

特集2 ● SCRUMによるプロジェクトマネジメント

Agile Project Management with SCRUM



第1章

SCRUMの誕生

「適材適所」のソフトウェア開発

注1) 参考文献は、第4章の文末に掲載しています。

注2) http://www.agilealliance.org/programs/roadmaps/Roadmap/xbreed/xbreed_index.htm

ソフトウェア開発、特に小〜中規模開発のプロジェクトマネージャは、なかなか思うようにいかないプロジェクト運営に悩まされています。見通しの立ちづらirik、いつまでも定まらない顧客からの要求、その後の度重なる変更、急速に進歩する技術への対応…プロジェクトの進展を妨げる要因はあまりにも多く、破綻するプロジェクトは後を絶ちません。プロジェクトマネージャは、こういった状況に対して何か打つ手はないのでしょうか？

上記のようなソフトウェア開発の特殊性を考慮していないプロセスに従って開発するのであれば、その通りかもしれません。しかし、こうした特殊性を前提にして考え出された新しいプロセスを導入すれば、今よりうまくプロジェクトを運営することも可能です。

本特集では、そのような新しいプロセスの中で最も人気があり、世界中で多くの実績をあげているSCRUM^{参考文献1)注1)}について紹介します。



本特集は、全部で4章の構成になっています。

第1章では、開発プロセスの歴史を簡単に紹介し、どのような問題意識からSCRUMが成立するに至ったかについて解説します。

第2章では、SCRUMのルールを具体的に解説します。プロジェクトへのSCRUM

の導入を検討する際には、第2章が役立つでしょう。

第3章では、SCRUMと他の開発プロセスを組み合わせたプロジェクトマネジメント方法について解説します。こうした組み合わせの中でも、特にXPはSCRUMと相性が良く、「XBreed」^{注2)}などの名称で、海外では広く利用されています。第3章では、こうした組み合わせによって相乗効果を最大限に引き出す効果的な導入法とともに、実際の開発プロジェクトにおける実績を紹介します。

第4章では、ソフトウェア開発・システム開発でなぜSCRUMが有効なのかを、SCRUMが理論的基盤としている理論や学説などを紹介しながら解説します。第4章は、SCRUMの理解を深めるだけでなく、ソフトウェア開発やプロジェクトマネジメントで有用な多くのヒントを提供します。



SCRUMの誕生



失敗と挫折の歴史

ソフトウェア開発をより効果的におこない、プロジェクトを成功させるために、戦略的に開発プロセスを定義し、導入している企業や組織は少なくありません。その中には、プロセスの導入によって大き

第1章 ● SCRUMの誕生



な成果をあげている企業もあれば、まったく期待された効果を得られない企業や組織もあります。前者の代表として真っ先に名前が挙がるのは航空宇宙や防衛産業などが多く、後者の代表として議論的になるのは情報システム開発に代表される商用アプリケーション開発でしょう。

なぜ、航空宇宙や防衛産業でうまく機能している開発プロセスが、システム開発の現場ではあまり効果的でないのでしょうか？

それは、両者の性格がまったく異なるからです。航空宇宙や防衛産業におけるソフトウェア開発には、長い経験と実績があり、比較的長期間にわたってきわめて大規模におこなわれるものです。これに対し、受託型の情報システム開発は、極端にビジネスの影響を受けやすいという特徴があり、顧客ごとに毎回異なるシステムを、短期間に小人数で開発するのが一般です。つまり、両者は「ソフトウェア開発」という点こそ共通するものの、実際には目的も規模も内容もまったく異なる開発ということになります。

このようにまったく異なる種類の開発について、条件などを明確に整理・分析せず、単純にプロセスを導入したり議論したりすることには、もともと無理がありました。そのため、大規模向けプロセスを導入した小～中規模開発プロジェクトの多くは失敗に終わることになったのです。

アジャイルな開発の台頭

「開発の種類が異なるのであれば、それに最適なプロセスも異なるのではないか」という問いかけは、きわめて自然なものです。数多くの失敗プロジェクトが繰り返



図1 アジャイルアライアンスのWebサイト (<http://www.agilealliance.org/home>) アジャイルな開発に関するさまざまな情報が提供されている。

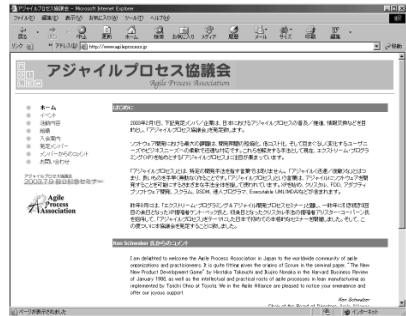


図2 アジャイルプロセス協議会のWebサイト (<http://www.agileprocess.jp>) 日本国内におけるアジャイルへの機運の高まりを受けて、2003年2月に発足した団体のWebサイト。

返される中で、現場の最前線に立つソフトウェア開発者たちがこのような疑問を持つようになったのも、特に驚くにはあたりません。そして、こうした考え方に基いて提唱されるようになったのが、「アジャイル」と呼ばれるソフトウェア開発スタイルであり、それに対応したプロセスです(図1、図2)。

「大規模な航空宇宙や防衛産業向けのプロセスを、小規模なシステム開発に無理に適用するよりは、小規模な開発に最適なプロセスを使おう」というアジャイルの提唱者たちの主張は、とてもシンプルです。それは、あらゆる方法には限界があ

注3) このあたりの考え方は、『アジャイルソフトウェア開発』に詳しく紹介されています(参考文献?)

特集2 ● SCRUMによるプロジェクトマネジメント

Agile Project Management with SCRUM

アジャイルソフトウェア開発宣言

我々は、自らアジャイル開発を実践するとともに、
人々がアジャイル開発を実践するための支援を通じて、
より優れたソフトウェア開発方法を見つけようとしている。

この活動を通じて、我々は

人と人同士の相互作用を、プロセスやツールよりも
動くソフトウェアを、包括的なドキュメントよりも
顧客との協力を、契約交渉よりも
変化に対応することを、計画に従うことよりも

尊重するに至った。

これは、右側にある項目の価値を認めつつも、
左側にある項目の価値をより一層重視する、ということである。

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Ker
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

(c) 2001, 上記の著者たち
この宣言は、この注意を含めた全文である場合に限り、
どのような形でも自由に複製してよい。

図3 アジャイルソフトウェア開発宣言（原文は<http://www.agilemanifesto.org>）
著者の中に、SCRUMの提唱者であるKen SchwaberやJeff Sutherlandの名前が見られる。

ることを認めたくて、規模や状況に応じて最適な方法を使っていこう^{注3}という価値観に裏付けられています。つまり、「大は小を兼ねる」よりも、「適材適所」を重視するという価値観です。

さらに、アジャイルなソフトウェア開発には、「小規模だからこそ可能になるメリットを最大限活用していこう」という思想も取り入れられています。小規模であることの最大のメリットは、**変化に対応しつつ俊敏（アジャイル）に動き回れる**ことです。この俊敏性を最大限に活かし、

刻々と変化する顧客ニーズに対応し、そして、それによって利益をあげることが、アジャイルなソフトウェア開発の目指すものなのです。このようなアジャイルの基本理念を示すマニフェストが、アジャイルアライアンスのWebサイトに掲載されています（図3）。

SCRUMは、アジャイルな開発を代表するプロセスの1つで、5～9人規模のチームで最大のマネジメント効果を発揮すると言われています^{参考文献21}。



カオスへの挑戦



明らかになった問題点

プロセスに関する研究が進むにつれ、航空宇宙分野には機能するプロセスが、ビジネス分野によく見られる小～中規模のソフトウェア開発で機能しない理論的根拠が、次第に明らかになってきました。それは、プロセスの成立経緯に端を発するものでした。

●従来型プロセスの特徴

以前から多くの開発プロジェクトで利用され、現在までのところ主流となっているソフトウェア開発プロセスやプロジェクト管理プロセスは、「予見型プロセス」や「最適化型プロセス」はなどと呼ばれています^{参考文献3)}。

予見型プロセスや最適化型プロセスは、「開発過程はすべて事前に予見可能であり、作業を綿密に計画すべし」「開発にあたり手順や成果物を事前に明確にし、作業を最適化しよう」とするプロセスの特徴から由来する呼び方です。一般に、「予見型プロセス」や「最適化型プロセス」には、「目標としている結果を得るためには、作業を計画し、管理が重要であり、必要である。そして、プロセスは管理可能であり、結果、プロセスから生成される成果や成果物に対して効果的である」というプロセスの哲学が存在します。

このことは、プロセスの内部（作業、成果物、作業間のフロー、成果物間の依存関係）を詳細に定義し、管理することを意図しています。これは、担当者（エンジニア、管理者など）にはあらかじめ

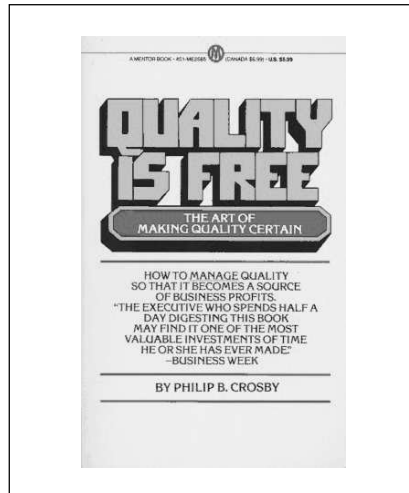


図4 Crosbyによる『Quality is Free』

決められた作業フロー、成果物を計画に沿って実践し、成果を出す任務が課せられていることを意味します。

●「生産」と「開発」

予見型プロセスや最適化型プロセスは、この20年来ソフトウェア開発プロセスの標準となってきましたが、こうしたプロセスは、実は「製造業」のプロセスを起源としています。それは、Taylorによる論文（1911年、1916年）をきっかけに、製造業で導入されたプロセスです。

その後Crosbyにより、製造業の生産性効率向上のためのMMM（Manufacturing Maturity Model）が、その著書『Quality Is Free』（図4）^{参考文献3)}で発表され、製造業界で成功を収めてきました。これをソフトウェア開発に導入・応用したものが、今日広く利用されているプロセスの基盤となっています。それは、ソフトウェア開発においても、製造業同様の効果を期待してのことです。

しかし、こうしたプロセスを適用した

特集2 ● SCRUMによるプロジェクトマネジメント

Agile Project Management with SCRUM

注4) 公平と正確を期すために、予見型プロセスや最適化型プロセスがソフトウェア開発で“常に”不適なわけではないことを改めて指摘しておきたいと思えます。事実、航空宇宙や防衛産業などは、こうしたプロセスによって大きな効果をあげています(参考文献:「ソフト開発プロセス改善手法と事例研究 品質や生産性を劇的に向上」日経コンピュータ1997.3.17号,「ソフト開発のプロセス導入効果と実践方法 米モトローラの事例に学ぶ」日経コンピュータ1997.12.22号)。

確かに、航空宇宙や防衛産業のソフトウェア開発も「開発」ではあります。しかし、これらの産業は競合企業が少なく、かつミッションクリティカルな分野なので、比較的時間をかけてじっくり開発する上に、新規性が他の分野に比べて少ないという特徴があります。したがって、「開発」といってもむしろその実態は「生産」に近く、そのために製造業の生産プロセスに近い開発プロセスが非常に向いているのです。

また、SEIのCMMIやISO-15504などは、開発者、プロジェクト管理者がおこなう作業としてどのようなことがあるかを知るには有用です。要は、自分たちのプロジェクトで開発するシステムの特徴や置かれている状況に適したプロセスを研究し、導入していくという「適材適所」の視点が重要ということです。

多くのプロジェクトが、大きく開発予算をオーバーしたりキャンセルされたりして、期待されたビジネス価値を達成できていません。いくつかの研究の結果、全プロジェクトのおよそ3分の2が、何かの形で見積りを越えるということが明らかになっています。

その理由は、一言で言えば簡単です。製造業のプロセスは「生産」のプロセスであるのに対し、ソフトウェア開発は文字通り「開発」だからです。

Taylorが発表した「科学的管理法」は、今から100年ほど前、20世紀前半の製造業界において、従業員をいかに管理し、効率を向上させることができるかという課題に対応したプロセスでした。当時、こうした業種の従業員が作業する内容は単純な作業の繰り返しであり、従業員に独自の創造性は原則として必要とされていませんでした。CrosbyによるMMMも、工場と同じ作業を繰り返す生産工程を対象に、その管理作業をいかに効率化するか、また不良品をいかに出さないかに焦点を当てている「生産プロセス」です。

これに対し、現代のソフトウェア開発の多くはまったく事情が異なります。ソフトウェア開発者には、可能な限りの創造性が要求されます。プロジェクトでは、システムの要求から定義し始め、新しい技術によって実装される新製品を開発します。ソフトウェア開発は、その名の通り「開発」の作業であって、同じ製品を大量に「生産」するわけではありません。もうおわかりの通り、「生産」のプロセスである製造業界のプロセスを、「開発」であるソフトウェア開発に持ち込んだことが、大きなミスマッチとなっていることは

否めないのです。



そしてSCRUMへ

このようなプロセスをめぐる問題について、参考文献[1]には象徴的な記述があります。SCRUMの著者たちは、かつてソフトウェア開発プロジェクトで、「予見的な(定義された)プロセス」を利用したもののうまくいかなかったという経験を度々していました。そんなあるとき、あるプロジェクトの業務で、コンサルテーションを依頼した“プロセスの専門家”から、

『ソフトウェア開発は、ある種のカオスであるから、あらかじめ「定義されたプロセス」は不適切である』

と指摘されたことが、SCRUMを編み出す大きなきっかけになったそうです⁴⁾。

ここでの「定義されたプロセス」とは、予見型や最適化型プロセスのことを指しています。そして、「開発」は入力に対して出力がいつも同じとは限らないから、「生産」とは異なるという説明を受けています。さらに、高度に知的かつ創造的な作業では「予見的プロセス」の採用はおしる品質を低下させることや、ソフトウェア開発は多くのコミュニケーションを必要とする複雑な作業であることも指摘されています。



SCRUMの効果



数百%の生産性向上

参考文献[1]には、「SCRUMの効果は

第1章●SCRUMの誕生



『プロセス改善によって5〜25%程度生産性が向上した』などという中途半端なものではない。SCRUMの提供する生産性向上とは、桁違いの向上、つまり数百%の生産性向上ということの意味している」という驚くべき主張が見られます。これは逆説的には、従来型のプロセスを小〜中規模開発に適用することが、それだけ効率の悪いものであったということを示しているのかもしれませんが。

優れた適応可能性

SCRUMは、プロジェクトにとってしばしば悩みの種となっている「根本的な変更への対応」が可能であることを約束しています。また、SCRUMはプロジェクトに関わる者すべてに充実したプロジェクトマネジメントを実現することも約束します。SCRUMでは、あらゆる事実を、早期にかつプロジェクトに携わるすべての人に対して明確にし、可能な限り早く調整することによって、リスクと不確実性を減少させ、プロジェクトを成功させることができますとされています。

さらに、SCRUMの導入および適用分野については、「SCRUMは現在までに、金融、貿易、銀行、通信、福利厚生管理、医療、保険、電子商取引、製造、科学的環境に至る広範な専門分野におけるソフトウェア開発への利用で何十億ドルも生み出してきた」と述べられています。

したがって、SCRUMは分野を選ばずさまざまな種類のソフトウェア開発に利用できると考えてよいでしょう。

記憶喪失からの復活

SCRUMには、驚くような方法や高度な手法は登場しません。むしろ、そのあまりのシンプルさに、驚くべき方法が何もないという事実が驚くかもしれません。この点は、従来型プロセスがプロジェクトメンバーに非常に多くのタスクを要求していたこととは対照的です。

SCRUMのシンプルさは、ソフトウェア開発の現場において長年忘れ去られていた、ある重要な事実を思い出させてくれます。それは、ソフトウェア開発とは決して「決まった作業をただ繰り返す」ものではなく、状況に応じて適切な行動や作業をおこなうものである、という事実です。

ソフトウェア開発プロジェクトのマネジメントが、「自然で、単純で、常識的」ということの重要性が、いつの間にかソフトウェア業界で忘れられたことで、業界は高い代償を支払っています。SCRUMの提唱者たちは、これを「ソフトウェア業界の記憶喪失」と述べています。そして、「SCRUMはこの記憶喪失から、読者を呼び戻すことだろう」と記述しており、SCRUMの利用を強く推奨しています。■