

MGL7460: Éléments pour l'examen final (Automne 2016)

- Gestion de la configuration et contrôle du code source
- Langages de script et dynamiques
 - * Principales caractéristiques
 - * Programmation Ruby: symboles, hashes, blocs, map/reduce/select, etc.
- Tests unitaires, cadres de tests et approche TDD
 - * Niveaux de tests
 - * MiniTest: Tests avec assertions vs. avec expectations
 - * Principe d'inversion des dépendances et injection de dépendances
 - * Doublures de tests: stubs et mocks
 - * Approche TDD
- Assemblage de logiciels
 - * Rôle
 - * Quelques outils: make, rake, ant, maven, etc.
- Tests d'acceptation et approche BDD
 - * Tests unitaires orientés "*behavior*"
 - * Tests d'acceptation
 - * Outils pour le BDD: cucumber/gherkin
 - * Autres outils pour les tests d'acceptation: headless browser, Selenium, etc.
- Style de programmation et qualité du code
 - * Abstraction procédurale et couplage
 - * KISS & DRY
- Maintenance de logiciels et qualité du code
 - * Différentes sortes de maintenance
 - * "Clean code" vs. "Code smells"
 - * Maintenabilité
 - * Lignes directrices du SIG pour la maintenabilité
 - * Dette technique et refactoring
 - * Outils d'analyse du code: CheckStyle, PMD, FindBugs, SonarQube
- Intégration, livraison et déploiement continus
 - * Intégration continue
 - * Livraison continue vs. déploiement continu
 - * DevOps
- Langages spécifiques au domaine
 - * Rôle et utilisation
 - * DSL externe vs. interne
 - * DSL internes et API coulante: chainage de méthodes, constructeur et bloc

- Laboratoires

- * Bases du langage Ruby
- * Utilisation du gem gli pour spécifier des applications en ligne de commandes
- * Utilisation de stubs et mocks
- * Spécification de tests "unitaires" avec cucumber
- * Outil d'analyse de code (SonarQube)
- * Outil d'intégration continue (Travis CI)
- * Spécification et mise en oeuvre d'un DSL interne avec API coulante