Introduction

Buts du cours

- Analyser les besoins d’un·e client·e et modéliser une situation
- Concevoir un logiciel de qualité
  - Correct, réponds au spécifications (soit ne pas faire quelque chose dont lea client·e n’a pas besoin)
  - Robuste, résiste aux situations difficiles
  - Extensible, peut être amélioré et étendu
  - Réutilisable, peut être réutilisable par d’autres logiciels. Soit éviter de réinventer la roue.
- Communiquer de manière claire
- Approfondir notre connaissance du langage UML (type diagrammes vu en bloc 1). L’UML est principalement un moyen de communication pour l’équipe de développement (et pour certain diagrames comme les use cases, avec lea client·e également).

Contenu du cours

- Modélisation des données
- Compléments UML (diagrames use cases, classes, séquences, etc)
- User stories (description d’une fonctionalité à implémenter) et IHM (Interface Human Machine, maquette de l’interface pour communiquer avec lea client·e)
- Méthode d’analyse (agile et scrum)
- Designs patterns (moyens de résoudre des problèmes en informatique quelque soit le langage de programmatoin)

Projet professionel

- Modélisation des besoins d’une vertiable entreprise de notre choix
- Il est très recommandé de ne pas l’avoir en seconde sess car on a un suivi durant l’année mais pas pendant la seconde session
- Travail effectué en groupe de 2

- Remise d’un rapport avec défense orale (pour poser des questions sur le rapport, et pour s’entrainer pour la défense orale du bloc 3), vers le 20 décembre

Examen

- QCM rapide
- Modélisation de situations, approfondissement des diagrammes dont on a déjà parlé en bloc 1 (use cases, séquence, classe, etc)

Types de métiers (débouchés de la formation)

- Administration
  - Administration système (monitoring, backup, restore, installation des logiciels et périphériques, gestion des comptes utilisatuers, droits d’accès, upgrade, etc)
  - Administration réseaux (extension des réseaux, sécurité, politique d’accès)
  - Administration base de données (utilisateurs, droits d’accès, configuration, fine-tuning, validation des scripts, monitoring des performance et volumes)
- Support (première ligne, assistance technique pour utilisateur·ice·s, suivi d’incidents et pannes, procédures, guides utilisateur·ice·s, formation utilisateur·ice·s)
- Informatique décisionnelle/Business Intelligence, prendre des décisions sur base de données en utilisant des programmes (collecte d’informations, reporting, recommendations, etc)
- Formation (concevoir des supports, dispositifs pédagogiques, évaluation, etc)
- Développement de logiciels (chefs de projets, analyste, architecte (définition de la structure de l’application, va définir quel langage, quel frameworks vont être utilisés, etc), dévelopeur, testeur)
  - Lea **chef de projet** est responsable du projet, toujours présent et suis tout le processus, calculer le budget, supervise le développement, organise les équipes de développement, contact avec lea client·e. Iel doit avoir des aptitudes techniques, une bonne communication, un bon leadership et de la rigueur.
  - L’**analyste** doit comprendre les besoins du client, traduire les besoins en modèles, est le lien entre l’équipe de développement et lea client·e. Iel doit avoir une bonne communication, des aptitudes techniques, de la rigueur et une bonne compréhension du domaine du problème ou une capcité à aquérir cette compréhension rapidement (l’informatique est rarement une fin en soit, on fait des programmes pour résoudre des problèmes d’autres personnes. Iel doit donc s’imerger dans le domaine du client pour comprendre comment résoudre le

problème. Le but est de faire gagner du temps au utilisateur·ice·s).
- L'**architecte** va gérer les aspects techniques du projet (structure du logiciel, réutilisation des composants, langages, serveurs, frameworks, etc) en fonction des exigences établies par les analystes (performance, contraintes économiques, *montée en charge*). Iel sera le point de référence technique pour l'équipe de développement et a une collaboration étroite avec le chef de projet (pour le budget et le planning). Iel doit avoir de bonnes compétences techniques, bonne connaissance du paysage IT en place, bonne communication, leadership technique et rigueur. *Voir tiobe*
- Lea **développeur·euse** va écrire le code sur base du travail de l'analyste et de l'architecte, tester le code, automatisation des tests (intégratoin continue), et collaborer avec l'analyste et l'architecte pour détecter des lacunes dans l'analyse, valider le respect de l'architecture.
- Lea **testeur·euse** doit faire des tests fonctionels (fonctionalités demandées), tests de performance (utilisateurs concurrents, bande-passante, pannes, etc), de robustesse (memory leaks, gros volume de données, etc), de vulnérabilité (resistant à differentes attaques, injection SQL, *fuzzing*). Iel doit communiquer les résultats avec lea chef de projet et l'équipe de developpement. Iel doit être rigoueu·euse, méthodique, avoir un bon esprit critique et une bonne communciation et diplomacie.

Mais il y a également plein d'autres nouveaux métiers en lien avec l'intégration continue, les réseaux sociaux, le cloud, le développement mobile, l'intelligence artificielle, etc.

Revision #5
Created 15 September 2023 17:44:10 by SnowCode
Updated 16 September 2023 02:10:32 by SnowCode