

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

MGL7460-90 HIVER 2020

REALISATION ET MAINTENANCE DE LOGICIELS

PROJET INDIVIDUEL / V1

PRÉSENTÉ PAR

JEAN-PIERRE PASCAL SUDRE

22 FÉVRIER 2020

Équipe Enseignante :

Professeur : Sébastien Mosser (UQAM)

Laboratoires : Jean-Philippe Gélinas (SFL, Lévis), Jonatan Cloutier (SFL, Montréal)

## Table des matières

1.	Introduction .....	3
2.	Objectifs .....	3
3.	Cadre d'analyse de maintenance .....	3
3.1.	Dimension "Équipe de développement" .....	4
3.2.	Dimension "Exigences" .....	4
3.3.	Dimension "Architecture logicielle" .....	4
3.4.	Dimension "Code source" .....	4
3.5.	Dimension "Tests" .....	5
3.6.	Dimension "Déploiement & Livraison" .....	5

## 1. Introduction

Dans ce projet individuel, l'hypothèse de travail repose sur le fait que ma hiérarchie m'a donné le mandat d'analyser un logiciel légataire afin d'estimer le risque pour notre compagnie d'accepter un projet de tierce maintenance applicative sur ce logiciel.

Dans le cadre de ce projet, je simulerai cette application légataire par une application open-source accessible publiquement.

Mon choix s'est porté sur le logiciel libre WordPress qui est un système de gestion de contenu gratuit, libre et open-source. Ce logiciel écrit principalement en PHP (autres langages : JavaScript, CSS, HTML et XML), repose sur une base de données MySQL ou MariaDB et est distribué par l'entreprise américaine Automattic.

## 2. Objectifs

Dans le cadre de ce projet d'analyse de maintenance de logiciel, il est à ma charge d'identifier la méthodologie à mettre en œuvre pour mener l'analyse de ce logiciel, de la mettre en œuvre, et de livrer mes conclusions vis à vis de la "maintenabilité" du logiciel étudié sous la forme d'un rapport de synthèse.

Voici la première ébauche et la liste des questions sur lesquels je souhaite m'attarder dans la deuxième version de ce rapport. Dans un premier temps, j'ai commencé à effectuer de la recherche sur certaines questions et élaboré une esquisse de réponse, pour les autres les réponses seront élaborées dans la version 2 de ce projet en tenant compte des conseils apportés par l'équipe professorale.

## 3. Cadre d'analyse de maintenance

Dans le cadre de cette analyse, nous avons commencé par effectuer les opérations suivantes :

dépôt de code sur GitHub :

<https://github.com/WordPress/WordPress>

dépôt Git public créé pour le projet:

<https://github.com/TPMGL746090>

dépôt de code importé dans sonarcloud aux fins d'analyse des données :

<https://sonarcloud.io/organizations/tpmgl746090/projects>

### 3.1. Dimension "Équipe de développement"

#### 3.1.1. Les développeurs principaux du projet.

Matt Mullenweg et Mike Little étaient co-fondateurs du projet. Les développeurs principaux sont Helen Hou-Sandí, Dion Hulse, Mark Jaquith, Matt Mullenweg, Andrew Ozz et Andrew Nacin.

WordPress est également développé par sa communauté, y compris les testeurs WP, un groupe de bénévoles qui testent chaque version. Ils ont un accès anticipé aux versions nocturnes, aux versions bêta et aux candidats aux versions. Les erreurs sont documentées dans une liste de diffusion spéciale ou dans l'outil Trac du projet.

Bien que largement développé par la communauté qui l'entoure, WordPress est étroitement associé à Automattic, la société fondée par Matt Mullenweg. Le 9 septembre 2010, Automattic a remis la marque WordPress à la nouvelle fondation WordPress, qui est une organisation soutenant WordPress.org (y compris le logiciel et les archives pour les plug-ins et les thèmes), bbPress et BuddyPress.

Du 5 août 2012 au 21 Février 2020, nous avons relevé 59 contributeurs directs et 39 contributeurs au master excluant les "merge commits".

Date de sortie initiale : 27 mai 2003; dernière version : 5.3.2 (18 décembre 2019)

#### 3.1.2. L'équipe de développement et sa stabilité.

#### 3.1.3. La répartition de la paternité du code source dans l'équipe.

#### 3.1.4. La communication entre les développeurs.

Les développeurs du projet WordPress utilisent [Slack](#) comme la principale plateforme de communication en temps réel. Leur réunion de développement principale a lieu tous les mercredis à 20h00 UTC sur la chaîne #core sur Slack.

### 3.2. Dimension "Exigences"

#### 3.2.1. Le code source et la trace aux exigences.

#### 3.2.2. La vélocité de l'équipe de développement sur la maintenance corrective.

### 3.3. Dimension "Architecture logicielle"

#### 3.3.1. Les composants principaux de l'application.

### 3.4. Dimension "Code source"

#### 3.4.1. La qualification de la qualité du code source.

Voici quelques points qui nous ont poussé à choisir SonarCloud pour l'analyse du code de notre projet. SonarCloud est le produit leader pour la qualité du code continu en ligne, totalement gratuit pour les projets open source. Il prend en charge

tous les principaux langages de programmation, y compris Java, JavaScript, TypeScript, C #, C / C ++ et bien d'autres.  
 Cette application SonarCloud GitHub simplifie plus que jamais l'intégration de nouveaux référentiels de code dans SonarCloud.  
 Notre analyse s'appuie sur l'utilisation de sonarcloud avec le "Quality Gate" par défaut : Sonar way.  
 Voir paramètres ci-dessous.

Metric	Operator	Value
Coverage	is less than	80.0%
Duplicated Lines (%)	is greater than	3.0%
Maintainability Rating	is worse than	A
Reliability Rating	is worse than	A
Security Rating	is worse than	A

  

TPWordpress **Passed** PUBLIC

Last analysis: February 19, 2020, 3:13 PM

886 **E** Bugs | 32 **E** Vulnerabilities | 18k **A** Code Smells | 9.4% Duplications

665k **XL** JavaScript, PHP, ...

JavaScript  
PHP  
CSS  
HTML  
XML

1 of 1 shown

3.4.2.La dépendance du code aux bibliothèques externes.

## 3.5. Dimension "Tests"

3.5.1.Les méthodes de tests mise en place dans le projet.

Les développeurs utilisent également des outils pour analyser les vulnérabilités potentielles, notamment WPScan, WordPress Auditor et WordPress Sploit Framework développé par 0pc0deFR. Ces types d'outils recherchent les vulnérabilités connues, telles que CSRF, LFI, RFI, XSS, injection SQL et énumération des utilisateurs. Cependant, toutes les vulnérabilités ne peuvent pas être détectées par les outils, il est donc conseillé de vérifier le code des plugins, des thèmes et autres compléments d'autres développeurs.

## 3.6. Dimension "Déploiement & Livraison"

3.6.1.Les outils d'intégration / déploiement continus mis en place.

3.6.2.La gestion des dépendances logicielles dans le projet.

3.6.3.La méthodologie de publication (releasing) est mise en place.