

Master 2 LITL

Programmation pour le TAL (SLT0905V)

Introduction : présentation de l'U.E. et développement logiciel

Franck Sajous/CLLE-ERSS

Séance 1

Savoir et savoir-faire

L'objectif :

- **n'est pas** de vous dispenser une formation qui fera de vous des informaticiens généralistes
- mais de vous amener vers une certaine autonomie, y compris votre capacité à l'auto-formation (vous savez déjà programmer !)
- **est** de vous donner une vision d'ensemble, théorique et pratique, de ce qu'est le développement logiciel en général :
 - en saisir les enjeux ;
 - comprendre le métier de futurs collaborateurs potentiels ;
 - acquérir un socle de culture commune, *i.e.* parler des mêmes concepts en utilisant la même terminologie.
- de vous faire saisir les spécificités de la programmation orientée objets
- de vous faire acquérir un ensemble de bonnes pratiques

Contenu

Thèmes

- Développement logiciel ;
- Méthodes (processus) de développement ;
- Analyse, conception et modélisation orientées objets ;
- Programmation orientée objets (POO).

Contenu

Thèmes

- Développement logiciel ;
- Méthodes (processus) de développement ;
- Analyse, conception et modélisation orientées objets ;
- Programmation orientée objets (POO).

Concrètement (techniquement)

- UML (langage de conception et modélisation... + communication et documentation si affinités)
Logiciel Dia
- Java (langage de POO)
- API Java
- Eclipse (environnement de développement)

Évaluation

Continue

- compte-rendus à effectuer en binômes :
 - synthèse du cours, ou de lectures
 - et/ou exercice (e.g. modélisation)
 - et/ou code source
 - → à envoyer par mail au plus tard pour le mercredi midi précédant la séance du vendredi

Terminale

- un travail à rendre à la fin du semestre
- dans la continuité des rendus précédents

→ une note globale qui tient compte de l'ensemble des restitutions

Organisation

Séances + autonomie

- séance : alternance de cours, de TD (modélisation) et de travail sur machine (programmation)
- autonomie :
 - dans la suite de la séance : continuer exercices de modélisation et TPs de programmation
 - préparation de la séance suivante : lecture + questions

Interaction importante

- entre vous : confrontez votre (in)compréhension du cours, vos idées/difficultés de mise en œuvre
- avec l'enseignant : faites part des difficultés/incertitudes au fur et à mesure
- apprentissage incrémental et cumulatif : besoin de maîtriser toutes les notions étudiées

Logiciels. . .

Questions préliminaires : un logiciel. . .

- qu'est-ce que c'est ?
- qu'est-ce que ça fait ?
- qui les utilise, et comment ?
- qui les conçoit ?

Logiciels. . .

Questions préliminaires : un logiciel. . .

- qu'est-ce que c'est ?
- qu'est-ce que ça fait ?
- qui les utilise, et comment ?
- qui les conçoit ?

TLFi

INFORMAT. [P. oppos. à *matériel*] „Ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de données“
(J. O., Vocab. de l'informat., 17 janv. 1982).
[<http://www.cnrtl.fr/definition/logiciel>]

Définitions (2)

Wikipédia

En informatique, un logiciel est un ensemble d'informations relatives à des **traitements effectués automatiquement** par un appareil informatique. Y sont incluses les **instructions de traitement, regroupées sous forme de programmes**, des **données** et de la documentation. Le tout est **stocké sous forme d'un ensemble de fichiers dans une mémoire**.

[<http://fr.wikipedia.org/wiki/Logiciel>]

Computer software (Wikipedia)

Computer software, or just software, is a **collection of computer programs and related data** that provide the **instructions for telling a computer what to do and how to do it**.

In other words, software is a conceptual entity which is a set of computer programs [...] concerned with the operation of a **data processing** system.

[http://en.wikipedia.org/wiki/Computer_software]

Définitions (3)

Logiciel (Encyclopédie Universalis)

Tout logiciel correspond à un problème à résoudre, dont il matérialise une solution possible parmi d'autres. On peut donc classer les logiciels en fonction du type de problèmes à résoudre :

- des **problèmes généraux**, pour lesquels on peut réaliser des **logiciels généraux**, qu'il faudra cependant adapter (« **paramétrer** »)
- des **problèmes spécifiques**, pour lesquels il faudra réaliser une **programmation ad hoc**.

Pour les logiciels généraux [...] il faut distinguer :

- les logiciels professionnels, appelés progiciels, conçus pour les entreprises, comme les progiciels de gestion intégrée
- les logiciels grand public, qui intéressent les particuliers et dont le meilleur exemple est Internet.

[<http://www.universalis.fr/encyclopedie/logiciels/2-typologie-des-logiciels/>]

Définitions (3)

Logiciel (Encyclopédie Universalis)

Tout logiciel correspond à un problème à résoudre, dont il matérialise une solution possible parmi d'autres. On peut donc classer les logiciels en fonction du type de problèmes à résoudre :

- des problèmes généraux, pour lesquels on peut réaliser des logiciels généraux, qu'il faudra cependant adapter (« paramétrer »)
- des problèmes spécifiques, pour lesquels il faudra réaliser une programmation ad hoc.

Pour les logiciels généraux [...] il faut distinguer :

- les logiciels professionnels, appelés progiciels, conçus pour les entreprises, comme les progiciels de gestion intégrée
- les logiciels grand public, qui intéressent les particuliers et *dont le meilleur exemple est Internet. (???)*

[<http://www.universalis.fr/encyclopedie/logiciels/2-typologie-des-logiciels/>]

Couches d'abstraction

Logiciel (Wikipédia)

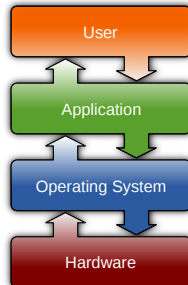
- Un **logiciel applicatif**, le type de logiciel le plus courant, aussi appelé **application informatique** : un logiciel dont les automatismes sont destinés à **assister un utilisateur dans une de ses activités**.
- Un **logiciel système** (ou logiciel de base). C'est un logiciel dont les automatismes contrôlent l'**utilisation du matériel informatique par les logiciels applicatifs**.
- Le système d'exploitation sert **d'interface entre le matériel et les logiciels applicatifs**. C'est l'ensemble de logiciels système central qui contrôle l'utilisation de l'appareil informatique par les autres logiciels.

[<http://fr.wikipedia.org/wiki/Logiciel>]

Définitions (6)

Logiciel/Computer Software (Wikipédia&Wikipedia, suite et fin)

- The term was coined to contrast to the old term hardware (meaning physical devices).
- Le logiciel peut être une application informatique complète, ou une pièce détachée telle qu'un composant logiciel.



Logiciels : premiers éléments de réponse

Qu'est-ce que c'est ?

- Un ensemble (ordonné) d'instructions, stockées sur un support (fichier, mémoire) qui spécifie à un système (matériel ?) quoi faire. . .
- . . . en interagissant avec un utilisateur ou en s'exécutant entièrement automatiquement.

Qu'est-ce que ça fait ?

- Ça manipule des données ;
- Ça résout des problèmes ;
- Ça permet à d'autres logiciels de faire quelque chose.

Autres éléments de typologie

Les logiciels. . .

- **qui les utilise ?**
- **qui les conçoit ?**

Autres éléments de typologie

Les logiciels. . .

- **qui les utilise ?**
- **qui les conçoit ?**

Qui utilise des logiciels ? (Où utilise-t-on des logiciels ?)

- particuliers ou professionnels, à la maison ou au bureau : des logiciels de bureautique (traitement de texte, tableur), client de messagerie, retouche d'images, etc.

Autres éléments de typologie

Les logiciels. . .

- **qui les utilise ?**
- **qui les conçoit ?**

Qui utilise des logiciels ? (Où utilise-t-on des logiciels ?)

- particuliers ou professionnels, à la maison ou au bureau : des logiciels de bureautique (traitement de texte, tableur), client de messagerie, retouche d'images, etc.
- calcul et simulation (Météo France, CERN, Airbus, etc.)

Autres éléments de typologie

Les logiciels...

- **qui les utilise ?**
- **qui les conçoit ?**

Qui utilise des logiciels ? (Où utilise-t-on des logiciels ?)

- particuliers ou professionnels, à la maison ou au bureau : des logiciels de bureautique (traitement de texte, tableur), client de messagerie, retouche d'images, etc.
- calcul et simulation (Météo France, CERN, Airbus, etc.)
- à peu près dans toutes les structures : systèmes d'information (entreprises, banques, universités, bibliothèques, commerces, etc.)

Autres éléments de typologie

Les logiciels. . .

- **qui les utilise ?**
- **qui les conçoit ?**

Qui utilise des logiciels ? (Où utilise-t-on des logiciels ?)

- particuliers ou professionnels, à la maison ou au bureau : des logiciels de bureautique (traitement de texte, tableur), client de messagerie, retouche d'images, etc.
- calcul et simulation (Météo France, CERN, Airbus, etc.)
- à peu près dans toutes les structures : systèmes d'information (entreprises, banques, universités, bibliothèques, commerces, etc.)
- sur le web : pages dynamiques (commerce électronique, déclaration d'impôts en ligne, etc.), moteurs de recherche

Autres éléments de typologie

Les logiciels. . .

- **qui les utilise ?**
- **qui les conçoit ?**

Qui utilise des logiciels ? (Où utilise-t-on des logiciels ?)

- particuliers ou professionnels, à la maison ou au bureau : des logiciels de bureautique (traitement de texte, tableur), client de messagerie, retouche d'images, etc.
- calcul et simulation (Météo France, CERN, Airbus, etc.)
- à peu près dans toutes les structures : systèmes d'information (entreprises, banques, universités, bibliothèques, commerces, etc.)
- sur le web : pages dynamiques (commerce électronique, déclaration d'impôts en ligne, etc.), moteurs de recherche
- pour l'éducation, le domaine médical, etc.

Autres éléments de typologie

Les logiciels. . .

- **qui les utilise ?**
- **qui les conçoit ?**

Qui utilise des logiciels ? (Où utilise-t-on des logiciels ?)

- particuliers ou professionnels, à la maison ou au bureau : des logiciels de bureautique (traitement de texte, tableur), client de messagerie, retouche d'images, etc.
- calcul et simulation (Météo France, CERN, Airbus, etc.)
- à peu près dans toutes les structures : systèmes d'information (entreprises, banques, universités, bibliothèques, commerces, etc.)
- sur le web : pages dynamiques (commerce électronique, déclaration d'impôts en ligne, etc.), moteurs de recherche
- pour l'éducation, le domaine médical, etc.
- logiciels embarqués : voitures, satellites, ascenseurs, machine-outils dans chaînes de montage (limite floue « soft/hard »)

Autres éléments de typologie

Les logiciels. . .

- **qui les utilise ?**
- **qui les conçoit ?**

Qui utilise des logiciels ? (Où utilise-t-on des logiciels ?)

- particuliers ou professionnels, à la maison ou au bureau : des logiciels de bureautique (traitement de texte, tableur), client de messagerie, retouche d'images, etc.
- calcul et simulation (Météo France, CERN, Airbus, etc.)
- à peu près dans toutes les structures : systèmes d'information (entreprises, banques, universités, bibliothèques, commerces, etc.)
- sur le web : pages dynamiques (commerce électronique, déclaration d'impôts en ligne, etc.), moteurs de recherche
- pour l'éducation, le domaine médical, etc.
- logiciels embarqués : voitures, satellites, ascenseurs, machine-outils dans chaînes de montage (limite floue « soft/hard »)
- des logiciels pour aider à créer des logiciels. . .

Concepteurs

Qui conçoit les logiciels ?

- **Éditeurs**

Identifient (ou créent) un besoin, développent un logiciel et le mettent sur le marché, à destination des particuliers ou des professionnels

- **Services informatiques (internes)**

Développent des logiciels spécifiques aux besoins d'une entité (entreprise, industrie, institution, etc.). Spécialisation dans le domaine d'activité.

- **SSII (Société de Service en Ingénierie Informatique)**

Idem, mais externalisé. Développent sur commande tout ou partie d'un logiciel. SSII généraliste ou (souvent) spécialisée dans un ou plusieurs secteurs donnés.

Interaction possible entre plusieurs SSII et/ou services internes.

- **Communautés de développeurs (→ logiciels libres)**

Tout un corps de métier

L'artisanat du logiciel

De nombreux rôles, obligatoires ou facultatifs, cloisonnés ou non :

- commerciaux
- chefs de projets, risk-managers
- développeurs (programmeurs)
- architectes
- experts techniques
- testeurs
- qualitaticiens
- graphistes
- ergonomes
- traducteurs
- RH, coaches, etc.

Génie logiciel, développement logiciel

Motivations

- Croissance de la complexité des logiciels vers la fin des 70s
- Baisse de la qualité, projets abandonnés, produits non conformes aux attentes, dépassement des coûts et délais → *“crise du logiciel”*
- Apparition des « micro-ordinateurs », omniprésence des logiciels
- Réalisation des enjeux économiques et de ceux liés à la sécurité engendrés par le développement logiciel

→ volonté de rationalisation de l'activité : qualité des applications critiques, maîtrise des coûts de développement et de maintenance

Génie logiciel, développement logiciel

Génie logiciel

- Arrêté ministériel, 1983 : « *ensemble des activités de conception et de mise en œuvre des produits et des procédures tendant à rationaliser la production du logiciel et son suivi* »

Génie logiciel, développement logiciel

Génie logiciel

- Arrêté ministériel, 1983 : « *ensemble des activités de conception et de mise en œuvre des produits et des procédures tendant à rationaliser la production du logiciel et son suivi* »
- Wikipédia : « *Le génie logiciel (anglais software engineering) est une science de génie industriel qui étudie les **méthodes de travail** et les **bonnes pratiques** des ingénieurs qui développent des logiciels. Le génie logiciel s'intéresse en particulier aux **procédures systématiques** qui permettent d'arriver à ce que des logiciels de grande taille correspondent aux attentes du client, soient fiables, aient un coût d'entretien réduit et de bonnes performances tout en respectant les délais et les coûts de construction* »

Génie logiciel, développement logiciel

Génie logiciel

- Arrêté ministériel, 1983 : « *ensemble des activités de conception et de mise en œuvre des produits et des procédures tendant à rationaliser la production du logiciel et son suivi* »
- Wikipédia : « *Le génie logiciel (anglais software engineering) est une science de génie industriel qui étudie les **méthodes de travail** et les **bonnes pratiques** des ingénieurs qui développent des logiciels. Le génie logiciel s'intéresse en particulier aux **procédures systématiques** qui permettent d'arriver à ce que des logiciels de grande taille correspondent aux attentes du client, soient fiables, aient un coût d'entretien réduit et de bonnes performances tout en respectant les délais et les coûts de construction* »
- Wikipédia : « *Est aussi appelée génie logiciel : l'ingénierie appliquée au logiciel informatique, l'activité par laquelle le code source d'un logiciel est spécifié puis produit. Le génie logiciel touche au cycle de vie des logiciels. Toutes les phases de la création d'un logiciel informatique y sont enseignées : l'**analyse du besoin**, l'**élaboration des spécifications**, la **conceptualisation du mécanisme interne** au logiciel ainsi que les techniques de programmation, le développement, la phase de test et finalement la maintenance.* »

Développement logiciel

Développement vs. génie

- **Génie** : une connotation plus industrielle, procédés plus systématiques.
- **Développement** : terme plus « mou » (beaucoup plus employé aujourd'hui), assume l'absence de méthode universelle et systématique, adaptée à toutes les situations (et qui marche à tous les coups)

Étapes traditionnelles

- Analyse
 - Recueil des besoins
 - Analyse fonctionnelle
- Conception
 - Architecture générale
 - Et conception détaillée (unitaire)
- Codage (programmation)
- Tests
- Recette

Développement logiciel

Développement vs. génie

- **Génie** : une connotation plus industrielle, procédés plus systématiques.
- **Développement** : terme plus « mou » (beaucoup plus employé aujourd'hui), assume l'absence de méthode universelle et systématique, adaptée à toutes les situations (et qui marche à tous les coups)

Étapes traditionnelles

- Analyse
 - Recueil des besoins
 - Analyse fonctionnelle
- Conception
 - Architecture générale
 - Et conception détaillée (unitaire)
- Codage (programmation)
- Tests
- Recette

Oui, mais... dans quel ordre ?

Processus de développement

Deux types principaux : cascade (waterfall) et itératif/incrémental

Le modèle cascade (plus ancien)

- 1 Analyse
- 2 Conception
- 3 Réalisation
- 4 Tests/Validation

Processus de développement

Deux types principaux : cascade (waterfall) et itératif/incrémental

Le modèle cascade (plus ancien)

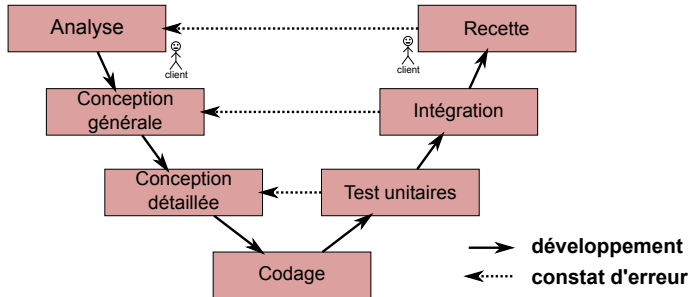
- 1 Analyse
- 2 Conception
- 3 Réalisation
- 4 Tests/Validation

Applications de plus grande taille, le “cycle en V” prévoit d'intégrer des sous-composants

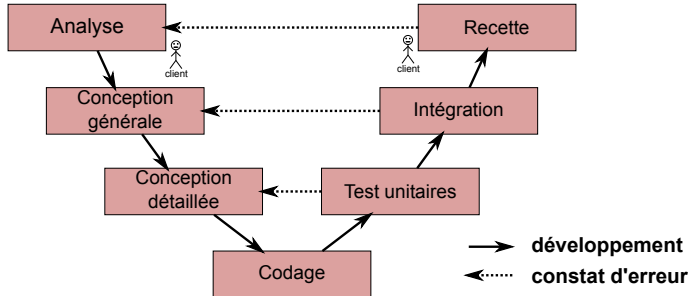
Le cycle en V

- 1 Analyse
- 2 Conception générale (architecture)
- 3 Conception détaillée (unitaire)
- 4 Réalisation (programmation)
- 5 Tests unitaires
- 6 Intégration
- 7 Validation/Tests

Processus de développement : cycle en V



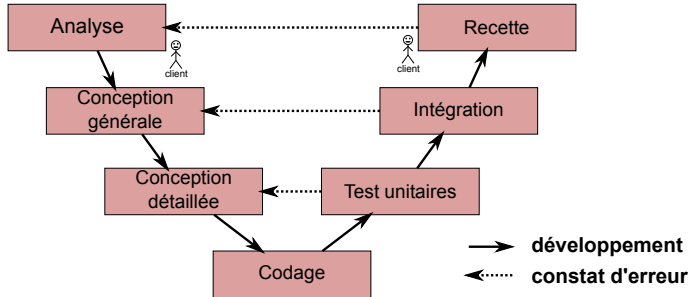
Processus de développement : cycle en V



Critiques

- lorsqu'une erreur est détectée tardivement, elle remet tout le cycle en cause → tout à refaire
- client/utilisateur consulté au tout début et à la toute fin (ne voit rien entre)

Processus de développement : cycle en V



Critiques

- lorsqu'une erreur est détectée tardivement, elle remet tout le cycle en cause → tout à refaire
- client/utilisateur consulté au tout début et à la toute fin (ne voit rien entre)

De la phase d'analyse au codage

- Qui recueille les besoins? Qui rédige le cahier des charges?
- À destination de qui? De(s) client(s)? De(s) programmeur(s)?

Tests unitaires/tests d'intégration : exemple

Recueil des besoins

« je voudrais une voiture facile à garer en ville et avec un siège très confortable »

Tests unitaires/tests d'intégration : exemple

Recueil des besoins

« je voudrais une voiture facile à garer en ville et avec un siège très confortable »

Conception
détaillée



Tests unitaires/tests d'intégration : exemple

Recueil des besoins

« je voudrais une voiture facile à garer en ville et avec un siège très confortable »

Conception
détaillée



Tests unitaires/tests d'intégration : exemple

Recueil des besoins

« je voudrais une voiture facile à garer en ville et avec un siège très confortable »

Conception
détaillée



Réalisation



Tests unitaires/tests d'intégration : exemple

Recueil des besoins

« je voudrais une voiture facile à garer en ville et avec un siège très confortable »

Conception
détaillée



Réalisation



Tests unitaires/tests d'intégration : exemple

Recueil des besoins

« je voudrais une voiture facile à garer en ville et avec un siège très confortable »

Conception
détaillée



Réalisation



Tests unitaires



Tests unitaires/tests d'intégration : exemple

Recueil des besoins

« je voudrais une voiture facile à garer en ville et avec un siège très confortable »

Conception
détaillée



Réalisation



Tests unitaires



Tests unitaires/tests d'intégration : exemple

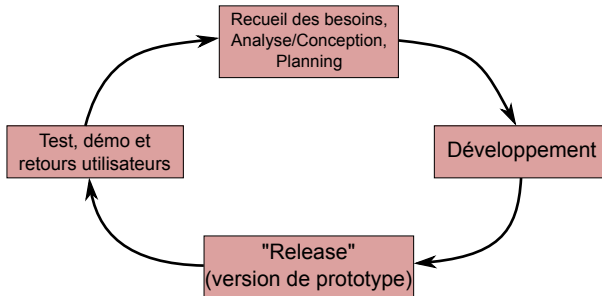
Recueil des besoins

« je voudrais une voiture facile à garer en ville et avec un siège très confortable »

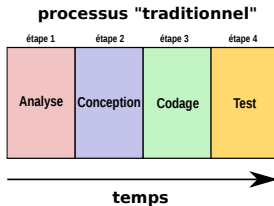


Test d'intégration...

Processus de développement : modèle itératif

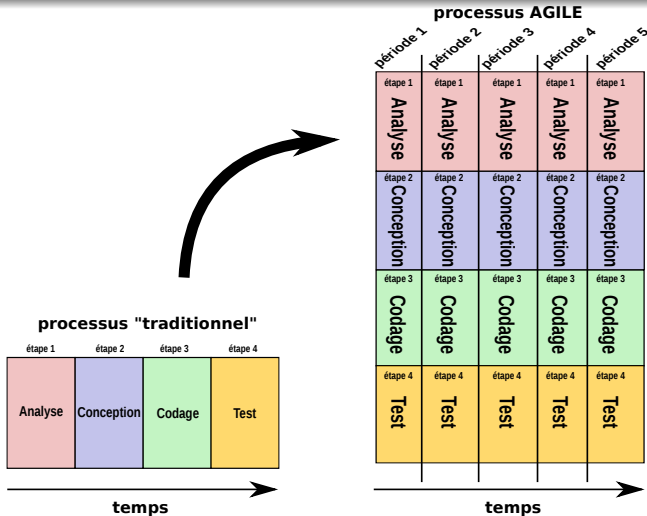


Processus de développement : cascade vs itératif



(adapté de *The Agile Samurai*, J. Rasmuson)

Processus de développement : cascade vs itératif



(adapté de *The Agile Samurai*, J. Rasmuson)

Origines du modèle itératif

AGILE manifesto

En 2001, 17 spécialistes du développement logiciel se rencontrent.
Constats partagés :

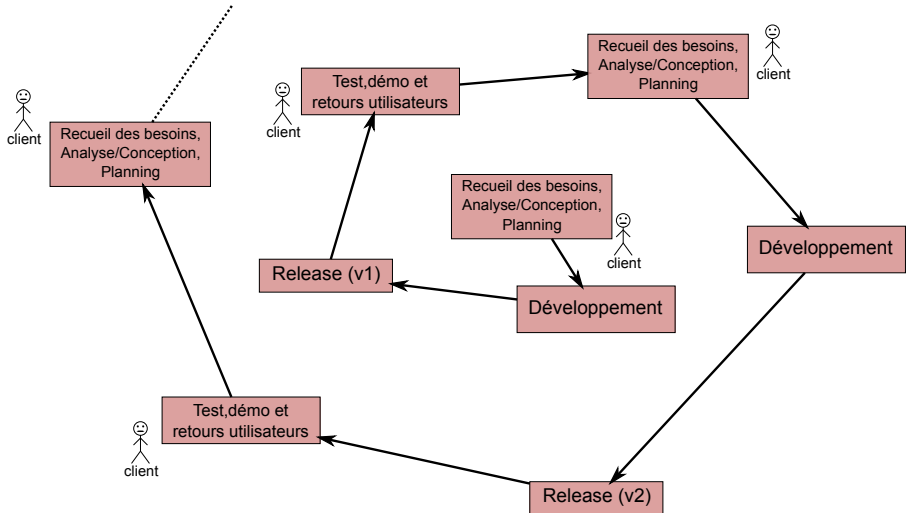
- Pas toujours une bonne intercompréhension entre clients/utilisateurs et analystes, puis analystes et développeurs, etc.
- Même les analystes les plus chevronnés peuvent commettre des erreurs
- Le client ne sait pas toujours formaliser ses besoins
(il ne sait pas toujours ce qu'il est techniquement possible de réaliser)
- Méthodes de développement : l'accent a été mis uniquement sur le processus, jamais sur l'humain
- → Rédaction d'un manifeste énonçant 12 principes généraux :
<http://agilemanifesto.org/>

Agilité

Principes

- Enchaîner des phases courtes de développement (e.g. 2 semaines) qui débouchent sur un prototype ;
- Time-boxing : respect impératif de la durée des phases. Renoncer à une fonctionnalité plutôt que reporter une release. *(Essayez... pourquoi ne pas appliquer ce principe au rendu de travaux universitaires ?) ;*
- Montrer le prototype au client. Selon les retours, raffiner le prototype ou revoir l'analyse ;
- Nécessité (et volonté) d'impliquer le client ;
- Communication directe et régulière entre tous les acteurs ;
- Place accordée à l'auto-organisation ;
- Erreurs (inévitables) détectées plus tôt → moindre impact.

Modèle itératif : représentation du cycle en spirale



De l'anarchisme organisationnel au coaching AGILE

Accueil des principes AGILE

- Des principes qui viennent de la communauté des développeurs (pas des décideurs) : des *"anarchistes organisationnels"* !
- Qui remettent en cause ce qui se faisait depuis des années
- Un désaveu des managers et des chefs de projets ?
- Prototyper ? Être prêt à "échouer" pour progresser (explorer pour sélectionner) : une *"culture de l'échec"* !
- Des équipes auto-organisées ? Des développeurs responsables ? Qui seraient autorisés (aptes ?) à communiquer ? Directement avec le client ???
- Des clients impliqués ?
- Prise en compte des valeurs humaines ?

De l'anarchisme organisationnel au coaching AGILE

Bientôt 20 ans d'Agilité

- Dès 2001, multiplication des méthodes (gestion de projets et programmation)
- Les principes AGILE ont fait leur preuve et gagné leurs lettres de noblesse
- Agilité présente dans toutes les SSII (si ce n'est par conviction, au moins par mode et pour ne pas laisser le champ libre à la concurrence)
- "Coach Agile" est devenu un métier
- Organismes de formation qui incluent les méthodes AGILE dans leur catalogue

Méthodes AGILE : la situation aujourd'hui

- Implantation croissante
- Une maturité dans la communauté et un certain recul (critique partielle de certaines méthodes), attitudes modérées : *"il n'existe pas une bonne méthode qui marche à tous les coups, seulement des principes, des bonnes pratiques"*
- Méthodes qui ne se cantonnent plus à la sphère informatique
E.g. campagnes d'annotation de corpus
- Méthodes pas nécessairement enseignées dans les formations spécialisées
- Implantation difficile dans certaines structures

AGILE

Finalement...

- Agile, juste un concentré de bon sens ?
- Enfin un nom respectable pour des principes et des méthodes déjà pratiquées auparavant !
- Dans la sphère politico-économique, une nouvelle connotation...

Et dans 20 ans ?