
Une approche situationnelle pour la définition et l'adaptation d'une méthode d'évolution logicielle pilotée par les données

Jolita Ralyté

*ISS, CUI, Université de Genève
Battelle bâtiment A, 7 Route de Drize, 1227 Carouge, Suisse
jolita.ralate@unige.ch*

RÉSUMÉ. Vous trouverez ici une courte présentation de l'article intitulé "A Situational Approach for the Definition and Tailoring of a Data-Driven Software Evolution Method" co-écrit avec X. Franch et al., et publié dans les actes de la conférence CAiSE 2018.

Mots-clés : évolution logicielle, ingénierie de méthodes situationnelles, processus logiciel

1. Introduction

Le succès de l'évolution des logiciels dépend en grande partie de la sélection des fonctionnalités appropriées à inclure dans la prochaine version. Une telle sélection est difficile et les entreprises relatent souvent de mauvaises expériences concernant l'acceptation des utilisateurs. Pour surmonter ce défi, il existe un nombre croissant d'approches proposant une utilisation intensive des données pour conduire l'évolution. Cette tendance a motivé la création de la méthode SUPERSEDE¹, qui propose la collecte et l'analyse des commentaires des utilisateurs et des données de surveillance comme base de référence pour déterminer et hiérarchiser les exigences, qui sont ensuite utilisées pour planifier la prochaine version. Cependant, chaque entreprise peut être intéressée par l'adaptation de cette méthode en fonction de facteurs contextuels tels que la taille du projet, la portée, l'implication des utilisateurs, etc. Notre objectif a été de définir SUPERSEDE en tant que méthode de référence pouvant être adaptée à différentes situations. Par conséquent, nous avons adopté l'ingénierie de méthodes situationnelles (IMS) (Henderson-Sellers et al., 2014) comme une approche de base pour concevoir la méthode SUPERSEDE et guider son adaptation à un contexte particulier.

¹ SUPERSEDE – projet EU H2020 (www.supersede.eu)

2. SUPERSEDE en tant que méthode situationnelle

Le processus de la méthode SUPERSEDE s'inspire de la boucle de contrôle autonome proposée pour les systèmes adaptatifs (Brun, et al., 2009) et inclus quatre étapes : la collecte de données, leur l'analyse, la décision concernant les nouvelles exigences d'évolution du logiciel, et l'opérationnalisation des décisions. A l'origine, chaque étape de la méthode avait été formalisée en plusieurs éléments : activités, artefacts, rôles et outils. Les activités sont des tâches qui impliquent un ou plusieurs rôles et sont exécutées à l'aide d'outils, qui reçoivent et produisent un ou plusieurs artefacts. Des exemples d'activités sont la collecte de commentaires (la phase de collecte), la définition d'une ontologie de domaine (analyse), la hiérarchisation des exigences (décision) et la planification de la mise en production (opérationnalisation).

Dans ce travail nous avons proposé une redéfinition systématique de la méthode SUPERSEDE en termes d'un catalogue de composants de méthode en suivant les principes et techniques d'IMS. En particulier, en s'inspirant de l'approche proposée dans (Mirbel et Ralyté 2006) nous avons défini le metamodel pour les composants de méthode de SUPERSEDE. En fait, chaque activité avec ses rôles, artefacts et outils liés est devenu un composant de méthode. Le concept de cadre de réutilisation proposé par Mirbel et Ralyté (2006) a servi de base pour définir un ensemble de critères permettant de décrire le contexte dans lequel chaque composant peut être sélectionné et utilisé. Finalement, nous avons suivi l'approche d'IMS par assemblage de composants (Ralyté et al., 2003) pour définir le processus permettant de guider la définition d'une personnalisation de la méthode SUPERSEDE dans un contexte particulier. L'application de la nouvelle mouture de la méthode SUPERSEDE a été testée et approuvée dans un cas de développement et d'évolution d'une plateforme de ville intelligente développé par un partenaire industriel.

Dans cet article nous avons également fait une contribution à la théorie d'IMS en proposant une nouvelle manière de définir le contexte de réutilisation des composants de méthode, notamment en utilisant les modèles de buts étendus avec des annotations contextuelles – une extension de iStar2.0 (Dalpiaz et al. 2016).

Bibliographie

- Brun Y. et al. (2009). Engineering Self-Adaptive Systems through Feedback Loops. *Software Engineering for Self-Adaptive Systems*, Springer.
- Dalpiaz F. et al. (2016). iStar2.0 Language Guide. <https://arxiv.org/abs/1605.07767>
- Franch X. et al. (2018). A Situational Approach for the Definition and Tailoring of a Data-Driven Software Evolution Method. *CAiSE 2018*, LNCS 10816, Springer, pp. 603-618.
- Henderson-Sellers B. et al. (2014). *Situational Method Engineering*. Springer.
- Mirbel I., Ralyté J. (2006). Situational Method Engineering: Combining Assembly-based and Roadmap-driven Approaches. *Requirements Engineering Journal*, 11(1): 58–78.
- Ralyté J., Deneckère R., Rolland C. (2003). Towards a Generic Model for Situational Method Engineering. *CAiSE 2003*, LNCS 2681, Springer, pp. 95-110.