

GEN: initiation au Génie Logiciel

Introduction

Thomas Genet
ISTIC/IRISA Université de Rennes 1

Qu'est-ce que le génie logiciel?

Wikipédia:

“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”

Qu'est-ce que le génie logiciel?

Wikipédia:

“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”

Qu'est-ce que le génie logiciel?

Wikipédia:

“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”

Qu'est-ce que le génie logiciel?

Wikipédia:

“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”

Qu'est-ce que le génie logiciel?

Wikipédia:

“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”

Qu'est-ce que le génie logiciel?

Wikipédia:

“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”

Qu'est-ce que le génie logiciel?

Wikipédia:

“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**”

Qu'est-ce que le génie logiciel?

Wikipédia:

“**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”

Qui utilise ces **procédures systématiques** du génie logiciel?

Qu'est-ce que le génie logiciel?

Wikipédia:

“**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”

Qui utilise ces **procédures systématiques** du génie logiciel?

Tous les développeurs de logiciels (en entreprise ou dans le logiciel libre)

Qu'allez vous faire pendant cette UE?

Vous ne pourrez plus dire:

- ce qu'on voit en cours ne nous servira jamais
- les cours ne parlent pas de choses utilisées dans la "vraie vie"

Qu'allez vous faire pendant cette UE?

Vous ne pourrez plus dire:

- ce qu'on voit en cours ne nous servira jamais
- les cours ne parlent pas de choses utilisées dans la "vraie vie"

Cette UE prend le contre-pied:

- vous allez être confrontés à de vrais problèmes de développement:

Qu'allez vous faire pendant cette UE?

Vous ne pourrez plus dire:

- ce qu'on voit en cours ne nous servira jamais
- les cours ne parlent pas de choses utilisées dans la "vraie vie"

Cette UE prend le contre-pied:

- vous allez être confrontés à de vrais problèmes de développement:
(classés par **importance décroissante**)

Qu'allez vous faire pendant cette UE?

Vous ne pourrez plus dire:

- ce qu'on voit en cours ne nous servira jamais
- les cours ne parlent pas de choses utilisées dans la "vraie vie"

Cette UE prend le contre-pied:

- vous allez être confrontés à de vrais problèmes de développement:
(classés par **importance décroissante**)
 - ▶ travailler en équipe
 - ▶ **communiquer!!**
 - ▶ organiser l'équipe
 - ▶ organiser son travail dans l'équipe
 - ▶ partager son code
 - ▶ **tester** son code
 - ▶ rendre son code lisible, utilisable et **modifiable** par d'autres
 - ▶ documenter son code
 - ▶ ... trouver des solutions algorithmiques, programmer

Qu'allez vous apprendre dans cette UE?

De quoi participer et organiser le développement de gros logiciels:

- Organisés en modules/classes/interfaces (Architecture logicielle)

Qu'allez vous apprendre dans cette UE?

De quoi participer et organiser le développement de gros logiciels:

- Organisés en modules/classes/interfaces (Architecture logicielle)
- Développés à plusieurs (Outil de développement collaboratif)

Qu'allez vous apprendre dans cette UE?

De quoi participer et organiser le développement de gros logiciels:

- Organisés en modules/classes/interfaces (Architecture logicielle)
- Développés à plusieurs (Outil de développement collaboratif)
- Documentés (Générateurs de documentation)

Qu'allez vous apprendre dans cette UE?

De quoi participer et organiser le développement de gros logiciels:

- Organisés en modules/classes/interfaces (Architecture logicielle)
- Développés à plusieurs (Outil de développement collaboratif)
- Documentés (Générateurs de documentation)
- Dont la fiabilité est éprouvée (Test automatisé et Contrats)

Qu'allez vous apprendre dans cette UE?

De quoi participer et organiser le développement de gros logiciels:

- Organisés en modules/classes/interfaces (Architecture logicielle)
- Développés à plusieurs (Outil de développement collaboratif)
- Documentés (Générateurs de documentation)
- Dont la fiabilité est éprouvée (Test automatisé et Contrats)

En bref: quelques outils de base du génie logiciel!

- (Bonus) Apprentissage d'un langage de prog. de pointe: Scala
- (Bonus) Initiation aux méthodes agiles

Pourquoi enseigner le génie logiciel dès le L2?

- Parce que c'est nécessaire dans la formation de tout développeur

Pourquoi enseigner le génie logiciel dès le L2?

- Parce que c'est nécessaire dans la formation de tout développeur
- Pour que vous puissiez, dès maintenant, vous impliquer dans des projets de développement collaboratifs (par ex. logiciel libre)

Pourquoi enseigner le génie logiciel dès le L2?

- Parce que c'est nécessaire dans la formation de tout développeur
- Pour que vous puissiez, dès maintenant, vous impliquer dans des projets de développement collaboratifs (par ex. logiciel libre)
- Pour que vous soyez plus rapidement autonomes en stage

Pourquoi enseigner le génie logiciel dès le L2?

- Parce que c'est nécessaire dans la formation de tout développeur
- Pour que vous puissiez, dès maintenant, vous impliquer dans des projets de développement collaboratifs (par ex. logiciel libre)
- Pour que vous soyez plus rapidement autonomes en stage
- Pour que vous soyez confronté au problème **suffisamment tôt**

Pourquoi enseigner le génie logiciel dès le L2?

- Parce que c'est nécessaire dans la formation de tout développeur
- Pour que vous puissiez, dès maintenant, vous impliquer dans des projets de développement collaboratifs (par ex. logiciel libre)
- Pour que vous soyez plus rapidement autonomes en stage
- Pour que vous soyez confronté au problème **suffisamment tôt**
Permet de comprendre l'importance de beaucoup de cours de L3/M1!

Pourquoi enseigner le génie logiciel dès le L2?

- Parce que c'est nécessaire dans la formation de tout développeur
- Pour que vous puissiez, dès maintenant, vous impliquer dans des projets de développement collaboratifs (par ex. logiciel libre)
- Pour que vous soyez plus rapidement autonomes en stage
- Pour que vous soyez confronté au problème **suffisamment tôt**
Permet de comprendre l'importance de beaucoup de cours de L3/M1!
- Parce que c'est nécessaire au bon déroulement des projets de L3/M1

Evaluation

- Examen terminal, 2h, avec documents (2/5 de la note)

Evaluation

- Examen terminal, 2h, avec documents (2/5 de la note)
- Note de contrôle continu, sans documents (1/5 de la note)
 - ▶ Plusieurs quizz notés pendant les CMs (10 minutes)

Evaluation

- Examen terminal, 2h, avec documents (2/5 de la note)
- Note de contrôle continu, sans documents (1/5 de la note)
 - ▶ Plusieurs quizz notés pendant les CMs (10 minutes)
- Note de travaux pratiques, 2 projets de TP, (2/5 de la note)

Evaluation

- Examen terminal, 2h, avec documents (2/5 de la note)
- Note de contrôle continu, sans documents (1/5 de la note)
 - ▶ Plusieurs quizz notés pendant les CMs (10 minutes)
- Note de travaux pratiques, 2 projets de TP, (2/5 de la note)
 - ▶ Conception de l'architecture logicielle du projet
 - ▶ Développement collaboratif du projet
 - ▶ Génération de la documentation
 - ▶ Test systématique et automatique du logiciel
 - ▶ Validation du logiciel par des tiers (enseignants, autres groupes de TP)

Evaluation

- Examen terminal, 2h, avec documents (2/5 de la note)
- Note de contrôle continu, sans documents (1/5 de la note)
 - ▶ Plusieurs quizz notés pendant les CMs (10 minutes)
- Note de travaux pratiques, 2 projets de TP, (2/5 de la note)
 - ▶ Conception de l'architecture logicielle du projet
 - ▶ Développement collaboratif du projet
 - ▶ Génération de la documentation
 - ▶ Test systématique et automatique du logiciel
 - ▶ Validation du logiciel par des tiers (enseignants, autres groupes de TP)

Attention: évaluation orale **individuelle** sur les projets de TP

- Explication du code produit
- Compréhension de l'architecture globale et de l'**intégralité** du code
- Implication dans le projet: êtes-vous un élément moteur?

Travaux pratiques (demo)

- Conception de l'architecture logicielle du projet
- Développement collaboratif du projet
- Génération de la documentation
- Test systématique et automatique du logiciel
- Validation par des tiers (enseignants, autres groupes de TP)

Outils utilisés en TP (tous libres):

- ScalaIDE 4.3 (Eclipse avec plugins Scala)
- Scala (2.11)
- JUnit 4
- ECLemma
- SVN

Cours et travaux pratiques

- CM1 – Scala: Types, classes, objets, fonctions
- CM2 – Scala: Modèle objet, fonctions récursives, . . .
- TP1 – Développement d'un type Graphe
- TP2 – Exploitation du type Graphe pour système de canalisations

Cours et travaux pratiques

- CM1 – Scala: Types, classes, objets, fonctions
 - CM2 – Scala: Modèle objet, fonctions récursives, . . .
 - TP1 – Développement d'un type Graphe
 - TP2 – Exploitation du type Graphe pour système de canalisations
-
- CM3 – Interfaces, héritage, architecture logicielle, case classes
 - TP3,4,5,6 – Développement d'un Robot web
 - CM4 – Gestion de versions, documentation et TDD
 - TP7 – développement dirigé par les tests (TDD)
 - CM5 – Programmation fonctionnelle et objet avancée

Cours et travaux pratiques

- CM1 – Scala: Types, classes, objets, fonctions
 - CM2 – Scala: Modèle objet, fonctions récursives,...
 - TP1 – Développement d'un type Graphe
 - TP2 – Exploitation du type Graphe pour système de canalisations
-
- CM3 – Interfaces, héritage, architecture logicielle, case classes
 - TP3,4,5,6 – Développement d'un Robot web
 - CM4 – Gestion de versions, documentation et TDD
 - TP7 – développement dirigé par les tests (TDD)
 - CM5 – Programmation fonctionnelle et objet avancée
-
- CM6 – Initiation au développement agile, fonctions d'ordre supérieur
 - TP 8 à 18 – Développement d'un assistant virtuel (office du tourisme)
 - CM7 – Programmation par contrats
 - TP19,20 – Contrats sur une carte à puce

Et les Travaux dirigés?

Pas de Travaux dirigés! (7 CMs et 20 TPs)

- Vous devez **poser des questions pendant les cours** (ou à la fin)
- Vous devez lire la documentation mise à votre disposition
- Vous devez **poser des questions en TP**

Et les Travaux dirigés?

Pas de Travaux dirigés! (7 CMs et 20 TPs)

- Vous devez **poser des questions pendant les cours** (ou à la fin)
- Vous devez lire la documentation mise à votre disposition
- Vous devez **poser des questions en TP**

Les annales des examens de GEN sont disponibles sur le site du **cours**

- Vous devez vous **entraîner sur ces annales**
- Vous pouvez poser des questions à vos encadrants de TP

Et les Travaux dirigés?

Pas de Travaux dirigés! (7 CMs et 20 TPs)

- Vous devez **poser des questions pendant les cours** (ou à la fin)
- Vous devez lire la documentation mise à votre disposition
- Vous devez **poser des questions en TP**

Les annales des examens de GEN sont disponibles sur le site du **cours**

- Vous devez vous **entraîner sur ces annales**
- Vous pouvez poser des questions à vos encadrants de TP

Des quizz notés (sans documents) en CM

- La mauvaise nouvelle: vous devez **connaître** et **comprendre** le cours
- La bonne nouvelle: le cours est réduit! (7 CMs)