

LOG100/GTI100

Programmation en génie logiciel

Automne 2024

Introduction au Génie Logiciel et au Cycle de Développement Logiciel

Chargé de cours : Anes Abdennebi

Les diapositives appartiennent à l'origine au
Valentin Vervondel, MRes
et Sègla Kpodjedo, PhD
Avec modifications

Bienvenue!



2

- Présentation du cours
- Objectifs : comprendre et appliquer les concepts de la **POO** (Programmation Orientée Objet)
 - ▣ Classe, objet, interface, heritage, encapsulation
 - ▣ Algorithmes de base, structures de données
- Site **Moodle**
 - ▣ Cours disponibles après la séance
 - ▣ Forum pour toute question

Bienvenue!

3

□ Évaluation

- ▣ Quiz (4^{ème} cours) : 5 % 
- ▣ Laboratoires : 20 %
 - Pas de retard possible (voir Règlement des études)
- ▣ Examen : 25 % 

Attention : une moyenne pondérée **inférieure à 50 %** dans les examens et les quiz entraîne automatiquement un **échec au cours**, peu importe les notes des laboratoires.

- Socrative : mini-quiz en classe (non évalués)
 - ▣ Sur socrative.com avec le numéro de la salle
- Moodle : **sondage sur vos attentes**
 - ▣ Possibilité d'adapter les horaires
 - ▣ Date du premier laboratoire sera affichée sur Moodle
- Toute absence ou retard à une évaluation doit être justifiée (maladie avec justificatif, etc.)
- Attention au plagiat! Il n'est pas toléré.
 - ▣ Lisez bien: Règlement relatif aux infractions de nature académique
 - ▣ Examens finaux: Directive relative aux examens finaux

Bienvenue!

5

Pour toutes vos questions et discussions académiques
dans ou en dehors du thème du GL

Email: - cc-anes.abdennebi@etsmtl.ca

Bienvenue!

6

- Nouveau système en test pour les laboratoires
 - ▣ IDE en ligne, depuis n'importe quelle machine
 - ▣ Exercices intégrés dans des guides
 - ▣ Tests d'évaluation exécutables plusieurs fois
 - ▣ Sauvegarde et remise automatique
 - ▣ Résultats plus rapides

Créez votre compte avec votre adresse
`prenom.nom.X@ens.etsmtl.ca`



Bienvenue!

7

Moodle: <https://ena.etsmtl.ca/> (Quiz et Intra)

ENAAQuiz: <https://enaquiz.etsmtl.ca/> (Examen final)

Plan

8

- Qu'est ce qu'un logiciel ?
- Génie logiciel : quel besoin ?
- Qu'est ce que le génie logiciel ?
- Ingénieur vs. programmeur
- Génie logiciel vs. d'autres disciplines
- Cycle de vie du logiciel

Un logiciel

9

« Ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de données. »



— *Dictionnaire Larousse*

Un logiciel

10

« Instructions that tell a computer what to do.

Software comprises the entire set of programs, procedures, and routines associated with the operation of a computer system»

Traduction: « Instructions qui indiquent à un ordinateur quoi faire. Le logiciel comprend l'ensemble des programmes, procédures et routines associés au fonctionnement d'un système informatique»

— *Britannica*

Un logiciel

11

- La nature d'un produit logiciel
 - ▣ Intangible
 - ▣ Ne s'use pas, il n'est pas détérioré par les tests
 - ▣ Produit artisanal
 - ▣ (trop ?) facile à modifier
 - ▣ Coûte (trop ?) cher



Qu'est ce que le génie logiciel ?

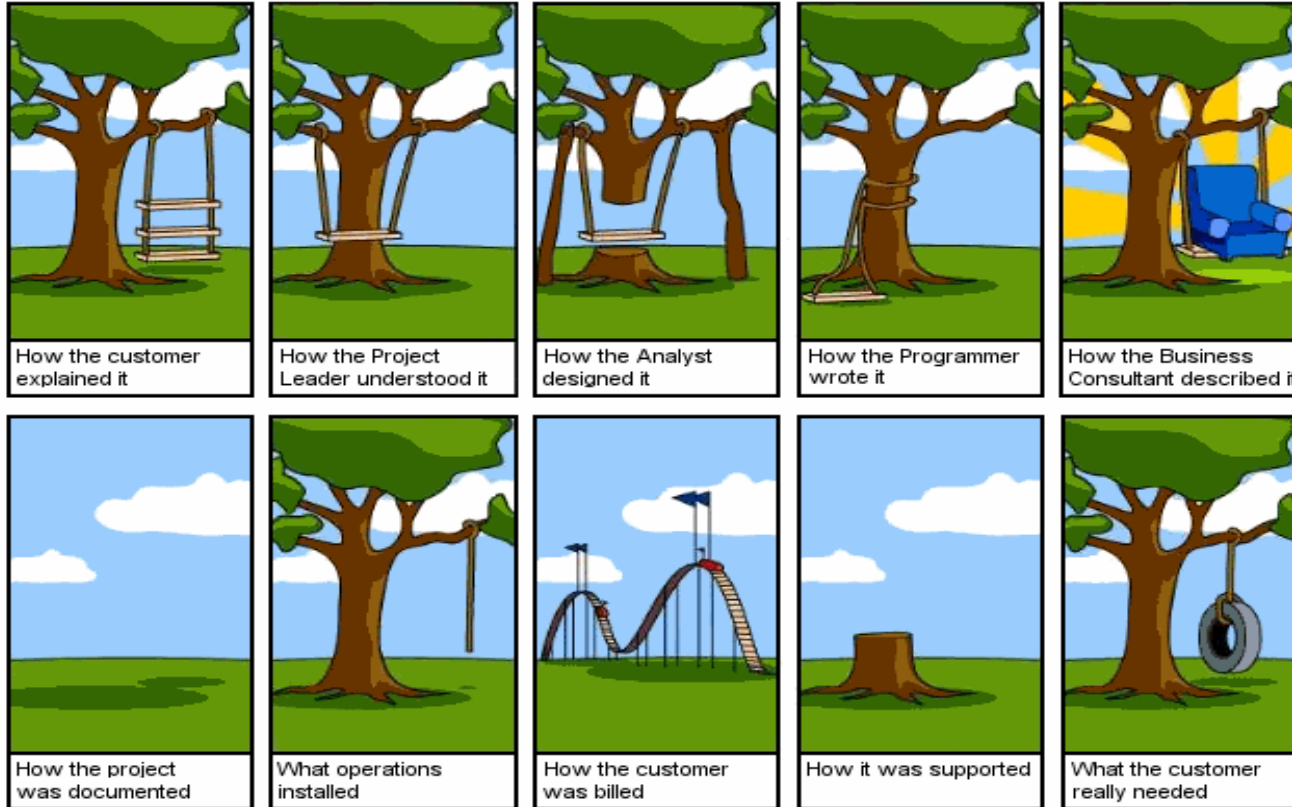
12

À votre avis ?

Pourquoi a-t-on besoin d'une discipline
telle que le génie logiciel ?

Génie logiciel : quel besoin ?

13



Source : projectcartoon.com

Génie logiciel : quel besoin ?

14

Client requirement....

Client : Why don't you clearly understand my requirements ?

His Requirement :



Source : programmerhumour.tumblr.com

Génie logiciel : quel besoin ?

15

- Un problème de gestion et de planification
 - ▣ Plusieurs intervenants
 - ▣ Un budget limité
 - ▣ Des échéanciers à respecter
- Un problème de communication
 - ▣ Les intervenants ont différentes perspectives sur le produit final



Génie logiciel : quel besoin ?

16

HealthCare.gov

Learn

Get Insurance

Log in

Español

Individuals & Families

Small Businesses

All Topics ▾

Search

SEARCH

The System is down at the moment.

We're working to resolve the issue as soon as possible. Please try again later.

- HealthCare.gov* (États Unis-2013)
 - ▣ Ce site Web sera l'échange officiel de soins de santé qui permettra aux résidents de comparer les prix des plans de soins de santé, d'identifier s'ils sont admissibles aux subventions fédérales et de s'inscrire au plan choisi.
 - ▣ Budget initial: 93.7m \$US → Budget final: 1.7b \$US

** Harvard Business School - The Failed Launch Of www.HealthCare.gov*

Génie logiciel : quel besoin ?

17

HealthCare.gov

Learn

Get Insurance

Log in

Español

Individuals & Families

Small Businesses

All Topics ▾

Search

SEARCH

The System is down at the moment.

We're working to resolve the issue as soon as possible. Please try again later.

- Une des multiples causes de cet échec:
 - ▣ « Les postes techniques n'étaient pas pourvus et les chefs de projet **avaient peu de connaissances** sur la quantité de travail requise et les processus typiques de développement de produits, ce qui leur laissait très **peu de temps** pour **tester et dépanner** le site Web »

* *Harvard Business School - The Failed Launch Of www.HealthCare.gov*

Génie logiciel : quel besoin ?

18

Problèmes rencontrés avec un logiciel :

En connaissez-vous ?

Échecs logiciels

20



- 2009, Royaume-Uni : un distributeur automatique donne **deux fois** le retrait demandé après une "erreur opérationnelle", un *computer glitch*
- *"It makes a change for us to have the joke on the banks for once."*
- *La banque : "If the people using the ATM see it as a bit of fun, so be it."*
- Événements similaires en 2008 / 2011 en Europe / Amérique du Nord

Échecs logiciels

21



Ariane 5, Vol 501 (1996)

← 4000 m et 40 s →

- Pertes : 370 millions \$ US
- Cause (entre autres) :
mauvaise gestion d'exceptions
PRE: $-32768 \leq x \leq +32767$
POST: $y = \text{int}16(x)$



Mars Climate Orbiter (1999)

- Écrasement du vaisseau sur Mars
- Pertes : 300 millions \$US
- Cause (entre autres) :
Mélange entre les systèmes impérial
et métrique

Machine de radio-thérapie Therac-25 (1985-87)

- Sur-Radiations (jusqu'à 100 fois la dose)
- Au moins 5 morts
- Cause : aucun test sur les entrées non-standard

Le bug de la division du Pentium (1994)

Une perte de quelques 500 millions de dollars par Intel

Échecs logiciels

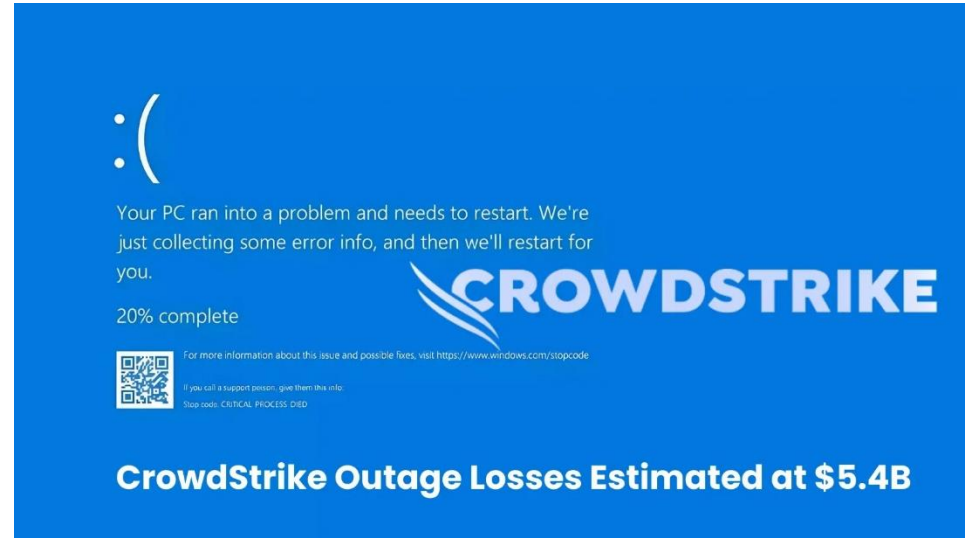
22

La panne de CrowdStrike-Microsoft (Juillet 2024)

“La panne a été déclenchée par une mise à jour logicielle défectueuse sur la plateforme Falcon de CrowdStrike, entraînant des pannes système et des erreurs d'écran bleu sur les appareils Windows” [Cyber Security News, 2024].

Pertes (globales):

- Santé: 1,94 milliards \$ (estimée)
- Les banques: 1,15 milliards \$
- Compagnies Aériennes: 860 millions \$



Annulations de projets

24

- Exemples d'annulation de projets de grande envergure :
 - ▣ LES (*London Stock Exchange*) a abandonné en 1993 son projet TAURUS qui devait assurer le suivi de l'exécution des transactions
 - 10 ans de développement
 - Coût de développement: £60 millions
 - Plus de £400 millions dépensés par les opérateurs sur le marché pour adapter leurs logiciels à TAURUS



London
Stock Exchange

TAURUS Creating a Paperless
Stock Trading Operation

Annulations de projets

25



- En Californie, le DMV (*Department of Motor Vehicles*) a renoncé en 1993 au projet informatique commencé en 1987 qui devait intégrer deux systèmes d'informations (immatriculation et permis)
 - Ce système avait coûté 45 millions \$US
 - Les coûts étaient de 6.5 fois l'estimation initiale

Annulations de projets

26

- En 1997, l'état de Washington a annulé son projet LAMP (*License Application Mitigation Project*)
 - Début 1990, livraison prévue pour 1995
 - Budget initial : 16 millions \$US
 - Coût en 1992 : 41.8 millions \$US
 - Coût en 1993 : 51 millions \$US
 - Coût estimé en 1997 à 67.5 millions \$US dont 40 millions ont été dépensé avant que le projet ne soit arrêté
 - On a estimé que le système coûterait \$US 4.2 millions par année d'entretien



Génie logiciel : quel besoin ?

27

- Problèmes rencontrés avec un logiciel
 - ▣ Non-conformité aux exigences de fonction
 - Peut résulter en un comportement non désiré et imprévu (potentiellement dangereux)
 - ▣ Non-conformité aux exigences de qualité
 - Temps de réponse
 - Non-disponibilité
 - Difficulté d'utilisation



28

- Ingénierie appliquée à la production des logiciels
 - ▣ Méthodes, principes, techniques, outils, langages... visant à concevoir, implémenter et maintenir des logiciels qui satisfont :
 - Les exigences fonctionnelles
 - Les exigences de qualité
 - Les contraintes budgétaires et temporelles

□ Définition selon le standard IEEE 610.12

- *(1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.*
- *(2) The study of approaches as in (1).*



Traduction:

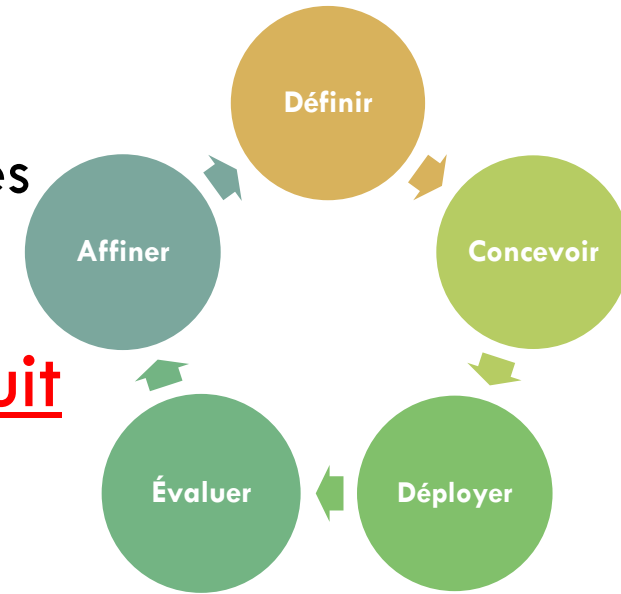
- (1) L'application d'une *méthode systématique, disciplinée et quantifiable* au développement, à la mise en œuvre et à la maintenance du logiciel.
- (2) L'étude des approches telles que dans (1).

*« Méthodes pour la construction multi-personnes de programmes multi-versions. » —
D. Parnas, 1974*

- ▣ Travail collaboratif
- ▣ Mise en œuvre d'un produit complexe
- ▣ Plusieurs versions du produit
 - Programmes
 - Documentation



- Organisation du processus de production
 - ▣ Cycles de développement
 - ▣ Gestion de projets
 - Ressources humaines et matérielles
 - Selon le temps et le budget
- Contrôle de la qualité du produit
 - ▣ Valider et vérifier
 - ▣ Corriger et améliorer



Ingénieur vs. programmeur

33

- « Un ingénieur est un professionnel exerçant des activités de conception et de direction de projets, de réalisation et de mise en œuvre de produits, de systèmes ou de services impliquant des **problèmes techniques complexes**. Ces responsabilités supposent alors un ensemble de connaissances techniques d'une part, économiques, sociales, environnementales et humaines d'autre part, reposant sur une solide culture scientifique. » [Wikipédia]

- Vous n'allez pas juste programmer, mais vous allez :
 - Analyser, Concevoir, Programmer, Tester, Documenter, Communiquer, Évaluer
 - Gérer, Diriger, Planifier, Formuler, etc.



Génie logiciel vs. d'autres disciplines

34

Le génie logiciel : une discipline relativement jeune



Le pont du Gard construit au 1^{er} siècle par les romains

Génie logiciel vs. d'autres disciplines

35

- La nature du produit logiciel
 - ▣ On peut « facilement » le modifier
 - ▣ On peut corriger ces défauts même après livraison et mise en œuvre
 - ▣ On le modifie de façon continue durant sa vie (maintenance)
- Imaginez cela dans le contexte de la construction !...

Génie logiciel vs. d'autres disciplines

36

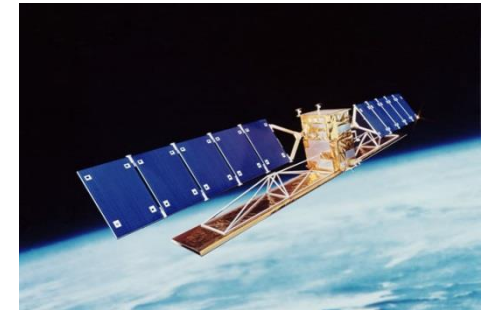


- C'est un domaine qui évolue rapidement
 - ▣ Les langages de programmation
 - ▣ Les technologies réseaux
 - ▣ Les plateformes matérielles
 - ▣ Les méthodes
- Les systèmes logiciels sont de plus en plus complexes et hétérogènes
- Le développement rapide de l'IA

Génie logiciel vs. d'autres disciplines

38

- Les systèmes temps réels sont produits de façon plus rigoureuse
 - ▣ Doivent réaliser les exigences attendues de façon **exacte** mais aussi ***dans des délais précis***
 - ▣ Ils sont présents dans les avions, les centrales nucléaires, les systèmes de contrôle de procédés, le domaine médical, les satellites, etc.
 - ▣ Ils nécessitent de gros budgets



Cycle de vie vs. de développement

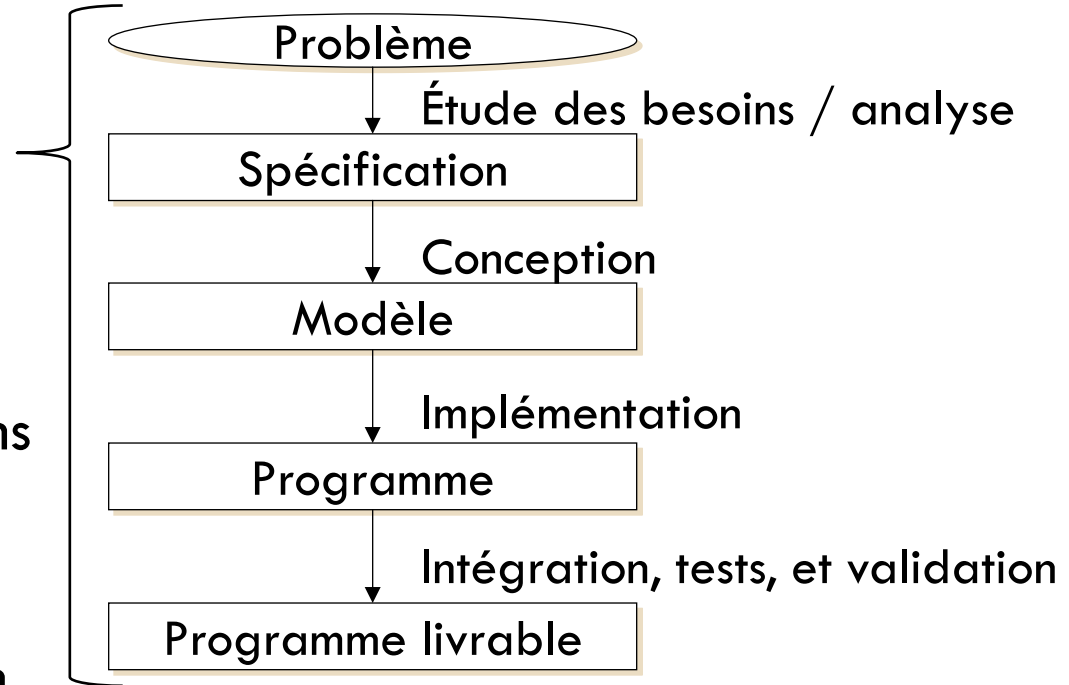
39

- Cycle de vie
 - ▣ Ensemble des phases par lesquelles passe un logiciel durant sa durée de vie
 - {conception, mise en œuvre, maintenance et mise hors-service}
- Cycle (ou processus) de développement
 - ▣ Ensemble des étapes menant à la mise en œuvre d'un logiciel
 - ▣ Il faut utiliser un processus bien défini
 - Des étapes et des livrables identifiés
- On va souvent utiliser les deux termes de façon égale

Processus de développement logiciel

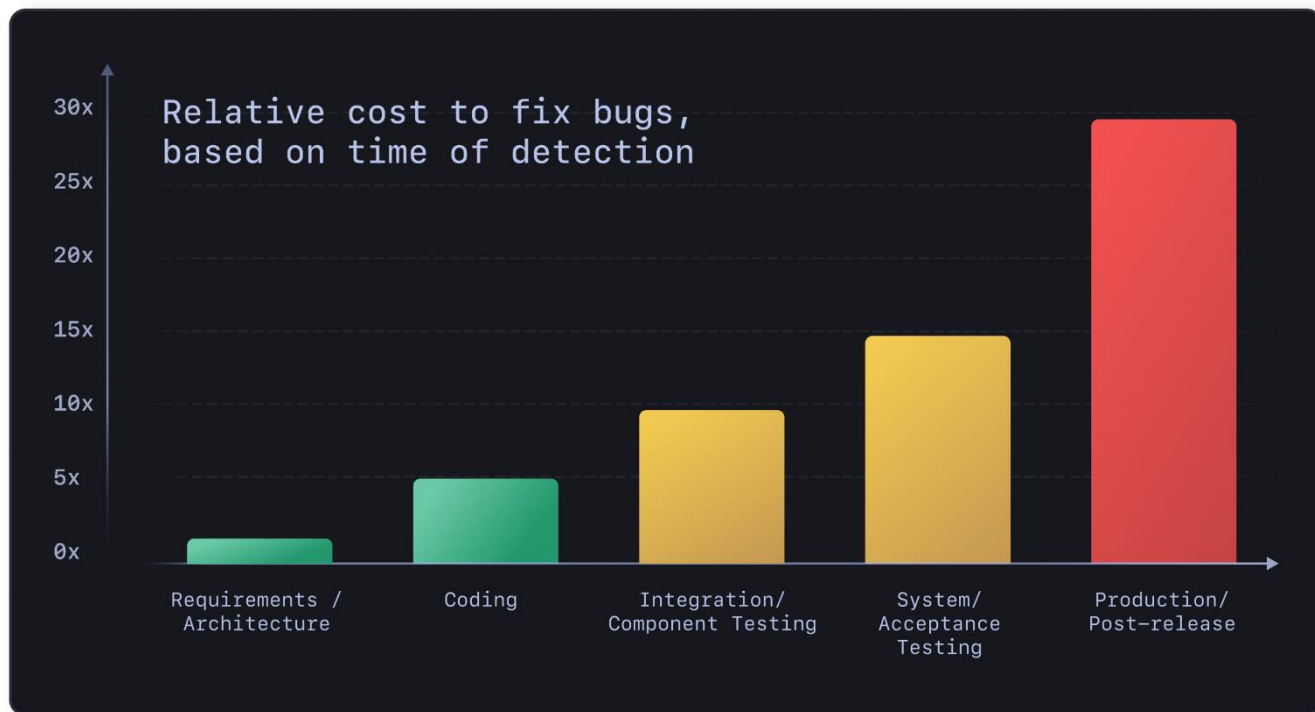
40

- ❑ Cycle traditionnel de développement (en cascades)
 - ❑ Nécessité de connaître parfaitement les besoins
 - ❑ Résultats vérifiés tard dans le cycle
 - ❑ Une faute détectée tard peut avoir un impact et un coût élevé



Processus de développement logiciel

41



Coût relatif pour corriger les bugs, basé sur le moment de la détection

D'après « The exponential cost of fixing bugs » de deepsource.com

Processus de développement logiciel

42

□ Cycle de développement itératif et incrémental

□ Itération

- Mini-projet à résultat testé et exécutable
- Feedback rapide et minimisation des risques

□ Incréments

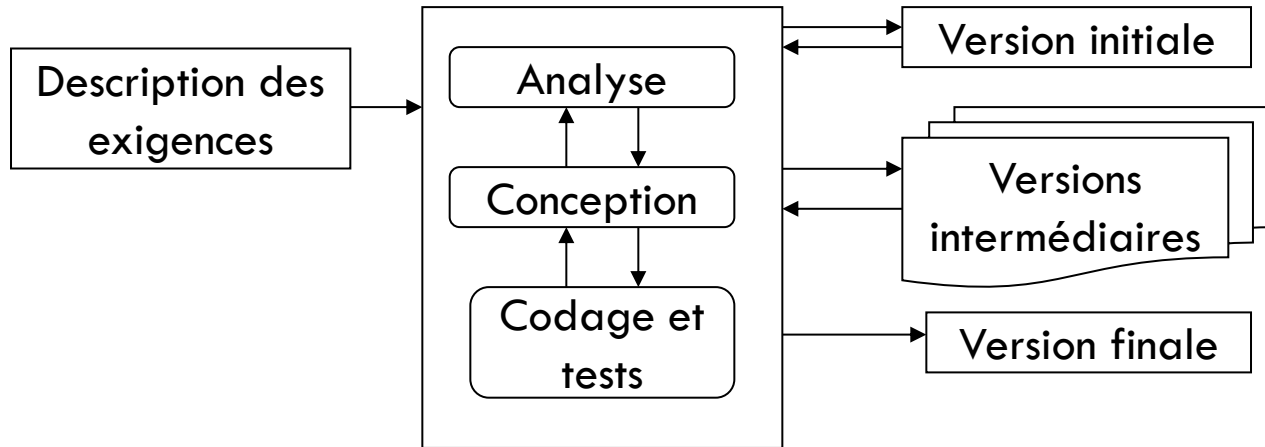
- Le système croît après chaque itération
- Convergence vers le produit final



Processus de développement logiciel

43

□ Cycle de développement itératif et incrémental



Modèle adapté de (Sommerville I., Software Engineering, Addison Wesley, 2001)

Conclusion

44

□ Les qualités des produits logiciels

- Exactitude
- Fiabilité
- Robustesse
- Performance
- Convivialité
- Facilité de vérification
- Facilité de maintenance
- Facilité de réparation
- Facilité d'évolution
- Facilité de réutilisation
- Portabilité
- Facilité de compréhension
- Interopérabilité
- ...

45

- [illegible]

```

import java.io.*;
import java.net.*;
import java.util.*;

public class Client {
    public void sendAuthent(String host, int port) throws IOException {
        OutputStream outStream = new Socket(host, port).getOutputStream();
        long t1 = (new Date()).getTime();
        double q1 = Math.random();
        byte[] protected1 = Protection.encode(q1);
        long t2 = (new Date()).getTime();
        double q2 = Math.random();
        byte[] protected2 = Protection.encode(q2);
        out.writeUTF(user);
        out.writeInt(protected1.length);
        out.write(protected1);
        out.writeInt(protected2.length);
        out.write(protected2);
        out.flush();
    }
}

public static void main(String[] args) {
    String host = args[0];
    int port = 7999;
    String user = "John";
    String password = "shh";
    Socket s = new Socket(host, port);
    Client client = new Client();
    client.sendAuthent(host, port);
}

```



Java™ time!

Question 1:

Quelle méthodologie de développement logiciel est la meilleure à utiliser ?

- A) En Cascades
- B) Agile
- C) Les deux
- D) Les deux sont obsolètes
- E) Je ne sais pas!

Question 2:

```
int valeurA = 256;
int valeurB = 128;
valeurB = (valeurA > valeurB) ? valeurB *=2 : valeurB;
if (valeurB > valeurA) {
    System.out.println("Bingo");
} else {
    System.out.println("C'est bizarre!!");
}
```

Quel message s'affichera?

- A) Erreur
- B) 128
- C) 256
- D) Bingo
- E) C'est bizarre!!