

LOG8430 - Architecture Logicielle et Conception Avancée

TP1: Étude et Analyse d'une architecture logicielle

Fabio Petrillo, Manel Abdellatif
Département de génie informatique et génie logiciel
École Polytechnique de Montréal, Québec, Canada
fabio@petrillo.com, manel.abdellatif@polymtl.ca

Session	Automne 2017
Pondération	30% de la note finale du TP
Taille des équipes	3-4 étudiants
Date importantes (23h55 au plus tard)	1 octobre 2017 Remise informelle non notée du TP1 pour avoir des recommandations 15 Octobre 2017 Remise Finale du TP1
Directives particulières	Soumission des travaux par moodle uniquement.
	Toute soumission du livrable en retard est pénalisée en raison de 10% par jour de retard.
En cas de problème lors de la soumission, veuillez contacter la chargée de laboratoire ou le responsable du cours.	

1 Introduction

L'objectif de cet exercice est de mettre en pratique les connaissances que vous avez acquies lors des cours précédents. Il s'agit d'étudier une architecture logicielle, de se familiariser avec, d'étudier la qualité du code, d'avoir une vue globale sur son architecture et son implémentation.

1.1 Déroulement du travail pratique

Vous devez former une équipe de 3 à 4 personnes pour travailler ce TP et envoyer à la chargée de laboratoire vos équipes et votre choix de l'application à analyser (voir Section 2) avant le **24 Septembre 2017**. Si vous désirez travailler sur une autre application veuillez envoyer votre choix également avant cette date pour le valider avec la chargée de laboratoire.

Pour le **1 octobre 2017**: Vous pouvez remettre une première version de votre travail pratique respectant le format des livrables (voir Section 3) et ayant une description pour les étapes 1 à 6. Cette remise ne sera pas notée. Suite à la remise de la première version, vous recevrez ensuite une rétroaction avec les commentaires sur votre travail. Vous devez utiliser ces commentaires pour éventuellement améliorer votre travail avant la remise finale le **15 Octobre 2017**.

2 Travail à réaliser

Veillez faire chaque étape du travail dans l'ordre et fournir votre démarche pour chacune d'elles:

1. Former une équipe de 3 à 4 étudiants.
2. Choisir un logiciel libre à analyser. Nous proposons:
 - Ring: <https://ring.cx/en/download>
 - Ptidej: <https://bitbucket.org/ptidejteam/ptidej-5>
 - JabRef: <http://www.jabref.org/>
 - RocketChat: <https://rocket.chat/docs/contributing>
 - Constellio: <https://constellio.com/how-to-contribute/?lang=en>
3. Analyse architecturale du projet selon le modèle architectural 4+1 view
4. Analyser la qualité du code en utilisant différentes métriques, détailler dans votre rapport ces métriques utilisées et expliquer votre propre analyse suite aux résultats trouvés.
5. Identifier des design patterns (au minimum un) présent dans le code.
 - (a) Définir ce patron de conception.
 - (b) Expliquer votre démarche, comment vous l'avez identifié ?
 - (c) Selon votre avis, pourquoi les développeurs ont utilisé ce patron dans leur code ? quelle est son utilité ? Quelle est son importance sur la maintenance et la qualité logicielle ? Quelle est son impact ?
 - (d) Schématiser à l'aide d'un diagramme ce patron de conception et justifier votre diagramme par un texte.
6. Identifier un défaut de conception (au minimum) présent dans le code.
 - (a) Définir ce défaut de conception.
 - (b) Expliquer votre démarche, comment vous l'avez identifié ?
 - (c) Quel est l'impact négatif sur le code et sur la maintenance ?
 - (d) Quelle solution proposez-vous afin d'améliorer la qualité du logiciel ?
 - (e) Expliquer votre solution et votre choix. (Le POURQUOI: La justification)
 - (f) Schématiser à l'aide d'un diagramme la situation AVANT (avec la mauvaise pratique) et APRES (après avoir appliqué votre solution). Expliquer les diagrammes.

3 Livrable

Chaque groupe doit fournir comme livrable un rapport selon le modèle 4+1 architectural view model et qui ne dépasse pas les 15 pages comprenant:

1. Une présentation générale du projet

2. Logical view: Diagramme de classe...
3. Development view: Diagramme de composants, diagramme de paquetage...
4. Process view: Diagramme de séquence, diagramme d'activité...
5. Physical view: Diagramme de déploiement
6. Scenarios: Diagrammes de cas d'utilisation
7. Analyse de la qualité du projet avec des métriques de qualités bien définies et avec justification des choix de métriques et interprétation des résultats
8. Description des design pattern et du défaut de conception comme indiqué dans les points 7 et 8 de la section 2.
9. Une synthèse de l'analyse architecturale du projet en identifiant les problèmes architecturaux et de la qualité du projet ainsi que les améliorations suggérées pour remédier aux problèmes identifiés.

4 Liens utiles

Liste des outils qui peuvent vous aider dans l'analyse:

- **Understand** <https://scitools.com>
- **CQSE** <https://www.cqse.eu/en/products/conqat/overview/>
- **Ptidej** <http://www.ptidej.net/downloads/tools/ptidejtoolsuite>
Accès au code source: <https://bitbucket.org/ptidejteam/ptidej-5>
- **CppCheck** outil d'analyse statique du code C++ <http://cppcheck.sourceforge.net/>
- **BetterHubCode** <https://bettercodehub.com/>
- **Code Scene** <https://codescene.io/>

Veuillez SVP regarder les exemples des rapports dans le lien ci-dessous et les suivre:

<https://github.com/delftswa2016/desosa2016/>