

# Art du Développement Moderne

## Partie 1a: Introduction à *Software Craftsmanship*

Licence Informatique 3<sup>e</sup> année - Tianxiao LIU - CY Cergy Paris Université

# Sommaire

---

- ▶ **Introduction au cours**
  - ▶ Objectifs, programme et organisation, évaluation
  
- ▶ **Introduction à Software Craftsmanship**
  - ▶ Principes de base
  - ▶ Compétences et façons de travail
  - ▶ Fonctionnement du monde de l'industrie logicielle
  - ▶ Carrière avec Software Craftsmanship (introduction générale, à suivre profondément plus tard dans le cours)



# Objectifs du cours

---

- ▶ **Au fur et à mesure**
  - ▶ Compréhension progressive de Software Craftsmanship
  - ▶ Découverte des qualités et compétences d'un craftsman
  - ▶ Pratique : exercices de développement et d'autres types
- ▶ **Buts**
  - ▶ Acquis des compétences de développement
  - ▶ Expériences en de vraies situations
    - ▶ Problématiques et résolutions



# Programme et organisation

---

- ▶ 12 séances de CM – TD
  - ▶ 8H – 12H (la fin avant 12H ou un peu après pour certains, cela dépend de votre niveau et efficacité)
- ▶ **Trois axes pour devenir petit à petit un craftsman**
  - ▶ Qualités de code et code refactoring
  - ▶ Domain Driven Design (DDD) et son application
  - ▶ Architecture logicielle : théorie et application



# Evaluation de l'UE (MCC)

---

## ▶ Note CC (50%)

- ▶ Les rendus des exercices en séances
- ▶ Travail seul pour certains exercices et en binôme pour certains autres
- ▶ Il n'y a pas de projet pour cette UE

## ▶ Examen écrit (50%)

- ▶ Tous documents manuscrits ou imprimés autorisés
  - ▶ **Sauf** livres et appareils électroniques
- ▶ Questions de cours, exercices techniques, solutions pour des cas concrets



# Software Craftsmanship

---

## ▶ Artisanat du logiciel

- ▶ Définition « officielle » : Approche de développement qui met l'accent sur les compétences de codage des développeurs.

## ▶ Un long chemin à parcourir

- ▶ Une façon de pensée : être responsable lui-même de sa carrière, apprentissage non stop...
- ▶ Etre « professionnel » en développement logiciel
- ▶ Beaucoup ont des qualités d'un craftsman mais ne le déclarent (savent) pas « officiellement »...



# Nature du développement logiciel

---

- ▶ **Qu'est-ce que c'est le développement logiciel ?**
  - ▶ Un art ? Les informaticiens sont-ils les artistes ?
  - ▶ Une science ou une ingénierie ?
- ▶ **Plutôt une métaphore, un métier (manuel)**
  - ▶ Ce n'est pas à suivre une procédure standardisée.
  - ▶ Beaucoup de compétences indispensables !
    - ▶ Programmation, conception, communication...
    - ▶ Pas de solutions « magiques » générales



# Principes de Software Craftsmanship

---

## ▶ Valeurs importantes à chercher

- ▶ Développer non seulement un logiciel fonctionnel, mais aussi un logiciel bien fait (artisanat)
- ▶ Développement → ajouter constamment de nouvelles valeurs
- ▶ Besoin d'une communauté de personnes : objectifs communs
- ▶ Logiciel orienté domaine et collaboration étroite avec clients

## ▶ Qualités et risques

- ▶ Réussite du développement d'un logiciel dans le monde industriel



# Un logiciel bien développé

---

- ▶ Critères d'évaluation évoluant
- ▶ Dans les années 90 - début des années 2000
  - ▶ Code incompréhensible, différente définition du travail d'équipe
  - ▶ Difficulté en réutilisation et maintenance
- ▶ Aujourd'hui
  - ▶ Code orienté domaine
  - ▶ Facile à comprendre pour les autres
  - ▶ Maintenable et prédictible



# Le prix du développement logiciel

---

- ▶ **Cela coûte une fortune !**
  - ▶ Il n'y a pas que les développeurs ou ingénieurs.
  - ▶ Testeurs, service de production, managers...
  - ▶ Commerciaux, marketing, back office
  - ▶ Matériels, infrastructure, bureaux...
  - ▶ Distribution, voyages...
- ▶ **Prêt à travailler dans une équipe à l'entreprise ?**
  - ▶ Allez-vous pouvoir y ajouter votre propre valeur ?



# Communauté des craftsmen

---

- ▶ **Partage et apprentissage**
  - ▶ Se développer → Se réaliser
  - ▶ Ceux plus expérimentés → Ceux moins
- ▶ **Les sites Web, les forums**
  - ▶ Bons éléments pour faire progresser l'industrie logicielle
- ▶ **Les gens ayant les valeurs et objectifs**
  - ▶ Expertises, publication et surtout vocation



# Collaboration de tous les acteurs

---

- ▶ **Relation entre employeur et employé**
  - ▶ Point de vue : l'employeur → comme un client
  - ▶ Très différent de la relation traditionnelle
  - ▶ Fierté en ce que l'on fait
  - ▶ Bons projets → réputation et bonnes expériences
- ▶ **Relation entre développeur et client**
  - ▶ Notre logiciel développé doit refléter au domaine
  - ▶ Les clients ont leurs métiers qui sont facilités par notre logiciel



# Problématiques

---

- ▶ **Entreprises encore « traditionnelles »**
  - ▶ Industrie logicielle : standard processus...
  - ▶ Communication Patron – Managers, les autres sont des « travailleurs » qui suivent des ordres...
  - ▶ Développeurs non qualifiés → mais ça coûte moins cher...
  
- ▶ **Résolutions**
  - ▶ Software Craftsmanship n'est pas un jeu « solo »
  - ▶ Une sorte d'attitude et culture pour tous



# S'enrichir avec les livres

---

- ▶ Différents types de livres pour différents objectifs
  - ▶ Livres pour les technologies spécifiques
    - ▶ Utiles mais peuvent vite devenir obsolètes (ne sont plus à jour)
  - ▶ Livres pour les concepts
    - ▶ Des bases solides pour acquérir les technologies plus facilement
  - ▶ Livres pour les « façons de faire »
    - ▶ Travail d'équipe, gestion de projet, agilité...
  - ▶ Livres « révolutionnaires »
    - ▶ Initiation de nouvelles méthodologies / concepts : changements radicaux



# Etre connecté aux bonnes personnes

---

- ▶ Sites Web : très nombreux
  - ▶ Comment sélectionner informations ?
  - ▶ Comment savoir si les informations sont viables ?
- ▶ Identifier les grands contributeurs dans le métier
  - ▶ Comme les catégories des livres
  - ▶ Les gens qui ont initié les idées innovantes et révolutionnaires
  - ▶ Innovant n'assure pas « être accepté par beaucoup de gens »



# Importance de la pratique

---

- ▶ **Apprendre à écrire du code de haute qualité**
  - ▶ Programmer et programmer !
  - ▶ Non seulement trouver une résolution pour une problématique
  - ▶ Trouver une bonne résolution
- ▶ **Difficultés**
  - ▶ Pas facile de changer sa mauvaise habitude
  - ▶ La théorie générale ne suffit pas → c'est pourquoi on fait ADM  
→ beaucoup de méthodes et bonnes pratiques



# Profiter des projets universitaires

---

- ▶ Des occasions très précieuses
  - ▶ Les projets très variés et il y en a un grand nombre
    - ▶ 4 en L2, 7-8 en L3, 7-8 en M1, 7-8 en M2 pro
  - ▶ Souvent des choix techniques ou algorithmiques
  - ▶ Conséquence très différente de celle à l'entreprise...
  - ▶ Vos premières expériences « professionnelles » simulées
  - ▶ Souvent accompagnés par votre passion



# Investissement en soi-même

---

- ▶ **Vous avez vos différents objectifs dans votre carrière**
  - ▶ Traiter chaque travail (simple ou non) comme une occasion de s'enrichir
  - ▶ Réflexion en ROI : Retour sur l'investissement pour soi-même
- ▶ **Ne pas essayer de juger : bon ou mauvais**
  - ▶ Acquérir des connaissances et compétences
  - ▶ Chercher plus de revenus, plus de stabilité (sécurité en travail) est bien légitime



# Autonomie, Progression et Initiative

---

- ▶ Comment trouver vos motivations d'un craftsman ?
- ▶ Autonomie
  - ▶ Avoir le contrôle de 5W + 1H
- ▶ Maîtrise
  - ▶ Devenir meilleur techniquement, professionnellement et humainement
- ▶ Raison d'être
  - ▶ Avoir un rôle important dans le projet / l'entreprise



# Carrière dans les entreprises

---

- ▶ **Choisir votre premier travail dans une entreprise ?**
  - ▶ Les entreprises semblent toutes bonnes ?
  - ▶ Passion au début et déception après ?
  - ▶ Carrière dans une entreprise et carrière personnelle
  - ▶ Intérêts personnels et intérêts de l'entreprise
  
- ▶ **Une carrière idéale en industrie logicielle ?**
  - ▶ Nous allons en parler plus en détail plus tard

