

アジャイルメトリクスパタンの提案

伊藤 宏幸 ヤフー株式会社
細谷 泰夫 三菱電機

アブストラクト

我々は、ソフトウェアプロダクト開発を「アジャイル」で実施する際の、「メトリクス」の取得・活用に関するパターン（以下「アジャイルメトリクスパターン」とする）について提案する。

「アジャイルメトリクスパターン」は、ソフトウェアプロダクト開発の課題を迅速に発見し、それらを仮説検証と改善の繰り返しによって解決していくことを促進する。また、仮説検証と改善の繰り返しの起因するボトルネックの発生・移動についても追跡し、その解決を図る。さらに、目的の共有やメトリクスの創出などの活動を通じて、プロダクト開発チームや組織を成長させることにも寄与する。

1. はじめに

ここ数年、アジャイルを採用した日本及び欧米のソフトウェアプロダクト開発において、アジャイルに関するメトリクスを取得・活用して仮説検証と改善を繰り返し実施している事例が増えている。事例の増加を鑑みるに、この傾向は一時的なものではなく、一定の普遍性・一般性を持つものと考えられる。また、メトリクスの取得・活用の事例を観察すると、そこに一定の法則性を見出すことができる。

そこで本論文では、特に日本の実際のアジャイルソフトウェアプロダクト開発の実例に基づき、「アジャイルメトリクスパターン」を抽出し提案することとする。

本論文で提案しているメトリクスの活用方法は、Goal-Quality-Metrics パラダイム（以下、GQM）と密接に関係している。GQMは「明確に目的を捉えて、目標に対して必要なメトリクスを対応付けるゴール指向（目的指向）な枠組み」[1]であるが、本論文のパターンランゲージはGQMパラダイムのようなメトリクスを用いた漸進的な改善活動を実施する際に直面する問題点と解決方法を提示するものである。

2. パターン一覧

本論文にて提案するパターンの一覧を以下に示す。

No	パターン名	要約
1	問題にフォーカスする	自分たちの抱えている問題にフォーカスしたメトリクスを計測しよう
2	近い未来を予測する	予測と実績の差分とその要因がわかるように、近い未来を予測してその予測を裏付けるメトリクスを計測、評価しよう
3	変化が見えるかに着目する	改善の効果がわかるように変化が見えるかに着目してメトリクスを計測しよう
4	自動化の恩恵に預かる	自動化によって、計測の手間を減らそう
5	複数を組み合わせる	複数のメトリクスを組み合わせで判断しよう

No	パターン名	要約
6	自分たちで創る	チームの問題を把握するために適した既存のメトリクスが無い場合は自分たちで計測するメトリクスを創ろう
7	目的を共有する	何を改善したくて、そのメトリクスを計測しているのかという目的を共有しよう
8	成長を実感する	チームが自らの成長を実感できるメトリクスを計測し、改善のモチベーションを上げよう
9	ボトルネックを追いかける	ボトルネックの移動に応じて、計測するメトリクスを変更しよう
10	会話を促す	メトリクスをキッカケにして自分たちの抱える問題や改善活動についての会話を促そう

3. パタン関連図

パタン同士の関連を示す。図における矢印はパタンの適用順序を表す。ただし、パタン「会話を促す」については、全てのパタンと並行して適用する可能性がある。これからメトリクスの活用を始める場合は、「問題にフォーカスする」から適用を開始する。すでにメトリクスを活用している場合は、現状のコンテキストに合っているパタンから適用を開始することができる。

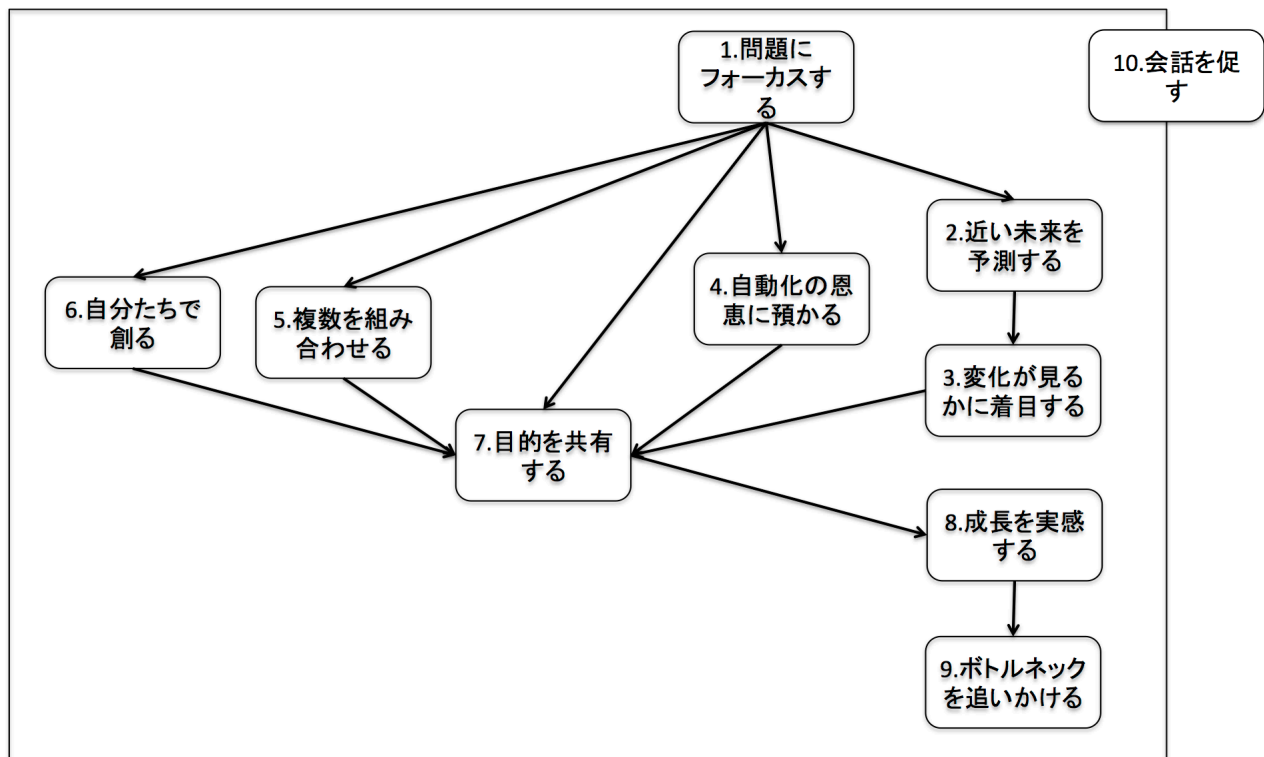


図 1 パタン関連図

4. パタンの説明

以下にそれぞれのパタンについて説明する。

4.1 問題にフォーカスする

4.1.1 要約

自分たちの抱えている問題にフォーカスしたメトリクスを計測しよう。

4.1.2 状況

チームのメンバーは、自分たちが問題を抱えているという意識を持っているが、何が問題であるかを明確に把握できておらず、メトリクスを計測して状況を見えるようにしたいと考えている。

4.1.3 問題

メトリクスを計測し改善に活用したいと考えているが、どんなメトリクスを計測したらいいかわからない。

4.1.4 フォース

世の中で計測すべきだと言われているメトリクスの種類は多く、計測するための手間が大きそうに見える。導入するメトリクスを絞り込むことによって、増える手間は小さくなるだろう。

4.1.5 解決方法

自分達がどのような問題を抱えているかをチームで話し合い、自分たちが抱えていると思っている問題が発生している度合い、影響にフォーカスして、計測対象のメトリクスを1つか2つに絞り込む。絞り込んだメトリクスが適切でなければ、都度見直す。

4.1.6 結果状況

計測するメトリクスの数を絞り込むことによって、メトリクスの計測を無理なくチームに適用することが可能となる。また、問題にフォーカスしたメトリクスを計測しているため、チームが抱える問題点を発見しやすい。

4.1.7 適用事例

過去にアジャイルの導入を支援したチームでは、常に事前にどこに問題があるかをメトリクス計測によって明確化し、それによって見つけた問題の解決にフォーカスしたアクションを取ることを徹底した。また、「完了率」や「割り込み率」など、その時点での問題に適したメトリクスのみを計測対象を絞った[2]。

4.2 近い未来を予測する

4.2.1 要約

予測と実績の差分とその要因がわかるように、近い未来を予測し、その予測を裏付けるメトリクスを計測、評価しよう。

4.2.2 状況

開発チームは問題を抱えており、メトリクスを計測して問題を把握して改善したいと考えている。開発はまだしばらく続くので継続的に改善したい。

4.2.3 問題

メトリクスの計測、評価のスパンを長く設定してしまうと、予測と実績の差分がある場合に、様々な要因が混在してしまっていて状況を把握することが困難になる。

4.2.4 フォース

近い未来の予測は、人によるばらつきも少なく共有しやすい。

4.2.5 解決方法

近い未来（例えば1～2週間後）に対して仮説に基づいた予測を行い、それを裏付けるメトリクスの計測、実績の評価を実施する。

4.2.6 結果状況

近い未来のことなので予測と実績の差分があった際に、その原因を特定することが容易である。原因をはっきり特定することで、改善のポイントを明確にすることができる。

4.2.7 適用事例

過去にテスト自動化普及の特命チームを編成した際、テスト自動化目標の達成度を1週間ごとにチームメンバー及びステークホルダーに共有するようにした。これをもとに1週間ごとに仮設定と検証を繰り返しながら施策を進めていき、最終的に当初目標を大きく上回る達成度を実現した[3]。

4.3 変化が見えるかに着目する

4.3.1 要約

改善活動の効果を確認するために、数値の変化に着目してメトリクスを計測しよう。

4.3.2 状況

チームのメンバーは、フォーカスしている問題に対する改善活動の効果の有無を確認したいと考えている。

4.3.3 問題

フォーカスしている問題に対するメトリクスを計測しているが、何を持って改善活動の効果が出ているのかを判断することができない。

4.3.4 フォース

取るべきメトリクスよりも、取りやすいメトリクスを計測対象にしてしまう。

4.3.5 解決方法

「問題にフォーカスする」ことを前提として、改善活動により数値が変化することが予想できるメトリクスを計測対象とする。「近い未来を予測する」によって設定した期間内で変化が予想できるメトリクスを計測することが効果的である。

4.3.6 結果状況

チームが抱える問題に対して実施した改善による変化が見えるようになり、改善の効果を知ることができる。

4.3.7 適用事例

過去にアジャイルの導入を支援したチームでは、常に事前にどこに問題があるかをメトリクス計測によって明確化し、それによって見つけた問題の解決にフォーカスしたアクションを取った。その際、アクション実施前後のメトリクスの変化に注目することで、効果の有無・程度を確認しながらアクションを取捨選択・実施した[4]。

4.4 自動化の恩恵に預かる

4.4.1 要約

自動化によって、計測の手間を減らそう。

4.4.2 状況

チームはメトリクスを計測したいが、メトリクス計測の手間が増えることを懸念している。

4.4.3 問題

メトリクス計測の手間を大きいと、負担感が大きくなってしまい導入することが困難になる。

4.4.4 フォース

計測の手間が大きく感じると、メリットを感じることができず導入のモチベーションが上がらない。

4.4.5 解決方法

テスト自動化、ビルド自動化等の自動化と組み合わせてメトリクスを自動で計測できるようにすることで、メトリクス計測の手間が減らし導入を容易にする。

4.4.6 結果状況

メトリクス計測を自動化することで、チームの負担感を減りメトリクスを導入することができる。一方で、自分たちの抱えている問題と自動で計測できるメトリクスが関連していない場合は、改善に繋がらないので注意が必要である。

4.4.7 適用事例

過去に支援したチームで、タスクの現状を把握したいという要望が出たことがあった。ちょうどタスク管理に JIRA を使用していたため、その CFD 及び dashboard を活用して、タスクの全体像と傾向を自動的に収集・確認できるようにした。チームはこの情報をもとに施策を実施するようになった[5]。

4.5 複数を組合せる

4.5.1 要約

複数のメトリクスを組み合わせて判断しよう。

4.5.2 状況

チームはメトリクスを計測しているが、単一のメトリクスで状況の良し悪しを判断している。その結果として、そのメトリクスの数値を改善すること自体が目的になってしまっている。

4.5.3 問題

単一のメトリクスの数値だけでは、本当に問題があるかどうか分からないことが多い。その結果として、数値の変化に一喜一憂してしまい、数値を上げる（下げる）こと自体が目的化してしまう。

4.5.4 フォース

単一の数値目標を設定して、達成するという活動はわかりやすく、数値自体が目的化しやすい。

4.5.5 解決方法

複数のメトリクスから多面的に状況を判断することによって、問題の有無を把握することができる。

4.5.6 結果状況

単一のメトリクスにより状況の良し悪しを単純に判断しないことによって、メトリクスの数値自体が目的化することがなくなり、数値そのものより問題の有無についての議論にフォーカスすることができる。

4.5.7 適用事例

過去にテスト自動化の普及を支援したチームで、「コードカバレッジ」のみを計測した際、この値を上げるためだけに価値の低いテストスクリプトを大量に作られてしまうという問題が生じた。これを解決するために、Pull Request のレビューコメント数・テストスクリプトの行数の変化も併せて計測するようにし、「価値の高いテストが作成されていること」を計測・追跡するようにした[6]。

4.6 自分たちで創る

4.6.1 要約

チームの問題を把握するために適した既存のメトリクスが無い場合は、自分たちで計測するメトリクスを創ろう。

4.6.2 状況

「問題にフォーカスする」によりチームが抱えている問題を把握するためのメトリクスを計測しようとしているが、よく知られたメトリクスに適したものが無い。

4.6.3 問題

チームが抱えている問題にフィットしない既存のメトリクスにより状況を判断しようとしても、適切に状況を把握することはできない。

4.6.4 フォース

世の中で知られている既存のメトリクスを利用しなければならないという思い込みがある。

4.6.5 解決方法

問題にフォーカスする」によって、チームが抱えている問題を把握するために最適なメトリクスを自分たちで考えて計測する。

4.6.6 結果状況

チームが抱えている問題を把握できるようになり、改善活動の効果を確認することができる。

4.6.7 適用事例

過去にアジャイルの導入を支援したチームで、割り込み作業の増加という課題に直面した際、稼働率などの既存のメトリクスでは現状をうまく捉えることができなかつた。そ

ここで、作業がどの程度割り込まれたのか、その時間と頻度そのものを計測することとし、これを「割り込み率」と命名した。結果、一定期間ごとの作業割り込みの程度を時間と回数で明示できるようになり、マネージャーやステークホルダーとの作業ボリュームの調整をスムーズに行えるようになった[7]。

4.7 目的を共有する

4.7.1 要約

何を改善したくて、そのメトリクスを計測しているのかという目的を共有しよう。

4.7.2 状況

チームはメトリクスを計測しているが、メトリクスの値を上げる（下げる）ことが目的になってしまい、不適切なアクションが取られている。

4.7.3 問題

メトリクスを計測する目的が共有されておらず、値を上げる（下げる）こと自体が目的化してしまっている。その結果、本来実施したい改善につながらないアクションが実行されてしまう

4.7.4 フォース

単純な数値目標はわかりやすく目的化しやすい。

4.7.5 解決方法

メトリクスを計測する目的を共有することで、メトリクスの数値を上げる（下げる）こと自体の目的化を防止する。「問題にフォーカスする」で検討した内容をメトリクス計測に関わる人全員と共有する。

4.7.6 結果状況

メトリクスを計測する目的を共有することで、数値の変化に一喜一憂せずに、「問題にフォーカスする」で改善対象とした問題についてメンバー全員が状況を把握し、改善活動の効果を知ることができる。

4.7.7 適用事例

過去にテスト自動化普及の特命チームを編成した際、テスト自動化目標の「達成度」をチームの達成すべき目標と定め、これを定期的にメンバーに示して共通認識を構築しながら施策を進めた。結果、メンバーの自発性を促すこととなり、最終的に当初目標を大きく上回る成果を実現した[8]。

4.8 成長を実感する

4.8.1 要約

チームが自らの成長を実感できるメトリクスを計測し、改善のモチベーションを上げよう。

4.8.2 状況

チームはメトリクスを計測しているが、「計測しているだけ」という状態になっており、改善のモチベーションも下がっている。

4.8.3 問題

チームが改善活動をした結果の変化がメトリクスの実績として現れないと、メトリクスを計測するメリットが感じられない。

4.8.4 フォース

チームの成長を実感する機会は明示的に設けられていないケースが多い。

4.8.5 解決方法

チームの改善活動による変化が実績として計測できるメトリクスを設定することにより、改善活動が有効だったかどうかを知ることができる。

4.8.6 結果状況

改善活動による変化を把握できるようになり、チームが成長を実感できるようになる。その結果として、新たな改善の試みをチームがより積極的に実施するようになる。

4.8.7 適用事例

過去にテスト自動化普及の特命チームを編成した際、テスト自動化目標の「達成度」を週次で計測し、変化がなければそこを徹底的に教育・支援するように振る舞った。またその成果を「数値の変化」としてチームメンバー全員に示し、自分たちのやっていることが正しいことを体感させた。これらを継続した結果、メンバーの自発性を促すこととなり、最終的に当初目標を大きく上回る成果を実現した[9]。

4.9 ボトルネックを迫りかける

4.9.1 要約

ボトルネックの移動に応じて、計測するメトリクスを変更しよう。

4.9.2 状況

チームは問題を抱えていたが、「問題にフォーカスしよう」によってメトリクスを計測し、問題を把握し、改善を進めることによって、抱えていた問題は解決した。しかし、チームのパフォーマンスは依然として悪いままである。

4.9.3 問題

当初抱えていた問題を解決してしまうと、ボトルネックが異なる問題に移動してしまい、現状計測しているメトリクスでは状況を把握できなくなる。

4.9.4 フォース

一つの問題を解決しても、全体として改善が進むかどうかわからない。複数の問題が組み合わさっている場合は、問題を解決すると、別の問題が阻害要因になることが多い

4.9.5 解決方法

ボトルネックが他の問題に移動したと感じたら、次に解決すべき問題にフォーカスしたメトリクスを計測する。

4.9.6 結果状況

ボトルネックの移動を認識し、次の問題に対応したメトリクス計測を実施することで、状況を共有し有効な改善を実施することができる。

4.9.7 適用事例

過去にアジャイルの導入を支援したチームでは、当初は手動による回帰テストの負荷が課題だったため、これをCI/CDに置き換えて作業時間を1/50以下に減らしたが、一方で作業割り込みの増加という新たな課題に遭遇した。そこで次は割り込み作業を計測し改善したところ、今度はユースケースレベルのバグの頻発という別の課題に直面した。プロダクト開発を続ける限り、課題は無くならない。課題解決は、ボトルネックの移動を産む。このことを念頭に置きながら、常にボトルネックを追跡・計測しておくことが、プロダクト開発の継続的な改善には重要である[10]。

4.10 会話を促す

4.10.1 要約

メトリクスをキッカケにして自分たちの抱える問題や改善活動についての会話を促そう。

4.10.2 状況

チームは問題を抱えているが、それぞれのメンバーはその問題について明示的に話し合う機会がなく、改善のモチベーションも上がらないメトリクスも計測しているが、計測した結果についてチームで話し合うことがない。

4.10.3 問題

チームで問題を抱えていても口に出して会話しなければ、改善する機会を得ることができない。

4.10.4 フォース

チームメンバーは問題を抱えていても、明示的に口に出しにくい。客観的に見えるものがあると、会話するキッカケになる。

4.10.5 解決方法

メトリクスをキッカケにチームで会話をすることによって、問題や改善について共有する。「問題にフォーカスする」「成長を実感する」など計測対象のメトリクスの選定や、実績の評価の際に、問題や改善方法そのものについて会話する。

4.10.6 結果状況

チーム内で、自分たちが抱える問題やその改善策について、メトリクスを使って活発に会話することによって、改善のモチベーションを上げることができる。

4.10.7 適用事例

過去に支援したチームで、ソースコードレビューの負荷の高さが議論になったことがあった。そこで、シール（ドット）を使ってレビュー頻度を実際に計測・見える化しようとしたところ、その手軽さ・面白さから、チームメンバーが積極的に協力してくれ、結果3日程度で問題の所在が一目瞭然となった。また、チームメンバー自ら話し合い、この問題の解決策を提示し実行してくれた[11]。

5. 結論と今後の課題

以上から、アジャイルを採用したソフトウェアプロダクト開発において、「アジャイルメトリクスパターン」が存在すること、及びそれらを活用することでソフトウェアプロダクト開発・チーム・組織をさらにより良い状態にすることができることを示した。一方で「アジャイルメトリクスパターン」には、以下の課題が存在することについても認識している。

5.1 さらに事例の収集・分析によるパタンの洗練

本論文で提案した「アジャイルメトリクスパターン」は、特定の会社の事例しか分析していないため、まだ広い一般性は確保できていない恐れがある。そのため、複数の会社からさらなる事例を収集・分析し、パターンそのものを洗練し、広い一般性を担保できるようにする必要がある。

5.2 日本におけるアジャイルを阻害する要因のパタンによる解決

特に日本では、アジャイルがソフトウェアプロダクト開発に広く浸透しているとはまだ言い難い状況である。そこで、日本でアジャイルを阻害する要因を分析し、それを解決する観点から、「アジャイルメトリクスパターン」を活用することを考えてみる必要があるであろう。

5.3 欧米の知見の継続的導入・統合

欧米ではアジャイルメトリクスを、ソフトウェアプロダクト開発のプラクティスのリストとして、一般化・統合する方向へ向かいつつある。我々は、この動きを逐一観察し、「アジャイルメトリクスパタン」へ継続的に導入・統合していく必要があると考える。

参考文献

1. 伊藤 宏幸. 2017. **アジャイルメトリクス実践ガイド**. Slideshare.
2. 伊藤 宏幸. 2016. **世界と事例から学ぶ、プロダクトオーナーの「素養」としてのアジャイルメトリクス**. Slideshare.
3. 伊藤 宏幸. 2015. **CI サーバを制圧せよ！ - プロジェクトメトリクスと自動化技術の活用による混乱の收拾と「最強」の組織の育成**. Slideshare.
4. 伊藤 宏幸. 2014. **メトリクスによる「見える化」のススメ: エssenシャル・リーン**. Slideshare.
5. 伊藤 宏幸. 2014. **現場実践主義としてのリーン開発とアジャイル**. Slideshare.
6. 鷲崎 弘宜. 2013. **メトリクスによるプロダクトの品質把握と改善 - Goal-Question-Metric (GQM) 法のコツ、および、組織目標との整合**. Slideshare.

-
1. 鷲崎 弘宜. 2013. **メトリクスによるプロダクトの品質把握と改善 - Goal-Question-Metric (GQM) 法のコツ、および、組織目標との整合**. Slideshare.
 2. 伊藤 宏幸. 2014. **現場実践主義としてのリーン開発とアジャイル**. Slideshare. P16-P38
 3. 伊藤 宏幸. 2015. **CI サーバを制圧せよ！ - プロジェクトメトリクスと自動化技術の活用による混乱の收拾と「最強」の組織の育成**. Slideshare. P75-P84
 4. 伊藤 宏幸. 2014. **現場実践主義としてのリーン開発とアジャイル**. Slideshare. P16-P38
 5. 伊藤 宏幸. 2014. **メトリクスによる「見える化」のススメ: エssenシャル・リーン**. Slideshare. P28-P31
 6. 伊藤 宏幸. 2017. **アジャイルメトリクス実践ガイド**. Slideshare. P54-P66
 7. 伊藤 宏幸. 2014. **現場実践主義としてのリーン開発とアジャイル**. Slideshare. P24-P31
 8. 伊藤 宏幸. 2015. **CI サーバを制圧せよ！ - プロジェクトメトリクスと自動化技術の活用による混乱の收拾と「最強」の組織の育成**. Slideshare. P75-P84
 9. 伊藤 宏幸. 2015. **CI サーバを制圧せよ！ - プロジェクトメトリクスと自動化技術の活用による混乱の收拾と「最強」の組織の育成**. Slideshare. P75-P84
 10. 伊藤 宏幸. 2014. **現場実践主義としてのリーン開発とアジャイル**. Slideshare. P16-P38
 11. 伊藤 宏幸. 2014. **メトリクスによる「見える化」のススメ: エssenシャル・リーン**. Slideshare. P15-P22