

LOG8371 : Ingénierie de la qualité en logiciel

Présentation de Cours

Hiver 2017

Fabio Petrillo
Chargé de Cours



This work is licensed under a Creative
Commons Attribution-NonCommercial-
ShareAlike 3.0 Unported License

Introduction

- Coordonnateur, Professeur et chargé de lab.
- Objectifs de l'informatique
- Objectifs du cours
- Méthodologie du cours
- Évaluation du cours
- Sondage

Coordonnateur du cours

Foutse Khomh

foutse.khomh@polymtl.ca

Local M-4123

Chargé de cours et labo

Fabio Petrillo

<http://fabiopetrillo.com>

fabio@petrillo.com

fabio.petrillo@polymtl.com

[@petrillofabio](#)

Local M-4221

Chargé de cours et labo - Fabio Petrillo

- Diplôme d'ingénieur en génie électrique
- Maîtrise en génie logiciel
 - Méthode agile pour l'industrie du jeux vidéo
- **Doctorat** en génie logiciel
 - Compréhension des programmes
 - Débogage
- Plus de 20 ans d'expérience en génie logiciel
 - Architecte logiciel au Brésil
 - Gestionnaire
 - Expérience en systèmes complexes
- Chargé de Course et Lab à Poly (INF3710, LOG8430 et LOG8371)

Objectifs

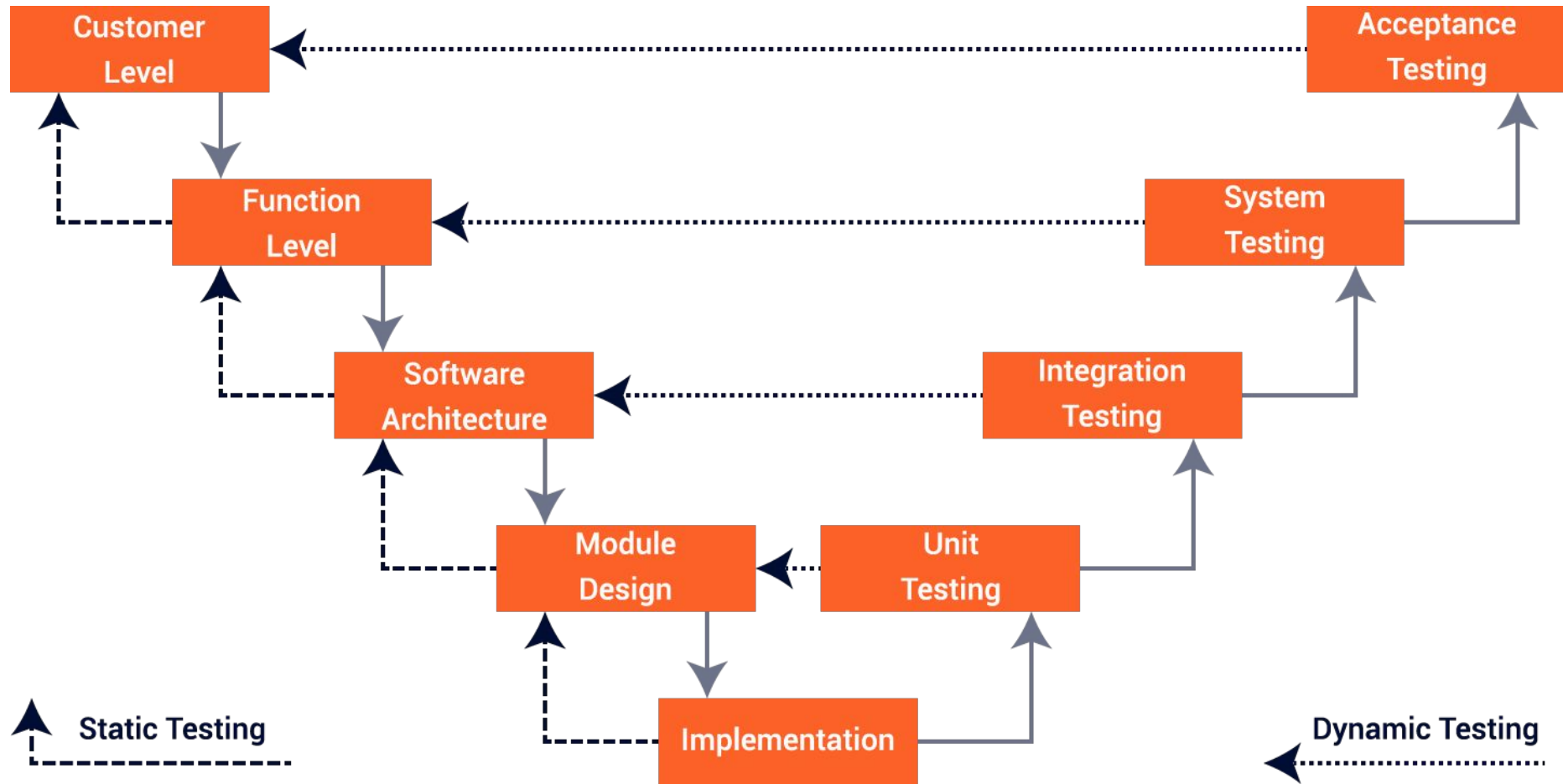
- présenter les concept de qualité logicielle ;
- sensibiliser aux coûts reliés à la qualité ;
- présenter les modèles de l'assurance qualité du logiciel ;
- étudier les normes définissant la qualité en logiciel ;
- explorer les critères pour l'évaluation de la qualité du logiciel ;
- étudier les différentes techniques de revue et d'audit ;
- connaître les pratiques pour la qualité de logicielle ;
- un approche **agile** pour la qualité ;

Contextes de la Qualité Logicielle

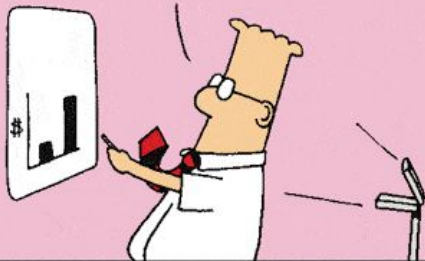
- Contexte de projet
 - processus
 - normes
 - outils
- Contexte des artefacts
 - documentation
 - pratiques de code
 - métriques/analyse de code
 - smells

Assurance Qualité Logicielle en Changement

- Avant, formellement, focus sur le processus
 - Idée centrale: le processus assurait la qualité, **indépendamment** des personnes
 - Waterfall process

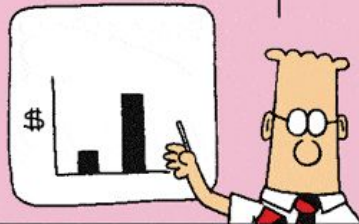


YOU SAVED ONE MILLION DOLLARS BY HAVING PROGRAMMERS IN ELBONIA WRITE SOFTWARE FOR US.



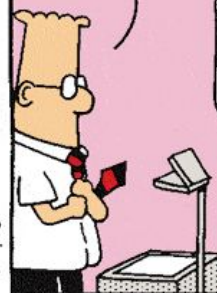
SAdams E-mail: SCOTTADAMS@AOL.COM

BUT WE WASTED FOUR MILLION DOLLARS TRYING TO DEBUG THE SOFTWARE.



2/24/96 © 1996 United Feature Syndicate, Inc.(NYC)

AND THE ENTIRE STAFF OF OUR QUALITY ASSURANCE GROUP QUIT TO BECOME MIMES.



LET'S BLAME THE MIMES; THEY WON'T TALK.



Assurance Qualité Logicielle en Changement

- Avant, formellement, focus sur le processus
 - Idée centrale: le processus assurait la qualité, **indépendamment** des personnes
 - Waterfall process
- Maintenant, focus sur les résultats - **Agilité**
 - Qualité - une responsabilité de **toutEs, tout le temps**
 - Agile testing
 - Clean code
 - Maintainability
 - Code Reviewing
 - Continuous Integration/Delivering
 - Cloud Computing

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

Through this work we have come to value:

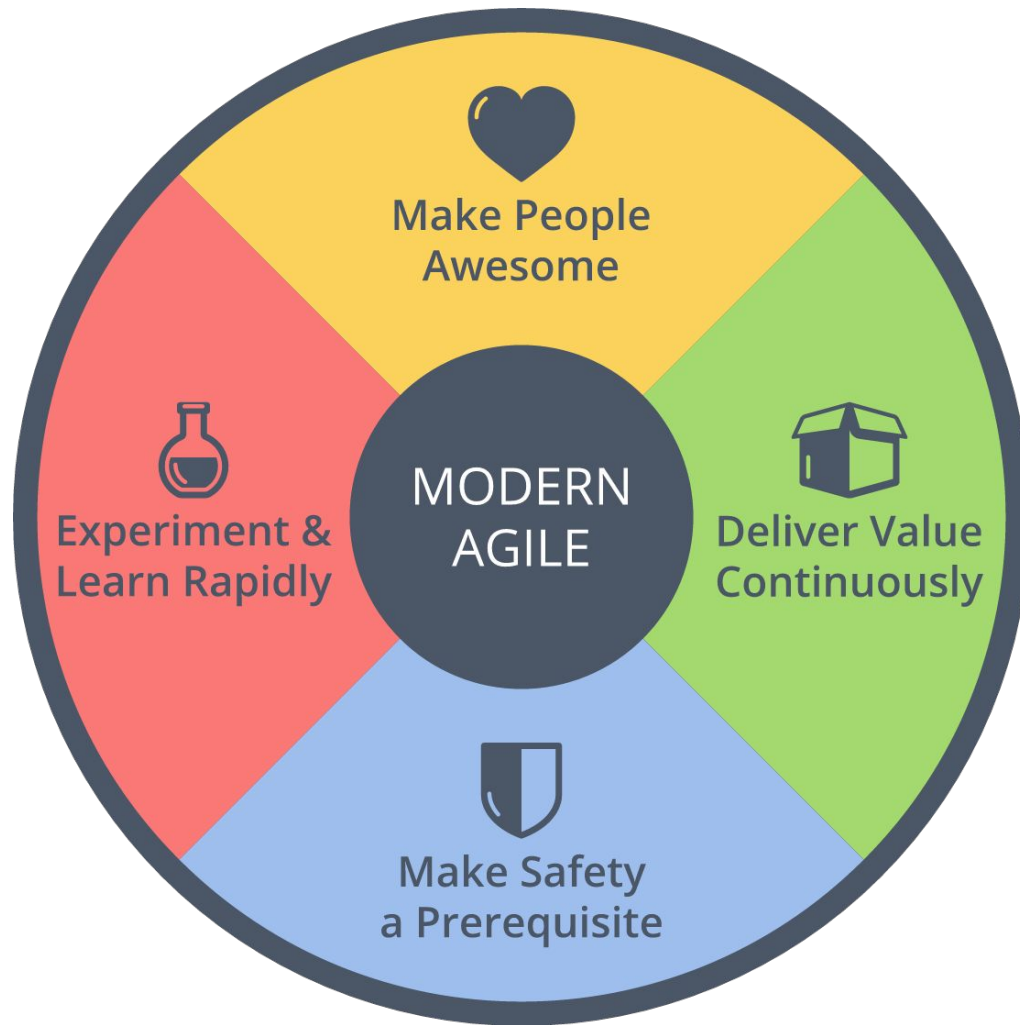
Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.



Software Quality Assurance Jobs Survey

- 13 Jobs
- Canada and USA
- Some big companies: Amazon, Dropbox, Verizon, ...
- Senior positions en general
- Salaty > 80K/year
- What are the most cited word?
- <https://www.jasondavies.com/wordcloud/>
- <http://tagcrowd.com/>

ability agile analyst analytical analyze application architecture assurance
automated bachelor bugs business cases change closely collaborate communication complex
computer continuous create data defects degree deliver design development direction
document drive engineer ensure environment excellent execute
experience framework functional hands-on highly identify implement improve
integration issues java jobs knowledge lead maintain management manual
methodologies mobile multiple oriented participate performance plans practices procedures process
product programming projects provide qa quality related release report
requirements resources responsible results review scripts selenium senior service
skills software specifications sql strategies support systems team technical
technology test tools track understand user validation verify web work years



20 most cited word

Senior Quality Assurance Analyst - Dropbox

*“As the Senior Quality Assurance Analyst, you will define, plan, and build-out QA **testing automation** using leading **best practices**. In this role, you’ll have a wide range of responsibilities including setting the overall testing automation strategy, ensuring the right level of production monitoring is in place, and running manual testing.”*

- Create manual and automated test cases and test scripts
- Execute regression testing, analyze test results, identify and log defects
- Analyze business requirements and technical design specifications to create appropriate test strategies, test plans and test scripts
- Bachelor's Degree in Computer Science, Software Engineering

Quality Assurance Lead - Tervita

*“The ideal candidate will have senior level experience as a QA in an **Agile/Scrum** based environment and will have experience using a variety of **Automated Testing Tools**.”*

- 5+ years hands on testing experience (Functional Web Service/SOA and User Interface Testing, Usability, Exploratory Testing, Performance Testing, Integration Testing, Regression Testing)
- Continuously review of all QA areas and apply best practices and processes
- 5+ years Release Management experience demonstrating ability to deliver quality software through testing, user acceptance, and production environments
- B.Sc. Computer or Software Engineering degree

Situation du cours

- LOG8430 est aussi un cours **gradu **
- Lectures et r sum  d'articles
 -  tre   l'aise avec l'anglais  crit (pr f rable)
- Faire de la recherche
 - Pens e critique
 - Cr er des connaissances
- Concepts avanc s
 -  tre   l'aise avec les concepts OO
 -  tre   l'aise avec des langage de programmation
 -  tre   l'aise installation des outils (se d brouiller)

Programme de cours

Cours	Contenus
01	Plan de cour
02 - 04	Ingénierie de la qualité
05 - 07	Tests en logiciel
08 - 10	Qualité du code
10 - 11	SCM, Gestion des services d'assistance, Intégration/Livraison continue
13	Révision

Organisation des travaux pratiques

TP	Sujet	Pondération	Remise (B1 - B2)
TP1	Analyse d'un processus de qualité logicielle libre ou entreprise	30%	05/02 - 12/02
TP2	Évaluation de la qualité logicielle libre	40%	12/03 - 19/03
TP3	Implémentation et exécution d'un processus de qualité logicielle	30%	02/04 - 09/04

Project d'investigation

- rapport technique un sujet de qualité logicielle
- N. de pages: 10 (\pm 2 pages)
- Format Latex/Doc IEEEtran
- Langue: anglais (préférable) ou français
- Outils d'édition: Overleaf (<http://www.overleaf.com>)

Structure de base pour le rapport

- Abstract: une synthèse de votre papier pour guider le lectures/attirer pour lire le papier
- Introduction: problème ciblé, motivation, proposition, organization du papier
 - avec une (ou des) question(s) de recherche
- Background: les concepts/contexte nécessaire pour comprendre votre papier (references)
- Approche: votre approche pour le problème ciblé
- Résultats: les résultats par rapport l'application de votre approche
- Discussion
- Threats to the validity
- Travaux connexes
- Conclusion: les conclusions et travaux futures

Rapport - Points évalués

- Importance de la contribution du papier et avantages que d'autres peuvent tirer de la contribution: pourquoi la contribution et les avantages sont-ils importants?
- Originalité du travail: quelles nouvelles idées, discussions ou approches sont introduites?
- Validité du travail présenté: comment les chercheurs et les praticiens peuvent-ils utiliser les résultats avec confiance?
- Clarté de présentation;
- Travaux antérieurs pertinents: le travail préalable a-t-il fait l'objet d'un examen adéquat?

Idées pour la recherche

- Révision de la littérature
- Surveys
 - Developers, QA Teams, Managers
 - Artefacts (Github, Stackover Flow, software projects)
- Évaluations
 - Outils, practices, patrons, techniques, ...
- New studies
 - Quality assurance on Open Source Projects: an exploratory study
 - Software Quality Assurance: Myth or Reality?
 - Is CMMI dead?
 - Software Quality Assurance: a developer's perception

Travail personnel du cours

- Triplet horaire
- 4,5 heures de travail personnel par semaine
- Étudier les notions présentés durant les séances de cours, en particulier les nombreux exemples
- Appliquer les concepts vu en cours ainsi que ceux présentés dans les articles discutés en cours
- Travailler sur les labs. du cours
- Préparer le papier

Outils

- Moodle
- Slack - <https://log8371.slack.com>
- Trello - <https://trello.com>
- Overleaf - <https://www.overleaf.com>
- Mendeley - <https://www.mendeley.com/>
- Git - <https://github.com/>

Important

- Toutes matière présentée pendant les séances de cours ou de lab. est contrôlable
- Les travaux remis en retard seront pénalisés de 10% par jour de retard
- Toute demande d'examen différé doit passer par le bureau des affaires académiques
- Les travaux se font par équipes de **3 étudiantEs (max)**. Chaque équipe doit soumettre le travaux sur Moodle.
- Le rapport de recherche est **individuelle et personnelle**.

Important

Les règlements relatifs au plagiat s'appliquent en tout temps



Important

- Assister aux séances de cours et de labs.
- La matière du cours vous intéresse
- Les explications aident à comprendre la matière
- Les étudiants qui assistent régulièrement au cours réussissent mieux le cours
- Les travaux pratiques et l'examen final porteront sur le contenu des séances de cours

Manuel du cours

- Software Quality Assurance – From theory to implementation, Daniel Galin, 2004.
- Assurance qualité logicielle 1, Alain April, Claude Y. Laporte, Hermes, 2011.
- Building Maintainable Software – Ten Guidelines for Future-Proof Code, Joost Visser, 2016.
- Code Quality – The Open Source Perspective, Diomidis Spinellis, 2006
- Clean Code – A Handbook of Agile Software Craftmanship, Robert Martin, 2008

Évaluation du cours

Rapports de TPs

- En équipe
- Objectifs
 - Compréhension
 - Application
- 3 rapports
- **Pondération 30%**

Évaluation du cours

Project d'investigation

- Individuel
- Objectif
 - Compréhension
 - Analyse
 - Synthèse
- **Pondération 30%**

Évaluation du cours

Examen Final

- Individuel
- Résumé fait **à la main** d'une feuille A3 (max) recto-verso
- 4~5 questions obligatoire et de questions optionnelle
- **Pondération 40%**
- Élaboration des questions par les étudiantEs
 - a chaque cours peut demander aux présent à proposer de questions avec la réponse
 - sujet de la séance ou les prochaines (étudiant en avance!!!)
 - je vais utiliser les bonnes questions dans l'examen
 - **l'auteur recevra les points sans faire la questions!**

Sondage

- Origine
 - Quel «génie» (Polytechnique) / École d'origine
- Intérêt à suivre le cours
- Attentes à l'égard du cours
- Répondre avec votre connaissance actuelle (papier ou par courriel)
 - Qu'est-ce que c'est l'assurance qualité logicielle?
 - Qu'est-ce que c'est/comment vous définirait un << code de haute qualité >>?
 - Décrivez votre expérience (professionnelle ou académique) en qualité logicielle?