

アジャイルソフトウェア開発と科学

鷲崎 弘宜[†]

アブストラクト

アジャイルソフトウェア開発方法論・プロセスを構成する種々のプラクティスに対する科学的アプローチの現状を紹介し、将来の展望を議論する。

Agile Software Development and Science

Hironori Washizaki[†]

Abstract

This paper reports the recent progress and future prospective of scientific researches on various practices in agile software development methodologies and processes.

1. はじめに

XP や Scrum に代表されるアジャイルソフトウェア開発方法論・プロセスは、主に経験や実績に基づいて有用性や一般性が確認されたプラクティスと称される行動指針のまとまりとして定義される。プラクティスの幾つかはもともと、文脈や制約に基づいて適宜選択・拡張されるようなプロセスやマネジメントに関する既存のパターン（パターンランゲージ）に基づくものである。また、パターンランゲージはもともと利用者を含む参加型の発展のプロセスを重視している 1)。

そのような背景に基づき、効率的かつ効果的なアジャイル開発の実施には、それぞれの組織やドメインに応じたプラクティスの取捨選択、拡張、新規策定、さらには全員参加型の継続的な見直しが求められる。

取捨選択や実践にあたり、個々のプラクティスが有効に機能する前提条件、および、その条件下でプラクティスを適用する際の最適なパラメータや期待される効果が明らかであることが望ましい。しかしながら、それらが客観性のある根拠を持って解明済みのプラクティスは少数であり、しばしば研究者からの「アジャイルには科学が（足り）ない」との批判を耳にする。

このような状況を受けて近年、個々のプラクティスの妥当性や有用性を検証する科学的アプローチが取り組まれ始めている。本稿では、我々の研究成果 2) を含めて、近年のアジャイル開発に対する科学的アプローチの幾つかを紹介し、今後の展望や期待を述べる。

2. アジャイルへの科学的アプローチ

「科学」の定義は様々であり文脈にも依存するが、

ここでは例えば、特定の現象に関する仮説（例えばその現象の原因や振る舞いを定性的・定量的に説明するモデル）について客観性のある実験や観察および適切な推論過程を経て検証することとする。「工学」とは、科学を応用した実用的技術の開発ということができる。

科学における推論過程はしばしば、帰納と演繹に大別される。ソフトウェアの場合、開発や運用の実績データから原理原則を見出す帰納的アプローチと、実証済みの原理原則および特定の仮定のもとで作成したモデルにより開発や運用を予測あるいは説明する演繹的アプローチの二つを考えることができる。

これらの二つに大別した形で、アジャイル開発を構成する既知のプラクティスや概念・定石への科学的アプローチをプラクティス毎に以下に紹介する。

(1) 帰納的アプローチ

- ペアプログラミング: Cockburn らは幾つかの実証的調査を通じて、ペアプログラミングが品質向上やコミュニケーションに有用なことを示している 3)。Salleh らは、(データによって裏打ちされた) 文献群の調査を通じてペアプログラミングの教育上の効果を分析している 4)。
- テストファースト (テスト駆動開発): テストファーストを取り入れた場合により多くのテストを記述し結果として生産性が高いことが学生実験の結果として報告されている 5)。
- トラックファクタ: Ricca らは OSS 開発のソースコードリポジトリにおけるコード作成・改変履歴を分析し、「トラックにひかれてプロジェクトの進行が困難となる人数」の導出を試み、その難しさを議論している 6)。ただし、各 OSS 開発プロジェクトにおける実際のトラックファクタを測定・観測したうえで上記の導出結果と照

[†]早稲田大学 Waseda University / 国立情報学研究所
GRACE センター NII GRACE Center

らしているわけではないため、厳密には帰納的
と言えない面もある。

(2) 演繹的アプローチ

- イテレーション: 我々は、イテレーションを繰り返して顧客からの定期的なフィードバックと段階的リリースを経る開発プロセスのモデル化の上で、モデル上でのシミュレーションを通じて、要求の変動性や規模に応じた最適なイテレーション期間を推測する手法を提案している 1)。
- 要求優先順位付け: Port らは、開発における要求の追加や得られる価値をモデル化し、シミュレーションを通じて、開発途中における要求発生確率が高い場合に、当初の計画ベースに開発を進めるよりも、イテレーション単位で価値に基づき要求を優先順位づけることで最終的に得られる価値が大きくなることを示している 7)。
- ペアプログラミング、リファクタリング: Cao らは、アジャイル開発の事例や、文献によるプラクティス等の特徴を抽出しシステムダイナミクスモデルを作成したうえで、モデル上でのシミュレーションを通じて、ペアプログラミングおよびリファクタリングがコストや生産性の観点から有用なことを示している 8)。

3. 展望と期待

上述のように、幾つかのプラクティス単体について帰納的・演繹的の両アプローチにより有効性や前提条件が検証されつつある。しかしながら、検証されつつあるプラクティスは依然として限られており、今後は以下のような取り組みが期待される。

- モデル化と演繹的アプローチの充実: (自然なことであるが) 実績データの観測を通じた帰納的アプローチが起り、続いて、そこで見出された原理原則を用いてモデル化して演繹的アプローチが取り込まれる、との流れが見受けられる。これに沿って、今後はアジャイル開発プロセスのダイナミクスを説明するモデル化の議論が活発となることが期待される。モデル化にあたり特に、プロジェクトの文脈・環境や特定個人に依存する部分や影響に留意が必要である。
- 人間系・組織系の扱い: 例えば「チーム全体」、「オンサイト顧客」のように、アジャイル開発では利害関係者間の信頼関係や対話、距離の近さ、全員参加型の取り組みを重視する。このような人間関係・チーム・組織系のプラクティスへの科学的アプローチが求められる。

- プラクティスの組み合わせ検証: プラクティス間における依存関係や相互作用が経験的に知られており、その科学的解明が求められる。
- 科学から工学へ: これまでの取り組みは仮説検証にとどまることが多く、それに基づいたアジャイル開発の実施前あるいは実施中における具体的な活動に応用可能な技術やツールとしてのまとめあげと活用が求められる。

以上の推進に向けて、アジャイル開発におけるプロセス・プロダクトデータの蓄積が求められる。特に、アジャイル開発の特徴である利害関係者間の対話や参加に関するデータが含まれることが望ましい。これらは帰納的アプローチの直接の入力となるばかりでなく、演繹的アプローチの結果の妥当性の検証にも必要である。ただしアジャイル開発においては、計画やドキュメントを重視する開発と比べると、開発途中の活動や成果の記録を詳細かつ電子的に把握困難な可能性があり、その記録と参照を意図した取り組みが求められる。さらに、こういったワークショップ等を通じて、モデルに対する実務家を含む集団によるレビューや比較・議論の機会設置が進むことを期待したい。

参考文献

- 1) 鷲崎, “ソフトウェアパターン概観”, 情報処理, 52(9), 2011.
- 2) 塩浜, 坂本, 久保秋, 鷲崎, 深澤, “アジャイル開発における適切なイテレーション期間のシミュレーションによる推定”, SES2011.
- 3) A. Cockburn and L. Williams, “The Costs and Benefits of Pair Programming,” 2nd International Conference on Extreme Programming and Flexible Processes in Software Engineering, 2001.
- 4) Norsaremah Salleh, et al., “Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review,” IEEE Transactions on Software Engineering, 37(4), 2011.
- 5) Hakan Erdogmus and Torchiano Morisio, “On the Effectiveness of Test-first Approach to Programming,” IEEE Transactions on Software Engineering, 31(1), 2005.
- 6) Filippo Ricca, et al., “On the difficulty of computing the Truck Factor,” 12th International Conference on Product-Focused Software Process Improvement, 2011.
- 7) Dan Port and Alexy Olkov, “Using Simulation to Investigate Requirements Prioritization Strategies,” 23rd IEEE/ACM International Conference on Automated Software Engineering, 2008.
- 8) Lan Cao, et al., “Modeling dynamics in agile software development,” ACM Transactions on Management Information Systems, 1(1), 2010.