

ウォーターフォール型ソフトウェア開発における 工程の重なりと利益率の関係

The relationship between process overlap and profitability on waterfall software development

日本電気株式会社 ソフトウェアエンジニアリング本部

NEC Corporation, Software Engineering Division

○上野 拓也 倉下 亮¹⁾ 三橋 二彩子¹⁾

谷口 修¹⁾ 齊藤 拓也¹⁾ 野中 誠²⁾

○Takuya Ueno Ryo Kurashita¹⁾ Fusako Mitsuhashi¹⁾

Osamu Taniguchi¹⁾ Takuya Saitou¹⁾ Makoto Nonaka²⁾

Abstract Overlapping processes in waterfall software development has both advantages and disadvantages. This study investigates the relationship between process overlap and profit margin based on 74 project data. The results indicate that excessive overlap decreases the chance of improving profitability. The study also shows the overlap between basic design and functional design processes has a positive relationship on the profit margin of a project, while those between other processes have a negative relationship on it. Improving the defect detection rate at upper processes also increases the profit margin. A decision tree model that explains the profit margin is also derived by using both process overlap measures and software quality measures. The results help project managers make a reasonably overlapped schedule.

1. はじめに

ウォーターフォール型ソフトウェア開発は前工程を終えたのちに後工程を開始し、後戻りすることなく逐次的に進めるものとして一般に認識されている^{[1][2]}。しかし、ウォーターフォールモデルにおいて隣り合う 2 工程の間には双方向の関係があり^[3]、この関係のため、それらの工程は部分的に並行して実施され得る^[4]。筆者らの経験においても、ウォーターフォール型ソフトウェア開発の日程計画において工程間に重なりを持たせることはしばしば行われている。ウォーターフォール型ソフトウェア開発は工程が逐次的に進められることが基本ではあるものの、工程間に重なりを持たせた日程計画を立てることは一般的なことであるといえる。

工程間に重なりを持たせることにはメリットとデメリットの両面がある。工程を重ねることで作業を並行して実施することができ、開発期間を短縮できたり^[5]、次工程の作業者の手持ち時間を

日本電気株式会社 ソフトウェアエンジニアリング本部
Software Engineering Division, NEC Corporation

東京都港区芝 4-14-1 Tel: 03-3798-9552 e-mail:t-ueno-pv@nec.com
4-14-1, Shiba, Minato-ku, Tokyo Japan

1) 日本電気株式会社 ソフトウェアエンジニアリング本部
Software Engineering Division, NEC Corporation

2) 東洋大学 経営学部
Faculty of Business Administration, Toyo University

【キーワード：】 ウォーターフォール型ソフトウェア開発, 工程の重なり, 利益率, 日程計画

減らせたりできるなどのメリットが得られる。これにより、プロジェクトの利益率向上が期待される。一方で、過度に工程を重ねてしまうと、前工程で生じた変更が次工程またはそれ以降の工程での後戻り作業を増やすことになり、計画時に比べて品質やコストに悪影響を及ぼしてしまう^[5]。これにより、プロジェクトの利益率が悪化するリスクが高くなってしまう。

工程間の重なりに関する検討は、これまでシミュレーション研究として扱われたり^{[5][6]}、並行ソフトウェア開発^{[7][8]}などのプロセスモデルとして論じられたりしてきた。しかし、ウォーターフォール型ソフトウェア開発における工程の重なりがプロジェクトの利益率にどのような影響を与えるのかについての実証的な取り組みは、筆者らが知る限り十分に示されていない。アジャイル開発が広がりを見せている一方で、ウォーターフォール型ソフトウェア開発は依然として広く利用されており^[9]、それを適用することが望ましい場合や^{[10][11]}、顧客都合などの理由によりそれを適用せざるを得ない場合もある。ウォーターフォール型ソフトウェア開発における工程間の重なりと利益率の関係に関する知見を実証的に導き出し、開発現場へと適用することは意義ある取り組みであると筆者らは考えている。

本論文では、ウォーターフォール型ソフトウェア開発プロジェクトを対象に収集したデータを基に、工程の重なりと利益率との関係について分析を行う。工程の重なりにはメリットとデメリットの両面があり、重なり具合は利益率に何らかの関係がある可能性がある。また、工程を個別に見たときに、どの工程とどの工程の重なりが利益率と関わりがあるのかについて掘り下げて検討する必要がある。さらに、品質メトリクスを含めて総合的に捉えたときに、利益率に関わる要因同士の構造が明らかになれば、プロジェクトの日程計画を立案する上で有益な知見が得られるものと期待できる。そのため、本研究では次の **Research Questions** を設定する。

RQ1：プロジェクト全体で見た工程の重なりと利益率との間には、どのような関係があるのか。

RQ2：工程の重なりを個別に見たときに、利益率と関わりのある工程はどれか。

RQ3：工程の重なりと品質メトリクスを用いて、利益率を説明することはできるか。

2 章では分析対象について述べたのちに、分析に用いた変数である工程の重なり、利益率予実差、および品質メトリクスについて、それぞれの概要とデータの概要を示す。3 章では RQ1 から RQ3 についての分析結果を示す。4 章で考察を述べ、最後に 5 章でまとめを述べる。

2. 分析方法

2.1 分析対象の概要

筆者らの所属企業内において、システムインテグレーション事業を手がける複数の事業部が 2018 年度に実施したウォーターフォール型ソフトウェア開発プロジェクトのデータを収集した。得られたデータのうち、開発規模が一定規模以下の改修プロジェクトで、出荷後の品質基準を満たしたものを対象とした。このような抽出条件とした理由は、これまでに筆者らが行ってきた品質データ分析^{[12][13]}において、妥当な結果が得られることが経験的に把握できているためである。データを精査し、本論文では 74 件のプロジェクトを分析対象とする。これらのプロジェクトの計画期間の平均値は 219.7 日、標準偏差は 83.80 日であった。IPA/SEC の『ソフトウェア開発データ白書』^[14]における改修・保守プロジェクトの工期の平均値が 7.4 カ月、標準偏差が 4.6 カ月であることを踏まえると、分析対象プロジェクトの計画期間は典型的なものであったといえる。

図 1 に示したウォーターフォール型開発プロセスのそれぞれの工程について、計画開始日と計画終了日を収集した。FD 工程から ST 工程までは欠測値はなかったが、BD 工程は欠測値比率が 58.7%と高かった。そのため、以降の分析での結果の解釈において注意が必要である。

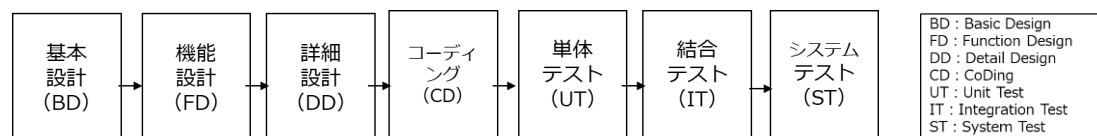


図1 ウォーターフォール型ソフトウェア開発プロセス

2.2 工程の重なり

工程 A とこれに後続する工程 B について、工程 A と工程 B の重なりを、工程 A の開始から工程 B の終了までの期間で割った値を「工程の重なり」と定義する。ただし、工程 A の終了日が工程 B の開始日より前であった場合は工程の重なりを 0 とし、負の値とならないようにする。

具体例を図 2 で説明する。先行する工程 A とこれに後続する工程 B が、図 2 のように重なりのある日程計画がそれぞれ立てられていたとする。このとき、工程 A と工程 B の重なり期間は①であり、工程 A の開始時点から工程 B の終了までの期間は②である。したがって、これらの工程の重なりは①÷②となる。そして、工程 A の終了日が工程 B の開始日よりも前の場合は①

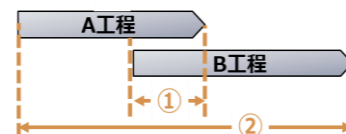


図 2 工程の重なり

が 0 となり（重なり 0）、工程 A と工程 B の開始日と終了日が同じ場合は①と②が等しくなる（重なり 1）。なお、工程 A の終了日が工程 B の終了日よりも後になる場合や、工程 A の開始日が工程 B の開始日よりも後になるパターンも論理的には想定されるが、そのような日程計画はこの文脈において現実的ではないため、考慮の対象外とする。

分析対象の 74 プロジェクトについて、工程の重なり平均値の分布を図 3 に示す。ここで「重なり平均」とは、あるプロジェクトについて、BD と FD の重なり、FD と DD の重なり、DD と CD の重なりのように隣り合う 2 工程の重なりをそれぞれ求め、それらの平均を求めた値である。

「重なり平均」の平均値は 0.308、中央値は 0.272 であった。また、図 3 に示した通り、重なり平均の平均が 0.0 から 0.1 の階級の度数がもっとも多い。なお、2.1 節で述べた通り、この値はそれぞれの工程の計画開始日および計画終了日に基づいている。したがって、工程の重なった期間において、それぞれの工程でつねに稼働が生じていたとは限らない点に留意する必要がある。

2.3 利益率予実差

プロジェクトを開始する前には、利益率の計画値を見積もる。そして、プロジェクトが完了すると、利益率の実績値が確定する。利益率の実績値から利益率の計画値を引くと、利益率予実差が求まる。本論文では、利益率予実差を用いて工程の重なりとの関係を分析する。ここで、利益率の計画値や実績値そのものを用いてない理由は、これらは組織の事業戦略や顧客との関係性などによって大きく左右されるためである。利益率予実差を用いることにより、利益率の計画に対する実績の乖離度合いに着目して評価することができる。

分析対象の 74 プロジェクトについて、利益率予実差を正規化した値の分布を図 4 に示す。ここで、「正規化した値」とは、それぞれのプロジェクトの予実差からその平均値を引いたものを標準偏差で割った値を意味する。したがって、正規化利益率予実差が負の値であることは、予実差の平均値を下回っていることを意味しており、予実差が負であったことを意味しているわけではない。なお、利益率の計画値と実績値、ならびに予実差の実測値は非公開とする。

以降では、利益率予実差の中央値以上を上位群、中央値未満を下位群に分けて分析する。正規化利益率予実差の平均値は 0.0、中央値は -0.104 であった。

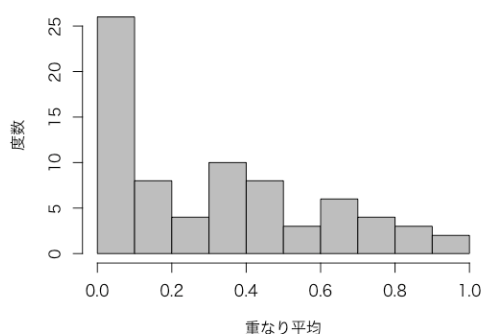


図 3 工程の重なり平均の分布

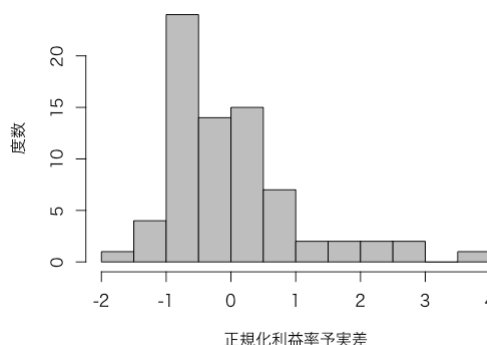


図 4 正規化利益率予実差の分布

2.4 品質メトリクス

品質メトリクスとして、上工程バグ摘出率、開發生産性、上工程新規バグ数/KL、テスト工程新規バグ数/KL、新規テスト項目数/KL、各工程の開発工数/KL、各工程のレビュー工数/KL、および各工程の新規バグ数/KLのメトリクスの値を収集する。ここで、「上工程」とは図1のBD工程からCD工程までを意味する。「テスト工程」とは図1のUT工程からST工程までを意味する。

これらの品質メトリクスを用いた理由は、筆者らの組織で定常的に収集されており^{[15][16]}、かつ、品質データ分析において有用な結果が得られることが経験的に把握できているためである。それぞれの定義については、紙幅の都合上ここでは省略する。詳しくは文献^{[15][16]}などを参照されたい。

これらのメトリクスの欠測値比率の平均値は3.06%であり、欠測値は極めて少ない。なお、これらの品質メトリクスの実測値は非公開とする。

3. 分析結果

3.1 工程の重なりと利益率予実差：プロジェクト全体に着目

まず、RQ1「プロジェクト全体で見た工程の重なりと利益率との間には、どのような関係があるのか」について、工程の重なりと利益率予実差の関係を分析する。これらを両軸にとった散布図を図5に示す。相関係数は -0.179 であり、有意水準5%で負の相関があるとはいえない(p 値 $=0.127$)。ただし、図5の右上部分にデータが分布していないことから、工程の重なりと利益率予実差の平均が一定以上高くなると、正規化利益率予実差が大きくなるという傾向があるように見える。

そこで、工程の重なり平均に閾値を設けてデータを二分し、さらに正規化利益率予実差の中央値でデータを分割し、両者の関係についてカイ二乗検定により分析する。工程の重なり平均の閾値を決めるにあたり、図3において頻度のもっとも多い0.1を基準とした場合と、図5の右上部分に相当する0.4を基準とした場合における。それらのクロス集計をそれぞれ表1と表2に示す。

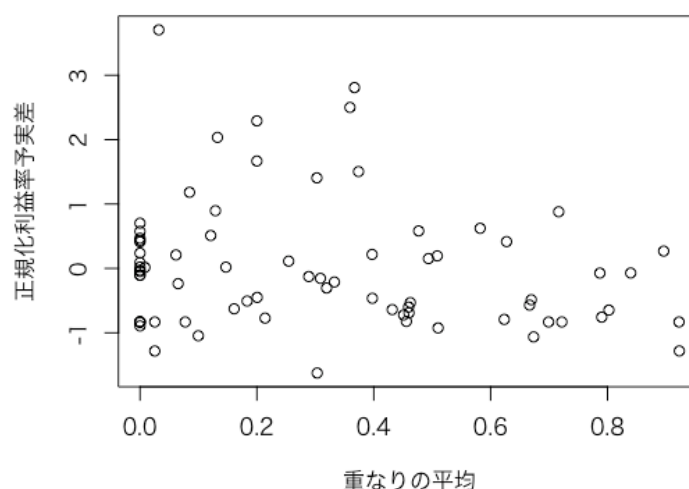


図5 重なり平均と利益率予実差の散布図

表1 重なり平均(0.1基準)と利益率予実差

	重なり平均		計
	0.1未満	0.1以上	
正規化利益率 上位群	16	21	37
予実差 下位群	10	27	37
計	26	48	74

表2 重なり平均(0.4基準)と利益率予実差

	重なり平均		計
	0.4未満	0.4以上	
正規化利益率 上位群	28	9	37
予実差 下位群	20	17	37
計	48	26	74

表 1 においてカイ二乗値は 2.13 であり、有意水準 5%で両者に関係があるとはいえない (p 値=0.223)。一方、表 2 においてカイ二乗値は 3.79 であり、有意水準 5%で両者に関係があるとはいえないものの、有意水準 10%では関係があるという結果が得られた (p 値=0.088)。これらの結果から、重なるの平均というプロジェクト全体の指標で見たときに、この値が 0.1 未満か否かは利益率予実差にはほとんど影響せず、0.4 未満までは利益率予実差の向上余地があるものの、0.4 以上では利益率予実差の大幅な向上が難しくなることが示された。

3.2 工程の重なりと利益率予実差：個別工程の組合せに着目

次に、RQ2「工程の重なりを個別に見たときに、利益率と関わりのある工程はどれか」について、隣り合う 2 工程の重なりと利益率予実差との関係を分析する。図 6 に、BD と FD の組合せから IT と ST の組合せまでについて、2 工程の重なりを利益率予実差の上位群と下位群に分けた箱ひげ図を示す。また、それらの平均値と中央値をそれぞれ表 3 と表 4 に示す。なお、 p 値について、表 3 は t 検定、表 4 はウィルコクソンの順位和検定の結果を示している。

図 6 より、それぞれの組合せにおける工程の重なりについて、利益率予実差の上位群と下位群のいずれにおいても計画時点で一定の幅で重なりのある日程計画が立てられていることがわかる。そして、BD と FD の組合せは利益率予実差の上位群の方が下位群よりも重なりが大きく、ほかの組合せはその逆であるように見える。表 3 と表 4 より、DD と CD、および CD と UT の組合せについては、利益率予実差の下位群の方が上位群に比べて、工程の重なりが平均値と中央値のいずれも有意に大きいことが示された。また、UT と IT、BD と FD の組合せでは、中央値については有意差があることが示された。そして、BD と FD は利益率予実差の上位群の方が工程の重なりが有意に大きいという、他の工程の組合せとは逆の結果が得られた。

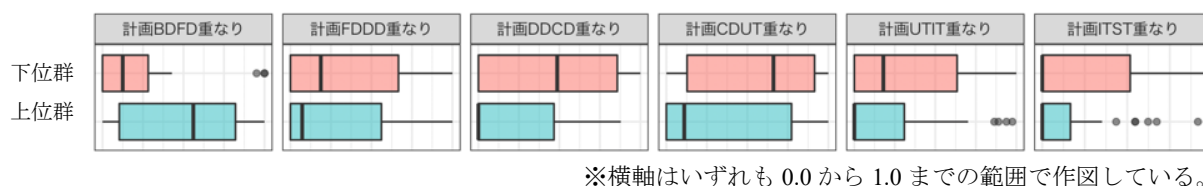


図 6 利益率予実差別の工程の重なり分布

表 3 利益率予実差別の工程の重なり平均値比較

利益率予実差	BDFD重なり	FDDD重なり	DDCD重なり	CDUT重なり	UTIT重なり	ITST重なり
下位群	0.266	0.323	0.448	0.554	0.317	0.239
上位群	0.478	0.277	0.209	0.359	0.202	0.134
n	31	73	71	69	70	73
p値	0.129	0.579	0.006***	0.047**	0.164	0.123

*** $p < .01$ ** $p < .05$ * $p < .1$

表 4 利益率予実差別の工程の重なり中央値比較

利益率予実差	BDFD重なり	FDDD重なり	DDCD重なり	CDUT重なり	UTIT重なり	ITST重なり
下位群	0.125	0.188	0.487	0.661	0.661	0.661
上位群	0.561	0.073	0.000	0.110	0.000	0.000
n	31	73	71	69	70	73
p値	0.099*	0.506	0.006***	0.043**	0.040**	0.107

*** $p < .01$ ** $p < .05$ * $p < .1$

3.3 利益率予実差を説明する要因

最後に、RQ3「工程の重なりと品質メトリクスを用いて、利益率を説明することはできるか」について、利益率予実差と、隣り合う2工程の重なりと品質メトリクスとの関係を分析する。ただし、品質メトリクスの分布を調べたところ、外れ値と思われる値を含んだプロジェクトが2件あったため、これらのデータを除外する。また、BDとFDの重なりは表3に示した通り欠測値が多いことから、このデータ項目は削除した上で分析する。さらに、上工程の状況がプロジェクト全体に影響するという観点から、個別工程の開発工数、レビュー工数、および新規バグ数については、上工程であるFD、DD、およびCDを対象とした。

まず、3.2節と同様、それぞれの品質メトリクスの分布を、利益率予実差の上位群と下位群に分けたときに、それぞれの平均値に有意差があるかどうかを分析した。その結果、上工程バグ摘出率のみ有意差が確認でき（ p 値=0.048）、利益率予実差の上位群の方が下位群に比べて上工程バグ摘出率が高いことが示された。別の言い方をすれば、上工程バグ摘出率以外の品質メトリクスは、開發生産性を含めすべて有意差が見られなかった。このことから、それぞれの品質メトリクスを単独で見たときに、利益率予実差と直接的な関係があるものは上工程バグ摘出率のみであり、それ以外の品質メトリクスについては直接的な関係がないといえる。

続いて、隣り合う2工程の重なりと品質メトリクスを説明変数としたときに、これらの説明変数を総合的に用いて利益率予実差を説明できるかについて分析する。図7に、決定木分析を行った結果を示す。図7に示した通り、工程の重なりについては、DDとCD、およびFDとDDの組合せが利益率予実差を説明する変数として得られた。また、品質メトリクスについては、FD開発工数/KL、DD開発工数/KL、およびFDレビュー工数/KLが利益率予実差を説明する変数として得られた。なお、これらの品質メトリクスが示されたノードにおける閾値は、2.3節で示したものと同一方法で正規化している。

図7より、次のことが読み取れる。まず、DDとCDの重なりが0.883以上と非常に高いプロジェクトは、そのすべてが利益率予実差の下位群である（Node 2）。それ以外では、FD開発工数/KLが正規化された値で-0.899未満と相対的に少ないものは、そのすべてが下位群である（Node 4）。それ以外では、DD開発工数/KLが正規化された値で0.753以上と相対的に多いものは、6割以上が下位群である（Node 6）。それ以外では、FDレビュー工数が正規化された値で-0.397以上と一定レベル以上の工数をかけているものは、約9割が上位群である（Node 11）。それ以外では、FDとDDの重なりが0.021未満と極めて低いものは、そうでないものに比べて下位群となる可能性が高い（Node 9, Node 10）。以上のことが決定木分析により示された。

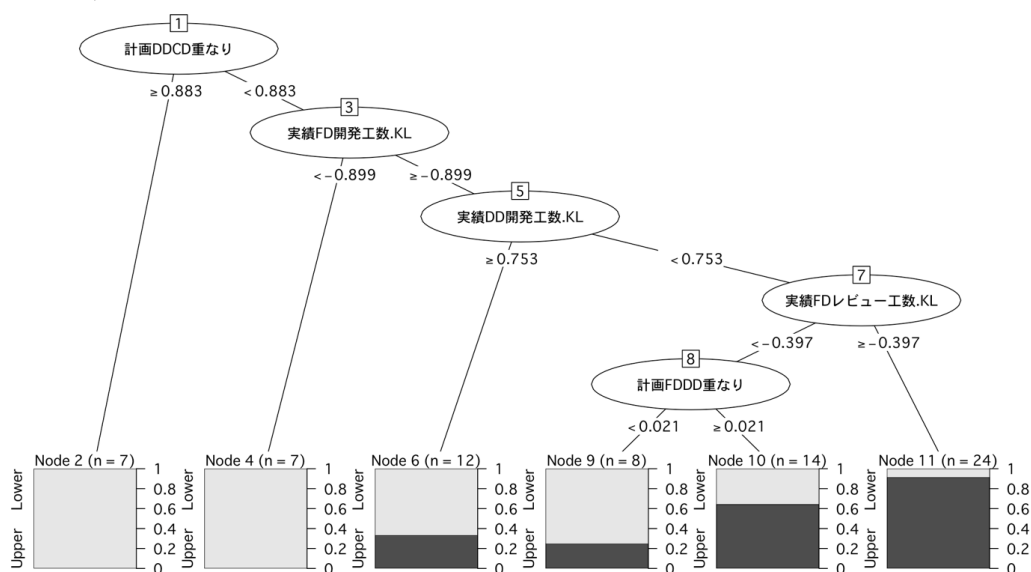


図7 利益率予実差の上位群・下位群を目的変数とした決定木分析

4. 考察

4.1 本研究の実務的および学術的意義

RQ1 について、プロジェクト全体で見た工程の重なり平均が 0.1 未満の場合、利益率予実差の上位群と下位群の比率に差はないが、工程の重なり平均が 0.4 未満の場合は、それ以上の場合に比べて利益率予実差が上位群になる可能性が高くなることがわかった。このことから、ある程度までは工程に重なりのある日程計画を立てることが、利益率を向上させる余地を残すものと考えられる。これは、工程に重なりを持たせることで、次工程の作業者の手待ち時間等を減らせたり、工程間での情報共有が促進されたりといった効果が得られ、作業が効率的に進められるものと考えられる。一方で、過度な重なりを持たせた日程計画は控えるべきといえる。

RQ2 について、BD と FD は重なりを大きくした方が利益率を向上させる可能性があることが示された。他の工程の組合せに比べて欠測値比率が高いことを考慮しなければならないものの、FD におけるフィージビリティ検討を深めながら BD の作業を行うことが、全体のコスト削減に結びつく可能性があるものと考えられる。一方で、DD と CD、CD と UT、UT と IT の組合せについては、重なりを小さくした方が利益率を向上させる可能性があることが示された。これらの工程については、過度な重なりを持たせた日程計画は控えるべきといえる。

RQ3 について、上工程バグ摘出率が利益率予実差と直接的に関係していることが示された。上工程バグ摘出率が高い水準となるように上工程の品質を良くすることで、総合的な改善につながるものと考えられる。このことは、これまでに筆者らが行ってきた品質データ分析からも、上工程重視により品質向上と生産性向上が両立可能との結果を得ており^[16]、ビジネスの観点からもより一層、上工程での品質を良くすることが必要であるといえる。

また、RQ3 の決定木分析について、DD と CD の重なりが大きい場合、CD の後戻り作業による工数への影響が大きくなり、これが全体の利益率に影響する可能性があるものと考えられる。FD 工数が少ない場合も、FD での検討不足のために後戻り作業による工数への影響が大きく、FD の品質による影響で DD の工数が多くかかり、利益率に影響がでているのではないかと推測する。反対に、FD レビュー工数に時間をかけ、品質を向上させた場合、その後の工程の品質も上がり、結果として利益率にプラスの影響がでているのではないかと考えられる。

本論文の学術的貢献は、ウォーターフォール型ソフトウェア開発という文脈において、工程の重なりと利益率との関係を実証的に示した点である。筆者らが知る限り、この文脈における工程の並行化については、シミュレーション研究によるもの^{[15][16]}や、並行ソフトウェア開発の方法論を概念的に示したもの^{[7][8]}はあるが、実証的な研究は十分に示されていない。工程の重なりが利益率に影響することを実証的に示したことは、学術的に見ても価値ある取り組みであると考えられる。

加えて、利益率に着目した点について新規性があるものと考えられる。品質データ分析では、品質や生産性を目的変数とすることが多い。論理的には、その延長上として利益率が定まるが、実際に品質データと利益率を紐付けた品質データ分析が報告された事例は多くないと思われる。これらの紐付けを行った分析結果を示したことに、本論文の意義があるといえる。

4.2 妥当性の脅威

本論文では工程の日程計画に基づいて議論しており、実際の稼働や工数に基づいた議論ではない点に留意が必要である。ある工程の開始日から終了日にわたって、つねに一定の工数がかかけられているわけではない。稼働のなされていない日もあり得る。そのため、二つの工程が重なっているからといって、それぞれの作業が同時進行しているとは限らない。このような日程計画だけでは見えない部分が分析結果に影響している可能性は否めない。

また、決定木分析はデータの変動によって構造が変わることが多いため、RQ3 の結果はデータによって変動する可能性がある。複数年度のデータを用いてサンプルサイズを増やし、ランダムフォレスト法などを適用するなどの検討が求められる。

5. おわりに

本論文では、ウォーターフォール型ソフトウェア開発の日程計画に基づいた工程の重なりと利益率との関係を分析した。その結果、工程の重なりが大きくなると利益率向上の余地が小さくなること、DD と CD, CD と UT, UT と IT の重なりは小さい方が望ましいが、BD と FD は逆に重なりが大きいが望ましいこと、さらには、工程の重なりと品質メトリクスにより利益率の上位群と下位群を分ける構造を示すことができた。今後の課題として、サンプルサイズを増やした上でより普遍的な分析結果を得ることなどが挙げられる。

謝辞 本研究を行う機会と貴重なご助言をいただいた菅田直美氏（元：日本電気株式会社ソフトウェア品質主席主管，現：株式会社イデソン代表取締役）に深く感謝いたします。

参考文献

- [1] Benington, H.D. (1956): Production of large computer programs. *In Proceedings, ONR Symposium on Advanced Programming Methods for Digital Computers*, June 1956, pp 15-27.
- [2] Bindal, N., & Mehta, A. (2015). Survey On Software Development Processing Models. *International Journal of Exploring Emerging Trends in Engineering (IJEETE)*, ISSN, 2394-0573.
- [3] Royce, W. W. (1987). Managing the development of large software systems: concepts and techniques. *In Proceedings of the 9th international conference on Software Engineering*, pp. 328-338.
- [4] Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 35(3), pp.8-13.
- [5] Nonaka, M., Zhu, L., Babar, M. A., & Staples, M. (2007). Project cost overrun simulation in software product line development. *In International Conference on Product Focused Software Process Improvement*, pp. 330-344.
- [6] Rahmandad, H., & Weiss, D. M. (2009). Dynamics of concurrent software development. *System Dynamics Review: The Journal of the System Dynamics Society*, 25(3), pp.224-249.
- [7] Aoyama, M. (1993). Concurrent-development process model. *IEEE software*, 10(4), pp.46-55.
- [8] Blackburn, J., Scudder, G., & Van Wassenhove, L. N. (2000). Concurrent software development. *Communications of the ACM*, 43(11es), 4-es.
- [9] ガートナー. (2019). アプリケーション開発に関する調査結果. <https://www.gartner.com/jp/newsroom/press-releases/pr-20190221> (2020 年 8 月 10 日参照).
- [10] Van Casteren, W. (2017). The Waterfall Model and the Agile Methodologies: A comparison by project characteristics. *Research Gate*.
- [11] Shaydulin, R., & Sybrandt, J. (2017). To agile, or not to agile: A comparison of software development methodologies. *arXiv preprint arXiv:1704.07469*.
- [12] 柳田礼子・野中誠・菅田直美. (2017). CMMI 成熟度レベル別に見たソフトウェア品質の良否にかかわる要因の複合的分析. *SEC journal*, 13(1), pp. 8-15.
- [13] 吉村直人・野中誠. (2018). ソフトウェア再利用が改修プロジェクトの生産性に与える影響の実証分析. プロジェクトマネジメント学会 2018 年度秋季研究発表大会予稿集, p.7
- [14] 独立行政法人情報処理推進機構社会基盤センター監修. (2018). ソフトウェア開発データ白書 2018-2019. 独立行政法人情報処理推進機構.
- [15] 菅田直美. (2010). ソフトウェア品質会計. 日科技連出版社.
- [16] 宮崎義昭・菅田直美. (2017). NEC における定量的管理: ソフトウェア品質会計と NEC データ白書に基づく改善活動の展開. *SEC journal*, 13(3), pp. 36-39.