

## Maîtrise en génie logiciel

### INF 743 : Architecture logicielle

#### Plan de cours

**Session :** A-2017

**Enseignants :** Pierre-Martin Tardif  
Joël Quimper

#### 1. Mise en contexte

Le spécialiste en architecture logicielle a la responsabilité de créer ou de sélectionner le *plan* le plus approprié pour concevoir des solutions mettant en relation plusieurs systèmes et intervenants au sein d'une entreprise. Cet informaticien de haut niveau est une personne clé dans tout développement logiciel puisqu'il est entre autre responsable de s'assurer que le *plan* conçu rencontre les besoins d'affaires, qu'il satisfasse aux exigences des parties prenantes et qu'il soit évolutif.

Au Québec comme ailleurs, la pénurie de spécialistes en architecture logicielle est un frein au développement de systèmes. Les grandes entreprises, tout comme les gouvernements, cherchent à mettre la main sur des personnes qui possèdent les compétences nécessaires à la réalisation d'architectures logicielles. La plupart du temps, ces organisations sont obligées de s'en remettre à quelques firmes informatiques qui prétendent être capables de combler ce besoin. Devoir confier la planification et la conception des architectures logicielles à des firmes externes fragilise les organisations et les rend dépendantes de ces firmes.

Ce cours s'inscrit donc dans un contexte où il y a un besoin de formation pour permettre aux organisations de faire face aux défis des développements logiciels actuels et futurs.

#### 2. Place du cours dans le programme

Ce cours s'inscrit notamment dans la maîtrise en génie logiciel. Les spécialistes en architecture logicielle interviennent à plusieurs stades dans le développement des logiciels de l'entreprise. Le regard qu'ils portent sur le développement logiciel peut être large. Il peut par exemple s'appliquer à l'ensemble du portfolio applicatif, dans le cas de l'architecte d'entreprise. Il peut aussi être plus spécifique et détaillé à un système en se préoccupant uniquement d'une application, voire d'un composant logiciel. Il existe un type de rôle qui intervient à un niveau intermédiaire entre ces deux pôles; on l'appelle généralement l'architecture de solutions. Ce cours vise à former un tel spécialiste.

### 3. Objectifs généraux

Cette activité pédagogique vise à développer chez l'étudiante ou l'étudiant la capacité de :

- O1. Définir une nouvelle architecture logicielle en fonction de l'analyse de besoins ;
- O2. Adapter une architecture logicielle existante à l'évolution des besoins ;
- O3. Migrer d'un système ou d'une architecture à un autre

### 4. Objectifs spécifiques

À la fin de cette activité pédagogique, l'étudiante ou l'étudiant sera capable de :

- OS1. Décrire les principales familles d'architectures logicielles ;
- OS2. Identifier les principaux attributs de qualité de ces familles en lien avec les principales normes en vigueur dans le domaine (ISO/IEC 9126, ISO/IEC 25000) ;
- OS3. Sélectionner l'architecture appropriée à une situation donnée ;
- OS4. Identifier les caractéristiques et les contraintes liées à une architecture existante ;
- OS5. Modifier une architecture existante en tenant compte de ses contraintes intrinsèques et de l'évolution des besoins ;
- OS6. Planifier la migration d'une architecture à une autre ;
- OS7. Procéder à la migration d'une architecture à une autre, tout en facilitant la coexistence, puis la transition entre l'ancienne architecture et la nouvelle.

### 5. Planification hebdomadaire

#### Séance #1 - Présentation du cours et des concepts clés (28 août, JQ)

- Logique d'enchaînement des séances et positionnement du cours avec le programme
- Préparation des équipes et explication du TP
- Définitions et mise en contexte de l'architecture applicative dans l'architecture d'entreprise

#### Séance #2 – Introduction à l'architecture logicielle (11 sept., JQ)

- Fondements et objectifs globaux
- Architectures de référence

#### Séance #3 – Construction d'une architecture logicielle (18 sept., JQ)

- Couches, services et composants
- Patterns
- Formalisme et représentation

#### Séance #4 – Les vues architecturales (25 sept., JQ)

- Fondements, définition et rôles des vues (conceptuelles, logiques, physiques...)
- Comment construire les vues

Considérations fonctionnelles, non-fonctionnelles et opérationnelles

#### Séance #5 – Présentation du TP (2 oct., PMT)

- Quiz #1 (séances 1, 2, 3 & 4 - 60 min.)
- Présentation initiale du TP (15 mn. max.)

#### Séance #6 – Les couches en architecture logicielle (16 oct., PMT)

- Définition et caractéristiques d'un composant
- L'architecture par couche

- Patterns architecturaux

**Séance #7 – Principes d’architecture logicielle appliqués (23 oct., JQ)**

- Patterns architecturaux
- Exemples

**Séance #8 – Intégration d’application et existant (30 oct., JQ)**

- Comprendre et évoluer des existants, quels sont les points d’intégration et les couches
- Architecture orienté service

**Séance #9 – Intégration d’applications et existant (6 nov., JQ)**

- Quiz #2 (séances 5, 6, 7 & 8 - 60 min.)
- Transition et évolution
- Patterns et anti-patterns
- Les logiciels commerciaux (COTS et « buy or build »)

**Séance #10 – Cycle de vie en développement logiciel (13 nov., PMT)**

- Définition et mise en contexte
- Agile, CMMI et Macroscopie
- DevOps
- Introduction aux pratiques pour supporter le SDLC

**Séance #11 – Cycle de vie en développement logiciel (20 nov., PMT)**

- Les pratiques pour supporter le SDLC
- Qualité, Tests et Mesures
- Gestion de configuration et intégration
- Présentation du plan de la solution du TP (15 mn. max)

**Séance #12 – L’architecture logicielle en entreprise (27 nov., PMT)**

- Quiz #3 (séances 9, 10 & 11 - 60 min.)
- TOGAF, principes, cadres de références, points de vue, outils
- Gestion de projet, gestion du « backlog » et travail en équipe
- Gestion des risques
- Gestion des requis

**Séance #13 – Présentation du TP (4 déc., PMT)**

- Trois équipes, quatre présentations de 30 minutes

**Séance #14 – Fin de la matière et révision (11 déc., PMT)**

**Séance #15 – Examen final récapitulatif (18 déc., JQ)**

## **6. Approche pédagogique préconisée**

Cours magistraux, discussions. Un travail d’équipe pratique sur des scénarios. Présentation au groupe et documentation.

## 7. Évaluation de l'apprentissage

Modalités d'évaluation	%
Trois quiz de 15% chacun	45 %
Travail pratique en équipe (10% présentation, 15% documentation)	25 %
Examen final	30 %

## 8. Plagiat

Un document dont le texte ou la structure se rapporte à des textes intégraux tirés d'un livre, d'une publication scientifique ou même d'un site Internet, doit être référencé adéquatement. Lors de la correction de tout travail individuel ou de groupe, une attention spéciale sera portée au plagiat, défini dans le Règlement des études comme « le fait, dans une activité pédagogique évaluée, de faire passer indûment pour siens des passages ou des idées tirés de l'œuvre d'autrui. » Le cas échéant, le plagiat est un délit qui contrevient à l'article 8.1.2 du Règlement des études : « tout acte ou manœuvre visant à tromper quant au rendement scolaire ou quant à la réussite d'une exigence relative à une activité pédagogique. ». À titre de sanction disciplinaire, les mesures suivantes peuvent être imposées : a) l'obligation de reprendre un travail, un examen ou une activité pédagogique et b) l'attribution de la note E ou de la note 0 pour un travail, un examen ou une activité évaluée. Tout travail suspecté de plagiat sera transmis à la vice-doyenne à l'enseignement de la Faculté des sciences.

## 9. Remise des travaux

Tous les travaux doivent être remis dans un format compatible avec Word, clairement identifiés tant au niveau de l'auteur ou des auteurs que de la nature du travail, ainsi qu'être remis de façon électronique le sera par Moodle.

## 10. Bibliographie

- Architecture Logicielle, Concevoir des applications simples, sûres et adaptables – 3<sup>e</sup> édition par Jacques Printz des éditions Dunod
- Software Engineering, Modern Approaches – second edition par Eric J. Braude et Michael E. Bernstein des éditions Wiley
- Software Architecture Foundation, Theory and Practice; Wiley; Richard Taylor, Nenad Medvidovic, Eric M. Dashofy
- The Open Group, TOGAF 9, <http://www.opengroup.org/togaf/>
- International Organization for Standardization, ISO/IEC 9126, [www.iso.org](http://www.iso.org)
- International Organization for Standardization, ISO/IEC 25000, [www.iso.org](http://www.iso.org)
- Software Engineering Institute du Carnegie Mellon University <http://www.sei.cmu.edu/>