RUPをパターン言語として捉える

- 開発の「苦」から解放するベスト・プラクティス

河 合 昭 男†

大規模開発プロセス RUP や短期小規模開発プロセス XP などのアジャイル系は，過去の様々な人達の経験や試行錯誤を通して得られた知見を集大成したベスト・プラクティスがベースになっている．ベスト・プラクティスの表現形式としてパターン言語はひとつの優れた方法である．今回 RUP に焦点を当て，パターン言語化を試みた．開発の「苦」から解放するベスト・プラクティスを，パターン言語の技法でさらに洗練化して行きたい．

RUP as a Pattern Language

- Best Practices to Relieve “Pain”of Development

Akio Kawai†

RUP for large systems and XP for small systems are based on best practices. A pattern language is a suitable technique to document best practices. This paper intends to explain RUP best practices as a pattern language. Let’s progress a pattern language based on the best practices which relieve “pain” of development.

1. はじめに

パターンとは繰返し起きる問題に対する解決策である．特定の問題領域のパターンを集めたものがパターン言語である．

ベスト・プラクティスとは，過去からの様々な人達による経験や試行錯誤を通してそこから得られた知見を集大成したものである．つまりひとつ一つのベスト・プラクティスはいわばパターンであり，特定の問題領域のベスト・プラクティス集はパターン言語として捉えることができる．

システム開発プロセスという問題領域で繰り返し起きるいつもの問題を解決する方法として，主に大規模開発を対象とした RUP や短期小規模開発を対象とした XP などのアジャイル開発プロセスがあるが，これらもベスト・プラクティスがベースとなっているのでパターン言語化できる．

今回 RUP に焦点をあててパターン言語化を試みる．前半で RUP2003 を，後半で大きく改訂された RUP7[1] について考察する．

†㈲オブジェクトデザイン研究所
Object Design Laboratory，Inc.

2. RUP の 6 つのベスト・プラクティス

RUP2003 は次の 6 つのベスト・プラクティスがベースになっている．それらは過去大規模開発に携わった人達の様々な経験や試行錯誤から得られた知見を集大成したものである．

---

(1)反復型開発
(2)要求管理
(3)コンポーネント・アーキテクチャ
(4)UML による可視化
(5)継続的品質検証
(6)変更管理

これら 6 項目はそれぞれ開発上の問題とそれを解決する方策を示しており，パターンである．これらのパターン言語化を試みた[2]が本稿では頁の都合で割愛する．

3. RUP7

RUP7 でこの 6 つのベスト・プラクティスは大きく見直され，6 つの基本原則(principles)として再構築された．ちなみに XP は「価値(values)―基本原則(principles)―実践(practices)」という構成になっているが，RUP7 の構

成も「基本原則(principles)―実践(practices)」となった．さらに，従来の RUP は開発者側を主体とした開発プロセスという位置づけであったが，改訂版では顧客満足が得られる製品開発のためのプロセス-顧客志向が強く打ち出されている．次にこの 6 つの基本原則を示す．

Adapt the process.（プロセスの適合）
Balance competing stakeholder priorities.（優先順位の調整）
Collaborate across teams.（チーム横断の協力）
Demonstrate value iteratively.（反復的に価値を実証）
Elevate the level of abstraction.（抽象化のレベルアップ）
Focus continuously on quality.（継続的品質注視）

個々の基本原則の記述形式は，(1)Benefit で利点を述べ，(2)Patterns には従来のベスト・プラクティスが含まれている．その関係を図 1 に示す．

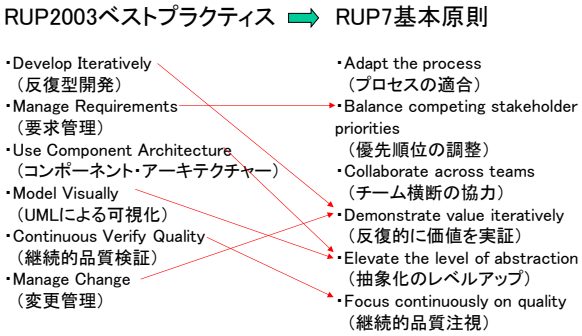

図 1　基本原則に吸収されたベスト・プラクティス

(3)Anti-pattern で起こりがちな失敗パターンが記述され，よく知られているアンチパターン[4] もここに含まれる．このようにパターンを意識した記述に改訂されていることが注目される．これらについてもパターン言語化を試みた[3]が本稿では頁の都合で割愛する．

4. おわりに－四諦八正道と RUP

釈迦が悟りを開き最初に弟子に説かれた法は四諦（したい）すなわち苦集滅道（くじゅうめつどう）だと言われている．

---

①苦：人生は苦である．
②集：では苦の原因は何なのか分析しよう．
③滅：原因が分かったらそれを取り除こう．
④道：それを取り除く方法論は八正道（はっしょうどう）である．

なぜこのような関係のない話を突然始めたのかいぶかられるでしょうが，筆者にはこれはパターン言語だと閃いた．正に人生で繰り返し起きる問題に対する解決策がここに解かれている．問題（苦）の原因を分析し（集），その原因を取り除く（滅）解決策（道＝八正道），つまり人生の問題の救済方法を「ソフトウェア危篤患者の救出[4]」と重ね合わせてみた．ソフトウェア開発上繰り返して失敗に陥るいつもの悪いパターン（アンチパターン）から抜け出すためのひとつの処方箋が RUP や XP という名前のパターン言語だと考えることができる．

開発の「苦」から解放するベスト・プラクティスを，パターン言語の技法でさらに洗練化して行きたい．

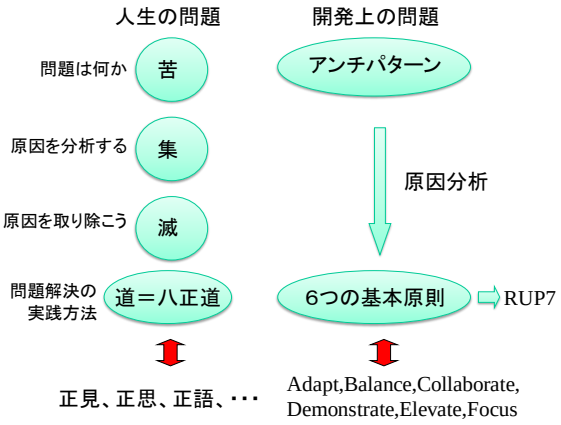

図 2　四諦八正道と RUP

参考文献

1）P.Kroll，W.Royce，“Key principles for business - driven development”，2005，http://www-128.ibm.com/developerworks/rational/library/oct05/kroll/
2）河合昭男，”RUP をパターン言語として考える”，2007，http://www.atmarkit.co.jp/im/carc/serial/world19/world19.html
3）河合昭男，” RUP7 で開発の「苦」から解放される”，2007，http://www.atmarkit.co.jp/im/carc/serial/world20/world20.html
4)W.J.Brown，岩谷宏[訳]，”アンチパターン”，ソフトバンク