



未来に向かって

アジリティを追求した ソフトウェア開発



鷲崎弘宜 (早稲田大学／国立情報学研究所／(株)システム情報)

アジリティが重要な時代

ソフトウェア、社会、人々が密接にかかわり、不確実性を増しつつある今日、市場や顧客の反応を素早く得て、要求や環境の変化に適応可能な俊敏さ（アジリティ、Agility）がソフトウェア開発に必要である。本稿では、アジリティを追及する「アジャイルソフトウェア開発」（Agile Software Development）について、よくある誤解も含めてさまざまな捉え方や起源、再定義の動きを紹介する。そのうえで研究との関係として、アジャイル開発に対する研究（Research for Agility）と、研究をアジャイルに進める取り組み（Agility for Research）の両面を取り上げて、最後に将来を展望する。

つまりアジャイル開発とは

ソフトウェア開発においてこれだけ普及し、これだけ誤解されている概念はほかにはないだろう。海外におけるアジャイル開発の導入は9割以上の開発組織に及び、国内においても約6割は導入済みというデータがある^{☆1}。この海外と国内の差の背景としては、Hofstede Indexに見て取れるように日本の組織が保守的で不確実性を忌避する傾向にあることや、雇用の流動性が低く個々人のプロフェッショナルの意識が育ちにくいことなどが挙げられる。また、誤解が蔓延していることも要因の1つであろう。

そのよくある誤解は、「アジャイルではドキュメントはいらない」「行き当たりばったりで各自が好きなように作ることがアジャイル」「特定の手法こそがアジャイル」といったものである。これらはすべて間違っている。アジャイル開発とは、特定の何かをする（Do Agile）ことではな

く、アジリティのマインドを持つこと（Be Agile）である。

そのマインドは、「軽量な」開発プロセスの関係者らが2001年に合意したアジャイルソフトウェア開発宣言^{☆2}にある。具体的には以下の4点に価値をおく姿勢である。

- （プロセスやツールよりも）人間と人間関係
- （ドキュメントよりも）動くソフトウェア
- （契約交渉よりも）顧客との協力
- （計画に従うことよりも）変化への対応

アジャイル開発の手法と起源

eXtreme Programming (XP) や Scrum に代表されるアジャイル開発手法やプロセスは、アジャイルマインドに基づきアジリティを発揮させる有効な手段である。おおむね共通のプロセスモデルを図-1に示す。XPが開発者視点、Scrumがマネージャ視点であり実践が推奨される事柄（プラクティス）に違いはあるものの、顧客参加を促しながら開発期間を短い単位（イテレーション）に区切って反復的に進めることで迅速かつ探索的・適応的に開発し顧客満足度を高める点は共通している。

それらの起源は、Christopher Alexander のパターンランゲージと、Edwards Deming らによる生産・品質管理の諸理論に遡ることができる。

前者は、心地よい都市環境や建築に共通の形や形成プロセスを、関係づけられたパターン群として言語化する考え方である。ソフトウェア開発の組織やプロセスに持ち込まれた結果、90年代の「段階的实施」「ペア開発」といったパターンをまとめた Coplien の『開発工程の生成的パターンランゲージ』や、「作業分割」といったパターンをまとめた Cunnigham の『エピソード』などにアジャイル開発プロセスの源流を見ることができる。

^{☆1} http://sec.ipa.go.jp/users/seminar/seminar_tokyo_20150708-01.pdf

^{☆2} <http://agilemanifesto.org/>

後者における最も著名な理論はPDCA/PDSAサイクルであり、それを実直に用いた歴史上初(1976年発表)のアジャイル開発手法としてEvoが知られている。さらにDemingの考え方をベースとして製造業で成功をおさめた考え方にトヨタ生産方式・リーン思考がある。これは顧客への価値を生み出す各工程における無駄や遅れを取り除き、価値を最大化する

流れ(バリューストリーム)を作り管理する考え方であり、Scrumやリーンソフトウェア開発の源流となっている。

論文における名称としての「アジャイルソフトウェアプロセス」の初出は、富士通(株)における開発とプロセス進化をまとめた青山教授のICSE'98におけるものである。

「すること」や手法の制度化(たとえばScrumマスターの認定制度)によりアジャイル開発は今日の隆盛を見ているが、結果として皮肉にもマインドが見失われがちである。これらの源流に触れることでソフトウェア開発におけるアジリティを再考することの意義は大きい。

アジャイル再定義

アジャイル宣言から時間が長く経過しツールやプロセスが成熟する中で、他分野の理論を応用してアジャイル開発の価値を再検証・再定義する動きが活発である。たとえば意思決定の状況を整理したCynefinフレームワーク²⁾(図-2)に基づくと、アジャイル開発における反復的・漸増的な開発とフィードバックは「複雑」な場合に適すると捉えられ、逆に言うところ「複合的」「単純」な状況下ではその必然性がない。しかし「複雑」な状況についても、プロセスそのものを反復的に改善し続ける仕組みが未確立であるため、アジャイル開発も含め実は十分ではないという指摘がある²⁾。

またJoshua Kerievskyは2016年にアジャイル宣言

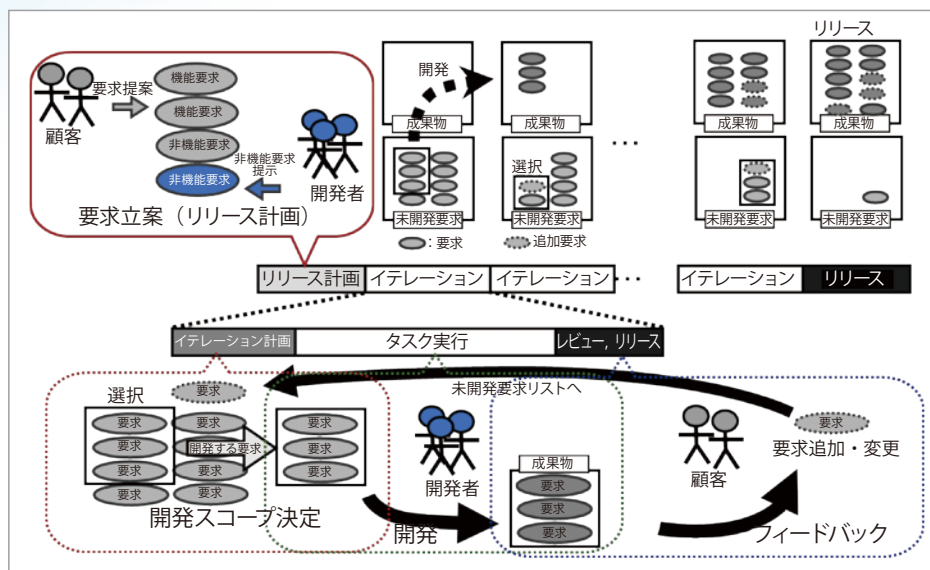


図-1 アジャイル開発プロセスの共通モデル(文献1)より)

を現代風に見直して「モダンアジャイル」と称し、集団心理学などを参考に以下の4点に価値を置くと再定義している^{☆3)}。

- 人々を尊重し輝かせること
- 心理的安全性を前提とすること
- 素早く実験し学習すること
- 価値を継続的に届けること

アジャイル開発のコミュニティにおいて、「アジャイル」そのものを絶えず見直しその時々状況に適應させようという動きは自然なものであり、それこそがアジャイル開発が今後も生き残り、主流であり続ける要因といえる。

アジャイル開発と研究

Research for Agilityとしては、前述のような理論的枠組みによる再考に加えて、アジリティを発揮するうえで有用であると経験的に知られるプラクティスに対してデータにより実証するものや³⁾、モデル化やシミュレーションさらには大規模サーベイを通じて個々のプラクティスおよびアジャイル開発全体の成功要因および留意点を明らかにするものがある。たとえば筆者らは、シミュレーションを通じて、要求の不確実性や関係の複雑さに応じて最適なイテレーション期間が異なることを明らかとした¹⁾。さらに、源流であるパターンやパ

☆3 <http://modernagile.org/>

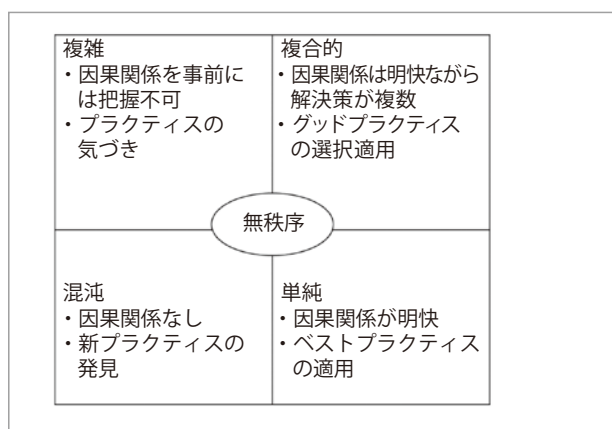


図-2 Cynefin フレームワーク (文献 2) より)

ターンランゲージに回帰して、プラクティスや関連する取り組みをパターン化する動きも活発である^{☆4}。たとえば筆者らは、アジャイル開発における品質保証の取り組みをパターン群として記述し「アジャイル品質」として提唱している⁴⁾。

Agility for Research については、ソフトウェア工学に限らず研究活動全般が本来的に探索的であることを考えると、それへのアジャイル開発の適用は自然なものといえる。事実、大学や企業においてソフトウェアやハードウェア等の研究活動へ多く適用され、コミュニケーションの効率化や、無数のアプローチの高速な適用・検証の繰り返しの効果が報告されている (たとえば文献 5))。

アジリティの行方

本稿ではソフトウェア開発におけるアジリティを、マイルドや手法、研究などのさまざまな角度から取り上げた。ソフトウェア開発においてアジャイル開発が定着し評価され再考されてきた今日、さらなる進展の方向性として適用対象や規模の拡大が挙げられる。

具体的には、前述のようにソフトウェアに限らず、ハードウェアを含むシステムの研究開発に対してアジャイル開発を適用するアジャイル・システムズエンジニアリング⁵⁾が1つの流れであり、今後の進展とシステムズエンジニアリング特有の課題識別や拡張が期待される。

もう1つは、組織全体とアジャイルの関係の進展であり、「エンタープライズ・アジャイル」と「アジャイル・

エンタープライズ」⁶⁾の2つがある。前者は、アジャイルソフトウェア開発のプラクティスの適用を組織レベルへと拡大する流れであり、Scaled Agile Framework や Disciplined Agile Delivery 等のフレームワークが用いられつつある。一方の后者は製造業を中心として、リーン思考やリーンマネジメントを起点に市場などの環境変化に適応的な組織レベルのアジリティを追求する1990年代から歴史ある流れである。両者は異なるコミュニティで研究実践され今日に至るが、変化への組織的な適応といった点は当然ながら共通である。リーンという共通の源流を持つこともあり、今後のコミュニティや領域を超えた議論と発展を期待したい。加えて、組織におけるプロセス改善の指針をまとめたCMMIについても、その共通プラクティスをアジャイル開発へ適用することに繋がる問いかけ^{☆5}が認知され、またCMMI Instituteから『A Guide to Scrum and CMMI』が2017年に発行されるなど、特に海外では融合が当たり前となり、日本でも進展が期待される。

参考文献

- 1) Shiohama, R., Washizaki, H. et al. : Investigating the Relationship between Project Constraints and Appropriate Iteration Length in Agile Development through Simulations, International Journal of Computer Applications in Technology, 54(4) (2016).
- 2) O'Connor, R. V. and Lepmets, M. : Exploring the Use of the Cynefin Framework to Inform Software Development Approach Decisions, International Conference on Software and System Process (ICSSP) (2015).
- 3) Dybå, T. and Dingsøyr, T. : Empirical Studies of Agile Software Development : A Systematic Review, Information and Software Technology, 50(9-10) (2008).
- 4) Yoder, J., Wirfs-Brock, R. and Washizaki, H. : QA to AQ Part Five : Being Agile at Quality, 6th Asian Conference on Pattern Languages of Programs (AsianPLoP) (2016).
- 5) Huang, P. M. : Agile Hardware and Software System Engineering for Innovation, IEEE Aerospace Conference (2012).
- 6) Ragin-Skorecka, K. : Agile Enterprise : A Human Factors Perspective, Human Factors and Ergonomics in Manufacturing & Service Industries, 26(1) (2016).

(2017年6月8日受付)

☆5 <http://xpjug.com/xp2016-session-c6-2/>

驚崎弘宜 (正会員) washizaki@waseda.jp

博士 (情報科学)。早稲田大学教授・グローバルソフトウェアエンジニアリング研究所所長、国立情報学研究所 客員教授、(株) システム情報 取締役 (監査等委員)。ISO/IEC/JTC1 SC7/WG20 Convenor, IEEE Computer Society Japan Chapter Vice-Chair, IEEE Computer Society Membership at Large for the Professional and Educational Activities Board, SEMAT Japan Chapter Chair, IPSJ Samurai Coding Director, CSEE&T'17 PC Co-Chair, APSEC'18 PC Co-Chair, Int. J. of Agile and Extreme Software Development Editor-in-Chief。本会シニア会員。

☆4 <http://www.scrumplop.org/>