université de TOURS

Génie logiciel

Thomas Devogele

Cours 1 : Introduction

Définition, Objectifs, Coûts

Bibliographie Génie logiciel

- **Précis de génie logiciel, de Gaudel, Masson**
- **Le Génie Logiciel, de Jacques Printz, Que sais-je? Presses Universitaires de France – PUF**
- **Architecture Logicielle : Concevoir Des Applications Simples, Sûres Et Adaptables, Printz, Dunod**
- **Scrum : Le guide pratique de la méthode agile la plus populaire - 2ème édition de Claude Aubry (7 septembre 2011) Dunod**

Bibliographie Génie logiciel

- **Le génie Logiciel (OpenClassRoom)**
  - https://openclassrooms.com/courses/creez-des-applications-de-qualite-avec-le-design-pattern-mvc/le-genie-logiciel-gl
- **Introduction au Génie logiciel (Alain.Tisserant, Mines Nancy)**
  - http://www.mines.u-nancy.fr/~tisseran/ cours/qualite-logiciel/qualite_logiciel.html
- **Génie logiciel cours de Jacques Lonchamp**
  - http://www.loria.fr/~jloncham/gl.html
- **Le génie logiciel cours d'Antoine Beugnard**
  - http://public.enst-bretagne.fr/~beugnard/
- **Gestion de projet (très bon Cours vidéo de Rémi Bachelet)**
  - http://gestiondeprojet.pm/
- **Cours vidéo de Xavier Blanc (Université de Bordeaux)**
  - https://www.youtube.com/playlist?list=PLuNTRFkYD3u4d5TLsUJmO_w9jT3mHy4Hq

Logiciel & système d'information

- **Logiciel (software)**
  - Partie non matérielle (électronique) d'un système informatique
  - ensemble de séquences d'instructions (programme) compréhensible par une machine et d'un jeu de données nécessaires à ces opérations.
  - Application sur ordinateur, smartphone, embarqué (véhicule, robotique, domotique…)
- **Système d'information (SI)**
  - ensemble de ressources qui permet de collecter, stocker, traiter et distribuer de l'information numérique
  - Regroupe du matériel, des logiciels et leurs données et du personnel

Génie logiciel GL
(software engineering)

- **Définition**
  - Science de l'ingénieur dont la finalité est la **fabrication de systèmes d'information** complexe
  - science 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 GL (software engineering)

- **Programmer n'est pas concevoir un système informatique**
- **La technique ? Nécessaire, mais pas si importante que ça !**
  - Les VRAIS problèmes difficiles :
    - l'organisation, la gestion
    - difficulté de formalisation
    - multitude de paramètres, facteurs
    - gestions des humains (informaticiens et futurs utilisateurs)
    - Reprise d'un existant
    - Gestion des changements d'objectifs,
    - Gestion des changements d'environnement (matériel, OS, SGBD…)

Définition du génie logiciel

- **Génie Logiciel (GL)**
  - domaine de l'ingénierie qui permet la conception, la réalisation et la maintenance des systèmes logiciels de qualité.
  - UK : software engineering

- Le **génie logiciel** représente l'application de principes d'**ingénierie** au domaine de la création de logiciels. Il consiste à identifier et à utiliser des **méthodes**, des **pratiques** et des **outils** permettant de maximiser les chances de réussite d'un projet logiciel.

Objectifs majeurs du GL

- Le génie logiciel regroupe l'ensemble des moyens qu'il faut réunir pour **spécifier**, **construire, distribuer** et **maintenir** des logiciels qui soient :
  - **Sûrs** (système déterminé même si erreur utilisateur)
  - **Purgés de toutes ses erreurs**
    - (ou s'assurer que les conséquences néfastes des erreurs résiduelles pourront être compensées rapidement)
  - **Conviviaux** (capacité des usagers)
  - Évolutifs
  - Économiques
- **Problèmes :**
  - ↑ puissance des matériels
    ➔ ↑ de la fréquence des erreurs
  - informatique répartie
  - couplage de logiciel

Paramètres économiques

- **Coût du logiciel**
  - **en hommes-mois (hm ou ha)**
  - **En personnes-mois (pm)**
- **Volume du logiciel**
  - **en nombre de lignes de code source (ls et kls)**
- **Délais**
  - durée
- **Productivité : ls/pm**

Exemples

- **Compilateur C :**
  - 10 ha, 20 à 30 kls délais 1 à 2 ans
- **Système d'exploitation : Windows 2000 (en 1999)**
  - 5 000 ingénieurs pendant 3 ans pour réécrire 70% des 35 000 kls de NT. Productivité = 4,8 lignes par homme jour
  - validation de Windows 2000 : 600 000 bêta testeurs, il restait pourtant au lancement de sa commercialisation 28 000 problèmes réels
- **Système de la navette spatiale**
  - >1000 ha, 2 200 kls, délais 6 ans

Les bugs : erreurs informatiques

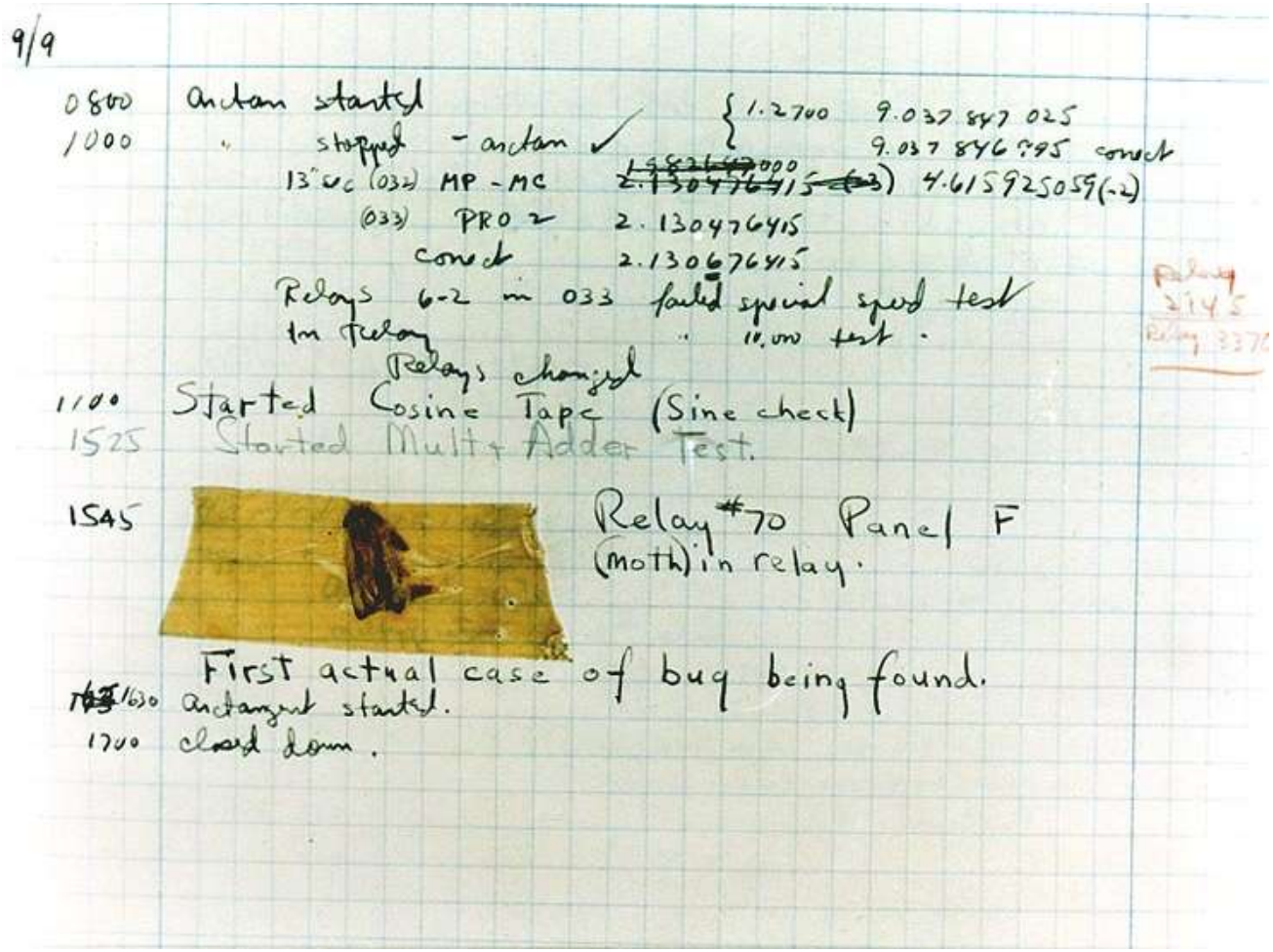
Anomalie de fonctionnement d'un programme informatique.

bibliographie

- 29 bugs informatiques aux conséquences catastrophiques
- 7 bugs informatiques catastrophiques Doc Seven vidéo Youtube 24/8/2016
- Top Ten Most Disastrous Software Bugs Choppedporks vidéo Youtube 4/11/2015

Les bugs

Grace Hopper (1906 – 1992)



Les bugs

- **Un bug peut provoquer**
  - un plantage c'est-à-dire un arrêt inattendu
  - d'importantes pertes d'informations
  - une véritable catastrophe
  - une faille de sécurité est un défaut mineur qui ne provoque pas de dysfonctionnement en utilisation courante, mais permet à un utilisateur malicieux ou un logiciel malveillant d'effectuer des opérations non autorisées à partir d'un exploit.

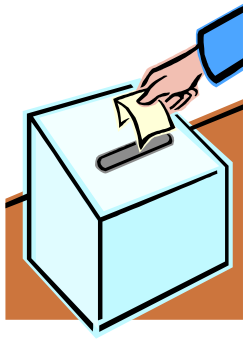
Différents types de systèmes logiciel

- **Système simple**
  - Un jeu,
- **Système d'exploitation (OS)**
- **Système critique**
  - un dispositif informatique dont le dysfonctionnement peut mettre en danger la santé et la vie des gens et des écosystèmes, provoquer des importants dégâts matériels ou avoir des répercussions sur la stabilité économique et politique.

Exemple d'erreur de programmation

- **Interface entre logiciels**
  - Cartes de crédit systématiquement avalées :
    - désaccord entre 2 calculateurs sur le calcul des années bissextiles

Exemple d'erreur de programmation



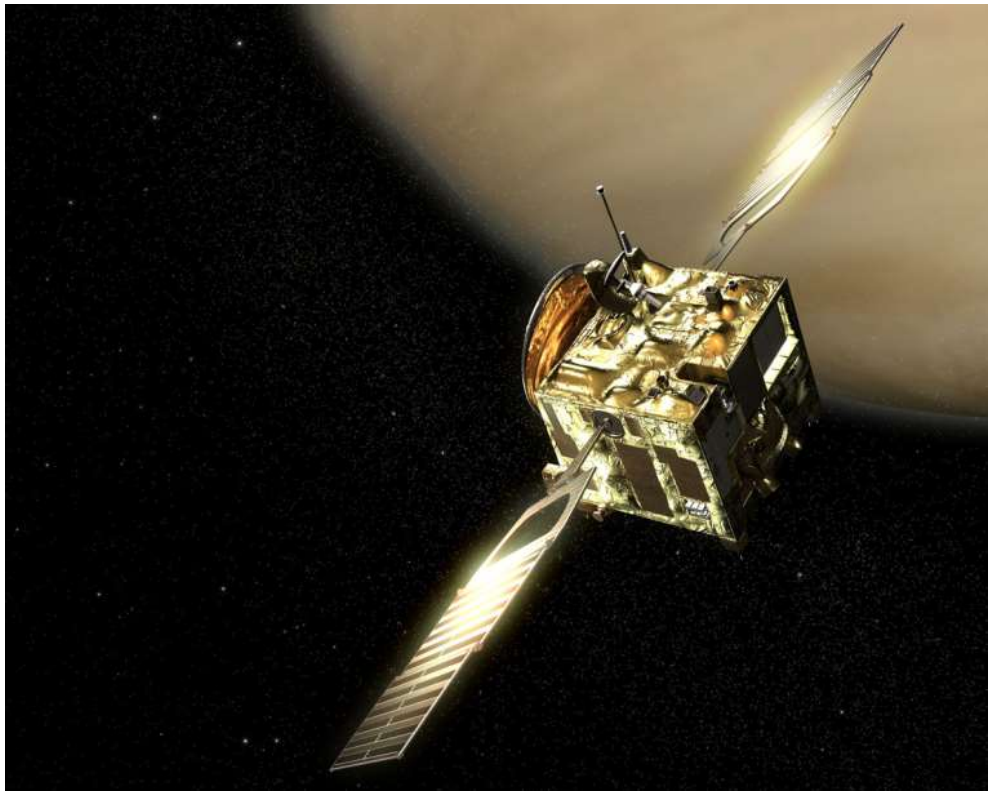
- **Problème d'arrondi**
  - Il faut rassembler 10% des inscrits pour pouvoir se maintenir au 2è tour.
  - En 2008, le candidat vert a obtenu au premier tour 1.765 voix sur 17.656 électeurs inscrits (9,998 %),
  - Le logiciel du ministère de l'Intérieur l'a autorisé à se représenter.
    - le logiciel ne va pas jusqu'à trois chiffres après la virgule, d'où "un arrondi très malheureux à 10%",
  - www.boulogne-billancourt.lesverts.fr/spip.php?article298

Exemple d'erreur de programmation

- **Problème d'initialisation à 0**
  - Laser de visée guidage terminal
  - Commando qui guide un missile en phase terminale
    - Le système de guidage transmet au missile un cap et une distance entre le commando et la cible
    - Le système indique batterie faible
    - Le commando change la pile
    - Initialisation des variables à zéro
    - Le missile arrive
      - 5 morts
  - Remarque : l'initialisation à 0 est une bonne pratique sauf dans certains cas …

Exemple d’erreur de programmation

- **Mission Vénus (Problème de syntaxe):**
  - passage à 500.000 km au lieu de 5.000 km : remplacement d'une virgule par un point



Exemple d’erreur de programmation

- **Mauvaise prise en compte des nombres négatifs**
  - Avion F16 déclaré sur le dos au passage de l'équateur :
    - non prise en compte du référentiel hémisphère sud (nombre négatif)

Erreur de gestion de projet

- **Taurus**
  - projet d'automatisation de la City abandonné après 4 années de travail et 100 millions de £ de pertes

Erreur de gestion de projet

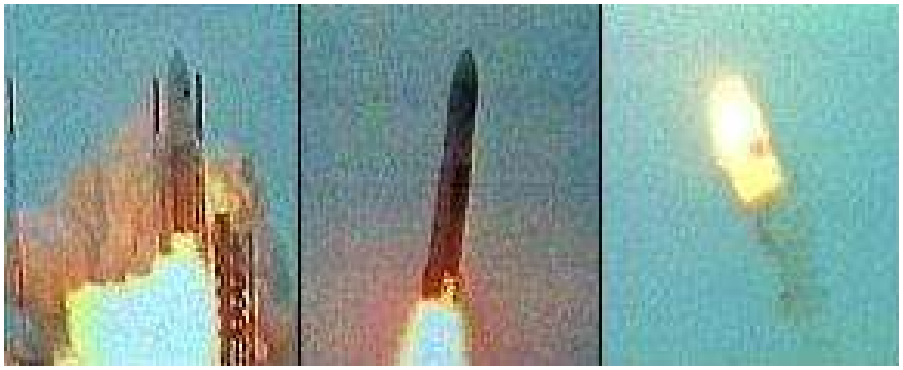
- **Fusion de l'ensemble des systèmes de paye de l'armée : Logiciel de Paye Louvois**
  - 40% des militaires français ont eu une paye non versée ou très erronée
  - Coût du système : 20 millions €
- **Problèmes**
  - Pas de réels experts côté clients (l'armée)
  - Remplacement des anciens systèmes sans vrais tests du nouveau
  - Pas de travail de préparation, pas de vraies spécifications des besoins détaillés
  - Système peu robuste
    - Traitement des erreurs pas bien pris en compte
  - Difficilement maintenable
    - Difficile à modifier une partie sans devoir tout modifier
- **Référence France TV info : autopsie d'une débâcle militaire**

Erreur de dimensionnement

- **21/05/2010 Panne du système de réservation lié à une demande trop importante de réservation (engorgement)**

Erreur de changement de version

- **Ariane 5 (1996)**
  - explosion de la fusée quelque temps après le décollage
    (coût 1 milliard de dollars)



  - La cause : logiciel de plate forme inertielle repris tel quel d'Ariane IV sans nouvelle validation. 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 au plan de tir (d'Ariane IV), et a provoqué l'ordre d'auto destruction. En fait tout se passait bien...

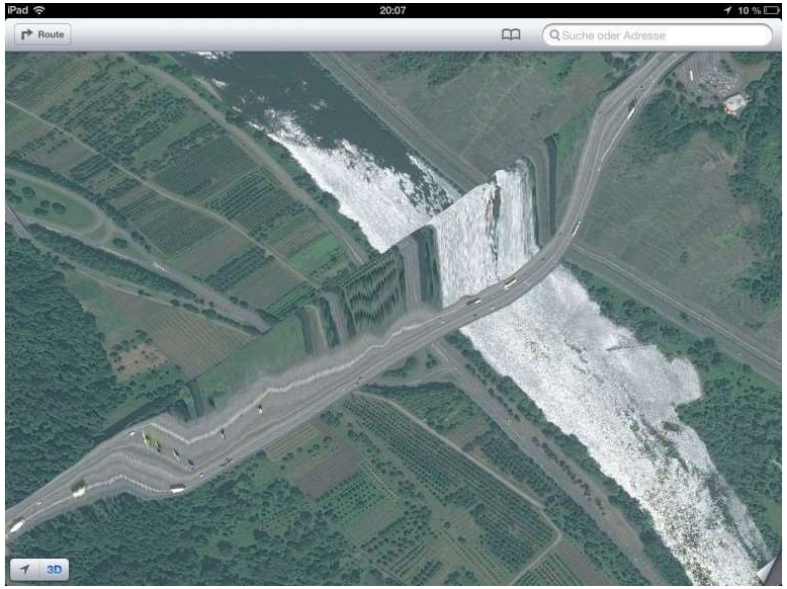
Erreur utilisateur : Problème d'envoi répété non désiré

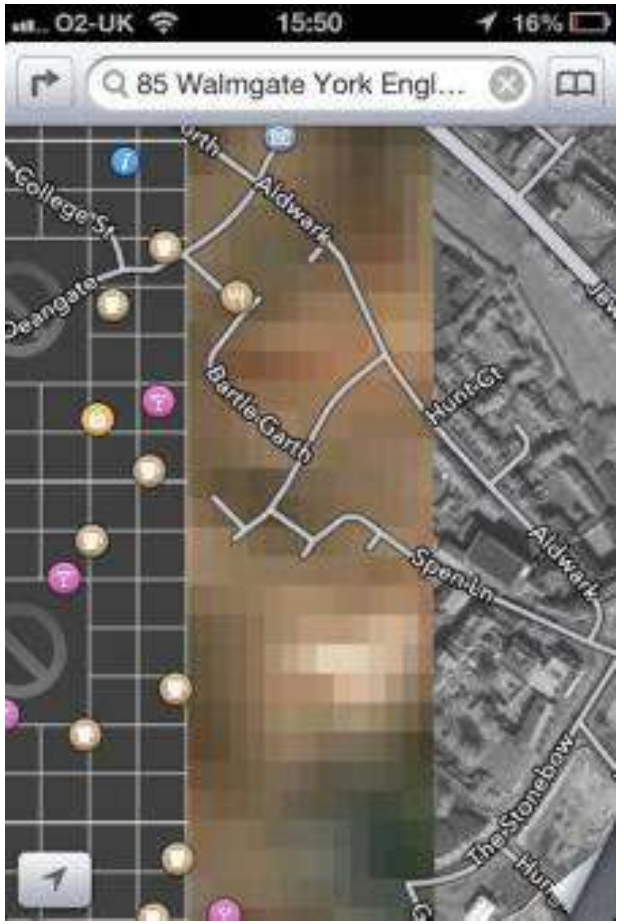
- **Deutsche Bank (mai 2010)**
  - envois d'ordres de vente à la Bourse japonaise d'Osaka,
    - pour un montant de 150 milliards d'euros
    - 6 millions de contrats à terme
    - dix fois le montant moyen des échanges quotidiens
  - a provoqué une baisse des marchés japonais
    - 1,1% de l'indice Nikkei (plusieurs milliards d'euros)

- ***Problème***
  - *un ordre de vente a été envoyé de façon répétée*
  - *Système très rapide sans mécanismes de contrôle suffisants*
  - **70% des ordres passés sur les Bourses américaines** sont le fait de ces robots virtuels
  - **37 microsecondes**, c'est la vitesse à laquelle les traders haute fréquence peuvent passer leurs ordres

Erreur d'intégration de données

- **Application Plans d'Apple pour Iphone (2012)**
  - Remplacement des données Google par des données issues de différentes source sans vérification de
    - la cohérence entre les données
    - La complétude





Erreur : Non récupération d'anciennes données

- **En changeant de système informatique les hôpitaux de Paris (2016) ont perdu la trace de 80 millions d'euros... qu'elle ne pourra plus jamais réclamer à ses créanciers**
  Vidéo France 2 ici

Problème de mode d'emploi

- **notice de fonctionnement des distributeurs de la Bank of Montréal (Sur Internet) contenant le code secret à six chiffres par défaut**
- **(juin 2014) 2 adolescents ont eu**
  - accès à l'ensemble des données du distributeur
  - la possibilité de modifier des paramètres (montant des frais, etc.)

Problèmes informatiques

- **La non-qualité des systèmes informatiques**
- **La non prise en compte des « erreurs humaines »**

  - ont des conséquences qui peuvent être très graves
    - Financières
    - Humaines

Pourquoi les bugs ?

- **Les bugs résultent d'erreurs humaines lors des travaux**
  - de spécification,
  - de conception,
  - de programmation
  - et de tests de logiciel et de matériel informatique.

La Crise du logiciel

- Constat des carences des logiciels depuis 1980
- Exemple
  - Etude sur 8 380 projets (Standish Group, 1995)
    - Succès : 16 %
    - Problématique : 53 % (budget ou délais non respectes, défaut de fonctionnalités)
    - Echec : 31 % (abandonne)
  - Le taux de succès décroît avec la taille des projets et la taille des entreprises

Besoin du GENIE LOGICIEL

Humour

- **« Tout programme non trivial possède au moins un bug. »**
  - Loi de Murphy
- **« L'erreur est humaine, mais un vrai désastre nécessite un ordinateur. »**
- **« Quand un logiciel n'a plus aucun bug, il est habituellement désuet. »**

Quelques mesures contre les bugs

- **Tests logiciels**
- **Utilisation de langages de hauts niveaux**
- **Débogage**
- **Maintenance corrective**
- **Prouver qu'il n'y a pas de Bugs**
  - Possible, mais chère
  - Réservé à des programmes critiques
- **Design pattern**
  - Cours Master SIAD (L3)

Tests de logiciels

- **Les tests de logiciels**
  - la première mesure pour contrer les bugs.
  - Impossible de tout tester
  - Exemple
    - Microsoft Word 2003 compte
      - 850 commandes
      - 1600 fonctions,

      ➔de plus de 500 millions de conditions à tester.

Langages de Programmation de haut niveau

- **Facilitent le travail de l'ingénieur.**
  - Délègue à l'environnement de développement des tâches de contrôle des erreurs

- **La mise en application de conventions de rédaction est une autre technique préventive destinée à diminuer le nombre de bugs.**

Le débogage

- **Le débogage est l'activité qui consiste à diagnostiquer et corriger des bugs.**
- **Deux types de débogage**
  - débogage en ligne
    - exécution du logiciel pas à pas et lancement après chaque pas d'une série de vérifications
  - débogage post-mortem
    - Examen du logiciel à la suite d'un crash informatique

Le débogage

- **Difficulté de la détection**
  - une faute est noyée au milieu d'une masse d'informations
- **Une erreur engendre la corruption des informations**
  - entre le début de l'erreur et son identification
- **La correction des erreurs doit comprendre**
  - la recherche de l'erreur et sa correction
  - La remise en état cohérent des informations si corruption
  - des corrections en cascade dans d'autres modules

Maintenance corrective

- **Lorsque le bug est décelé et corrigé après la distribution du logiciel,**
  - Distribution d'un *patch*,
    - un kit qui remplace les parties défaillantes du logiciel par celles qui ont été corrigées.

Design pattern

- **« Chaque patron décrit un problème qui se manifeste constamment dans notre environnement, et donc décrit le cœur de la solution à ce problème, d'une façon telle que l'on puisse réutiliser cette solution des millions de fois, sans jamais le faire deux fois de la même manière »**
  - Christopher Alexander, 1977.
- **« Les patrons offrent la possibilité de capitaliser un savoir précieux né du savoir-faire d'experts »**
  - Buschmann, 1996.

Design pattern

- **Exemple**
  - Le patron **Modèle-Vue-Contrôleur** (MVC) des IHM est une combinaison des patrons Observateur, Stratégie et Composite.
- **Décomposition en 3 parties**
  - un **modèle**
    - modèle de données
  - une **vue**
    - présentation, interface utilisateur
  - un **contrôleur**
    - logique de contrôle, gestion des événements, synchronisation

Conduite de Projet et phases du projet

Partie tirée de la vidéo de Xavier Blanc (Université de Bordeaux)

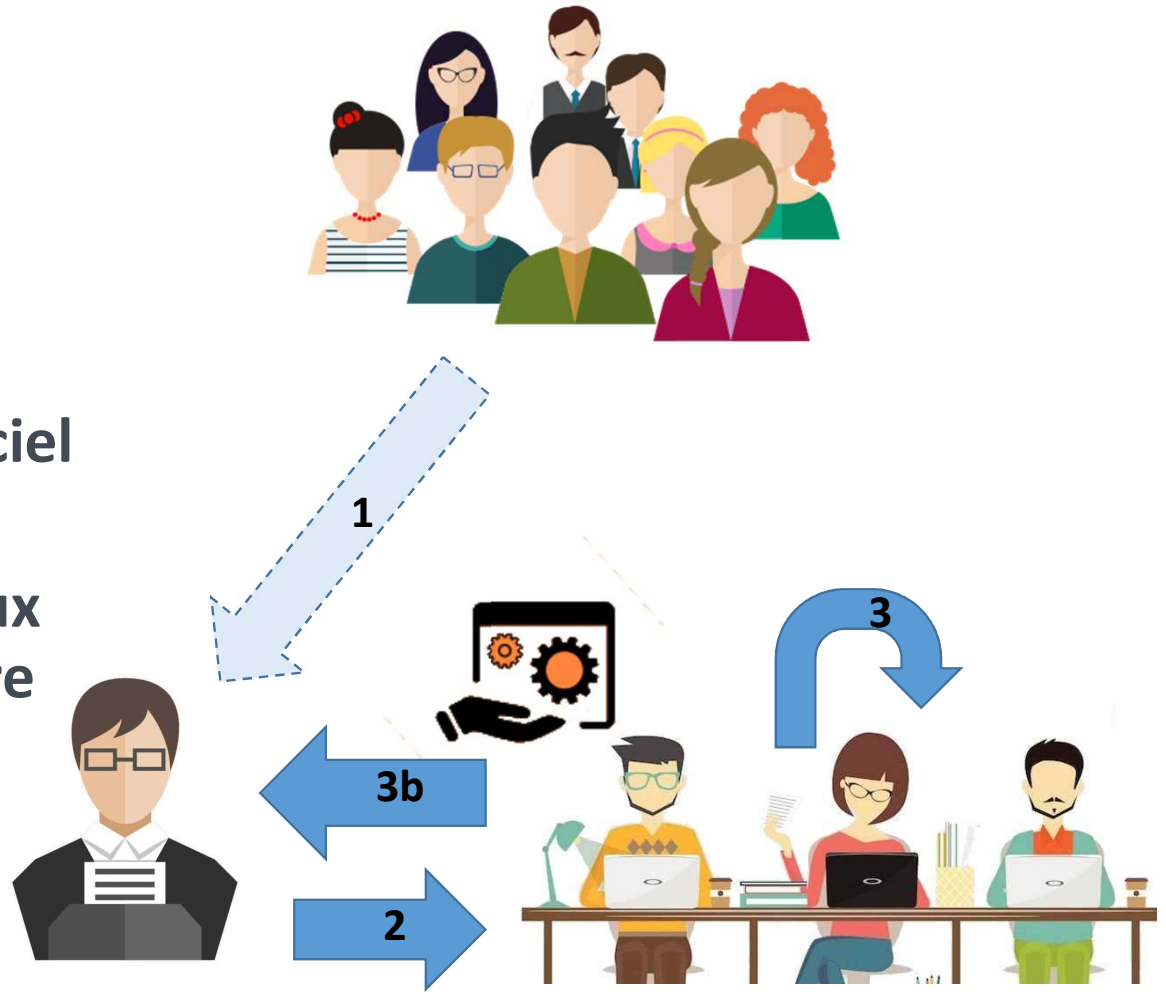
https://www.youtube.com/watch?v=vhpcngRVE_A&t=656s

La production d'un logiciel ou d'un SI

- **Les participants et leurs rôles**
  - Les **utilisateurs** (end user)
    - Vont utiliser le logiciel pour répondre à des **besoins** (connu ou nouveau)
  - Les **développeurs** (software developer)
    - Va produire le code source, la documentation, les améliorations … et définir le SI
  - Le **propriétaire** (owner)
    - Possède le logiciel
    - Paye le développeur
    - Fournit le logiciel aux utilisateurs
    - Se fait payer par les utilisateurs si le logiciel est payant

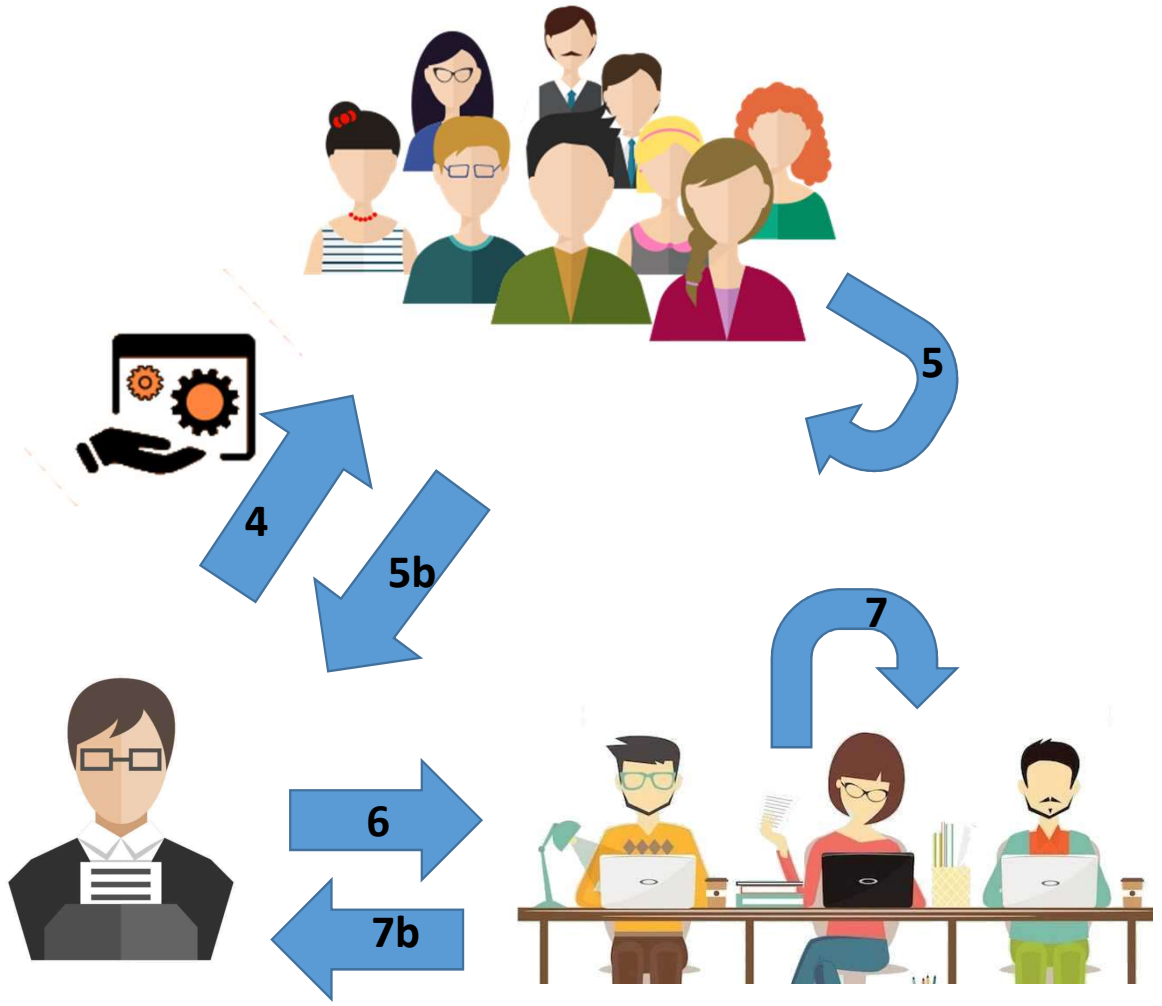
La création

1. **Les utilisateurs ont des besoins**
2. **Le propriétaire demande aux développeurs de réaliser le logiciel**
3. **Les développeurs réalisent le logiciel qui doit correspondre aux besoins des utilisateurs et le livre au producteur**



L’exploitation

4. **Le propriétaire déploie le logiciel**
5. **Les utilisateurs utilisent le logiciel, ils rencontrent des anomalies et aimeraient voir des évolutions**
6. **Le propriétaire demande au développeur de corriger les anomalies ou de développer les évolutions**
7. **Les développeurs corrigent les anomalies et réalisent les évolutions et le livre**
8. **Le propriétaire déploie la nouvelle version (proche de la phase 4)**



Les bases du cycle de vie d'un logiciel

- **Développement**
  - Identifications des besoins auxquels le logiciel doit répondre (Propriétaire)
  - Développement (Développeur)
  - Livraison au propriétaire
  - Test de déploiement (pré production)
- **Production & maintenance**
  - Livraison
  - Maintenance corrective
  - Maintenance évolutive
  - Livraison de mise à jour

Le cahier des charges, les spécifications et le chiffrage

- **Le cahier des charges**
  - Vision utilisateur du logiciel ou du SI exprimé dans un langage **métier** par le propriétaire
  - Exigence fonctionnelle
    - ce que doit faire le logiciel ou le SI
  - Contraintes métier
    - Règle à respecter
    - Reprise d'un existant

Exemple (simplifié) de cahier des charges pour une société d'assurance, qui désire **un système pour suivre le traitement des sinistres de ses clients**

qui regroupe les différentes étapes

- Déclaration du sinistre,
- Désignation d'un ou plusieurs experts si nécessaire,
  - Récupération des expertises
- Négociation avec le client
- devis des entreprises
- Validation du devis
- Paiement des entreprises
- Historisation des différents sinistres de mes clients

Qui a des Contraintes

- Ce système doit être relié à mes fichiers clients,
- les clients doivent pouvoir suivre l'évolution du traitement via notre site web.
- Le SI doit répondre aux normes ISO XXX des assurances et à la RGPD (Règlement général sur la protection des données)

Le cahier des charges, les spécifications et le chiffrage

- **Les spécifications**
  - Réponses des développeurs aux cahiers des charges
  - Proposition d'un logiciel ou d'un SI qui « répond » aux besoins exprimés par le propriétaire.
    - Certains besoins peuvent ne pas être couverts ou couverts partiellement pour des raisons de coût ou de sécurité par exemple
  - Elles peuvent être plus ou moins détaillées
  - C'est la traduction du cahier des charges en termes plus techniques (informatique)
  - Elles décrivent comment les exigences fonctionnelles vont être implémentées dans le SI, et donc permettre le développement du logiciel.
  - Elles ajoutent des contraintes informatiques : sauvegarde, sécurité, maintenance…
  - Les spécification doivent être validées par le propriétaire
  - Il doit être compréhensible par les développeurs et le propriétaire

Le cahier des charges, les spécifications et le chiffrage

- **Le chiffrage**
  - C'est le devis
    - Exprimé en personne/ mois pour chaque tâche
- **Le chiffrage en informatique est complexe**
  - Développement de nouvelles tâches : coût difficile à prédire
  - Refacturation de tâches déjà réalisés dans d'ancien projet
  - Cette activité sera vue en L3 en gestion de projet

Exercices

- **Imaginer un cahier des charges pour créer un logiciel de gestion des inscriptions à l'université**
- **Imaginer une réponse au cahier des charges sous forme de spécifications informatiques**


1 PLANNING
2 DESIGN
3 PROGRAMMATION
JAVA
CSS3
PHP
HTML5
4 TESTS
5 LIVRAISON
6 MAINTENANCE