

# Génie Logiciel 1

## Introduction



Ecole d'ingénieurs et d'architectes de Fribourg  
Hochschule für Technik und Architektur Freiburg

Télécom 2ème année  
Semestre d'automne  
2018-2019

.....  
**Hes-SO** // Fribourg  
Haute Ecole Spécialisée  
de Suisse occidentale  
Fachhochschule Westschweiz



Pierre Kuonen : [pierre.kuonen@hefr.ch](mailto:pierre.kuonen@hefr.ch)  
Julien Tscherrig : [julien.tscherrig@hefr.ch](mailto:julien.tscherrig@hefr.ch)  
Elena Mugellini : [elena.mugellini@hefr.ch](mailto:elena.mugellini@hefr.ch)



.....  
**Hes-SO** // Fribourg  
Haute Ecole Spécialisée  
de Suisse occidentale  
Fachhochschule Westschweiz  
University of Applied Sciences  
Western Switzerland  
.....

# Motivation

*Les exigences et les attentes à l'égard de la qualité logicielle sont de plus en plus grandes. La taille et la complexité des systèmes informatiques ne cessent d'augmenter.*

*Le développement logiciel ne peut plus être le simple fait du travail « artisanal » d'un développeur aussi ingénieux soit-il. Considérant le rôle critique de certains logiciels, le coût de leur réalisation et les courts échéanciers de production, il est essentiel que développement logiciel intègre une analyse et une conception rigoureuses, une implantation de grande qualité et la vérification.*

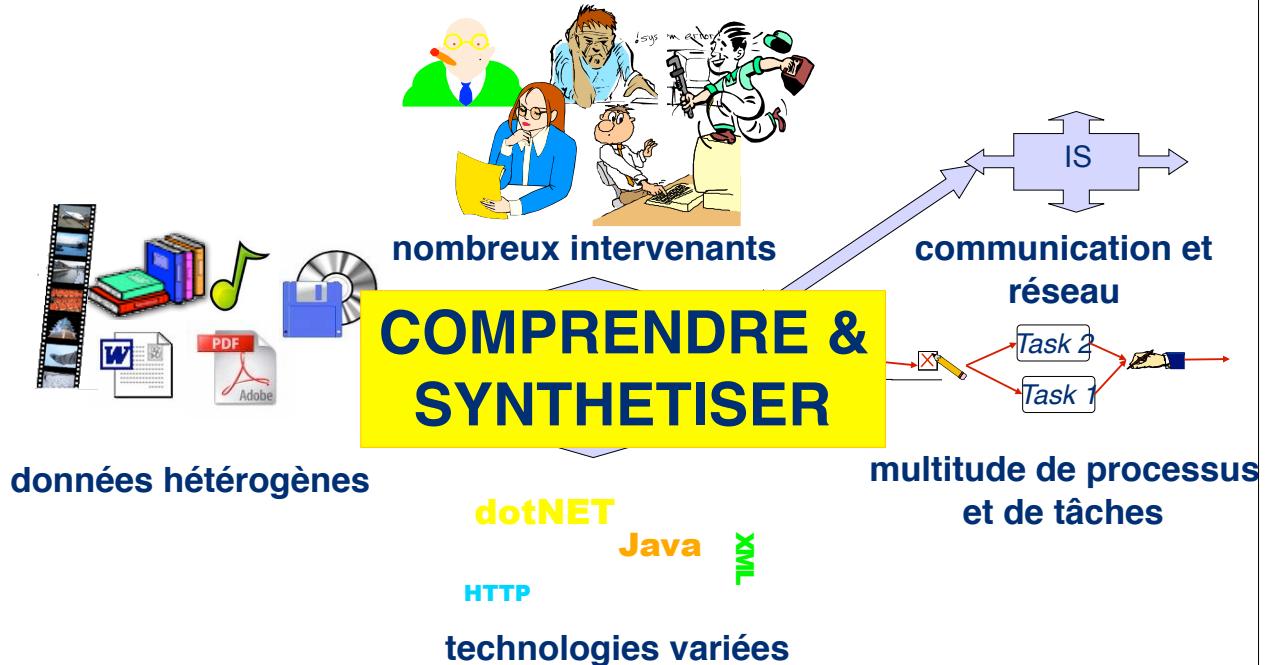
*La culture des « bonnes pratiques » et l'application de certains principes de développement contribuent à l'élaboration de logiciels fiables, performants et facilement modifiables.*

Yann-Gaël Guéhéneuc

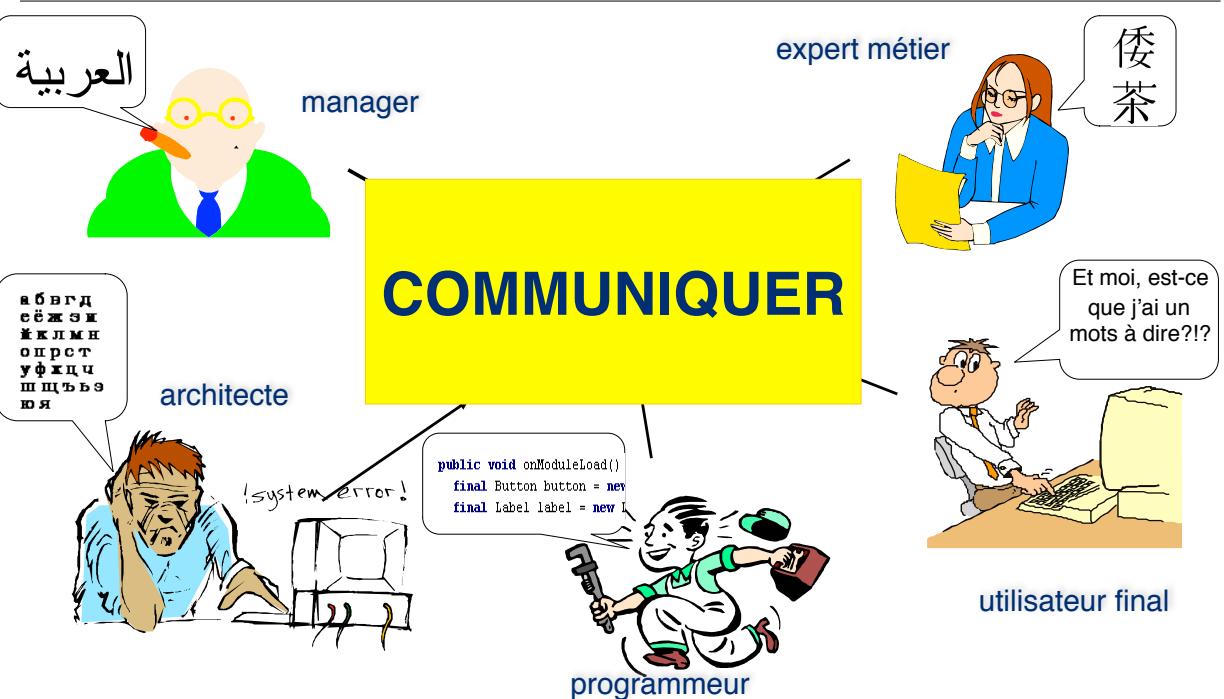
# Problèmes liés à la mis-en place d'un système

- Système (logiciel) très complexe
- Multitâches and multi-processus
- Distribué
- Environnement hétérogènes (*plusieurs acteurs avec différents profils*)
- Hétérogénéité des technologies
- Evolution du système
- Intégration

# Problèmes liés à la mise en place d'un système (suite)



# Problèmes liés à la mise en place d'un système (suite)



# Définitions du Génie Logiciel

- Domaine des « Science de l'ingénieur » dont la finalité est la **conception et la fabrication** de systèmes informatisés complexes, sûrs et de qualité.  
(Jacques Lonchamp, CNAM)
- Un ensemble de **méthodes**, de **techniques** et d'**outils** permettant d'améliorer le processus de production et de **maintenance** des logiciels dans ses aspects techniques et dans son organisation.  
(Bertrand DAVID, Ecole Centrale de Lyon)
- L'application au développement, à la mise en oeuvre et à la maintenance du logiciel d'**une approche systématique, disciplinée et mesurable**; en fait l'application des méthodes de l'ingénieur au logiciel. (IEEE: Institute of Electrical and Electronical Engineers)
- Le génie logiciel est un « **savoir-faire** »

## Domaine adressé par le génie logiciel

- **Domaine du génie logiciel**
  - La construction collective d'un système complexe, constituée par un ensemble de programmes et de documentations associées avec souvent de multiples versions (« multi-person construction of multi-version software »)
- **Par opposition au domaine de la programmation**
  - La production d'un programme par un individu isolé (domaine de la programmation)

# Problème adressé par le génie logiciel

- Se préoccupe des procédés de fabrication des logiciels de façon à s'assurer que
  - Ce qui est fabriqué répond aux besoins des utilisateurs (correction fonctionnelle) et possède les qualités attendues
  - La qualité correspond au contrat de service initial, en termes de performance, sûreté de fonctionnement et robustesse, sécurité, facilité d'évolution, portabilité, convivialité (« user friendliness »)...
  - Les coûts et délais restent dans les limites prévues au départ

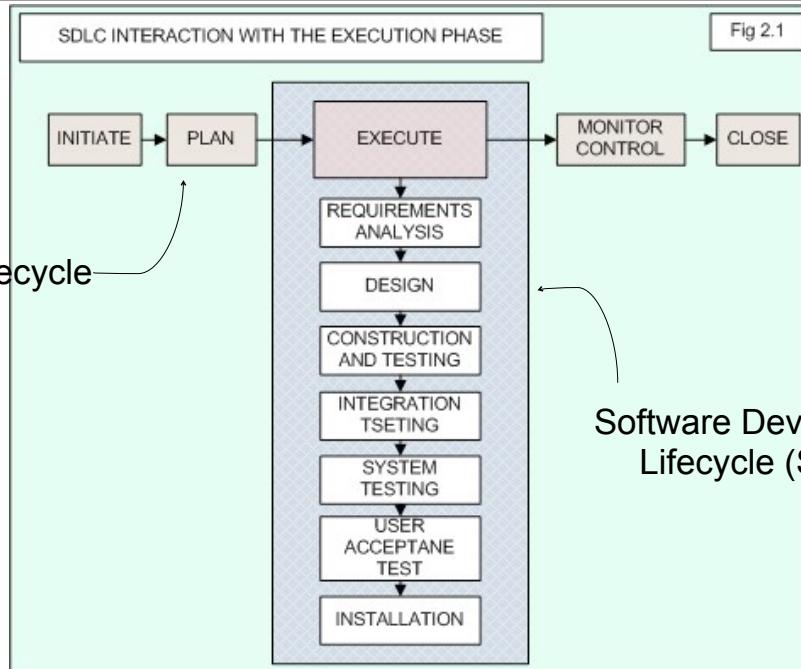
# Deux aspects du génie logiciel

- **La conduite du projet informatique**
  - Utilise des méthodes générales qui ne sont pas forcément spécifiques à l'informatique
  - C'est la partie gestion du travail : chef de projet
    - Gestion des ressources, gestion des coûts, suivi des phases
- **La réalisation du produit logiciel**
  - Utilise des méthodes et outils spécifiques à la réalisation de produits logiciels
    - Méthode COO (Conception Orienté Objet)
    - UML (Unified Modeling Language)
  - C'est la partie technique du travail : suivi des activités
    - Analyse, Conception, Codage, Test

# PLC vs SDLC ?!

Project Lifecycle  
(PLC)

Fig 2.1



## Constatations

- Une des clés de la réussite des projets logiciels réside à **appliquer un processus adapté qui permet de contrôler, de décider et de mesurer le développement du logiciel**

- De plus en plus les informaticiens ne développent plus des logiciels “ex nihilo” mais assembleront des composants “pré-fabriqués” (maturité)

### Raisons des échecs

- Illusion : l'outil va résoudre le problème...
- Promesses non-tenues des standards : ne garantit pas le résultat mais garantit que vous suivez un processus (centré sur la méthode pas sur le résultat)
- Le risque d'échec est proportionnel à la taille du logiciel développé

# Mise en garde aux chefs de projets

- Un livre ne suffit pas à remplacer l'apprentissage d'une méthode pertinente de développement logiciel !
- Un équipement informatique à la fine pointe de la technologie ne garantit en rien une production logicielle de qualité !
- Il ne suffit pas d'ajouter des programmeurs pour accélérer la production logicielle !

# Mise en garde aux clients

- Une description générale des besoins n'est pas suffisante pour débuter le développement d'un logiciel !
- Bien que les logiciels soient flexibles, les modifications tardives peuvent être beaucoup plus coûteuses qu'on ne le pense !

# Mise en garde aux développeurs

- **Le travail du développeur ne se limite pas à produire un programme qui fonctionne !**
  - 60% à 80% du travail devra être consacré à sa maintenance
- **Il existe d'autres méthodes que les tests pour évaluer la qualité d'un logiciel**
  - Méthodes de révisions techniques formelles (par exemple : preuve mathématique, méthode B, etc.)
- **La production de documents lors du développement logiciel n'est pas du temps perdu !**
  - La qualité gagnée réduira les efforts de correction subséquents

## Analogie

- S'il s'agit de construire **une niche de chien**, cela ne nécessite guère un travail exagéré de planification. Les matériaux et outils utilisés seront guère problématiques.
- Dans le cas d'**un garage** il s'agit déjà de respecter un certain nombre de **prescriptions** concernant les dimensions et l'isolation (protection contre l'écoulement d'huile ou d'essence par exemple). Les travaux de planification pourront toujours rester relativement faibles.



# Analogie (suite)

- Dans le cas d'un projet d'**une villa familiale** la situation se présente déjà autrement. En particulier l'**intervention de nombreux corps de métier** nécessite déjà **une planification plus complexe** faisant appelle à une certaine **méthode de travail**.
- Enfin dans le cas de la construction d'**un complexe industriel, commerciale ou publique**, il est absolument indispensable de disposer de **méthodes et d'outils de planification de réalisation et d'un suivi très élaborés**.

