CYCLE DE VIE D'UN LOGICIEL

Analyse et conception orientée objet

www.elbahihassan.com

elbahihassan@gmail.com

ISTA Oulad Teima

Software is more important than hardware.

— Bill Gates —

AZ QUOTES

Rappel

Information :

- L'information est un ensemble de données ayant un sens compréhensible par l'esprit humain. Elle peut donc prendre plusieurs formes : texte, image, son, vidéo, symbole …

Système d'informations :

- Le système d'informations est un ensemble organisé des ressource humains et matériels ayant pour but la collection, le stockage, le transport, le traitement, la diffusion, la présentation et la destruction de l'information au sein d'une organisation. Cela est réalisé en général grâce à un ordinateur.

Rappel

Informatique :

- L'informatique est la science du traitement automatique des informations

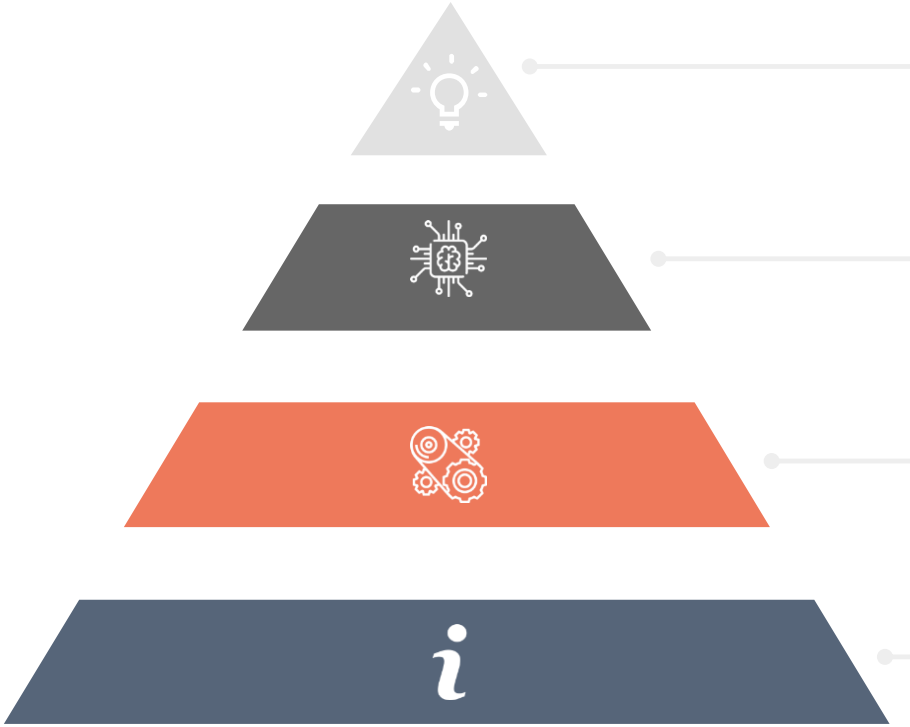
Système informatique :

- Le système informatique est l'ensemble des éléments matériels et logiciels destinés au traitement automatique de l'information. **Exemple** : Ordinateur.

Un logiciel :

- Un logiciel est un système d'information automatisé.

Rappel



Logiciels

Système informatiques

Système d'informations

Informations

Système informatique

Partie matérielle

Partie logicielle

Système informatique

Partie matérielle

Partie logicielle

Système informatique

Partie matérielle

Partie logicielle

Comparaison entre le logiciel et le matériel

- Le « Ordinateur » a besoin du « Software » pour être piloté
- Le « Ordinateur » a besoin du « Hardware » pour être exécuté
- Évolution des capacités des logiciels est intimement liée à l'évolution du hardware et aussi d'autre facteurs :
    - Amélioration de la puissance du processeur
    - Amélioration des capacité de stockages
    - Changement des dispositifs d'entrée ou de sortie (écran tactile, stylo optique, etc.)
    - Augmentation de la mobilités des unités mobiles (Smartphones, Tablettes, Smart Watch, etc.)

Impacts d'un logiciel de mauvaise qualité

Plusieurs dégâts ont été causés par des erreurs dans des logiciels :

- **Le bug du sonde Mariner-1** en **1962** : une fusée spatiale pour mission de survol de venus, a dérouté de sa trajectoire après 5 min de son lancement
  → **Cause** : Une formule mathématique qui a été mal transcrite en code source
  → **Coût** : 18,5 millions de dollars

Impacts d’un logiciel de mauvaise qualité

- **Processeur Pentium**, **1994**: Bug dans la table de valeurs utilisée par l'algorithme de division.
- Intel a décidé de remplacer tous les processeurs Pentium défectueux, ce qui aurait pu représenter un coût énorme pour la compagnie.

Impacts d’un logiciel de mauvaise qualité

La présence du bug (Processeur Pentium, 1994) peut être vérifiée manuellement en effectuant le calcul suivant dans toute application utilisant des nombres à virgule, y compris la calculatrice Windows ou Microsoft Excel dans Windows 95/98.

- La valeur correcte est:

$$\frac{4{,}195{,}835}{3{,}145{,}727} = 1.3338204491362410 02$$

- La valeur renvoyée par un processeur Pentium défectueux est :

$$\frac{4{,}195{,}835}{3{,}145{,}727} = 1.333739068902037589$$

- Toutefois, seule une petite fraction des possesseurs de processeurs défectueux a demandé l'échange.

Impacts d'un logiciel de mauvaise qualité

- Ariane V vol 501, 1996
  - → Explosion après 40 secondes de vol
  - → Coût : 370 millions de dollars
  - → Panne du système de navigation due à un dépassement de capacité (arithmetic overflow)

Impacts d'un logiciel de mauvaise qualité

- La société de tests de logiciels *Tricentis* a analysé 606 défaillances logicielles sur 314 entreprises afin de mieux comprendre l'impact commercial et financier des défaillances logicielles.
- Ces défaillances de logiciels avaient affecté 3,6 milliards de personnes et entraîné des pertes financières de 1 700 milliards de dollars.

SOFTWARE

Report: Software failure caused $1.7 trillion in financial losses in 2017

Software testing company Tricentis found that retail and consumer technology were the areas most affected, while software failures in public service and healthcare were down from the previous year.

By Scott Matteson | January 26, 2018, 7:54 AM PST

*Référence : https://www.techrepublic.com/article/report-software-failure-caused-1-7-trillion-in-financial-losses-in-2017/

Raisons principales des bugs

- Erreurs humaines
- Taille et complexité des logiciels
- Taille des équipes de conception/développement
- Manque de méthodes de conception
- Négligence de la phase d'analyse des besoins du client
- Négligence et manque de méthodes et d'outils des phases de validation/vérification

Critères de qualité d'un logiciel

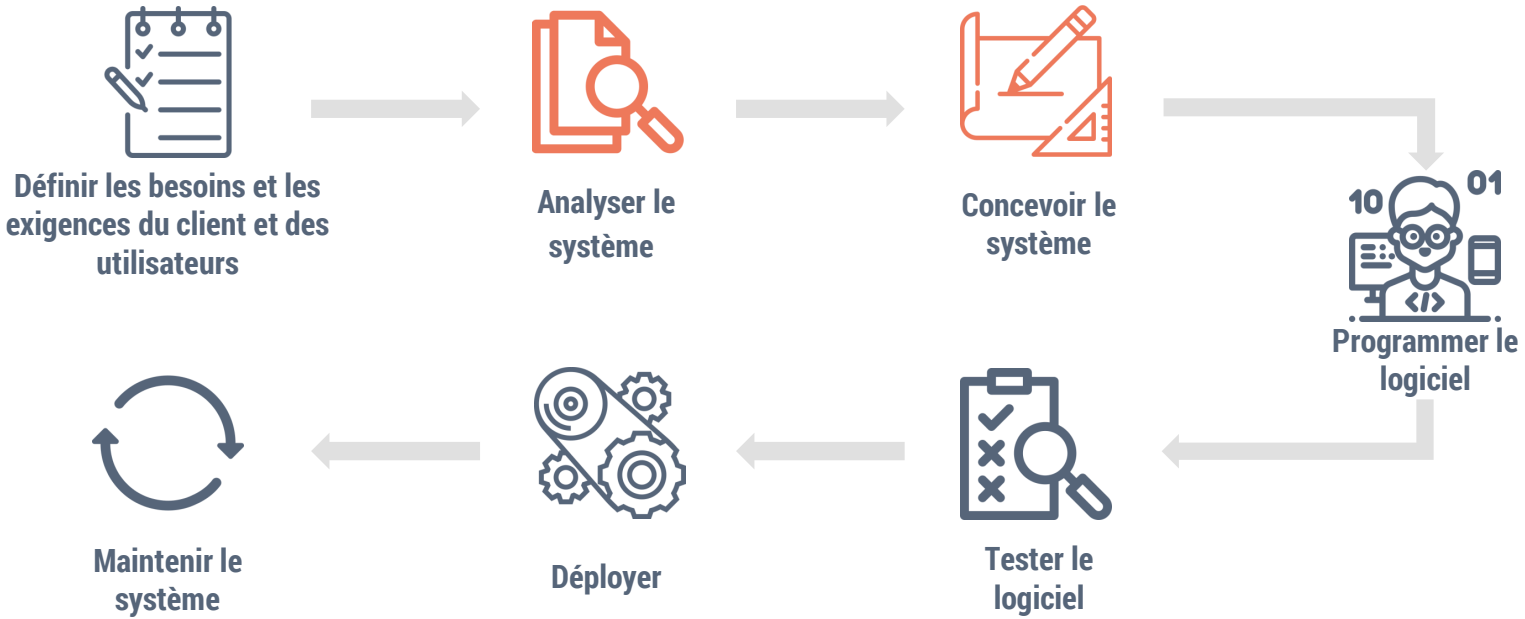
- **Validité :** aptitude d'un produit logiciel à remplir exactement ses fonctions, définies par le cahier des charges et les spécifications.
- **Fiabilité ou robustesse :** aptitude d'un produit logiciel à fonctionner dans des conditions anormales.
- **Extensibilité (maintenance) :** facilité avec laquelle un logiciel se prête à sa maintenance, c'est-à-dire 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.
- **Compatibilité :** facilité avec laquelle un logiciel peut être combiné avec d'autres logiciels.

Critères de qualité d'un logiciel

- **Efficacité :** Utilisation optimale des ressources matérielles.
- **Portabilité :** facilité avec laquelle un logiciel peut être transféré sous différents environnements matériels et logiciels.
- **Vérifiabilité :** facilité de préparation des procédures de test.
- **Intégrité :** aptitude d'un logiciel à protéger son code et ses données contre des accès non autorisés.
- **Facilité d'emploi :** facilité d'apprentissage, d'utilisation, de préparation des données, d'interprétation des erreurs et de rattrapage en cas d'erreur d'utilisation.

Les étapes de développement logiciel

Le processus de développement logiciel contient un certain nombre d'étapes :



Définition des besoins et des exigences

Les étapes de développement logiciel

- La définition des besoins et des exigences correspond à l'étape dans laquelle nous discutions avec le client et les futurs utilisateurs afin de comprendre de quoi ils ont besoin : QUI doit pouvoir faire QUOI ?

- Lors de cette étape, nous définissions également les demandes précises, telles que le respect de certaines normes graphiques, les temps de réponse, le matériel sur lesquels le logiciel devrait fonctionner, etc.

Analyser le système

Les étapes de développement logiciel

- L'analyse du système permet d'affiner ce qui a été défini dans l'étape précédente.

- Dans cette étape de développement logiciel on détaille davantage le fonctionnement interne du futur logiciel (COMMENT cela doit-il fonctionner ?).

Conception du système

Les étapes de développement logiciel

- La conception du système correspond à la définition de choix techniques.

- La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

Programmer le logiciel

Les étapes de développement logiciel

- La programmation est l'étape dans laquelle les informaticiens réalisent le logiciel à l'aide de langages de programmation, de systèmes de gestion de bases de données, etc.

Tester le logiciel

Les étapes de développement logiciel

- Durant les tests, les informaticiens vérifient que le logiciel fonctionne et répond aux besoins définis en début du projet.
- Cette phase de tests peut intégrer des validations du logiciel avec le client et/ou les utilisateurs. C'est même plus que souhaité.

Déployer

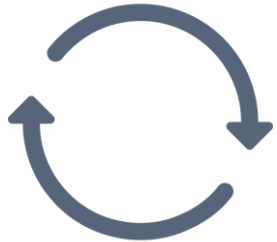
Les étapes de développement logiciel

- Lors du déploiement, les informaticiens installent le logiciel sur le matériel et réalisent des ajustements pour faire fonctionner le logiciel dans l'environnement de travail des utilisateurs.

Maintenir le système

Les étapes de développement logiciel

- La maintenance correspond à la période qui suit l'installation et pendant laquelle les anomalies et problèmes doivent être corrigés.



Cycle de vie d'un logiciel

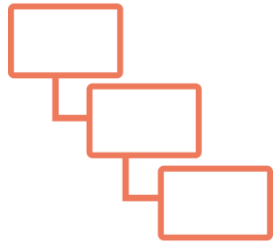
Les étapes de développement logiciel ne sont pas forcément utilisées de façon linéaire. On parle souvent de cycles de vie, qui ont pour but d'organiser ces étapes de différentes manières en fonction d'un certain nombre de critères relatifs au projet de développement. Ci-dessous, quelques exemples de cycle de vie.

- **La cascade**
- **Le modèle en V**
- Le modèle en W
- La spirale
- Le RAD
- Etc.

Le Cycle de vie en cascade

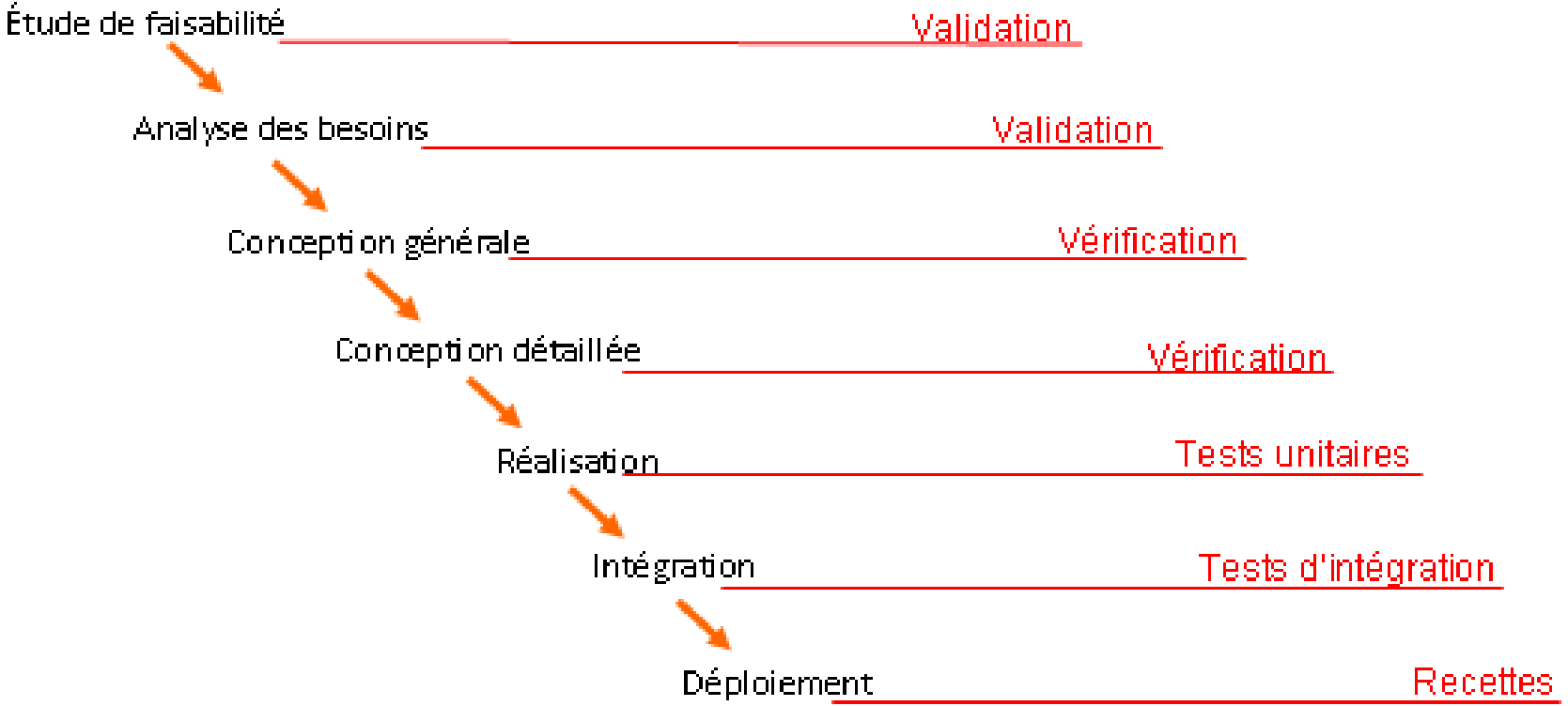
Cycle de vie d'un logiciel

- Le cycle de vie en cascade peut être utilisé lorsqu'un projet est relativement simple, c'est-à-dire pour lequel nous avons la quasi-certitude que les besoins ou exigences n'évolueront pas en cours de projet.
- En pratique, une étape ne démarre que si la précédente ait été validée par le client et/ou les utilisateurs.



Le Cycle de vie en cascade

Cycle de vie d'un logiciel



Le Cycle de vie en cascade

Cycle de vie d'un logiciel

1. **Étude de faisabilité:** Cette étape permet de décider de la nécessité et de l'opportunité de lancer le projet.
2. **Analyse des besoins:** Cette étape correspond à l'étape n°1 énoncé ci-dessus.
3. **Conception générale** : correspond à l'étape Analyse du système.
4. **Conception détaillée** Cette étape est équivalente à l'étape Conception du système.
5. **Réalisation:** Cette étape équivalente à l'étape La programmation.
6. **Intégration:** Cette étape équivaut à une partie de l'étape La programmation. Notez que l'étape Tests n'est pas vu ici comme une étape, mais comme des validations.
7. **Le déploiement:** Cette étape équivalente à l'étape Déployer.

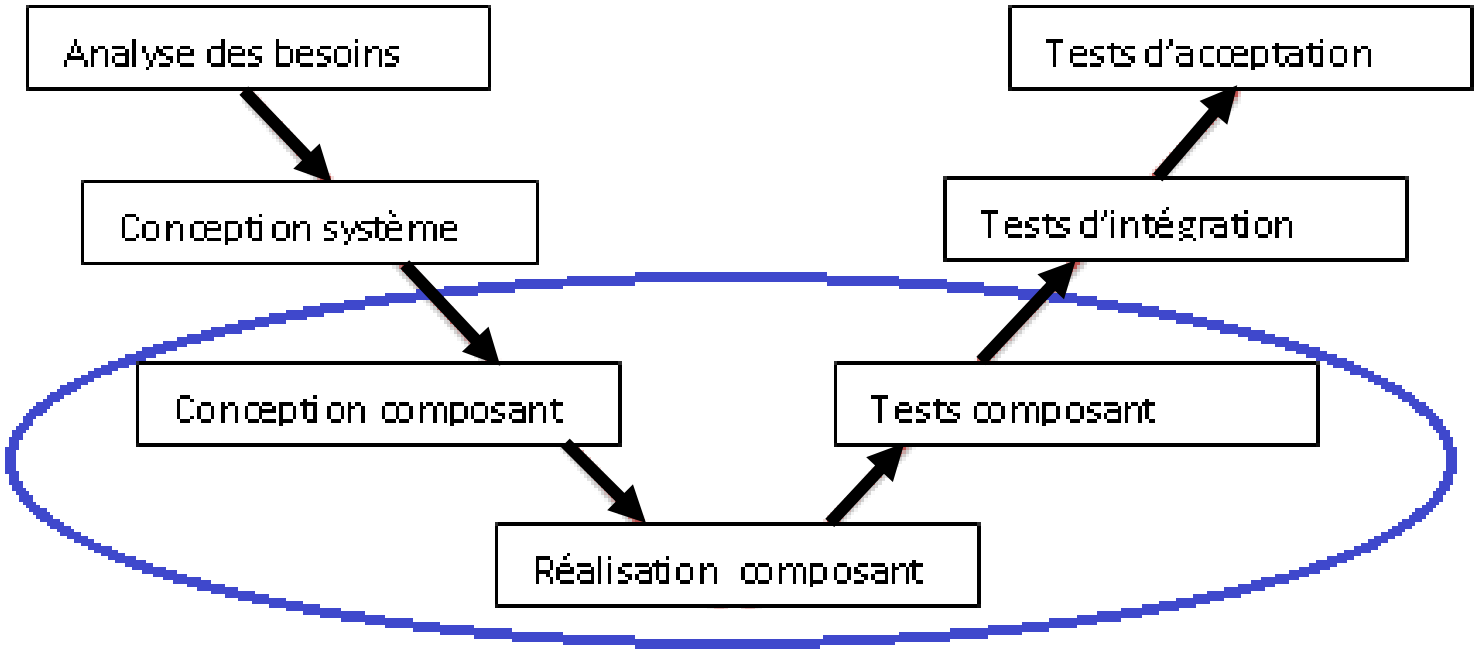
Le Cycle de vie en V

Cycle de vie d'un logiciel

- Un projet plus important dont les besoins et les exigences risquent d'évoluer pourrait avoir **un cycle en V**.
- Comme illustré dans le diapositive suivant, le système à développer serait décomposé en modules. Chaque module serait **conçu, développé et testé séparément**.
- Les différents modules pourraient alors être intégrés dans le système global au fur et à mesure.
- Des tests d'intégration permettraient alors de garantir que l'ensemble fonctionne de façon à répondre à la conception générale du système.
- Les tests d'acceptation permettent ensuite de vérifier la cohérence du système avec les besoins.

Le Cycle de vie en V

Cycle de vie d'un logiciel



Le Cycle de vie en cascade

Cycle de vie d'un logiciel

1. **Analyse des besoins** : identique à l'étape Définition des besoins et des exigences.
2. **Conception système :** Cette étape couvre une partie de l'étape : l'analyse du système et de l'étape : la conception du système. Il s'agit d'une vision globale du logiciel à développer.
3. Le système est divisé en composants pour lesquels on réalisera :
4. **Conception composant** : l'analyse et la conception détaillée du composant
5. **Réalisation** : composant La réalisation ou la programmation du composant
6. **Test composant :** Cela correspond donc aux étapes la programmation et les tests.
7. **Tests d'intégration :** Les différents composants sont alors intégrés dans un logiciel.
8. **Tests d'acceptation:** correspondent à la validation du logiciel par le client et les utilisateurs.

Résumé

- Un logiciel est un système d'information automatisé.
- Le processus de développement logiciel contient un certain nombre d'étapes :
    - Définir les besoins et les exigences du client et des utilisateurs, Analyser le système, Concevoir le système, Programmer le logiciel, Tester le logiciel, Déployer et Maintenir le système.
- L'objectif d'un cycle de vie est d'organiser les différentes étapes du processus de développement logiciel en fonction d'un certain nombre de critères relatifs au projet de développement.
    - Les cycles de vie en cascade et en V sont les plus utilisés.