

Modélisation et développement objet

8INF334

Introduction générale

Génie logiciel

Aymen Sioud
Département Informatique et Mathématique
Courriel : aymen.sioud@uqac.ca

Objectif du GL

Introduction

- **Vos compétences:** *programming in the small*
 - programmation individuelle sur de petits problèmes
 - Algo., langages de programmation, structure de données
 - (parfois) un peu de méthodologie: analyse descendante
- **En entreprise:** *programming in the large*
 - travail en équipe pour de projets longs et complexes
 - spécifications des exigences de départ peu précises
 - dialogue avec le client-utilisateur: parler métier
 - organisation, planification, gestion de risque
 - Conséquence: démarche ingénierie : **génie logiciel**

Introduction: pourquoi le GL?

- Alors pourquoi le génie logiciel ?
 - Pour rationaliser la *production*, le *déploiement* et la *maintenance* du logiciel.
- Quelle portée de la discipline?
 - On s'intéresse au logiciel en tant que **produit**.
 - On s'intéresse à son cycle de vie en tant que **processus** et aux participants humains
- Qu'est-ce qu'on vise?
 - La **qualité** du logiciel: multiples facteurs, difficilement mesurables.
 - La **qualité** du processus (**maturité**): contrôle, reproductibilité, améliorations.

Génie logiciel: définition

- Le GL doit fournir des méthodes de conception de systèmes complexes permettant:
 - Une prise en compte du client
 - Une démarche de qualité
 - Une organisation du travail en équipe
- *IEEE Standards Collection: Software Engineering* «The application of a **systematic, disciplined, quantifiable** approach to the **development, operation, and maintenance** of software, that is the application of engineering to software»

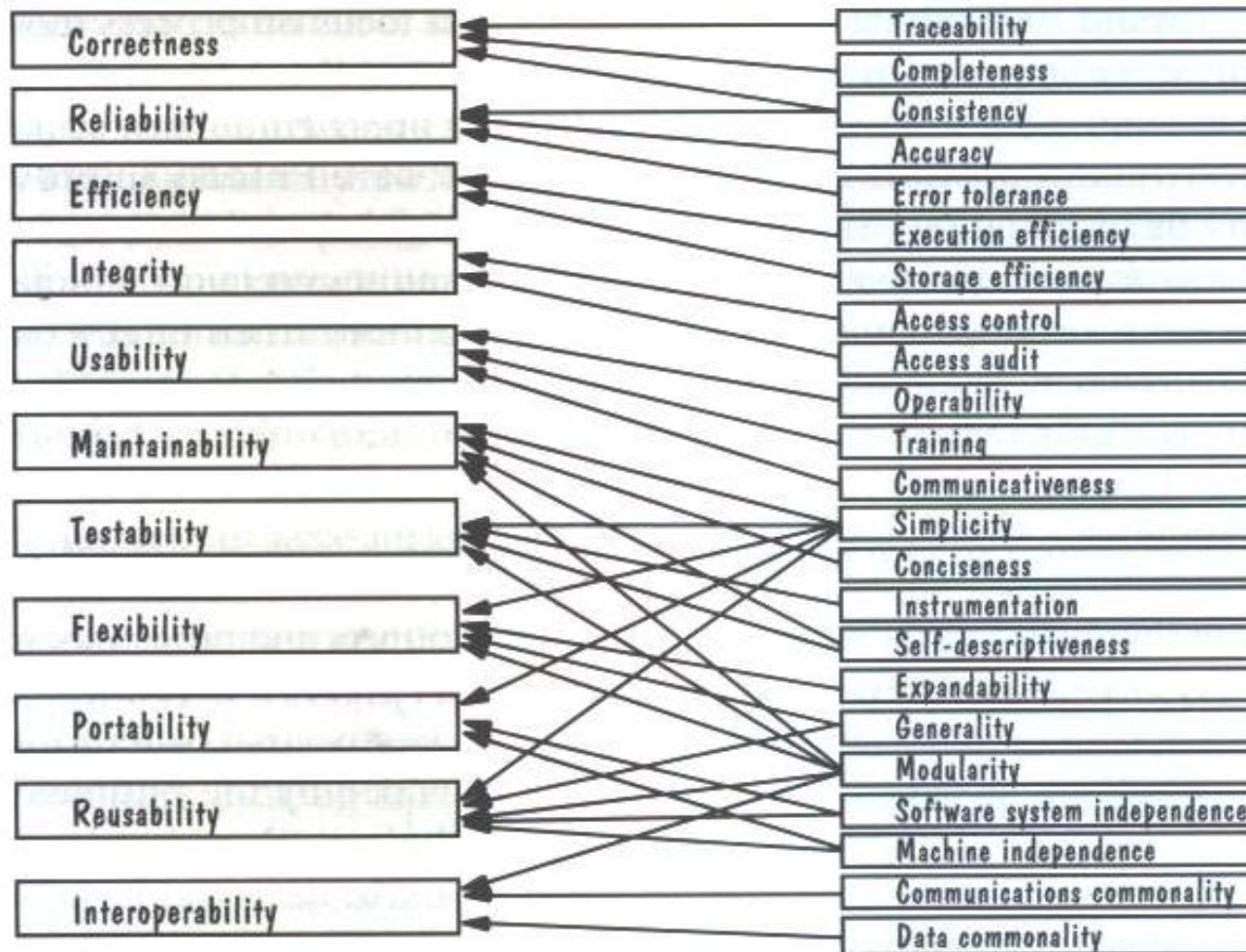
Génie logiciel: définition

- Le GL consiste à **développer** et à **utiliser** des **méthodes** et des **outils** dans le but de produire un logiciel de **qualité** en respectant les contraintes de budgets en **coût** et en **temps**
- GL: Méthodologie de construction en **équipe** d'un logiciel complexe et à multiples versions
- Programmation Vs. Génie logiciel (approximation):
 - **Programmation**: activité **personnelle**
 - **Génie logiciel**: activité **d'équipe**, la partie programmation ne représentera qu'entre 10% et 30% du coût total

Logiciel: aspects économiques

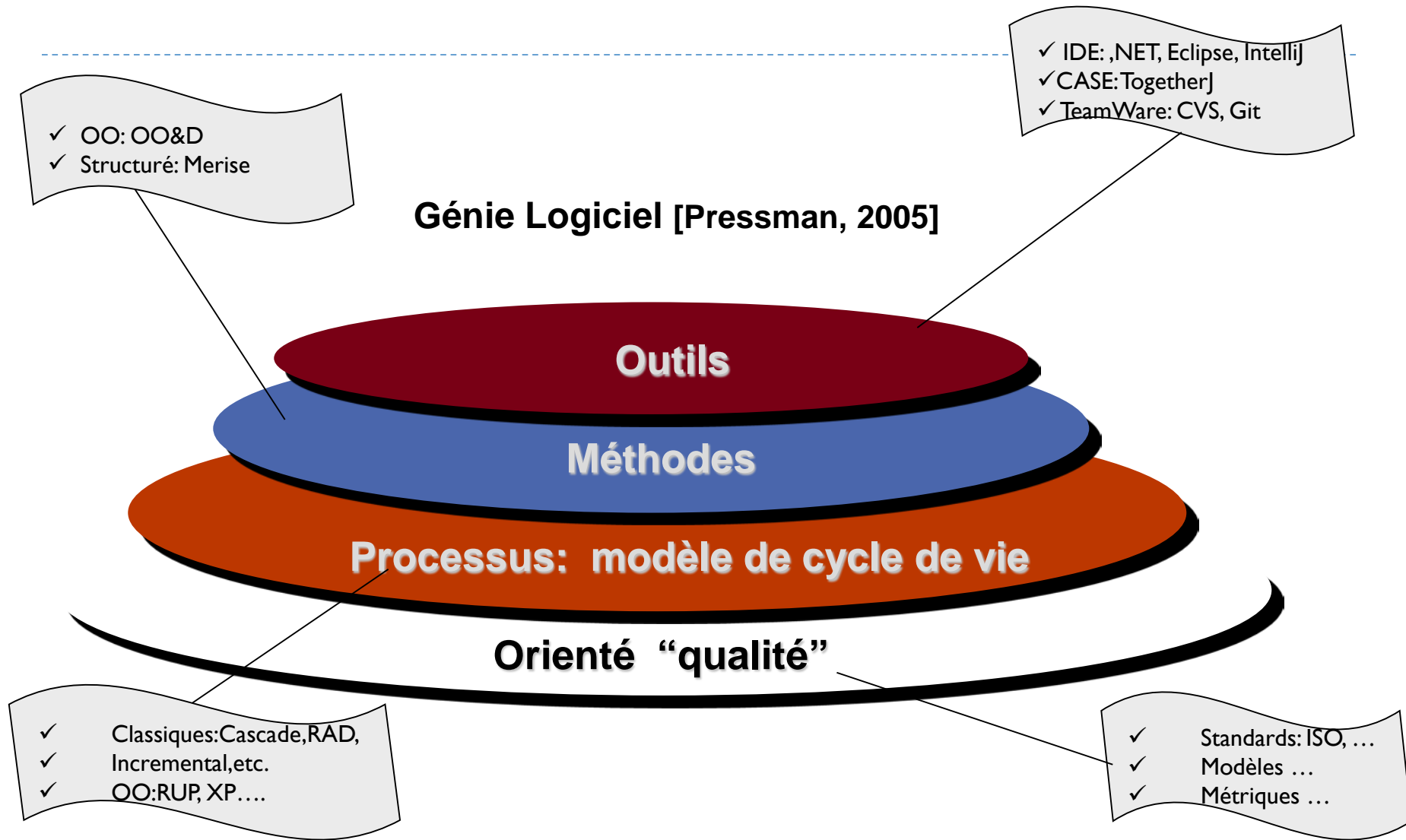
- *The economies of ALL developed nations are dependent on software: [Sommerville, 2008]*
- Importance économique du logiciel:
 - Importance croissance de l'informatique dans l'économie
 - Coût de logiciel supérieur à celui de matériel
 - Coût de maintenance supérieur au coût de conception
- Une amélioration de la **qualité** du logiciel est indispensable

Qualité de logiciel [Pfleeger et al. 06]



McCall's quality model.

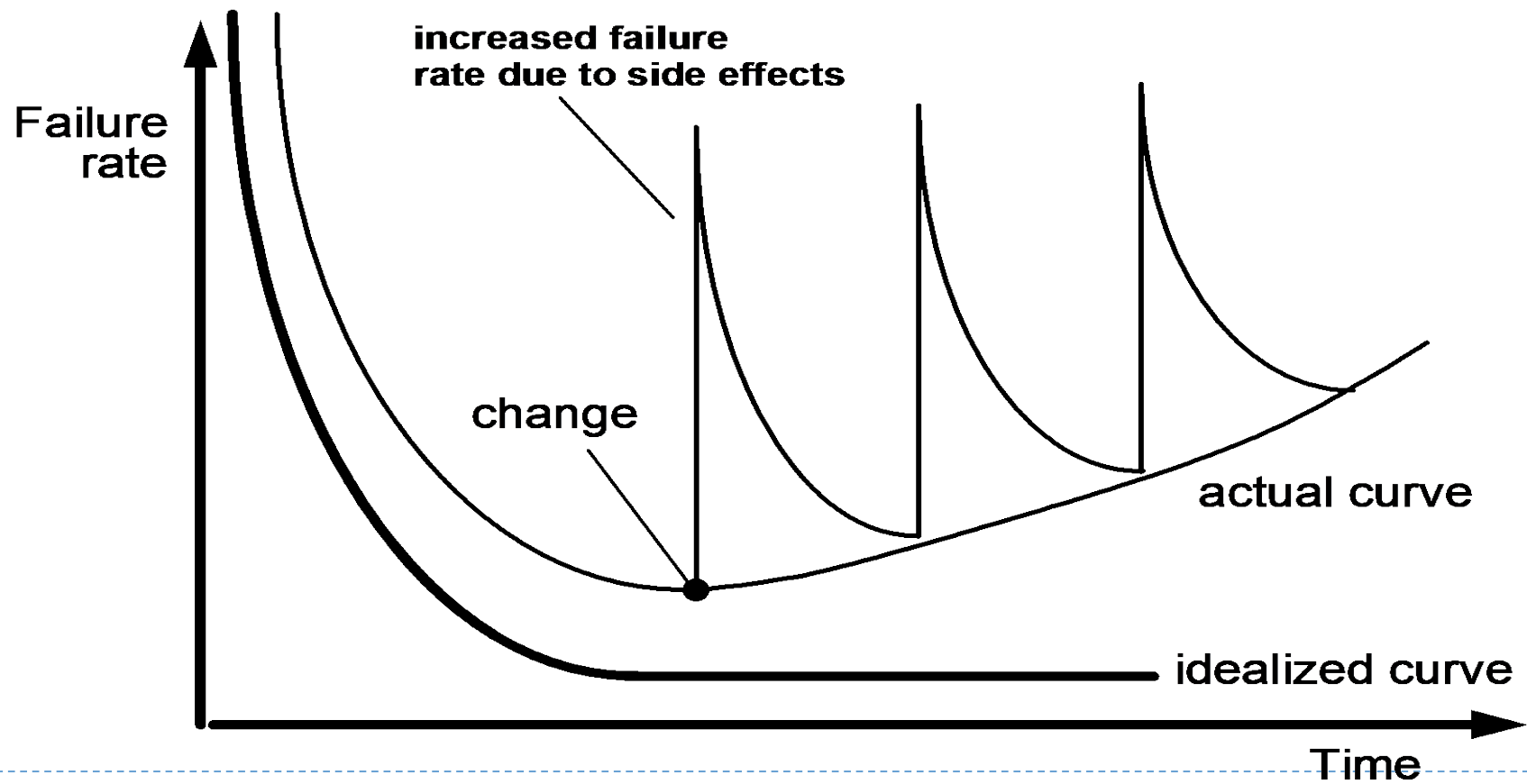
Génie logiciel: une technologie par couches



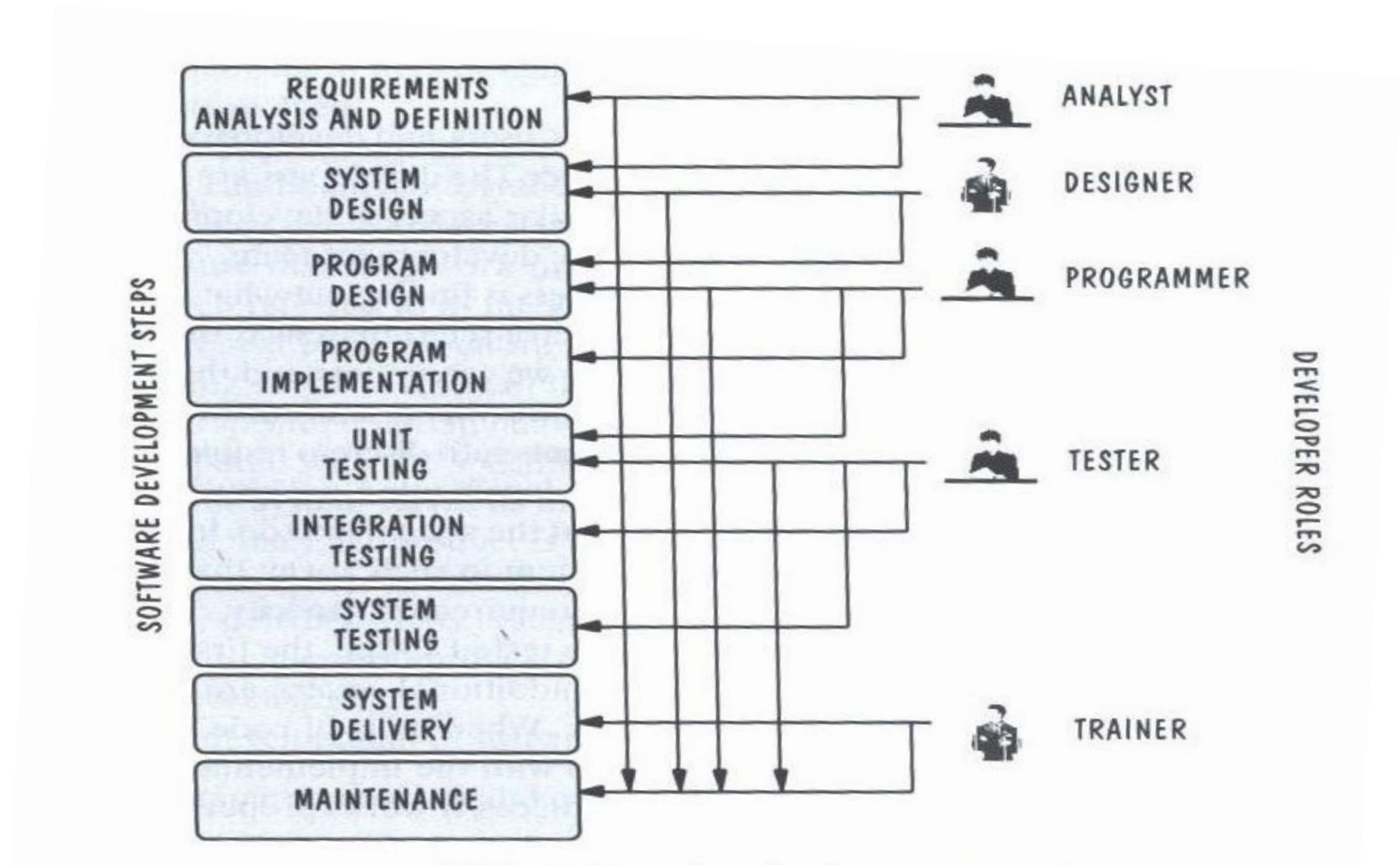
Définition d'un logiciel

- Software is both a **product** (programme, document, ...) and a **vehicule** (OS, Network, ...) that delivers a product [Pressman, 2005]
- Un logiciel est composé de:
 - programmes
 - structure de données
 - documents

Coût de logiciel [Pressman, 2005]



Rôles des développeurs [Pfleeger et al.,06]



The roles of the development team.

Historique et évolution

- Évolution du matériel et du logiciel:
 - 1950-1964: traitements en lots, logiciel personnalisé
 - 1964-1975: multi-usagers, temps réels, base de données, progiciel
 - 1975-1985: systèmes distribués, systèmes embarqués, matériel à bas coût
 - 1985-2002: ordinateurs personnels, Orienté-objet, systèmes experts
 - 2002-...: ordinateurs multicores, orienté-aspect, orienté composant, Entrepôt de données (datawarehouse), bioinformatique, cloud, mobile ...

Le logiciel aujourd'hui c'est

- ▶ Système
- ▶ Applications
- ▶ Logiciel scientifiques / ingénierie
- ▶ Logiciel embarqué
- ▶ Embedded software
- ▶ Ligne de production logiciel (CMMI)
- ▶ Application Web/Mobile
- ▶ Logiciel IA (robotique, gaming, neural)
- ▶ Logiciel en grille (GRID)

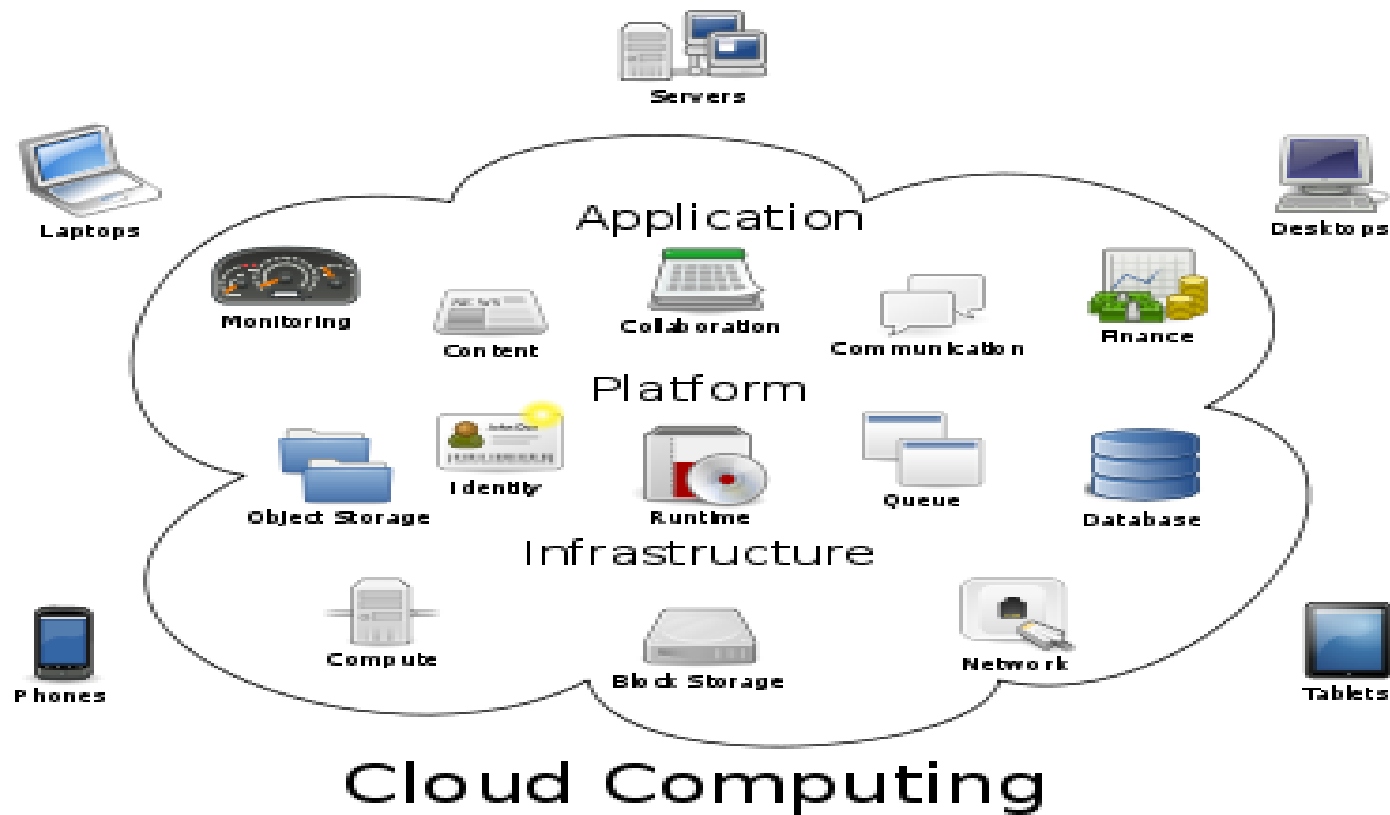
Web App

- ▶ Ce n'est plus que des liens hypertextes
- ▶ Plusieurs technologies
 - ▶ HTML5, CSS3, JQuery, Bootstrap, Modernizr, Java, XML, etc.
- ▶ Web sémantique (Web 3.0)
 - ▶ Applications d'entreprise et consommateurs plus complexes
 - ▶ BD sémantiques
 - ▶ Représentation de données flexible
 - ▶ Accès aux Interfaces via API
- ▶ IHM plus importante pour déterminer la qualité du produit

Applications mobiles

- ▶ Téléphone, Tablette, Borne, Voiture, etc.
- ▶ Prise en charge des composants de l'appareil
 - ▶ Accéléromètre, APN, Boussole, géolocalisation, etc,
- ▶ Accès à des données distantes et locales
- ▶ Persistance
- ▶ 2 choix
 - ▶ Web
 - ▶ Natif
- ▶ One Web / One Application ?

Cloud Computing



[Pressman, 2014]

Cloud Computing

- ▶ Ressources de stockage de données distribuées
- ▶ Dispositifs de traitements informatiques en réseau
- ▶ Calcul effectué à l'extérieur du Cloud avec des données dans le Cloud
- ▶ Architecture contenant deux niveaux :
 - ▶ Aval : les machines clientes et le logiciel d'application pour permettre l'accès
 - ▶ Amont : les serveurs, le stockage de données et les applications serveur
- ▶ Restrictions des accès

Ligne de production

- ▶ Ensemble de systèmes qui partagent un ensemble commun de caractéristiques et répondant aux besoins d'un marché particulier pour la production d'un logiciel
- ▶ Utilisation d'un noyau commun et de composants réutilisables
- ▶ Partage un ensemble d'actifs
 - ▶ exigences, l'architecture, design patterns, des composants réutilisables, les test cases, et d'autres livrable
- ▶ Permet d'élaborer de nombreux produits qui sont conçus en capitalisant sur les points communs entre tous ces produits,

Historique et évolution

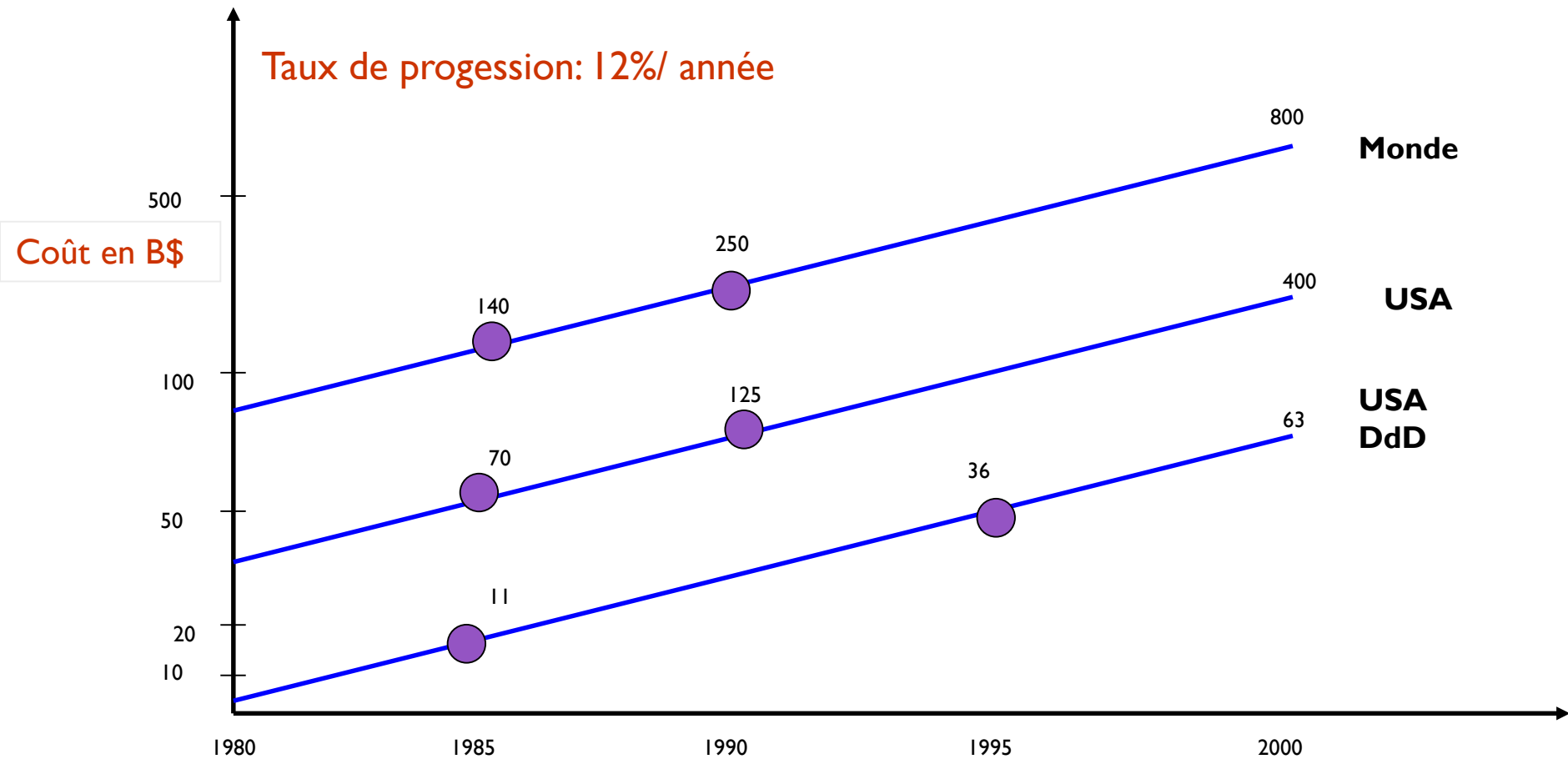
- Langages et outils logiciels:
 - 1960-1972: assembleur
 - 1972-1985: Fortran, Cobol, C, Pascal
 - 1985-1995: OO, Smalltalk, C++, interfaces usagers
 - 1995-...: outils Case: Rational Rose, Norme de conception, Java, ...
 - 2002: C#, Ruby
 - 2011-...:HTML5, PhoneGap, Sencha, XAMARIN,ASP.NET, Kotlin, Swift...

Crise du GL

Problème du génie logiciel

1. Accroissement de la demande et des coûts de développement
2. Taille et complexité des logiciels
3. Taille croissante des équipes
4. Spécifications peu précises
5. Évolution rapide des applications

Accroissement de la demande et des coûts de développement



Accroissement de la demande et des coûts de développement

- Standish group: rapport CHAOS: *a Recipe for Success*, ([1994](#), [2004](#)) [Retraillé, 2005]
 - 29%: taux de réussite de projets en 2004
 - 53%: taux des projets aboutis mais sans respecter le planning, le budget ou le périmètre fonctionnel prévus
 - 18% sont arrêtés

Taille et complexité des logiciels

- ▶ Le logiciel offre de plus en plus de fonctionnalités (système d'information, data warehouse)
- ▶ Logiciel souvent non unique, entités à interfacer
- ▶ Technologie en mutation (OS et langages en évolution)
- ▶ Complexité architecturale (machines distantes, hétérogènes, client-serveur, Intranet, ...)
- ▶ **Solution** : décomposer le processus de développement, découper en sous-systèmes, se rapprocher d'un découpage naturel proche de la réalité.

Taille croissante des équipes

- ▶ Gestion des compétences variées
- ▶ Coordination des travaux et circulation de l'information
- ▶ Gestion en parallèle du travail sur une même tâche
- ▶ Utilisation d'un langage non ambigu et compréhensible par tous les acteurs du problème
- ▶ Problème relié aux délais de plus en plus courts.

- ▶ **Solution** : Unification du vocabulaire et méthode d'organisation du travail

Spécifications peu précises

- ▶ Spécifications précises, cohérentes et complètes
- ▶ Représentent une vue fonctionnelle du système à réaliser : rôle important du client et donc proche du domaine d'application vs. Analyse et conception
- ▶ Nécessité d'un formalisme simple et concis compris par tous les partis.
- ▶ **Solution** : Modèles pour éclaircir, récapituler et montrer les points clés des spécifications.

Évolution rapide des applications

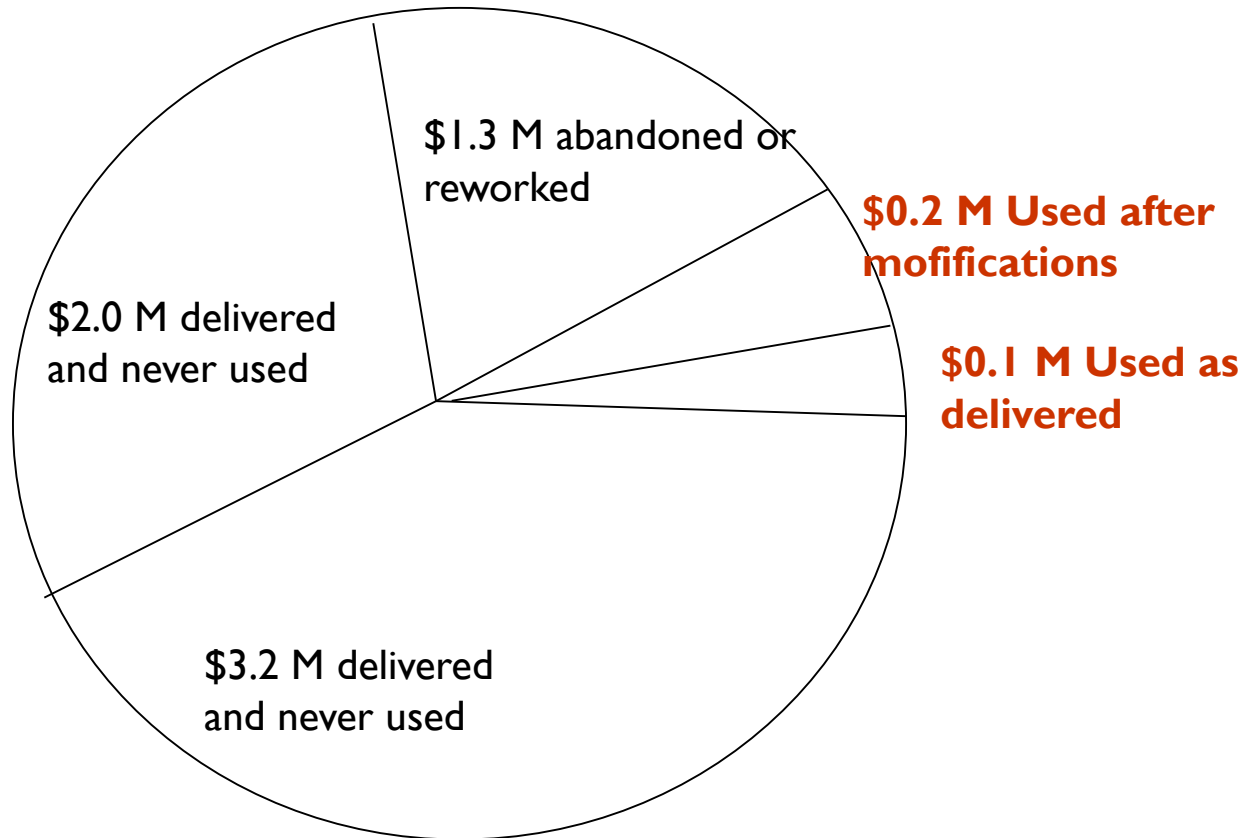
- ▶ Évolution fonctionnelle et technique
 - ▶ Modification des besoins du client
 - ▶ Modification de l'activité du client
 - ▶ Modification de l'environnement technique.
-
- ▶ **Solution** : cycle de vie itératif et incrémental, comme le cycle de vie objet.

Crise du logiciel

- Les cause de l'échec sont:
 - Mauvaise division et répartition du travail dans une équipe de personnes
 - Mauvaise communication entre les personnes ou entre les équipes-transfert des connaissances
 - Taille du logiciel par rapport à l'effort estimé
 - Relation entre client et concepteur est d'abord un problème de langage (transformation de l'informel vers le formel, mauvaise compréhension des besoins)
 - Logiciels livrés mais jamais utilisés avec succès parce qu'ils ne respectent pas la qualité requise
 - ...

Évidence de la crise du logiciel

1980:



A breakdown of 6,8 M in software projects commissioned by the US government in 1979

Crise du logiciel: loi de l'évolution de logiciel

- **Loi du changement continuuel:** un logiciel doit être continuellement adapté/changé, sinon il devient moins satisfaisant à l'usage!
- **Loi de complexité croissante:** lorsqu'un logiciel change, sa structure tend à devenir plus complexe. Des ressources additionnelles doivent être consacrées à maintenir et à préserver sa structure
- **Loi de croissance constante:** un logiciel doit doter constamment de nouvelles fonctionnalités afin de maintenir la satisfaction des utilisateurs tout au long de sa vie
- **Loi de la qualité déclinante :** la qualité d'un logiciel tend à diminuer face aux changements!

Solution à cette crise: Une méthodologie de développement

1. **Améliorer la productivité :**

- faciliter le recensement des «vrais» besoins
- Réutilisation des composants
- Prototypage rapide

2. **Réduire la complexité**

- automatisation de la documentation et de la programmation
- développement incrémental

3. **Améliorer la qualité**

- Test de logiciel
- Gestion de projet

Caractéristique du logiciel

Caractéristiques (1)

- Pourquoi est-ce que les systèmes d'exploitation ne peuvent-ils être construits comme on construit les ponts ou les gratte-ciels ?
- Il n'y a pas la même nature que le matériel (hardware)
 - Un logiciel est développé et non fabriqué comme un ordinateur
- **Un logiciel ne s'use pas mais il se détériore**
 - *When a hardware component wears out, it is replaced by a spare parts. There are no software spare parts!*
 - Alors, **la maintenance de logiciel implique plus des complexités que la maintenance de matériel**
- La plupart des logiciels sont développés sur demande et ne sont pas assemblés à partir de composants existants

Caractéristiques (2)

- Dire un logiciel de qualité sous-entend qu'il satisfait les caractéristiques de **qualité externe et interne**
 - Externe: vision client
 - Interne: vision développeur
- Parmi les caractéristiques de qualité externe, on distingue:
 - Fonctionnalité
 - Fiabilité
 - Facilité d'utilisation
 - Rendement
 - Maintenabilité

Caractéristiques (3)

- **Fonctionnalité:** est la capacité d'un logiciel de fournir les fonctions qui répondent aux besoins indiqués durant l'utilisation du logiciel. Les sous-caractéristiques sont:
 - **Aptitude:** concerne la présence et la convenance d'un ensemble de fonctions pour accomplir des tâches précises
 - **Exactitude:** concerne l'exactitude ou la convenance des résultats ou d'effets
 - **Interopérabilité:** est la capacité du système d'agir avec d'autres système
 - **Sécurité:** est la capacité de prévenir les accès non autorisés aux programmes et aux données

Caractéristiques (4)

- **Fiabilité:** est la capacité d'un logiciel de maintenir son niveau de performance quand il est utilisé dans des conditions spécifiques. Les sous-caractéristiques sont:
 - **Maturité:** est la convenance de la fréquence d'échec
 - **Tolérance aux fautes:** est la capacité de maintenir un niveau de performance dans les cas d'erreurs
 - **Possibilité de récupération:** est la capacité de rétablir son niveau de performance et récupérer directement, en un temps et effort déterminés, les données affectées en cas d'échec

Caractéristiques (5)

- **Facilité d'utilisation**: est la capacité d'un logiciel à être compris, appris, utilisé et être aimé par l'utilisateur. Les sous-caractéristiques sont:
 - **Facilité de compréhension**: concerne les efforts que les utilisateurs fournissent pour connaître le logiciel et ses possibilités
 - **Facilité d'apprentissage**: concerne les efforts que les utilisateurs fournissent pour apprendre l'utilisation du logiciel (opérations de contrôle, entrées, sorties)
 - **Facilité d'exploitation**: concerne les efforts que les utilisateurs fournissent pour exploiter les services offerts par le logiciel

Caractéristiques (6)

- **Rendement:** est la capacité d'un logiciel de fournir la performance requise, relative aux ressources utilisées. Les sous-caractéristiques sont:
 - **Comportement vis-à-vis du temps:** est la convenance du champs de réponse et de traitement
 - **Comportement vis-à-vis des ressources:** est la convenance de la quantité de ressources utilisées et la durée d'une telle utilisation

Caractéristiques (7)

- **Maintenabilité:** est la capacité d'un logiciel d'être modifié. La modification peut inclure les corrections, l'amélioration ou l'adaptation du logiciel aux changements de l'environnement. Les sous-caractéristiques sont:
 - **Facilité d'analyse:** concerne l'effort fournis pour diagnostiquer les déficiences ou les causes d'une panne ou pour identifier des parties à modifier
 - **Facilité de modification:** concerne les efforts fournis pour modifier ou éliminer les erreurs ou pour des changements environnementaux
 - **Stabilité:** concerne les risques d'effets inattendus dus aux changements
 - **Facilité de test:** concerne les efforts fournis pour valider le logiciel modifié

Caractéristiques (8)

- Parmi les caractéristiques de la **qualité interne**, on distingue:
 - **Réutilisabilité**: de plus en plus importante aujourd'hui
 - **Efficacité**: bonne utilisation des ressources matérielles
 - **Portabilité**: adaptation à des nouveaux environnements

Domaines d'applications

- Les principaux domaines d'application
 - Logiciels système
 - Logiciels scientifiques
 - Logiciels embarqués
 - Logiciels d'intelligence artificielle
 - Logiciels d'application Web
 - Jeux vidéo
 - ...