Processus de Développement du Logiciel

L.Kzaz Novembre 2019

Plan.

- Introduction.
- Le génie logiciel
- La qualité du logiciel.
- Le cycle de vie du logiciel.
- Le processus de développement.

Processus de Développement du Logiciel

INTRODUCTION

Introduction

Place du Logiciel dans les SI:

La construction d'un SI Informatisé, SII, passe obligatoirement par :

- *La mise en place d'une infrastructure matérielle comportant des d'ordinateurs des serveurs, des réseaux etc.*

Mais aussi, et surtout par :

- *La mise en place d'une infrastructure logicielle.*

La première est de nature : Physique et tangible.

Alors que, la seconde est : Immatérielle et intangible.

Introduction

Place du Logiciel dans les SI:

L'OCDE considère que le logiciel est déjà le « secteur le plus important économiquement, et qu'il est à l'économie fondée sur le savoir ce que les secteurs de l'acier et de l'automobile étaient à l'économie industrielle ».

La part du logiciel dans le coût d'un Système Informatique est estimée à 80%, alors que le matériel n'est que est de 20%.

La fabrication du matériel est assurée par quelques grands constructeurs. Il est relativement fiable et le marché est standardisé.

Les problèmes liés à l'informatique sont essentiellement des problèmes de Logiciel.

Introduction

Spécificités du Logiciel:

- *Le logiciel est un produit immatériel dont l'existence est indépendante du support physique. Il est semblable à une œuvre d'art (Roman, Partition musicale, etc.)*

- *C'est un objet technique fortement contraint*
  - *Fonctionne ou ne fonctionne pas*
  - *Structure complexe*
  - *Relève des modes de travail du domaine technique.*

- *Il a un cycle de production spécifique:*
  - *La reproduction pose peu de problèmes, seule la première copie d'un logiciel a un coût*
  - *Production a l'unité*
  - *Semblable au Génie Civil (ponts, routes...)*

Introduction

Spécificités du Logiciel:

- *Le logiciel est un produit très malléable au sens de « facile à modifier »,*
- *es caractéristiques attendues sont difficiles à figer au départ et souvent remises en cause en cours de développement,*
- *les défaillances et erreurs ne proviennent ni de défauts dans les matériaux ni de phénomènes d'usure dont on connaît les lois mais d'erreurs humaines, inhérentes à l'activité de développement,*
- *le logiciel ne s'use pas, il devient obsolète (par rapport aux concurrents, par rapport au contexte technique, par rapport aux autres logiciels...),*

Introduction

Spécificités du Logiciel:

Le logiciel est produit pour répondre aux besoins d'une large variété de domaines et systèmes:

- *Gros systèmes de gestion (ou systèmes d'information) ; le plus souvent des systèmes transactionnels construits autour d'une base de données;*
- *Systèmes temps réel, qui doivent répondre à des événements dans des limites de temps prédéfinies et strictes ;*
- *Systèmes distribués sur un réseau de machines (distribution des données et/ou des traitements);*
- *Systèmes embarqués et systèmes critiques, interfacés avec un système à contrôler (ex: aéronautique, centrales nucléaires...).*

Introduction

Le Génie Logiciel:

La production du logiciel est une activité qui relève du domaine du « Génie logiciel / Software engineering », définie comme étant un domaine des « sciences de l'ingénieur » dont la finalité est :

- *La conception,*
- *La fabrication,*
- *La maintenance de systèmes logiciels complexes.*

Le GL se définit souvent par opposition à la 'programmation', c'est-à-dire la production d'un programme par un individu unique, considérée comme « facile ».

Dans le cas du GL, il s'agit de la fabrication collective d'un système complexe, concrétisée par un ensemble de documents de conception, de programmes et de jeux de tests avec souvent de multiples versions, et considérée comme « difficile ».

Introduction

Le Génie Logiciel:

Le GL est difficile à étudier car:

- *Très vaste,*
- *Pas toujours très précis (beaucoup de discours généraux),*
- *Foisonnant dans les concepts et le vocabulaire,*
- *Sensible aux effets de modes,*
- *Les aspects techniques nécessitent une bonne maîtrise des outils fondamentaux de l'informatique (programmation, BD, système / réseau...).*

Introduction

Le Génie Logiciel:

Le GL se préoccupe des procédés de fabrication des logiciels de façon à s'assurer que les quatre exigences suivantes soient satisfaites.

- *Exigences Fonctionnelles (Besoins, Spécifications fonctionnelles) des utilisateurs.*
- *Exigences de Qualité spécifiées dans un contrat de service initial.*
- *Exigences de Coût. Le coût de production doit rester dans les limites prévues au départ.*
- *Exigences de Délai. Le délai de production doit rester dans les limites prévues au départ.*

Processus de Développement du Logiciel

Qualité du Logiciel

La qualité du logiciel

La qualité du logiciel est une notion multiforme. L'ISO/IEC 9126 propose 6 caractéristiques de qualité du produit logiciel:

- *Capacité fonctionnelle (Functionality)*
- *Fiabilité (Reliability)*
- *Facilité d'utilisation (Usability)*
- *Rendement (Efficiency)*
- *Maintenabilité (Maintainability)*
- *Portabilité (Portability)*
- *……*

La qualité du logiciel

L'ISO/IEC 9126 propose 6 caractéristiques de qualité du produit logiciel:

1. *Capacité fonctionnelle (Functionality)*

   La capacité qu'ont les fonctionnalités d'un logiciel à répondre aux exigences et besoins explicites ou implicites des usagers.

   En font partie:

   La précision, l'interopérabilité, la conformité aux normes et la sécurité.

   - *Aptitude : présence et adéquation d'une série de fonctions pour les tâches données.*
   - *Exactitude : résultats ou effets justes ou convenus.*
   - *Interopérabilité : interactions avec d'autres systèmes.*
   - *Sécurité : accès non autorisé (accidentel ou délibéré) aux programmes et données*

La qualité du logiciel

2. Facilité d'utilisation (Usability)

Elle porte sur l'effort nécessaire pour apprendre à manipuler le logiciel.

En font partie:

- *Facilité de compréhension : effort de l'utilisateur pour comprendre la logique et la mise en œuvre.*
- *Facilité d'apprentissage : effort de l'utilisateur pour apprendre son utilisation.*
- *Facilité d'exploitation : effort que doit faire l'utilisateur pour exploiter et contrôler l'exploitation du logiciel.*

La qualité du logiciel

3. Fiabilité (Reliability)

C'est-à-dire la capacité d'un logiciel de rendre des résultats corrects et à maintenir un niveau de service quelles que soient les conditions d'exploitation .

En font partie:

- *Maturité : fréquence des défaillances dues à des défauts .*
- *Tolérance aux pannes : aptitude à maintenir un niveau de service donné en cas de défaut ou d'attaque.*
- *Possibilité de récupération : capacité à rétablir son niveau de service et de restaurer les données directement affectées en cas de défaillance ; temps et effort nécessaire pour le faire.*

La qualité du logiciel

3. *Fiabilité (Reliability)*

- *Correction, justesse, conformité : le logiciel est conforme a ses spéciations, les résultats sont ceux attendus*
- *Robustesse, sûreté : le logiciel fonctionne raisonnablement en toutes circonstances, rien de catastrophique ne peut survenir, même en dehors des conditions d'utilisation prévues*

Mesures :

- *MTBF : Mean Time Between Failures.*
- *Disponibilité (pourcentage du temps pendant lequel le système est utilisable)*
- *Taux d'erreur (nombre d'erreurs par KLOC Kilo Line Of Code)*

Solutions :

- *Utiliser des méthodes formelles, des langages et des méthodes de programmation de haut niveau*
- *Vérifications, tests*
- *Recourir aux Progiciels*

La qualité du logiciel

4. Le Rendement / Performance (Efficiency)

C'est le rapport entre la quantité de ressources utilisées (moyens matériels, temps, personnel), et la quantité de résultats délivrés.

En font partie:

- *Temps : Temps : temps de réponses et de traitement ; débits lors de l'exécution de sa fonction.*
- *Ressources : quantité de ressources utilisées ; durée de leur utilisation par fonction.*

La qualité du logiciel

5. Maintenabilité (Maintainability)

Elle porte sur l'effort nécessaire pour corriger ou transformer le logiciel.

En font partie:

- *Facilité d'analyse : effort nécessaire pour diagnostiquer les déficiences et leurs causes ou pour identifier les parties à modifier.*
- *Facilité de modification : effort nécessaire pour modifier, remédier aux défauts ou adapter à l'environnement.*
- *Stabilité : risque des effets inattendus des modifications.*
- *Facilité de test : effort pour valider le logiciel modifié.*

La qualité du logiciel

6. *Portabilité (Portability)*

c'est-à-dire l'aptitude d'un logiciel de fonctionner dans un environnement matériel ou logiciel différent de son environnement initial

En font partie:

- *Facilité d'adaptation : possibilité d'adaptation à différents environnements donnés sans que l'on ait recours à d'autres actions ou moyens que ceux prévus à cet effet par le logiciel.*
- *Facilité d'installation : effort nécessaire pour installer le logiciel dans un environnement donné.*
- *Conformité aux règles de portabilité : conformité aux normes et aux conventions ayant trait à la portabilité.*
- *Interchangeabilité : possibilité et effort d'utilisation du logiciel à la place d'un autre logiciel donné dans le même environnement.*

La qualité du logiciel

Il existe d'autres critères de la qualité:

Réutilisabilité: C'est la possibilité de réutiliser certains éléments produits lors du processus de développement d'un logiciel donné, dans le processus de développement d'un autre logiciel.

Le but de la réutilisation est d'obtenir des gains en terme de temps et de coûts.

La réutilisation peut concerner: des parties du code, des morceaux de modèles de conception, ou encore des parties de documents.

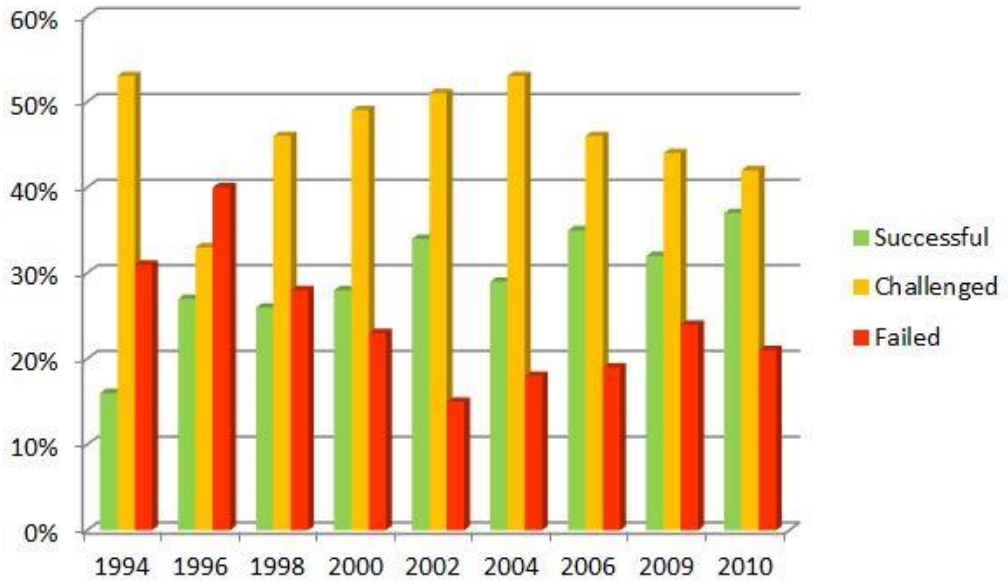
On estime que 80% du code est le même dans la plupart des logiciels, et que seul 20% sont spécifiques.

Solutions:
- *Abstraction, généricité (ex : Diagramme des classes Gestion des Comptes).*
- *Patrons de conception « Design Patterns »*
- *Ingénierie des composants*

Introduction

La qualité

Le tableau suivant donne les éléments depuis 1994 jusqu'à la dernière publication du Standish Group, le Chaos Manifesto 2011 (portant sur l'année 2010). http://alain.battandier.free.fr/spip.php?article55



	Successful	Challenged	Failed
1994	16%	53%	31%
1996	27%	33%	40%
1998	26%	46%	28%
2000	28%	49%	23%
2002	34%	51%	15%
2004	29%	53%	18%
2006	35%	46%	19%
2009	32%	44%	24%
2010	37%	42%	21%

Le génie logiciel tente d'apporter des réponses méthodologiques et des outils pour maitriser les processus de développement et garantir la qualité du logiciel.

Processus de Développement du Logiciel

CYCLE DE VIE

Le cycle de vie du logiciel

Notion générale de Cycle de Vie :

Un cycle de vie est un ensemble de phases séquentiellement cohérentes, dont le nom et le nombre de séquences sont déterminés en fonction du type et de la nature du produit ou du service à réaliser.

On distingue les principales phases de:

- *Création d'un produit ,*
- *Distribution sur un marché,*
- *Disparition.*

Le but du découpage est de :

- *Maîtriser les risques,*
- *Maîtriser au mieux les délais et les coûts,*
- *Obtenir une qualité conforme aux exigences.*

Le cycle de vie du logiciel

Le logiciel, en tant que produit, possède son propre « Cycle de Vie » (Software Lifecycle).

Le «cycle de vie du logiciel» désigne toutes les étapes du « processus de développement » d'un logiciel, depuis sa conception jusqu'à sa disparition.

Il comprend généralement les activités suivantes :

1. *Définition des objectifs, consistant à définir la finalité du projet et son inscription dans une stratégie globale.*

2. *Analyse des besoins et faisabilité, c'est-à-dire l'expression, le recueil et la formalisation des besoins du demandeur (le client) et de l'ensemble des contraintes.*
3. *Conception générale. Il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel.*

Le cycle de vie du logiciel

4. *Conception détaillée, consistant à définir précisément chaque sous-ensemble du logiciel.*

5. *Codage (Implémentation ou programmation), soit la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception.*

6. *Tests unitaires, permettant de vérifier individuellement que chaque sous-ensemble du logiciel est implémentée conformément aux spécifications.*

7. *Intégration, dont l'objectif est de s'assurer de l'interfaçage des différents éléments (modules) du logiciel. Elle fait l'objet de tests d'intégration consignés dans un document.*

Le cycle de vie du logiciel

8. *Qualification (ou recette), c'est-à-dire la vérification de la conformité du logiciel aux spécifications initiales.*

9. *Documentation, visant à produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs.*

10. *Mise en production, le logiciel est mis en service et est effectivement utilisé par les utilisateurs.*

11. *Maintenance, comprenant toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.*

Processus de Développement du Logiciel

Les Processus/ Modèles de développement

Les Processus de développement

Introduction:

C'est un ensemble structuré d'activités nécessaires pour développer un logiciel.

Un modèle de développement de logiciel est une représentation abstraite d'un processus

Il existe de nombreux modèles de développement; ils comportent tous, en général, les activités suivantes:

- *Spécification / Analyse*
- *Conception.*
- *Programmation / Implémentation.*
- *Test / Validation.*
- *Maintenance / Evolution*

Les Processus de développement

Introduction:

La description d'un modèle de développement nécessite:

- *La description des activités.*
- *La description des produits résultats des activités.*
- *Les rôles, qui reflètent les responsabilités des personnes impliquées dans le processus.*
- *Les pré- et post-conditions, qui sont des conditions vraies avant et après l'activité d'un processus.*

Le Processus de développement

Les étapes générales

Dans un projet de développement « bien conduit », l'effort se répartit comme suit :

- *Programmation : 15 à 20 %.*
- *Spécification et conception : 40 %.*
- *Validation et vérification: 40 % .*

Le Processus de développement

Les étapes générales

Il existe différents modèles de processus de développement, qui se distinguent par la manière dont les étapes s'enchainent.

Il y'a lieu de distinguer entre 3 principaux modèles de cycle de vie:

- *Cascade appelé aussi séquentiel, Waterfall Model (Bohem 1976).*
- *en V (Golberg).*
- *de la Spirale, appelé aussi incrémental ou itératif;*

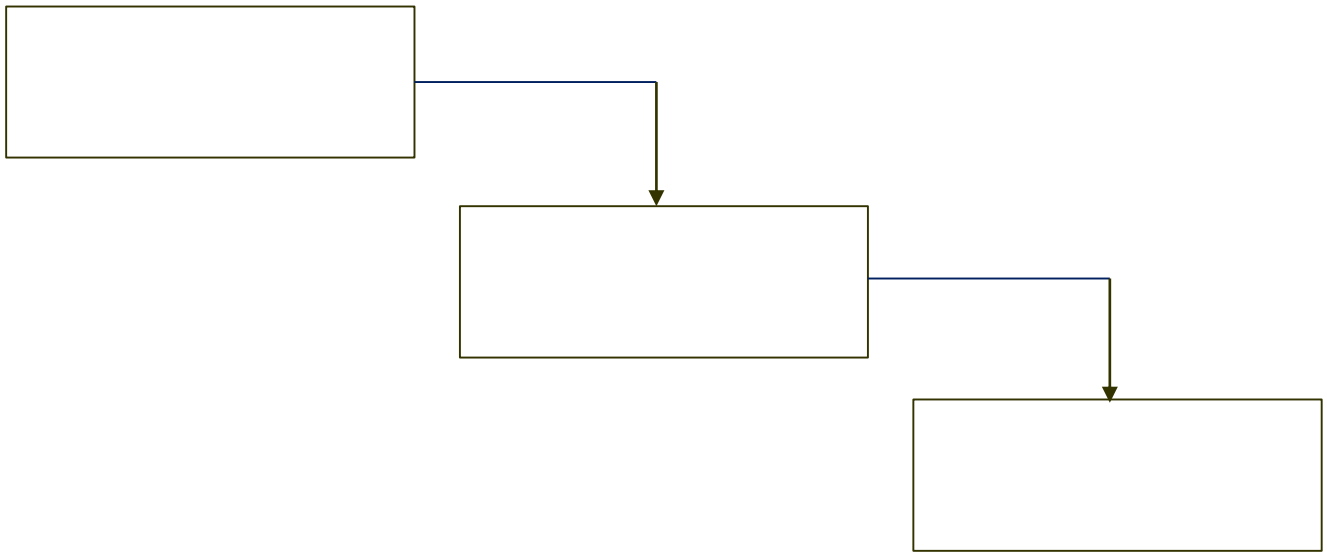
Ces modèles se différencient essentiellement par la manière avec laquelle s'enchaînent les différentes étapes.

Le Processus de développement

Le modèle de la Cascade: Water fall

Principes du modèle de la cascade :

- *Le développement du logiciel est considéré comme une succession d'étapes réalisées de façon strictement séquentielle.*



Le Processus de développement

Le modèle de la Cascade: Water Flow

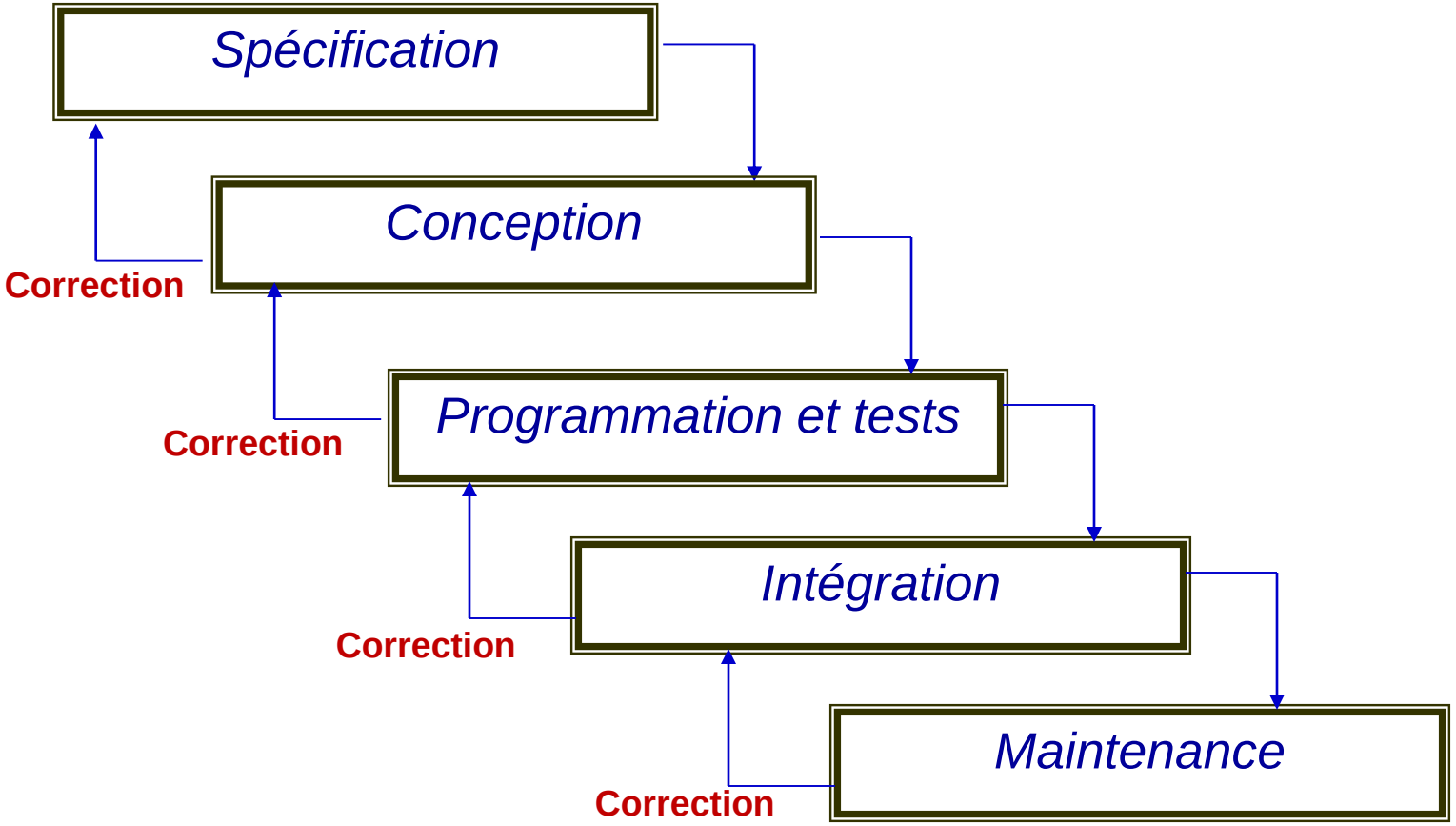
Principes du modèle :

- *Chaque étape correspond à une activité de base qui s'appuie sur les résultats de l'étape qui précède.*
- *Chaque étape se termine par un résultat (un livrable).*
- *Les résultats d'une étape sont soumis à une revue approfondie, et on ne passe en "principe" à l'étape suivante, que lorsqu' ils sont jugés satisfaisants.*
- *Il n'y a pas (ou peu) de retours en arrière.*

Le Processus de développement

Le modèle de la Cascade: Water Flow

Version initiale attribuée à Bohem en 1976



Le Processus de développement

Le modèle de la Cascade: Water Flow

Avantages du modèle :

- *Facile à comprendre.*
- *Force la documentation. Chaque étape donne lieu à un livrable.*
- *Chaque étape comporte ses propres tests.*

Le Processus de développement

Le modèle de la Cascade: Water Flow

Limites du modèle:

- *Pas de modification possible des besoins, après le démarrage du projet.*
- *Le coût de correction des erreurs est fort élevé, en particulier si elles sont découvertes tardivement.*
- *Des erreurs constatées dans les résultats d'une étape remettront en cause toutes les étapes qui la suivent : les erreurs se propagent du haut (début) vers le bas (la fin).*
- *Chaque étape fait appel à des compétences différentes et donc à des personnes différentes, ce qui augmente les risques d'incohérence.*

Le Processus de développement

Le modèle de la Cascade: Water Flow

Limites du modèle :

- *Ne convient qu'aux applications intelligibles dont les résultats sont prévisibles de l'analyse à la conception.*

- *Ne convient pas aux applications comportant une part substantielle d'incertitude au niveau des exigences.*

- *Ne permet pas d'obtenir un système utilisable avant la fin complète du processus.*

Le Processus de développement

Le modèle en V: Les principes

Le modèle en V repose sur deux principes fondamentaux:

- *Une décomposition du logiciel en petits morceaux appelés : modules ou composant. Chaque module couvre une fonction du logiciel.*

- *Une Validation systématique de chaque étape du processus.*

Le Processus de développement

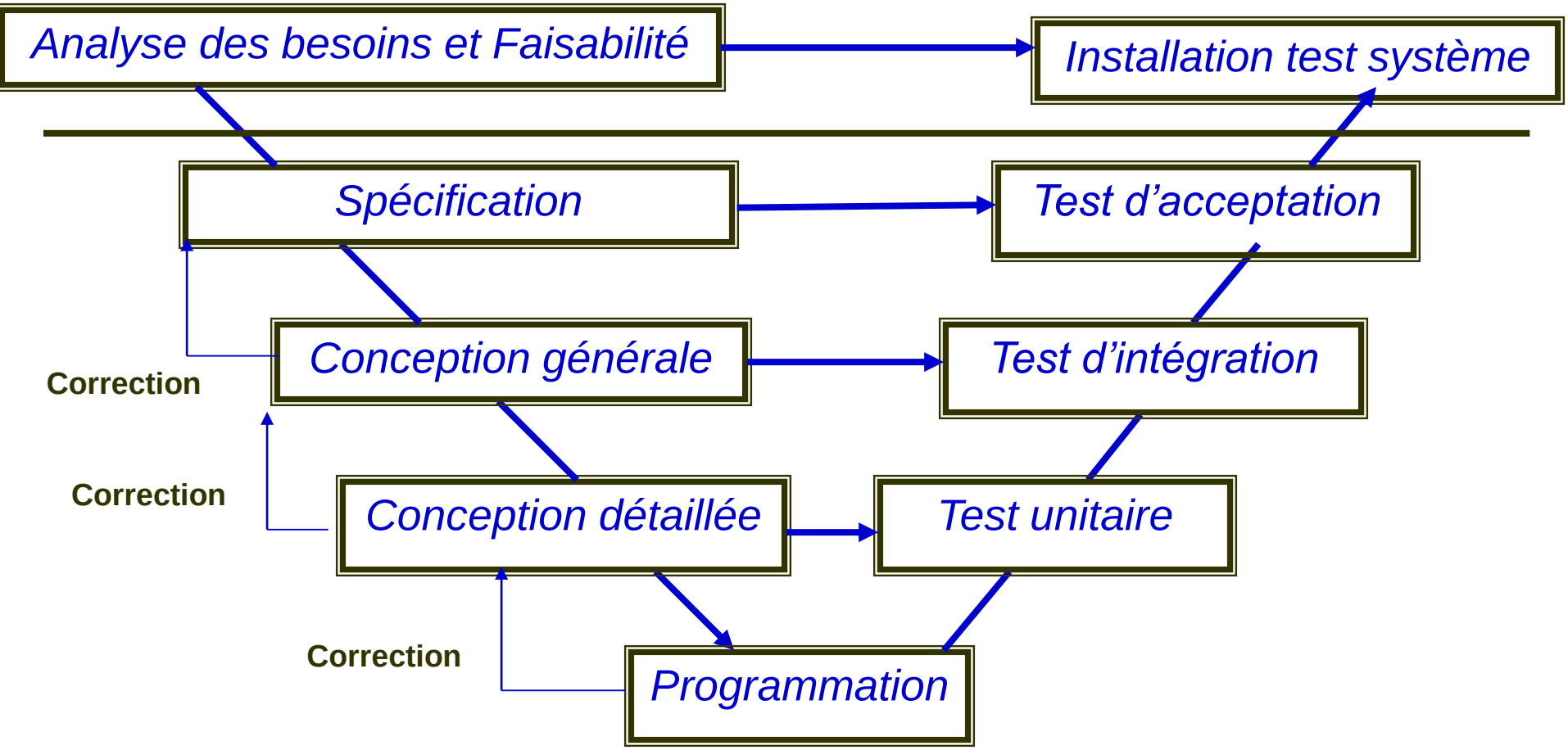
Le modèle en V: Les principes

Ainsi, Chaque composant (ou module) du logiciel, issu de la décomposition, est produit en enchaînant les étapes selon deux axes :

- *L'axe suivant le V : enchaînement selon le modèle de la cascade; mais appliqué au module.*
- *L'axe transversal : Chaque étape est validée avant de passer à la suivante.*

Le Processus de développement

Le modèle en V: Les principes



Le Processus de développement

Le modèle en V

Les points forts du modèle :

- *Il permet d'identifier et d'anticiper très tôt les éventuelles évolutions des besoins.*
- *C'est aussi un moyen de vérifier de la maturité des utilisateurs, car s'il en était autrement, ils se trouveraient dans l'incapacité de fournir des test de recettes dès la phase de spécification.*
- *C'est un modèle avantageux pour une maîtrise d'oeuvre, rassurant pour une maîtrise d'ouvrage qui doit cependant s'engager significativement.*
- *Bien adapté aux logiciels de complexité moyenne.*

Le Processus de développement

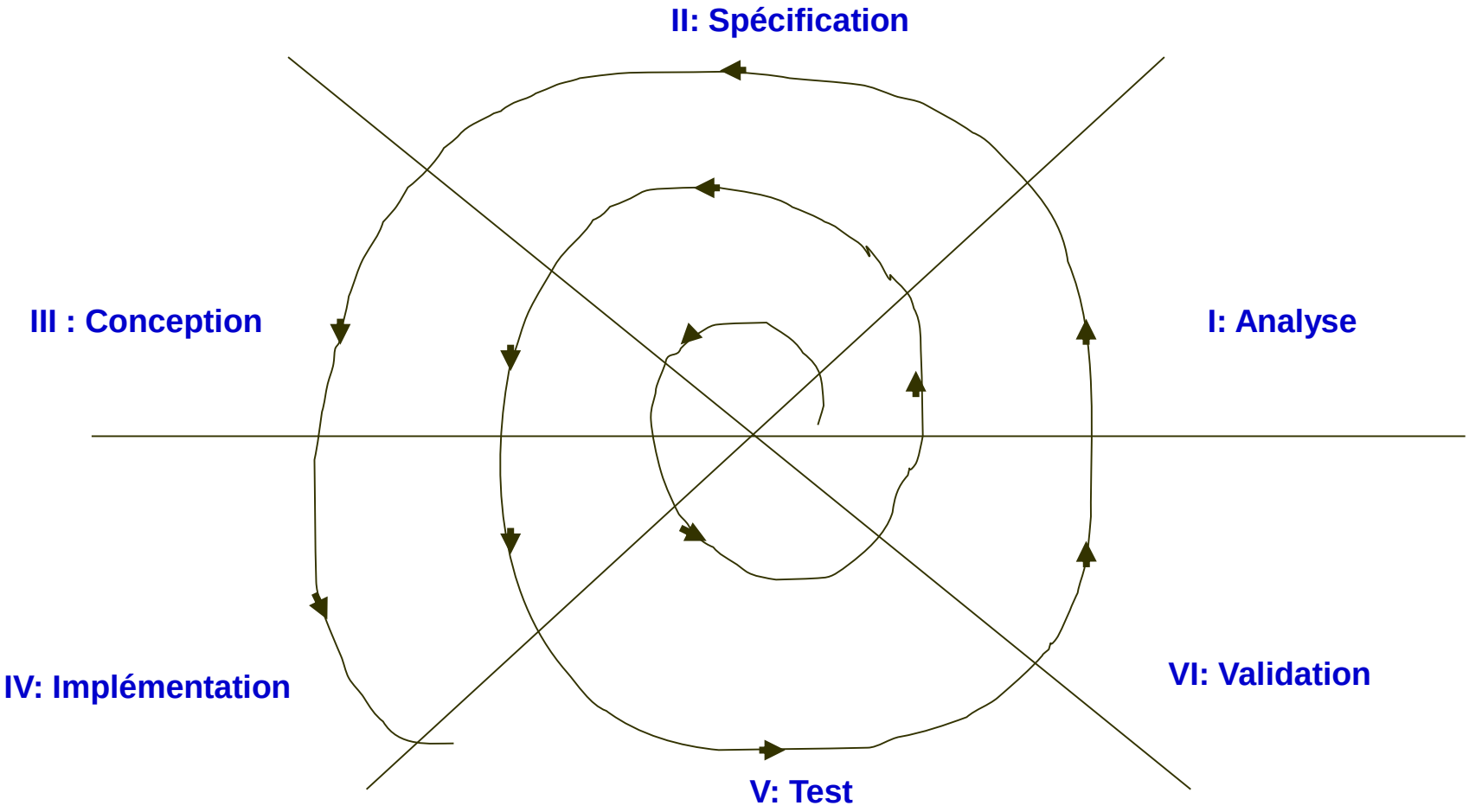
Le modèle en Spirale (Itératif)

Principes du modèle:

- *Le logiciel est produit par une succession d'itérations.*
- *Chaque itération étend le résultat de l'itération précédente.*
- *Chaque itération consiste à produire un logiciel (un prototype) à partir du logiciel produit lors de la précédente itération.*
- *Chaque itération se déroule selon les étapes des modèles en Cascade ou en V.*

Le Processus de développement

Le modèle en Spirale (Itératif)



Le Processus de développement

Le modèle en Spirale (Itératif)

Portée des itérations:

- *Le nombre et la durée des itérations dépendent de la taille du projet.*
- *La durée des itérations est de deux à quatre semaines dans le cas d'un projet court de moins de 6 mois.*
- *Elle peut s'étaler de 3 à quatre mois pour des projets s'étalant sur plusieurs années.*
- *Si les itérations sont trop courtes , leur coût sera trop élevé.*
- *Si les itérations sont trop longues , les points de contrôle seront trop espacés, les erreurs commises en début d'une itération seront détectées tardivement, et leur correction sera couteuse.*

Le Processus de développement

Le modèle en Spirale (Itératif)

Déroulement des itérations:

On distingue cinq phases dans le déroulement de chaque itération (cycle de la spirale) :

- *Analyse: Détermination des objectifs, des alternatives et des contraintes ;*
- *Spécification : Modèle des Besoins, Analyse des risques, évaluation des alternatives ;*
- *Conception : Modèle du Prototype ;*
- *Développement du prototype;*
- *Revue et vérification des résultats (Prototypes) d'une itération ou du cycle.*

Le Processus de développement

Le modèle en Spirale (Itératif)

Il y'a lieu pour chaque itération de :

- *Définir sa portée : estimer sa durée, la quantité de travail minimum pour une progression pertinente .*
- *Construire dès le départ des parties vitales ainsi que les fragments de code fréquemment exécutés par l'application ;*
- *Veiller à privilégier le développement des fragments significatifs du logiciel;*

Le Processus de développement

Le modèle en Spirale (Itératif)

Chaque itération doit proposer au minimum:

- *Un retour sur investissement. Valeur ajoutée – Cout > minimum ;*
- *Une nouvelle fonctionnalité ajoutée;*
- *Une interaction utilisateur améliorée;*
- *Une meilleure efficacité;*
- *Une fiabilité accrue;*
- *Une infrastructure renforcée pour la maintenance et les itérations futures;*

Le Processus de développement

Le modèle en Spirale (Itératif)

Les cas d'utilisation constituent une excellente base pour l'attribution des ressources et la planification des itérations:

- *Chaque itération peut se concentrer sur quelques cas d'utilisation;*
- *Un cas d'utilisation peut aussi s'étendre sur plusieurs itérations;*

Ainsi :

« Une fonctionnalité de base d'un Cas d'Utilisation peut être implémentée lors d'une itération, alors qu'une fonctionnalité plus avancée sera traitée lors d'une deuxième itération, et la gestion des erreurs dans une troisième itération ».

Le Processus de développement

Le modèle en Spirale (Itératif)

Règles de mise en œuvre du modèle:

- *Les priorités sont à établir de manière réfléchie, selon leur degré d'importance;*
- *Eviter le syndrome du « tout est d'importance égale » : Lorsqu'un élément est considéré comme important, il faut réduire l'importance d'un autre élément.*
- *Eviter de rendre le résultat de chaque itération au client; veiller à rester dans les délais et à s'assurer que les différentes composantes de l'application concordent.*
- *Chaque itération doit commencer avec un référentiel commun et se terminer par un nouveau référentiel commun.*
- *Les développeurs doivent intégrer toutes les versions des artefacts d'un système et les vérifier à la fin de l'itération..*

Le Processus de développement

Le modèle en Spirale (Itératif)

Règles de mise en œuvre du modèle:

- *Toute l'équipe doit travailler sur avec le même jeu d'hypothèses et une version actualisée du système; le respect de cette règle est vital pour la réussite du projet.*

- *Chaque itération doit générer une version exécutable, qu'il faut tester et intégrer au système obtenu à l'itération précédente.*

- *Chaque itération doit être planifiée de telle sorte qu'il y'ait suffisamment de temps pour réaliser toutes les étapes du développement, à savoir: l'analyse , la conception, l'implémentation, le test et l'intégration. Le modèle de la cascade est donc applicable à petite échelle (celui d'une itération).*

Le Processus de développement

Le modèle en Spirale (Itératif)

Application du modèle itératif à la modélisation:

La modélisation par itération permet de découvrir très tôt les problèmes du logiciel. Philosophie du « fail fast » (« Echouer très tôt » « mieux vaut tôt que tard »)

- *Commencer par construire un modèle par le recueil des exigences,*
- *Revoir le modèle après une première itération,*
- *Intercaler modélisation et développement, au fur et à mesure.*

Le Processus de développement

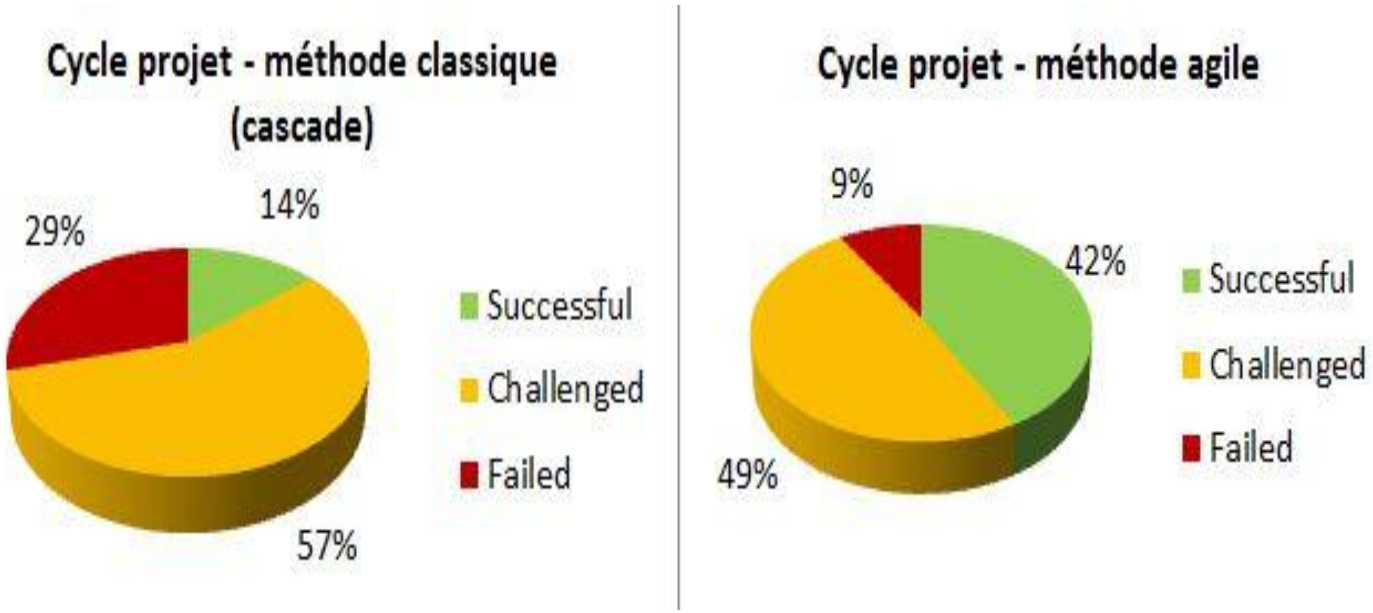
Le modèle en Spirale (Itératif)

Points forts :

- *Adapté aux projets très complexes dont les besoins ne peuvent être déterminées de façon précise à l'avance, ou bien qui sont très évolutifs.*
- *Les besoins sont exprimés et pris en compte au fur et à mesure que les prototypes sont développés et testés.*
- *Permet de découvrir les écueils et les erreurs dès les toutes premières étapes du développement, grâce aux multiples contrôles.*
- *Le système en construction est plus aisé à modifier en cas d'erreur*
- *Les utilisateurs s'expriment plus aisément au sujet d'un produit opérationnel (le prototype), que face à des descriptions formelles (Cahier des Charges).*
- *Permet une meilleure maitrise du projet et une réduction des risques*

Le Processus de développement

Choix d'un modèle



Le Processus de développement

Choix d'un modèle

Le choix d'une modèle de développement est une décision qui relève de la maîtrise d'œuvre.

Ce choix dépend de plusieurs paramètres :

- *Taille du domaine.*
- *Complexité du domaine.*
- *Degré de stabilité des besoins*
- *Expérience antérieure des utilisateurs dans les projets de développement de logiciels .*

Le Processus de développement

Choix d'un modèle

Modèle en Cascade:

Points forts

- *Facile à comprendre et à utiliser*
- *Adapté pour une équipe inexpérimentée*
- *Les limites de chaque étape sont visibles*
- *Facilite un management du projet*
- *La définition des besoins est non-évolutive*
- *La qualité prime sur le coût*

Points faibles

- *Tous les besoins doivent être bien spécifiés au départ*
- *Donne une fausse impression de l'avancée des travaux*
- *Pas d'interaction entre les phases de développement*
- *L'intégration n'a lieu qu'à la fin du cycle*
- *Le client peut se retrouver non satisfait*
- *Pas de retour en arrière d'une phase à l'autre*

Quand l'utiliser ?

- *La phase de spécification a été très bien faite*
- *La définition du produit est stable*
- *Il s'agit d'une nouvelle version d'un produit existant*
- *L'implantation d'un produit existant sur une nouvelle plate-forme*
- *Une bonne maîtrise de la technologie*

Le Processus de développement

Choix d'un modèle

Modèle en V:

Points forts

- Facile à utiliser
- Les tests sont effectués à chaque étape
- Le contrôle se fait progressivement à chaque étape
- Les phases de validation sont prises en main très tôt dans le processus de développement

Points faibles

- Une mauvaise prise en compte des évènements concurrents
- Le processus n'est pas itératif
- Une mauvaise prise en compte des changements de la spécification des besoins
- Ne contient pas les activités d'analyses de risques

Quand l'utiliser ?

- Les spécifications de besoins doivent être bien faites
- La solution à développer et la technologie à utiliser doivent être parfaitement connue
- Les changements doivent être faits avant l'analyse
- Excellent pour les systèmes requérant une grande sûreté

Le Processus de développement

Choix d'un modèle

Modèle en Spirale:

Points forts

- *Sans coût élevé, donne des indications sur les risques majeurs*
- *Les fonctions critiques à haut risque sont développées en premier lieu*
- *La conception ne doit pas forcément être terminée*
- *Les utilisateurs finaux sont intimement associés à toutes les étapes du développement.*
- *Le développement se fait en interaction avec les clients*
- *L'évolution du coût de développement est sous contrôle*
- *Les utilisateurs ont dès le départ une vue globale du système*

Points faibles

- *Le temps consacré à l'évaluation des risques est trop élevé pour des petits projet*
- *Le temps mis à planifier, évaluer les risques, fixer les objectifs, les prototypes peut être excessif*
- *Ce modèle est complexe*
- *Une expertise en évaluation des risques est nécessaire*
- *La spirale peut être infinie*
- *les développeurs travaillent par intermittence*
- *il est difficile de définir les objectifs et les points de validation intermédiaires entre les différentes étapes*

Le Processus de développement

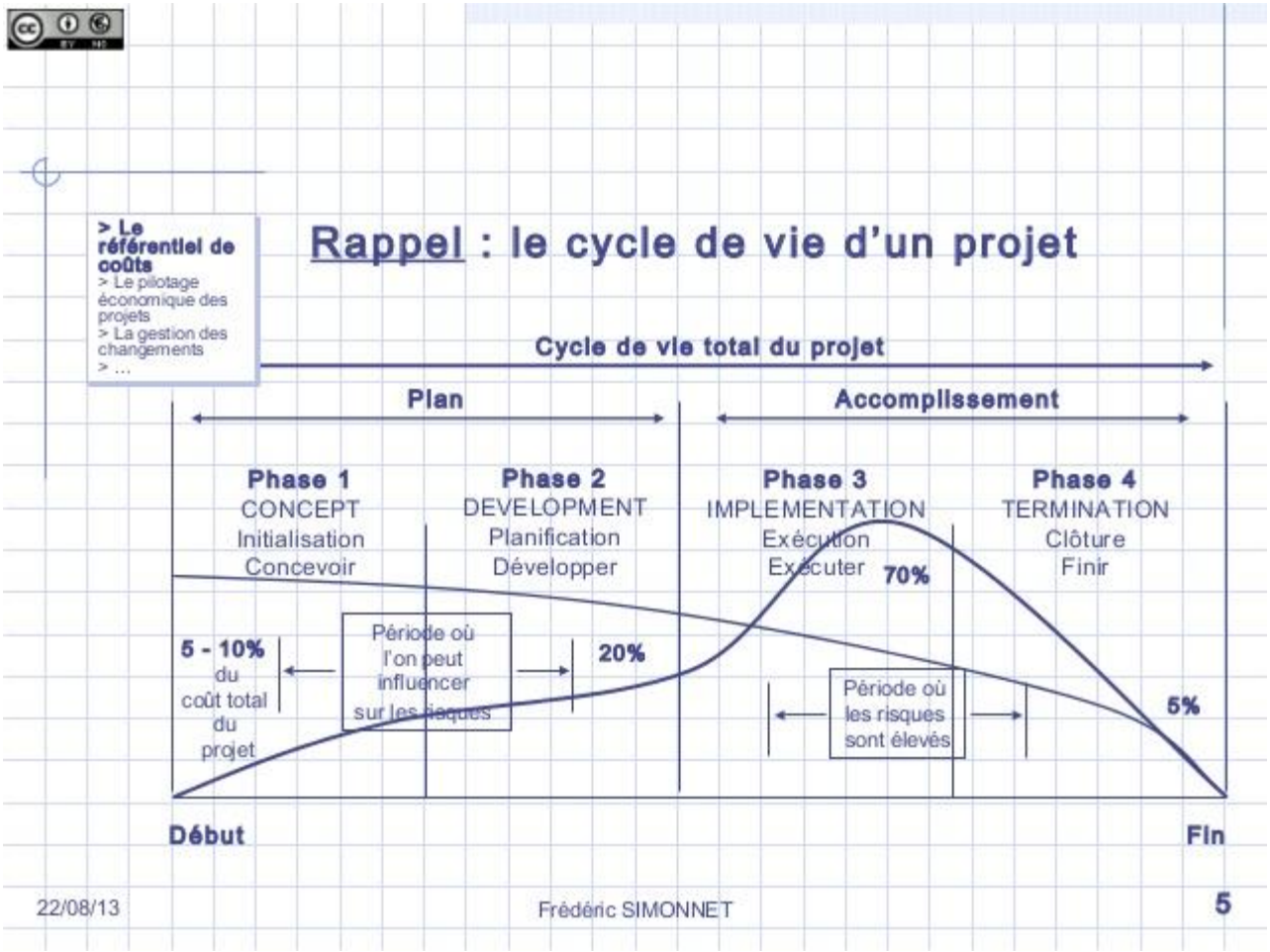
Choix d'un modèle

Modèle en Spirale:

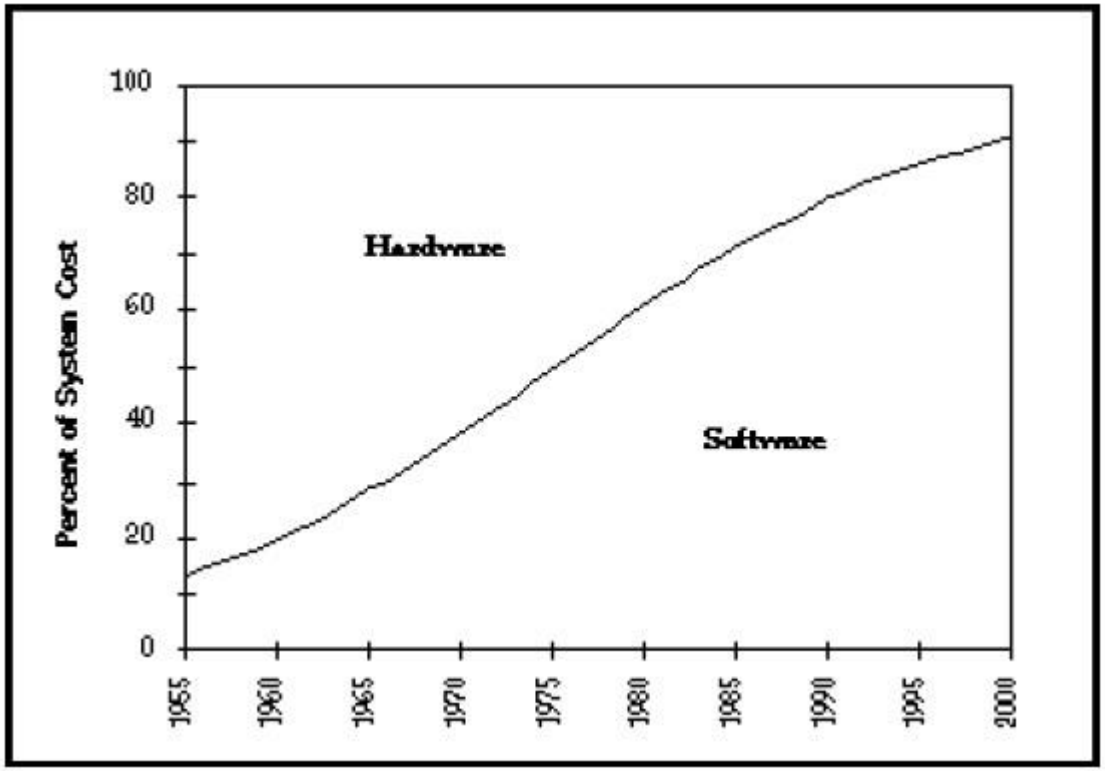
Quand l'utiliser ?

- *les coûts et l'évaluation des risques est important*
- *pour des projets à risque au moins moyennement élevé*
- *pour des projets à long terme dont les financements peuvent varier*
- *les utilisateurs ne définissent pas clairement leurs besoins*
- *la spécification des besoins est complexe*
- *il s'agit d'une nouvelle gamme de produits*
- *des changements significatifs peuvent intervenir à cause par exemple de l'évolution de la recherche ou de l'exploration*

Introduction

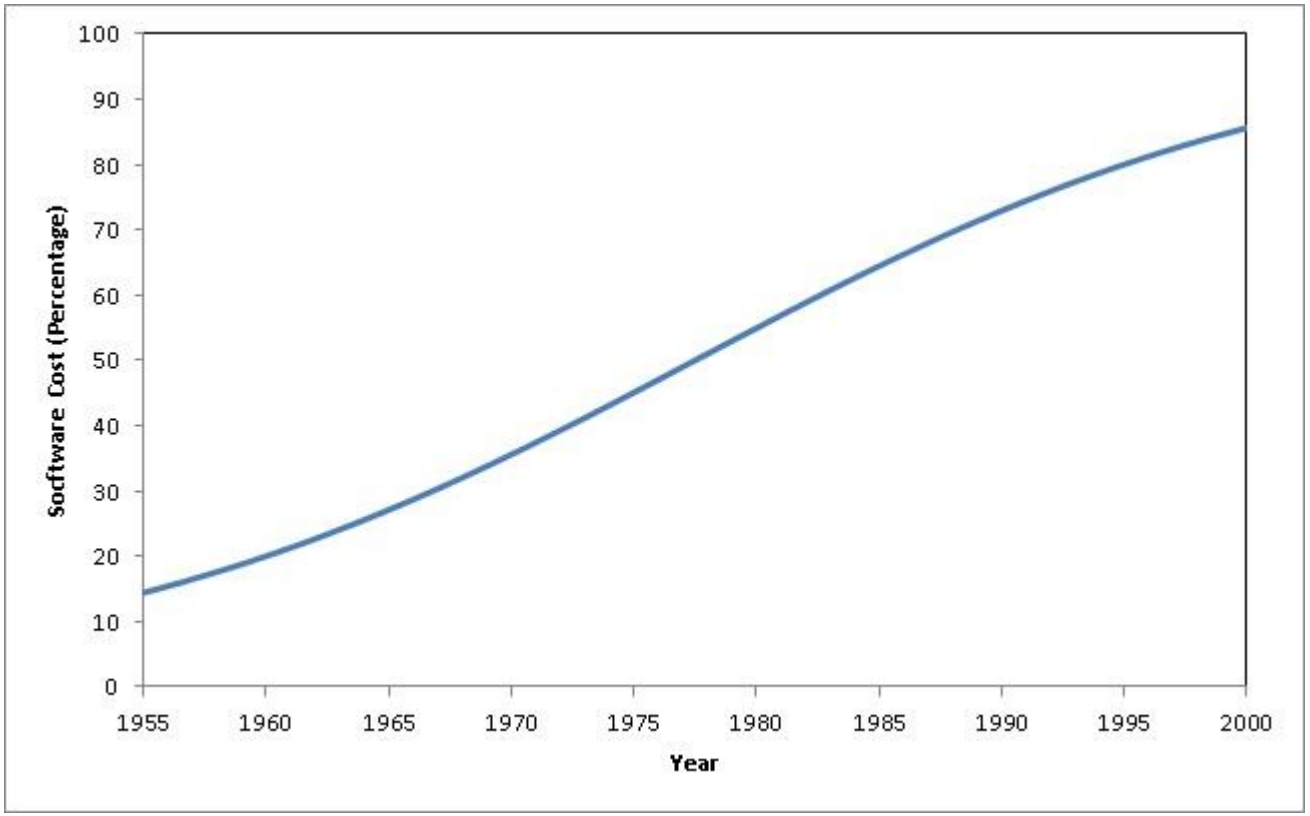


Introduction



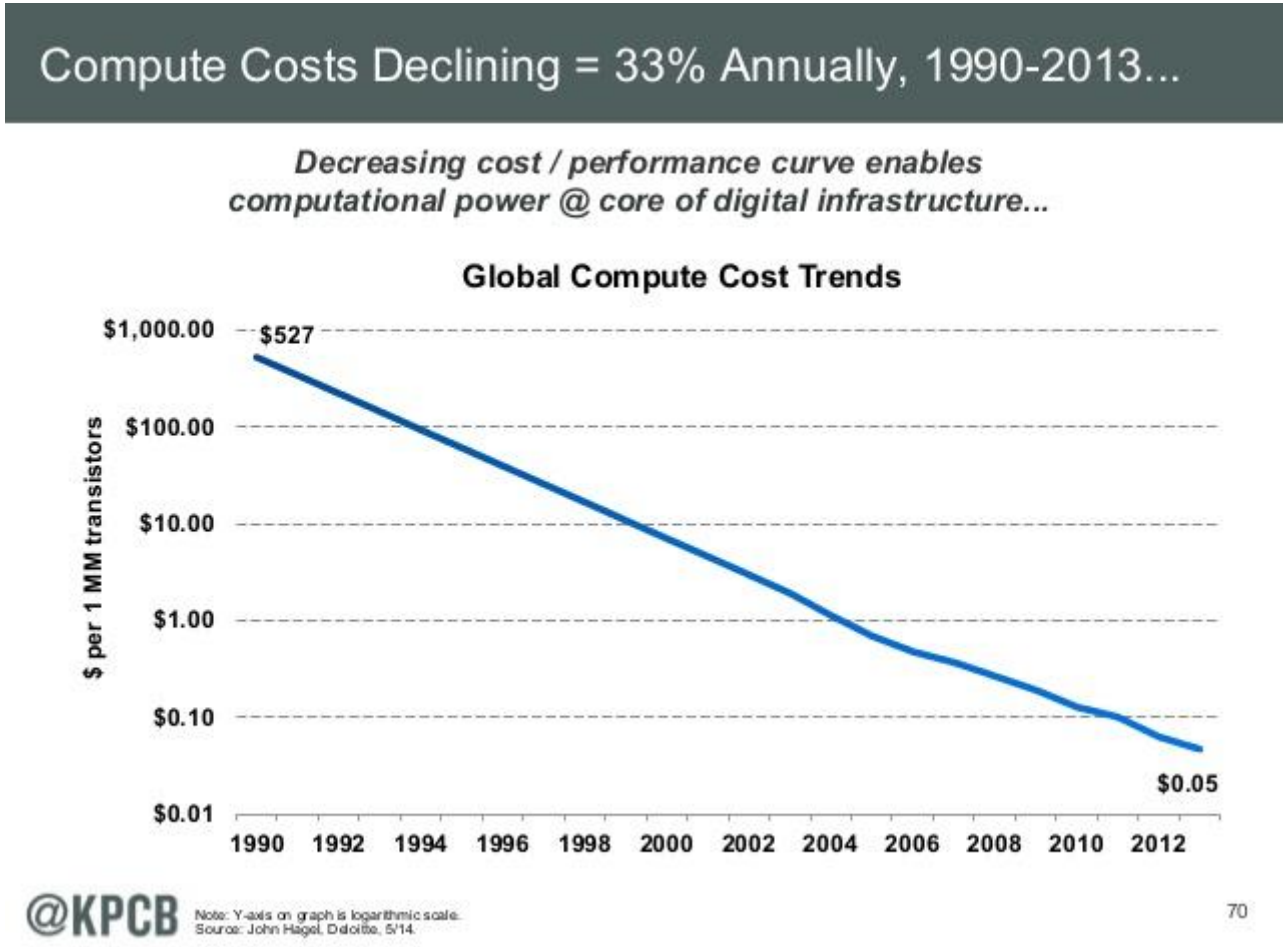
Alors que, la seconde est : *Immatérielle et intangible.*

Introduction



Alors que, la seconde est : *Immatérielle et intangible.*

Introduction



Alors que, la seconde est : *Immatérielle et intangible.*

Les cycles du développement logiciel

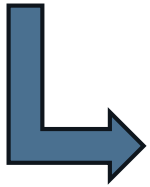
Bailly Gabriel, Bacchiocchi Julien, Moly Corentin ,Verny Thiago et Glasson Emma

Sommaire

1. Introduction
2. Le cycle en cascade
3. Le cycle en V
4. Le cycle itératif
5. Le cycle en spirale
6. Les méthodes agiles
7. Conclusion

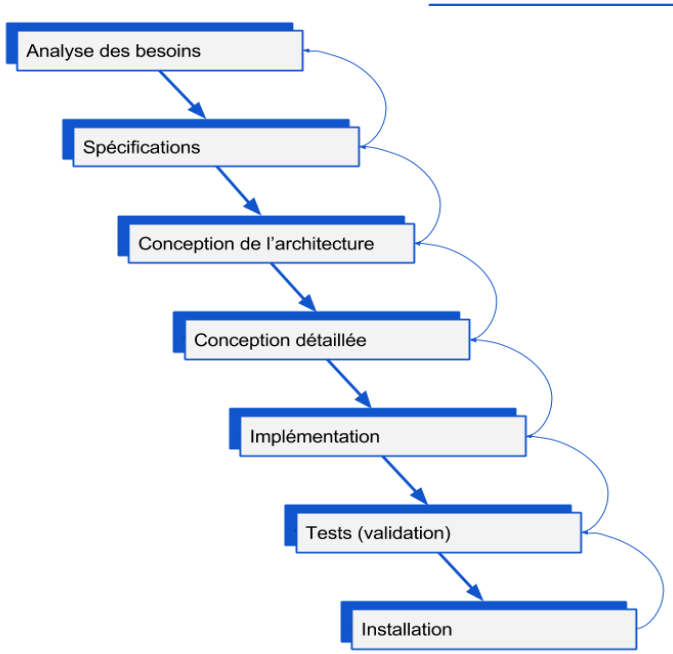
Introduction

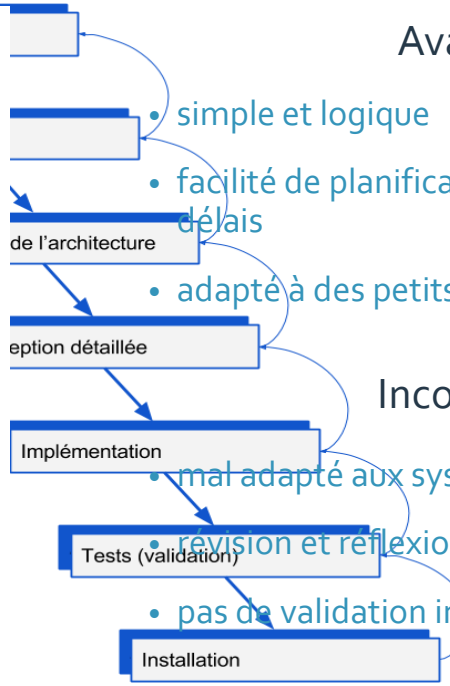
Qu'est ce que le processus de développement d'un logiciel?



un ensemble structuré d'activités nécessaires pour développer un logiciel

Le cycle en cascade





http://www.responsive-mind.fr/cycles-developpement-informatique/ *publié le 2/03/2015*

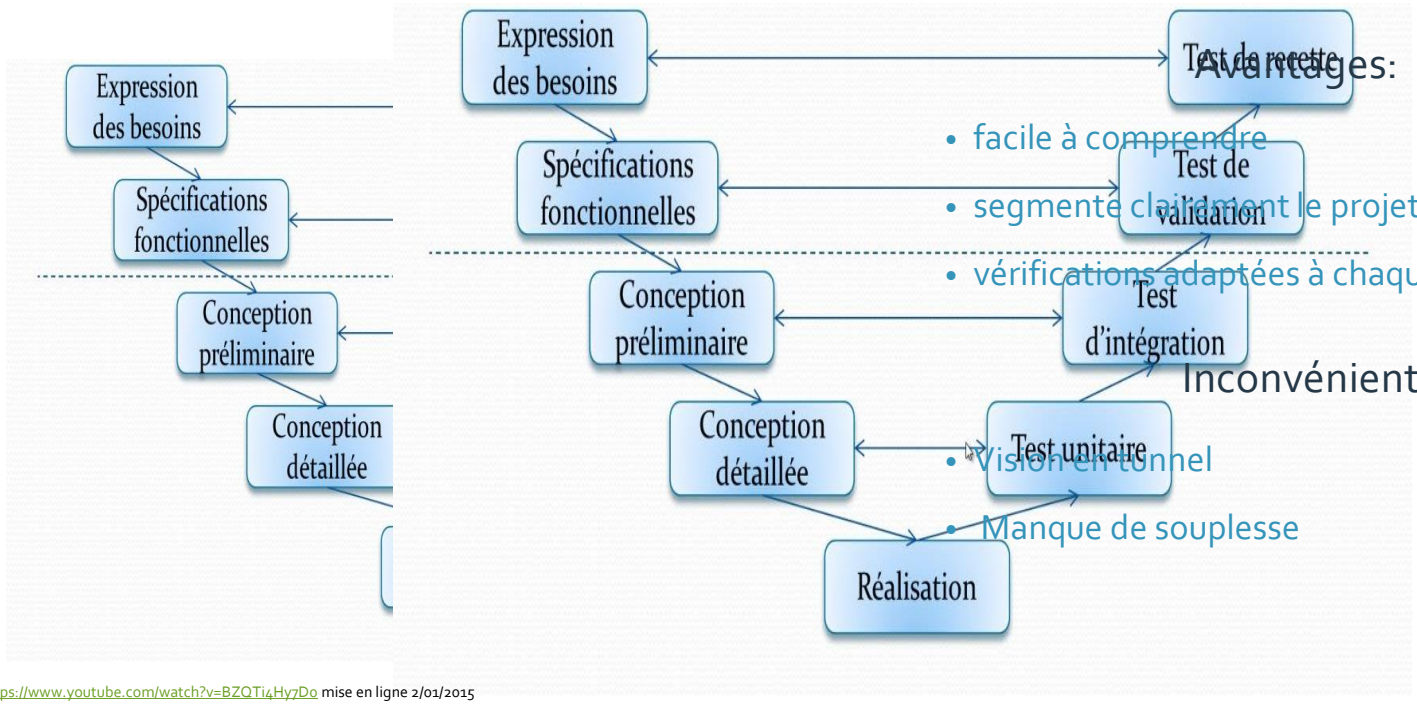
Avantages:

- simple et logique
- facilité de planification des étapes et des délais
- adapté à des petits systèmes

Inconvénients:

- mal adapté aux systèmes complexes
- révision et réflexion quasi impossible
- pas de validation intermédiaire

Le cycle en V



Avantages:

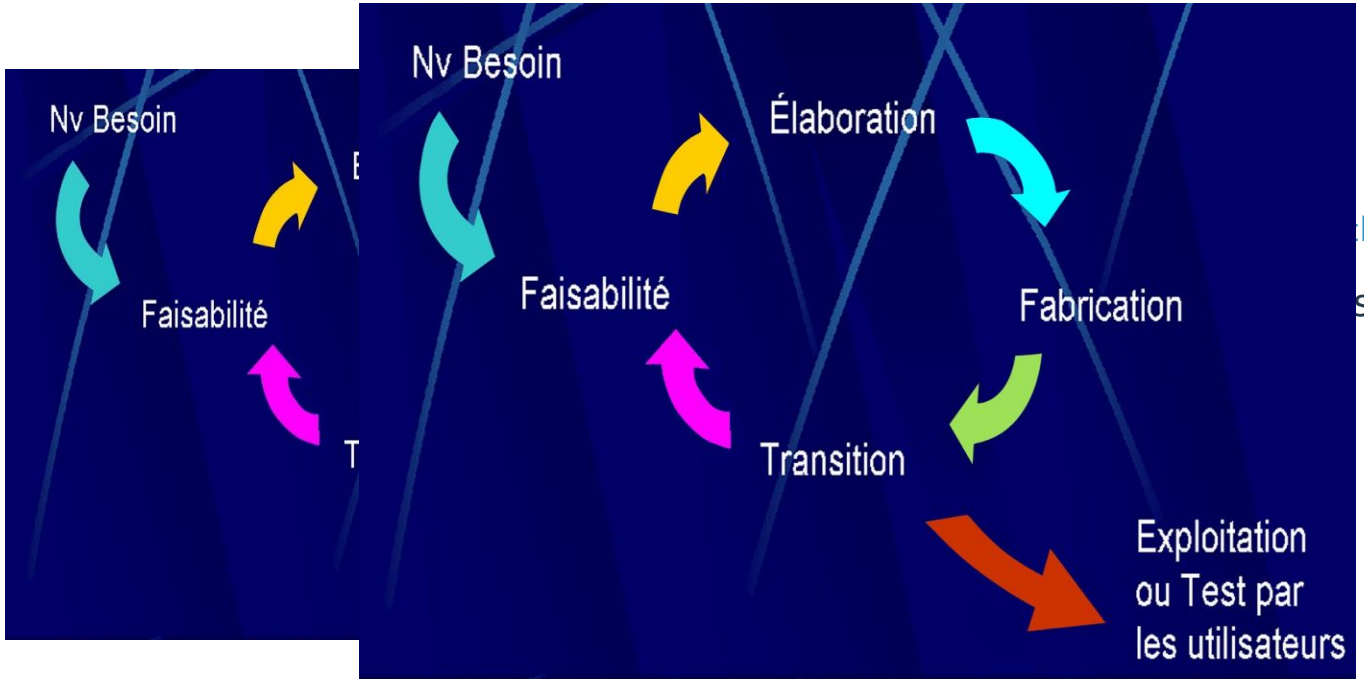
- facile à comprendre
- segmente clairement le projet
- vérifications adaptées à chaque étape

Inconvénients:

- Vision en tunnel
- Manque de souplesse

https://www.youtube.com/watch?v=BZQTi4Hy7Do mise en ligne 2/01/2015

Le cycle itératif



haque cycle

s:

intégrées

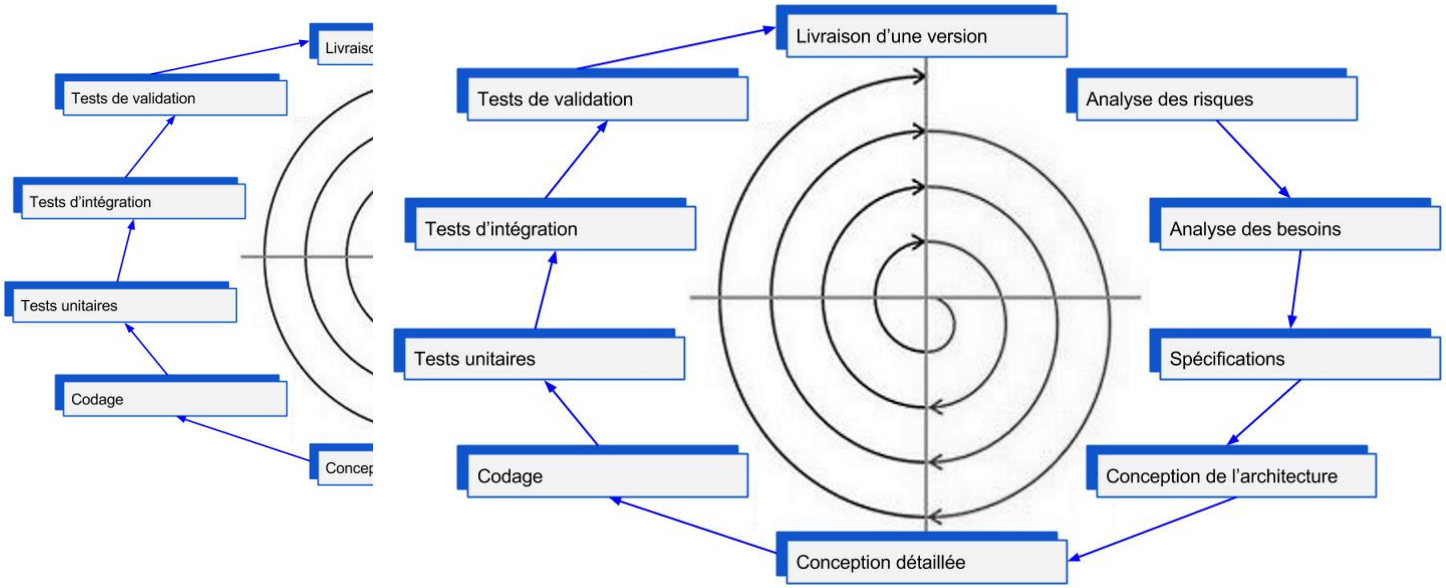
https://commons.wikimedia.org/wiki/File:Mod%C3%A8le_It%C3%Agratif.PNG - publié 9 avril 2005

Le cycle en spirale

Avantages

acile

jet coûteux



http://www.responsive-mind.fr/wp-content/uploads/2015/03/cycle-en-spirale2.png

Les méthodes agiles

12 principes généraux

- Satisfaction du client
- Acceptation de demandes de changement
- Livraison régulière de versions opérationnelles du produit
- Coopération client/équipe
- Des individus motivés
- Conversation face à face
- Mesure de l'avancement via les fonctionnalités du produit
- Un rythme soutenable et continu
- Excellence technique et conception surveillées
- Faire simple
- Responsabilisation des équipes
- Ajustement des processus pour plus d'efficacité

4 valeurs:

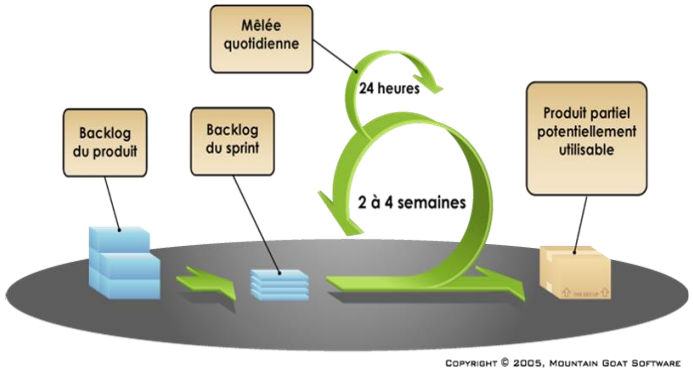
- L'équipe
- Un logiciel qui fonctionne
- La collaboration
- L'acceptation du changement

Inconvénients

- difficile d'établir un contrat
- difficile de limiter les risques
- compliqué de progresser sur le rôle du client et du testeur
- très différents des autres modèle donc demande des efforts importants

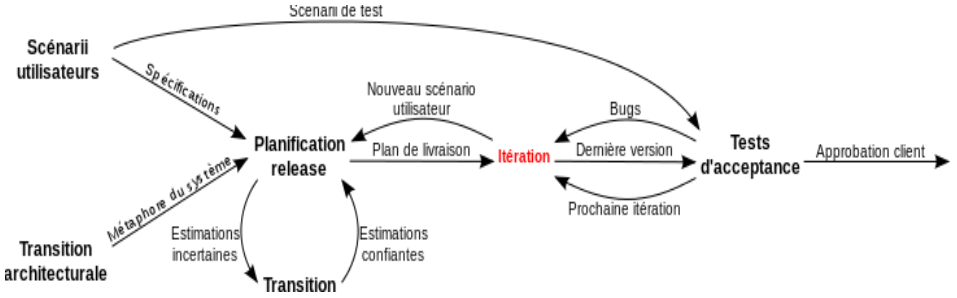
Les méthodes agiles

Méthode SCRUM



http://igm.univ-mlv.fr/~dr/XPOSE2008/SCRUM/presentation.php

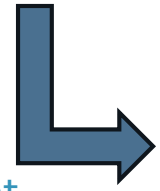
Méthode XP



http://theses.ulaval.ca/archimede/fichiers/24937/ch04.html de 2007

Conclusion

Quel est le meilleur modèle ?



Pas de modèle idéal, tout dépend des caractéristiques de votre projet

Sitographie

Introduction

- http://membres-lig.imag.fr/dubousquet/docs/2.2_CyclesDeVie.pdf
- http://perso.univ-st-etienne.fr/jacquene/gl/cours/partie2.pdf

Le cycle en cascade

- http://www.lemagit.fr/definition/Modele-en-cascade-Waterfall
- http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/
- http://www.responsive-mind.fr/cycles-developpement-informatique/

Le cycle en V

- https://www.pentalog.fr/notre_demarche/methodologie_cycle_en_v.htm
- http://patjo82.over-blog.com/article-etude-comparee-des-differents-cycles-de-vie-de-logiciels-111005786.html

Le cycle itératif

http://www.responsive-mind.fr/cycles-developpement-informatique/

Sitographie

Le cycle en spirale

- *Différents modèles de cycle de vie*, Anne-Marie Hugues (Décembre 20002) http://users.polytech.unice.fr/~hugues/GL/chapitre2.pdf
- *Avantages et inconvénients du modèle en spirale*, Keegan Larson (Octobre 2012)
- http://portableoccasionordinateur.blogspot.fr/2012/10/avantages-et-inconvenients-du-modele-en.html
- *Les différents cycles de développement en informatique*, Renaud Mariage Gaudron
- (Mars 2015) http://www.responsive-mind.fr/cycles-developpement-informatique

Les méthodes agiles

- http://medina.developpez.com/cours/extreme-programming/
- http://methodesagiles.info/methode_Agile.php
- https://fr.wikipedia.org/wiki/M%C3%A9thode_agile
- https://fr.wikipedia.org/wiki/Manifeste_agile
- http://www.agiliste.fr/introduction-methodes-agiles/

Bibliographie

- *Gestion de projets informatiques 1*, M.A. KHALDI (2016-2017)

L’intégration des applications d’entreprise.

Par L.Kzaz Janvier 2008

Le concept EAI :Entreprise Application Intégration

Problématique.

- *Les entreprises se sont équipées au fur et à mesure, de solutions logicielles issus de différentes stratégies et sources :*

  - *Solutions spécifiques développées soit en interne soit par des prestataires externes.*

  - *Progiciels de gestion standards issus d'un ou de plusieurs éditeurs.*

  - *Progiciel de gestion intégrée couvrant un certain nombre de fonctions de base.*

Le concept EAI :Entreprise Application Intégration

Problématique.

- *Cet ensemble d'applicatifs constitue un patrimoine de l'entreprise, qu'il est souvent coûteux d'abandonner au profit de nouvelles solutions.*

- *L'intégration des applications d'entreprise vise à remédier à certains problèmes des SI construits à l'aide d'applications hétérogènes.*

- *l'EAI représente "un ensemble d'outils, de progiciels d'intégration qui offrent des moyens de connecter et de faire communiquer entre elles les applications de l'entreprises, existantes ou à venir".*

Le concept EAI :Entreprise Application Intégration

Problématique.

- *Les objectifs d'une telle intégration sont les suivants :*

  - *Conserver le patrimoine applicatif de l'entreprise ;*

  - *Disposer des avantages des solutions intégrées ;*

  - *Présenter à l'utilisateur final une vision unifiée de l'information gérée par les différentes applications afin de l'aider dans ses prises de décision ;*

  - *Résoudre le problème de la cohérence entre des systèmes qui s'entrecroisent.*

Le concept EAI :Entreprise Application Intégration

Problématique.

- *L'EAI vise à " lier " de façon optimale des applications et des systèmes d'entreprise afin de :*

  - Constituer un support efficace aux processus de l'entreprise (processus manufacturiers, processus économiques, etc.),

  - Permettre à l'entreprise de répondre aux changements du marché;

Le concept EAI :Entreprise Application Intégration

Problématique.

•L'American Management Systems propose un cadre permettant aux entreprises de valider le degré d'intégration de leurs applications.

• Ce modèle a pour but de guider les entreprises dans les étapes conduisant à la gestion optimale de leurs capacités d'intégration d'applications en vue de satisfaire les objectifs de l'entreprise (économiques, technologiques, etc.).

Le concept EAI :Entreprise Application Intégration

Problématique.

Pré-intégration

- Systèmes indépendants ne disposant que de rares interfaces
- Processus peu réutilisables et définis comme des "usines à gaz"
- Synchronisation manuelle des données entre applications

Le concept EAI :Entreprise Application Intégration

Problématique.

Niveau 1: *Intégration Point-à-Point*

- Interfaces personnalisées pour des interaction point-à-point

- Utilisation d'interfaces programmables (API) ou d'outils de synchronisation de données.

- Outils et bus permettant l'échange de messages.

- Systèmes faiblement couplés.

Le concept EAI :Entreprise Application Intégration

Problématique.

Niveau 2 : *Intégration structurelle.*

•l'architecture des interfaces suit un modèle en étoile.

•le bus logiciel comprend des courtiers de messages ou des serveurs d'applications.

caractéristiques des bus logiciels:

- Transformation de données
- Système de règles
- Intégrité des transactions

•Existence d'un modèle des interfaces des applications de l'entreprise.

Le concept EAI :Entreprise Application Intégration

Problématique.

Niveau 3: *Intégration par le processus*

- l'architecture des interfaces du niveau 2 comme fondements du niveau 3
- l'information entre les systèmes n'est plus simplement échangée mais elle est gérée.
- le bus logiciel comprend des outils permettant d'automatiser la modélisation des processus

caractéristiques des bus logiciels :

- modélisation des flux (workflow)
- automatisation des routages
- automatisation des décisions

- Existence d'un modèle économique de l'entreprise (business model)

Le concept EAI :Entreprise Application Intégration

Problématique.

Niveau 4 : *Intégration externe*

- Enrichit le niveau 3 et l'étend pour prendre en compte plusieurs organisations (inter-organisationnel)

- Infrastructure réseau commune (ex. Internet)

- Adoption de standards pour l'échange de données (exemple,XML)

- Caractéristiques des bus logiciels :
  - transactions sécurisées
  - agents intelligents
  - définition des sémantiques de l'application
  - adaptation des interfaces

Le concept EAI :Entreprise Application Intégration

Problématique.

Définition de l'interopérabilité

L'interopérabilité est définie comme étant la capacité d'un ou de plusieurs systèmes ou composants d'échanger de l'information et d'utiliser l'information échangée [IEEE 1990].

Comme nous le remarquons, cette définition couvre des aspects très divers que sont :

- la caractérisation de l'information échangée,
- les moyens permettant les échanges,
- l'exploitation qui est faite de cette information.

L'ensemble de ces aspects concernent les composants et les systèmes logiciels.

Remarque : dans la pratique, la notion d'interopérabilité préserve l'identité des composants interopérant, contrairement au phénomène d'intégration pour lequel les composants perdent leur identité (ils sont absorbés").

Questions:

- *Expliquer comment et pourquoi une solution progiciel peut être une source d'enrichissement des fonctions de l'entreprise.*
- *Pourquoi le cycle de développement en V d'une solution logicielle nécessite une certaine maturité et un engagement important de la part des utilisateurs.*
- *Donner deux exemples de besoins non fonctionnels concernant le domaine gestion et suivi des stages traités en travaux dirigés.*
- *L'école a décidé de confier la réalisation du logiciel à une société externe. La livraison du produit est prévue pour le début de l'année universitaire. Énumérer les travaux à réaliser pour aboutir à la mise en service du logiciel.*
- *En quoi consiste l'intégration de la solution avec l'existant dans le cas gestion et suivi des stages.*

Questions:

- *Proposer un modèle de formulaire permettant de réaliser le cas : Choisir trois offres de stage. Vous pouvez vous inspirer des modèles étudiés en travaux pratiques.*

- *Préciser les contrôles qui vous semblent convenir pour la réalisation de ce formulaire.*
- *Le fichier des entreprises, actuellement géré à l'aide d'un tableur , comporte les informations sur 2500 entreprises. Quelle solution proposez vous pour disposer des informations. Ce point n'est pas inclu dans le contrat passé avec la société de services chargée de de réalisation du logiciel .*
- *Commenter l'approche adoptée par l'école pour informatiser la gestion et le suivi des stages. Préciser le type d'approche : Fonctionnel, systémique. Quels sont les impacts de cette approche sur le système d'information.*

Questions:

- *Énumérer les bénéfices que peut tirer cette école de la mise en place de cette solution.*

- La **licence** BSD (*Berkeley software distribution license*) est une licence libre utilisée pour la distribution de logiciels. Elle permet de réutiliser tout ou partie du logiciel sans restriction, qu'il soit intégré dans un logiciel libre ou propriétaire.
- La version originale de la licence BSD incluait une clause de publicité particulièrement contraignante qui obligeait la mention du copyright dans toute publicité ou document fourni avec le logiciel, ce qui pouvait provoquer quelques problèmes en cas d'utilisation d'un grand nombre de composants sous cette licence. La nouvelle version de cette licence ne contient pas cette clause de publicité.