

成功のカギはノウハウと人¹

阪井 誠

関西事業部

はじめに

開発標準で定められたプロセスモデルに従っていても、プロジェクトの業務や人によってうまくいく場合とそうでない場合があります。この記事では、プロセスモデルを補完してソフトウェア開発プロセス（以下プロセス）を成長させる方法について議論します。プロセスはプロセスモデルだけでなく、実践するノウハウや人が重要です。モデルに不足するノウハウを議論することで、プロセスに追加すべきものが明確になり、プロセス改善に役立てることができるでしょう。この記事を引きかけにして、プロセスをどのように改善し、成長させていくかといった前向きでオープンな議論を期待しています。

モデル・ノウハウ・人

良いプロダクトの開発には良いプロセスモデルだけでなく、実践に必要なノウハウや人が必要です。これまでプロセスモデルやマインドセットの議論が多く行われてきましたが、これらはプロセスの理解に役立つものの、より良いプロセスに成長させるには不十分でした。ソフトウェアシンポジウムの過去の議論では、プロセスは文化だと言われてきました。組織内のノウハウや人を無視して、プロセスを単純化したモデルや、その実現に必要なマインドセットを議論してもプロセス改善にはつながらないでしょう。

¹ この記事はソフトウェア技術者協会主催ソフトウェアシンポジウム 2022 における Future Presentation 「プロセスモデルの補完方法 -モデル・ノウハウ・人-」 を加筆修正したものです

アジャイル開発を例に挙げると「ウォーターフォールでもしっかり開発出来ないのにアジャイル開発なんて出来るわけがない」[1]という議論があります。アジャイル開発に対する理解不足の一面もありますが、ソフトウェア開発に共通するプロセスモデル以外の要素の存在を示しています。基本的な技術や業務知識など、開発で考慮すべき点はモデルに関係なく共通点多いでしょう。アジャイル開発を始める際などに「守破離」という言葉を使って厳格に実践することが求められる場合もありますが、本当にそれだけで良いのでしょうか？それが本記事を書いたきっかけです。

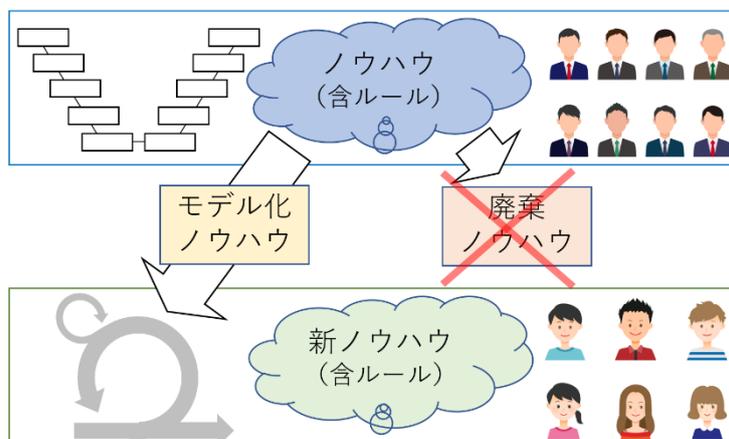
プロセスを成長させるにはモデルだけでなく、多くのノウハウや経験に基づく様々な工夫が必要です。チーム内だけでなく社内外の情報を利用して成長させる必要があります。

新しいプロセスへの移行の問題

図1に新しいプロセスへの移行を示します。従来のプロセスモデルで開発を行う中で、モデルにうまく適応するための工夫が行われていました。その工夫の中で、追加のルールや手順、注意事項など様々なノウハウが生まれ、人と組織は成長してきました。しかし、モデルに問題がある場合や、ノウハウの実践が簡単ではない場合など、何らかの問題があると、問題の解決を期待できる新しいモデルと新しいノウハウに移行されます。

移行時に従来のノウハウには新しいモデルや新しいノウハウに取り入れられるものもありますが、廃棄されるノウハウもあります。また、廃棄されたノウハウを知っている経験者が新しいプロセスに参画できない場合もあります。こうして古いプロセスモデルと共に従来の文化が失われてしまいます。

新しいプロセスモデルに移行しても文化はすぐには成熟しません。開発チームなどの学びはありますが、過去を知らない人への入れ替えによって文化の断絶が発生する一方で、プラセボ（偽薬）効果によって新しいものは良いものであるとの思い込みが生まれがちです。こうして新しいモデルの守りに入ってしまう、失敗が顕在化するまで改善が進みません。



この記事では、この様な例としてアジャイル開発を取り上げて、新しいプロセスモデルに不足するノウハウの補完法を議論します。

図1 新しいプロセスへの移行

🚧 アジャイル開発ケーススタディ

実現が難しかったノウハウとアジャイル開発

「アジャイル開発は、ソフトウェア工学の正統な進化形」[2]と言われる様に実現が困難だったノウハウをアジャイル開発はモデル化しています。スクラムを例に、解決した問題例の一部を以下に示します。

- ✓ 「現地現物」はプロジェクト管理のノウハウですが、従来は成果物、特に実装を確認するまでに時間がかかり、実装確認後のフィードバックが困難でした。スクラムでは繰り返しの単位であるタイムボックス毎のスプリントごとにレビューを実施することで、適切なフィードバックが可能になっています。
- ✓ ハンフリーはソフトウェア開発に「自律的なチーム」や「その最大限の能力を最大限発揮できるようメンバーを動機付け、コーチし、後押しする」ことが重要だとしましたが[3]、規律に基づく組織はトップダウン的で困難な場合もありました。スクラムではスクラムマスターがチームを支援する奉仕者であり、真のリーダーとなっており[4]。

このように従来は困難だったノウハウがスクラムのプロセスモデルでは解決されています。

スクラムにおけるプロセスモデルの補完

ベームはリスク管理されていることをリスク駆動と呼び、スクラムに対してマネジメントプロセスは制約の源泉として完全に適用できるが、リスク/機会は制約として適用できるのは部分的とし、強いリスク駆動とはされていません[5]。スクラムはプロダクトバックログ（PBL）につけられた優先順序に応じてストーリーを実現する仕組みだからです。

これに対して、ジェフサザーランドは、プロダクトオーナー研修で「PBLは価値最大、リスク最小で優先付け」と説明して[6]、強いリスク駆動でないスクラムのモデルを補完するノウハウを示しています。また、開発チームにおいては妨害リストが管理されるなど、様々なノウハウでプロセスモデルを補完してリスクが管理されるようになっています。

その一方で、イテレーティブな開発を「はじめに小さく失敗する」と説明して、小さな単位で開発することであたかも全てのリスクが減るように表現されることもあります。また、リスクに関して記述の少ない書籍も多くあります。これはスクラムのプロセスがリスクに対して未熟というよりは、対象とする業務によっては、リスクに対する考慮が必ずしも必要ではないからでしょう。つまり、モデルに組み込まれていないノウハウは、業務などに応じて取捨選択されている可能性があるでしょう。

議論

ここではスクラムという汎用的なプロセスをリスク管理で考えましたが、各企業のプロセスも同様です。プロセスモデルだけでなく、それぞれの対象業務に応じてプロセスモデルを補うノウハウの議論が必要でしょう。

プロセスは、プロセスモデルだけでなく、ノウハウや人を含めた文化であり、モデルを忠実に実践するだけでは成長させることが困難な場合があります。社内外の知識の利用が必要で、コミュニティや書籍、経験者の知恵を借りることで、より良いプロセス改善を実施することが可能になるでしょう。

例に挙げたリスクに関しても、複数チームでの開発などではリーンソフトウェア開発の「決定をできるだけ遅らせる」プラクティスなども参考になるでしょう[7]。また業務固有のリスクについて、社内外の知見者の意見は大いに役立つと考えられます。このように収集したノウハウを組織に合わせて取捨選択し、モデルを補完すればプロセスを成長させられるでしょう。

おわりに

CMM ブームの後、TQC を懐かしむ声を聞くことができました。新しいプロセスを導入した際に、それまでに積み上げてきた文化を無くして移行してしまったからです。単に捨ててしまうのではなく、重要なノウハウを利用してプロセスを成長させていくべきでしょう。この記事をきっかけに、プロセスをどのように改善し、成長させていくかといった前向きでオープンな議論発展することを期待しています。

引用文献

- [1] アジャイルよろず相談室, 「ウォーターフォールでもしっかり開発出来ないのにアジャイル開発なんて出来るわけがない」という言い方をどう思いますか?, <https://qr.ae/pGLjdc>, Quora, Inc. .
- [2] Yasuharu NISHI, ちなみに僕のスタンスは、アジャイルはソフトウェア工学の正統な進化形だということです。進化がアンチテーゼを必須にするというなら、アジャイルは一度ソフトウェア工学を否定しなきゃいけないんでしょうし、それも含めて正統な進化形です。 , <https://twitter.com/YasuharuNishi/status/1164336449967099904>, Twitter, 2019.
- [3] 阪井, デブサミ運営事務局・SEshop.com 編集部編, リーダーに求められる大切なこと, 100 人のプロが選んだソフトウェア開発の名著, pp.20-21, 翔泳社, <https://www.slideshare.net/MakotoSAKAI/ss-16581244>, 2012.
- [4] Ken Schwaber, Jeff Sutherland, スクラムガイド,
<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Japanese.pdf>, ScrumGuides.org, 2020.
- [5] バリー・ベーム, アジャイルと規律, pp.205-210, 日経 BP 社, 2004.
- [6] Sucusuku Scrum, スクラムプロジェクト準備(公開用) No.31, <https://www.slideshare.net/SucusukuScrum/no31-13338313/41>, p.41, 2012.
- [7] メアリー・ポッペンディーク, トム・ポッペンディーク, リーンソフトウェア開発, pp.79-109, 日経 BP 社, 2004.

GSLetterNeo Vol.170

2022年9月20日発行

発行者 株式会社 SRA 技術本部 先端技術研究室

編集者 熊澤努 方学芬

バックナンバー <https://www.sra.co.jp/public/sra/gsletter/>

お問い合わせ gsneo@sra.co.jp



〒171-8513 東京都豊島区南池袋 2-32-8

夢を。



夢を。Yawaraka Innovation
やわらかいのべーしょん