

AgileXDDP

八木 将計
株式会社日立製作所
masakazu.yagi.zd@hitachi.com

斎藤 賢一
株式会社エクスマーシオン
kenichi.saito@exmotion.co.jp

会田 圭司
テクマトリックス株式会社
aita@techmatrix.co.jp

永田 敦
ソニー株式会社
inagworld@gmail.com

山田 謙輔
派生開発推進協議会 T6 研究会
yamaken1979@gmail.com

石川 宏保
派生開発推進協議会 T6 研究会

星野 充史
アンリツエンジニアリング株式会社
Atsushi.Hoshino@anritsu.com

要旨

本報告では、広がりを見せている Agile 開発[1][2][3]と、派生開発手法である XDDP (eXtreme Derivative Development Process) [4][5]の二つの手法において、双方の特徴を組み合わせることで、それぞれの課題を解決する「AgileXDDP」を提案する。AgileXDDP は、Agile 開発に XDDP の考え方を導入したものと、XDDP に Agile 開発の考え方を導入したものの 2 つを提案する。

1. はじめに

近年、ソフトウェア開発において、スクラムなどの Agile 開発が一般的に広まっている[1][2][3]。Agile 開発の多くは、イテレーション(反復開発)することで、リスクを最小限にしつつ、迅速で柔軟な開発を実現する。そのため、自動テストなどの反復によるデグレードなどのトラブルを防止する仕組みが含まれる。しかし、多くの反復を繰り返していくうちに、自動テスト等の仕組みの保守が難しくなり、デグレードなどのトラブルが発生することがある。また、自動テスト等の仕組みがないソフトウェアに機能追加をする場合などに、導入が難しくなりやすい。

一方、ソフトウェアの派生開発の手法として XDDP (eXtreme Derivative Development Process)が注目されてきている[4][5]。XDDP は、派生開発における変更点に注目し、それに特化したドキュメントを作成・レビューすることで、デグレード等のトラブルを防止する手法である。しかし、この変更に関するドキュメントは作成スキルが必要であり、担当者のスキルが未熟な場合、通常のレビュープロセスでは、ドキュメント作成で手戻りが発生するという

課題がある。

派生開発推進協議会 T6 研究会では、「Agile 開発との連携」というテーマで、XDDP と Agile 開発の連携を研究している[6]。T6 研究会では、本報告にて、上記の Agile 開発での課題と XDDP での課題を解決する手法として、「AgileXDDP」を提案する。提案手法は、Agile 開発に XDDP の要素を持ち込んだ「機能追加の AgileXDDP」と XDDP に Agile 開発の要素を持ち込んだ「変更の AgileXDDP」の二つがある。

2. Agile と XDDP の課題

2.1. Agile 開発の課題

Agile 開発では、構成管理や継続的インテグレーションによる自動テストによって、通常よりもデグレードなどのトラブルは発生しづらいが、イテレーションを繰り返していくなど、ソフトが増大化していくと、デグレードなど、一般的な派生開発と同じような問題が発生する。

また、Agile 開発は、開発の成果物が刻々と変化するような新規開発に適した手法であり、既に動いているものに機能追加するには、困難な場合がある。特に、仕様やテストケースが失われている場合、デグレードを防ぐのが難しく、適用が困難になる。

2.2. XDDP の課題

XDDP は、機能追加プロセスと変更プロセスを分離しており、特に変更プロセスに「変更 3 点セット(変更要求仕様書, TM, 変更設計書)」というドキュメントを作成・レビュー

一することで、変更によるデグレードといった手戻りを防止する方法である。しかし、この変更3点セット、特に、変更要求仕様書に用いられる USDM (Universal Specification Describing Manner)[7]の記載にスキルが必要になる。

そのため、担当者のスキルが未熟な場合、変更3点セットが完成してからレビューを行うと手戻りが発生しやすいという課題がある。また、レビューで十分に精査できていない状況で開発を進めると、後工程での大きな手戻りになるとともに、担当者のスキル向上に繋らず、最終的に変更3点セットが形骸化する可能性もある。

3. 二つの「AgileXDDP」

3.1. 機能追加の AgileXDDP

2.1 節に示す Agile 開発の課題を解決するため、Agile 開発に XDDP の考え方・やり方を導入する。本報告では、主に機能追加プロセスに関する手法なので、本手法を「機能追加の AgileXDDP」と呼ぶ。

機能追加の AgileXDDP は、Agile 開発の一つの機能追加において、図 1、図 2 に示す通り、一度 Master から Feature ブランチを切り、機能を実現し、顧客等とレビューを行う。結果、よければそれを母体である Master に反映することになるが、その際に、XDDP の「移植」を用いる。

Feature ブランチを試作台として用いることで、Agile 開発特有のスピードを実現しつつ、XDDP の移植を用いることで Master に対する影響分析し、デグレードなどの手戻りの悪影響を防ぐ方法となる。一見、移植部分が無駄なように感じられるが、「移植」はそもそも別のソースコードから別のソースコードへ機能を持ってくる作業なので、Develop などで Master が違っていても、その差による影響を最小化することができると考えられる。

機能追加の AgileXDDP では、Master が異なっても対応可能になるため、図 3、図 4 のように一部の作業を並行にする応用も可能になると考えられる。また、Master への悪影響を最小化できるので、複数チームで同一の Master に対して機能追加することも可能となると考えられる。

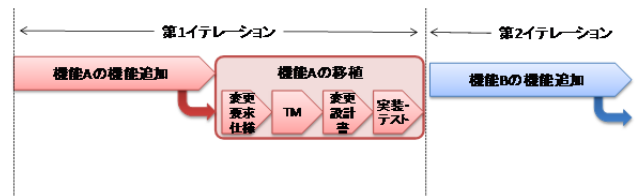


図 1 機能追加の AgileXDDP

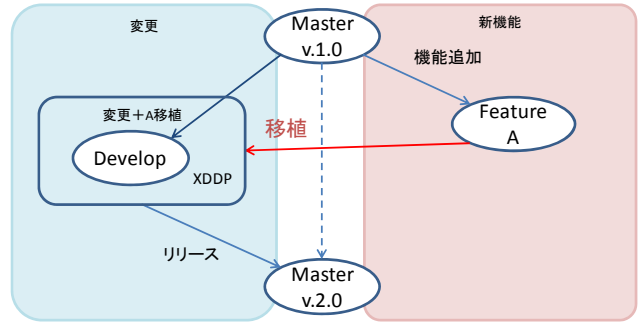


図 2 機能追加の AgileXDDP の構成管理

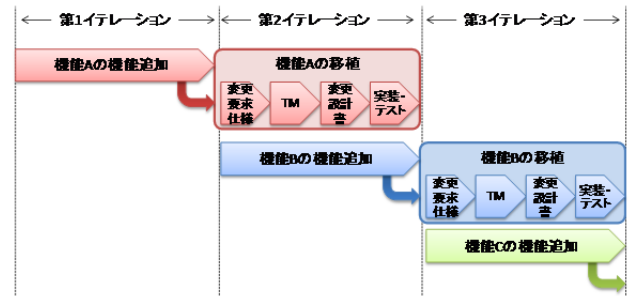


図 3 機能追加の AgileXDDP ver.2

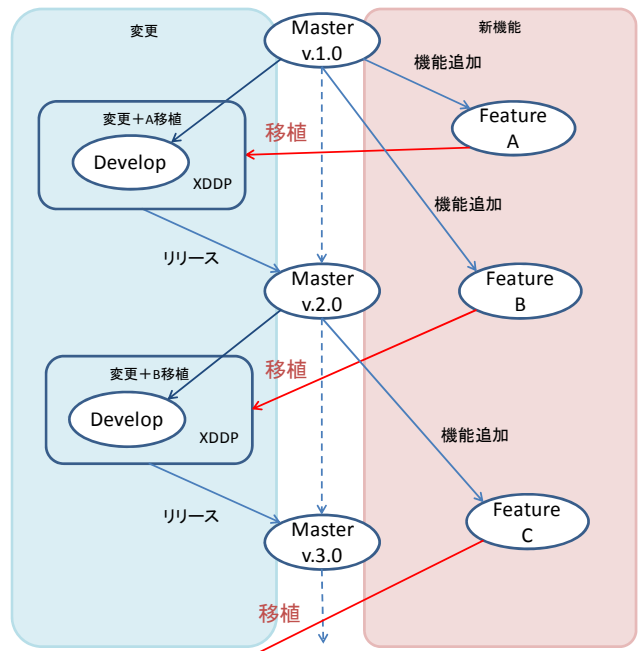


図 4 機能追加の AgileXDDP ver.2 の構成管理

3.2. 変更の AgileXDDP

2.2 節に示す XDDP の課題を解決するため、XDDP に Agile 開発の考え方・やり方を導入する。本報告では、主に変更プロセスに関する手法なので、本手法を「変更の AgileXDDP」と呼ぶ。

変更の AgileXDDP は、XDDP の変更プロセスの変更 3 点セットの作成において、タイムボックスを用いて、イテレーティブに成果物を作成する。特に、変更要求仕様書である USDM では、記載技術が必要になるので、「アジャイルインスペクション¹」を用いて、担当者のスキル向上を含める。また、このアジャイルインスペクションを用いることで、全ての仕様を記述してからレビューするのではなく、要求定義の早期の段階から要求者へフィードバックすることで手戻り防止の効果を狙う。

この変更の AgileXDDP を用いることで、変更 3 点セットの作成技術を醸成しながら、後工程での手戻りも防ぐことができると考えられる。

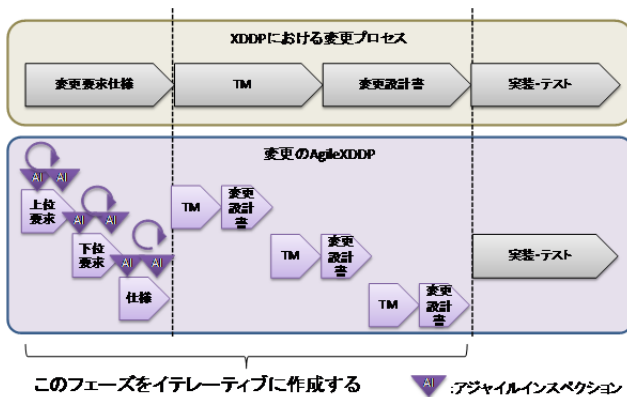


図 5 変更の AgileXDDP

4. 今後の課題

本報告で示した、二つの AgileXDDP は、まだ構想の段階であり、実際に適用することが今後の一番大きな課題となる。

また、この二つの AgileXDDP は、それぞれ機能追加と変更に着目しており、両方をハイブリッドすることも可能になると考えている。そちらの検討も進めていきたい。

¹ アジャイルインスペクションとは、ドキュメント作成途中でサンプリング、短いタイムボックスでレビューし、決められた品質になるまで修正/サンプリング/レビューを繰り返すことで、ドキュメントと書き手の品質レベルを向上する手法である[8][9]。

5. まとめ

本報告では、派生開発推進協議会 T6 研究会の研究成果として、Agile 開発と XDDP の特徴を活かした二種類の「AgileXDDP」を提案した。まだ、実証ができていない方法ではあるが、この二つの AgileXDDP により Agile 開発、XDDP 双方における課題が解決できることを期待している。

参考文献

- [1] Beck, et. al., Manifesto for Agile Software Development. <http://www.agilemanifesto.org/>
- [2] 独立行政法人情報処理推進機構, アジャイル型開発におけるプラクティス活用事例調査 調査報告書, 東京, 2013.
- [3] 平鍋健児, 野中郁次郎, アジャイル開発とスクラム, 翔泳社, 2013.
- [4] 清水吉男, 「派生開発」を成功させるプロセス改善の技術と極意, 技術評論社, 2007.
- [5] K. Kobata, E. Nakai and T. Tsuda, “Process Improvement Using XDDP: Application of XDDP to the Car Navigation System,” Proc. 5th World Congress for Software Quality, (Nov. 2011).
- [6] 派生開発推進協議会 AFFORDD <http://www.affordd.jp/>
- [7] 清水吉男, 改訂第 2 版 [入門+実践] 要求を仕様化する技術・表現する技術～仕様が書けていますか?, 技術評論社, 2010.
- [8] T.Gilb, “Inspection Facts: To Help You Make Good Decisions to Do Software Inspections and Reviewers Properly,” ソフトウェア品質シンポジウム 2008 (SQiP2008), 2008.
- [9] 永田敦, 森崎修司, “アジャイルインスペクションの実際,” ソフトウェアテストシンポジウム 2010 (JaSST’10 Tokyo), 2010.