

# リーンのコンセプトによるアジャイル開発の生産性指標の提案とウォーターフォール開発との比較

秦泉寺 久美<sup>1,a)</sup> 神明夫<sup>1,b)</sup> 村本 達也<sup>1,c)</sup>

受付日 2020年2月19日, 採録日 2020年9月10日

**概要:** 近年のデジタルディスラプション時代の到来によりアジャイル開発の重要性が認識されつつあるものの、受託でソフト開発をしている発注側組織において、ウォーターフォール型の開発からアジャイル型の開発へ組織的な方針転換に踏み出せない場合がある。その理由の1つとして旧来からウォーターフォール型の開発データや生産性の評価をしてきているなかで、アジャイル開発の生産性（費用、期間）の善し悪しを判断できないことが考えられる。本稿では、価値の構築に着目し生産性を期間および費用の指標として定式化し、同一の基準で案件間の生産性を比較評価する方法を提案する。筆者らはアジャイル開発を推進し、準委任契約での運用と開発の両立を確認するためにアジャイル開発トライアルを2018年から実施しており、本指標を用いてトライアル案10案件および過去のウォーターフォール型開発の統計値を比較したところ、NTT研究所の例では、アジャイル開発において10%以上プロダクトの機能使用率が向上すれば、ウォーターフォール開発に対して費用、期間面で優位な開発になりうることが分かった。

**キーワード:** アジャイル開発, 生産性, 指標, リーン

## Productivity Evaluation for Agile Development Using Lean and Comparison of Waterfall Development

KUMI JINZENJI<sup>1,a)</sup> AKIO JIN<sup>1,b)</sup> TATSUYA MURAMOTO<sup>1,c)</sup>

Received: February 19, 2020, Accepted: September 10, 2020

**Abstract:** With the advent of the digital disruption era in recent years, the importance of agile development is becoming increasingly recognized. However, ordering company in contract development of software are unable to make a systematic change from waterfall development to agile development. One of the reasons is that it is difficult to judge whether the productivity (cost, duration) of agile development is superior or not in the context of traditional waterfall development data and productivity evaluations. In this paper, we propose a method to compare and evaluate the productivity of projects based on the same criteria. We have been conducting agile development trials using delegation agreements since 2018. Using this indicator, we compared 10 agile trial projects to waterfall development in the past with statistics. In the example of NTT Laboratories, if the usage rate of features in agile-developed software increases by more than 10%, then agile development can be significant in terms of cost and duration comparing to waterfall development.

**Keywords:** agile development, productivity, Lean

### 1. はじめに

デジタルディスラプションの時代が到来し、従来の収益構造の見直しが望まれているなかで、迅速に顧客や市場に対し価値を提供し、フィードバックを得られるアジャイル開発が世界的に普及しつつある [1]。一方で、旧来から受託でウォーターフォール開発を実施してきた発注側組織のなか

<sup>1</sup> 日本電信電話株式会社 NTT ソフトウェアイノベーションセンター  
NTT Software Innovation Center, Nippon Telegraph and Telephone Corporation, Minato, Tokyo 108-0023, Japan

a) kumi.jinzenji.hz@hco.ntt.co.jp

b) akio.jin.ra@hco.ntt.co.jp

c) tatsuya.muramoto.sg@hco.ntt.co.jp

にはアジャイル開発の戦略的な導入に踏み切れない場合がある。アジャイル開発への導入障壁として以下の2つが考えられる。

- ウォータフォール型の受託開発を想定した厳格な契約の運用をしており、アジャイル開発のように要件が進化し短期間に繰り返しプロダクトをリリースすることを想定しておらず、労働者派遣法、下請法、税制などの現行法制度に十分対応するための企業内の運用や制度が未整備であること。
- アジャイル開発の生産性（以降、「開発期間」、「費用」の前提）評価法が確定していないため、他の案件やウォータフォール開発などの過去のデータとの比較ができず、プロダクトの品質やプロジェクトの能力を相対的に客観的に評価できないこと。

前者の障壁については、現行法制度を遵守するための契約形態 [2], [3] や社内運用制度設計の検討 [4] がなされてきている。また、その場合、アジャイル開発に親和性が高い準委任契約を採用 [4] する傾向がある。しかし、準委任契約は受託側に完成責任がない契約であるため、請負契約と比較して発注側のリスクが高くなる。リスクを受容してでもアジャイル開発を採用・推進するためには、アジャイル開発が請負契約によるウォータフォール開発に比較して生産性が勝っているといえる後者の障壁の克服が必要となる。

以下にアジャイル開発の生産性がウォータフォール開発と比較して劣勢になるとされる要因を3つ記す。

- 受託側にフルスタックエンジニアのスキルが要求されることによる人月単金の高止まり。
- 繰り返しリリースによる検証稼働の増加
- 発注側人材が受託側と密なコミュニケーションを必要とすることによる労務費の増加

これらのマイナスと思われる要因があるなかで、費用、開発期間の面でアジャイル開発の優位性を主張することが課題となる。

本稿ではリーンのコンセプトにおける「価値の寄与」に着目しウォータフォールやアジャイルなどの開発方法論によらず共通的に使える生産性に寄与するパラメータを定義し、過去のデータや案件間で費用と開発期間の両面から生産性を評価する指標を提案する。また、筆者らは、2018年8月以降、NTTの全研究所において実施しているアジャイル準委任契約でのトライアル [4]（以降、アジャイルトライアルと表記）10案件に対して提案指標を適用して有効性を評価した。また、実際に使われている機能の割合がどれだけ向上すればアジャイル開発がウォータフォール開発より生産性が高いといえるのかをアジャイルトライアルと過去のウォータフォール開発のそれぞれの統計値を用いて検証し、知見を得た。以降、2章では関連研究、3章では提案指標、4章ではアジャイルトライアル案件への適用結果と過去のウォータフォール開発との比較を示し、最後にまとめ

と今後の課題について述べる。

## 2. 関連研究

伝統的な生産性評価方法として、ライン数やファンクションポイント [5] といった生産量あたりの費用で評価する方法がある。これらは費用あたりの生産量が多いほど生産性は高いと評価されるものである。しかし、オブジェクト指向のプログラミング言語の普及により生産性の高さとソフトウェアの凝集性の高さが同義に語られるようになり、生産量が生産性の指標としてはふさわしくないものとなってきている。

一方、開発サイドのメトリクスとしてはWIP (Work in Progress) に代表されるように主にベロシティを計測するメトリクスが多数提案 [6], [7], [8] されているが、費用や期間にかかわる生産性を直接的に計るものではないこと、案件に閉じた指標になるため、過去も含めた他の案件との比較が困難であるという課題があった。この課題に対して、実際に開発活動をしないうステークホルダーへの品質、進捗などの報告を目的として、目標アンケートの考え方を導入したアプローチがある [9]。項目の標準化はされるが評価自体が主観的であるため、横並びの評価には適用しにくい。また、リーンにおける Value Stream Mapping によってムダなプロセスをあぶりだし、総時間に対する付加価値時間の割合を「プロセス効率」として計測する方法 [10] があるが、費用も含めた生産性は評価されていない。さらに、費用の見積もり方法については過去の文献調査 [11] が行われているが、個々の案件の費用の妥当性は主張できるが、横並びでの比較については言及されていない。また、筆者らはアジャイル開発の導入障壁を克服することを目的に、アジャイル開発のプロセスを整備し [12]、リーン [13], [14], [15], [16] の考え方を基本にしたアジャイル開発のQCDの検討を行い、マネージメントサイドへの支援を目的としたアジャイル開発の開発期間を予測する方法を提案している [17]。しかし、開発期間は総開発費用に関連する重要な要因ではあるものの、直接的な生産性評価への言及には至らなかった。

## 3. 提案指標

本提案指標の要件を下記にまとめる。

- 費用と期間の観点で生産性を評価が可能
- 複数の事案との比較評価が可能

本章では、上記の要件を満たし、リーンの考え方を利用した生産性指標を提案する。

### 3.1 リーンの考え方

リーンの原点はトヨタ生産方式 [13] にある。当時日本の市場がそれほど大きくないなかで、“どのように自動車の少量生産を行えば、大量生産並みに価格をおさえることができるのか” [14] という問題に対して、“ムダを排除せよ”

というリーンの根本が考え出された。以下にトヨタ生産方式における7つのムダ [14], [15] を列記する。

- 在庫
- 加工しすぎ
- 作りすぎ
- 運搬
- 待ち時間
- 動作
- 不良と手直し

在庫, 加工しすぎ, 作りすぎは, ソフトウェア開発においては, 使われないプロダクトもしくは機能に関するもので, 作業や費用に大きく関わるムダである。運搬, 待ち時間, 動作は時間に関するムダである。不良と手直しは作業(費用)と時間の双方にかかわるムダである。

また, Bellらは顧客の価値に着目した。顧客の価値とは“顧客が要求し支払うに値するもの”と定義されている。価値を直接もたらす作業をVA (Value Added), 価値をもたらさない作業をNVA (Not Value Added), 価値をもたらさないが必要な作業をNNVA (Necessary but Not Value Added)と定義している [16]。ただし, 個々のソフトウェア開発プロセスがVA/NNVA/NVAのどれに属するかは明確にされていない。

一方, Royce [18]は“すべてのソフトウェア開発の基本ステップは分析とコーディングである”とし, “そのほか多くの開発ステップが必要だが, そのうちの1つとして分析やコーディングほどに最終的な製品に直接寄与するものはなく, それらすべてが開発コストを引き上げている” [14]とされている。つまりソフトウェア開発プロセスのうち, 分析とコーディング以外は価値に直接寄与しない, と明言している。

こういったリーンに関する動向に鑑みて, 我々は生産性を向上させるには顧客にとっての価値を最大化しムダを極力排除するという視点に立ち, 生産性を新たに定義するアプローチをとる [19], [20]。

まず, 表1に示すように, ソフトウェア開発プロセスを検討, 構築, 検証, 管理調整, 文書作成, 環境整備, 不具合修正の7つの作業カテゴリに分類した。ソフトウェア開発は検討(分析), 構築(コーディング), 検証(試験)が基本的なプロセスである。これらのなかには管理調整や文書作成的な部分も含むが, 会議体や設計書作成, 報告書作成などのより包括的なプロセスは管理調整や文書作成として独立してカテゴリ化した。

次に表2に示すように, これらの作業カテゴリを顧客の価値に直接寄与するか否かでVA/NNVA/NVA(以降, 便宜的に“リーンフラグ”と記載)に分類した。まずRoyce [18]の考え方を踏襲し, 「検討」, 「構築」をVAとした。単体試験はコーディングと切り離せないプロセスであるので, 「構築」に分類した。よってVAに含まれる。さらに, 「不

表1 ソフトウェア開発プロセスと作業カテゴリ

Table 1 Software development process and work categories.

作業カテゴリ	ソフトウェア開発プロセス例
検討 (VC)	ソフトウェア全体もしくは特定開発機能/非機能に関する分析, 検討. 例) ユーザストーリー/受け入れ基準作成, アーキテクチャ検討, 実装方式検討, 設計など.
構築 (VC)	特定開発機能/非機能に関する構築, 試験 例) コーディング, コードレビュー, 単体試験など.
検証 (VF)	ソフトウェア全体に対する試験. 機能/非機能試験, 回帰試験, フィールド試験など.
管理調整 (MC)	より包括的, 共通的, 定例的な会議, 打ち合わせなど. 例) スプリント計画/レビュー, デイリースタラム, 共通的タスクとしての開発管理作業. 例) 進捗管理, 仕様管理, リスク管理, 問題管理など. 対外的な調整, オフラインなど.
環境整備 (EP)	開発/試験環境構築, 現場調整, 物品購入など.
文書作成 (DC)	より包括的, 共通的な文書作成 例) 設計書, 計画書, 各種報告書, マニュアルなど.
不具合修正 (DF)	不具合事象の解析, バグ修正, 不具合に伴う説明など.

表2 作業カテゴリとリーンフラグ

Table 2 Work categories and Lean flags.

リーンフラグ	作業カテゴリ
VA	検討, 構築
NNVA	検証, 管理調整, 環境整備, 文書作成
NVA	不具合修正

具合修正」をNVAと分類した。VAにもNVAにもあてはまらない「検証」, 「管理調整」, 「環境整備」, 「文書作成」についてはNNVAとした。NNVAは直接価値には貢献しないがNVAを生じさせることに対するリスク回避として有効な作業である。一方, これらのリスクマネジメントの効果を保ちながら自動化やコミュニケーションツールを利用することで作業時間をできるだけ削減し時間と費用を削減していくことができる作業カテゴリである。

なお, 作業がベースとなるので, 以降, 作業時間を生産性計測の基本的な単位とする。

### 3.2 生産性の基本的な考え方

図1に本稿で提案する生産性指標の考え方を示す。ある機能群を開発するために要した作業時間(実作業時間)を

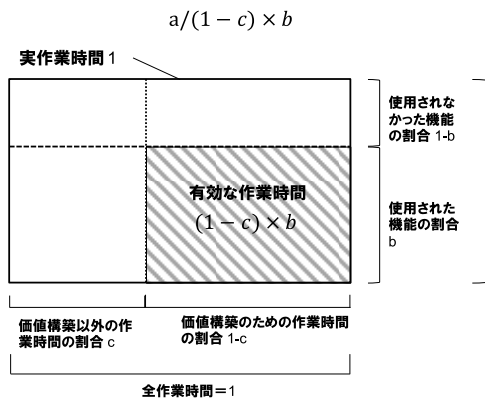


図 1 生産性指標算出の考え方

Fig. 1 Concept of productivity calculation.

1として、以降説明する。ここで、顧客に直接価値をもたらさない、もしくは、費用の面でムダ（在庫，加工しすぎ，作りすぎ）につながる2つの事象に着目する。

- 開発におけるVA以外の作業時間の割合
- リリース後に使用されなかった機能の割合

ここで、機能とは顧客の価値を代表する外部仕様項目，ユースケース，ユースストーリーなどを想定する。なお非機能要求項目は実現された機能を検証する中で確認されるものなので，機能数には含めない。図1に示すように実作業時間を1とした場合，前述の2つの観点から価値に直接寄与し，かつ実際に使われた機能に対する作業時間を有効作業時間とすると，有効作業時間は斜線部分に相当する。これは，有効な価値1を達成するためには前述の有効作業時間の逆数倍の作業時間（期間）が必要となる。さらに，作業時間あたりの費用（単金）を乗算することで有効な価値1を達成するための費用を算出できる。

ここで，生産性とはチームの生み出した結果から推測されるチームの能力であると考え，本稿ではソフトウェア開発の生産性を「一定の価値1を達成するために必要な期間および費用」と定義する。一定の価値を少ない費用で達成できれば生産性は高く，一定の価値を短い期間で達成できれば生産性は高いという考え方である。この生産性の考え方は，使用された機能の“割合”や価値を作るための作業時間の“割合”を用いることで，実際の作業時間や機能の粒度，個数に依存しないという特長を持つ。

### 3.3 4つの説明パラメータ

生産性にプラスおよびマイナスに働く観点を数値化したものを本生産性指標算出時の説明パラメータとする。以下に説明パラメータ a-d を説明する。

#### a 人月単金

受託側に支払われる月あたりの費用。基本となる案件の単金に対する単金比を用いてもよい。

#### b 機能使用率

実現した機能のうち実際に使われる割合である。ウォー

タフォール型開発では64%の機能が使われていない[21]，という報告がされているが，この場合の機能使用率は0.36である。アジャイル開発についてはMVP (Minimum Viable Product) を積み重ねるため，1.0が理想であるが，最終的にプロダクトとしてソフトウェアをリリースするときに機能の取舍選択が行われる場合もあるため，機能使用率は1.0よりも若干小さくなることが予想される。

#### c NVA-NNVA 率

価値の構築に寄与していない作業時間の割合である。ラインにおけるムダな作業の割合ともいえる。NVA-NNVA率は全体の作業時間に対するNVAおよびNNVAの作業時間の割合である。表1で示すように，NVAの例としてはフィードバック先（顧客）から指摘されたバグ修正に関連する作業時間がこれにあたる。ウォータフォール開発の場合，開発終了後に顧客にソフトウェアがリリースされる。よって，開発期間中の顧客からのフィードバックはないとしてNVAの作業量を0とする。また，NNVAの例としては検証や環境整備などがこれにあたる。文献[10]のプロセス効率率は1.0からNVA-NNVA率を引き算したものと考えることができる。

#### d 内製率

本説明パラメータは内製作業時間と外注作業時間の両方がある場合に利用できる。全費用における内製分費用の割合である。内製のみの場合は100% (= 1.0) となる。

これら4つの説明パラメータはウォータフォール開発やアジャイル開発などの開発方法論によらずに算出可能で，単独でも観点別に横並びの比較評価が可能である。また，説明パラメータ a, c, d は序章で述べたアジャイル開発がウォータフォール開発と比較して生産性面で劣勢に働くと考えられる要因である。

### 3.4 生産性指標（期間指標と費用指標）

ソフトウェア開発の生産性を評価するための期間指標  $I_{delivery}$ ，費用指標  $I_{cost}$ ，総費用指標  $I_{total\_cost}$  を下記式(1)–(3)に定義する。考え方は3.1節に記載したものにとり，これらの式には3.2節で紹介した単金 a，機能使用率 b，NVA-NNVA率 c，内製率 d の4つのパラメータを用いる。

$$I_{delivery} = \frac{1}{(1-c) \times b} \quad (1)$$

$$I_{cost} = I_{delivery} \times a = \frac{a}{(1-c) \times b} \quad (2)$$

$$I_{total\_cost} = \frac{I_{cost}}{1-d} = \frac{a}{(1-d) \times (1-c) \times b} \quad (3)$$

式(1)で記述される期間指標  $I_{delivery}$  は，一定の作業時間において，使われないものを作りこむ確率や価値を生まない作業を実施する確率を除去したうえで，価値1が構築される期間を定義したものである。式(2)で記述される

費用指標  $I_{cost}$  は、期間指標に単金を乗算したものである。また、外注する場合の総費用指標  $I_{total\_cost}$  は内製比率  $d$  と費用指標  $I_{cost}$  から求めることができる。これらは同一の観点および横並びの比較が可能な 4 つパラメータの乗算によって算出されるので、案件間での評価が可能である。これら 3 つの指標をまとめて生産性指標と呼ぶ。

一般的に生産性は入力に対して出力の度合いで定義される。入力を作業時間  $e$ 、出力を機能  $f$  とする。理想的な生産性  $p$  は

$$p = f/e \quad (4)$$

と書ける。ここで、入力に対して  $1 - c$  の効率なので、実際に  $f$  の出力を得るには、 $e'$  の入力が必要である。

$$e' = e/(1 - c) \quad (5)$$

出力に対して  $b$  の効率なので、実際に使用される機能は  $f'$  である。

$$f' = bf \quad (6)$$

よって実質的な生産性は  $p'$  となる。

$$p' = \frac{f'}{e'} = \frac{bf(1 - c)}{e} = p(1 - c)b \quad (7)$$

ここで、実質的な生産性に対する理想的な生産性の比は  $p/p'$  となる。

$$p/p' = \frac{1}{(1 - c)b} \quad (8)$$

これは式 (1) の定義に他ならない。

### 3.5 案件間評価のための生産性指標比

案件間での評価には、式 (9)–(11) で示すように、期間指標比  $R_{Delivery}^k$ 、費用指標比  $R_{cost}^k$ 、総費用指標比  $R_{total\_cost}^k$  を用いる。

$$R_{delivery}^k = I_{delivery}^k / I_{delivery}^{ref} \quad (9)$$

$$R_{cost}^k = I_{cost}^k / I_{cost}^{ref} \quad (10)$$

$$R_{total\_cost}^k = I_{total\_cost}^k / I_{total\_cost}^{ref} \quad (11)$$

これらの指標比は、基準となる案件に対する期間指標  $I_{delivery}$ 、費用指標  $I_{cost}$ 、総費用指標  $I_{total\_cost}$  の比である。それぞれ式 (1)–(3) で算出された指標を基準案件の値で除算することで算出される。値が 1.0 を下回るほど基準値と比較して生産性が良い、すなわちより少ない費用、期間で価値を達成できると考える。

## 4. 検証実験

本章では提案した生産性指標を用い、アジャイルトライアル案件 10 件における生産性評価と、過去 4 年間の 200 件以上の請負契約におけるウォーターフォール開発の統計データを用いたアジャイル開発との比較評価を行う。

### 4.1 データ収集方法

アジャイルトライアル案件は、代表的な開発手法であるスクラムで実施され、スプリント（短い時間で区切られた開発の単位）ごとの受諾側へ清算のために作業終了報告書 [4] に記載されるランクごとの人月単金、作業カテゴリに紐づいた作業時間を収集した。

また、過去のウォーターフォール開発案件については、従来から使用している生産性評価のための帳票に記載された大まかな工程ごとの作業時間と総額の情報を収集した。大まかな工程とは、本稿で定義した検討、構築、検証、環境整備の 4 種類である。

どちらも人月単金  $a$  は、総外注費用を総外注作業時間で除して算出した。なお、以降の議論では作業時間、機能数、金額などの実際のデータの提示は割愛するが、本稿の指標はこういった絶対的なデータには依存しないので評価の妥当性には影響ない。

### 4.2 前提条件

アジャイルトライアル案件、ウォーターフォール案件ともドメインはネットワークシステム開発からツール作成まで幅広い。

アジャイルトライアル案件はすべて準委任契約での受託開発である。2019 年 6 月末時点でのデータを使用して評価した。スプリント期間は 2~4 週間である。各案件ではリーンの考え方にに基づき各作業時間を作業カテゴリに紐付けて計測した。過去のウォーターフォール開発案件は検討、構築、検証および環境整備が作業時間として別計されているのでそれぞれの作業カテゴリの作業時間として計上した。

なお、NVA-NNVA 率の算出には、アジャイルトライアル案件では NVA および NNVA 作業として「バグ修正」、「検証」、「環境整備」の作業カテゴリ、ウォーターフォール開発では「検証」、「環境整備」の作業カテゴリの作業時間を用いた。これは、ウォーターフォール開発では「コミュニケーション」および「文書作成」が分計されていなかったことによる。

また、機能使用率  $b$  は未計測であるために便宜上  $b = 1.0$  とした。アジャイル開発は顧客の要望を優先順位に従ってソフトウェアとして逐次実現する開発なので、機能使用比率  $b$  はどのプロジェクトも 1.0 に近いと考えられる。よって、以降の生産性評価にはほとんど影響を与えないと考える（なお、4.3 節では機能比率  $b$  の比を可変パラメータとしてシミュレーションを実施）。

アジャイル開発とウォーターフォール開発の比較には統計値を用いた。前者は全案件の平均値、後者は案件数が多いため中央値を使用した。

### 4.3 アジャイルトライアル案件間の生産性比較

全 10 案件に対し本稿で定義した説明パラメータを計測

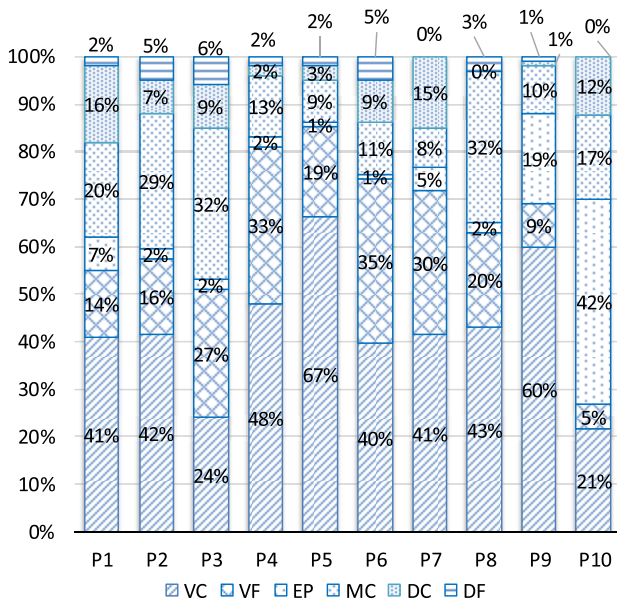


図 2 アジャイルトライアル案件の作業カテゴリごとの作業時間の割合

Fig. 2 Distribution of work effort by each work category in Agile trial projects.

表 3 説明パラメータ (アジャイルトライアル)

Table 3 Description parameters (Agile trials).

Projects	Description parameters		
	a	c	d
P1	1.28	0.59	0.14
P2	1.19	0.58	0.03
P3	1.18	0.76	0.02
P4	1.07	0.52	0.14
P5	1.01	0.33	0.14
P6	1.44	0.60	0.09
P7	1.06	0.59	0.13
P8	1.36	0.57	0.12
P9	1.25	0.40	0.14
P10	1.21	0.79	0.20

し、期間指標、費用指標および総費用指標を算出した。図 2 にアジャイルトライアル案件 10 件における作業カテゴリごとの作業時間の分布を示す。図中の VC などの記号は表 1 における作業カテゴリを示す。なお、参考に付録 A-1 に図 2 の元データを示す。図 2 よりプロジェクトが作業時間の  $R_{cost}^k$  面で何に注力したかが分かる。表 3 に各案件の機能比率  $b$  を除く 3 つの説明パラメータを示す。

また、図 3、図 4、図 5 に P1 を基準とした場合の期間指標比  $R_{delivery}^k$ 、費用指標比  $R_{cost}^k$ 、総費用指標比  $R_{total\_cost}^k$  を示す (付録 A-2 に期間指標、費用指標、総費用指標の算出結果を示す)。表 3 の説明パラメータ  $c$  は図 2 の作業の内訳を反映しており、また、図 3 の期間指標 (比) から  $c$  の値が小さい (ムダが少ない) ほど生産性が高いことが分

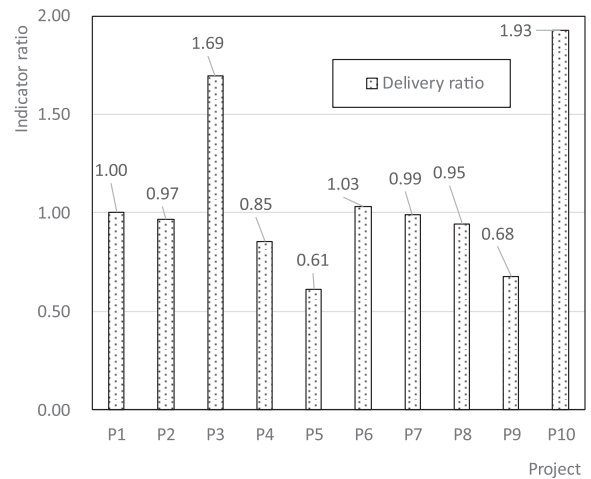


図 3 P1 を基準とした場合の期間指標比  $R_{Delivery}^k$   
Fig. 3 Delivery ratio as reference is P1.

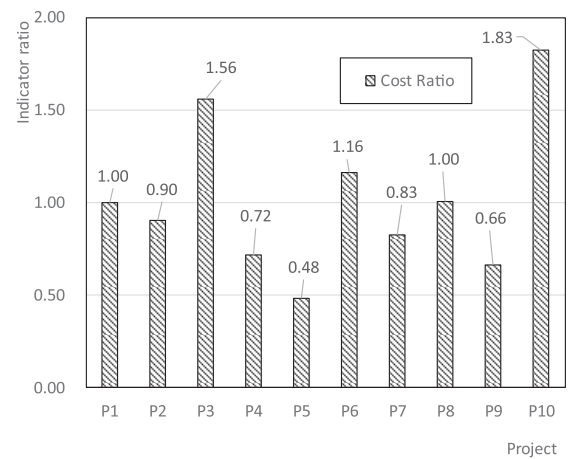


図 4 P1 を基準とした場合の費用指標比  $R_{cost}^k$   
Fig. 4 Cost ratio as reference is P1.

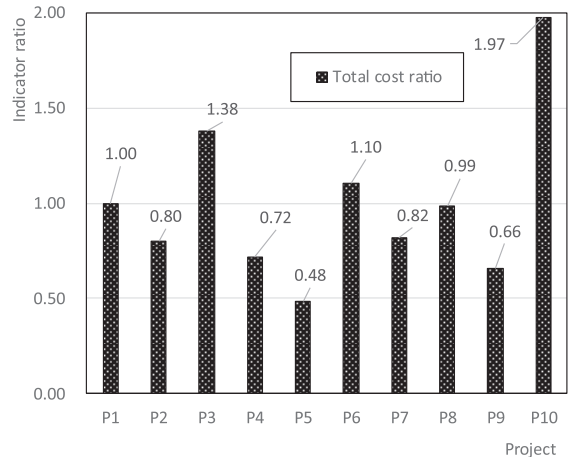


図 5 P1 を基準とした場合の総費用指標比  $R_{total\_cost}^k$   
Fig. 5 Total cost ratio as reference is P1.

かる。すなわち、指標はプロジェクトの状態をよく表している。

図 3~図 5 から、基準となる案件と比較して期間、費用の評価が可能であることが分かる。ここで P1 と P10 に着

表 4 説明パラメータ (AGL/WF 比較)

Table 4 Description parameters in comparison with agile and waterfall projects.

Project	Description parameters		
	a	c	d
AGL	1.30	0.35	0.14
WF	1.00	0.45	0.15

表 5 WF を基準とした場合の AGL の各生産性指標

Table 5 Productivity indicators of agile projects as reference is waterfall project.

Ratio of productivity indicators		
$R_{delivery}^{agl}$	$R_{cost}^{agl}$	$R_{total\_cost}^{agl}$
0.85	1.10	1.09

目すると、P10 は表 2 より単金 a は同程度であるにもかかわらず、図 3～図 5 からは期間指標比  $R_{delivery}^k$ 、費用指標比  $R_{cost}^k$ 、総費用指標比  $R_{total\_cost}^k$  は 2 倍近い値となった。表 3 より、価値に貢献していない作業時間の割合を示すパラメータ c が P10 は P1 の約 34% 増しであることが分かる。その理由として図 2 より P10 は環境整備 (EP) に作業時間の 42% を使っていることなどが考えられる。このように、P10 は P1 と比較して、単金は同程度だが、同じ価値を達成するのに約 2 倍の期間、約 2 倍の費用がかかるプロジェクトだということが分かる。

4.4 過去のウォーターフォール案件との生産性比較

過去のウォーターフォール開発の案件とアジャイルトリアル案件の生産性の比較を行う。アジャイルトリアル 10 案件の平均値とウォーターフォール案件の過去 4 年間の 200 件以上の統計値 (中央値) と比較した。

表 4 にアジャイルトリアル (AGL) 案件およびウォーターフォール (WF) 開発の機能使用率 b を除く 3 つのパラメータ、表 5 におよび費用指標、期間指標、総費用指標を示す (付録 A-3 に作業カテゴリごとの作業時間内訳、付録 A-4 に表 5 の根拠となる期間指標、費用指標、総費用指標の算出結果を示す)。

表 4 からアジャイル開発の人月単金はウォーターフォール開発の 1.3 倍であることが分かる。アジャイル開発はウォーターフォール開発と比較して検証頻度が高いので、より高い NVA-NNVA 率 c を示すと予想されたが、両者にて極端な差異は見られなかった。内製率 d についてもアジャイル開発の方が高くなることが予想されたが、ほぼ同じ割合となった。また、表 5 からアジャイルトリアル案件はウォーターフォール開発案件に対して同じ価値を達成するのに、0.85 倍の期間、1.10 倍の外注費用、1.09 倍の総費用がかかることが分かる。

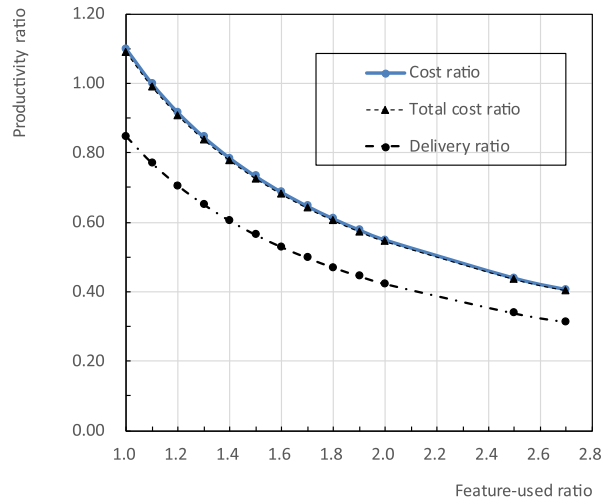


図 6 機能使用比率  $b_{agl}/b_{wf}$  を用いたアジャイル開発とウォーターフォール開発の生産性シミュレーション

Fig. 6 Productivity simulation of agile and waterfall projects using feature-used ratio  $b_{agl}/b_{wf}$ .

4.5 機能使用比率比を用いた生産性シミュレーション

式 (9)–(11) における期間指標、費用指標、総費用指標の比を期間指標比、費用指標比、総費用指標比とすると、ウォーターフォール開発に対するアジャイル開発の指標比  $R_{delivery}^{agl}$ 、 $R_{cost}^{agl}$ 、 $R_{total\_cost}^{agl}$  は下記に示す式 (12)–(14) で記すことができる。

$$\begin{aligned}
 R_{delivery}^{agl} &= I_{delivery}^{agl} / I_{delivery}^{wf} \\
 &= \frac{1 - c_{wf}}{1 - c_{agl}} \times \frac{b_{wf}}{b_{agl}} \\
 &\equiv f_{delivery} \left( \frac{b_{agl}}{b_{wf}} \right)
 \end{aligned} \tag{12}$$

$$\begin{aligned}
 R_{cost}^{agl} &= I_{cost}^{agl} / I_{cost}^{wf} \\
 &= \frac{a_{agl}}{a_{wf}} \times \frac{1 - c_{wf}}{1 - c_{agl}} \times \frac{b_{wf}}{b_{agl}} \\
 &\equiv f_{cost} \left( \frac{b_{agl}}{b_{wf}} \right)
 \end{aligned} \tag{13}$$

$$\begin{aligned}
 R_{total\_cost}^{agl} &= I_{total\_cost}^{agl} / I_{total\_cost}^{wf} \\
 &= \frac{1 - d_{agl}}{1 - d_{wf}} \times \frac{a_{agl}}{a_{wf}} \times \frac{1 - c_{wf}}{1 - c_{agl}} \times \frac{b_{wf}}{b_{agl}} \\
 &\equiv f_{total\_cost} \left( \frac{b_{agl}}{b_{wf}} \right)
 \end{aligned} \tag{14}$$

式 (11)–(14) において、各指標比はそれぞれ機能使用率の比である  $b_{agl}/b_{wf}$  (便宜上、機能使用比率比と呼ぶ) の関数として表現できる。

ここで、機能使用比率比  $b_{agl}/b_{wf}$  を用いたシミュレーションを行う。アジャイルトリアル案件と過去事案のウォーターフォール開発案件の機能使用比率  $b_{agl}/b_{wf}$  比を 1.0~2.7 まで変化させて期間指標比  $R_{delivery}^{agl}$ 、費用指標比  $R_{cost}^{agl}$ 、総費用指標比  $R_{total\_cost}^{agl}$  を算出した様子を図 6 に示す。

機能使用比率比  $b_{agl}/b_{wf} = 1.0$  とはアジャイルトリア

ルもウォータフォール開発も同程度の機能使用率の場合で、表4に示す結果に他ならない。これは、同じ価値をアジャイル開発にて実現するのにウォータフォール開発の85%の期間、110%の費用で達成できることを示す。

機能使用比率比  $b_{agl}/b_{wfl} = 2.7$  とはアジャイル開発が100%、ウォータフォール開発が36%の機能使用率 [21] の場合に各々相当する。この場合、同じ価値を構築するために、アジャイル開発がウォータフォール開発に比較して25%の期間、40%の費用で達成可能であることを示す。

さらに、費用について着目すると、機能使用比率比  $b_{agl}/b_{wfl}$  が1.1のとき、生産性比がほぼ1.0となり、機能使用比率比  $b_{agl}/b_{wfl}$  が1.0よりも大きければ、生産性比は1.0より小さくなることが分かる。これは、アジャイル開発の機能使用比率がウォータフォール開発より10%以上改善されればアジャイル開発が費用面でウォータフォール開発の生産性を上回ることを示している。

一連の結果から、アジャイル開発の単金はウォータフォール開発の単金の1.3倍ではあるが、アジャイル開発は機能指標比率を10%改善できれば、ウォータフォール開発に対して生産性の面で有意な開発といえる。アジャイル開発はMVPを積み重ねていく開発手法であるので、ムダな機能を作りこむ可能性が高いウォータフォール開発 [21] からアジャイル開発に開発手法を切り替えるだけで機能使用比率の10%改善は容易に達成可能と予想される。

#### 4.6 考察

10件のアジャイルトライアルデータを通じ、リーンを利用した生産性指標によって期間と費用の観点から案件横並びで生産性を評価できることを確認した。また、単金、NVA-NNVA率、内製率など案件の特性がどのように生産性に寄与しているかの説明も可能であることを確認した。また、過去のウォータフォール開発の統計値と比較することで、10%以上機能使用率が改善されればアジャイル開発が費用と期間の両面でウォータフォール開発に比べて優位であることが示された。アジャイル開発はMVPを積み重ねる開発であるため、ウォータフォール開発に対してムダが少ないと予測されるため、10%の機能使用率の改善は容易であることが予想され、アジャイル開発がウォータフォール開発に対して生産性面で有意である可能性が示された。

一方で、今回の指標の前提は、生産性を「価値1を達成するための費用指標」、「価値1を達成するための期間指標」を定義し、「1人月で達成できる架空の理想的な作業によってもたらされる価値」を“1”とする前提の基に成り立っている。リーンの考え方にに基づき、開発におけるムダを考慮した指標である。これに加えて、今後は新たな概念であるプロジェクトの難易度、開発の結果としての収益との関連性の考慮が課題である。

## 5. まとめと今後の課題

本稿では、凝集性が生産性と結び付く中で従来型の従量制の生産性評価が意味をなさなくなっていることへの解決策として価値への寄与に着目し、人月単金、ムダな作業、使われない機能などの評価観点を入れ込み、費用と期間で評価できる新たな生産性指標を提案した。アジャイルトライアルの10案件に対して提案指標を適用し、案件間での生産性の比較が可能であることを確認した。また、単金、NVA-NNVA率、内製率など組織の特性に鑑みた評価が可能となると同時に、過去のウォータフォール開発の統計値と比較することで、アジャイル開発のポテンシャルを数値的に説明ができた。

本稿での知見はあくまで筆者らの組織のアジャイルトライアルおよびウォータフォール開発の過去のデータから得られた結果ではあり、一般論ではない。しかし、アジャイルトライアルの結果、アジャイル開発とウォータフォール開発の生産性の比較と根拠を客観的に主張できたため、筆者らの所属組織はアジャイル開発をトライアルから本格運用に移行する判断をすることにつながった。本研究の目的である、アジャイル開発を導入するための障壁の1つであった、生産性を客観的に説明できない点は、本指標を用いることで解決できたといえる。

今後は、プロジェクトの難易度、収益との関連性を指標に取り入れる必要があると考える。難易度の高い案件、収益に貢献する案件は高い単金、長期間にわたる開発によって価値が実現されたとしても生産性が悪いと評価されることが重要である。

**謝辞** アジャイルトライアルへ参加いただいたNTT研究所員の皆さま、受託会社としてアジャイルトライアルに参加いただいたNTTテクノクロス株式会社、NTTアドバンステクノロジー株式会社、およびNTT-ATテクノコミュニケーションズ株式会社の皆さま、シミュレーションにご協力いただいた株式会社サイバー創研の皆さま、アジャイル準委任契約の検討を実施したNTT研究企画部門、総務部門法務担当、情報ネットワーク総合研究所企画、知的財産センタ、契約センタの皆さまへ深く感謝申し上げます。

#### 参考文献

- [1] available from (<https://www.solinea.com/blog/gartners-mode-1-mode-2-dangerous-thinking>)
- [2] 独立行政法人情報処理推進機構：アジャイル開発外部委託モデル契約 (2020).
- [3] 情報処理学会情報処理に関する法的問題 (LIP) 研究グループ：アジャイル開発向けソフトウェア開発委託契約書 (準委任型) (2020).
- [4] 秦泉寺久美, 神 明夫, 夏川勝行：アジャイル開発のための準委任契約制度設計と課題, 情報処理学会電子化知的財産社会基盤 (EIP) 研究会研究報告, Vol.2019-EIP-85, No.13 (2019).



[5] available from (<http://www.ifpug.org/>)

[6] Davis, C.: *Agile metrics in action*, Manning (2015).

[7] Budacu, E. and Pocatilu, P.: Real time agile metrics for measuring team performance, *Informatica Economica*, Vol.22, No.4, (2018).

[8] Destefanis, G., Counsell, S., Conca, G. and Tonelli, R.: Software metrics in agile software: An Empirical study, *Business Information Processing* (May 2014).

[9] Boerman, M.P., Lubsen, Z., Tamburri, D.A. and Visser, J.: Measuring and Monitoring Agile Development Status, *Proc. 2015 IEEE/ACM 6th International Workshop on Emerging Trends in Software Metrics*, pp.54–62 (2015).

[10] Verbruggen, F., Sutherland, J., van der Werf, J.M., Brinkkemper, S. and Sutherland, A.: Process Efficiency – Adapting Flow to the Agile Improvement Effort, *HICSS 2019*, Computer Science Published (2019).

[11] Mansor, Z., Yahya, S., Habibah, N. and Arshad, H.: Review on Traditional and Agile Cost Estimation Success Factor in Software Development Project, *International Journal on New Computer Architectures and Their Applications (IJNCAA)*, Vol.1, No.3, pp.965–975 (2011).

[12] Jinzenji, K., Williams, L., Hoshino, T. and Takahashi, K.: An experience report for software quality evaluation in highly iterative development methodology using traditional metrics, *IEEE International Symposium on Software Reliability Engineering Symposium (ISSRE 2013)*, pp.310–319 (2013).

[13] 大野耐一：トヨタ生産方式—脱規模の経営目指して，ダイヤモンド社 (1978).

[14] Poppendieck, M. and Poppendieck, T.: *Lean Software Development*, Addison-Wesley Professional (2003).

[15] 新郷重夫：ノン・ストック生産方式への展開—トヨタ生産システムの真の意義，日本能率協会 (1987).

[16] Bell, S.C. and Orzen, M.A.: *Lean IT*, CRC Press (2010).

[17] Jinzenji, K. and Hamuro, D.: A concept of QCD prediction and control in agile software development for portfolio management, *IEEE International Symposium on Software Reliability Engineering Symposium (ISSRE 2018)*, pp.48–52 (2018).

[18] Royce, W.W.: Managing the Development of Large Software Systems, *Proc. IEEE WESCON* (Aug. 1970).

[19] 秦泉寺久美，神明夫，夏川勝行：VSMを用いたアジャイル開発の生産性指標の提案とウォーターフォール開発との比較，情報処理学会ソフトウェア工学研究会 (SE) (2019).

[20] Jinzenji, K., Jin, A. and Muramoto, T.: Productivity evaluation indicators based on Lean and their application to compare agile and waterfall, *IEEE 44th Annual Computers, Softwares, and Applications Conference (COMPSAC2020)*, pp.444–450 (2020).

[21] Standish Group: *The CHAOS Report 2005* (2005).

## 付 録

### A.1 作業カテゴリごとの作業時間内訳 (アジャイルトリアル)

Project	Work categories					
	VC	VF	EP	MC	DC	DF
P1	0.41	0.14	0.07	0.20	0.16	0.02
P2	0.42	0.16	0.02	0.29	0.07	0.05
P3	0.24	0.27	0.02	0.32	0.09	0.06
P4	0.48	0.33	0.02	0.13	0.02	0.02
P5	0.67	0.19	0.01	0.09	0.03	0.02
P6	0.40	0.35	0.01	0.11	0.09	0.05
P7	0.41	0.30	0.05	0.08	0.15	0.00
P8	0.43	0.20	0.02	0.32	0.00	0.03
P9	0.60	0.09	0.19	0.10	0.01	0.01
P10	0.21	0.05	0.42	0.17	0.12	0.03

### A.2 生産性指標 (アジャイルトリアル)

Project	Productivity indicators		
	$I_{delivery}$	$I_{cost}$	$I_{total\_cost}$
P1	2.45	3.13	3.63
P2	2.37	2.82	2.90
P3	4.15	4.88	5.00
P4	2.09	2.24	2.60
P5	1.50	1.51	1.75
P6	2.53	3.63	4.00
P7	2.43	2.58	2.98
P8	2.31	3.15	3.58
P9	1.66	2.07	2.39
P10	4.72	5.71	7.16

### A.3 作業カテゴリごとの作業時間内訳 (AGL/WF 比較)

Project	Work categories					
	VC	VF	EP	MC	DC	DF
AGL	0.44	0.21	0.09	0.16	0.08	0.03
WF	0.55	0.35	0.10	---	---	0.0

### A.4 生産性指標 (SGL/WF 比較)

Project	Productivity indicators		
	$I_{delivery}$	$I_{cost}$	$I_{total\_cost}$
AGL	1.54	2.00	2.32
WF	1.82	1.82	2.13



秦泉寺 久美 (正会員)

1991年早稲田大学大学院理工学研究科修士課程修了。同年、日本電信電話株式会社 (NTT) 入社。以降、画像処理、符号化、通信の研究開発に従事。NTT コミュニケーションズでの大規模ネットワークシステム構築・維持管理を経て、現在は、ソフトウェア開発標準策定・運用、ソフトウェア工学研究に従事。博士 (国際情報通信学)。電子情報通信学会会員。



神 明夫

1994年名古屋大学大学院工学研究科博士課程前期課程修了。2008年筑波大学博士 (工学)。1994年よりNTTにてオーディオ符号化・音声信号処理の研究に従事。2003年以降NTT コミュニケーションズ, NTT レゾナント, NTT データの各社にてシステム開発等を担当。その後現職にてソフトウェア開発方式の研究等に従事。1997年日経BP技術賞, 1999年電気通信普及財団賞受賞。



村本 達也

1996年筑波大学社会工学研究科修士課程修了。DBMSのコア技術研究, アプリケーションサービス開発業務を経て、現在はソフトウェア開発工学研究に従事。NTT ソフトウェアイノベーションセンター主幹研究員。