
Java Productivity Primer

Douze recommandations pour augmenter votre
productivité avec une usine logicielle

Juillet 2009



Cabinet d'Architectes en Systèmes d'Information

Table des matières

Introduction	5
Recommandations pour la productivité	8
Recommandation 01 - Choisissez une plate-forme Java productive taillée pour votre projet	8
Recommandation 02 - Mettez en place une usine logicielle adaptée à votre équipe de développement	12
Recommandation 03 - Pour les petits projets, mettez en place une petite usine logicielle	14
Recommandation 04 - Pour les projets plus grands, mettez en place une usine logicielle sur une plate-forme partagée	16
Recommandation 05 - Ne perdez pas votre temps : automatisez votre processus de construction ...	19
Recommandation 06 - Gérez élégamment les développements de l'équipe : utilisez un système de gestion de versions	21
Recommandation 07 - Evitez l'effet tunnel : mettez en place un processus d'intégration et de test en continu	25
Recommandation 08 - Utilisez un système de suivi des tâches	31
Recommandation 09 - Utilisez un environnement de développement intégré productif	32
Recommandation 10 - Mettez les tests au centre de vos développements	34
Recommandation 11 - Déployez tranquille : externalisez les références vers des ressources externes ...	37
Recommandation 12 - Soyez attentif à votre productivité	40
Références	42
À propos d'OCTO Technology	44
Bibliographie OCTO Technology	46

Résumé

Les équipes de développement attendent de Java qu'il les aide à augmenter leur productivité. Le présent document fournit un ensemble de recommandations qui vous aideront à être plus productif avec Java lorsque vous développez des applications d'entreprise. Il comprend une brève présentation des principes du développement moderne, et vous guide à travers les bases de l'organisation de vos activités autour d'une usine logicielle productive, c'est-à-dire un ensemble d'outils pour automatiser les tâches répétitives et aider à mettre en œuvre un workflow de développement efficace. Les experts seront sans doute déjà familiers avec tout ou partie de ces recommandations. Néanmoins, le présent document est un bon moyen de lancer les discussions et de partager les connaissances.

Les auteurs, une équipe d'architectes d'OCTO Technology, conçoivent et fournissent l'assistance pour une telle usine logicielle au sein d'une grande institution financière. Ce document est le fruit de leur expérience sur le terrain, des leçons qu'ils ont retenues dans leur quête de productivité, et des avis de la communauté des architectes logiciels d'OCTO.

Introduction

La plate-forme Java est en perpétuelle évolution. Adopter les nouveaux modèles d'architecture (par exemple les applications Web) et développer des logiciels d'entreprise en utilisant cette plate-forme devient un véritable défi. L'écosystème Java dont la grande vivacité rime aussi trop souvent avec grande complexité, est souvent quelque peu déroutant pour ceux qui découvrent la plate-forme pour la première fois. Certains sont habitués à la productivité que l'on peut attendre d'environnements de développement clients/serveurs classiques comme PowerBuilder ou VB. Du coup, ils éprouvent des difficultés à trouver le moyen d'atteindre, pour ne pas dire dépasser, un tel niveau de productivité avec Java.

Etre productif avec Java ?

Les performances au niveau de la productivité, mises en avant par les utilisateurs et par les fournisseurs Java, est l'une des questions les plus discutées dans la communauté Java. Ce souci de productivité est au cœur des évolutions de Java Enterprise Edition (JEE 5 puis JEE 6 [1]) de SUN. Il a entraîné des changements notables, à commencer par le langage Java lui-même avec Java 5 [2], ainsi que l'adoption de modèles de programmation plus performants, issus de la mouvance des conteneurs légers (les lightweight containers). Toutefois, le développement Java en entreprise s'apparente encore à un parcours semé d'embûches et d'impasses.

Dans les entreprises, les systèmes existants sont souvent intégrés de multiples façons, avec de nombreux flux et référentiels de données de qualité très variables. Ils forment ensemble un empilement de couches logicielles avec lesquelles il faut interagir dans nos projets Java.

Face à ces défis, la communauté des développeurs Java adopte des pratiques améliorant la productivité. Elles combinent des outils de développement et des processus agiles et efficaces, regroupés dans la notion d'usines logicielles (ou software factories). Le présent document a pour ambition de vous présenter ces formes de développement. Il est structuré sous forme de douze recommandations pratiques, saupoudrées de conseils directement applicables.

C'est quoi, la productivité, finalement ?

Même si la productivité est une notion délicate à définir, nous pensons qu'un bon indicateur pour la mesurer est le nombre de fonctionnalités sans erreur, ajoutées au logiciel au cours d'une période donnée.