

---

# Formalisation de bonnes pratiques dans les procédés de développement logiciels

Vincent Le Gloahec\* — Regis Fleurquin\*\* — Salah Sadou\*\*\*

\*Alkante SAS, Rennes, France

\*\*IRISA/Triskell, Campus Universitaire de Beaulieu, Rennes, France

\*\*\*Laboratoire VALORIA, Université de Bretagne Sud, Vannes, France

v.legloahec@alkante.com, regis.fleurquin@irisa.fr, salah.sadou@univ-ubs.fr

---

*RÉSUMÉ. Les bonnes pratiques sont des techniques et des méthodes, capitalisées par une entreprise ou par une communauté, dont l'efficacité a été constatée lors des projets de développement. Les bonnes pratiques sont conservées par les entreprises au mieux de manière informelle au détour de guides ou de manuels qualité, au pire elles font partie d'une culture implicite fortement tributaire de certaines individualités. Dans cet article, nous proposons d'exprimer de manière formelle les bonnes pratiques orientées processus. Cela offre deux avantages : d'une part une conservation de manière non ambiguë et structurée et d'autre part la possibilité d'automatiser le contrôle de leur respect. Nous évaluons l'intérêt du langage SPEM pour exprimer les bonnes pratiques. Une expérimentation a été menée avec un partenaire industriel spécialisé dans des projets de développement Web.*

*ABSTRACT. Good practices are techniques and methods capitalized by a company or a community, which effectiveness has been observed in development projects. Those good practices are kept in companies at best in a informal manner through guidelines or quality manuals, and at worst they form a part of a tacit knowledge which strongly depends on some individuals. In this paper, we propose to express in a formal manner those good practices. This approach offers two advantages : on the one hand, the storage in a non ambiguous and structured way, and on the other hand, the possibility to automate the checking of their conformance. We evaluate the relevance of the SPEM language to express good practices. An experiment has been carried out in collaboration with our industrial partner, which is specialized in Web development projects.*

*MOTS-CLÉS : Bonnes pratiques, Ingénierie des procédés, Ingénierie des modèles, SPEM, OCL*

*KEYWORDS: Good practices, Process engineering, Model-driven engineering, SPEM, OCL*

---

## 1. Introduction

Afin de mieux cerner les activités de développement logiciel, d'identifier les liens de cause à effet, et in fine de dégager des zones de gain potentiel, de nombreuses études empiriques ont été conduites. Ces études ont permis d'identifier des "bonnes pratiques" (BP) qui, lorsqu'elles sont mises en oeuvre de manière pertinente, contribuent à augmenter l'efficacité des développements sur le plan de la qualité et de la productivité. Ces pratiques d'origine interne ou externe, mobilisées à bon escient lors des projets, constituent l'expertise, la valeur ajoutée d'une entreprise. C'est un capital précieux pour garantir la satisfaction de ses clients, se distinguer, briguer des labels, des certificats et faire face à la concurrence (Gratton *et al.*, 2005).

Mais faute de moyens qu'elles jugent adéquats, la majorité des entreprises ne rationalisent pas la gestion de ce type de connaissance. Les rares entreprises qui tentent de capitaliser ces connaissances, le font au travers de documents informels, souvent incomplets, parfois mal référencés et épars, conduisant à un usage inadéquat et donc inefficace des BP (Shull *et al.*, 2005). Des travaux proposent d'introduire des processus et des outils informatiques qui facilitent le stockage, le partage et la diffusion de BP au sein des entreprises (Fragidis *et al.*, 2006), (Zhu *et al.*, 2007). Cependant, les BP référencées par ces systèmes ne sont consultables au final qu'au travers de documents textuels et informels.

Nous proposons dans cet article de rendre productives les BP. Nous montrons que, dès lors, il est possible d'adopter une démarche d'ingénierie dirigée par les modèles ; une démarche qui facilite, d'une part le processus de capitalisation des BP et, d'autre part, la vérification automatique de leur respect dans les outils offrant des mécanismes de contrôle de l'activité de développement. Nous nous focalisons, dans la suite du papier, uniquement sur le verrou qui pèse sur la première étape de cette démarche : la définition d'un langage dédié, acceptable par les développeurs et les responsables qualité, qui permettrait de rendre productives les BP. Nous présentons ensuite le langage SPEM et son intérêt théorique pour asseoir les fondations d'un tel langage, en nous appuyant sur une expérimentation conduite au sein d'une entreprise partenaire pour évaluer, cette fois ci in vivo, l'intérêt pratique de SPEM.

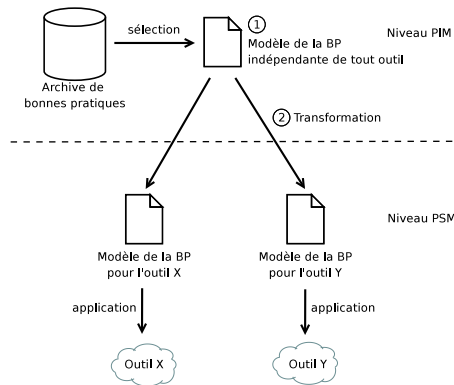
## 2. Une démarche pour la gestion des bonnes pratiques

Notre objectif est, à terme, comme le montre la figure 2, de bâtir un pont entre, d'une part l'univers des dépôts et d'autre part celui des outils offrant des moteurs capables de vérifier le respect de BP. Nous proposons, pour cela, d'adopter une démarche d'ingénierie dirigée par les modèles (IDM) en rendant les BP productives.

Il est important que les BP soient formalisées et stockées dans les dépôts sous la forme de modèles dédiés de niveau PIM, c'est-à-dire indépendant de tout outil. Cette indépendance vis-à-vis des outils est la seule garantissant un savoir pérenne capable de survivre aux évolutions et changements affectant inévitable-

ment les plates-formes de développement. La capitalisation s’opèrerait donc à ce niveau PIM dans un langage dédié aussi puissant et accessible que possible.

Il devient envisageable de transformer, à la demande, de manière aussi automatique que possible les modèles PIM en modèles de niveau PSM conformes aux langages interprétables par les extensions de chaque outil. La mise en place effective d’une telle démarche suppose la levée de deux verrous : i) l’existence d’un langage de niveau PIM pour exprimer les BP, ii) la définition de transformations pour passer des modèles PIM aux modèles PSM. Dans la suite de cet article, nous nous consacrerons uniquement au premier de ces deux verrous.



**Figure 1.** Démarche de transformation des BP.

### 3. Nature des bonnes pratiques

Les BP sont des techniques et des méthodes, capitalisées par une entreprise ou par une communauté, dont l’efficacité a été constatée lors des projets de développement. Dans une entreprise, les BP capitalisées proviennent de deux types de source : des sources externes à l’entreprise (en provenance d’autres entreprises par Benchmarking, d’articles, de rapports, de guides, de méthodes, d’ouvrages faisant autorité, etc.) et des sources internes (par analyse de sa propre activité). Au final, on constate une extrême variété des BP sur le plan de la forme, du niveau d’abstraction et du contexte d’usage : BP de conception, d’architecture, de codage, etc.

Dans cet article, nous nous intéressons uniquement aux BP orientées processus. On peut trouver des exemples de telles BP dans (Thayer, 2002), (McConnell, 1993) et (Phillips, 2004). Nous qualifions une BP “d’orientée processus” lorsqu’elle exprime des contraintes sur la forme que peut prendre tout ou partie d’un processus de développement. Souvent ces BP s’expriment sous la forme d’activités ordonnées agrégées sous la forme de processus. Le respect par exemple d’un cycle de développement en V peut être considéré comme une BP orientée processus. Son expression revient à donner le graphe de séquençement des activités à mener lors d’un projet avec la liste des documents en entrée et sortie de chacune des étapes.

#### **4. SPEM pour modéliser les bonnes pratiques**

Dans cette section nous allons évaluer l'intérêt d'un langage comme SPEM porté par l'OMG pour modéliser les BP. Nous décrivons, en particulier, deux stratégies possibles de modélisation, avant de conclure avec quelques exemples de formalisation de BP à l'aide d'une stratégie implicite.

Nous venons d'illustrer le fait que nombre de bonnes pratiques (BP) intègrent des aspects processus, d'où la nécessité de pouvoir représenter les activités, leur séquençement, les personnes et artefacts qui les composent. L'OMG a justement proposé un langage, SPEM, dédié à l'ingénierie des procédés et intégrant l'ensemble de ces aspects (OMG, 2008). Le standard SPEM2.0 se présente sous deux formes équivalentes : un méta-modèle complet exprimé en MOF et s'appuyant sur le méta-modèle UML2.0, et un profil UML. L'intérêt majeur de ce standard est qu'il présente la particularité d'avoir sa syntaxe abstraite décrite à l'aide de MOF. Il est donc utilisable en l'état dans une démarche IDM avec les outils de transformation existant. Il est aussi possible de raffiner SPEM standard en usant du mécanisme de profil défini dans UML.

Ces raisons font qu'il était naturel d'évaluer la pertinence de ce langage pour servir de base à un futur langage de modélisation de BP. Ce qui nous intéresse maintenant est de savoir comment des BP, souvent décrites de manière informelle, peuvent être représentées dans le contexte de l'ingénierie des procédés à l'aide de SPEM. On peut toutefois distinguer deux stratégies possibles pour modéliser des BP en utilisant le langage SPEM : par modélisation explicite ou par modélisation implicite. Nous allons détailler successivement chacune de ces stratégies.

##### **4.1. Deux stratégies de modélisation**

Il est d'une part possible d'user d'une stratégie par modélisation explicite qui consiste à modéliser directement les bonnes pratiques sous la forme de modèles SPEM à part entière, en les décrivant sous la forme de processus particuliers ou comme des éléments méthodologiques. Cette stratégie a le mérite de permettre un usage complet du langage SPEM pour décrire une BP. En effet, la BP existe au travers d'un modèle SPEM. Le principal inconvénient de cette stratégie est de déconnecter de tout contexte la bonne pratique. Celle-ci ne peut être positionnée dans tel ou tel étape, processus ou domaine.

D'autre part, la stratégie par modélisation implicite consiste à représenter une BP non pas en SPEM mais comme un ensemble de contraintes OCL venant contraindre un processus SPEM particulier défini à un niveau entreprise (un processus générique) ou projet (une instance particulière de processus générique). Cette stratégie permet cette fois-ci de contextualiser la BP en l'associant à un processus cible. Mais elle limite le pouvoir d'expression à disposition à celui d'OCL. L'inconvénient de cette stratégie

implicite est qu'une BP ainsi exprimée est, à l'usage, clairement moins puissante, lisible et maintenable que pour la stratégie précédente.

#### 4.2. *Expérimentation dans un contexte de développement Web*

Face à la complexité croissante des projets et aux problématiques liées à la maintenance logicielle, notre partenaire industriel Alkante tend à faire évoluer ses processus de développement classiques. Dans ce contexte, nous proposons un retour d'expérience sur l'usage des langages SPEM et OCL pour la formalisation de BP que souhaitent documenter la société. Il s'agit aussi bien de BP de la littérature issues de méthodes reconnues que de pratiques internes émanant de l'expérience acquise par la société lors de ses développements.

La méthode RUP, par exemple, propose de nombreuses BP d'ordre général. Il est souvent nécessaire de raffiner les règles générales afin de pouvoir les exprimer et les formaliser en OCL sur le méta-modèle SPEM. En utilisant uniquement les constructions offertes par SPEM2.0 et le stéréotype *Iteration*, la pratique « Une itération est une activité de haut niveau, qui doit être répétée dans le temps, et qui se décompose en sous-activités » peut être exprimée en OCL de la façon suivante :

```
context: Iteration
inv: self.isRepeatable = true and
    self.nestedBreakdownElement->asSet()
    ->select( be:BreakdownElement | be.oclIsKindOf( Activity ) )
    ->asSet()->notEmpty()
```

Une autre pratique, spécifique à la société, consiste cette fois-ci à saisir à l'aide d'un outil dédié les temps passés par chaque développeur lors des différentes phases d'un projet, ou de toute autre activité de production (documentation, recherche, maintenance, etc.). Dans SPEM, il est possible de représenter cette pratique sous la forme d'une tâche intitulée « Saisie hebdomadaire des temps passés », et d'une contrainte OCL associée. Pour cela, le stéréotype "SaisieDesTemps" qui est de type *TaskUse* est défini dans un Profil SPEM pour RUP spécifique à l'entreprise, et la BP peut être modélisée comme suit :

```
context: SaisieDesTemps
inv: self.isPlanned = true and
    self.planningData.oclIsTypeOf( WeeklyPlannedData ) and
    self.planningData.oclAsType( WeeklyPlannedData ).dayOfWeek=1
```

## 5. Conclusion

Les bonnes pratiques intégrant souvent des aspects process, nous avons évalué l'intérêt du langage SPEM pour servir de base à la formalisation de BP orientées processus. Notre étude a montré que SPEM se révèle compliqué à appréhender par les développeurs. Il se montre également souvent trop abstrait pour décrire certaines

BP. Leur expression suppose l'existence de profils dont la définition reste à la charge de l'entreprise. Les profils spécialisent les concepts manipulés ; des contraintes OCL permettent alors la formulation d'une expression, sur la base de ces concepts.

Passées ces quelques difficultés SPEM se montre à l'aise dans son domaine de prédilection : la modélisation des aspects processus apparaissant dans les BP. Les problèmes se posent de manière insoluble lorsqu'il faut tisser de manière cohérente des aspects processus et modèles inhérents à certaines pratiques. Cette expérimentation a également mis en valeur le besoin d'organiser les BP selon des structures multiples de type hiérarchique voire de type graphe : selon leur granularité (certaines se construisant comme agrégation d'autres) et selon leur niveau d'application (générales à l'entreprise, propres à un projet, à un domaine, etc.), selon leur niveau d'abstraction dans la chaîne IDM (PIM, PSM, avec éventuellement des stades intermédiaires). Cette structuration et les aspects de tissage entre modèles et processus seront au coeur de nos préoccupations pour l'élaboration d'un langage basé sur SPEM dédié à la formalisation des bonnes pratiques. Sur le second point nous pensons nous inspirer des travaux qui abordent l'ingénierie des méthodes de Génie Logiciel (Henderson-Sellers, 2003), (Gonzalez-Perez *et al.*, 2007).

## 6. Bibliographie

- Fragidis G., Tarabanis K., « From Repositories of Best Practices to Networks of Best Practices », *Management of Innovation and Technology, 2006 IEEE International Conference on*, vol. 1, p. 370-374, June, 2006.
- Gonzalez-Perez C., Henderson-Sellers B., « Modelling software development methodologies : A conceptual foundation », *J. Syst. Softw.*, vol. 80, n° 11, p. 1778-1796, 2007.
- Gratton L., Ghoshal S., « Beyond Best Practices », *Sloan Management Review*, vol. 46, n° 3, p. 49-57, 2005.
- Henderson-Sellers B., « Method engineering for OO systems development », *Commun. ACM*, vol. 46, n° 10, p. 73-78, 2003.
- McConnell S., *Code complete : a practical handbook of software construction*, Microsoft Press, Bellevue, WA, USA, 1993.
- OMG, « Software and Systems Process Engineering (SPEM), formal/2008-04-01 », <http://www.omg.org/spec/SPEM/2.0/>, 2008.
- Phillips D., *The Software Project Manager's Handbook : Principles That Work at Work*, John Wiley & Sons, 2004.
- Shull F., Turner R., « An empirical approach to best practice identification and selection : the US Department of Defense acquisition best practices clearinghouse », *Empirical Software Engineering, 2005. 2005 International Symposium on*, vol. , p. 8 pp.-, Nov., 2005.
- Thayer R. H., *Project Manager's Guide to Software Engineering's Best Practices*, IEEE Computer Society Press, Los Alamitos, CA, USA, 2002.
- Zhu L., Staples M., Gorton I., « An Infrastructure for Indexing and Organizing Best Practices », *REBSE '07 : Proceedings of the Second International Workshop on Realising Evidence-Based Software Engineering*, IEEE Computer Society, Washington, DC, USA, p. 4, 2007.