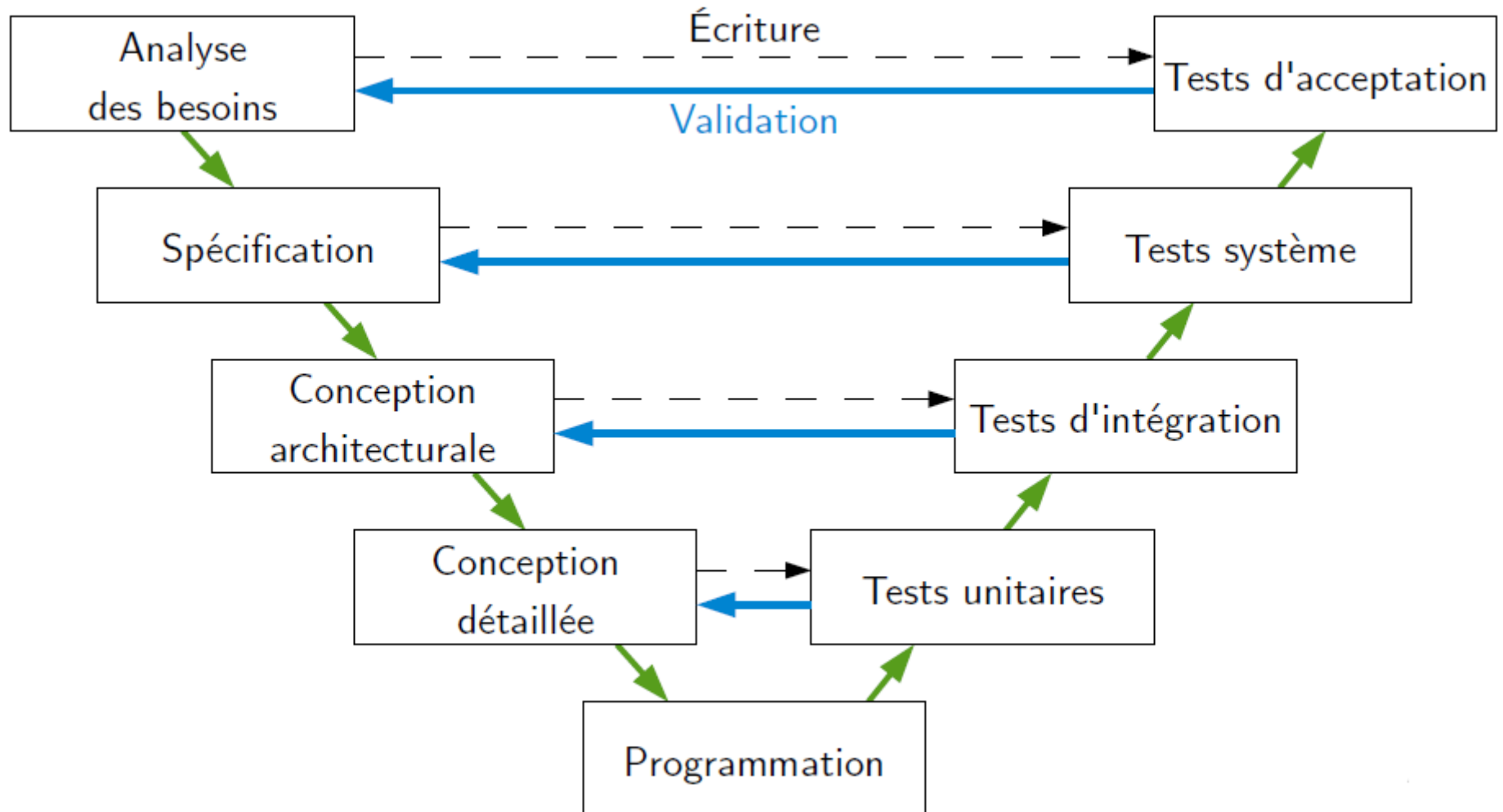
Processus en cascade

Chaque étape doit être terminée avant que ne commence la suivante À chaque étape, production d'un document base de l'étape suivante



Génie Logiciel

Processus en V



Modèle et modélisation

Modèle:

- Une abstraction de la réalité, centrée sur la représentation conceptuelle et physique d'un système.
- La simplification/abstraction d'une entité (Phénomène, processus, système etc.) du monde réel.

Modéliser un concept ou un objet pour :

- Mieux le comprendre (modélisation en physique)
- Mieux le construire (modélisation en ingénierie)



Aider la réalisation d'un logiciel à partir des besoins du client

Modèle et modélisation

Conception orientée objet

Principes

- Concept du domaine d'application = **objet**
Décrit par **état** (attributs) + **comportement** (opérations)
- Liens entre concepts : héritage, agrégation, composition...

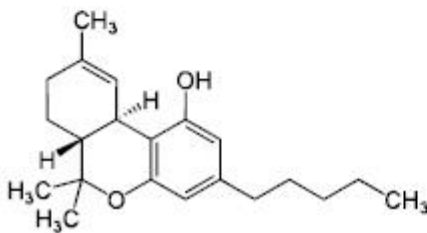
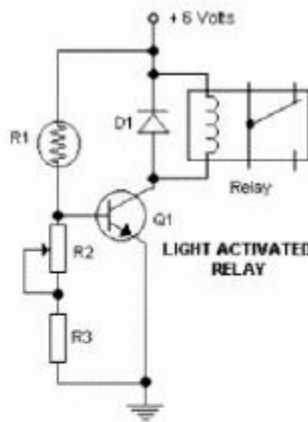
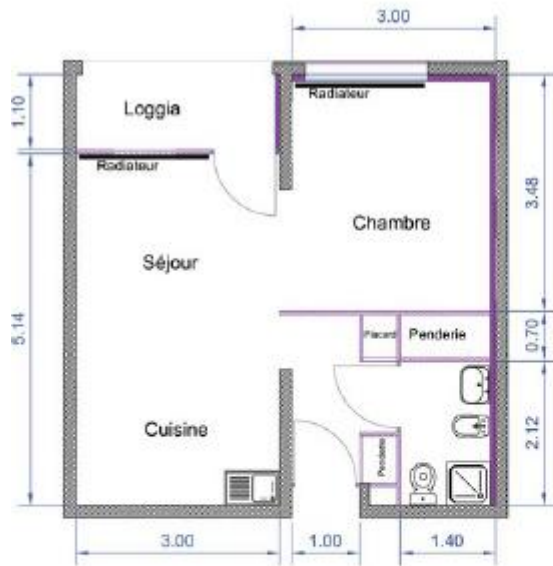
Caractéristiques des objets

- **Identité** : objet = entité unique
- **Classification** : regroupement des objets de même nature (attributs + opérations)
- **Polymorphisme** : comportement différent d'une même opération dans différentes classes
- **Héritage** : partage hiérarchique des attributs et opérations

Modèle et modélisation

Modélisation graphique

Principe : « Un beau dessin vaut mieux qu'un long discours »
Seulement s'il est compris par tous de la même manière



Modèle et modélisation

UML : Unified Modeling Language

Langage :

- Syntaxe et règles d'écriture
- Notations graphiques normalisées

... de modélisation :

- Abstraction du fonctionnement et de la structure du système
- Spécification et conception

... unifié :

- Fusion de plusieurs notations antérieures : Booch, OMT, OOSE
- Standard défini par l'OMG (Object Management Group)
- Dernière version : UML 2.5.1 (Décembre 2017)

En résumé : Langage graphique pour visualiser, spécifier, construire et documenter un logiciel



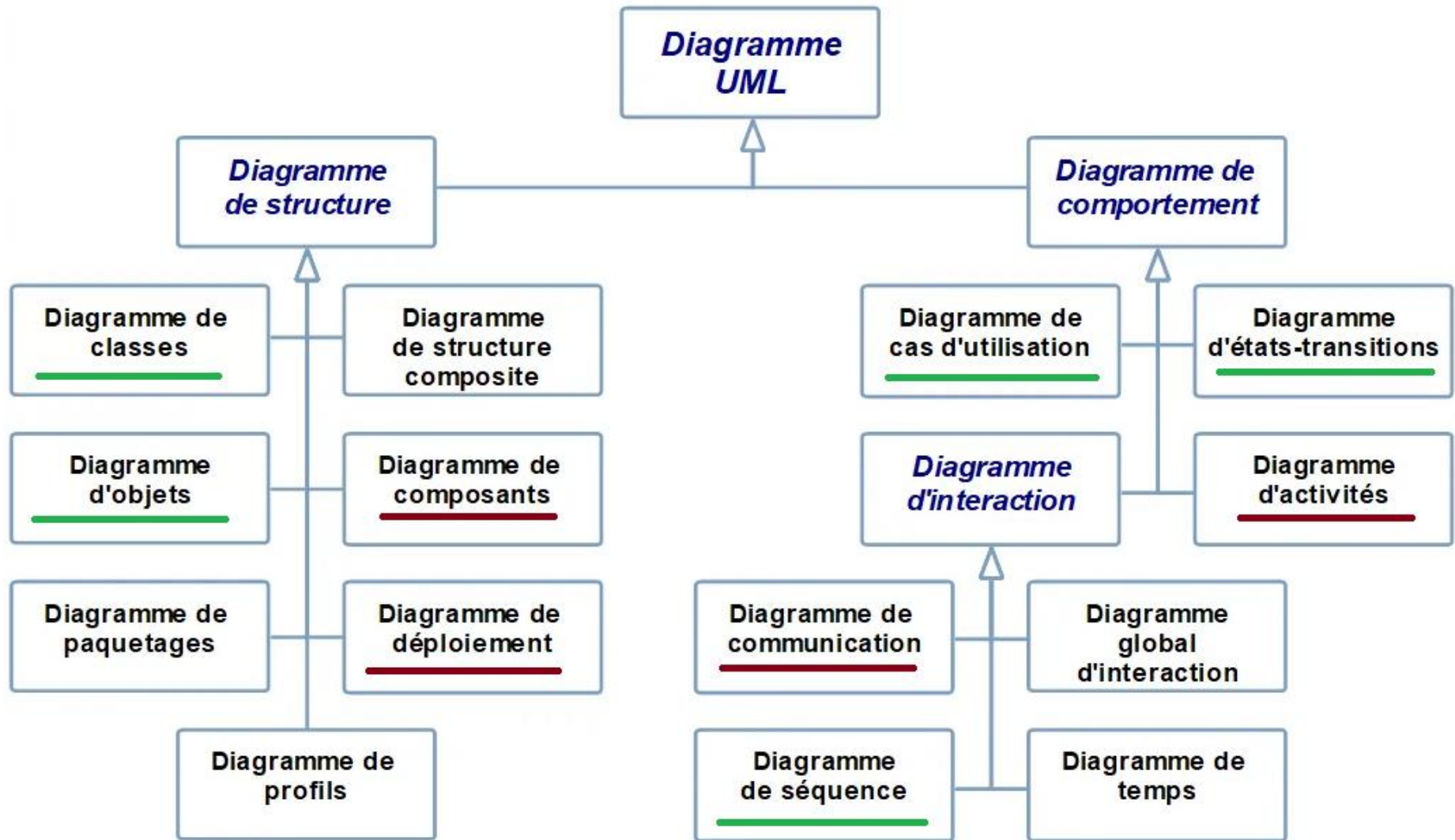
Modèle et modélisation

UML

- **Langage graphique** : Ensemble de **diagrammes** (UML1: 9 diagrammes / UML2: 14 diagrammes) permettant de modéliser le logiciel selon différentes vues et à différents niveaux d'abstraction
- **Modélisation orientée objet** : modélisation du système comme un **ensemble d'objets** interagissant

UML **n'est pas une méthode** de conception
UML **est un outil indépendant** de la méthode

Modèle et modélisation



Modèle et modélisation

Exemple d'utilisation des diagrammes

Pendant la Spécification

- Diagrammes de **cas d'utilisation** : **besoins** des utilisateurs
- Diagrammes de **séquence** : **scénarios** d'interactions entre les utilisateurs et le logiciel, vu de l'extérieur
- Diagrammes **d'activité** : enchaînement d'actions représentant un **comportement** du logiciel

Pendant la Conception

- Diagrammes de **classes** : **structure interne** du logiciel
- Diagrammes **d'objet** : **état interne** du logiciel à un instant donné
- Diagrammes **états-transitions** : évolution de l'état d'un objet
- Diagrammes de **séquence** : scénarios d'interactions avec les utilisateurs ou **au sein du logiciel**
- Diagrammes de **composants** : composants **physiques** du logiciel
- Diagrammes de **déploiement** : organisation **matérielle** du logiciel

- **Références:**

- Aïcha CHOUTRI: **Génie Logiciel 1**. Support de cours destiné aux étudiants L2 Informatique. Université Constantine 2
- Delphine LONGUET: **Introduction au génie logiciel et à la modélisation**. Support de cours destiné aux étudiants L3 Informatique. Polytech Paris-Sud
- Laurent AUDIBERT: **UML 2: De l'apprentissage à la pratique**.