

Examen *DevOps* & *Continuous Testing*

- Date: 20 mai 2016
- Durée: 2 heures
- **Aucun document autorisé; barème donné à titre indicatif.**

Vous devez répondre aux questions posées dans le sujet sur la copie d'examen fournie, avec d'éventuelles feuilles intercalaires. Vous ne pouvez pas sortir de la salle d'examen durant la première heure (avant 15h00), ni durant le dernier quart d'heure (après 15h45). Lisez bien tout le sujet avant de commencer à répondre aux questions. Les parties 1 et 2 sont prévues pour représenter environ 45 minutes de travail. Soyez synthétique dans vos réponses (quitte à utiliser des listes à puces), nous n'attendons pas un roman mais une analyse pertinente et précise. N'hésitez pas à mobiliser des compétences acquises dans d'autres cours pour répondre aux questions d'un point de vue global.

Toute fraude identifiée sera transmise au conseil de discipline de l'UNS.

Partie 1: *Question de recul* (5 points)

Durant le mois de mars 2016 à eu lieu l'affaire du "*NPM Gate*". NPM est le gestionnaire de *package* de référence dans l'écosystème Javascript (4 millions d'utilisateurs). Suite à une problématique légale sur le nom d'un de ses *packages* (une entreprise de même nom en réclamant la paternité), un développeur a retiré les 250 *packages* qu'il avait publié dans NPM (aka "*rage quit*"). Or, un de ses *package* (*left-pad*, faisant approximativement 15 lignes de codes permettant d'aligner à gauche du texte) était utilisé en moyenne 2,486,696 de fois par mois par d'autres *packages* (directement ou indirectement). Cette action a eu pour effet de faire s'effondrer tout l'écosystème NPM jusqu'à ce que les gestionnaires de NPM décident de re-publier le *package* en question sans l'accord de son auteur pour remettre sur pied leur système.

Les moyens mis en oeuvre par NPM pour que la situation ne se reproduisent pas lors du retrait d'un *package* sont les suivants¹ :

- *If the version is less than 24 hours old, you can unpublish it. The package will be completely removed from the registry. No new packages can be published using the same name and version.*
- *If the version is older than 24 hours, then the unpublish will fail, with a message to contact support@npmjs.com.*
- *If you contact support, they will check to see if removing that version of your package would break any other installs. If so, we will not remove it. You'll either have to*

¹ <http://blog.npmjs.org/post/141905368000/changes-to-npms-unpublish-policy> (politique officielle)

transfer ownership of the package or reach out to the owners of dependent packages to change their dependency.

- *If every version of a package is removed, it will be replaced with a security placeholder package, so that the formerly used name will not be susceptible to malicious squatting.*
- *If another member of the community wishes to publish a package with the same name as a security placeholder, they'll need to contact support@npmjs.com. npm will determine whether to grant this request. (Generally, we will.)*

Que pensez vous de l'affaire du "NPM Gate", et des moyens mis en oeuvre par NPM pour y remédier (argumentez votre réponse) ?

Partie 2: Continuous Testing (3 points)

Dans le cadre de vos études en Génie Logiciel, vous avez été confronté aux tests unitaires, aux tests d'intégration, aux tests fonctionnels, et aux tests par mutation. Une autre catégorie de test est le "test dirigé par les modèles", i.e., "*Model-based Testing*" (MBT). Vous trouverez ci-dessous la définition de l'approche MBT par Wikipedia² :

"Le model-based testing (MBT) est une activité qui permet de concevoir et de dériver (de manière automatique ou non) des cas de tests à partir d'un modèle abstrait et haut niveau du système sous test (SUT). Le modèle est dit abstrait car il offre bien souvent une vue partielle et discrète des comportements attendus d'un logiciel ou d'un système. Sur la base de modèles abstraits, des cas de test peuvent être dérivées sous la forme de suites de tests. Ces suites de tests ne sont pas directement exécutables, car elles n'ont pas le même niveau d'abstraction que le code exécutable. Cela demande souvent une intervention manuelle de la part d'un ingénieur de test qui doit concevoir une couche d'adaptation permettant de passer d'une suite de tests abstraites en suite de tests exécutables. Cette étape est généralement appelée étape de concrétisation. Une fois les cas de tests exécutés, une comparaison est possible entre le comportement réel du logiciel (le logiciel développé) et le comportement attendu (décrit dans le modèle). La comparaison entre ce qui est attendue et ce qui se passe réellement permet d'assigner un verdict de test. Un test est dit non-passant lorsque le comportement réel du logiciel, ou du système, diffère du comportement attendu. Les outils MBT sont des outils qui peuvent automatiser le design des tests fonctionnels (tests boîte noire)."

Comment mettriez vous en place du MBT sur un projet de développement ? Quels modèles utiliseriez vous, et quelle valeur en serait retiré par le projet (métier ou non-fonctionnelle) ?

Partie 3: Étude de cas (12 points)

² https://fr.wikipedia.org/wiki/Model-based_testing

Raymond et Huguette, deux brillants étudiants de Polytech, ont monté une start-up autour de leur idée, un quizz en ligne pour aider les étudiants à réviser leurs examens.

Ils ont choisi d'héberger leur code dans un dépôt *Github*, depuis laquelle chacun récupère un clone du dépôt sur son ordinateur. Comme ils ont suivi des cours de génie logiciel, ils décident de mettre en place une solution d'intégration continue basée sur *Travis CI*³, qui récupère tout le code, compile et produit le *war* d'une application web. Localement sur leur ordinateur, ils peuvent récupérer la dernière version de l'application web, la tester, et quand ils sont satisfaits, la publier vers une solution d'hébergement gratuite *mycheaphosting.com* qui leur offre un conteneur d'application *Tomcat* et une base de données *MySQL* pour la persistance des résultats. Les étudiants du campus Polytech peuvent ainsi pointer leur navigateur sur l'adresse web <http://mycheaphosting.com/jebossemesexams> et tester leurs connaissances avec les quizzes.

L'idée plaît, et une grande entreprise avec pignon sur rue, Pythie, rachète la start-up. Elle forme une équipe d'une vingtaine de personnes, incluant bien sûr Raymond et Huguette, et leur demande de faire passer le produit à l'échelle. Pythie impose quelques contraintes: le code doit être hébergé sur un serveur de gestion de version SVN interne à l'entreprise. Aucune donnée ne doit jamais sortir du réseau interne, et le développement doit pouvoir être réalisé par une équipe distribuée sur plusieurs continents. De plus, la solution construite doit pouvoir être hébergée sur les *data-centers* de Pythie et monter en charge pour un déploiement sur de nombreux campus.

Vous êtes un membre de l'équipe de **JeBosseMesExams** en charge de mettre en place toute l'infrastructure qui permet à l'équipe distribuée de travailler le plus efficacement possible, jusqu'à la mise en conditions opérationnelles de l'application.

1. Décrivez 3 problèmes avec l'infrastructure de la *startup* avant son rachat
2. Proposez une solution respectant les contraintes de Pythie, reprenant au maximum l'existant et permettant un développement à l'échelle mondiale. Votre solution doit s'attacher à résoudre les problèmes que peuvent rencontrer une équipe distribuée de 20 personnes. Vous couvrirez le besoin de construire l'application jusqu'au point où elle devient deployable dans les *data-centers* de Pythie.
3. Quelle est votre stratégie de test, comment l'implémentez-vous dans la solution décrite en 2 ?
4. Comment envisagez vous de gérer les déploiements des versions successives dans les *data-centers* ? On suppose avoir la contrainte que l'application soit disponible 24 heures sur 24, 7 jours sur 7 (i.e., 24/7).

Vous serez évalués sur votre capacité à justifier vos choix de manière pertinente.

³ Travis CI est une plateforme d'intégration continue offerte "as a service", qui permet de construire des projets hébergés sur github et héberger les artifacts résultants. Tout l'administration se fait à travers le web, et via un fichier de configuration yml décrivant le lien entre projets github et les builds à faire effectuer à la plateforme.