

Le Génie Logiciel

Zineb Rachik

Ecole Hassania des travaux publics

Année universitaire : 2016-2017

Objectifs du cours

- Être en mesure d'exprimer des besoins et de suivre un développement de logiciel.
- Pouvoir s'intégrer dans une équipe de professionnels.
- Travail en équipe sur des projets longs et complexes
- Spécifications de départ peu précises
- Dialogue avec le client/utilisateur : parler métier et non informatique
- Organisation, planification, gestion du risque

Naissance du premier programme

- Entre 1842 et 1843, Ada Lovelace traduisait le mémoire d'un mathématicien italien sur la machine analytique. Elle avait ajouté ses propres notes dont l'une décrivait de façon détaillée une séquence progressive d'opérations pour résoudre certains problèmes mathématiques. Le premier programme était né.

Programmation au fil des années

- au début des années 1950 les langages d'assemblage ont été inventés.
- L'assembleur améliorait nettement la situation grâce à un premier pas vers l'abstraction
- un défaut fondamental subsistait : un programme en assembleur est écrit en termes de *ce que sait faire la machine* ; or, on souhaite une expression en termes de *ce que veut faire l'utilisateur*.

Programmation au fil des années

- Au cours de la décennie 1950-1960, la programmation des ordinateurs a connu une véritable révolution, passant d'un codage en notation binaire à l'usage de langages de programmation de haut niveau, ancêtres de ceux que nous connaissons aujourd'hui.

Naissance des langage de haut niveau

- De 1957 à 1959 on note le développement des langages Fortran, Lisp et COBOL. À partir de 1970 on assiste à la naissance du Pascal.
- À la fin des années 1960 éclate la « crise du logiciel », prise de conscience des difficultés que rencontre le développement des grands projets informatiques.

La crise du logiciel

- Dans les années 1960, l'informatique conquiert de nouveaux champs d'application.
- Elle s'impose dans la gestion des entreprises.
- Naissance des sociétés de services informatiques.
- La production de logiciels devient l'objet d'une industrie spécifique.
- La taille des systèmes informatique ainsi que leur complexité connaissent une croissance exponentielle.

La crise du logiciel

- La qualité du logiciel livré est souvent déficiente
- Les performances étaient très souvent médiocres
- Les coûts de développement d'un logiciel sont excessifs
- L'invisibilité du logiciel,
- La maintenance du logiciel est difficile, coûteuse et souvent à l'origine de nouvelles erreurs
- Il est rare qu'on puisse réutiliser un logiciel existant ou un de ses composants pour confectionner un nouveau système, même si celui-ci comporte des fonctions similaires.

Erreurs célèbres

- La sonde Mariner qui devait effectuer un passage à 5 000 km de Vénus s'est perdue à 5 000 000 km de ladite planète à cause d'une erreur de programme Fortran : le point avait été remplacé par une virgule.

Erreurs célèbres

- Le 4 juin 1996, lors du premier lancement de la fusée Ariane V, celle-ci explose en vol. La cause : un logiciel utilisé sous Ariane IV et intégré sans nouvelle validation dans Ariane V. Ariane V ayant des moteurs plus puissants s'incline plus rapidement que Ariane IV, pour récupérer l'accélération due à la rotation de la Terre. Les capteurs ont bien détecté cette inclinaison d'Ariane V, mais le logiciel l'a jugée non conforme à son plan de tir, et a provoqué l'ordre d'autodestruction alors que tout se passait bien. Coût : 1/2 milliard de dollars

Erreurs célèbres

- En 1983, toute la vallée du Colorado a été inondée. La cause ? Une mauvaise modélisation dans le logiciel du temps d'ouverture du barrage.
- Bug du Pentium en 1994. Coût : 500 millions de dollars
- Bug de l'an 2000. Coût de la mise à niveau des logiciels à la France : 500 milliards de francs.

Erreurs célèbres

- Le 22 décembre 2001, en pleine période des achats de Noël, les 750 000 terminaux de paiement ne répondent plus chez les commerçants, entraînant de longues files d'attente de clients excédés dont beaucoup finissent par abandonner leurs chariots. La cause : la saturation des serveurs de la société Atos en charge des autorisations de paiement dépassant 600F. Les autorisations de débit qui prennent habituellement quelques dizaines de secondes, nécessitent ce jour-là quasiment la demi-heure. Le coût du préjudice pour le seul groupe Leclerc : 2 millions d'Euros.

Une solution?

- une prise de conscience se manifeste dans la communauté informatique : la production de logiciel manque du fondement rationnel présent dans d'autres techniques plus anciennes

Une solution?

- De manière analogue au génie civil, au génie électrique et au génie chimique, il faut promouvoir un génie logiciel.
- La division des affaires scientifiques de l'OTAN organise en octobre 1968, une conférence de travail sur les difficultés de la production de logiciel et les moyens de les surmonter.
Intitulée *Working Conference on Software Engineering*, traduit en français par « génie logiciel ».

Qu'est ce que c'est génie?

- les connaissances et techniques de l'ingénieur.
- Synonyme de science de l'ingénieur.

Qu'est ce que c'est logiciel?

- Mot inventé en 1967 par Philippe Renard
- La partie « soft » de l'ordinateur
- Un ensemble de programmes apportant des fonctionnalités supplémentaires à la machine.

Types de logiciels

- Les logiciels propriétaires et payants.
- Les logiciels gratuits.
- Les logiciels libres.

Types de logiciels

- Sur mesure (custom)
 - Pour un client spécifique
- Générique (generic)
 - Vendu sur le marché
 - un tableur (spreadsheet), un outil de base de donnée (database)
 - un outil de traitement de texte (word processor)
 - ...
- Embarqués (embedded)
 - exécutent dans du matériel électronique isolé
machine à laver, télévision, lecteur DVD, téléphone mobile,
magnétoscope, four à micro-ondes, réfrigérateur, joueur MP3, ...
 - Difficile à modifier

Types de logiciels

- Logiciel à temps réel (real-time)
 - systèmes de contrôle et de surveillance
 - manipulent et contrôlent le matériel technique
 - Réaction immédiate requise
 - Environnement souvent très contraignant

Types de logiciels

- Logiciel de traitement de données (data processing)
 - Ils stockent, recherchent, transforment et présentent l'information aux utilisateurs
 - Grandes quantités de données avec des corrélations complexes, enregistrées dans les bases de données
 - Largement utilisés en administration des affaires
 - Fiabilité des résultats
 - Sécurité dans l'accès aux données
- Quelques fois les 2 aspects sont présents dans un logiciel

Types de logiciels

- Les systèmes distribués
 - synchronisent la transmission, assurent l'intégrité des données et la sécurité, ...
 - Technologies utilisées
CORBA, DOM/DCOM, SOAP, EJB, ...
- Les systèmes de matériel
 - Systèmes d'exploitation, exécutions de matériel de bas niveau
- Les systèmes d'entreprise
décrivent les buts, les ressources, les règles et le travail réel dans une entreprise

Qu'est ce que c'est Génie logiciel

- Le génie logiciel est donc l'art de spécifier, de concevoir, de réaliser, et de faire évoluer, avec des moyens et dans des délais raisonnables, des programmes, des documentations et des procédures de qualité en vue d'utiliser un ordinateur pour résoudre certains problèmes.

Qualité exigée d'un logiciel

Bertrand Meyer Conception et programmation orientées objet
Editions Eyrolles, 2000

- validité La validité (correction, justesse, conformité) est la capacité que possède un produit logiciel à remplir exactement ses fonctions, définies par le cahier des charges et les spécifications.
- Adéquation entre : Le besoin effectif de l'utilisateur Les fonctions offertes par le logiciel
- Solutions : Emphase sur l'analyse des besoins Améliorer la communication (langage commun, démarche participative)
Travailler avec rigueur

Qualité exigée d'un logiciel

- Fiabilité ou Robustesse La robustesse (fiabilité, sûreté) est la capacité qu'offrent des systèmes logiciels à réagir de manière appropriée à la présence de conditions anormales (i.e. rien de catastrophique ne peut survenir, même en dehors des conditions d'utilisation prévues).
- Solutions : Utiliser des méthodes formelles, des langages et des méthodes de programmation de haut niveau Vérifications, tests

Qualité exigée d'un logiciel

- Facilité d'utilisation La facilité d'utilisation est la facilité avec laquelle des personnes présentant des formations et des compétences différentes peuvent apprendre à utiliser les produits logiciels et s'en servir pour résoudre des problèmes. Elle recouvre également la facilité d'installation, d'opération et de contrôle.
- Facilité d'apprentissage : comprendre ce que l'on peut faire avec le logiciel, et savoir comment le faire Facilité d'utilisation : importance de l'effort nécessaire pour utiliser le logiciel à des fins données Solutions : Analyse du mode opératoire des utilisateurs Adapter l'ergonomie des logiciels aux utilisateurs

Qualité exigée d'un logiciel

- Compatibilité La compatibilité est la facilité avec laquelle des éléments logiciels peuvent être combinés à d'autres.
- Compatibilité, interopérabilité, couplabilité
- Un logiciel doit pouvoir interagir en synergie avec d'autres logiciels
- Solutions : Bases de données (découplage données/traitements)
« Externaliser » certaines fonctions en utilisant des «
Middleware » avec une API (Application Program Interface)
bien définie Standardisation des formats de fichiers (XML...)
et des protocoles de communication (CORBA...) Les ERP
(Entreprise Resources Planning)

Qualité exigée d'un logiciel

- Efficacité, performance
- Les logiciels doivent satisfaire aux contraintes de temps d'exécution
- Solutions : Logiciels plus simples Veiller à la complexité des algorithmes Machines plus performantes

Qualité exigée d'un logiciel

- Portabilité
- La portabilité est la facilité avec laquelle des produits logiciels peuvent être transférés d'un environnement logiciel ou matériel à l'autre
- Un même logiciel doit pouvoir fonctionner sur plusieurs machines
- Solutions : Rendre le logiciel indépendant de son environnement d'exécution (voir interopérabilité) Machines virtuelles

Qualité exigée d'un logiciel

- Réutilisabilité
- La réutilisabilité est la capacité des éléments logiciels à servir à la construction de plusieurs applications différentes
- On peut espérer des gains considérables car dans la plupart des logiciels :
- 80 % du code est du « tout venant » qu'on retrouve à peu près partout
- 20 % du code est spécifique
- Solutions : Abstraction, généricité Construire un logiciel à partir de composants prêts à l'emploi Design Patterns

Qualité exigée d'un logiciel

- Maintenabilité
- La maintenabilité est le degré de facilité de la maintenance d'un produit logiciel.
- Facilité de maintenance
- Un logiciel ne s'use pas
- Pourtant, la maintenance absorbe une très grosse partie des efforts de développement (représente 67 % de l'effort de développement) ;
- Les coûts de maintenance se jouent très tôt dans le processus d'élaboration du logiciel
- Solution : Réutilisabilité, modularité Vérifier, tester Anticiper les changements à venir.

Qualité exigée d'un logiciel

- Extensibilité
- L'extensibilité est la facilité d'adaptation des produits logiciels aux changements de spécifications.
- Intégrité
- Aptitude d'un logiciel à protéger son code et ses données contre des accès non autorisé.
- Ponctualité
- La ponctualité est la capacité d'un système logiciel à être livré au moment désiré par ses utilisateurs, ou avant.

Principes du génie logiciel

- Généralisation:
- Structuration
- Abstraction
- Modularité
- Documentation
- Vérification

Cycle de vie du logiciel:

Définition

- Cycle de vie : ensemble des étapes de la réalisation, de l'énoncé des besoins à la maintenance ou au retrait du logiciel de façon générale,
- C'est la période de temps s'étalant du début à la fin du processus du logiciel.

Cycle de vie du logiciel: Etapes

- Définition et analyse des besoins: spécification
- Conception architecturale et élaboration des tests d'intégration
- Conception détaillée et élaboration des tests unitaires
- Codage (production du code source)
- Tests unitaires et d'intégration

Cycle de vie du logiciel: Etapes

- Implantation chez le commanditaire, essais avec les utilisateurs et validation
- Formation des utilisateurs, utilisation, maintenance, évolution
- Retrait Ces étapes ne doivent pas être vues comme se succédant les unes aux autres de façon linéaire.

Spécifications: Etude préalable

- Comment procéder pour faire ce développement?
- Quels moyens faut-il mettre en oeuvre?
- Dresser un état de l'existant et faire une analyse de ses forces et faiblesses;
- Identifier les idées ou besoins de l'utilisateur;
- Formuler des solutions potentielles;
- Faire des études de faisabilité;
- Planifier la transition entre l'ancien logiciel et le nouveau, s'il y a lieu;
- Affiner ou finaliser l'énoncé des besoins de l'utilisateur.
-

Spécifications: Analyse

- description de l'environnement du logiciel;
- spécification fonctionnelle (functional specification), qui définit toutes les fonctions que le logiciel doit offrir;
- comportement en cas d'erreurs, c'est-à-dire dans les cas où le logiciel ne peut pas accomplir une fonction;
- performances requises (performance requirements), par exemple : temps de réponse, encombrement en mémoire, sécurité de fonctionnement;
- interfaces avec l'utilisateur (user interface), en particulier le dialogue sur terminal, la présentation des écrans, la disposition des états imprimés, etc.
- interfaces avec d'autres logiciels;
- interfaces avec le matériel;
- contraintes de réalisation, telles que l'environnement de développement, le langage de programmation à utiliser, les procédures et normes à suivre, etc.

Spécifications: Cahier de charge

- 1. introduction : présentation générale, motivations, définitions des termes
- 2. contexte : environnement matériel et humain, acteurs et utilisateurs, interaction avec d'autres systèmes et logiciels, existant
- 3. spécifications fonctionnelles : grandes fonctionnalités du système, acteurs et autres systèmes qu'elles impliquent
- 4. spécifications non fonctionnelles, contraintes : – charte graphique – matériel : marques, RAM, débit de connexion Internet... – interfaçage : protocoles de communication, formats de fichiers, etc., pour l'interaction avec des matériels, logiciels, systèmes d'exploitation... – performances : temps réel... – sécurité : sauvegardes, confidentialité, ... – charge à supporter : volume des données, nombre d'utilisateurs simultanés, ... – comportement en cas de panne...
- 5. priorités relatives des spécifications, versions à prévoir, délais
- 6. évolutions à prévoir
- 7. annexes

(norme IEEE 830 :1993)

Spécifications: Exemple

- Un Guichet Automatique de Banque (GAB) offre les services suivants :
- Distribution d'argent à tout Porteur de carte de crédit, via un lecteur de carte et un distributeur de billets.
- Consultation de solde de compte, dépôt en numéraire et dépôt de chèques pour les clients porteurs d'une carte de crédit de la banque.
- Un opérateur de maintenance se charge de la collecte des dépôts d'argent et de la recharge du distributeur.

Spécifications: Exemple

- Besoins d'interface Homme/Machine (IHM) Les dispositifs d'entrée/sortie à la disposition du Porteur de carte doivent être:
- Un lecteur de carte bancaire.
- Un clavier numérique (pour saisir son code), avec des touches "validation", "correction" et "annulation".
- Un écran pour l'affichage des messages du GAB.
- Des touches autour de l'écran pour sélectionner un montant de retrait parmi ceux qui sont proposés.
- Un distributeur de billets.
- Un distributeur de tickets.

Spécifications: Exemple 2

- Créer un cahier de charge pour la gestion informatisée d'une bibliothèque.