

LOG1000 – Ingénierie logicielle TP #3

Processus de développement logiciel et d'assurance qualité d'un projet open source

Objectifs:

Explorer les outils utilisés lors du processus de développement et de l'assurance qualité d'un logiciel open source moderne.

Comprendre le processus de développement logiciel dans le cadre des projets open source.

Contribuer au projet open source Ring.

Déroulement du travail pratique

Répondez aux questions reliées aux énoncés concernant les outils utilisés pour le développement et l'assurance qualité d'un logiciel open source.

Ensuite, vous serez ramené à contribuer au projet open source « Ring ». **La contribution pourra être considérée et utilisée réellement par les développeurs et utilisateurs de Ring. En envoyant votre contribution à Ring, vous donnez droit aux développeurs de Ring de modifier votre documentation, tout en gardant vos noms comme contributeurs principaux de la documentation biensure.**

L'ensemble des ressources qui seront consultées en ligne sera en anglais. Si vous voulez de l'aide pour traduire certains termes techniques, n'hésitez pas à en faire part au chargé de laboratoire.

Format du travail remis

Vous devez remettre dans votre répertoire Git un rapport textuel contenant les réponses aux questions des différents exercices. Faites bien attention à rédiger des commentaires constructifs

lorsqu'exigé, une bonne partie de l'évaluation consiste à valider si vous comprenez bien la matière et avez le bon jugement lors de différentes mises en situation.

Pondération

Qualité du rapport	[/15]
E1	[/13]
E2	[/7]
E3	[/10]
E4	[/10]
E5	[/25]
Total	[/80]

Exploration d'un projet open source

Dans la première partie de ce TP (E1-E4), vous aurez à explorer les différents outils en ligne utilisés pour le développement et l'assurance qualité du projet open source WireShark (www.wireshark.org). Ce logiciel est très populaire dans le domaine de télécommunication pour lire et visualiser les paquets transportés dans un réseau (TCP-IP, UDP, etc.).

E1 Révision technique de code source [/ 13]

Dans cette question, nous allons explorer le système de révision technique de code source de Wireshark. Pour cela, connectez-vous à l'adresse <https://code.wireshark.org/review/> , et explorez le site en cliquant sur les différents éléments de la page. Si le site demande de vous connecter, annulez

- Expliquez à quoi sert le dépôt gerrit exactement. [/1]
- Que signifient “open“, “merged” et “abandoned”? [/1]
- Qu'est-ce qu'un “patch”? [/1]
- Cherchez la révision technique Id32864d324f72ffee9c1033f0f36a79eb2651334.

Travail pratique #3: Processus de développement logiciel open source

- e. Identifiez les fichiers qui ont été modifiés et décrivez brièvement ce que fait ce patch.
[1]
- f. Est-ce que le patch a été accepté? Comment est-ce que l'on peut voir ça? [1]
- g. Identifiez les développeurs qui critiquent ce patch. [1]
- h. Identifiez les valeurs de la section "Code-Review" à droite. Que signifient ces valeurs? [1]
- i. Naviguez la partie "History" en bas, qu'est-ce qui s'est passé pour cette révision? Pourquoi cette partie est importante? [1]
- j. Faites le même travail pour la révision technique « Ic34c9fd45740b834ce826d2ce083c615191bdef3 » du développeur "Michael Oed".
[5]

E2 Le répertoire de gestion des versions [/ 7]

Comme on a vu avec Git lors du TP1, les projets open source utilisent également des répertoires pour gérer les différentes révisions des fichiers dans un projet dont le développement est partagé au sein d'une communauté de développeurs. Git est aussi utilisé par WireShark.

Trouvez le répertoire Git de WireShark en parcourant le site Web www.wireshark.org et accédez au "commit" lié à la révision « Id32864d324f72ffee9c1033f0f36a79eb2651334 » (voir ci-dessus), soit à l'aide de l'auteur de ce commit ou à partir de la page Gerrit.

- a. Selon vous, est-ce que cet auteur contribue beaucoup au développement du logiciel (selon l'ensemble de ses "commit") en comparaison avec le commit de ce commit? Comment est-ce que vous avez trouvé la réponse? [2]
- b. Copiez le message complet de ce commit. [1]

Sous la même page, accédez au lien "diff" en bas.

- c. Pourquoi cette section est utile? [1]
- d. Copiez les lignes de code qui ont été modifiées par l'auteur. (Cela inclut les lignes enlevées et ajoutées) [1]

Visitez maintenant le commit 2ddd92b6f8f587325b9e14598658626f3a007c5c:

- a. Est-ce que ce commit ajoute une nouvelle fonctionnalité ou résoud un bogue? Comment est-ce que vous avez déterminé ça? [/2]

E3 Gestion des bogues [/ 10]

Chaque release d'un logiciel a des défauts (bogues). Pour que la communauté utilisant le logiciel soit en mesure de rapporter ces bogues observés, une base de données accessible par une interface web est typiquement utilisée. Dans le cas du logiciel WireShark, l'outil Bugzilla est utilisé. Connectez-vous à l'adresse suivante pour avoir accès au répertoire de bogues de WireShark: <https://bugs.wireshark.org/bugzilla/>.

- a. Trouvez le bogue #9887 (astuce: il y a une barre de recherche). Quel est le nom de la personne qui a trouvé le bogue? Est-ce que ce rapporteur est un être humain? [/1]
- b. Quel est le niveau d'importance de ce bogue? Comment est déterminé le niveau d'importance d'un bogue? Est-ce que ce bogue est critique à un tel point qu'il compromet le fonctionnement du logiciel? (dites pourquoi) [/3]
- c. Combien de personnes on commenté ce problème? (à part la description originale) [/2]
- d. Est-ce que la communauté a confirmé la validité du bogue? Comment? [/2]
- e. Est-ce que le bogue a été résolu? Comment est-ce que vous avez déterminé ça? [/2]

E4 Intégration continue [/ 10]

Nous allons explorer l'outil d'intégration continue du projet Wireshark. Accédez au répertoire Buildbot de Wireshark en allant vers <https://buildbot.wireshark.org/wireshark-master/waterfall>.

- a. Expliquez l'utilité de "Buildbot" dans le contexte d'un projet "open source". [/2]
- b. Identifiez les différents "builders" disponible pour ce projet cliquant sur "Builders" en haut de la page. [/1]

Visitez la page "waterfall":

- a. Que signifient les couleurs vert, rouge et orange? [/1.5]
- b. Que signifient les différentes colonnes? [/1.5]

- c. Identifiez un build échoué ou avec des avertissements, faites un screenshot et expliquez la cause des problèmes en cliquant sur le lien pour voir des messages du build. [/2]
- d. Quel est le numéro de la révision technique de ce build échoué, et qui est le développeur? [/2]

E5 Contribuer au projet Ring: [/ 25]

Dans la deuxième partie du TP, vous allez contribuer à l'amélioration de la documentation du projet open source « Ring ». Pour cela, il faut que vous choisissiez une fonctionnalité à documenter dans le fichier <https://docs.google.com/spreadsheets/d/1agEWeQOY3reVOYrVVSbo2pEca78ciwezU64sAE02jNA>. Si vous avez un UWP ou Linux, svp choisissez une fonctionnalité pour ces plateformes afin de permettre chaque groupe d'exécuter une fonctionnalité sur un appareil personnel.

Le déroulement de cette partie sera comme suit: la première étape consiste à cloner le projet auquel vous allez contribuer, la deuxième section consiste à vous familiariser avec la documentation du projet Ring en faisant un « Hello word », la troisième partie consiste à documenter la fonctionnalité que vous avez choisie. Et dans la dernière étape, vous serez ramené à envoyer votre contribution à l'équipe de Ring.

Attention, cette partie du travail pratique est accompagnée par un document décrivant les étapes nécessaires pour contribuer au projet Ring. Notez qu'il faut d'abord consulter ce document pour avoir plus de contexte, il faut également suivre les consignes, sinon votre contribution risque de directement être rejetée par le projet.

Etape 1 : Préparer l'environnement :

Après avoir choisi la fonctionnalité à documenter, vous devez ouvrir le répertoire « Gerrit » du projet « Ring » et vous authentifier en utilisant un compte Github ou votre propre compte GMail. Cependant, utilisez le lien « Sign-in » de la page suivante <https://gerrit-ring.savoirfairelinux.com> (le répertoire « Gerrit » du projet « Ring »). Et configurer votre local Git comme indiqué dans le paragraphe 1 du tutoriel accompagnant ce TP.

Dans la page « Projects », cliquez sur le lien « List » ensuite « ring-project », copiez la commande « git clone ... » et exécutez-la dans une ligne de commande afin de cloner le projet auquel vous allez contribuer. (paragraphe 2 du tutoriel accompagnant ce TP)

Copiez le résultat de la commande « git clone » dans votre rapport. [2]

Créer ensuite une branche Git sur laquelle vous allez faire votre contribution (paragraphe 3 du tutoriel accompagnant ce TP).

Etape 2 : Faire un « Hello Word » de la documentation de Ring:

Pour générer la documentation, il faut utiliser la commande « make -f Makefile.packaging docs ».

Le dossier « build » sera ainsi créé sous le répertoire « docs ». Pour visualiser la documentation existante, ouvrez avec un navigateur web (Chrome, Firefox ...) le fichier « index.html » qui est dans le répertoire « docs/build/html ».

Les fichiers sources utilisés pour construire cette documentation sont dans le répertoire « docs/source ». Par exemple, le fichier « index.rst » est utilisé pour construire la page web « index.html » que vous avez visualisé. En ouvrant le fichier « docs/source/index.rst » avec un éditeur de texte, vous pouvez remarquer que la phrase « As a new server » qui est dans le premier paragraphe de la page web « index.html » est écrite dans ce fichier.

D'une manière similaire, vous allez faire la documentation de la fonctionnalité que vous avez choisie.

La documentation de Ring est écrite en utilisant RST, inspirez-vous du tutoriel suivant pour rédiger votre documentation <http://docutils.sourceforge.net/docs/user/rst/quickref.html>

Ensuite, sous le répertoire « docs/source/dev/log1000Doc » (créez le dossier log1000Doc s'il n'existe pas) créez un dossier dont le nom est "nomEtudiant1_nomEtudiant2" (remplacez bien sûr nomEtudiant1 et nomEtudiant2 avec vos propres noms) :

- a. Dans le dossier que vous avez créé, ajoutez un fichier dont l'extension est « .rst » qui contiendra votre documentation en respectant le format du nom de fichier suivant « matriculeEtudiant1_matriculeEtudiant2.rst ». Faites une capture d'écran du fichier ajouté dans votre rapport. [1]
- b. Insérez dans ce fichier le texte suivant : [1]

=====

Ceci est un "hello word" de notre documentation.

Changer Matricule1 et Matricule2 par vos propres matricules d'étudiants.

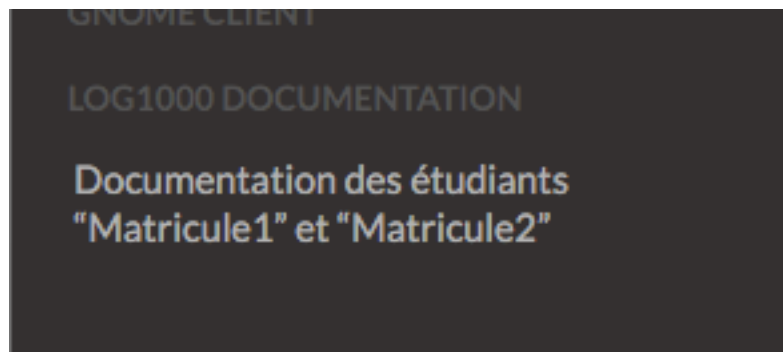
Pour visualiser votre documentation,

- c. Revenez à la ligne de commande et exécutez « `make -f Makefile.packaging docs` » comme indiqué précédemment. Faites une capture d'écran du résultat de cette commande (`make ...`) dans votre rapport. [2]
- d. Ensuite, comme indiqué précédemment, ouvrez la page « `index.html` » pour voir vos modifications. En bas de page, vous pouvez trouver un lien vers votre documentation qui est ajouté. Faites une capture d'écran du lien ajouté dans votre rapport. [2]

LOG1000 documentation

- [Documentation des étudiants "Matricule1" et "Matricule2"](#)

De même, vous pouvez remarquer qu'il y a un lien vers votre documentation dans le menu à gauche de la page :



- e. Cliquez sur l'un de ces deux liens pour voir votre modification comme indiquée sur la figure suivante. Faites une capture d'écran de la page créée dans votre rapport. [1]

Documentation des étudiants "Matricule1" et "Matricule2"

Ceci est un "hello word" de notre documentation.

 Previous

Etape 3 : Documentez Ring:

Maintenant, après avoir appris à faire une documentation dans le projet « Ring » en faisant un « Hello Word » dans la section précédente, vous allez élaborer la documentation de votre fonctionnalité choisie. Assurez-vous de **supprimer le fichier et le dossier** avec vos noms que vous avez créé dans la section précédente.

Notez bien que le dossier dans lequel il faut ajouter la documentation doit être "docs/sources/dev/log1000Doc/", il ne faut surtout rien changer en dehors de ce dossier. En plus, pour éviter des problèmes de conflits avec les autres équipes du TP, les fichiers que vous allez créer doivent impérativement respecter le format du fichier indiqué dans le document des fonctionnalités (google sheet). Et les captures d'écran doivent être ajoutées dans un dossier avec le même format de nom indiqué dans le google sheet.

Par exemple, si vous faite la documentation de la fonctionnalité téléchargement, le nom des fichiers que vous allez créer doivent être sous la forme, "telechargement_<nom de la plateforme>" (comme l'indique le google sheet), par exemple "telechargement_android.rst". Pour les captures d'écran, il faut créer un dossier avec le même nom "telechargement_<nom de la plateforme>" comme "telechargement_android", et dans lequel vous pouvez ajouter vos images de captures d'écran. Attention, ce dossier doit biensure être créé dans le répertoire "docs/sources/dev/log1000Doc/". Par exemple, dans le dossier "telechargement_android", ajoutez la capture d'écran "capture1.png". De la même manière il faut considérer le reste des fonctionnalités.

La documentation de votre fonctionnalité doit contenir du texte et des captures d'écrans, le but est d'expliquer à un nouvel utilisateur de « Ring » la fonctionnalité que vous documenter. Cette explication doit contenir du texte expliquant le but de la fonctionnalité, et les étapes nécessaires pour l'effectuer. Cependant, chacune de ces étapes doit contenir une description et des captures d'écran.

Une fois vous avez terminé votre documentation, faites une capture d'écran de la page web que vous avez créée (la page de documentation). Notez que vous serez évalué aussi par la qualité de votre documentation. [10]

Etape 4 : Envoyez votre documentation aux développeurs de Ring:

Après avoir effectué votre contribution, il faut la proposer aux développeurs de Ring. Pour cela, il faut suivre le tutoriel joint à l'énoncé du laboratoire. Et principalement en suivant les instructions du paragraphe 4 et 5.

Prenez des captures d'écrans de la procédure de révision technique (chacune des étapes du tutoriel joint à l'énoncé du TP, et aussi des captures d'écrans pour les feedbacks que vous allez probablement recevoir sur votre contribution) pour le rapport du TP. [5]

Notez que s'il y a des modifications à faire et à ré-envoyer, il faut exécuter les étapes 4 et 5 du tutoriel joint à l'énoncé.

Notez que l'acceptation ou le refus de votre contribution ne sera pas évalué dans le cadre de ce TP. Notez aussi qu'avant de soumettre votre contribution qu'il faut s'assurer de sa bonne qualité, car elle sera visualisée et révisée par les développeurs de « Ring ».

Considérations importantes pour la fin du TP

Toujours faire un « git add » des nouveaux fichiers, un « git commit » et un « git push » de vos dernières modifications pour que l'on puisse voir la dernière version de votre travail lors de la correction. Si vous ne faites pas de commit, il se peut que l'on évalue une version différente de votre TP local sur le serveur Git.

Vérification que vos travaux sont présents dans le répertoire Git du serveur:

Vous pouvez utiliser la commande « git ls-files » afin de voir les fichiers dans le dernier snapshot de l'entrepôt Git lui-même. Remplacez XX par le numéro de votre équipe. Ce que vous verrez dans cette liste correspond à ce que l'on verra pour la correction.

!! DATE LIMITE DE REMISE !!

Lundi 13 Mars 2017 avant 23h55.

Tout ce qui est «commit» après cette date ne sera pas considéré dans la correction.

!! Pénalités pour retard: 10% par jour !!