

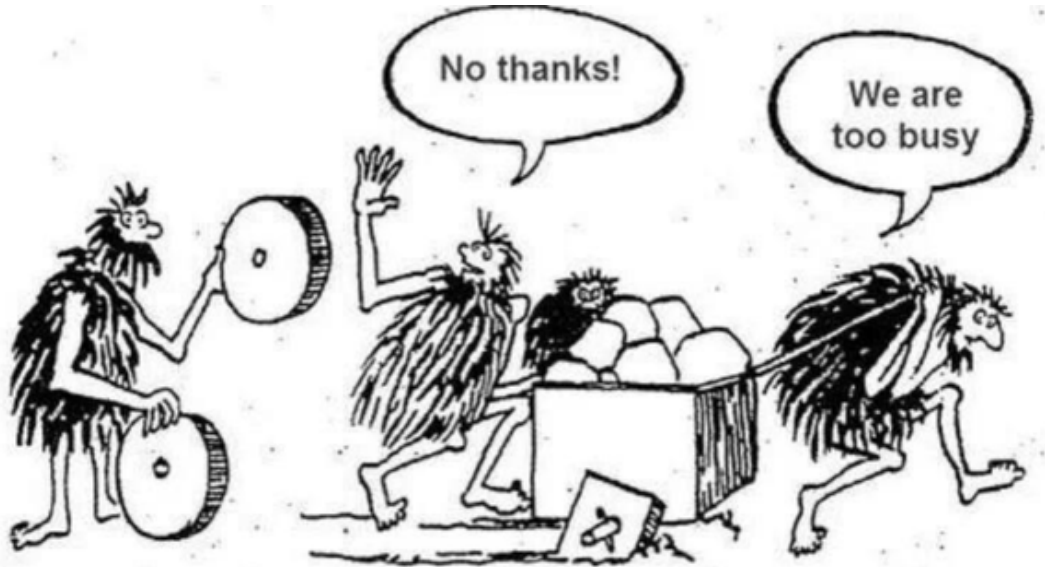
Génie Logiciel

Dominique Revuz, Philippe Finkel, Philippe Cluzeau

ESIPE

mars 2021

Au départ...



Plan

- ➊ Positionnement et objectifs des cours/tds de génie logiciel
- ➋ Pourquoi développer des logiciels
- ➌ Définition du génie logiciel
- ➍ Mythes
- ➎ Qualité

Objectifs

Objectifs généraux

- ❶ Culture générale projet logiciel
- ❷ 2 drôles de termes : *Création* et *Élaboration*

Hors périmètre de ce cours

- Le **COMMENT**. On se concentre sur le **POURQUOI** et le **QUOI**
 - par ex : diagramme de classes et conception
- Gestion de projet, Work Breakdown Structure (WBS) que vous verrez en INFO3 (cours *simulation de pilotage*)
- DevOps (en INFO2 aussi)

Indirectement lié

- Agilité : on fera une matinée avec Yifan pour faire le lien.
 - orga & réalisation. important pour parler de priorisation et de roadmap

Comment obtenir les besoins **réels**

point d'attention systématique : on va devoir travailler l'écoute et l'empathie parce que le client est compétent dans son domaine mais pas en GL !!

- il faudra **lire entre les lignes**
- apprendre à **poser des questions** ! même voire surtout des questions idiotes !!
- **reformuler** souvent pour s'assurer d'une bonne compréhension mutuelle
- chercher **plusieurs axes** "d'attaque" des besoins pour mettre le client en face de ses contradictions et mieux comprendre les **priorités**
- identifier et clarifier les **hypothèses** et les **contraintes**

Comment décrire au mieux un logiciel à réaliser

- pour que les concepteurs aient tous les éléments en vue,
- et que le résultat corresponde au mieux aux réels besoins des clients et à leurs priorités
- en s'assurant que ça tient dans le budget, voire les délais ! donc en cadrant suffisamment le périmètre du projet

Création ... Élaboration

- 1 Culture générale
- 2 Comprendre et s'appropriier les phases de **création** et d'**élaboration**
- 3 Travailler et s'entraîner avec les outils, diagrammes, terminologies, checklists

Synonymes pour création : inception, cadrage, voire précadrage

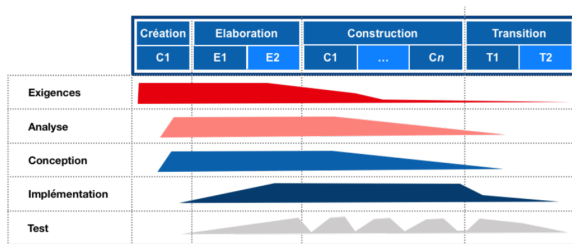


Figure 1: Processus unifié

source: https://fr.wikipedia.org/wiki/Processus_unifié

INFO2 ... INFO3

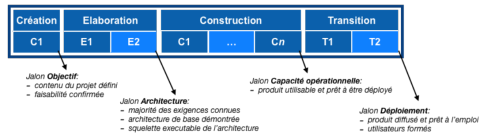


Figure 2: Phases et jalons

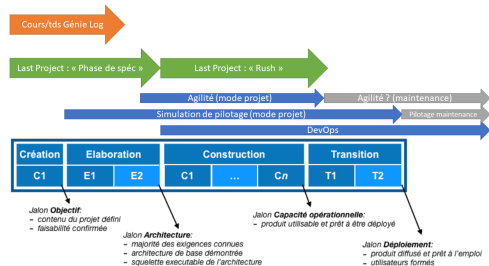


Figure 3: Et à l'ESIPE ?

Objectifs

Ces cours et td de GL doivent vous aider à mieux structurer votre réflexion

Objectifs “techniques” :

- diagrammes d’acteurs, de cas d’utilisation, d’interaction générale
- glossaire
- acteurs (humain, non-humain) / partie prenantes
- objets du domaine
- cas d’utilisation
 - et cas limites, cas d’erreur des UC
 - hiérarchisation
 - essentiels / détaillés
- FQM
- IHM, architecture générale

Nos rêves :-)

- A la fin de votre formation, vous serez en capacité d'intégrer comme ingénieur une équipe de développement.
- A la fin du module génie logiciel, vous serez capables de participer aux phases suivantes d'un développement : création, (inception, démarrage, cadrage qui vise à définir le produit et les objectifs du projet), élaboration (toutes les facettes de l'élaboration), tests, recette.
- Pour les interviews, vous pourrez mettre en application les savoir être principaux d'empathie, reformulation et pédagogie au cours d'une interview basée sur un projet concret; et appliquer les principes du QQQQCCP et de ses 7 questionnements.
- Qualité transverse : vous montrerez dans ces productions (dessins, listes, scénarios, descriptions) les qualités de précision et de concision. Et appliquerez le principe *DRY* (Don't Repeat Yourself), *“Dans un système, toute connaissance doit avoir une représentation unique, non-ambiguë, faisant autorité”*. Cela commencera avec la terminologie.

Remarque: Nous définissons le “projet type” pour cet enseignement comme un projet d'une dizaine de mois homme (tout compris. soit 200 j.h).

Pourquoi développons nous des logiciels

- ① Pour le Plaisir
- ② Pour gagner sa vie
- ③ Pour répondre à des **besoins** ! Et régler des **problèmes**

Pour répondre à des besoins !

- ① Faire fonctionner des appareils
- ② Fournir des services aux utilisateurs
 - ① Services de Communication
 - ② Services variés
- ③ Réaliser des Tâches
 - ① Travailler
 - ② Culture détente etc

Deux Grands cas de figure

- ① Le fonctionnement de machines électroniques
 - ① Les systèmes d'exploitation sont historiquement les premiers logiciels importants
 - ② Mais aujourd'hui le nombre d'appareils pilotés par des logiciels est colossal
- ② permettre à une personne de réaliser une tâche de manipulation de données
 - ① En Général c'est une activité liée au travail
 - ② les jeux ...

- ① Le Génie logiciel que nous allons voir ensemble est plus orienté vers la deuxième mission
 - ① En effet, nous allons nous intéresser à **l'aspect humain de la compréhension d'un objectif de développement.**
 - ② ex: pilote automatique => spécifications peuvent être écrites.
- ② Cette mission est double
 - ① Comprendre et formaliser les besoins et proposer une solution qui les satisfasse
 - ② Développer la solution
- ③ Ainsi nous avons une partie plus science humaine puis une partie plus technique mais elles doivent être judicieusement associées
 - ① Compréhension humaine sans les aspects techniques => risque d'être *hors sol*
 - ② Expertise technique sans compréhension profonde des besoins => risque de faire des jolis logiciels bibelots inutiles

La plupart des projets informatiques se passent mal !

- dépassement important des **délais**
- dépassement important des **coûts**
- non **conformité** :
 - bugs,
 - limitations,
 - “flou” car comportement attendu “discutable” (imprécision, quiproquo, ...)
- difficulté et coût important de de maintenance

Et qualité souvent insuffisante

la qualité concerne bien sûr toute la chaîne :

- expression de besoins
 - cahier des charges fonctionnel
 - préparation des cas tests
 - organisation des recettes,
 - ...
- qualité et l'implication des participants !!!
- qualité et efficacité des organisations
- Qualité du développement logiciel : conception, développement, cycle de vie du logiciel

Manque de :

- Précision
- Stabilité
- Complétude

Depuis quand ?

- crise du logiciel en 1968.
 - baisse significative de la qualité des logiciels
 - puissance de calcul des ordinateurs en hausse
 - complexité des logiciels en hausse
 - Retards, dépassement de coût,
 - baisse de la fiabilité,
 - baisse des performances
- Coût important de maintenance
- coût du matériel informatique en baisse // coût du logiciel en hausse
 - études sur les méthodes de travail adaptées à la complexité des logiciels contemporains
=> **génie logiciel**
- l'utilisation des méthodes de génie logiciel se répand *doucement* dans l'industrie du logiciel. **A VOUS DE JOUER :-)**

Vie d'un logiciel

- POURQUOI: Besoins / Problèmes / Exigences
 - *Peuvent être flous, imprécis, irréalistes, incomplets, et changer ! Maturité, évolution marché, contexte entreprise, contexte réglementaire, réorganisation, ...*
- QUOI: Cahier des charges fonctionnel
- COMMENT : Architecture
- COMMENT : Conception, Spécifications techniques
- Dev + tests unitaires, + tests d'intégration
- Maintenance corrective / évolutive
- Réutilisation / évolutions majeures / refonte

Classique mais tellement vrai !

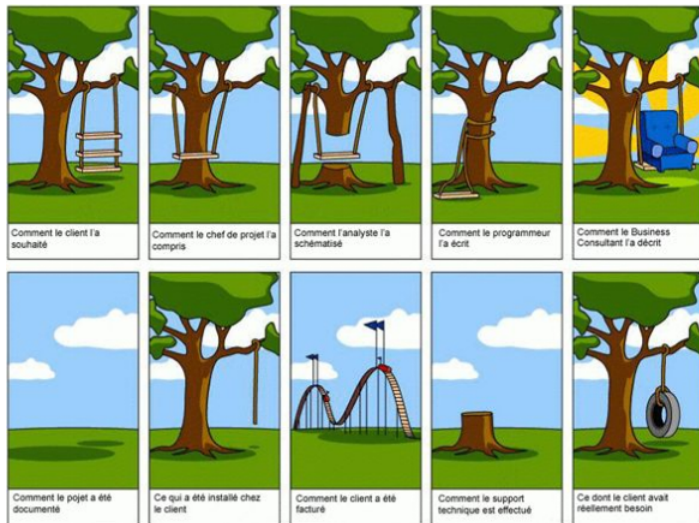


Figure 4: Vous avez connu ?

Définition du Génie Logiciel

- 1) L'objectif du génie logiciel est d'assurer la qualité logicielle et d'industrialiser la production
- 2) Le génie logiciel s'applique à des systèmes de toute taille, et à toutes les phases du cycle de vie d'un logiciel.
- 3) Le Génie Logiciel est la discipline qui se préoccupe de l'application de principes, de pratiques et de connaissances à la construction efficace et effective de systèmes logiciels qui satisfont aux exigences des clients et des utilisateurs.

Cycle de Vie et Outils

- ① Le génie logiciel utilise des techniques d'ingénierie : méthodes, processus, techniques et mesures. Le génie logiciel peut s'appuyer sur l'utilisation d'outils qui gèrent le processus de conception et développement, analysent et modèlent des artefacts logiciels ; montrent et contrôlent la qualité; assurent que l'approche de l'évolution et de la réutilisation est disciplinée et contrôlée.
- ② Ce cycle de vie comprend analyse des exigences et écriture des spécifications, conception (design), construction, tests et recette, installation, mise en œuvre, et maintenance.
- ③ Attention trop de processus tue le processus.

- ❶ L'écriture de logiciel qui peut impliquer soit un unique développeur soit une équipe, nécessite que l'on choisisse les outils, les méthodes, et les approches les plus adaptés et applicables à l'environnement de développement visé.
- ❷ La « boite à out' » du génie logiciel s'améliore et s'étend, les contrats, les patrons de conceptions, en font partie et sont considérés comme des bonnes pratiques.

Outils et contexte

- ❶ Les apports du génie logiciel sont précieux pour le développement d'un logiciel dans n'importe quel domaine d'application où l'on se soucie de professionnalisme, qualité, coût, délais.
- ❷ En résumé, quelque soit le contexte il est nécessaire de réfléchir aux outils que l'on doit mettre en œuvre. Ces choix, d'ordre organisationnel et technique, impactent sur la taille de l'effort nécessaire à produire le logiciel.

Pour ces cours et tds ... pas d'outils autre que le cerveau et le crayon (ou Plantuml).

Deux Grandes Cultures

Développement Discipliné

- ① des Contrats
 - engagement des deux parties
- ② des Documents
 - permettant de définir les objectifs et la recette
- ③ des Métriques
 - pour ne pas avoir d'ambiguïté et pouvoir utiliser des arguments rationnels
- ④ un Langage
 - communs aux deux parties
- ⑤ une Stratégie de décomposition

Développement Agile

- ① un Client
 - Présent en continue qui pilote les évolutions
- ② une Équipe
 - responsable et engagée
- ③ une Relation de confiance entre les parties
- ④ une Stratégie de décomposition

Ingénieur Logiciel ?

Ce qui distingue des autres ingénieurs

- ① Une science à part : Modification Mineure -> Défaut Majeur
 - ① Des grandeurs discrètes au lieu de grandeurs continues
 - ② Phase de MEP et production particulières
 - ③ Une malléabilité traîtresse
 - ④ Un delta de productivité colossal entre différents programmeurs
- ② Des ethnies différentes pour la qualité
 - ① Le debuggage continue MS
 - ② Tout doit être prouvé : Spécifications Mathématiques
 - ③ la qualité c'est d'abord l'architecture : Unix
 - ④ Code and Test
 - ⑤ Test and Code (TDD)

Ce qui distingue II

- ① l'absence de récurrence
 - ① Chaque projet est une découverte
 - ② Iceberg de complexité
 - ③ l'innovation est une nécessité
- ② La maintenance
 - ① Les logiciels ont une maintenance complexe
 - ② Il n'y pas de garagistes, que des "ingénieurs"

Ce qui rapproche

- 1) Qualité
 - 1) Comme dans toutes les activités d'ingénierie c'est la qualité du produit fabriqué qui en fait sa valeur.
- 2) Engagement
- 3) Gestion du risque
- 4) Technicité

Nos questions ?

Quoi/Quand/Comment

- ① Les trois questions auxquelles nous devons répondre
- ② Faire **quoi** dans quel **ordre** avec quelles **techniques** et quels **outils** ?
- ③ Comment contrôler et piloter le processus ?
- ④ Comment faire mieux ?
 - ① Efficacité
 - ② gestion des Risques

QQOQCCP

« Qui ? Quoi ? Où ? Quand ? Comment ? Combien ? Pourquoi ? »

Une méthode de questionnement parmi d'autres. L'essentiel est de se poser toutes les questions, et plusieurs fois !

Mythes du génie logiciel

Mythes de Management

- ① Les managers s'accrochent à des mythes. Comme une personne à une paille pour se sauver de la noyade.
- ② Mythe : Le livre des Standards
 - ① “Nous avons un livre avec tous les standards de la profession”. Cela ne donne t-il pas à tous nos professionnels tout ce qu'ils ont à savoir ?
 - ② Il existe (en fait plusieurs) ! mais est-il utilisé ?
 - ③ Le personnel est-il au courant de son existence ?
 - ④ Ce guide est-il à jour ?
 - ⑤ Est-il complet ?
 - ⑥ Réponses: Non pour toutes ! (et ils sont parfois indigestes...)

Mythes du Management II

- ① Mythe: ordinateurs dernier cri
 - ① Mythe : Mon personnel a ce qu'il faut, j'ai acheté les ordinateurs dernier cri.
 - ② Ce n'est pas l'ordinateur qui fait le logiciel.
 - ③ Il faut mieux de bon logiciel de gestion et développement de projets logiciels que des ordinateurs puissants
 - ④ La plupart des équipes de développement sous-utilisent les logiciels d'aide au développement.
- ② Mythe: Ajouter des Hommes
 - ① Mythe : Si l'on est en retard sur le développement alors ajoutons du personnel
 - ② Réalité : Ajouter des personnes à un projet en retard le rend encore plus en retard (Brooks 75).
 - ③ En effet les nouveaux doivent être formés par les anciens ...

Mythes Clients

- ❶ Les clients ne sont pas informés des difficultés de l'écriture de logiciels. Ils ont des attentes irréalistes du point de vue des développeurs
- ❷ Mythe: Exigences faibles
 - ❶ Mythe : On peut commencer le processus de dev à partir de quelques phrases de description
 - ❷ Réalité : Une mauvaise définition des exigences et des spécifications est la cause la plus importante de l'échec pour les projets logiciels.
 - ❸ Les caractéristiques d'un logiciel ne peuvent être proprement déterminées qu'avec une communication approfondie entre le client et les développeurs.

Mythes Clients II

1) Mythe: Changement

- 1) Mythe : Les exigences changent tout le temps mais ce n'est pas grave car le logiciel est un objet flexible.
- 2) Réalité : le coût d'un changement **croît exponentiellement** avec la phase dans lequel il est introduit. Petite Modification => grosse catastrophe
- 3) en phase de Définition : 1X
- 4) en phase de Développement : 1,5 à 6X
- 5) en phase de Maintenance : 60X à 100X

2) Mythes de la complexité

- 1) Les clients ont une très très mauvaise appréciation de la complexité.
- 2) exemple : Un DAB (distributeur automatique de billets) c'est simple !

Mythes Développeurs

- ① Mythes des Développeurs
 - ① Longuement vu comme un art le passage à l'ingénierie se fait dans la douleur.
 - ② La diffusion dans le grand public complique la donne.
- ② Mythe : ça tourne => c'est fini
 - ① Mythe : Une fois le code écrit notre travail est fini.
 - ② Réalité : 50 à 80 % de l'effort sera fourni après la première livraison au client.

Mythes Développeurs II

- ❶ Mythe: La Qualité se mesure sur le code
 - ❶ Mythe : On ne peut pas mesurer la qualité d'un code avant qu'il ne tourne.
 - ❷ Réalité : Bien des Bugs/Erreurs/C...es peuvent être identifiés dans les étapes de spécifications, d'analyse et de conception.
 - ❸ Certains défauts logiciels ne sont pas détectés par les tests mais par une lecture attentive du code (*Revue de code*)

“Heureusement, nous, utilisateurs, ne devons pas découvrir tous les bugs du logiciel une fois livré ...”

Mythes Développeurs III

- 1) Mythe: l'exécutable est le livrable
 - 1) Mythe: le code est le seul livrable d'un projet réussi
 - 2) Réalité:
 - La documentation
 - Le plan de déploiement
 - Les données
 - Les configurations
 - La formation
 - voire la gestion du changement et la migration et l'organisation du *rollout*

- ① Mythe : les bugs
- ② Mark II : décembre 1947
- ③ Fils de Bugs

0800 Arctan started
1000 " stopped - arctan ✓
1300 (032) MP-MC { 1.2700 9.037 847 025
~~1.30476415~~ 9.037 846 995 convect
(033) PRO 2 4.615925059(-2)
convect 2.130476415
2.130676415
Relays 6-2 in 033 failed special speed test
in relay " " 11,000 test.

Relay 2145
Relay 3376



First actual case of bug being found.
1700 antenae started.
1700 closed down.



La qualité

- ① L'industrie du logiciel est réputée pour la faible qualité de sa production
- ② Dépassement des Délais
- ③ Dépassement des Coûts
- ④ Non conformité aux Exigences ou aux besoins
- ⑤ Un niveau de qualité des livrables suboptimal (présence d'erreurs appelées bugs)



Figure 6: Fix the Problem before it happens

Définir la qualité

- 1) Comment définissez vous la qualité ?
 - 1) pour un sportif
 - 2) une œuvre d'art
 - 3) un produit manufacturé (toaster)
 - 4) café
- 2) un état d'esprit ?
- 3) Référence le cours de Gestion de projet
 - 1) Analyse de la Valeur

Data Quality



Figure 7: La qualité des données

La Norme ISO 9126

- ① La norme ISO 9126 propose une liste de Qualité pour définir la qualité logicielle
- ② Capacité fonctionnelle : Est-ce que le logiciel répond aux besoins fonctionnels exprimés ?
- ③ Fiabilité : Est-ce que le logiciel maintient son niveau de service dans des conditions précises et pendant une période déterminée ?
- ④ Rendement / Efficacité : Est-ce que le logiciel requiert un dimensionnement rentable et proportionné de la plate-forme d'hébergement en regard des autres exigences ?
- ⑤ Maintenabilité : Est-ce que le logiciel requiert peu d'effort à son évolution par rapport aux nouveaux besoins ?
- ⑥ Portabilité : Est-ce que le logiciel peut être transféré d'une plate-forme ou d'un environnement à un autre ?

https://fr.wikipedia.org/wiki/ISO/CEI_9126

Capacité fonctionnelle

- 1) Est-ce que le logiciel répond aux besoins fonctionnels exprimés ?
- 2) Pertinence
- 3) Exactitude
- 4) Interopérabilité
- 5) Sécurité
- 6) Conformité

Fiabilité

- 1) Est-ce que le logiciel maintient son niveau de service dans des conditions précises et pendant une période déterminée ?
- 2) Maturité
- 3) Tolérance aux pannes (voire *Chaos Engineering* !)
- 4) Facilité de récupération
- 5) Conformité

Rendement / Efficacité

- 1) Est-ce que le logiciel requiert un dimensionnement rentable et proportionné de la plate-forme d'hébergement en regard des autres exigences ?
- 2) Comportement temporel
- 3) Utilisation des ressources
- 4) Conformité

Maintenabilité

- 1) Est-ce que le logiciel requiert peu d'effort à son évolution par rapport aux nouveaux besoins ?
- 2) Facilité d'analyse
- 3) Facilité de modification
- 4) Stabilité
- 5) Testabilité ... **vous avez entendu parler des TU et des TNR ?**
- 6) Conformité

Portabilité

- 1) Est-ce que le logiciel peut être transféré d'une plate-forme ou d'un environnement à un autre ?
- 2) Facilité d'adaptation
- 3) Facilité d'installation
- 4) Coexistence
- 5) Interchangeabilité
- 6) Conformité

Usage de cette norme

- ❶ C'est bien d'avoir une norme ... c'est un des mythes du Génie logiciel ...
- ❷ La norme permet d'avoir un référentiel
- ❸ Quand faites vous le test de ce référentiel ?
- ❹ Utile pour donner une direction
- ❺ Quel est le coût de la qualité ?

Le standard qui tue



Figure 8: Standard l'approche capitaliste