

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

MGL7460-90 HIVER 2020

RÉALISATION ET MAINTENANCE DE LOGICIELS

PROJET INDIVIDUEL / V2

PRÉSENTÉ PAR

JEAN-PIERRE PASCAL SUDRE

12 AVRIL 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 .....	3
3.2.	Dimension Exigences .....	5
3.3.	Dimension Architecture logicielle.....	6
3.4.	Dimension Code source.....	6
3.5.	Dimension Tests .....	7
3.6.	Dimension Déploiement & Livraison.....	8
4.	Note de synthèse sur le logiciel .....	8

# 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.

## 3. Cadre d'analyse de maintenance

Dans le cadre de cette analyse, nous avons commencé par créer un Git public pour le projet, nous avons modifié le dépôt Git pour regrouper sous 2 repository toutes les informations aussi bien analyse de code avec SonarCloud ou données statistiques provenant de GitStats :

<https://github.com/JPSUQAM/ProjetIndividuel>

### 3.1. Dimension Équipe de développement

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

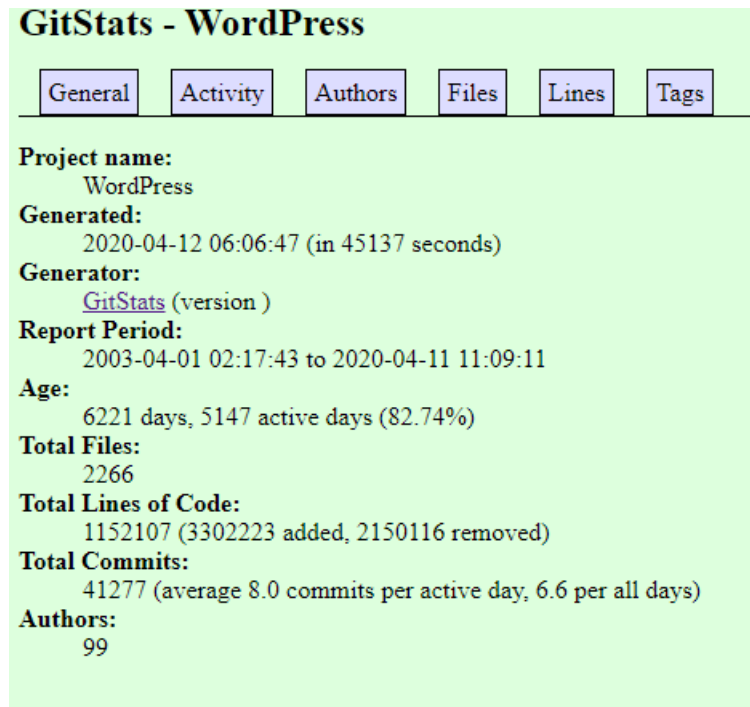
Matt Mullenweg et Mike Little étaient co-fondateurs du projet. Les développeurs principaux sont Helen Hou-Sandí, Dion Hulse, Mark Jaquith, Matt Mullenweg, Andrew Ozz et Andrew Nacin.

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 qui soutient WordPress.org (y compris le logiciel et les archives pour les plugins et les thèmes), bbPress et BuddyPress.

De Mai 2003 à Avril 2020, nous avons relevé 59 contributeurs direct et 39 contributeurs au master avec un total de 41 277 (Source rapport GitStats).

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.

D'après le tableau ci-dessous provenant de GitStats, nous pouvons déduire que l'équipe de développement a été stable au cours des 17 années de développement de WordPress car ryan par exemple durant ses 8 ans d'activité a contribué à 17,33 % du total des commits, il en va de même pour Sergey Biruykov qui est également un important contributeur depuis 7 ans. Nous remarquons que les autres contributeurs de moindre importance contribuent également sur de longues périodes 3 ans et plus).

List of Authors								
Author	Commits (%)	+ lines	- lines	First commit	Last commit	Age	Active days	# by commits
ryan	7153 (17.33%)	677450	452597	2005-02-20	2012-08-09	2726 days, 13:08:32	1630	1
Sergey Biryukov	3442 (8.34%)	253639	419792	2013-01-22	2020-04-11	2635 days, 19:24:41	997	2
nacin	2705 (6.55%)	134481	166188	2010-02-12	2012-08-08	907 days, 4:53:16	524	3
Andrew Nacin	2258 (5.47%)	288330	251331	2012-08-13	2016-12-04	1574 days, 0:37:01	394	4
Scott Taylor	2047 (4.96%)	168418	136798	2013-08-21	2017-08-24	1463 days, 20:47:35	331	5
azaozz	1686 (4.08%)	162279	131644	2008-07-16	2012-08-08	1483 days, 20:46:38	613	6
Drew Jaynes	1664 (4.03%)	53788	40296	2013-09-18	2018-12-19	1917 days, 13:55:40	386	7
Andrew Ozz	1332 (3.23%)	255116	201464	2012-08-10	2020-04-07	2797 days, 17:44:04	678	8
westi	1201 (2.91%)	81897	49296	2007-07-18	2012-08-08	1847 days, 10:30:39	461	9
saxmatt	1200 (2.91%)	81227	63712	2003-04-01	2005-02-17	688 days, 1:45:40	291	10
Dominik Schilling	1150 (2.79%)	294652	65431	2013-04-10	2020-01-06	2461 days, 20:21:10	434	11
markjaquith	1057 (2.56%)	54490	43685	2006-10-03	2012-08-05	2132 days, 14:42:43	423	12
John Blackburn	882 (2.14%)	53092	55379	2014-01-07	2020-03-28	2272 days, 5:36:57	326	13
Boone Gorges	810 (1.96%)	19319	8479	2014-09-30	2019-09-19	1815 days, 0:59:38	351	14
Lance Willett	758 (1.84%)	51673	32071	2012-08-15	2015-07-06	1055 days, 13:00:04	196	15
matt	714 (1.73%)	78169	54365	2005-02-21	2012-02-06	2541 days, 16:30:32	273	16
Gary Pendergast	694 (1.68%)	186023	103090	2014-10-28	2019-09-26	1793 days, 10:23:39	263	17
Weston Ruter	619 (1.50%)	106191	17017	2015-05-25	2019-06-12	1479 days, 1:24:25	286	18
dd32	603 (1.46%)	33022	33053	2010-01-13	2012-08-05	935 days, 3:10:14	182	19
Helen Hou-Sandi	512 (1.24%)	145765	83144	2012-12-18	2018-02-07	1877 days, 2:56:42	263	20

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

Au niveau de la paternité du code, dans les premières années de développement de WordPress, 4 ou 5 développeurs (saxmatt, mikelittle, alex\_t\_king, michelvaldrighi, rboren, ryan, matt) se partageaient la paternité du code avec des contributeurs secondaires mais qui participaient aux commits de manière très prononcée. Depuis les années 2012, la paternité s'est élargie avec plus de diversité au niveau des développeurs qui effectuent de nombreux commits.

### 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.

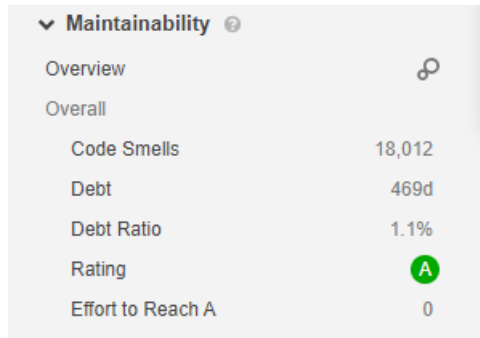
## 3.2. Dimension Exigences

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

Le code source et sa relation avec la trace aux exigences s'effectue entre les développeurs et les utilisateurs au sein de la très grande communauté WordPress.org dans le cadre d'échanges, de forums.

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

Nous relevons que la vélocité de l'équipe de développement en ce qui a trait à la maintenance corrective est très pro-active et rapide à apporter des correctifs car la dette technologique globale est de 1,1%, étant inférieure à 5% il est possible de confirmer que la maintenabilité de l'application est d'excellente qualité.



▼ Maintainability ?	
Overview	
Overall	
Code Smells	18,012
Debt	469d
Debt Ratio	1.1%
Rating	A
Effort to Reach A	0

Source : <https://sonarcloud.io/organizations/jpsuqam/projects>

### 3.3. Dimension Architecture logicielle

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

WordPress se compose d'un système de modèles Web utilisant un processeur de modèles. Son architecture repose sur un contrôleur frontal, acheminant toutes les demandes d'URI non statiques vers un seul fichier PHP qui analyse l'URI et identifie la page cible.

### 3.4. Dimension Code source

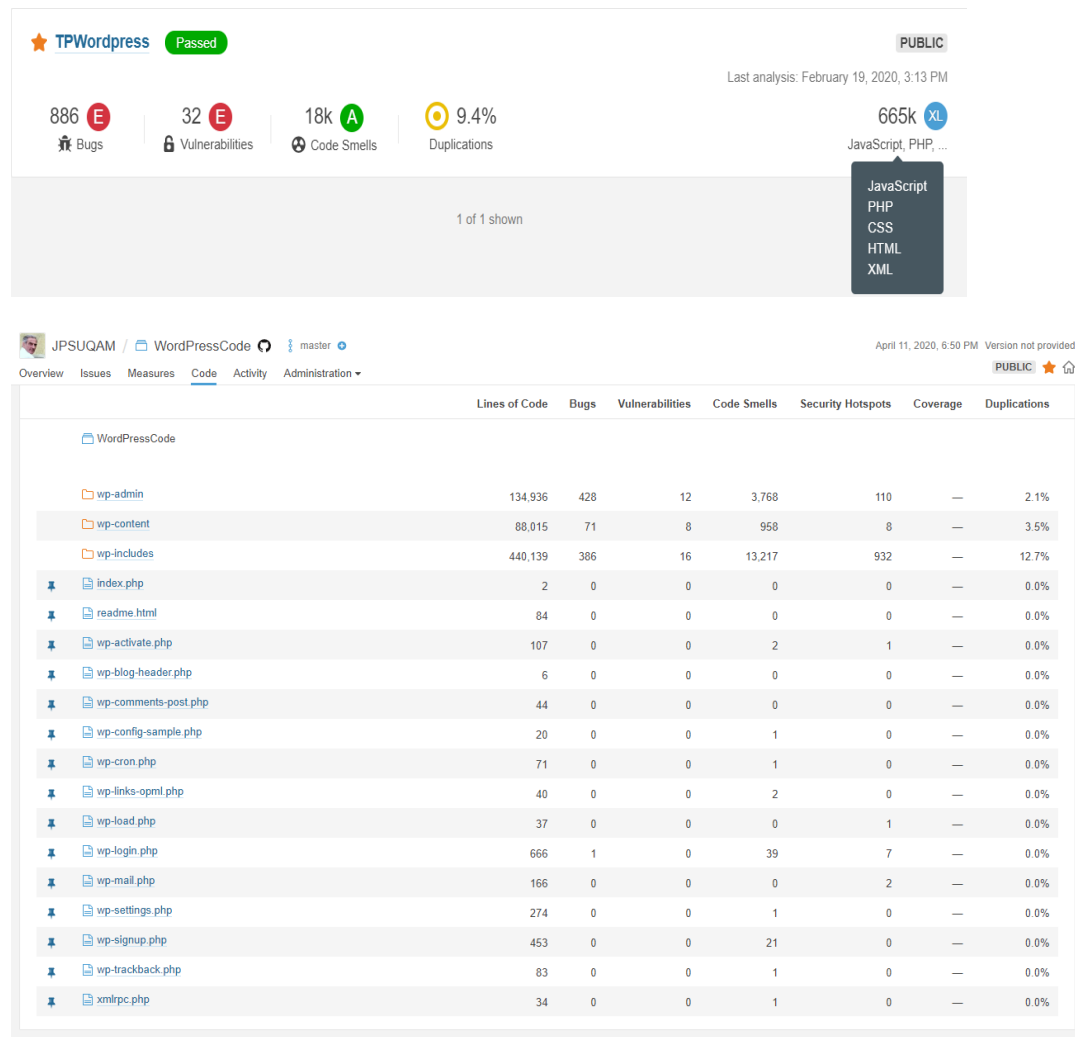
#### 3.4.1. La qualification de la qualité du code source.

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 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



### 3.4.2. La dépendance du code aux bibliothèques externes.

Il est de plus en plus fréquent que pour limiter la dépendance du code aux bibliothèques externes, dans les projets utilisant Wordpress, il est intéressant d'intégrer WordPress avec Composer, Packagist et WPackagist afin de produire un meilleur code.

## 3.5. Dimension Tests

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

PHPUnit est le framework de test officiel choisi par l'équipe principale pour tester leur code PHP.

WordPress utilise PHPUnit, la norme pour les tests unitaires des projets PHP. Les instructions d'installation se trouvent dans le manuel PHPUnit ou sur le référentiel PHPUnit Github. Toutefois, la dernière version de PHPUnit est 8.x, WordPress n'est actuellement compatible qu'avec l'arborescence des versions 7.x.

Côté utilisateur de l'application WordPress, 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.

## 3.6. Dimension Déploiement & Livraison

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

L'approche CI/CD est utilisée par l'équipe de développement.  
CI/CD ou Continuous Integration / Continuous Delivery avec GitLab.

### 3.6.2. La gestion des dépendances logicielles dans le projet.

Nous remarquons que sur WP comme sur n'importe quel CMS ou logiciel open-source, l'exposition aux failles de sécurité. Une installation doit être constamment mise à jour et sauvegardée pour prévenir au maximum l'intégrité du site de tout dysfonctionnement à venir.

La maintenance préventive consiste en deux actions principales : mettre à jour le noyau WP, et les extensions et thèmes utilisés. Sauvegarder souvent et archiver ces sauvegardes de façon incrémentale. Il est donc primordial de connaître les dépendances logicielles (extensions) de l'installation et de faire un suivi de leurs évolutions à venir pour prévenir les risques de régression suite à une mise à jour.

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

Chaque cycle de publication WordPress est dirigé par un ou plusieurs des principaux développeurs WordPress. Un cycle de publication dure généralement environ 4 mois entre la réunion de cadrage initiale et le lancement de la version.

Un cycle de publication suit le schéma suivant :

Phase 1: Planification et sécurisation des chefs d'équipe. Cela se fait dans la salle de chat core sur Slack.

Phase 2: Début des travaux de développement. Les chefs d'équipe assemblent les équipes et travaillent sur les fonctionnalités qui leur sont attribuées.

Phase 3: Beta. Les betas sont publiés et les beta-testeurs sont invités à commencer à signaler les bogues.

Phase 4: Libération du candidat. Le travail est ciblé uniquement sur les régressions et les bloqueurs.

Phase 5: lancement. La version WordPress est lancée et mise à disposition dans l'administrateur WordPress pour les mises à jour.

## 4. Note de synthèse sur le logiciel

Au vu des rapports d'analyse du code de SonarCloud et des données statistiques de GitStats il est à noter que WordPress a le soutien d'une grande communauté de développeur à travers le monde qui facilitent son évolution.



WordPress est la plate-forme CMS préférée pour de nombreuses raisons. Voici les avantages majeurs : Accès à la base de données; accès aux fichiers sources; accès aux feuilles de style CSS; mise à jour régulière possible de la plateforme, de ses modules et de son Template sans frais d'abonnement; évolution du site avec l'ajout de fonctionnalités grâce aux modules développés par la communauté, multitude de Template disponibles créés par la communauté avec le support du développeur inclus; optimisation facile du référencement naturel grâce à l'utilisation de modules dédiés; lecture facile des statistiques Google Analytics (via l'installation de modules); maintenance annuelle suffisante pour des sites simples, propriétaire de son site et de son contenu et autonomie de gestion d'un site internet professionnel.

Il y a quelques inconvénients majeurs : Le soutien dépend du développeur du thème, il faut sélectionner seulement les modules pertinents pour éviter une maintenance trop lourde.

La plateforme WordPress a vu le jour en 2003. À cette époque, ce système de gestion de contenu (CMS) était utilisé principalement par les blogueurs.

WordPress aujourd'hui est une plate-forme de création de site internet open source écrit en PHP (dynamique et statique). C'est probablement le plus simple et le plus puissant système de gestion de contenu de blog et de site Web (ou CMS) existant aujourd'hui.

À cette époque, ce système de gestion de contenu (CMS) était utilisé principalement par les blogueurs. WordPress aujourd'hui est une plate-forme de création des sites internet open source écrit en PHP (dynamique et statique). C'est probablement le plus simple et le plus puissant système de gestion de contenu de blog et de site Web (ou CMS) existant aujourd'hui. WordPress est la plate-forme CMS la plus populaire au monde, alimentant plus d'un tiers des sites Web dans le monde. Des initiatives comme, « Five for the Future »<sup>i</sup> reflète bien le désir de continuité et d'ouverture vers la communauté de développeurs, elle vise à promouvoir la contribution de la communauté WordPress à la croissance de la plateforme. En tant que projet open source, WordPress est créé par une palette diversifiée de personnes du monde entier.

Limitations rencontrées pour ce projet personnel livraison V2 :

J'ai dû prêter à ma fille le portable que j'ai acheté spécifiquement pour le cours lorsque les écoles ont fermé mi-Mars et qu'elle est allée chez sa mère en région pour pouvoir suivre ses cours à distance. Au vu des limitations de mon environnement de travail (Laptop Desjardins, pas Administrateur sur le poste) j'ai même dû demander à un de mes collègues admin sur son poste de me rouler l'analyse GitStats pour le code de WordPress car je ne pouvais pas faire certaines opérations sur mon laptop. J'ai essayé de faire l'Import de mon projet GitHub dans <https://sourceforge.net/> sans succès.

---

<sup>i</sup> Source: <https://wordpress.org/five-for-the-future/>