

# **Chapitre I : Introduction au Génie logiciel**

**Rania MZID**

**Institut Supérieur d'Informatique (ISI Ariana)**

# Qu'est ce un logiciel?

---

- ▶ Ensemble des **programmes, procédés et règles**, et éventuellement de la **documentation**, relatifs au fonctionnement d'un ensemble de traitement de l'information
- ▶ Autrement dit, un logiciel est l'ensemble de :
  - ▶ Programmes informatiques (code)
  - ▶ Documentations (manuel d'utilisation, de maintenance, etc..)
- ▶ Réalise une spécification
  - ▶ Son comportement vérifie un ensemble de critères qui régissent ses interactions avec son environnement



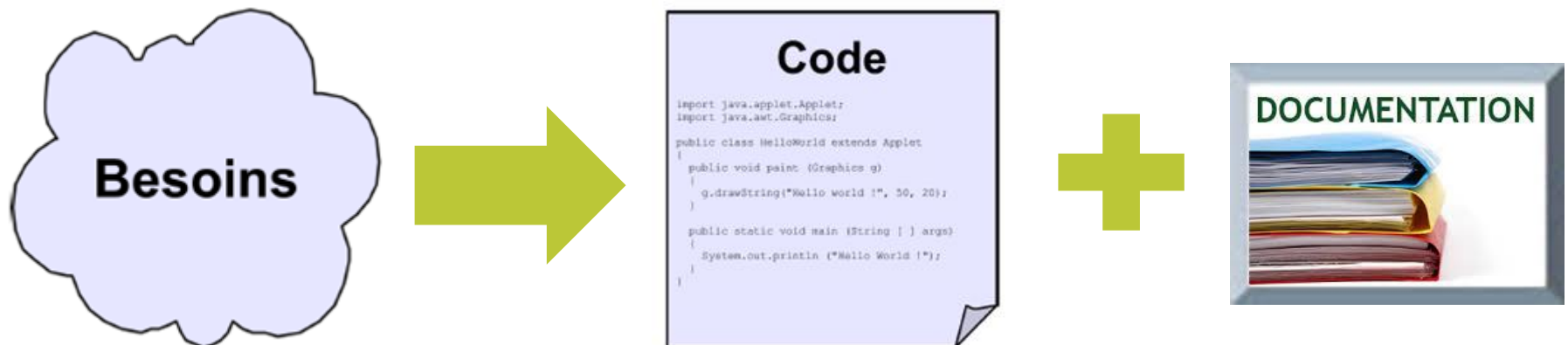
# Qu'est-ce que le génie logiciel ?

---

- ▶ L'art de **spécifier**, de **concevoir**, de **réaliser**, et de **faire évoluer**, avec des moyens et dans des délais raisonnables, des *programmes*, des *documentations* et des *procédures de qualité* en vue d'utiliser un ordinateur pour résoudre certains problèmes. [

**Gaudel et al, 1996**

**Précis de génie logiciel, Edition Masson]**



# Objectifs du génie logiciel

---

- ▶ Le génie logiciel vise à assurer que
  - La spécification répond aux besoins clients
  - Le logiciel respecte sa spécification
  - Les coûts et les délais alloués pour sa réalisation sont respectés
  
- ▶ Qu'est ce qu'une spécification ?
  - Traduit les besoins clients
  - Besoin simple, exact : facile à spécifier
  - Besoin complexe, ambiguë : difficile à spécifier



# Quelques exemples de spécifications complexes (ambigües)

---

- ▶ Un traducteur automatique : est-ce qu'un texte anglais « bien écrit » ?
- ▶ Un jeu vidéo : Comment spécifier ce qui est amusant ?



# Qualité du logiciel

---

## ▶ Enjeu du génie logiciel

« le bon logiciel bien fait » !

## ▶ Quelques exigences de qualité logiciel (Bertrand Meyer)

### ■ **Correction**

→ « La correction est la capacité que possède un produit logiciel de mener à bien sa tâche, telle qu'elle a été définie par sa spécification »

### ■ **Robustesse**

→ « La robustesse est la capacité qu'offrent des systèmes logiciels à réagir de manière appropriée à la présence de conditions anormales. »

### ■ **Extensibilité**

→ « L'extensibilité est la facilité d'adaptation des produits logiciels aux changements de spécifications. »

---



# Qualité du logiciel

---

▶ Quelques exigences de qualité logiciel (Bertrand Meyer)

■ **Réutilisabilité**

→ « La réutilisabilité est la capacité des éléments logiciels à servir à la construction de plusieurs applications différentes. »

■ **Efficacité**

→ « L'efficacité est la capacité d'un système logiciel à utiliser le minimum de ressources matérielles, que ce soit le temps machine, l'espace occupé en mémoire, ou la bande passante des moyens de communication. »

■ **Portabilité**

→ « La portabilité est la facilité avec laquelle des produits logiciels peuvent être transférés d'un environnement logiciel ou matériel à l'autre »

---



# Qualité du logiciel

---

- ▶ Un bon logiciel de point de vue **client**





# Qualité du logiciel

---

- ▶ Un bon logiciel de point de vue **fournisseur**



# Eléments du génie logiciel

---

## Activités

- Expression des besoins, Spécification et Analyse, conception, Programmation
- Vérification et Validation, Maintenance

## Langages

- **Formalisent les activités**
- UML, Merise, Ecore, SDL, etc.

## Processus

- **Organisent l'exécution les activités**
  - Agile, RUP, XP, cycle en V, etc.
- 



# Activités du génie logiciel

---

- ▶ Spécification et expression des besoins
- ▶ Analyse des besoins
- ▶ Conception
- ▶ Programmation
- ▶ Validation et vérification
- ▶ Maintenance

=> Pour chaque activité : Utilisation et production de documents



# Activité 1: Spécification & expression des besoins

---

- ▶ **Comprendre les besoins du client**
  - ▶ Objectifs généraux, environnement du futur système, ressources disponibles, contraintes de performance
  - ▶ Fournie par le client (expert du domaine d'application, futur utilisateur...)
  
- ▶ **Les besoins exprimés par l'utilisateur sont**
  - ▶ Définis dans le cahier des charges rédigé en langage naturel
  - ▶ Formalisé en utilisant un langage (Diagramme de cas d'utilisation UML)



# Activité 2: Analyse des besoins

---

- ▶ Établir une description claire de ce **que doit faire** le logiciel :
  - ▶ fonctionnalités détaillées
  - ▶ exigences de qualité
  - ▶ Elaboration des maquette des interfaces
- ▶ **Etude de faisabilité**
  - ▶ Est-il techniquement et financièrement faisable de construire le système ?
- Clarifier le cahier des charges (ambiguïtés, contradictions)
  - ▶ Lister les exigences fonctionnelles et non fonctionnelles en détail
  - ▶ Validation des exigences



# Activité 3: Conception

---

- ▶ Élaborer une **solution concrète (comment ?)** réalisant la spécification
- ▶ **Conception de l'architecture**
  - ▶ Identification de la structure globale du système
  - ▶ Les principaux composants
  - ▶ Leurs relations
- ▶ **Conception des interfaces**
  - ▶ On définit les interfaces du système
- ▶ **Conception des composants**
  - ▶ Conception de chaque composant de façon indépendante
  - ▶ Conception de la structure de la base de données



# Activité 4: Programmation

---

- ▶ **Implantation** de la solution conçue
  - ▶ Traduction de la conception en un code compilable



# Activité 5: Vérification & Validation

---

## ○ Vérification

- Avons-nous construit le produit bien ?
  - Vérifier une spécification par rapport aux **attentes techniques**
  - Détection des **Erreurs** par rapport aux définitions précises établies lors des phases antérieures de développement
- **Le logiciel doit être conforme à sa spécification**

## ○ Validation

- Avons-nous construit le bon produit?
  - Vérifier une spécification par rapport aux **attentes métiers**
  - Détection des **défauts** par rapport aux besoins que le produit doit satisfaire
- **Le logiciel doit faire ce que l'utilisateur a besoin**





# Activité 5: Vérification & Validation

---

## ○ Erreur/Défaut (bug)/Défaillance

- Erreur (méprise) : Action humaine produisant un résultat incorrect
- Défaut (Bug) : Manifestation d'une erreur dans un logiciel
- Défaillance : Modification d'un comportement requis d'un système
  - La défaillance est un événement, Un bug est l'état d'un logiciel causé par une erreur



# Activité 5: Vérification & Validation

---

## ○ Les types de V&V

### ○ Statique : Analyse, contrôle

- ▶ → Vérifier des propriétés avant exécution afin de découvrir des problèmes ou prouver sa correction (manuel ou automatique)

### ○ Dynamique : évaluer, expérimenter

- ▶ → Le système est exécuté sur des données de test et son exécution est observé



# Activité 5: Vérification & Validation

---

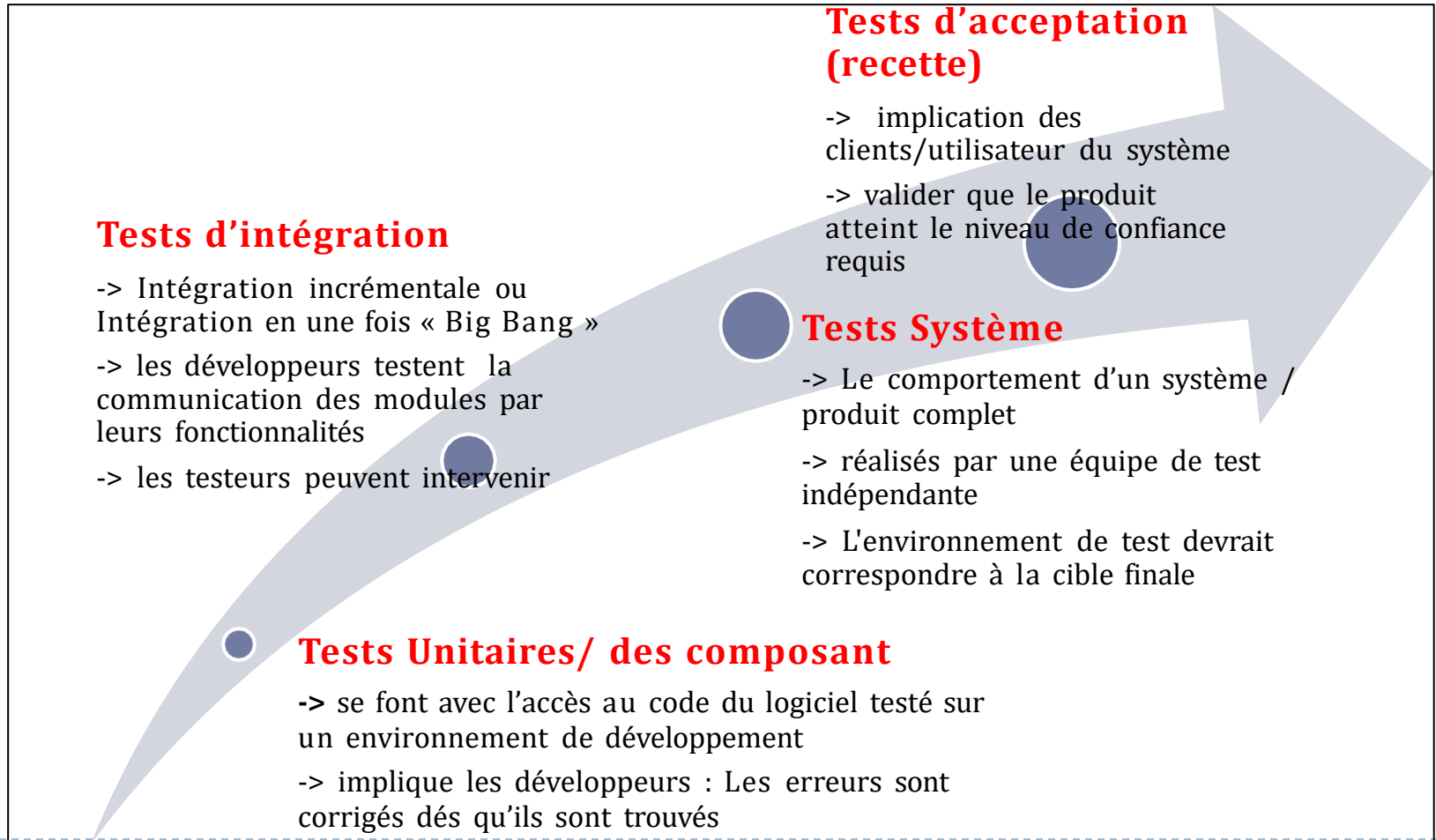
- ▶ Techniques de V&V: **les tests**
- ▶ « Le test est un processus **manuel** ou **automatique**, qui vise à établir qu'un système vérifie les **propriétés exigées** par sa **spécification**, ou à détecter des **différences** entre les **résultats** engendrés par le système et ceux qui sont **attendus** par la **spécification** » [IEEE 729, 1993]



# Activité 5: Vérification & Validation

---

## ► Les niveaux de tests



# Activité 5: Vérification & Validation

---

## ► Les types de tests

### **Test boîte blanche**

- tests basés sur une analyse de la structure interne du composant ou Système
- Tests structurels souvent **pour la vérification**

### **Test boîte noire**

- Tests sans référence aux structures internes du composant ou du système
- Tests fonctionnels ou non-fonctionnels souvent **pour la validation**



# Activité 6: La maintenance

---

## ▶ Types de maintenance :

- ▶ **Correction** : identifier et corriger des erreurs trouvées après la livraison
- ▶ **Adaptation** : adapter le logiciel aux changements dans l'environnement (format des données, environnement d'exécution...)
- ▶ **Perfection** : améliorer la performance, ajouter des fonctionnalités, améliorer la maintenabilité du logiciel



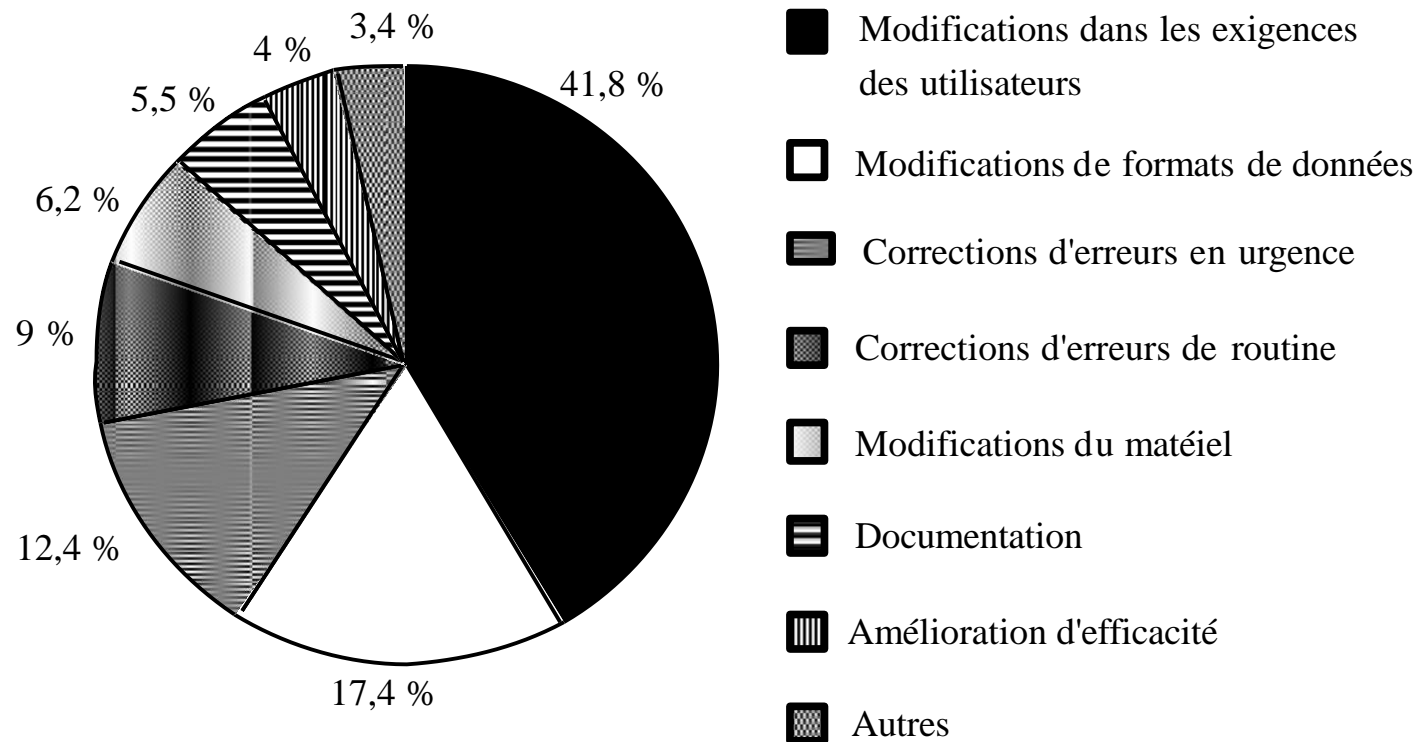
## Activité 6: La maintenance

---

- ▶ Plusieurs études ont été consacrées à l'évaluation des coûts de production des logiciels répartis sur tout le cycle de développement (i.e., analyse, conception, implémentation et maintenance).
  - ▶ Avec les méthodes classiques de développement, toutes ces études s'accordent pour attribuer à l'étape de maintenance la moyenne de 70% du coût global de développement.
- 



# Activité 6: La maintenance



*Répartition des coûts de maintenance ([Lientz 1979] in [Meyer 1988])*