



Génie logiciel avancé

L3 21 – 2022-2023

Marie NDIAYE

Introduction au Génie Logiciel

Le logiciel

- ▶ Qu'est-ce qu'un logiciel ?
 - ▶ Ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de l'information. *Arrêté du 22 décembre 1981 sur l'enrichissement de la langue française définit le terme « logiciel ».*
 - ▶ Ensemble des programmes et des procédures nécessaires au fonctionnement d'un système informatique. *Futura Sciences.*
- ▶ Software

Typologie des logiciels

- ▶ logiciels système, logiciels applicatifs, middleware.
- ▶ Utilisateurs cibles : grand public, spécialistes, développeurs, entreprises, particuliers, etc.
- ▶ niveau de standardisation
 - ▶ Logiciel standard
 - ▶ Logiciel spécifique
- ▶ Droits accordés par le contrat de licence
 - ▶ Logiciel propriétaire, logiciel libre, logiciel open source, logiciel gratuit, partagiciel(shareware)

La crise du logiciel

- ▶ Survenue à la fin des années 60
- ▶ Provient d'un décalage entre les progrès matériels d'une part et logiciels d'autre part :
 - ▶ Apparition des ordinateurs de la troisième génération, de plus en plus puissants et de moins en moins coûteux.
 - ▶ Construction de logiciels restait dans le domaine de l'artisanat et du folklore

Causes et manifestations de la crise du logiciel

► Causes

- Les difficultés de communication entre les acteurs
- Les difficultés dans la conception et le développement
- La négligence et le manque de méthodes et d'outils dans les phases de validation/vérification

► Manifestations

- Coût de construction de logiciels élevé
- Délais de livraison non respectés
- Les logiciels n'étaient pas évolutifs, ce qui les rendait très rapidement obsolètes
- Performances faibles
- Fiabilité aléatoire
- Convivialité discutable (interfaces homme/machine inexistantes)

Naissance du génie logiciel

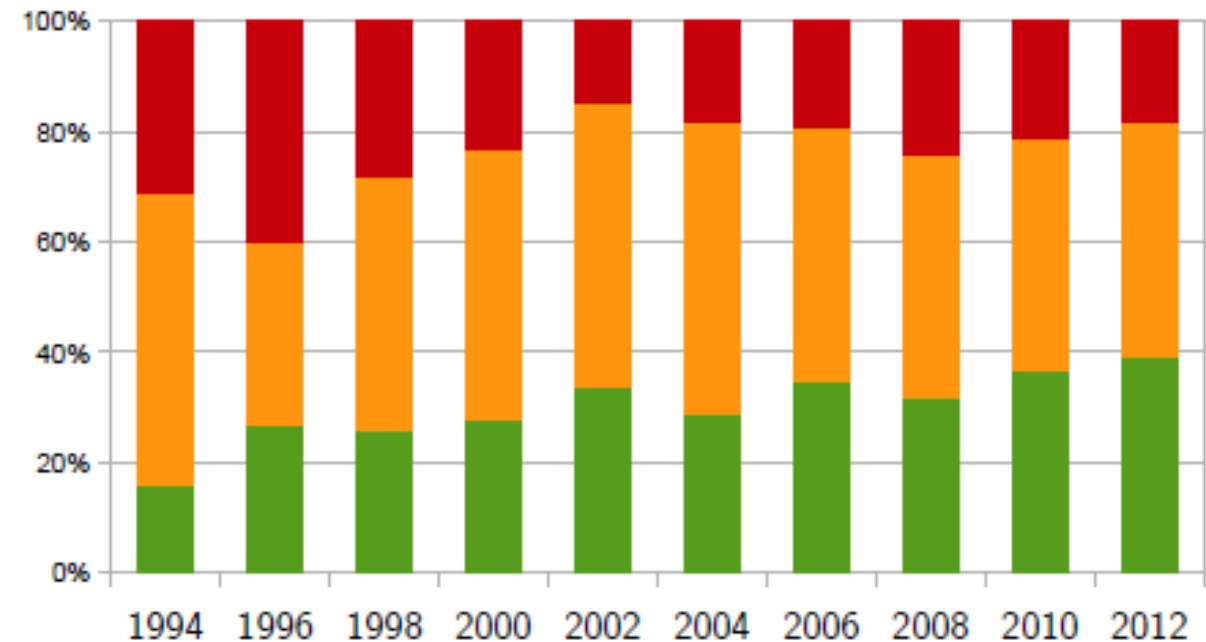
- ▶ Prise de conscience dans la communauté informatique
 - ▶ La production de logiciel manque du fondement rationnel présent dans d'autres techniques plus anciennes.
- ▶ Solution
 - ▶ Promouvoir un génie logiciel de manière analogue au génie civil, au génie électrique et au génie chimique...
- ▶ Événement fondateur de cette nouvelle discipline
 - ▶ Initiateur : **Friedrich Ludwig Bauer**, professeur à l'université technique de Munich
 - ▶ Octobre 1968 puis octobre 1969 : Conférence de travail sur les difficultés de la production de logiciel et les moyens de les surmonter.
 - ▶ Résultats
 - ▶ Popularisation du terme de *software engineering* (génie logiciel).
 - ▶ Mise au point sur la situation du domaine, identification des grandes lignes de la discipline naissante du génie logiciel et de définition de quelques-uns de ses objectifs ainsi que des pistes de recherche.

Crise du logiciel

- ▶ Fin des années 60
 - ▶ Coûts de production non maîtrisés
 - ▶ Délais de livraison non respectés
 - ▶ Logiciels pas toujours conformes aux attentes du client ou de l'utilisateur
 - ▶ Utilisation et maintenance difficiles
 - ▶ Logiciels non évolutifs

Enquête sur les logiciels 1/2

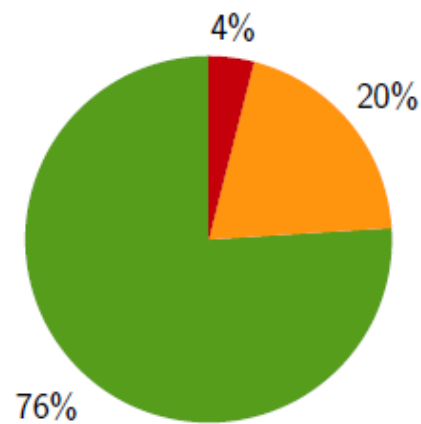
- ▶ Enquête sur des milliers de projets, de toutes tailles et de tous secteurs
 - ▶ **Projets réussis** : achevés dans les délais et pour le budget impartis, avec toutes les fonctionnalités demandées
 - ▶ **Projets mitigés** : achevés et opérationnels, mais livrés hors délais, hors budget ou sans toutes les fonctionnalités demandées
 - ▶ **Projets ratés** : abandonnés avant la fin ou livrés mais jamais utilisés



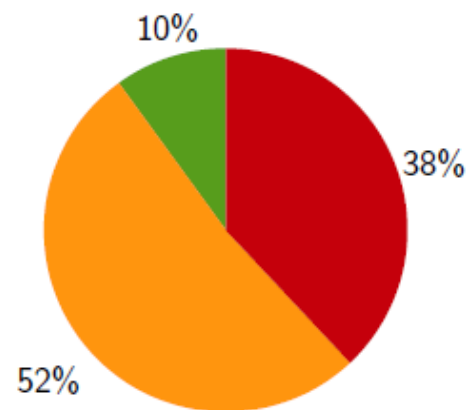
Standish group, *Chaos Manifesto 2013 - Think Big, Act Small*, 2013

Enquête sur les logiciels 1/2

► Petits projets/ grands projets

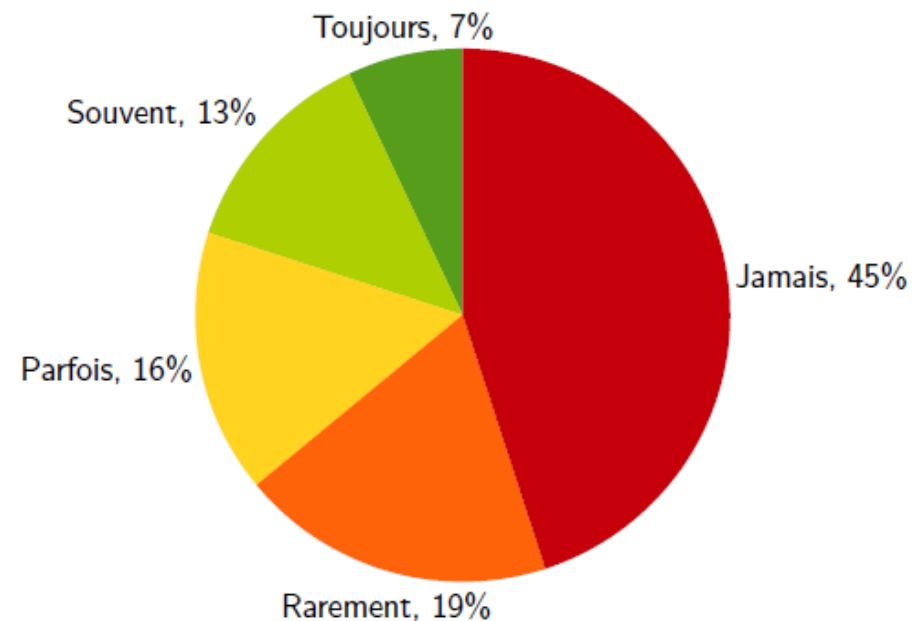


Petits projets
budget \leq \$1 million



Grands projets
budget \geq \$10 millions

► Utilisation des fonctionnalités implantées



Standish group, *Chaos Manifesto 2002*, 2002

Les échecs 1/6

■ Défaillances : les spectaculaires

- 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 (format US des nombres).
- 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



Les échecs 2/6

■ Défaillances : les navrantes

- Le Therac-25, un appareil d'irradiation thérapeutique a provoqué la mort de 2 personnes, et l'irradiation de 4 autres, à cause d'une erreur logicielle.
- Un navire de guerre anglais a été coulé par un Exocet français, au cours de la guerre des Malouines. Bilan de la catastrophe : plusieurs centaines de morts. Le vaisseau anglais n'avait, en effet, pas activé ses défenses, l'Exocet n'étant pas répertorié comme missile ennemi.
- En 1983, toute la vallée du Colorado a été inondée. La cause ? Une mauvaise modélisation dans le logiciel du temps d'ouverture du barrage.

Les échecs 3/6

■ Défaillances : les risibles

- Une personne s'est retrouvée saisie pour une dette impayée de 0,01F et cela bien qu'elle se soit acquittée de l'intégralité de sa dette. Le coupable ? un arrondi mal maîtrisé dans les calculs.
- Une centenaire (106 ans) s'est retrouvée convoquée à l'école. La cause ? le codage de l'année de naissance sur deux caractères.
- Une victime de l'an 2000 s'est vue adresser une amende de 91 500 \$ au retour d'une cassette vidéo louée. La raison ? Le retard calculé étant de cent ans car là encore le codage de l'année des emprunts avait été effectué sur deux caractères, pour gagner un peu de place.

Les échecs 4/6

■ Défaillances : les coûteuses

- 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.
- 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.

Les échecs 5/6

■ Défaillances : et c'est pas fini...

- La SNCF a rencontré des difficultés importantes pour la mise en service de son système Socrate (système de réservation des places). Plantages fréquents, mauvaise ergonomie, manque de formation du personnel, ont amené un report important et durable de la clientèle vers d'autres moyens de transport. La cause : le déploiement par la SNCF d'un système de réservation de places destiné aux compagnies aériennes, sans réadaptation véritable du cahier des charges au domaine du transport ferroviaire.

Les échecs 6/6

■ Dépassement des délais et des coûts :

- En mars 1993, la bourse de Londres a renoncé au projet informatique Taurus qui devait assurer complètement le suivi de l'exécution des transactions. Le système avait coûté directement 60 millions de livres et les opérateurs sur le marché avaient dépensé 400 millions de livres pour y adapter leurs propres logiciels.
- Toujours en 1993, en Californie, le DMV a renoncé au projet informatique démarré 6 ans plus tôt qui devait intégrer les systèmes d'information destinés à gérer les immatriculations et les permis. Les coûts étaient de 6,5 fois ceux prévus initialement et la date de livraison probable avait été reportée à 1998.
- Question pour un champion : Mars 1992, vous êtes nommé(e) Directeur du projet Taurus... quel est l'enjeu du projet ? Quelle est la stratégie à adopter ?

Les causes

- ▶ Communication difficile
- ▶ Conception et développement difficile
- ▶ Négligence et manque de méthodes et d'outils des phases de validation/vérification

Le génie logiciel 1/2

- ▶ Qu'est-ce que le Génie Logiciel?
 - ▶ D'après la Norme *IEEE 610.12* : Le Génie Logiciel est l'application d'une approche **systématique, disciplinée et quantifiable** au **développement**, à l'**exploitation** et à la **maintenance** du logiciel. C'est-à-dire, l'application de l'ingénierie au logiciel.
 - ▶ D'après *Classical and Object-Oriented Software Engineering with UML and Java* de Schach : Le Génie Logiciel est une discipline qui a pour but la fabrication du logiciel **sans faute, livré dans le délai et le budget prévus à l'avance**, et qui **satisfait aux besoins du client**.
 - ▶ D'après Marie Claude Gaudel : Le Génie Logiciel est 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 système informatique pour résoudre certains problèmes
 - ▶ Le Génie Logiciel est l'art et la science de concevoir et de construire, avec économie et élégance, **des applications, des objets, des frameworks et d'autres systèmes informatiques**, qui soient **corrects, robustes, extensibles, réutilisables, sûrs, efficaces, faciles à maintenir et à utiliser**.

Le génie logiciel 2/2

- ▶ Principe
 - ▶ Appliquer les méthodes classiques d'ingénierie au domaine du logiciel
- ▶ Objectifs
 - ▶ Maîtrise du processus de fabrication de logiciels
 - ▶ Respect des coûts et des délais
 - ▶ Satisfaction du client
 - ▶ Production de logiciels de qualité

Problème du Génie Logiciel

- ▶ Faire en sorte que les développeurs construisent un logiciel qui répond parfaitement aux exigences des utilisateurs.
- ▶ **Question fondamentale :**
 - ▶ Comment traduire les exigences dans des fichiers qui constituent un logiciel ?
- Ou**
- ▶ Comment rendre les exigences des utilisateurs interprétables par un ordinateur ?

Solutions proposées par le Génie Logiciel

- ▶ Séparer les aspects fonctionnels et technologiques
- ▶ Décomposition en sous-systèmes \Rightarrow approche objet
- ▶ Démarche itérative \Rightarrow approche objet

Qualité du logiciel

- ▶ Privilégier la **qualité** et l'**efficacité**
 - ▶ Démarche qualité : **ISO 9126**
 - ▶ **Prévention** (erreurs) moins coûteuse que **correction**
- ▶ **Qualité externe** vs **qualité interne**
 - ▶ Externe \Rightarrow Client
 - ▶ Interne \Rightarrow Développeur

Références

- ▶ Support de cours de Laurent TICHIT et de Noël NOVELLI
 - ▶ <http://www.dil.univ-mrs.fr/~novelli/GL/>
- ▶ Support de cours de Delphine Longuet
 - ▶ <https://www.lri.fr/~longuet/Enseignements/14-15/Miage-GLA/>
- ▶ Autres sites
 - ▶ <https://perso.liris.cnrs.fr/christine.solnon/agl.html#crise>
 - ▶ https://fr.wikipedia.org/wiki/Friedrich_L._Bauer
 - ▶ <https://interstices.info/la-naissance-du-genie-logiciel/>
- ▶ *Le logiciel système*, coll. Que Sais-je no 2681, Thierry Falissard, 1997
- ▶ <https://www.encyclopedie.fr/definition/Application>
- ▶ <https://www.encyclopedie.fr/definition/Middleware>
- ▶ <https://reponsesrapides.fr/la-vie/quel-type-de-logiciel-est-construit-pour-repondre-a-cette-demande/>
- ▶ <https://fr.wikipedia.org/wiki/Logiciel>
- ▶ <https://www.wiki.fr-fr.nina.az/Logiciel.html>
- ▶ <https://www.talend.com/fr/resources/guide-middleware/>