Introduction au génie logiciel

Jean-Claude Royer

16 novembre 2009

Organisation du cours

Organisation du cours

Nature du logiciel

Génie Logiciel

- Génie Logiciel : principes, méthodes, techniques et outils

Génie Logiciel

- Génie Logiciel : principes, méthodes, techniques et outils
- Domaine vaste qui couvre de nombreuses et diverses activités touchant aux technologies du logiciel, à la gestion des ressources et à la mesure de la qualité

Génie Logiciel

- Génie Logiciel : principes, méthodes, techniques et outils
- Domaine vaste qui couvre de nombreuses et diverses activités touchant aux technologies du logiciel, à la gestion des ressources et à la mesure de la qualité
- Essayer de rationaliser, de maîtriser et de rendre efficace le processus de production du logiciel

Une définition : wikipedia

Le génie logiciel est en fait de l'ingénierie appliquée au logiciel informatique. Cette branche de l'informatique s'intéresse donc plus particulièrement à la manière dont le code source d'un logiciel est spécifié puis produit. Le génie logiciel touche donc au cycle de vie des logiciels. Toutes les phases de la création d'un logiciel informatique y sont donc enseignées : le développement, l'analyse du besoin, l'élaboration des spécifications, à la conceptualisation du mécanisme interne au logiciel ainsi que des techniques de programmation et finalement à la maintenance. Les projets relatifs à l'ingénierie logicielle sont de l'ordre du "Programming in the large", c'est à dire que les projets sont généralement de grande envergure et dépassent souvent les 10000 lignes de code. Ces projets nécessitent donc une équipe de développement bien structurée. La gestion de projet se retrouve donc le complément naturel du génie logiciel.

Définition de l'IEEE

- The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software ; that is, the application of engineering to software.

Définition de l'IEEE

- The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software ; that is, the application of engineering to software.
- Le génie logiciel est considéré comme une discipline émergente

The SWEBOK Knowledge Areas (10)

- Software requirements

The SWEBOK Knowledge Areas (10)

- Software requirements
- Software design

The SWEBOK Knowledge Areas (10)

- Software requirements
- Software design
- Software construction

The SWEBOK Knowledge Areas (10)

- Software requirements
- Software design
- Software construction
- Software testing

The SWEBOK Knowledge Areas (10)

- Software requirements
- Software design
- Software construction
- Software testing
- Software maintenance

The SWEBOK Knowledge Areas (10)

- Software requirements
- Software design
- Software construction
- Software testing
- Software maintenance
- Software configuration management

The SWEBOK Knowledge Areas (10)

- Software requirements
- Software design
- Software construction
- Software testing
- Software maintenance
- Software configuration management
- Software engineering management

The SWEBOK Knowledge Areas (10)

- Software requirements
- Software design
- Software construction
- Software testing
- Software maintenance
- Software configuration management
- Software engineering management
- Software engineering process

The SWEBOK Knowledge Areas (10)

- Software requirements
- Software design
- Software construction
- Software testing
- Software maintenance
- Software configuration management
- Software engineering management
- Software engineering process
- Software engineering tools and methods

The SWEBOK Knowledge Areas (10)

- Software requirements
- Software design
- Software construction
- Software testing
- Software maintenance
- Software configuration management
- Software engineering management
- Software engineering process
- Software engineering tools and methods
- Software quality

Organisation

- Responsable : Jean-Claude Royer

Organisation

- Responsable : Jean-Claude Royer
- Intervenants : Jean-Claude Royer, Mario Südholt, Cédric Dumas, Fabien Hermenier, Gilles Chabert, Xavier Lorca

Organisation

- Responsable : Jean-Claude Royer
- Intervenants : Jean-Claude Royer, Mario Südholt, Cédric Dumas, Fabien Hermenier, Gilles Chabert, Xavier Lorca
- Jours : lundi et mercredi

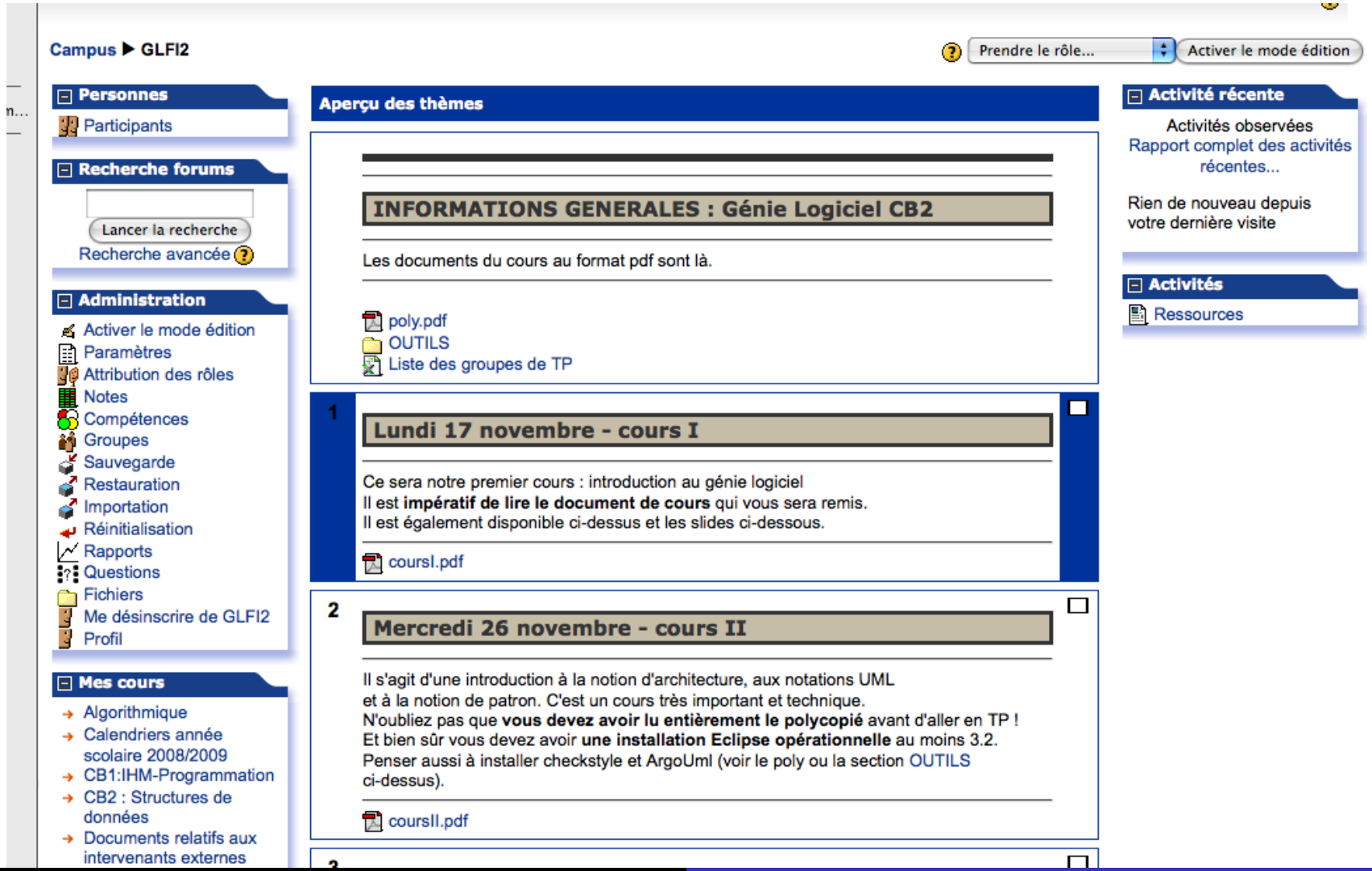
Organisation

- Responsable : Jean-Claude Royer
- Intervenants : Jean-Claude Royer, Mario Südholt, Cédric Dumas, Fabien Hermenier, Gilles Chabert, Xavier Lorca
- Jours : lundi et mercredi
- Salle de TP et utilisation d'Eclipse **obligatoire**

Organisation

- Responsable : Jean-Claude Royer
- Intervenants : Jean-Claude Royer, Mario Südholt, Cédric Dumas, Fabien Hermenier, Gilles Chabert, Xavier Lorca
- Jours : lundi et mercredi
- Salle de TP et utilisation d'Eclipse **obligatoire**
- Installer : **ArgoUml**, voir le poly pour explications

https ://nte.gemtech.fr/campus/course/view.php ?id=371



Planning

Date	Séances	Contenu
Lundi 17/11/08	C	C1 : Introduction au génie logiciel
Lundi 25/11/08	C	C2 : Éléments pour les architectures lo
Mercredi 7/12/08	2TP	TP0
Lundi 9/12/08	2TP	TP1
Mercredi 14/12/08	4TP	TP2 Noté
Lundi 4/01/10	4TP	TP3
Mercredi 6/01/10	2TP	TP4 Noté

Contrôles

- Le premier TP non noté

Contrôles

- Le premier TP non noté
- TP 2 et TP4 noté

Contrôles

- Le premier TP non noté
- TP 2 et TP4 noté
- La formule savante est : 0.5*tp + 0.5*tp

Contrôles

- Le premier TP non noté
- TP 2 et TP4 noté
- La formule savante est : 0.5*tp + 0.5*tp
- Mais bien sûr modulable suivant votre comportement, activité, réactivité etc

Bibliographie

- *Génie logiciel : principes, méthodes et techniques*, PPUR, A. Strohmeier et Didier Buchs

Bibliographie

- *Génie logiciel : principes, méthodes et techniques*, PPUR, A. Strohmeier et Didier Buchs
- *Software Design Using Java 2*, Kevin Lano, José Luiz Fiadero, Luis Andrade, 2002, Palgrave MacMillan.

Bibliographie

- *Génie logiciel : principes, méthodes et techniques*, PPUR, A. Strohmeier et Didier Buchs
- *Software Design Using Java 2*, Kevin Lano, José Luiz Fiadero, Luis Andrade, 2002, Palgrave MacMillan.
- *Dictionnaire encyclopédique du génie logiciel*, Henri Habrias, Masson, 1997.

Bibliographie

- *Génie logiciel : principes, méthodes et techniques*, PPUR, A. Strohmeier et Didier Buchs
- *Software Design Using Java 2*, Kevin Lano, José Luiz Fiadero, Luis Andrade, 2002, Palgrave MacMillan.
- *Dictionnaire encyclopédique du génie logiciel*, Henri Habrias, Masson, 1997.

- Pas facile de trouver un livre qui recouvre parfaitement ce cours

Bibliographie

- *Génie logiciel : principes, méthodes et techniques*, PPUR, A. Strohmeier et Didier Buchs
- *Software Design Using Java 2*, Kevin Lano, José Luiz Fiadero, Luis Andrade, 2002, Palgrave MacMillan.
- *Dictionnaire encyclopédique du génie logiciel*, Henri Habrias, Masson, 1997.
- Pas facile de trouver un livre qui recouvre parfaitement ce cours
- Cours sur Web : EPFL, IRO, Bordeaux I, ...

Le programme réel

- Introduction au génie logiciel

Le programme réel

- Introduction au génie logiciel
- Éléments pour les architectures logicielles

Le programme réel

- Introduction au génie logiciel
- Éléments pour les architectures logicielles
- Outils et environnements de développement

La nature du logiciel

- D'abord de l'information !

La nature du logiciel

- D'abord de l'information !
- Produit immatériel, intellectuel, aspect "artistique"

La nature du logiciel

- D'abord de l'information !
- Produit immatériel, intellectuel, aspect "artistique"
- Travail d'ingénieurs et de programmeurs, pas de manufaction

La nature du logiciel

- D'abord de l'information !
- Produit immatériel, intellectuel, aspect "artistique"
- Travail d'ingénieurs et de programmeurs, pas de manufaction
- Pas d'usure, copie et destruction facile

La nature du logiciel

- D'abord de l'information !
- Produit immatériel, intellectuel, aspect "artistique"
- Travail d'ingénieurs et de programmeurs, pas de manufaction
- Pas d'usure, copie et destruction facile
- Coût du logiciel >> coût du matériel

La nature du logiciel

- D'abord de l'information !
- Produit immatériel, intellectuel, aspect "artistique"
- Travail d'ingénieurs et de programmeurs, pas de manufaction
- Pas d'usure, copie et destruction facile
- Coût du logiciel >> coût du matériel
- Coût de maintenance >> coût de développement

La nature du logiciel

- D'abord de l'information !
- Produit immatériel, intellectuel, aspect "artistique"
- Travail d'ingénieurs et de programmeurs, pas de manufaction
- Pas d'usure, copie et destruction facile
- Coût du logiciel >> coût du matériel
- Coût de maintenance >> coût de développement
- La fiabilité est fondamentale (ou devrait l'être !)

Une autre explication !

" La programmation est aujourd'hui une course entre les ingénieurs informaticiens qui essaient de construire des programmes plus grands et mieux à l'épreuve des idiots, et l'univers qui essaie de produire des idiots plus grands et plus idiots. Jusqu'à présent, l'univers gagne."

Rich Cook

Génie logiciel et économie

- L'économie de toute nation développée dépend du logiciel

Génie logiciel et économie

- L'économie de toute nation développée dépend du logiciel
- De plus en plus de systèmes sont contrôlés par logiciel

Génie logiciel et économie

- L'économie de toute nation développée dépend du logiciel
- De plus en plus de systèmes sont contrôlés par logiciel
- C'est une tendance industrielle mais aussi tertiaire, loisir, famille, etc

Génie logiciel et économie

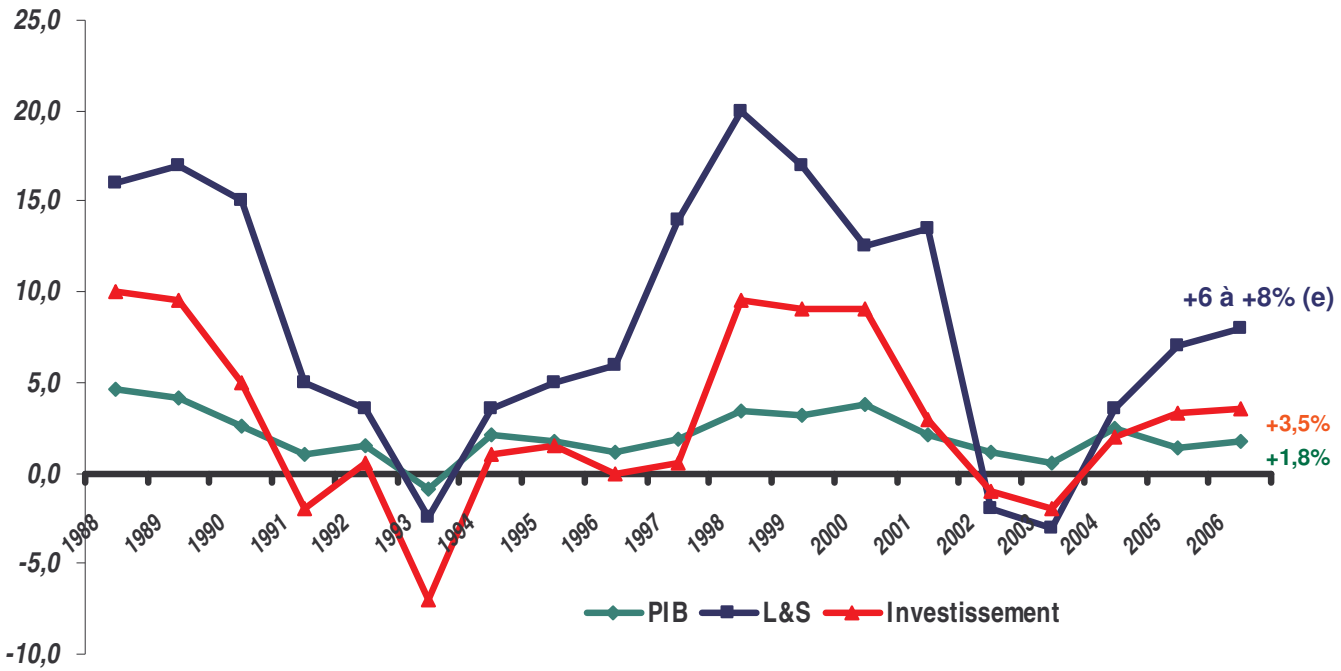
- L'économie de toute nation développée dépend du logiciel
- De plus en plus de systèmes sont contrôlés par logiciel
- C'est une tendance industrielle mais aussi tertiaire, loisir, famille, etc
- La part de l'informatique représente une partie importante du PIB d'un pays développé

Génie logiciel et économie

- L'économie de toute nation développée dépend du logiciel
- De plus en plus de systèmes sont contrôlés par logiciel
- C'est une tendance industrielle mais aussi tertiaire, loisir, famille, etc
- La part de l'informatique représente une partie importante du PIB d'un pays développé
- L'emploi informatique est toujours un secteur important (actuellement premier secteur en France avec plus d'un quart des emplois de cadres)

Génie logiciel et économie

2005-2006 : Croissance soutenue du secteur L&S dans un contexte économique peu porteur (2/3)



Contraintes d'utilisation

- Le logiciel est autorisé si aucune augmentation du taux de mortalité "normale" n'est constatée

Contraintes d'utilisation

- Le logiciel est autorisé si aucune augmentation du taux de mortalité "normale" n'est constatée
- Exemples : taux de défaillance acceptable :
  10^{-7} pannes/heures (avion)
  10^{-9} pannes/heures (train)

Contraintes d'utilisation

- Le logiciel est autorisé si aucune augmentation du taux de mortalité "normale" n'est constatée
- Exemples : taux de défaillance acceptable :
  10^{-7} pannes/heures (avion)
  10^{-9} pannes/heures (train)
- Ceci implique des probabilités de défaillance plus faible pour la partie logicielle :
  10^{-9} pannes/heures (avion)
  10^{-11} pannes/heures (train)

Exemples récents de problèmes

- Ariane 5 : problème de débordement de calculs

Exemples récents de problèmes

- Ariane 5 : problème de débordement de calculs
- Mars Pathfinder Mission : problème de reset dû à une tâche trop longue

Exemples récents de problèmes

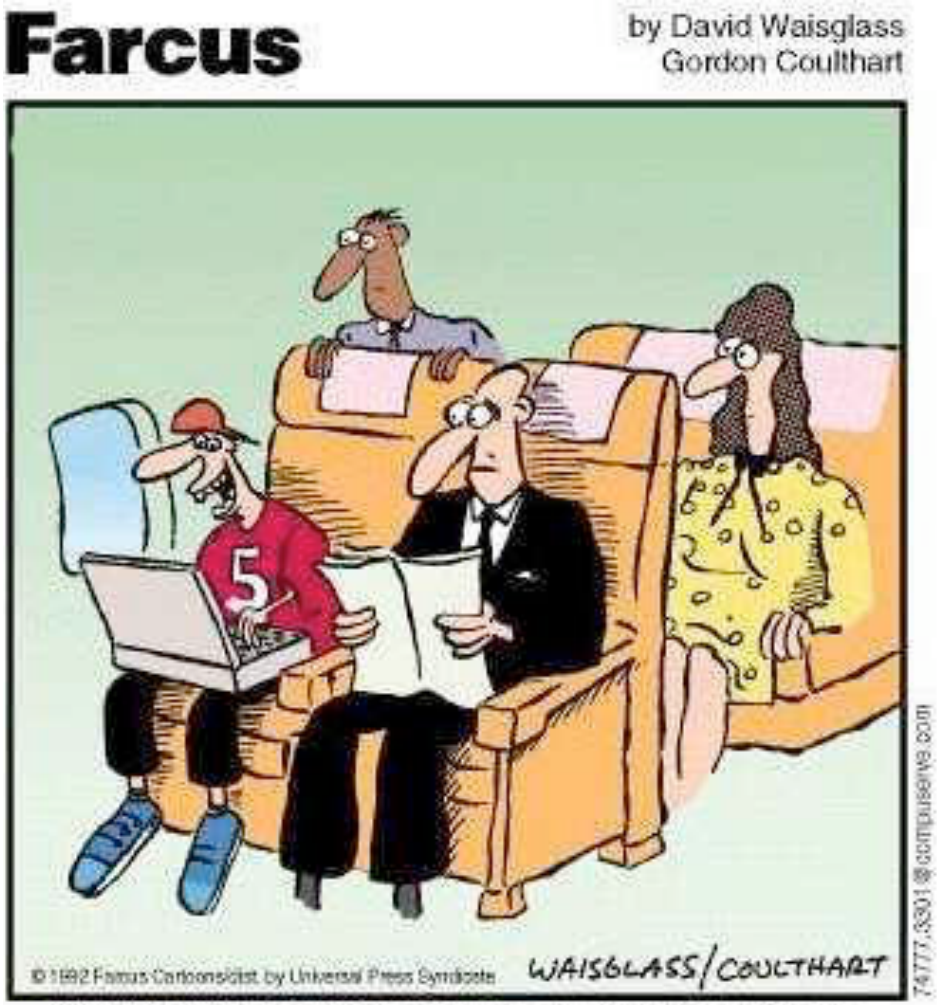
- Ariane 5 : problème de débordement de calculs
- Mars Pathfinder Mission : problème de reset dû à une tâche trop longue
- Robot Spirit : débordement des fichiers et redémarrage

Exemples récents de problèmes

- Ariane 5 : problème de débordement de calculs
- Mars Pathfinder Mission : problème de reset dû à une tâche trop longue
- Robot Spirit : débordement des fichiers et redémarrage
- Erreur du Pentium (calculs flottants) : 470 10^6 $

Exemples récents de problèmes

- Ariane 5 : problème de débordement de calculs
- Mars Pathfinder Mission : problème de reset dû à une tâche trop longue
- Robot Spirit : débordement des fichiers et redémarrage
- Erreur du Pentium (calculs flottants) : 470 10^6 \$
- Anomalies “spatial” en 1999 : 8 de lanceurs, 16 de satellites, 4 sondes, télescopes, ...



Yaahooo !!! J'ai réussi à faire planter le système de la tour de contrôle !!!


Bonne Année 2000
Le bogue ? Quelle plaisanterie !
Aucun problème à l'horizon.

La crise du logiciel

- Crise du logiciel dès 1969

La crise du logiciel

- Crise du logiciel dès 1969
- Dépassement moyen de budget : 70%

La crise du logiciel

- Crise du logiciel dès 1969
- Dépassement moyen de budget : 70%
- Dépassement moyen de délai : 50%

La crise du logiciel

- Crise du logiciel dès 1969
- Dépassement moyen de budget : 70%
- Dépassement moyen de délai : 50%
- La maintenance est aussi un gros problème

La crise du logiciel

- Crise du logiciel dès 1969
- Dépassement moyen de budget : 70%
- Dépassement moyen de délai : 50%
- La maintenance est aussi un gros problème
- Une grande partie du coût du logiciel provient de la maintenance (40 à 70 %)

La crise du logiciel

- Crise du logiciel dès 1969
- Dépassement moyen de budget : 70%
- Dépassement moyen de délai : 50%
- La maintenance est aussi un gros problème
- Une grande partie du coût du logiciel provient de la maintenance (40 à 70 %)
- Coût logiciel >>> coût du matériel

La crise du logiciel

- Crise du logiciel dès 1969
- Dépassement moyen de budget : 70%
- Dépassement moyen de délai : 50%
- La maintenance est aussi un gros problème
- Une grande partie du coût du logiciel provient de la maintenance (40 à 70 %)
- Coût logiciel >>> coût du matériel
- Rapport logiciel/matériel : 50-50 en 1965, 70-30 en 75, 80-20 en 1985 ...

État du génie logiciel

- Débute dans les années 60, discipline jeune et en pleine évolution

État du génie logiciel

- Débute dans les années 60, discipline jeune et en pleine évolution
- Beaucoup de techniques trop informelles

État du génie logiciel

- Débute dans les années 60, discipline jeune et en pleine évolution
- Beaucoup de techniques trop informelles
- Quelques tendances plus rationnelles mais encore très insuffisantes et peu répandues

État du génie logiciel

- Débute dans les années 60, discipline jeune et en pleine évolution
- Beaucoup de techniques trop informelles
- Quelques tendances plus rationnelles mais encore très insuffisantes et peu répandues
- Peu de consensus sur l'ensemble des technologies

État du génie logiciel

- Débute dans les années 60, discipline jeune et en pleine évolution
- Beaucoup de techniques trop informelles
- Quelques tendances plus rationnelles mais encore très insuffisantes et peu répandues
- Peu de consensus sur l'ensemble des technologies
- Une "anomalie" par rapport aux autres sciences de l'ingénieur

État du génie logiciel

- Débute dans les années 60, discipline jeune et en pleine évolution
- Beaucoup de techniques trop informelles
- Quelques tendances plus rationnelles mais encore très insuffisantes et peu répandues
- Peu de consensus sur l'ensemble des technologies
- Une "anomalie" par rapport aux autres sciences de l'ingénieur
- Manque de maturité ?

Évolution de l'informatique

- Promouvoir une véritable science de l'ingénieur du logiciel

Évolution de l'informatique

- Promouvoir une véritable science de l'ingénieur du logiciel
- Facteurs positifs
  - Économie, recherche, intérêts industriels, systèmes critiques, améliorations constantes des formations

Évolution de l'informatique

- Promouvoir une véritable science de l'ingénieur du logiciel
- Facteurs positifs
  - Économie, recherche, intérêts industriels, systèmes critiques, améliorations constantes des formations
- Facteurs négatifs
  - Monopoles, approche trop commerciale et à court terme, manque de formation, échecs de projet

Rôle de l'ingénieur Génie Logiciel 1/2

- Communication avec le client ou l'utilisateur dans les termes de l'application

Rôle de l'ingénieur Génie Logiciel 1/2

- Communication avec le client ou l'utilisateur dans les termes de l'application
- Lecture (voire aide à l'écriture) du cahier des charges

Rôle de l'ingénieur Génie Logiciel 1/2

- Communication avec le client ou l'utilisateur dans les termes de l'application
- Lecture (voire aide à l'écriture) du cahier des charges
- Traduction de cahier des charges en spécifications

Rôle de l'ingénieur Génie Logiciel 1/2

- Communication avec le client ou l'utilisateur dans les termes de l'application
- Lecture (voire aide à l'écriture) du cahier des charges
- Traduction de cahier des charges en spécifications
- Maîtrise des niveaux d'abstraction

Rôle de l'ingénieur Génie Logiciel 1/2

- Communication avec le client ou l'utilisateur dans les termes de l'application
- Lecture (voire aide à l'écriture) du cahier des charges
- Traduction de cahier des charges en spécifications
- Maîtrise des niveaux d'abstraction
- Maîtrise des phases et des différentes approches de développement

Rôle de l'ingénieur Génie Logiciel 2/2

- Maîtrise des techniques de programmation

Rôle de l'ingénieur Génie Logiciel 2/2

- Maîtrise des techniques de programmation
- Capacité à élaborer des modèles, y compris des modèles formels

Rôle de l'ingénieur Génie Logiciel 2/2

- Maîtrise des techniques de programmation
- Capacité à élaborer des modèles, y compris des modèles formels
- Connaissance et pratique des ateliers de génie logiciel

Rôle de l'ingénieur Génie Logiciel 2/2

- Maîtrise des techniques de programmation
- Capacité à élaborer des modèles, y compris des modèles formels
- Connaissance et pratique des ateliers de génie logiciel
- Capacité d'organisation, de gestion et de communication

Rôle de l'ingénieur Génie Logiciel 2/2

- Maîtrise des techniques de programmation
- Capacité à élaborer des modèles, y compris des modèles formels
- Connaissance et pratique des ateliers de génie logiciel
- Capacité d'organisation, de gestion et de communication
- C'est plus qu'un métier !

Responsabilités de l'ingénieur génie logiciel

- Pas uniquement une activité technique informatique

Responsabilités de l'ingénieur génie logiciel

- Pas uniquement une activité technique informatique
- Responsabilités éthiques, sociales et professionnelles

Responsabilités de l'ingénieur génie logiciel

- Pas uniquement une activité technique informatique
- Responsabilités éthiques, sociales et professionnelles
  - Confidentialité

Responsabilités de l'ingénieur génie logiciel

- Pas uniquement une activité technique informatique
- Responsabilités éthiques, sociales et professionnelles
  - Confidentialité
  - Compétence

Responsabilités de l'ingénieur génie logiciel

- Pas uniquement une activité technique informatique
- Responsabilités éthiques, sociales et professionnelles
  - Confidentialité
  - Compétence
  - Droit de propriété individuel

Responsabilités de l'ingénieur génie logiciel

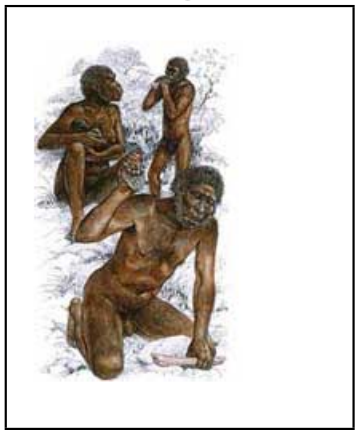
- Pas uniquement une activité technique informatique
- Responsabilités éthiques, sociales et professionnelles
  - Confidentialité
  - Compétence
  - Droit de propriété individuel
  - Mauvaise utilisation de l'ordinateur

Responsabilités de l'ingénieur génie logiciel

- Pas uniquement une activité technique informatique
- Responsabilités éthiques, sociales et professionnelles
  - Confidentialité
  - Compétence
  - Droit de propriété individuel
  - Mauvaise utilisation de l'ordinateur
- Pas de droits clairs ou bien formalisés sur ces problèmes

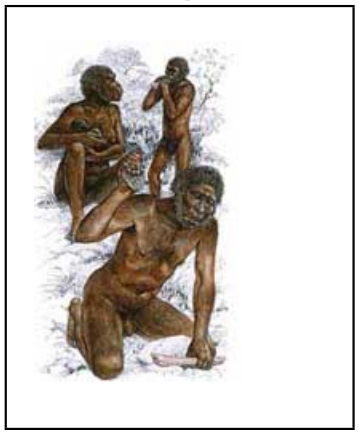
Un mot sur les outils

- Un longue histoire indissociable de l'intelligence humaine ?



Un mot sur les outils

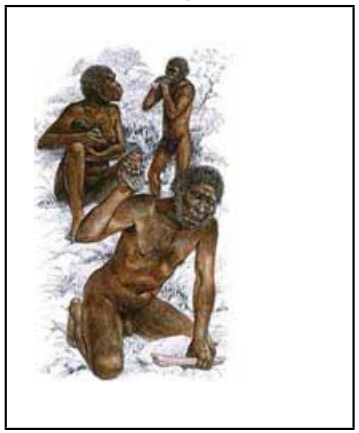
- Un longue histoire indissociable de l'intelligence humaine ?



- Indispensable pour parler d'ingénierie et améliorer la qualité et la productivité

Un mot sur les outils

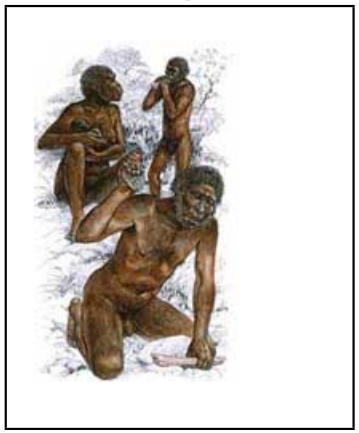
- Un longue histoire indissociable de l'intelligence humaine ?



- Indispensable pour parler d'ingénierie et améliorer la qualité et la productivité
- Aucun outil n'est parfait, particulièrement vrai malheureusement en informatique !

Un mot sur les outils

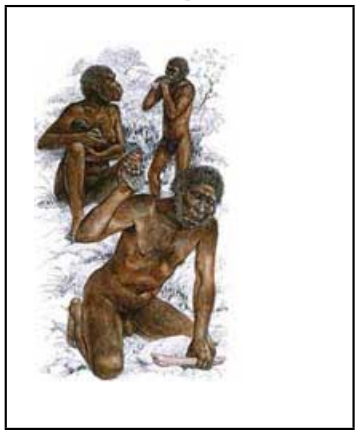
- Un longue histoire indissociable de l'intelligence humaine ?



- Indispensable pour parler d'ingénierie et améliorer la qualité et la productivité
- Aucun outil n'est parfait, particulièrement vrai malheureusement en informatique !
- Demande toujours un temps d'apprentissage et une utilisation parfois laborieuse

Un mot sur les outils

- Un longue histoire indissociable de l'intelligence humaine ?



- Indispensable pour parler d'ingénierie et améliorer la qualité et la productivité
- Aucun outil n'est parfait, particulièrement vrai malheureusement en informatique !
- Demande toujours un temps d'apprentissage et une utilisation parfois laborieuse
- Apprendre à apprendre à trouver et se servir des outils

Qualités du logiciel

- Il faut s'intéresser aux qualités du produit et aux qualités du processus de production

Qualités du logiciel

- Il faut s'intéresser aux qualités du produit et aux qualités du processus de production
- Qualités externes (pour l'utilisateur)

Qualités du logiciel

- Il faut s'intéresser aux qualités du produit et aux qualités du processus de production
- Qualités externes (pour l'utilisateur)
- Qualités internes (pour le développeur)

Qualités du logiciel

- Il faut s'intéresser aux qualités du produit et aux qualités du processus de production
- Qualités externes (pour l'utilisateur)
- Qualités internes (pour le développeur)
- Ces éléments ne sont pas indépendants

Qualités externes

- Fonctionnalité :

Qualités externes

- Fonctionnalité :
  - **Correction** : satisfait à ses spécifications

Qualités externes

- Fonctionnalité :
  - **Correction** : satisfait à ses spécifications
  - **Fiabilité** : probabilité d'exécution sans défaillance

Qualités externes

- Fonctionnalité :
  - **Correction** : satisfait à ses spécifications
  - **Fiabilité** : probabilité d'exécution sans défaillance
  - **Robustesse** : comportement raisonnable dans des conditions anormales

Qualités externes

- Fonctionnalité :
  - **Correction** : satisfait à ses spécifications
  - **Fiabilité** : probabilité d'exécution sans défaillance
  - **Robustesse** : comportement raisonnable dans des conditions anormales
- **Performance** : gestion efficace des ressources (temps, espace, matériels), mesures et statistiques sont possibles

Qualités externes

- Fonctionnalité :
  - **Correction** : satisfait à ses spécifications
  - **Fiabilité** : probabilité d'exécution sans défaillance
  - **Robustesse** : comportement raisonnable dans des conditions anormales
- **Performance** : gestion efficace des ressources (temps, espace, matériels), mesures et statistiques sont possibles
- **Ergonomie** : facilité d'utilisation et d'apprentissage, interface utilisateur conviviale (voir le cours d'IHM)

Qualités internes

- **Maintenabilité** : le plus coûteux

Qualités internes

- **Maintenabilité** : le plus coûteux
- **Vérifiabilité** : liée aux possibilités de vérification et de mesurabilité du produit

Qualités internes

- **Maintenabilité** : le plus coûteux
- **Vérifiabilité** : liée aux possibilités de vérification et de mesurabilité du produit
- **Réutilisation** : reprise de tout ou partie pour la construction d'autres systèmes

Qualités internes

- **Maintenabilité** : le plus coûteux
- **Vérifiabilité** : liée aux possibilités de vérification et de mesurabilité du produit
- **Réutilisation** : reprise de tout ou partie pour la construction d'autres systèmes
- **Portabilité** : indépendance vis à vis de l'environnement

Qualités internes

- **Maintenabilité** : le plus coûteux
- **Vérifiabilité** : liée aux possibilités de vérification et de mesurabilité du produit
- **Réutilisation** : reprise de tout ou partie pour la construction d'autres systèmes
- **Portabilité** : indépendance vis à vis de l'environnement
- **Interopérabilité** : capacité à interagir avec d'autres systèmes

Illustrations

- Le changement de numération téléphonique de 8 à 10 chiffres à entrainer une évolution des logiciels gérant les centraux téléphoniques

Illustrations

- Le changement de numération téléphonique de 8 à 10 chiffres à entrainer une évolution des logiciels gérant les centraux téléphoniques
- Un simulateur de vol doit respecter les contraintes du système réel (inertie par exemple)

Illustrations

- Le changement de numération téléphonique de 8 à 10 chiffres à entrainer une évolution des logiciels gérant les centraux téléphoniques
- Un simulateur de vol doit respecter les contraintes du système réel (inertie par exemple)
- Un module de compte bancaire avec ses variations peut-être réutilisé pour la conception de plusieurs systèmes

Illustrations

- Le changement de numération téléphonique de 8 à 10 chiffres à entrainer une évolution des logiciels gérant les centraux téléphoniques
- Un simulateur de vol doit respecter les contraintes du système réel (inertie par exemple)
- Un module de compte bancaire avec ses variations peut-être réutilisé pour la conception de plusieurs systèmes
- Utiliser le même logiciel sous Windows ou Linux ? (oui avec les applications Java)

Illustrations

- Le changement de numération téléphonique de 8 à 10 chiffres à entrainer une évolution des logiciels gérant les centraux téléphoniques
- Un simulateur de vol doit respecter les contraintes du système réel (inertie par exemple)
- Un module de compte bancaire avec ses variations peut-être réutilisé pour la conception de plusieurs systèmes
- Utiliser le même logiciel sous Windows ou Linux ? (oui avec les applications Java)
- Système bancaire distribué qui doit opérer des transactions entre des établissements ni du même groupe ni du même pays

Qualités du processus

- **Productivité** : efficacité du processus

Qualités du processus

- **Productivité** : efficacité du processus
- **Prédictivité temporelle** : capacité à estimer les délais

Qualités du processus

- **Productivité** : efficacité du processus
- **Prédictivité temporelle** : capacité à estimer les délais
- **Visibilité** : le processus et l'état du développement doivent être faciles à comprendre

Qualités du processus

- **Productivité** : efficacité du processus
- **Prédictivité temporelle** : capacité à estimer les délais
- **Visibilité** : le processus et l'état du développement doivent être faciles à comprendre
- **Supporté** : par un ou des outils

Qualités du processus

- **Productivité** : efficacité du processus
- **Prédictivité temporelle** : capacité à estimer les délais
- **Visibilité** : le processus et l'état du développement doivent être faciles à comprendre
- **Supporté** : par un ou des outils
- **Maîtrisé** : par tous les acteurs du développement

Cycle de développement

- Un bon développement doit reposer sur une méthode et un cycle de développement

Cycle de développement

- Un bon développement doit reposer sur une méthode et un cycle de développement
- On identifie différentes phases dans un développement : analyse, conception, ...

Cycle de développement

- Un bon développement doit reposer sur une méthode et un cycle de développement
- On identifie différentes phases dans un développement : analyse, conception, ...
- Elles produisent en général des documents qui vont servir à une autre phase

Cycle de développement

- Un bon développement doit reposer sur une méthode et un cycle de développement
- On identifie différentes phases dans un développement : analyse, conception, ...
- Elles produisent en général des documents qui vont servir à une autre phase
- Le cycle décrit les enchaînements prévus des différentes phases

Cycle de développement

- Un bon développement doit reposer sur une méthode et un cycle de développement
- On identifie différentes phases dans un développement : analyse, conception, ...
- Elles produisent en général des documents qui vont servir à une autre phase
- Le cycle décrit les enchaînements prévus des différentes phases
- Il existe plusieurs cycles de vie du logiciel

Le développement dans sa globalité

- Autour de l'aspect purement logiciel on trouve :

Le développement dans sa globalité

- Autour de l'aspect purement logiciel on trouve :
  - Gestion administrative

Le développement dans sa globalité

- Autour de l'aspect purement logiciel on trouve :
  - Gestion administrative
  - Documentation

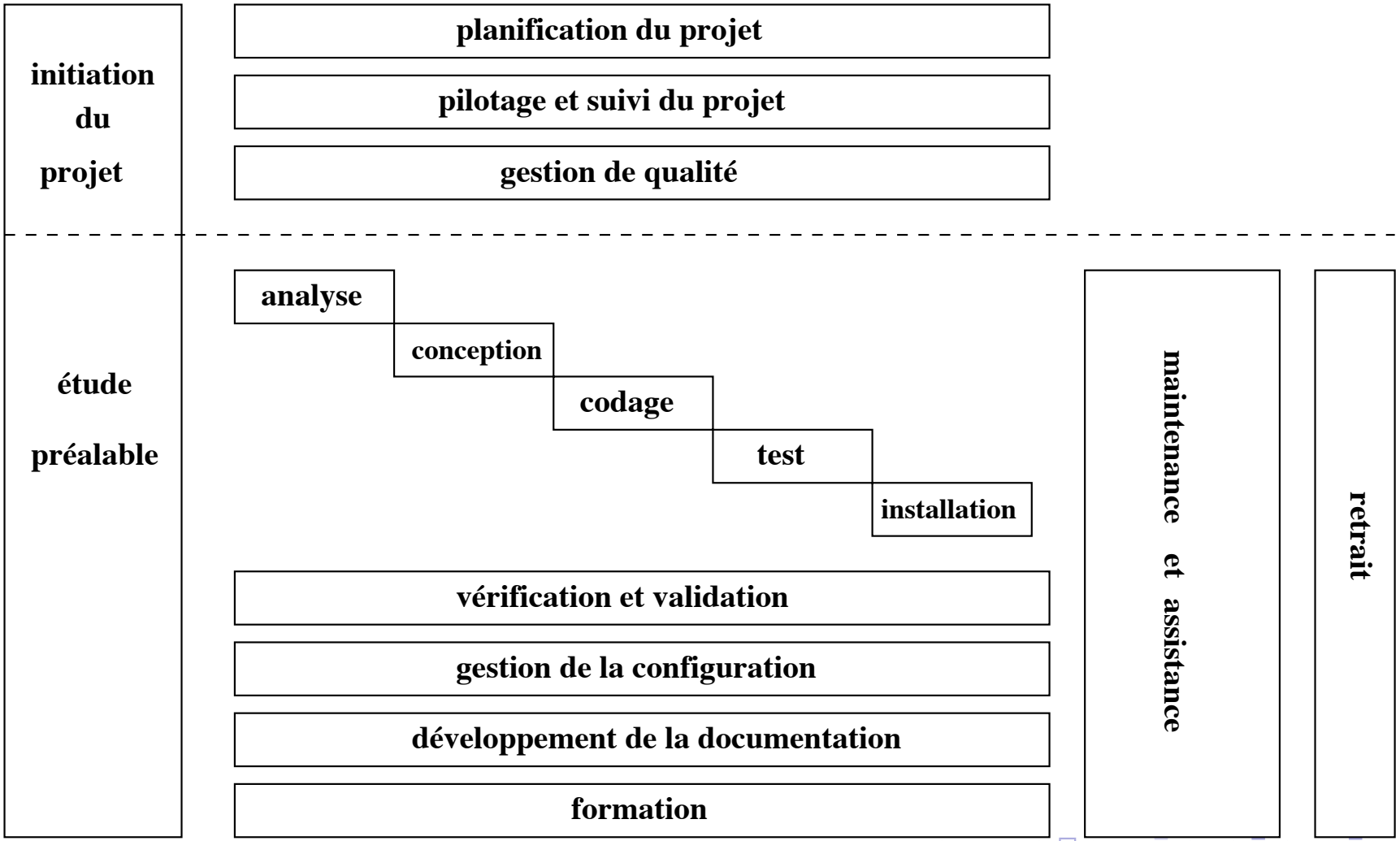
Le développement dans sa globalité

- Autour de l'aspect purement logiciel on trouve :
  - Gestion administrative
  - Documentation
  - Gestion des versions et configurations

Le développement dans sa globalité

- Autour de l'aspect purement logiciel on trouve :
  - Gestion administrative
  - Documentation
  - Gestion des versions et configurations
- Ces activités sont menées en parallèle avec le développement du logiciel proprement dit

Schéma des grandes tâches



Les autres activités 1/2

- Planification du projet : décomposition en sous-tâches

Les autres activités 1/2

- Planification du projet : décomposition en sous-tâches
- Suivi du projet : état d'avancement et contrôle

Les autres activités 1/2

- Planification du projet : décomposition en sous-tâches
- Suivi du projet : état d'avancement et contrôle
- Gestion de la qualité : mesure de la qualité, recommandations

Les autres activités 1/2

- Planification du projet : décomposition en sous-tâches
- Suivi du projet : état d'avancement et contrôle
- Gestion de la qualité : mesure de la qualité, recommandations
- La documentation est un élément essentiel du logiciel : de gestion du projet, technique ou de référence, d'utilisation, d'installation et d'exploitation

Les autres activités 2/2

- Configuration : maîtriser l'évolution du logiciel et de sa documentation

Les autres activités 2/2

- Configuration : maîtriser l'évolution du logiciel et de sa documentation
- Vérification et validation

Les autres activités 2/2

- Configuration : maîtriser l'évolution du logiciel et de sa documentation
- Vérification et validation
- Formation

Normes

- Classification des normes du génie logiciel ANSI/IEEE 1002

Normes

- Classification des normes du génie logiciel ANSI/IEEE 1002
- **Ingénierie du produit** : analyse des besoins, conception, codage, intégration, debogage, support du produit, maintenance.

Normes

- Classification des normes du génie logiciel ANSI/IEEE 1002
- **Ingénierie du produit** : analyse des besoins, conception, codage, intégration, debogage, support du produit, maintenance.
- **Vérification et validation** : revues et audits, analyse du produit, test

Normes

- Classification des normes du génie logiciel ANSI/IEEE 1002
- **Ingénierie du produit** : analyse des besoins, conception, codage, intégration, debogage, support du produit, maintenance.
- **Vérification et validation** : revues et audits, analyse du produit, test
- **Gestion technique** : gestion du processus, gestion du produit, gestion des ressources

Cahier des charges

- Contient le contexte, les besoins et leur évolution

Cahier des charges

- Contient le contexte, les besoins et leur évolution
- Spécifications fonctionnelles

Cahier des charges

- Contient le contexte, les besoins et leur évolution
- Spécifications fonctionnelles
- Document de référence du produit

Cahier des charges

- Contient le contexte, les besoins et leur évolution
- Spécifications fonctionnelles
- Document de référence du produit
- Notion de client-fournisseur

Cahier des charges

- Contient le contexte, les besoins et leur évolution
- Spécifications fonctionnelles
- Document de référence du produit
- Notion de client-fournisseur
- Informel !

Cahier des charges

- Contient le contexte, les besoins et leur évolution
- Spécifications fonctionnelles
- Document de référence du produit
- Notion de client-fournisseur
- Informel !
- La validation n'est donc jamais une preuve formelle de l'adéquation

Un modèle

Positionnement et objectifs : dans quel contexte le cahier des charges est-il établi et quels sont ces objectifs

Un modèle

Positionnement et objectifs : dans quel contexte le cahier des charges est-il établi et quels sont ces objectifs

Spécifications applicatives ou fonctionnelles : descriptions des données, traitements, présentations, interfaces

Un modèle

Positionnement et objectifs : dans quel contexte le cahier des charges est-il établi et quels sont ces objectifs

Spécifications applicatives ou fonctionnelles : descriptions des données, traitements, présentations, interfaces

Spécifications techniques : informations plus spécifiques sur le contexte

Un modèle

Positionnement et objectifs : dans quel contexte le cahier des charges est-il établi et quels sont ces objectifs

Spécifications applicatives ou fonctionnelles : descriptions des données, traitements, présentations, interfaces

Spécifications techniques : informations plus spécifiques sur le contexte

Contraintes de réalisation :

Un modèle

Positionnement et objectifs : dans quel contexte le cahier des charges est-il établi et quels sont ces objectifs

Spécifications applicatives ou fonctionnelles : descriptions des données, traitements, présentations, interfaces

Spécifications techniques : informations plus spécifiques sur le contexte

Contraintes de réalisation :

Spécifications administratives : budget, délais, propriété, clauses légales

Un modèle

Positionnement et objectifs : dans quel contexte le cahier des charges est-il établi et quels sont ces objectifs

Spécifications applicatives ou fonctionnelles : descriptions des données, traitements, présentations, interfaces

Spécifications techniques : informations plus spécifiques sur le contexte

Contraintes de réalisation :

Spécifications administratives : budget, délais, propriété, clauses légales

Spécifications d'évaluation : les éléments qui permettront de valider l'offre faite par le futur contractant

Exemple : le carnet d'adresse

Objectifs : la société machin veut gérer un carnet d'adresse de ses clients et intégrer ceci dans son système d'information ...

Exemple : le carnet d'adresse

Objectifs : la société machin veut gérer un carnet d'adresse de ses clients et intégrer ceci dans son système d'information ...

Fonctions : Elle veut pouvoir créer des fiches de ses contacts avec des informations, pouvoir retrouver une fiche, la modifier ou bien la supprimer. Tout ceci doit pouvoir se faire d'une façon agréable avec utilisation de menus, souris et clavier, point sur lequel il a plein d'idées originales ...

Exemple : le carnet d'adresse

Objectifs : la société machin veut gérer un carnet d'adresse de ses clients et intégrer ceci dans son système d'information ...

Fonctions : Elle veut pouvoir créer des fiches de ses contacts avec des informations, pouvoir retrouver une fiche, la modifier ou bien la supprimer. Tout ceci doit pouvoir se faire d'une façon agréable avec utilisation de menus, souris et clavier, point sur lequel il a plein d'idées originales ...

Techniques : ce sera sous système CPM, avec 10KO de RAM et un processeur à relais électromagnétiques !

suite

Contraintes : standard XXL et pas le droit de fumer ni de manger dans les locaux.

suite

Contraintes : standard XXL et pas le droit de fumer ni de manger dans les locaux.

Administration : 3 mois sur place, 4 personnes avec portables, 10MF en 10 versements ...

suite

Contraintes : standard XXL et pas le droit de fumer ni de manger dans les locaux.

Administration : 3 mois sur place, 4 personnes avec portables, 10MF en 10 versements ...

Evaluation : tests intensifs en interne pendant un mois sans restriction

Le CDC selon J-R. Abrial

Le cahier des charges d'un système informatique est constitué par l'ensemble des documents qui rassemblent, du point de vue du client, tous les éléments techniques à partir desquels un futur contractant va pouvoir réaliser le système en question.
A côté de cette finalité technique, le cahier remplit aussi souvent un rôle contractuel : il contient en fait une certaine définition "légale" du système à réaliser. cela signifie que tout litige, qui pourrait apparaître entre le client et le contractant au sujet du système finale, devrait pouvoir être tranché, en dernière analyse, après consultation du cahier.
Cette double caractéristique (technique et contractuelle) du cahier des charges justifie son importance et donc le soin que l'on doit apporter à son écriture.

Analyse

- Terme général désignant la compréhension et la formulation d'un problème

Analyse

- Terme général désignant la compréhension et la formulation d'un problème
- Une analyse part du cahier des charges et produit une spécification

Analyse

- Terme général désignant la compréhension et la formulation d'un problème
- Une analyse part du cahier des charges et produit une spécification
- L'analyse des besoins produit des spécifications fonctionnelles

Analyse

- Terme général désignant la compréhension et la formulation d'un problème
- Une analyse part du cahier des charges et produit une spécification
- L'analyse des besoins produit des spécifications fonctionnelles
- Analyse formelle et spécification formelle

Analyse

- Terme général désignant la compréhension et la formulation d'un problème
- Une analyse part du cahier des charges et produit une spécification
- L'analyse des besoins produit des spécifications fonctionnelles
- Analyse formelle et spécification formelle
- Langage de spécification semi-formel et formel

Analyse

- Terme général désignant la compréhension et la formulation d'un problème
- Une analyse part du cahier des charges et produit une spécification
- L'analyse des besoins produit des spécifications fonctionnelles
- Analyse formelle et spécification formelle
- Langage de spécification semi-formel et formel
- Attention une **spécification décrit seulement le problème**

Spécification formelle

- Description d'un problème dans un langage ayant une sémantique formelle (une approximation parlante est *mathématique*)

Spécification formelle

- Description d'un problème dans un langage ayant une sémantique formelle (une approximation parlante est *mathématique*)
- Permet des vérifications, traduction automatique en code, tests automatiques, documentation, rigueur, ...

Spécification formelle

- Description d'un problème dans un langage ayant une sémantique formelle (une approximation parlante est *mathématique*)
- Permet des vérifications, traduction automatique en code, tests automatiques, documentation, rigueur, ...
- Important pour le développement de logiciel dit *critique*

Spécification formelle

- Description d'un problème dans un langage ayant une sémantique formelle (une approximation parlante est *mathématique*)
- Permet des vérifications, traduction automatique en code, tests automatiques, documentation, rigueur, ...
- Important pour le développement de logiciel dit *critique*
- Approche importante : transformation formelle

Spécification formelle

- Description d'un problème dans un langage ayant une sémantique formelle (une approximation parlante est *mathématique*)
- Permet des vérifications, traduction automatique en code, tests automatiques, documentation, rigueur, ...
- Important pour le développement de logiciel dit *critique*
- Approche importante : transformation formelle
- Mais ce n'est pas si simple ...

La conception

- Activité destinée à produire **une solution à un problème**

La conception

- Activité destinée à produire **une solution à un problème**
- Elle part du résultat de l'analyse

La conception

- Activité destinée à produire **une solution à un problème**
- Elle part du résultat de l'analyse
- Son résultat est un ensemble de documents décrivant : architecture, solutions fonctionnelles (algorithmes), considérations de performances, format des données d'entrée et de sortie ...

La conception

- Activité destinée à produire **une solution à un problème**
- Elle part du résultat de l'analyse
- Son résultat est un ensemble de documents décrivant : architecture, solutions fonctionnelles (algorithmes), considérations de performances, format des données d'entrée et de sortie ...
- Conception détaillée

Codage

- Implémentation du résultat de la conception dans un langage exécutable (codage, implémentation, programmation ...)

Codage

- Implémentation du résultat de la conception dans un langage exécutable (codage, implémentation, programmation ...)
- Il s'agit du codage dans un langage de programmation, *i.e.*, de la traduction de la conception dans un langage

Codage

- Implémentation du résultat de la conception dans un langage exécutable (codage, implémentation, programmation ...)
- Il s'agit du codage dans un langage de programmation, *i.e.*, de la traduction de la conception dans un langage
- Activité bien connue des programmeurs

Intégration

- Le logiciel doit interagir avec d'autres éléments

Intégration

- Le logiciel doit interagir avec d'autres éléments
- Système intégré = composé et assemblé

Intégration

- Le logiciel doit interagir avec d'autres éléments
- Système intégré = composé et assemblé
- Il peut y avoir intégration matériel et logiciel (co-conception)

Intégration

- Le logiciel doit interagir avec d'autres éléments
- Système intégré = composé et assemblé
- Il peut y avoir intégration matériel et logiciel (co-conception)
- Éventuellement ce matériel peut être non informatique (robotique, capteurs, etc)

Intégration

- Le logiciel doit interagir avec d'autres éléments
- Système intégré = composé et assemblé
- Il peut y avoir intégration matériel et logiciel (co-conception)
- Éventuellement ce matériel peut être non informatique (robotique, capteurs, etc)
- L'aspect architecture matérielle du système est apparente ici

Débogage

- Détection et correction des erreurs

Débogage

- Détection et correction des erreurs
- Activité classique en programmation

Débogage

- Détection et correction des erreurs
- Activité classique en programmation
- Activité coûteuse et assez “artistique”

Débogage

- Détection et correction des erreurs
- Activité classique en programmation
- Activité coûteuse et assez “artistique”
- Se base sur des traces d’exécution ou des tests

Débogage

- Détection et correction des erreurs
- Activité classique en programmation
- Activité coûteuse et assez “artistique”
- Se base sur des traces d’exécution ou des tests
- Les erreurs sont souvent détectées suite à des tests effectués par le programmeur mais aussi par le client avant ou après la livraison

Débogage

- Détection et correction des erreurs
- Activité classique en programmation
- Activité coûteuse et assez “artistique”
- Se base sur des traces d’exécution ou des tests
- Les erreurs sont souvent détectées suite à des tests effectués par le programmeur mais aussi par le client avant ou après la livraison
- Les séries de tests peuvent être parfois automatisées

Débogage, oui mais ...

- **Insuffisance fondamentale** : trouve des erreurs mais ne prouve pas qu'il n'y en a plus

Débogage, oui mais ...

- **Insuffisance fondamentale** : trouve des erreurs mais ne prouve pas qu'il n'y en a plus
- Problème avec les systèmes infinis ou très gros en particulier, tests non exhaustifs

Débogage, oui mais ...

- **Insuffisance fondamentale** : trouve des erreurs mais ne prouve pas qu'il n'y en a plus
- Problème avec les systèmes infinis ou très gros en particulier, tests non exhaustifs
- Toutefois des outils et des méthodes indispensables en pratique

Maintenance

- Activités destinés à améliorer, adapter ou corriger un logiciel

Maintenance

- Activités destinés à améliorer, adapter ou corriger un logiciel
- *Perfectionnement* : amélioration des performances, pas de changement fonctionnel

Maintenance

- Activités destinés à améliorer, adapter ou corriger un logiciel
- *Perfectionnement* : amélioration des performances, pas de changement fonctionnel
- *Adaptation* : à l'évolution de l'environnement

Maintenance

- Activités destinés à améliorer, adapter ou corriger un logiciel
- *Perfectionnement* : amélioration des performances, pas de changement fonctionnel
- *Adaptation* : à l'évolution de l'environnement
- *Correction* : débogage alors que le produit est livré et en utilisation

Inspection ou revue de code

- Pas une phase du cycle mais une activité importante pour la qualité du logiciel

Inspection ou revue de code

- Pas une phase du cycle mais une activité importante pour la qualité du logiciel
- Peut prendre des formes très diverses

Inspection ou revue de code

- Pas une phase du cycle mais une activité importante pour la qualité du logiciel
- Peut prendre des formes très diverses
- Peut s'appliquer à tous les produits du cycle de développement

Inspection ou revue de code

- Pas une phase du cycle mais une activité importante pour la qualité du logiciel
- Peut prendre des formes très diverses
- Peut s'appliquer à tous les produits du cycle de développement
- Analyse manuelle avec une ou un groupe de personnes

Inspection ou revue de code

- Pas une phase du cycle mais une activité importante pour la qualité du logiciel
- Peut prendre des formes très diverses
- Peut s'appliquer à tous les produits du cycle de développement
- Analyse manuelle avec une ou un groupe de personnes
- Un guide des règles de dévelopement en usage dans l'entreprise (standard, norme, etc)

Inspection ou revue de code

- Pas une phase du cycle mais une activité importante pour la qualité du logiciel
- Peut prendre des formes très diverses
- Peut s'appliquer à tous les produits du cycle de développement
- Analyse manuelle avec une ou un groupe de personnes
- Un guide des règles de dévelopement en usage dans l'entreprise (standard, norme, etc)
- Souvent avec interaction des auteurs

Inspection ou revue de code

- Pas une phase du cycle mais une activité importante pour la qualité du logiciel
- Peut prendre des formes très diverses
- Peut s'appliquer à tous les produits du cycle de développement
- Analyse manuelle avec une ou un groupe de personnes
- Un guide des règles de dévelopement en usage dans l'entreprise (standard, norme, etc)
- Souvent avec interaction des auteurs
- Groupe hétérogène avec programmeur expérimenté

suite ...

- Lisent les codes sources et signalent des erreurs ou les doutes

suite ...

- Lisent les codes sources et signalent des erreurs ou les doutes
- Peuvent tester manuellement des cas

suite ...

- Lisent les codes sources et signalent des erreurs ou les doutes
- Peuvent tester manuellement des cas
- Prouvé comme étant une activité essentielle pour la qualité !

suite ...

- Lisent les codes sources et signalent des erreurs ou les doutes
- Peuvent tester manuellement des cas
- Prouvé comme étant une activité essentielle pour la qualité !
- Des estimations donnent de 50 à 90 % d'élimination de problèmes

suite ...

- Lisent les codes sources et signalent des erreurs ou les doutes
- Peuvent tester manuellement des cas
- Prouvé comme étant une activité essentielle pour la qualité !
- Des estimations donnent de 50 à 90 % d'élimination de problèmes
- Contre 30 % avec des tests !

suite ...

- Lisent les codes sources et signalent des erreurs ou les doutes
- Peuvent tester manuellement des cas
- Prouvé comme étant une activité essentielle pour la qualité !
- Des estimations donnent de 50 à 90 % d'élimination de problèmes
- Contre 30 % avec des tests !
- `http://www-lor.int-evry.fr/~raffy/Poly/v&v.htm`

suite ...

- Lisent les codes sources et signalent des erreurs ou les doutes
- Peuvent tester manuellement des cas
- Prouvé comme étant une activité essentielle pour la qualité !
- Des estimations donnent de 50 à 90 % d'élimination de problèmes
- Contre 30 % avec des tests !
- `http://www-lor.int-evry.fr/~raffy/Poly/v&v.htm`
- wikipedia

suite ...

- Lisent les codes sources et signalent des erreurs ou les doutes
- Peuvent tester manuellement des cas
- Prouvé comme étant une activité essentielle pour la qualité !
- Des estimations donnent de 50 à 90 % d'élimination de problèmes
- Contre 30 % avec des tests !
- `http://www-lor.int-evry.fr/~raffy/Poly/v&v.htm`
- wikipedia
- `http://www.site.uottawa.ca/~ssome/Cours/SEG3603/reviews.pdf`

Atelier de Génie Logiciel

- AGL ou environnement de programmation

Atelier de Génie Logiciel

- AGL ou environnement de programmation
- Ensemble d'outils logiciels et matériels assistant le développement

Atelier de Génie Logiciel

- AGL ou environnement de programmation
- Ensemble d'outils logiciels et matériels assistant le développement
- Il a pour but d'aider à réaliser, contrôler et suivre un logiciel tout au long de son cycle de vie

Atelier de Génie Logiciel

- AGL ou environnement de programmation
- Ensemble d'outils logiciels et matériels assistant le développement
- Il a pour but d'aider à réaliser, contrôler et suivre un logiciel tout au long de son cycle de vie
- Il doit faciliter la coordination des différentes tâches, de mesurer et d'améliorer la qualité du produit

Atelier de Génie Logiciel

- AGL ou environnement de programmation
- Ensemble d'outils logiciels et matériels assistant le développement
- Il a pour but d'aider à réaliser, contrôler et suivre un logiciel tout au long de son cycle de vie
- Il doit faciliter la coordination des différentes tâches, de mesurer et d'améliorer la qualité du produit
- Voir `campus` : projet OSE3 ou cours IHM de C. Dumas

Atelier de Génie Logiciel

- AGL ou environnement de programmation
- Ensemble d'outils logiciels et matériels assistant le développement
- Il a pour but d'aider à réaliser, contrôler et suivre un logiciel tout au long de son cycle de vie
- Il doit faciliter la coordination des différentes tâches, de mesurer et d'améliorer la qualité du produit
- Voir `campus` : projet OSE3 ou cours IHM de C. Dumas
- Mais aussi la suite de ce cours ...

Les “approches” du développement

- Modèle de la cascade et cycle en V

Les “approches” du développement

- Modèle de la cascade et cycle en V
- Programmation exploratoire

Les “approches” du développement

- Modèle de la cascade et cycle en V
- Programmation exploratoire
- Prototypage

Les “approches” du développement

- Modèle de la cascade et cycle en V
- Programmation exploratoire
- Prototypage
- Par réutilisation

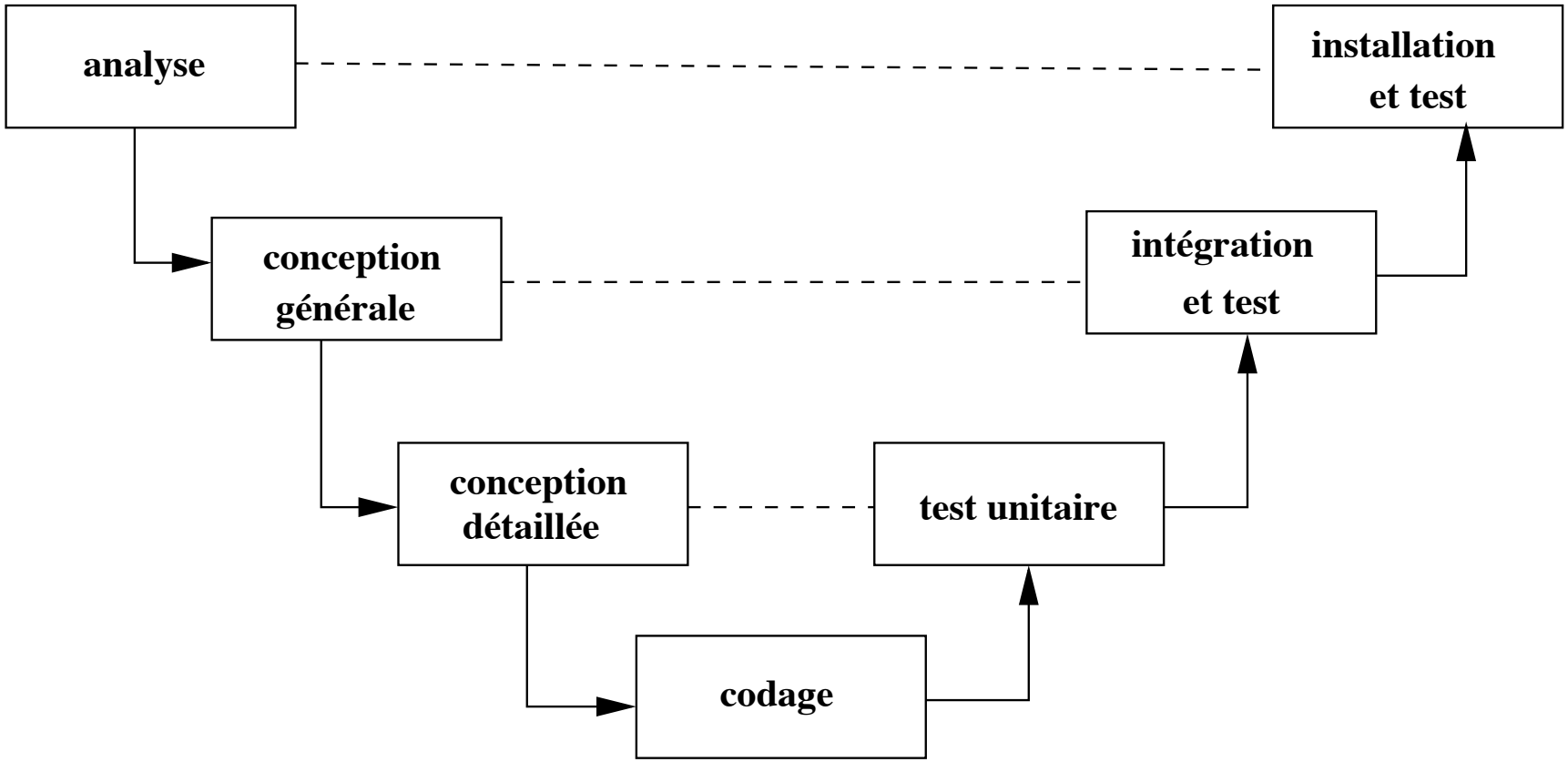
Les “approches” du développement

- Modèle de la cascade et cycle en V
- Programmation exploratoire
- Prototypage
- Par réutilisation
- Transformation formelle (je passe !)

Les “approches” du développement

- Modèle de la cascade et cycle en V
- Programmation exploratoire
- Prototypage
- Par réutilisation
- Transformation formelle (je passe !)
- Autres : hybride, par exemple le modèle en spirale ou RUP

Le cycle en V


analyse
installation et test
conception générale
intégration et test
conception détaillée
test unitaire
codage

Programmation exploratoire

- Part d'une spécification informelle

Programmation exploratoire

- Part d'une spécification informelle
- Itère codage et évaluation par tests

Programmation exploratoire

- Part d'une spécification informelle
- Itère codage et évaluation par tests
- RAD, par l'exemple, eXtreme programming

Programmation exploratoire

- Part d'une spécification informelle
- Itère codage et évaluation par tests
- RAD, par l'exemple, eXtreme programming
- Processus trop simple

Programmation exploratoire

- Part d'une spécification informelle
- Itère codage et évaluation par tests
- RAD, par l'exemple, eXtreme programming
- Processus trop simple
- Prévision des coûts impossible

Programmation exploratoire

- Part d'une spécification informelle
- Itère codage et évaluation par tests
- RAD, par l'exemple, eXtreme programming
- Processus trop simple
- Prévision des coûts impossible
- Mauvaise qualité du produit

Programmation exploratoire

- Part d'une spécification informelle
- Itère codage et évaluation par tests
- RAD, par l'exemple, eXtreme programming
- Processus trop simple
- Prévision des coûts impossible
- Mauvaise qualité du produit
- Souvent la seule connue des amateurs

Le prototypage

- Prototype : logiciel développé rapidement sans soucis de performance

Le prototypage

- Prototype : logiciel développé rapidement sans soucis de performance
- Jetable ou évolutif

Le prototypage

- Prototype : logiciel développé rapidement sans soucis de performance
- Jetable ou évolutif
- Intéressant car permet de compléter les spécifications au faire et à mesure de l'évolution

Le prototypage

- Prototype : logiciel développé rapidement sans soucis de performance
- Jetable ou évolutif
- Intéressant car permet de compléter les spécifications au faire et à mesure de l'évolution
- Processus peu structuré, difficile à gérer en général

Le prototypage

- Prototype : logiciel développé rapidement sans soucis de performance
- Jetable ou évolutif
- Intéressant car permet de compléter les spécifications au faire et à mesure de l'évolution
- Processus peu structuré, difficile à gérer en général
- Peu servir à une meilleure compréhension des besoins ou des interfaces

Par réutilisation

- On définit une architecture à partir d'autres éléments logiciels disponibles

Par réutilisation

- On définit une architecture à partir d'autres éléments logiciels disponibles
- Il faut assembler le tout

Par réutilisation

- On définit une architecture à partir d'autres éléments logiciels disponibles
- Il faut assembler le tout
- Processus gérable et économique

Par réutilisation

- On définit une architecture à partir d'autres éléments logiciels disponibles
- Il faut assembler le tout
- Processus gérable et économique
- Difficulté : interopérabilité, fabrication des composants, recherche et adaptation des composants

Par réutilisation

- On définit une architecture à partir d'autres éléments logiciels disponibles
- Il faut assembler le tout
- Processus gérable et économique
- Difficulté : interopérabilité, fabrication des composants, recherche et adaptation des composants
- Techniques à base de *patron*, de généricité, de template, ...

Par réutilisation

- On définit une architecture à partir d'autres éléments logiciels disponibles
- Il faut assembler le tout
- Processus gérable et économique
- Difficulté : interopérabilité, fabrication des composants, recherche et adaptation des composants
- Techniques à base de *patron*, de généricité, de template, ...
- Un des principes de base de lignes de produits logiciels

Cycle en spirale