

Génie logiciel - Introduction

Nuwan Herath

2022-2023

Cours est basé sur ceux de Claude Godart et de Myriam Servières.

- 1 Définitions
- 2 La qualité du logiciel
- 3 Historique du génie logiciel
- 4 Processus logiciel
Présentation
Les processus orientés plan
- 5 Pour aller plus loin

Définitions

Définitions du génie logiciel

Le génie logiciel est l'ensemble des moyens techniques, industriels et humains qu'il faut réunir pour spécifier, construire, distribuer et maintenir les logiciels.

¹Le génie Logiciel, Jacques Printz, Que sais-je?, presses universitaires de France, 1995.

Définitions du génie logiciel

Le génie logiciel est l'ensemble des moyens techniques, industriels et humains qu'il faut réunir pour spécifier, construire, distribuer et maintenir les logiciels.

Ils doivent être :

- sûrs,
- conviviaux,
- évolutifs,
- économiques.¹

¹Le génie Logiciel, Jacques Printz, Que sais-je?, presses universitaires de France, 1995.

Définitions du génie logiciel

Ils doivent être :

- **sûrs**, ils réagissent de façon déterministe à toutes les sollicitations, éventuellement entachées d'erreurs qui correspondent à sa mission,
- **conviviaux**, ils sont adaptés aux capacités réelles et non supposées, des usagers,
- **évolutifs**, ils s'adaptent aux nouveaux besoins dans des délais raisonnables,
- **économiques**, ils réalisent l'optimum entre le service rendu (fonctions offertes, retour sur investissement, contrat de service, etc.) et les coûts de développement-maintenance initialement annoncés.¹

¹Le génie Logiciel, Jacques Printz, Que sais-je?, presses universitaires de France, 1995.

Les disciplines connexes du génie logiciel

- Génie informatique
- Informatique
- Mathématiques
- Management
- Management de projet
- Qualité
- Ergonomie logicielle
- Ingénierie des systèmes

Définition d'un logiciel

Un logiciel est

- un ensemble de **programmes**, qui permet à un ordinateur ou à un système informatique d'assurer une tâche ou une fonction en particulier,
- les **données** qu'ils utilisent,
- les **documents** qui servent à concevoir ces programmes et ces données, à les mettre en œuvre, à les utiliser et à les modifier.

Le coût des logiciels

L'unité de coût des logiciels est le homme-mois (hm) ou le homme-ans (ha) différent de la durée de développement

Exemple

Emploi de 3 ingénieurs pendant 18 mois :

- 54 hm
- 4,5 ha

- Le coût d'un logiciel est
 - d'autant plus important que sa durée de vie est longue
 - très sensible à la qualité de l'architecture initiale du logiciel
- Les coûts sont globaux : Développement + Maintenance + Exploitation

La taille des logiciels

Le volume ou taille d'un logiciel est exprimé en nombre de lignes de code source que comporte le logiciel final (ls ou kls pour millier de lignes sources)

Ce paramètre a été retenu comme indicateur principal de la quantité d'information contenue dans le logiciel

<https://www.informationisbeautiful.net/visualizations/million-lines-of-code/>

Remarque

Un logiciel de 100 kls avec ses annexes représente environ 10 ouvrages de 400 pages

La productivité du développement

La productivité d'un développement s'exprimera en ls par hm
Cet indicateur dénote la difficulté de fabrication du logiciel

La qualité du logiciel

Les critères de qualité des logiciels

Les différentes qualités d'un logiciel sont :

- la **fiabilité** (se comporte tel qu'espéré durant un intervalle de temps donné)
- la **facilité de maintenance** (l'entretien peut être réalisé à moindre coût)
- l'**évolutivité** (autorise la prise en compte de nouvelles exigences)

²Introduction au génie Logiciel, Guillaume Raschia, polytech' Nantes, 2004.

Les critères de qualité des logiciels

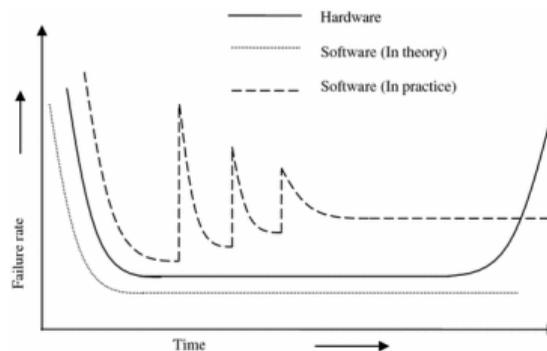
Les différentes qualités d'un logiciel sont :

- la fiabilité (se comporte tel qu'espéré durant un intervalle de temps donné)
- la facilité de maintenance (l'entretien peut être réalisé à moindre coût)
- l'évolutivité (autorise la prise en compte de nouvelles exigences)
- la correction (ou validité/conformité) (se comporte exactement tel que décrit par sa spécification)
- la robustesse (se comporte de façon convenable, même dans des circonstances non prévues par la spécification)
- la performance (ou efficacité) (exploite les ressources de façon au moins rentable, sinon optimale)
- la convivialité (les usagers le trouvent agréable et facile à utiliser)
- la vérifiabilité (peut être aisément vérifié/testé)
- la réutilisabilité (tout ou partie peut être utilisé à nouveau)
- la portabilité (peut être utilisé sur des plateformes différentes)
- l'interopérabilité (capable de cohabiter voire de coopérer avec d'autres systèmes)²

²Introduction au génie Logiciel, Guillaume Raschia, polytech' Nantes, 2004.

Les critères de qualité des logiciels

La maintenance et l'évolutivité affectent la fiabilité



Source : Reliability and Safety Engineering, V. Ajit Kumar, A. Srividya et K. Durga Rao, Springer London, 2016

Comment construire des programmes à la fois aisément déterminables et évolutifs ?

Les critères de qualité des logiciels

La qualité d'un logiciel est perçue différemment selon :

- le point de vue de l'évaluateur
 - Développeur
 - Usager
 - Client
- le domaine d'application
 - Système transactionnel
 - Système temps réel
 - Système embarqué
 - Système distribué

Comment définir la qualité d'un logiciel ?

- Les différents facteurs de qualité du logiciel peuvent être incompatibles (ex : la portabilité d'un programme peut se faire au détriment de ses performances)
- Pour chaque logiciel, il faut établir des priorités parmi les critères de qualité qui guideront la conception

De la qualité logicielle à la qualité du développement logiciel

Recherche de qualité des logiciels → qualité du développement logiciel

De la qualité logicielle à la qualité du développement logiciel

Le génie logiciel doit assurer :

- La qualité du logiciel
- La qualité du procédé de développement :
 - Productivité :
 - Diminution des coûts
 - Réutilisation
 - Outils et environnements adaptés
 - Ponctualité : livraison d'un produit utilisant les techniques "actuelles"
 - Visibilité (transparence) : documentation et autres documents de production

Historique du génie logiciel

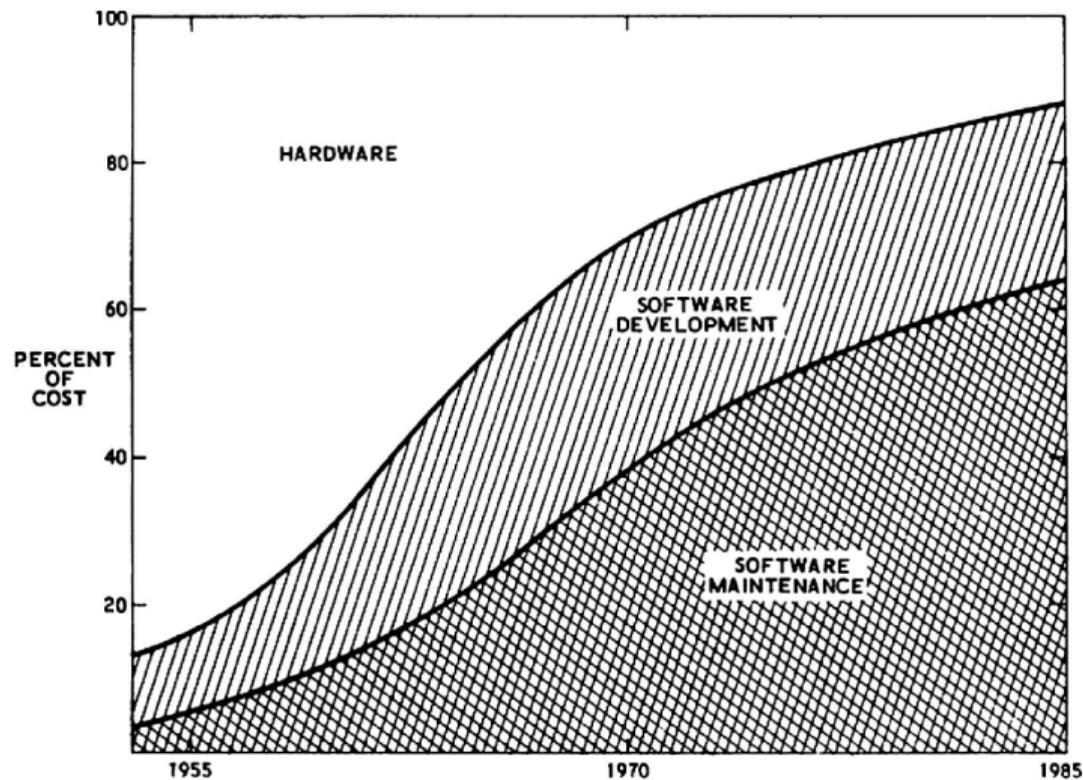
Bref historique

- **Années 50** : petits programmes sur mesure
 - Le client est souvent le programmeur lui-même
 - Les débuts de la profession de programmeur
- **Années 60 et 70** : les premiers logiciels de taille importante voient le jour
 - Centaines de clients
 - Bases de données

Crise du logiciel

- **Années 80** : l'industrie du logiciel se développe en même temps que les ordinateurs personnels, les réseaux, ...
 - Le coût du matériel baisse, les logiciels apparaissent en grand nombre
 - Les réseaux et les systèmes répartis se développent, augmentant encore l'offre et la complexité des logiciels

La crise du logiciel



Les symptômes de la crise du logiciel

- Coûts de développement difficiles à prévoir
- Date de rendu du projet souvent dépassée
- Qualité des logiciels produits faible
- Maintenance difficile et coûteuse
- Efficacité limitée des programmes
- Cahier des charges non respecté

Les causes de la crise du logiciel

Il est difficile de développer un logiciel et de gérer un projet de développement logiciel car un logiciel est :

- complexe
- invisible
- facile à altérer ³

³Software Engineering, Why? What, Alfred Strohmeier, École Polytechnique Fédérale de Lausanne, 2000.

Les causes de la crise du logiciel

Il est difficile de développer un logiciel et de gérer un projet de développement logiciel car un logiciel est :

- complexe → difficile à comprendre
- invisible
- facile à altérer ³

³Software Engineering, Why? What, Alfred Strohmeier, École Polytechnique Fédérale de Lausanne, 2000.

Les causes de la crise du logiciel

Il est difficile de développer un logiciel et de gérer un projet de développement logiciel car un logiciel est :

- complexe → difficile à comprendre
- invisible → qualité difficile à mesurer et progression pas facile à suivre
- facile à altérer ³

³Software Engineering, Why? What, Alfred Strohmeier, École Polytechnique Fédérale de Lausanne, 2000.

Les causes de la crise du logiciel

Il est difficile de développer un logiciel et de gérer un projet de développement logiciel car un logiciel est :

- complexe → difficile à comprendre
- invisible → qualité difficile à mesurer et progression pas facile à suivre
- facile à altérer → difficile d'appréhender toutes les conséquences³

³Software Engineering, Why? What, Alfred Strohmeier, École Polytechnique Fédérale de Lausanne, 2000.

La naissance du génie logiciel

Le génie logiciel (*software engineering*) est né en octobre 1968 à Garmisch-Partenkirchen en Allemagne, lors d'une conférence de l'OTAN

Objectifs du génie logiciel

Développer des logiciels qui :

- remplissent les besoins des clients
- peuvent être maintenus (i.e. adaptés à des changements continus)
- en respectant le budget
- en respectant les délais

Principes du génie logiciel

- Utiliser des principes et des méthodes de programmation
- Suivre des méthodes, procédures et standards bien identifiés
- Utiliser des outils logiciels
- Utiliser des techniques de management

Quelques cas d'école⁴

- 1962 : la sonde Mariner 1 vers Venus a été détruite après 290 secondes de vol, à cause d'un trait d'union dans un programme FORTRAN (coût 80 million de dollars)
- 1971 : lors d'une expérience météorologique en France, 72 ballons contenant des instruments de mesure furent instantanément détruits à la suite d'un défaut dans le logiciel
- 1981 : un problème logiciel retarda de 2 jours le premier lancement de la navette spatiale Columbia
- juin et au début juillet 1991 : une série de coupures affectèrent les abonnés au téléphone de Los Angeles, San Francisco, Washington, Baltimore, Virginia dans l'État de Virginie et Greensboro en Caroline du Nord à cause d'une réaction en chaîne due au changement de version du logiciel

⁴Précis de génie logiciel, M-C Gaudel et al., Masson, 1996.

Un cas d'école français⁵

Le 22/12/2001 750 000 terminaux de paiement ne répondent plus

→ longues files d'attente

- Saturation des serveurs de la société Atos chargées des autorisations de paiements dépassant 600F. Habituellement quelques dizaines de secondes, ce jour-là : 30 min.
- Des clients abandonnent leurs chariots pleins. Le groupe Leclerc a chiffré son préjudice à 2 millions d'euros.

⁵Cours de Génie Logiciel, François Jacquenet, Université de Saint-Étienne.

Le fameux vol 501 d'Ariane V

Explosion lors du premier lancement de la fusée Ariane V le 4 juin 1996

- Logiciel de plateforme inertielle repris tel quel d'Ariane IV sans nouvelle validation
- Ariane V ayant des moteurs plus puissants s'incline plus rapidement que Ariane IV, pour récupérer l'accélération due à la rotation de la Terre
- Les capteurs ont bien détecté cette inclinaison d'Ariane V, mais le logiciel l'a jugée non conforme au plan de tir (d'Ariane IV), et a provoqué l'ordre d'autodestruction
- En fait tout se passait bien...
- Coût du programme d'étude d'Ariane V : 38 milliards de Francs, pour 39 secondes de vol après 10 années de travail....

Processus logiciel

Processus logiciel

Présentation

Objectif des processus logiciel

- Rationaliser
- Formaliser
- Automatiser le développement du logiciel

“Software processes are software too”

— L. Osterweil, 1987

Définition

- Un processus fournit des principes (des règles flexibles) pour le bon enchaînement des activités de développement de logiciel
- C'est un guide qui définit des garde-fous mais au sein desquels il existe beaucoup de libertés (planification imprécise) et il doit exister beaucoup de liberté (approche agile)

Les processus logiciel

- Les bonnes pratiques :
 - la planification
 - le développement incrémental
 - la ré-utilisation
 - la libération de la créativité
- Un modèle de processus utilise ces ingrédients à des doses différentes :
 - les méthodes **en cascade** et **en V** favorisent la planification
 - les méthodes **en spirale** et **RUP** favorisent le développement incrémental avec une bonne dose de planification
 - la méthode **agile** favorise la créativité avec une bonne dose de développement incrémental

Aujourd'hui on ne fait pas de développement logiciel sans réutiliser (.Net, JEE, services web...)

Y a-t-il un processus universel ?

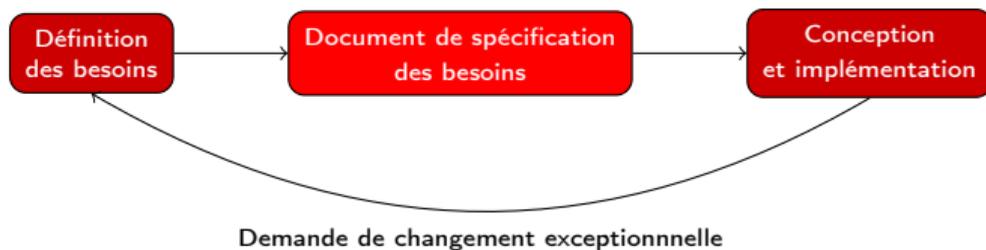
- S'il y a une tendance aujourd'hui à favoriser les processus agiles, ceux-ci ne semblent pas adaptés aux gros projets
- Le **Processus unifié** est construit autour de la notation (universelle) UML mais est loin d'être universellement utilisé
- Les processus réels sont très influencés par la réutilisation.

Deux approches plus ou moins antagonistes... ou plus ou moins complémentaires

- Approches orientées plan
- Approches agiles

Approche orientée plan

Diagramme



Approche orientée plan

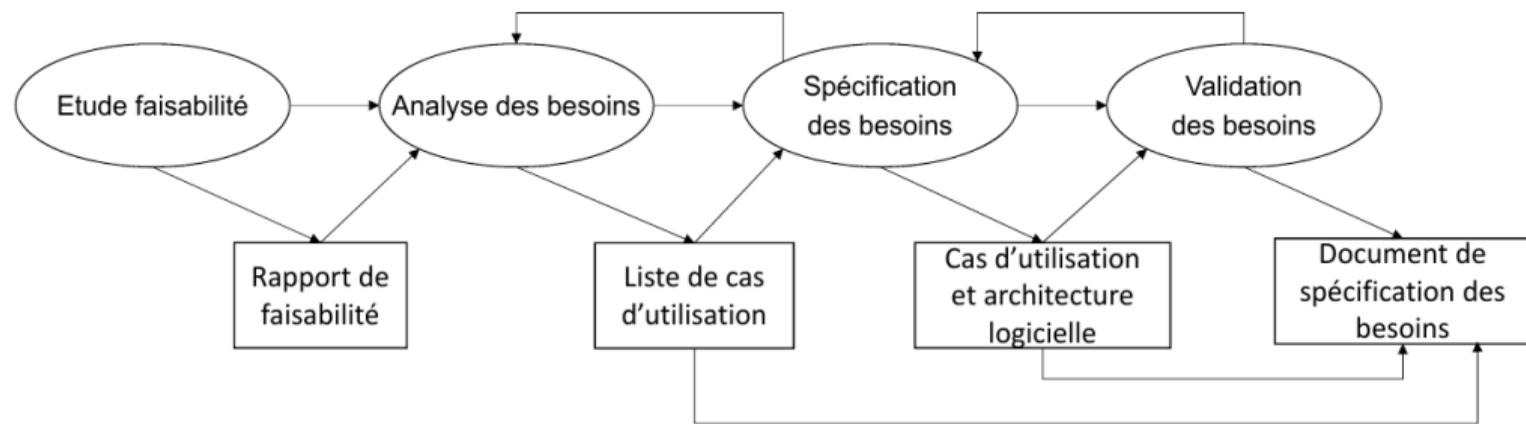
Valeurs

- Planifier précisément en termes d'usage, de coût, de délais
- Respecter les contrats avec le client
- Documenter le logiciel pour en faciliter la maintenance
- Eviter les changements en cours de développement

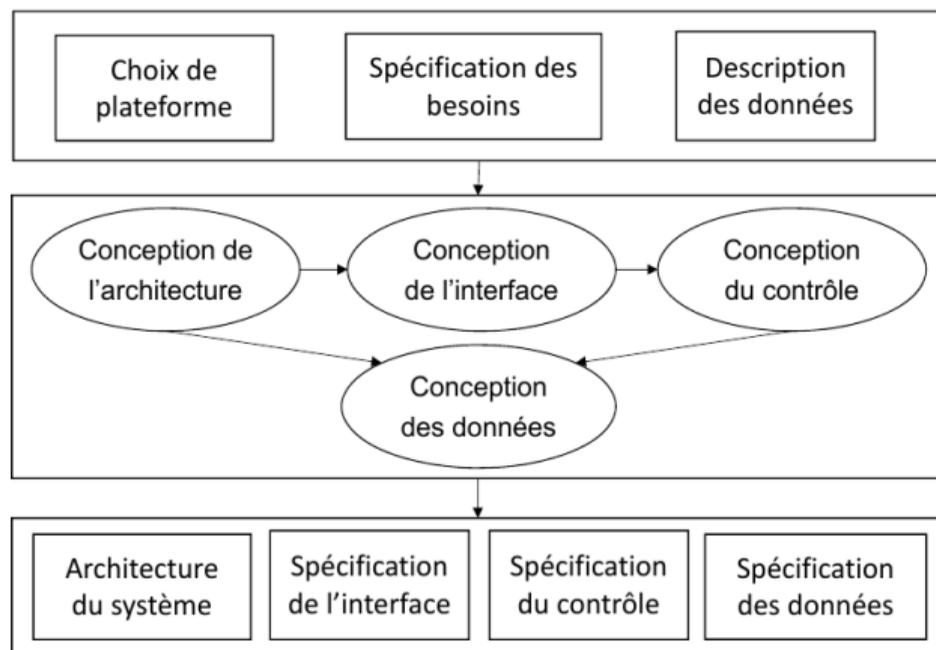
Les activités du développement logiciel

- Les activités de développement sont plus ou moins nombreuses en fonction du niveau de détail et en fonction du modèle de développement considérés
- On considère ici les concepts généraux suivants :
 - la spécification des besoins
 - la conception
 - l'implémentation
 - la validation
 - l'évolution

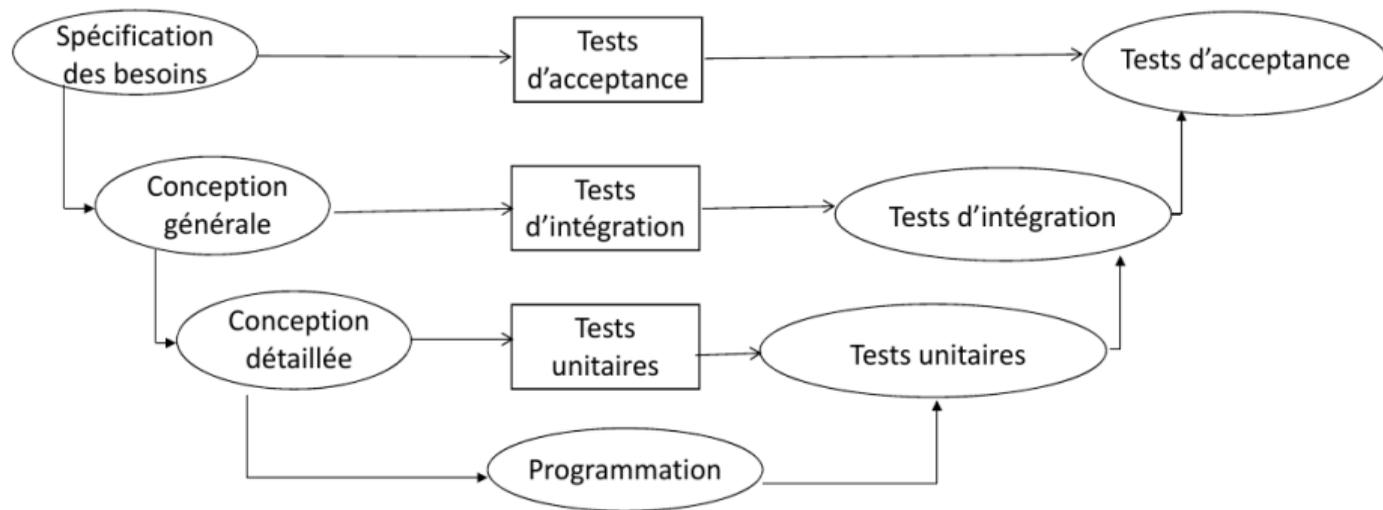
La spécification des besoins



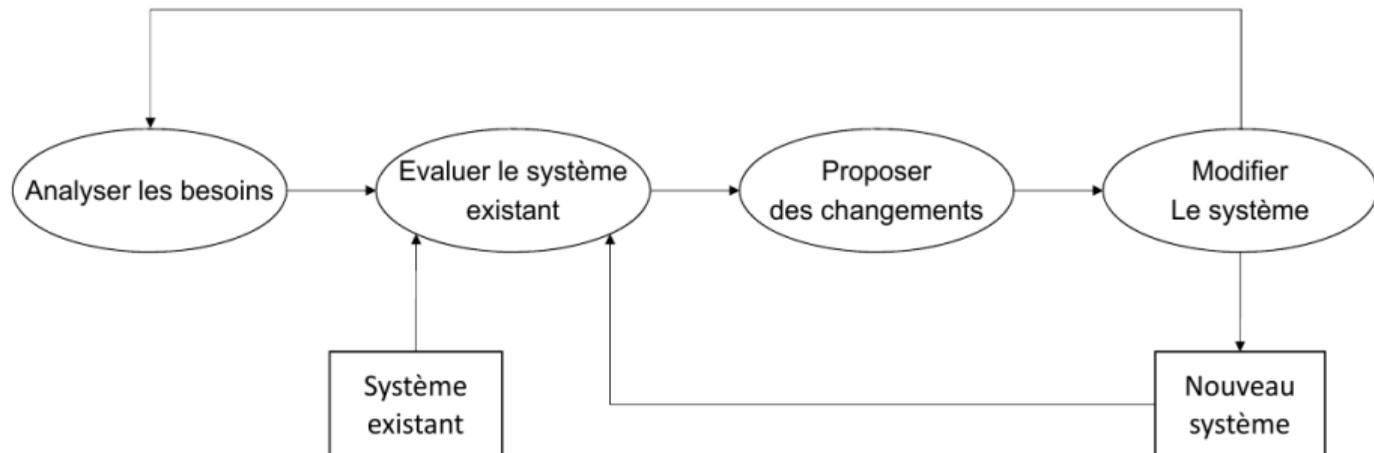
La conception et l'implémentation



La validation



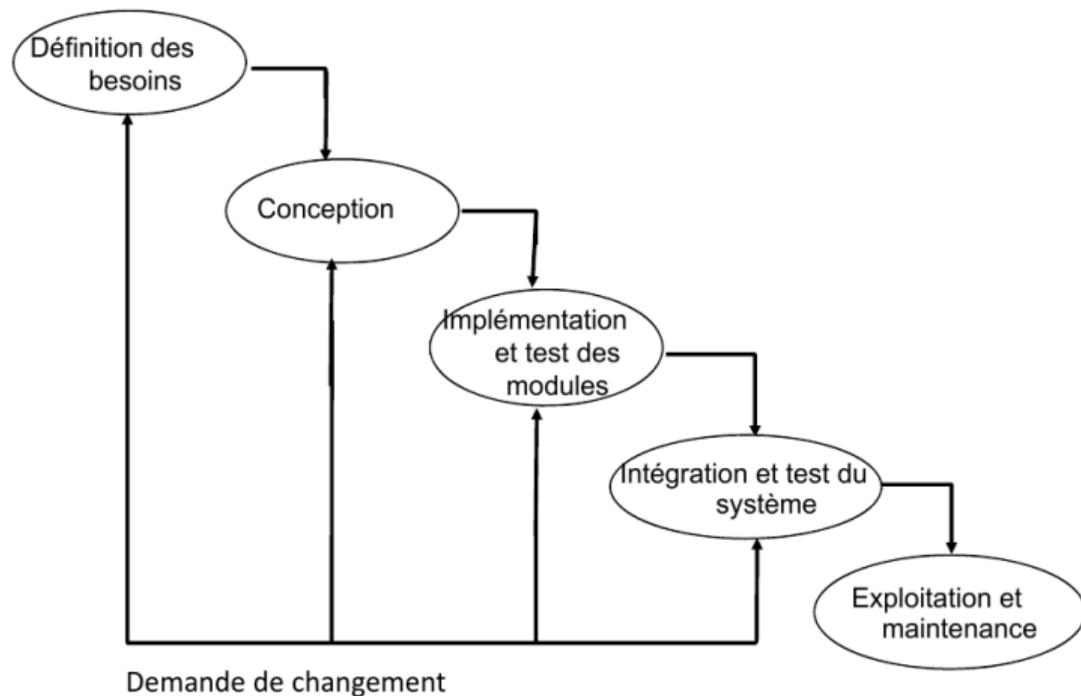
L'évolution



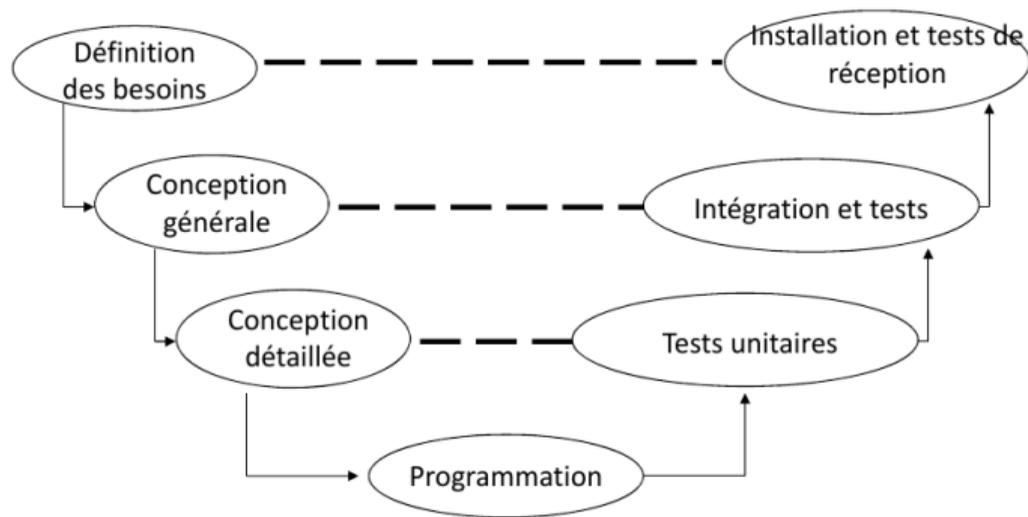
Processus logiciel

Les processus orientés plan

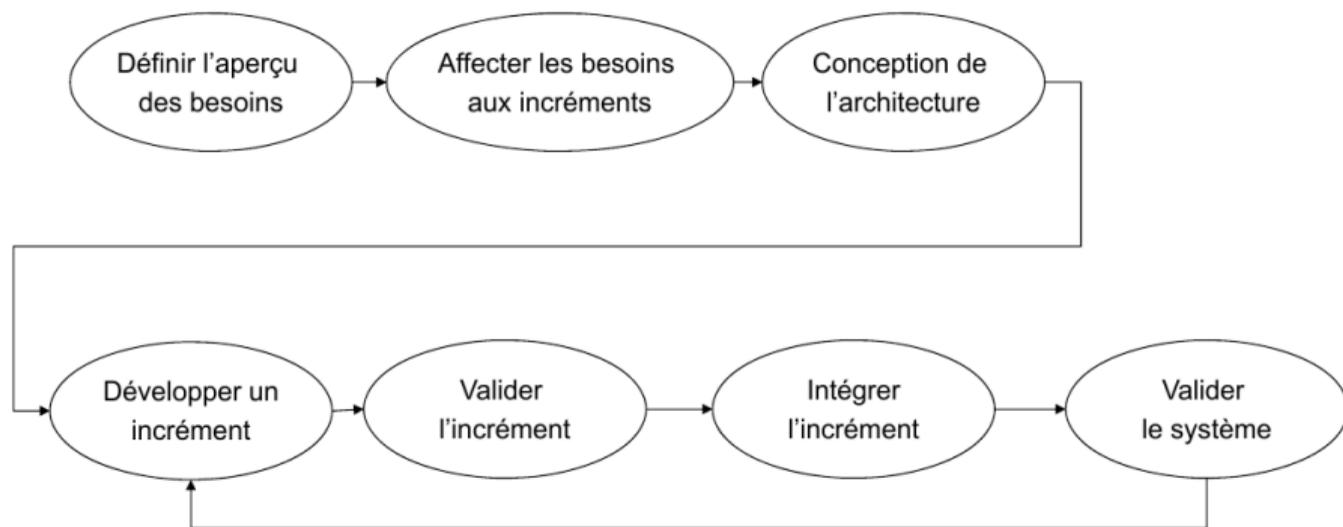
Modèle en cascade



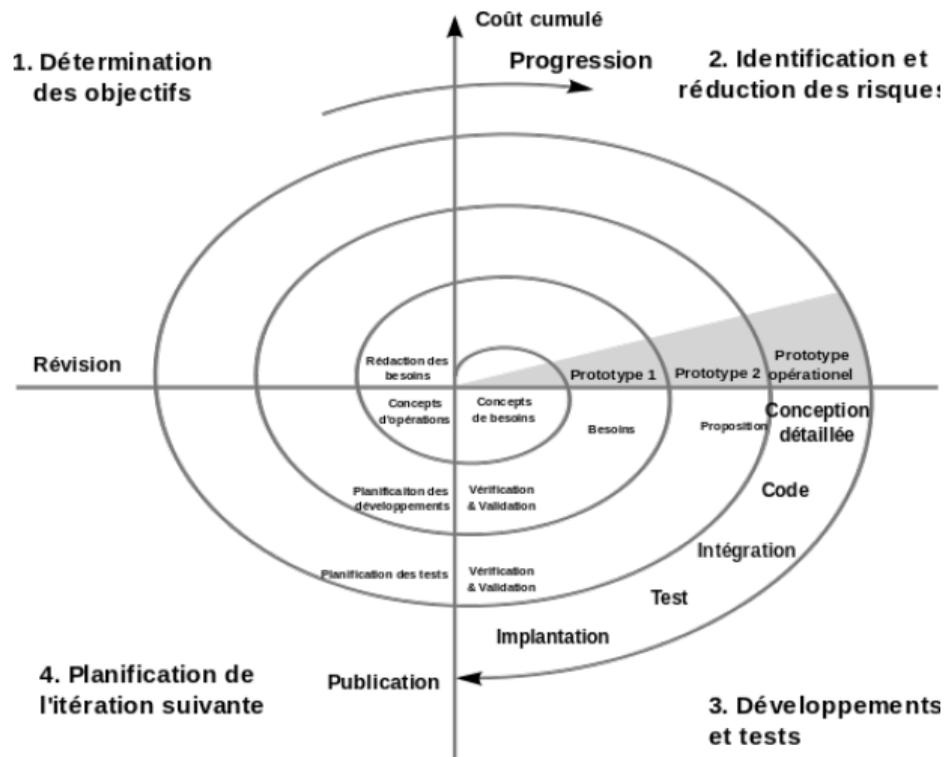
Modèle en V



Modèle incrémental



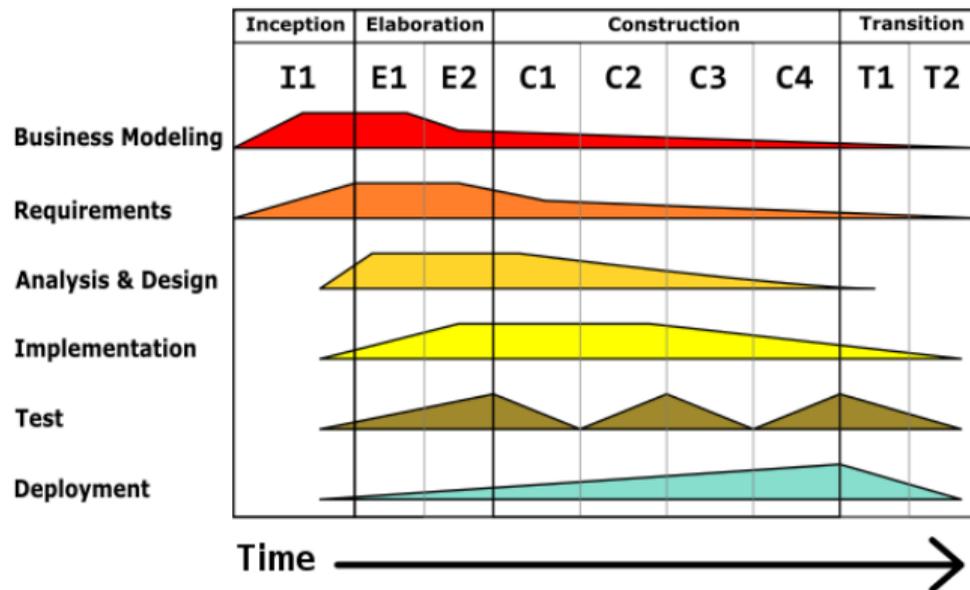
Modèle en spirale



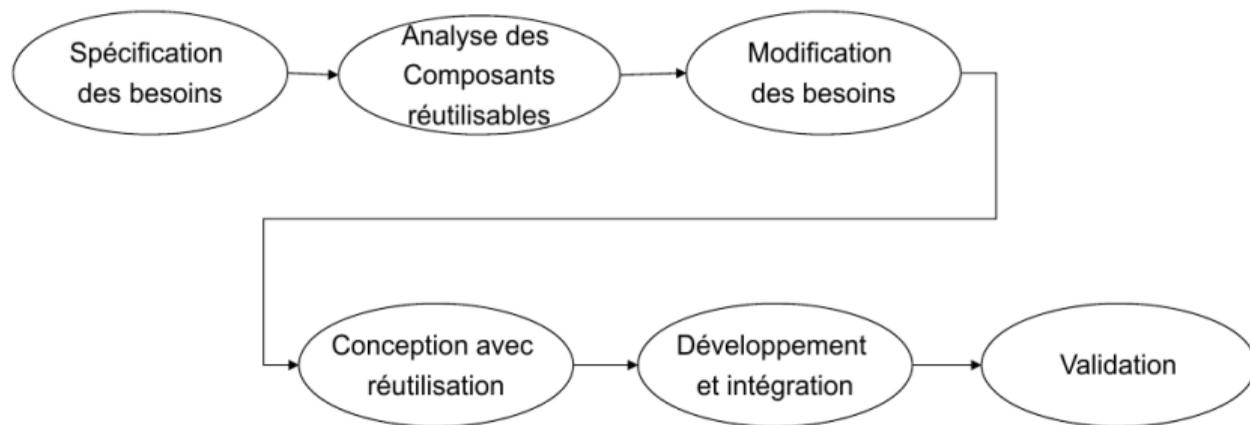
Organisation matricielle (RUP)

Iterative Development

Business value is delivered incrementally in time-boxed crossdiscipline iterations.



Modèle par réutilisation



Pour aller plus loin

Des références bibliographiques

- Software Engineering, Why? What, Alfred Strohmeier, École Polytechnique Fédérale de Lausanne, 2000.
- Guide to the Software Engineering Body of Knowledge, IEEE Computer Society Professional Practices Committee, 2004.
- Précis de génie logiciel, M-C Gaudel et al., Masson, 1996
- Le génie logiciel et ses applications, Ian Sommerville, InterEditions, 1988.
- Processus d'ingénierie du logiciel - Méthodes et Qualité, Claude Pinet, Pearson Education, 2002.