

筑波大学大学院博士課程

システム情報工学研究科特定課題研究報告書

SWOT 分析支援ツールの開発
ーヘルプ機能の開発および定量的な進捗把握
に基づく遅延への対策ー

胥 徳文

修士（工学）

（コンピュータサイエンス専攻）

指導教員 田中二郎

2015年 3月

概要

本報告書は、筑波大学「高度 IT 人材育成のための実践的ソフトウェア開発専修プログラム」における「研究開発プロジェクト」の一つである SWOT 分析支援ツール開発プロジェクトを進める際に、行ったヘルプ機能の開発および定量的な進捗把握に基づく遅延への対策について報告するものである。

本プロジェクトの目的は、顧客が抱える問題を解決するためのシステムを提案して開発することである。本プロジェクトの顧客は山戸昭三研究員であり、顧客は教員としてまた IT コンサルタントとしてクライアント（学生チームまたは企業の経営改革チーム）に SWOT 分析をさせる場面がある。SWOT 分析は、企業の現状を内部要因と外部要因を分けて分析し、その分析内容をクライアントで共有するための手法の一つである。顧客は SWOT 分析を行わせる際に、いくつかの問題を抱えている。本プロジェクトは、顧客これらの問題解決するために、ヒアリングして要求を分析し、顧客の実施シーンや実施手順を分析した上で、利用システム開発要件を洗い出し、顧客と合意しながらシステム開発を実施した。

筆者は本システム開発において、顧客が求める価値を実現するために、SWOT 分析支援ツールの開発をチームメンバとともに実現した。

本プロジェクトにおける筆者の貢献は主にタイムマネジメントによる進捗管理することとヘルプ機能を開発すること 2 つがある。まず、タイムマネジメントによる進捗管理では、プロジェクトの目標の確実に実現するために、進捗報告の曖昧性削減を工夫し、EVM を用いて定量的にプロジェクト進捗を把握した。定量的な進捗把握に基づき、進捗遅れに対して実現可能性がある対策を提案して、実施まで働きかけることで、進捗遅れ問題の解決を支援した。また、筆者がヘルプ機能の開発による、システム運用教育での問い合わせ回数削減し、運用教育の時間短縮を実現でき、システム習得性向上へ貢献した。顧客が求める価値を提供し、顧客の気の利いたガイドが出る要望を満たすために、ヘルプシステムを調査した上で、分かりやすいアニメーションヘルプ機能を設計、実装した。最後に、実証実験でヘルプ機能の評価を行った。

目次

第1章	はじめに	1
第2章	プロジェクト背景	2
2.1	本プロジェクト概要	2
2.2	SWOT 分析とは	2
2.3	顧客の概要	4
	初回 SWOT 分析の実施	4
	個別に分析内容の検討	4
	検討結果の共有と分析の再実施	5
	分析結果の発表	5
2.4	顧客が抱えている問題	6
2.5	顧客の要望	7
第3章	提案システム	8
3.1	システム概要	8
第4章	開発概要	10
4.1	開発体制	10
4.2	開発環境	11
4.3	開発計画	12
4.4	開発実績	13
第5章	筆者の貢献	14
5.1	タイムマネジメント	14
5.1.1	定量的な進捗把握	14
5.1.2	進捗遅れへの対策	22
5.2	ヘルプ機能の設計と開発	28
5.2.1	ヘルプ機能提案	28
5.2.2	ヘルプ機能の設計と実装	29
5.2.3	ヘルプ機能の実証実験	31
第6章	まとめ	35
	謝辞	36
	参考文献	37

図目次

図 2-1	SWOT マトリックス	2
図 2-2	SWOT 分析手順	3
図 2-3	SWOT 分析のプロセス	5
図 3-1	提案システム構成	8
図 4-1	開発スケジュール	12
図 5-1	第1 イテレーションのタスク進捗報告例①	15
図 5-2	第1 イテレーションのタスク進捗報告例②	16
図 5-3	第1 イテレーション タスクチケット例	17
図 5-4	第1 イテレーションのタスク実績例①	18
図 5-5	第1 イテレーションのタスク実績例②	19
図 5-6	第1 イテレーションのタスク実績例③	19
図 5-7	第1 イテレーションのタスク実績例④	19
図 5-8	第1 イテレーションのタスク実績例⑤	20
図 5-9	第1 イテレーションのタスク実績例⑥	20
図 5-10	第1 イテレーション 進捗報告画面	21
図 5-11	タスクステータスの進捗率設定	21
図 5-12	レビューなしタスクステータスの遷移設定	21
図 5-13	チームレビューだけのタスクステータスの遷移設定	22
図 5-14	チームレビューと顧客レビューがあるタスクステータスの遷移設定	22
図 5-15	第2 イテレーション 進捗報告画面	22
図 5-16	開発1ヶ月間の EVM	24
図 5-17	開発1ヶ月間 SV と CV	24
図 5-18	単体テストまでの EVM	25
図 5-19	単体テストまでの SV と CV	25
図 5-20	第1 イテレーションの EVM	26
図 5-21	第1 イテレーションの SV と CV	27
図 5-22	第1 イテレーションの PC	27
図 5-23	第1 イテレーションの TEAC	27
図 5-24	ヘルプ機能画面イメージ	29
図 5-25	ヘルプ機能ユースケース	29
図 5-26	AndroidView のライフサイクル	31
図 5-27	筑波大学 SWOT 分析図の例	32

表目次

表 3.1 機能一覧.....	9
表 4.1 プロジェクトステークホルダー.....	10
表 4.2 機能開発の役割分担.....	10
表 4.3 マネジメントの役割分担	11
表 5.1 実装タスク実績の時間割①.....	18
表 5.2 レビューなしタスク出来高計測基準	18
表 5.3 タスク実績の時間割②.....	19
表 5.4 チームレビューだけのタスク出来高計測基準	19
表 5.5 タスクの実績時間割③.....	20
表 5.6 チームレビューと顧客レビューがあるタスク出来高計測基準.....	20
表 5.7 第1イテレーション評価での操作に関する問い合わせ回数	28
表 5.8 自由テーマでの必須作成項目	33
表 5.9 グループ1のアンケートと計測したデータ	33
表 5.10 グループ2のアンケートと計測したデータ	34

第1章 はじめに

本プロジェクトは、実際的な顧客に対して、システムを提案し、開発することで、顧客が抱えている課題を解決すると目指す。本プロジェクトの顧客はクライアント（学生チームまたは企業の経営改革チーム）に SWOT 分析をさせる際に、SWOT 分析を実施する方法について課題を抱えている。本プロジェクトは顧客にヒアリングを実施して、顧客が求める価値を抽出し、その価値を満たすために、要件定義から評価まで一連の研究開発を行う。

本プロジェクトの顧客は SWOT の実施プロセスや実施手順について課題を抱えている。企業の現状分析を行われる際に SWOT 分析という手法が広く知られ、活用されている。SWOT 分析とは、企業の現状を強み (Strength)、弱み (Weakness)、機会 (Opportunity)、脅威 (Threat) の 4 つのカテゴリで、企業に影響を与える要因の分析を行う。現在、顧客が SWOT 分析の実施を行わせる際に、プロセスが以下 4 つに抽象できる。①初回 SWOT 分析を実施する。②初回 SWOT 終了後、参加者が個別に分析結果を検討する。③後日に、参加者が個別に検討した結果を統合し、継続して分析を行う。④SWOT 分析結果を用いてプレゼンテーションする。また、プロセス毎に SWOT を実施する手順としては以下の通りである。①参加者アイディアを付箋に書き出す。②付箋を集めて、山作りやグループとしてまとめる。③付箋やグループの相互関係を示すため、関連線を繋げる。これらの業務プロセスと業務手順について、現在、ホワイトボード、模造紙、付箋などを用いて SWOT 分析を行う顧客が課題を抱えている。

本報告書では、本プロジェクトの顧客の業務課題を分析し、顧客へのシステム提案や開発などのプロセスやプロジェクトにおける筆者の貢献を報告する。本文の構成としては、2 章で、プロジェクトの背景として、顧客の業務現状、抱えている課題、及び要望について述べる。そして、これらの課題を解決し、顧客の要望を満たすために、顧客へ提案するシステム要件について説明する。3 章では、提案システムの概要、構成、開発する機能など提案システムの詳細を述べる。4 章では、提案システムを開発する際のステークホルダー、開発計画、開発スケジュールと本プロジェクトでのマネジメントの役割分担などを述べる。5 章では、本プロジェクトにおける筆者のタイムマネジメントによる進捗管理及びヘルプ機能開発の貢献について述べる。最後の 6 章で本報告書をまとめ、筆者の観点から本プロジェクトを振り返る。

第2章 プロジェクト背景

本章では、本プロジェクトの概要を紹介し、本プロジェクトの顧客及び顧客の業務現状、抱えている課題や顧客を求める価値などを説明した上で、競合製品の調査に結果を説明する。本プロジェクトは「SWOT 分析支援ツールの開発」プロジェクトであり、SWOT 分析を実施について課題を抱えている顧客に対して、システム提案することにより、顧客の問題解決を目指す。

2.1 本プロジェクト概要

本プロジェクトは、筑波大学大学院の「高度 IT 人材育成のための実践的ソフトウェア開発専修プログラム（高度 IT プログラム）」[18]の一環として、学生がソフトウェア開発に必要なスキルを身に付けるため、実施された実践的なプロジェクトである。本プロジェクトは、特定課題研究開発プロジェクトと呼び、筑波大学大学院システム情報工学研究科・コンピュータサイエンス専攻・高度人材育成のための実践的ソフトウェア開発専修プログラムにおいて実施されている。

特定課題研究開発プロジェクトに参加する学生はチームを組んで、実際の顧客に対して、身に付けたスキルを活用し、ソフトウェア開発することで、顧客の抱えている課題を解決する。開発チームは要件定義、設計、実装、テスト、評価など一連の開発プロセスを通して、顧客の要望を抽出し、顧客の抱えている課題を解決するために、要件を定義し、顧客にシステム提案し、設計と開発をした。最後に、実証実験で開発したシステムの評価を行い、成果物を顧客に納品する。

2.2 SWOT 分析とは

SWOT 分析とは、企業の内部要因と外部要因を把握するための、経営戦略に係るフレームワークの一つである。図 2-1 SWOT マトリックスに示す、企業の現状を強み (Strength)、弱み (Weakness)、機会 (Opportunity)、脅威 (Threat) の 4 つのカテゴリで、企業に影響を与える要因の分析を行う。現在、顧客を含め、日本の企業、自治体、NPO が自社の現状を把握するために、SWOT 分析が広く知られ、活用している手法である[1][2]。

	好影響	悪影響
内的要因	S(強み)	W(弱み)
外的要因	O(機会)	T(脅威)

図 2-1 SWOT マトリックス

SWOT 分析を実施する時、図 2-2 SWOT 分析手順に示すように実施する。

- ① SWOT 分析参加者がブレインストーミングにより、アイデアを付箋に書き出し、項目として各領域内で貼り付ける。
- ② 項目を貼り付けることに伴い、参加者が項目を分類し、似たようなアイデアを書いた付箋を集め、グループとしてまとめる。
- ③ 項目とグループを整理した後、項目やグループ間の関連関係がある場合、参加者が関連線を繋げることで関連関係を示す。

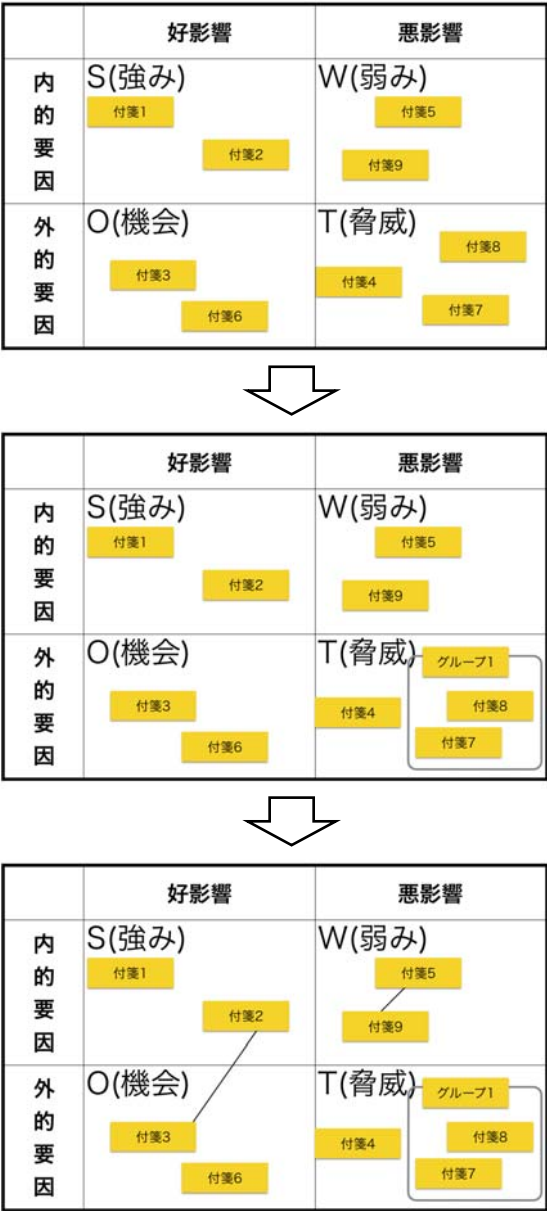


図 2-2 SWOT 分析手順

2.3 顧客の概要

本プロジェクトの顧客は2つの場面でSWOT分析の実施を指導し、SWOT分析の流れは4つのプロセスに分類した。顧客の利用場面について、まず、顧客が大学の講師として学生に授業を行う際に、学生に例題を提示し、学生のSWOT分析を指導する、次に、ITコンサルタントとしてクライアント企業の教務改革のために、SWOT分析を行い、クライアントが抱えている業務課題の抽出を行う。SWOT分析の流れは初回SWOT分析の実施、個別に分析内容の検討、個別検討結果の共有とSWOT分析の再実施、分析結果の発表、4つプロセスに分類できた。

顧客は講師として学生にSWOT分析を指導する場面では、授業中複数人の共同実施と放課後個別の検討2つの利用シーンがある。授業中に顧客はSWOT分析の手法を説明して事例を提示し、学生チームにSWOT分析の議論を行わせ、学生チームは提示された事例に対して議論しながら、アイデアを書き出して、SWOT分析を実施する。放課後に、宿題として、引き続き学生は個別に検討を行う。後日に、学生が集め、個別の検討結果をチームに統合し、SWOT分析を継続して行うこともある。最後に、全員の前でSWOT分析の検討結果を発表する。

また、ITコンサルタントとして企業に出向き、企業の現状認識をサポートするために、SWOT分析の実施を指導する。企業のSWOT分析を実施する際に、その際に企業の社員はSWOT分析を実施する主体であり、顧客は時に建言して分析に参加する、社員は企業現状について、内部環境要因と外部環境要因を分析する。その後、社員は個人ごとに4つの領域S強み、弱み、機会、脅威の中の1つの領域を担当し、個別に検討を行う場合もある。次回のSOWT分析では、社員が個別検討した結果を統合し、継続して業務課題の抽出を行う。最後に、後日、SWOT分析の結果を上司に発表する。

以上の2つの場面をまとめ、顧客はSWOT分析を指導する際に、SWOT分析の流れは4つのプロセスに分類できる。図 2-3 SWOT分析のプロセスに示すように、SWOT分析のプロセスは1回だけを実施することではなく、複数回を実施する場合は多い。最初は、まず顧客が指導のもとに参加者があるテーマを中心にSWOT分析を実施する。その後、参加者が実施した分析内容について個別に検討を行う。後日、参加者が集めて、個別に検討した内容を統合し、継続してSWOT分析を行う。最後に、検討した結果を発表する。以下で各プロセスの詳細を説明する。

初回 SWOT 分析の実施

まず、初回SWOT分析を実施する際に、参加者はあるテーマを中心に分析を行う。顧客の現状の一つの例としては、ホワイトボード、模造紙や付箋などを用いて、SWOT分析の実施を行わせている。その場合、参加者は付箋にアイデアを書き出し、ホワイトボードや模造紙に貼り付けながら、議論を行い、グループになるアイデアをグループで囲い、関連線で繋がることでアイデアやグループの相互関係を表示する。

個別に分析内容の検討

次に、参加者が個別に分析結果を検討する場合がある。最初のSWOT分析を実施した後、参加者が個別に実施したSWOT分析の内容を持ち帰り、検討を行う。このプロセスでは、参加者が個別に分析結果の関連内容を調べたり、ある要因を深く考えたりすることができ、分析結果を振り返る。

検討結果の共有と分析の再実施

後日に、参加者が個別の検討結果を統合し、再議論を行う。このプロセスでは、参加者は集めて、個別に検討した内容を統合して、継続して SWOT 分析を行う。このプロセスで各要因や要因間の関連性及びグループの構成を修正する場合もある。例えば、企業の経営戦略分析の場合は、社員は前回の SWOT 分析結果を再現し、各自担当した検討結果を共有して加え、再議論することができる。

分析結果の発表

最後に、SWOT 分析の結果を発表する。このプロセスは参加者が現状の内部要因と外部要因は何があるか、要因の間どのような関係があるかなどの SWOT 分析の内容を発表する。学生の場合は、SWOT 分析の成果をクラスの全員に報告する。企業 SWOT 分析の場合は、社員が上司や役員に向けて抽出した経営戦略課題や戦略シナリオを発表する。発表する際にホワイトボードで SWOT 分析結果を再現して発表する場合があります、PC とパワーポイントを用いて、スクリーンに SWOT 分析結果を映して発表する場合もある。



図 2-3 SWOT 分析のプロセス

2.4 顧客が抱えている問題

前節で述べた顧客の利用場面や利用プロセスを分析し、顧客にヒアリングを実施した上で、現在顧客が SWOT 分析に対して抱えた問題を抽出した。現在、顧客が指導する SWOT 分析を行う時、主に模造紙、ホワイトボード、付箋を用いているため、分析結果を保存して後日に再利用することが難しい、発表する際に手書き文字が読みにくい、模造紙や付箋など道具を出先に持ち運ぶ手間がかかるという問題を顧客が抱えている。問題の詳細は以下で述べる。

まず、グループや関連線両端の付箋を移動することが難しい問題があった。ホワイトボード、紙付箋、模造紙で SWOT 分析を実施する場合は、作成済みのグループを移動することは手数をかかり、作成した関連線の修正が難しい。グループを移動する場合は、囲まれたグループ内の付箋を一枚ずつ移動する必要がある。ペンで書いた関連線の両端の付箋を移動する場合は、元の関連線を消し、付箋を移動した後、関連線を書き直す。元の関連線が消せない場合は、関連線両端の付箋の位置が変えられない、修正が難しいという問題がある。

次に、SWOT 分析が中断される場合は再開が難しいという問題があった。SWOT 分析の結果或いは検討途中で中断された結果を保存して再編集することが難しい。模造紙と付箋を保存する時、付箋が落ちたり、紛失したりすることも発生する可能性がある。ホワイトボードで実施した SWOT 分析結果を再現する時、グループの構成や関連線がペンで書いてある場合が多いため、後日に個別の検討結果を反映する時、完全に前回の分析結果を再現することが難しい、手数もかかる。ホワイトボードで実施する場合は、保存するための空間がかかるため、大量的な結果を保存することや長期間の保存に向いていないという問題もある。

そして、検討結果を論文やプレゼン資料に反映することが難しいという問題があった。アイデアの書き出しは手書きで行うので、書き手によって字が読めない場合が存在し、その内容を用いて発表する、或いは論文に引用される時に、書き手により文字が読みにくい、付箋に書いた文字が小さくて見えにくいという問題がある。

最後に、実施のため、道具の持ち運びに手間がかかるという問題があった。実施する前の準備段階や実施した結果を用いて発表を行う際に、道具の持ち運びが必要があり、手間がかかるという課題が抱えている。模造紙、ホワイトボードを利用する場合は、出先にこれらの道具を用意していない時、持って行く必要がある。複数回実施する場合は、分析結果を別の場所へ持ち運ぶ時、手間がかかってしまうという問題がある。

2.5 顧客の要望

開発チームは顧客が抱えている問題を分析し、顧客の要望，すなわち，実現して欲しい価値を抽出した．開発チームが顧客の SWOT 分析の現状に対する不満足点について，価値の階層構造（VBS[Value Breakdown Structure]）という分析で抽出した．抽出した価値がつぎに通りでまとめた．

1.簡単に付箋を作り，動かせる．SWOT 分析はブレインストーミングであるので，ボトムアップ的に思考を集約できるため，ブレインストーミングのしやすさ，アイディアの発想に集中することが重要である．価値としては，SWOT 分析を実施する時，簡単に付箋をすること．グループ，関連線で付箋間や山同士の関係を表現できること．作成した付箋，グループや関連線を動かせること．色づけ，拡大縮小で仕様を修正すること．

2.検討内容を共有しやすい．SWOT 分析基本は 3-4 人で現状の問題，課題，戦略などの思考を共有しながら，検討をするため，小人数のアイディア共有が重要な価値であり，また，検討した結果を大人数向けに発表するため，検討の結果をプレゼンテーションも重要な価値である．

3.検討の中断と再開が容易にできる．SWOT 分析を実施する際に，途中で中断される場合があり，参加者個別の検討を含まれる複数回で実施する場合がある．このように，中断された検討の結果を保存して，次回の検討で，再開が容易にできることが顧客の要望である．

4.検討結果を論文やプレゼン資料に反映しやすい．SWOT 分析の結果を発表したり，論文の中に引用したりすることが多いため，分析結果を引用や文章に挿入でき，特定のファイル形式で出力することが顧客の要望である．

5.デザインが洗練されている．SWOT 分析支援するツールは SWOT 分析の参加者に使ってみたいという思いをさせるため，ユーザインタフェースのデザインなどを配慮する必要がある．

6.初心者にも操作法が容易にわかる．SWOT 分析支援するツールを利用する時，ユーザの操作を支援し，気の利いたガイドが出ると希望する．

7.システム価格が 10 万円以下で実現できる．

8.SWOT 分析の準備が手軽にできる．SWOT 分析の実施を準備する際に，手間のかからないことが顧客の要望である．最初回の実施だけではなく，個人別に検討した結果を統合して議論を再開する場合，結果を発表するために実施場所を変更する場合も手軽に準備できること．

第3章 提案システム

本プロジェクトでは、顧客の要望を満たすために、SWOT分析支援するツールを提案した。本提案システムは Android アプリケーションとして開発を行い、アプリケーション名は SWOT Analysis Application（以下、SAA と略称する）である。SAA の提案では、顧客の利用場面や業務プロセスに合わせて、顧客の求める価値を実現するため、システムの要件を定めた。

3.1 システム概要

SAA は Android アプリケーションとして開発する SWOT 分析を支援するツールであり、SAA によって顧客が求める価値を実現する。顧客が SAA を利用することで、アイデアを入力して因子を作成することができる。作成した因子は指で自由に移動し、再編集することができる。実施した SWOT 分析の分析図がファイルとして保存でき、読み込むことができる。また、複数の端末で共同に SWOT 分析を行うことができ、入力したアイデアや作成した因子をリアルタイムに他端末上で反映し、SWOT 分析図を共有することができる。本提案システムの構成を図 3-1 提案システム構成に示す。

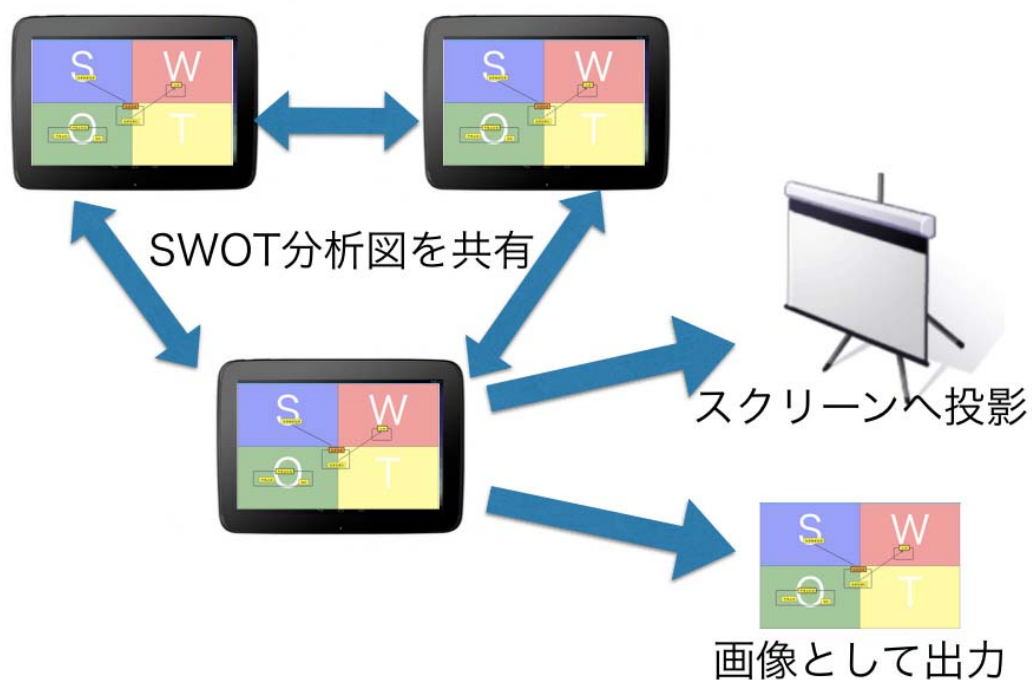


図 3-1 提案システム構成

対象ハードウェアはタブレット端末 Nexus10 と顧客に提案した。指で付箋を自由に移動させることができる, という要求を満たすためにタブレット端末でのソリューションを提案し, その中でも大型液晶画面やプロジェクタを使って内容を表示し参加者で状況を共有するためには, Nexus10 が最適な端末であると判断し, それを含めたシステムを提案した。

本システムの機能は顧客の利用場面及び利用プロセスに合わせ, 顧客の価値を実現するために要件定義を行った。本システムの機能を設計する時, 顧客の価値分析結果に基づいて実現する要件で要件定義を行い, 要件を満たすために, 開発する機能を表 3.1 機能一覧に定めた。

表 3.1 機能一覧

機能名	機能概要
SWOT 分析図の編集機能	画面上の SWOT 分析図を編集する機能
SWOT 分析図の保存と読み込み機能	作成した SWOT 分析図をファイルへ保存, 読み込みする機能
保存済み SWOT 分析図の検索機能	検索ワードに対応した, 保存したファイルを検索する機能
分析内容送信機能	画面上の全要素を他端末に一括送信する機能
リアルタイム共有機能	SWOT 分析図上に表示された要因や関連線などのオブジェクトを複数台の端末でリアルタイムに同期する機能
SWOT 分析図の画像出力機能	作成した SWOT 分析図を画像として出力する機能
ヘルプ機能	アニメーションによってユーザの操作を支援する機能
UI デザインの変更	SAA の操作性やデザインの統一性を図るために UI デザインを変更する

第4章 開発概要

本章では，本プロジェクトの開発体制を紹介し，開発環境，開発計画と開発実績を説明する．開発体制の説明では，本プロジェクト開発に関わるステークホルダーを紹介し，本プロジェクトの役割分担を説明する．そして，本プロジェクトの開発で利用するツールや開発環境を説明する．最後に，本プロジェクトの開発計画を述べ，開発の実績を報告する．

4.1 開発体制

本プロジェクトのステークホルダーは，課題担当教員である早瀬先生，顧客兼担当顧問である山戸研究員及び学生 4 名に構成されている．表 4.1 プロジェクトステークホルダーに示したように，学生 4 人が開発チームを組んで，顧客に対して研究開発を行い，顧客とのコミュニケーションを円滑にするため，課題担当教員である早瀬先生から指導を受ける．開発チームはコアタイムを週 3 回設定し，開発チームメンバがミーティング，レビュー，及び集中作業を行う，開発チームミーティングが週毎に開く，開発チームメンバの報告を行う．顧客ミーティングが月 2 回程度で行い，開発チームが開発状況を顧客に共有する．

表 4.1 プロジェクトステークホルダー

課題担当教員	早瀬先生
顧客兼担当顧問	山戸研究員
開発チーム	大塚優樹
	鈴木良生
	田中靖成
	胥徳文

本プロジェクト機能開発の役割分担は表 4.2 機能開発の役割分担に示しように，チームメンバ 4 人が 2 つのイテレーションを分けて，8 つの機能開発を行った．第 1 イテレーションで，SWOT 分析図の保存と読み込み機能，保存済み SWOT 分析図の検索機能，析内容送信機能 4 つの機能を開発され，第 2 イテレーションでは，リアルタイム共有機能，SWOT 分析図の画像出力機能，ヘルプ機能を開発され，UI デザインの変更が行われた．

表 4.2 機能開発の役割分担

開発メンバ	開発機能
大塚優樹	分析内容送信機能 リアルタイム共有機能
胥徳文	ヘルプ機能
鈴木良生	SWOT 分析図の保存と読み込み機能 SWOT 分析図の画像出力機能 保存済み SWOT 分析図の検索機能
田中靖成	SWOT 分析図の編集機能 UI デザインの変更

本プロジェクトでは、機能の開発だけではなく、開発チームメンバがそれぞれにプロジェクトマネジメントも実践できた。より高い品質、より低いリスクでプロジェクトを進めるために、開発チームメンバがマネジメントの知識を活用し、プロジェクトのスコープ管理、品質管理、リスク管理、スケジュール管理などを工夫した。本プロジェクトにおける各メンバのマネジメントの役割分担を表 4.3 マネジメントの役割分担に示す。その中に筆者はタイムマネジメントを担当し、プロジェクト目標を確実に実現するために、進捗管理を行った。具体的な貢献は、進捗報告の曖昧性を削減、EVMにより定量的に進捗把握に基づき進捗遅れへの対策について、5章で詳細を述べる。

表 4.3 マネジメントの役割分担

大塚優樹	プロジェクトリーダー スコープマネジメント
鈴木良生	リスクマネジメント
田中靖成	品質マネジメント
胥徳文	タイムマネジメント

4.2 開発環境

本プロジェクトでは開発環境として、システム機能を「Android ADT Bundle」と「Git」を利用し、プロジェクト進捗を管理では「Redmine」を利用した。Android ADT Bundle は、グーグル社が提供した Android アプリケーション開発環境のである。Git は、オープンソースの分散型バージョン管理システムである。Redmine は、オープンソースのプロジェクト管理ソフトウェアであり、タスク管理、進捗管理、進捗情報の共有できるツールである。

機能開発する際は、Eclipse に基づく Android ADT Bundle という開発環境を用いて、ソースコードを作成、修正し、Git によってソースコードを統合する。Android ADT Bundle は Java プログラミング言語に対応し、開発する時に、ソースコードの自動補完、プログラムのデバッグができる。Git で複数人が同時システムを開発する時の、ソースコードの変更履歴を記録、追跡し、ソースコードのバージョン管理を行った。

また、本プロジェクトの進捗管理が Redmine を利用した。Redmine 上でタスクをチケットとして発行することで、進捗報告の場を設け、タスクの進捗状況を監視でき、進捗の管理を行った。開発チームメンバがチケットの責任者として設定され、現在の進捗率や投入した工数などチケットに記録することで、進捗報告を行った。

4.3 開発計画

本プロジェクトではイテレーション型開発を採用し、2つのイテレーションを分けて開発を行った。図 4-1 開発スケジュールは全体のスケジュールである。各イテレーションでは、要件定義から評価まで行い、最後に顧客に納品する。

イテレーション型開発を採用した理由は、段階を分けて開発することで、顧客に密なコミュニケーションを実現でき、顧客の満足度を高めることができる。本プロジェクトは新規システムを開発するため、実施したヒアリングでは、顧客の要件をすべて定めることが難しいため、顧客が成果物を作成した後で要求が変化する可能性もある。複数回のイテレーションを実施することで、顧客の変化に柔軟に対応でき、顧客の要求を満たすプロジェクト開発を目指す。

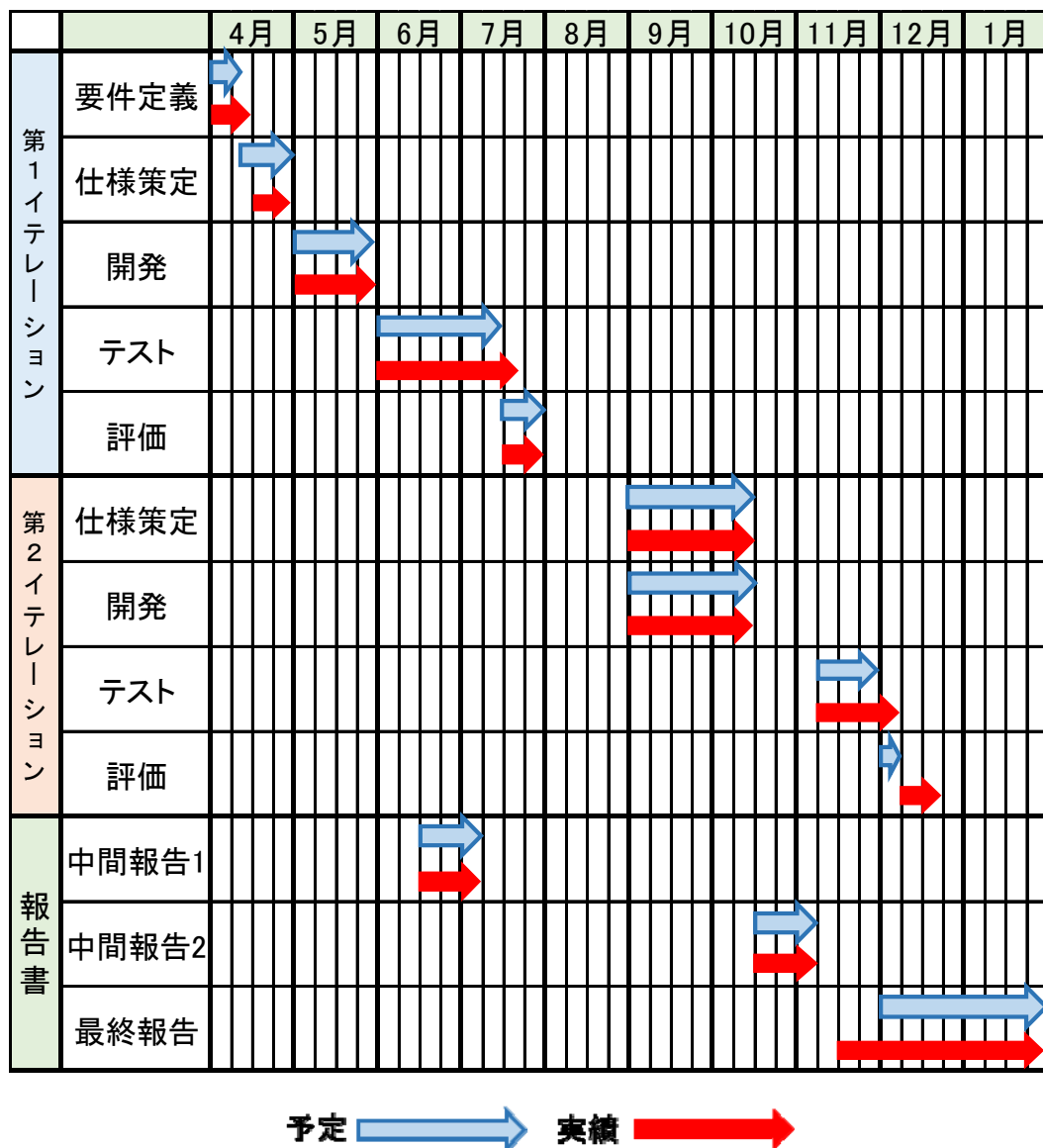


図 4-1 開発スケジュール

4.4 開発実績

本プロジェクトの開発では、計画通りに開発を行い、発生した問題に対策で解決し、顧客に納品できた。第1イテレーションの開発で、開発中の遅延へ対策することで、納期まで納品できた。第2イテレーションの開発で、2週間程度の遅れが発生したため、納期延長のという計画変更を行われ、結果として、改定した納期まで納品できた。

第1イテレーションでは要件定義から評価まで一連の開発を行い、顧客に納品できた。要件定義フェーズ顧客との間の打ち合わせに時間がかかったため、合意した設計内容のドキュメント化するタスクに遅れが発生した。また、開発工程では、タスクの粒度が多きいため、進捗計測の曖昧性が存在した。進捗が遅れなかったが、進捗把握のミスが発生した。単体テストフェーズで、進捗管理を見直し、進捗が正しく反映された。また、結合テストでは、結合テストの再実施が発生するため、結合テストの期間を1週間伸ばした。最後にチーム全体投入工数を増やすことで、納期まで納品することができた。

第2イテレーションは、もっと密にコミュニケーションを取りながら、必要となる成果物を揃えていくアジャイル開発で進んだ。開発する機能も顧客と仕様や確認しながら、開発を進んで、スケジュール上、仕様策定工程と開発工程を同時並行的に進めていた。テスト工程では、早めに開始した特定課題研究報告書と並行的に進んでいて、相互に影響を与えたため、テストの期間を延長され、納期を2週間程度延長した。納期の延長について、顧客ミーティングで顧客の合意をもらった上で、開発計画の変更を行った。最後に、改定した納期に納品できた。

第5章 筆者の貢献

本プロジェクトにおいて、筆者はタイムマネジメント[3]を担当し、進捗管理を行うことで、プロジェクトが円滑に進行することへ貢献し、ヘルプ機能を開発することによりシステムの習得性向上へ貢献した。タイムマネジメントでは、筆者は EVM[4]手法を用いて進捗を計測し、監視することにより、定量的に進捗を把握した。進捗が遅れる際に、実現可能性のある対策で進捗遅れの解決を支援した。また、分かりやすい GIF アニメーションを用いて、ヘルプ機能を設計、実装することで、SAA 運用教育での問い合わせ回数削減、教育時間短縮を実現でき、SAA の習得性向上を支援した。

5.1 タイムマネジメント

本プロジェクトにおけるマネジメントでは、筆者はプロジェクトタイムマネジメント[3]を担当し、プロジェクトの進捗管理を行った。筆者はタイムマネージャとして、EVM 手法を用いて進捗管理プロセスを設計し、プロジェクトに適合する出来高測定基準を設定して、統一した出来高測定基準で精度の高い進捗を計測でき、早期に進捗遅れを把握することができた。開発チームの変更を伴い、新たなプロジェクト開発プロセスに適合する進捗管理計画、出来高測定基準がない、進捗報告環境が整備していないため、進捗報告の際に曖昧性や進捗報告もれが存在し、進捗と進捗遅れの把握が難しいという問題があり、筆者はこれらの問題を解決し、精度の高い進捗を計測でき、進捗の遅れを把握することを実現した。

5.1.1 定量的な進捗把握

進捗管理では、進捗管理計画がない、タスク進捗報告の漏れや曖昧性が存在する問題があった。筆者は本チームに参加した際に、開発チームのメンバーの変更を伴い、進捗管理計画ない、進捗管理環境を整備していないため、進捗監視の漏れが発生し、プロジェクトに適合する出来高測定基準を設計していないため、進捗管理する際に、開発チームメンバーが個別に感覚で報告するため、曖昧性があった。

これらの問題で、進捗の遅れが把握しにくかった。具体的に、まず、進捗管理計画がないため、タスクの進捗把握の漏れが発生し、プロジェクトのスコープの変更に対して、計測すべくタスクの範囲調整を行っていないという問題があった。例えば、4月末、開発チームメンバーの変更が発生したため、プロジェクトの開発スコープが変更された。本来、第1イテレーションでは、5つの開発機能を開発する予定がある。変更後、開発する機能が3つになった。しかし、4月の4週目の進捗計測のデータがない、進捗把握の漏れが発生した。開発する機能が5つから3つまで変更したことにも対応せず、進捗把握すべくタスク範囲の調整していないという問題があった。

次に、第1イテレーションでは、タスクの出来高計測基準を設定していない、タスク担当者が進捗率を報告する時、自分の判断で入力するため、個人による誤差が発生し、曖昧性の存在が問題であった。具体的には、図 5-1 第1イテレーションのタスク進捗報告例①に示す。第1イテレーションのある作業では、責任者が自分の判断で進捗を 60%から 90%に更新した後、進捗を 90%から 80%に戻し、最後 80%を 100%に更新した。また、図 5-2 第1イテレーションのタスク進捗報告例②に示す。タスク担当者が自分の判断でタスクの進捗率直接に 10%から 60%に更新され、進捗率が 80%に到達した後、複数回の作業内容が更新された。しかし、進捗率の更新がなかった。このような、タスクの出来高計測基準を設定

しない場合は、進捗報告に曖昧性と恣意性が存在するという問題がある。

Taskwork Summary 27/04/2019 更新 進捗 % を 10 から 30 に変更 ※5/19のチケットです。 このチケットの進捗率は30%に変更されました。 このチケットの進捗率は30%に変更されました。	#9
Taskwork Summary 27/04/2019 更新 進捗 % を 30 から 60 に変更 ※5/20のチケットです。 このチケットの進捗率は60%に変更されました。 このチケットの進捗率は60%に変更されました。	#10
Taskwork Summary 27/04/2019 更新 進捗 % を 60 から 90 に変更 ※5/21のチケットです。 このチケットの進捗率は90%に変更されました。	#11
Taskwork Summary 27/04/2019 更新 このチケットの進捗率は90%に変更されました。 このチケットの進捗率は90%に変更されました。 このチケットの進捗率は90%に変更されました。 このチケットの進捗率は90%に変更されました。	#12
Taskwork Summary 27/04/2019 更新 進捗 % を 90 から 80 に変更 ※5/29の作業分 このチケットの進捗率は80%に変更されました。 このチケットの進捗率は80%に変更されました。 このチケットの進捗率は80%に変更されました。 このチケットの進捗率は80%に変更されました。	#14
Taskwork Summary 27/04/2019 更新 進捗 % を 80 から 100 に変更	#15
Taskwork Summary 27/04/2019 更新 進捗 % を 100 から 100 に変更	#16

図 5-1 第1イテレーションのタスク進捗報告例①

Task: 進捗率の更新	#15
進捗 % を 10 から 60 に変更	
Task: 進捗率の更新	#16
進捗 % を 60 から 80 に変更	
Task: 進捗率の更新	#17
進捗 % を 80 から 100 に変更	
Task: 進捗率の更新	#18
進捗 % を 100 から 120 に変更	
Task: 進捗率の更新	#19
進捗 % を 120 から 140 に変更	
Task: 進捗率の更新	#20
進捗 % を 140 から 160 に変更	
Task: 進捗率の更新	#21
進捗 % を 160 から 180 に変更	

図 5-2 第1イテレーションのタスク進捗報告例②

以上の問題に対して、筆者は EVM[4][5][6]手法を用いて進捗管理プロセスを設計し、進捗管理環境を整備して定期的にプロジェクトの進捗状況を監視することを工夫し、プロジェクトに適合する出来高測定基準を設定と改善を行った。本プロジェクトの第1イテレーションでは、チームメンバーの人数が変更したため、開発のタスクの変更が発生した。変更する前に、プロジェクトが5つの機能を開発する予定であり、変更後、3つの機能を開発することになった。変更を伴い、筆者は進捗管理プロセスを設計し、進捗管理計画書を作成した。Redmineを利用し[7]、各タスクの進捗を監視できる進捗管理環境を整備した。第1イテレーションでの出来高計測基準がない進捗報告する方法を改善し、タスクをレビューの実施する方式に基づいて分類し、タスクを種類毎に出来高測定基準を設定した。

具体的に、まず、進捗管理計画がない問題に対して、EVM 手法を用いて進捗管理プロセスを設計し、進捗管理計画書を作成した。進捗管理計画書で計測すべきタスクを明確し、プロジェクト開発スコープの変更に伴い、管理するタスク範囲の調整を行った。EVM という進捗管理手法を用い、計測して顧客と共有する指標を決め、計測や顧客に共有する頻度を決めた。第1イテレーションで開発メンバーの変更が発生したため、開発する予定の機能が5つから3つに変更した。筆者はプロジェクトスコープが変更した時、新しいプロジェクトスコープに応じて管理すべきタスク一覧表を直し、早期に進捗把握を支援した。

次に、進捗管理環境を整備し、進捗状況を監視してタスクの責任者に進捗報告を催促することで、進捗報告漏れの防止を支援した。進捗管理環境整備では、進捗管理支援するツール Redmine を用いて、進捗管理計画書で明確した計測すべきタスクをチケットとして発行する。例えば、図 5-3 第1イテレーション タスクチケット例に示す、第1イテレーションのタ

スクの 1 部である。Redmine で各タスクの開始日と終了日を設定し、作業中のタスクの進捗更新履歴を監視した。タスクの責任者にタスクの進捗報告を催促した。タスク進捗更新の監視と催促することにより、作業の進捗報告の漏れの防止を支援した。

チケット	ステータス	優先度	題名	担当者	開始日	期日	予定工数	進捗 %
	Closed	High	SY総研プロジェクト	Shimizu, J	2014/04/01	2014/07/25	448.0	
Feature-#707	Closed	Normal	SWOT分析機能	Shimizu, T	2014/04/01	2014/07/18	126.0	
Feature-#198	Closed	Normal	要件定義	Shimizu, T	2014/04/01	2014/04/18	15.0	
Feature-#696	Closed	Normal	技術調査書	Shimizu, T	2014/04/01	2014/04/18	15.0	
Feature-#697	Closed	Normal	SWOTの定義付け	Shimizu, T	2014/04/01	2014/04/07	3.0	
Feature-#697	Closed	Normal	SWOT総括に関する技術調査	Shimizu, T	2014/04/08	2014/04/09	6.0	
Feature-#697	Closed	Normal	技術調査書初版作成	Shimizu, T	2014/04/09	2014/04/10	2.0	
Feature-#697	Closed	Normal	チームレビュー	Shimizu, T	2014/04/09	2014/04/11	2.0	
Feature-#697	Closed	Normal	顧客レビュー	Shimizu, T	2014/04/14	2014/04/18	1.0	
Feature-#697	Closed	Normal	調査書修正	Shimizu, T	2014/04/11	2014/04/18	1.0	
Feature-#697	Closed	Normal	技術調査書顧客承認	Shimizu, T	2014/04/16	2014/04/18	0.0	
Feature-#198	Closed	Normal	設計	Shimizu, T	2014/04/14	2014/04/30	31.0	
Feature-#704	Closed	Normal	外部設計	Shimizu, T	2014/04/14	2014/04/30	24.0	
Feature-#207	Closed	Normal	画面定義書	Shimizu, T	2014/04/14	2014/04/25	13.0	
Feature-#303	Closed	Normal	画面イメージの作成	Shimizu, T	2014/04/14	2014/04/16	3.0	
Feature-#303	Closed	Normal	画面イメージ顧客レビュー	Shimizu, T	2014/04/16	2014/04/18	1.0	
Feature-#303	Closed	Normal	画面定義書初版作成	Shimizu, T	2014/04/21	2014/04/23	5.0	
Feature-#303	Closed	Normal	画面定義書の修正	Shimizu, T	2014/04/23	2014/04/25	2.0	
Feature-#303	Closed	Normal	チームレビュー	Shimizu, T	2014/04/23	2014/04/23	1.0	
Feature-#303	Closed	Normal	顧客レビュー	Shimizu, T	2014/04/23	2014/04/25	1.0	
Feature-#207	Closed	Normal	ユースケース図	Shimizu, T	2014/04/17	2014/04/30	6.0	
Feature-#304	Closed	Normal	ユースケース図の初版作成	Shimizu, T	2014/04/17	2014/04/18	3.0	
Feature-#304	Closed	Normal	チームレビュー	Shimizu, T	2014/04/21	2014/04/23	1.0	
Feature-#304	Closed	Normal	ユースケース図修正	Shimizu, T	2014/04/25	2014/04/25	1.0	
Feature-#304	Closed	Normal	ユースケース図の顧客レビュー	Shimizu, T	2014/04/23	2014/04/25	1.0	
Feature-#304	Closed	Normal	ユースケース図顧客承認	Shimizu, T	2014/04/30	2014/04/30	0.0	
Feature-#207	Closed	Normal	ユースケース記述	Shimizu, T	2014/04/18	2014/04/30	5.0	
Feature-#450	Closed	Normal	ユースケース記述初版作成	Shimizu, T	2014/04/18	2014/04/18	2.0	
Feature-#450	Closed	Normal	チームレビュー	Shimizu, T	2014/04/21	2014/04/23	1.0	
Feature-#450	Closed	Normal	顧客レビュー	Shimizu, T	2014/04/23	2014/04/25	1.0	
Feature-#450	Closed	Normal	ユースケース記述修正	Shimizu, T	2014/04/25	2014/04/25	1.0	
Feature-#450	Closed	Normal	ユースケース記述顧客承認	Shimizu, T	2014/04/25	2014/04/30	0.0	
Feature-#704	Closed	Normal	内部設計	Shimizu, T	2014/04/16	2014/04/30	7.0	
Feature-#208	Closed	Normal	クラス図の作成	Shimizu, T	2014/04/16	2014/04/18	3.0	
Feature-#208	Closed	Normal	チームレビュー	Shimizu, T	2014/04/24	2014/04/24	2.0	
Feature-#208	Closed	Normal	クラス図の修正	Shimizu, T	2014/04/24	2014/04/30	1.0	
Feature-#208	Closed	Normal	クラス図チーム承認	Shimizu, T	2014/04/30	2014/04/30	1.0	
Feature-#198	Closed	Normal	開発	Shimizu, T	2014/05/01	2014/05/26	32.0	
Feature-#705	Closed	Normal	ソースコード	Shimizu, T	2014/05/01	2014/05/26	32.0	
Feature-#198	Closed	Normal	テスト	Shimizu, T	2014/05/26	2014/07/18	48.0	
Feature-#464	Closed	Normal	単体テスト	Shimizu, T	2014/05/26	2014/06/04	19.0	
Feature-#465	Closed	Normal	単体テスト計画書	Shimizu, T	2014/05/26	2014/05/28	6.0	
Feature-#466	Closed	Normal	計画書初版作成	Shimizu, T	2014/05/26	2014/05/28	3.0	
Feature-#466	Closed	Normal	チームレビュー	Shimizu, T	2014/05/26	2014/05/28	2.0	
Feature-#466	Closed	Normal	計画書修正	Shimizu, T	2014/05/26	2014/05/28	1.0	

図 5-3 第 1 イテレーション タスクチケット例

また、進捗管理の曖昧性を削減するため、プロジェクトに適合する出来高計測基準を設定し、改善を行い、タスク進捗報告する際に、出来高計測基準に合わせてタスク進捗率を自動的に補完するように設定をした。進捗報告する際に、タスクの責任者が自分の判断で進捗率を入力するため、曖昧性が存在する。この曖昧性を削減するため、第 2 イテレーションでは、タスクを 3 種類に分け、出来高測定基準を設定した。本プロジェクトのタスクはレビューなし、チームレビューだけ、第 1 イテレーションのチームレビューと顧客レビューあり、3 種類がある。それぞれに出来高測定基準を設定した。

出来高計測基準を設定する際に、第 1 イテレーションの開発実績を参照し、改善を加え、本開発プロジェクトに適性の良い指標で設定した。具体的に、レビューなし、チームレビューだけ、チームレビューと顧客レビュー両方ありの 3 種類のタスクに対し、出来高計測基準は以下の通りに設定した。

① レビューなしタスク

レビューなしのタスクは技術調査とソースコードなど個人作業だけのタスクである。第1イテレーションの1つ機能実装する実績の例を図5-4 第1イテレーションのタスク実績例に示す。実績に時間割を表5.1に示す、投入した工数が46.5時間、その中に明確的に記録したフィードバックの時間（プログラム統合の時間を含め）が6時間、実装時間は40.5時間である、実装時間が大体8割占めると推算し、修正時間が2割と推算した。レビューなし工程の出来高計測基準が表5.2 レビューなしタスク出来高計測基準に示すように設定した。

チケット	コメント	時間	
Feature-#766: プログラムの統合	みんなのプログラムを統合した。	3.00	🔧 🗨
Feature-#414: ソースコード	5/27の作業分。Javadocの挿入やプログラムの見直しを行った。	2.00	🔧 🗨
Feature-#414: ソースコード	5/26の作業分。細かい修正を行っていた	1.00	🔧 🗨
Feature-#414: ソースコード	5/23の作業分。ダイアログ関係を実装していた	3.00	🔧 🗨
Feature-#414: ソースコード	5/22の作業分。作業内容同期機能の実装を進めた	4.00	🔧 🗨
Feature-#414: ソースコード	5/20の作業分。作業内容の同期機能の実装を進めた	7.00	🔧 🗨
Feature-#414: ソースコード	作業内容同期機能の実装を進めた	5.00	🔧 🗨
Feature-#414: ソースコード	wifi対応機能の開発を進めた	3.00	🔧 🗨
Feature-#414: ソースコード	5/14の作業分。	8.00	🔧 🗨
Feature-#414: ソースコード	5/13の作業分。	2.00	🔧 🗨
Feature-#414: ソースコード	5/12の作業分。	0.50	🔧 🗨
Feature-#414: ソースコード	5/8の作業分	1.00	🔧 🗨
Feature-#414: ソースコード	5/7の分、開発環境の設定を行った。	1.00	🔧 🗨
Feature-#414: ソースコード	5/7の作業分である	6.00	🔧 🗨

図 5-4 第1イテレーションのタスク実績例①

表 5.1 実装タスク実績の時間割①

タスク	作成時間 (割合)	修正時間
実装	40.5 時間 (87%)	6 時間(13%)

表 5.2 レビューなしタスク出来高計測基準

工程	進捗率
未着手	0%
作業中	40%
作業完了	80%
フィードバック	90%
チケット完了	100%

② チームレビューだけのタスク

チームレビューだけのタスクは単体テストや結合テストなど開発チームメンバーの合意をもらう必要があるタスクである。第1イテレーションの単体テスト計画書、結合テスト計画書、結合テスト仕様書作成するタスクを例として挙げる。各タスクの実績工数が図5-5 第1イテレーションのタスク実績例②図5-6 第1イテレーションのタスク実績例③と図5-7 第1イテレーションのタスク実績例④に示す、実績の時間割は表5.3 タスク実績の時間割②示す。作成作業時間が大体6割、レビューと修正する時間が4割と推算し、チームレビューだけのタスクの出来高計測基準が表5.4 チーム

レビューだけのタスク出来高計測基準を示すように設定した。

プロジェクト	チケット	コメント	時間	
研究開発プロジェクト	Feature-#468: チームレビュー		1.00	 
研究開発プロジェクト	Feature-#469: 計画書修正		1.00	 
研究開発プロジェクト	Feature-#467: 計画書初版作成		2.00	 
研究開発プロジェクト	Feature-#467: 計画書初版作成		1.00	 

図 5-5 第1イテレーションのタスク実績例②

プロジェクト	チケット	コメント	時間	
研究開発プロジェクト	Project_Management-#484: 計画書修正		1.00	 
研究開発プロジェクト	Project_Management-#491: 計画書初版作成		2.50	 
研究開発プロジェクト	Project_Management-#482: チームレビュー		0.75	 

図 5-6 第1イテレーションのタスク実績例③

プロジェクト	チケット	コメント	時間	
研究開発プロジェクト	Feature-#492: 仕様書修正		0.50	 
研究開発プロジェクト	Feature-#490: チームレビュー		2.00	 
研究開発プロジェクト	Feature-#492: 仕様書修正		1.00	 
研究開発プロジェクト	Feature-#489: 仕様書初版作成		3.50	 

図 5-7 第1イテレーションのタスク実績例④

表 5.3 タスク実績の時間割②

タスク	作成時間 (割合)	チームレビュー時間 (割合)	レビュー修正時間 (割合)
単体テスト計画書	3 時間 (60%)	1 時間 (20%)	1 時間 (20%)
結合テスト計画書	2.5 時間 (59%)	1 時間 (24%)	0.75 時間 (18%)
結合テスト仕様書	3.5 時間 (50%)	2 時間 (29%)	1.5 時間 (21%)

表 5.4 チームレビューだけのタスク出来高計測基準

工程	進捗率
未着手	0%
作業中	30%
作業完了※1	60%
チームレビュー※2	80%
チケット完了	100%

③ チームレビューと顧客レビューがあるタスク

チームレビューと顧客レビュー両方があるタスクは内部設計, 外部設計など顧客と合意するタスクである. 第1イテレーションの画面定義書のタスクを例として挙げる. 実績工数が図 5-8 第1イテレーションのタスク実績例⑤図 5-9 第1イテレーションのタスク実績例⑥に示す, 実績の時間割は表に示す. 作成作業時間が大体 6 割, チームレビューと修正時間が大体 3 割, 顧客レビューと修正時間が大体 1 割になるため, チームレビューと顧客レビュー両方が含めるタスクの出来高計測基準は表 5.5 タスクの実績時間割③に示すように設定した.

プロジェクト	チケット	コメント	時間	
研究開発プロジェクト	Feature-#288: 画面定義書の作成	顧客レビューを行った	2.00	 
研究開発プロジェクト	Feature-#259: 画面定義書の修正	画面定義書の修正を行った	2.00	 
研究開発プロジェクト	Feature-#746: 画面定義書チームレビュー	画面定義書のチームレビューを行った	1.50	 
研究開発プロジェクト	Feature-#745: 画面定義書初版作成	画面定義書の初版を作成した	6.00	 

図 5-8 第1イテレーションのタスク実績例⑤

プロジェクト	チケット	コメント	時間	
研究開発プロジェクト	Project_Management-#828: 結合テスト報告書顧客レビュー		1.00	 
研究開発プロジェクト	Project_Management-#829: 結合テスト報告書修正		1.50	 
研究開発プロジェクト	Project_Management-#721: 結合テスト報告書初版		4.00	 
研究開発プロジェクト	Project_Management-#721: 結合テスト報告書初版		2.00	 
研究開発プロジェクト	Project_Management-#827: 結合テスト報告書チームレビュー		1.00	 

図 5-9 第1イテレーションのタスク実績例⑥

表 5.5 タスクの実績時間割③

タスク	作成時間 (割合)	チームレビュー (割合)	顧客レビュー (割合)
画面定義書	6 時間 (52%)	3.5 時間 (30%)	2 時間 (17%)
結合テスト 報告書	6 時間 (63%)	2.5 時間 (26%)	1 時間 (11%)

表 5.6 チームレビューと顧客レビューがあるタスク出来高計測基準

工程	進捗率
未着手	0%
作業中	30%
作業完了	60%
チームレビュー	80%
顧客レビュー	90%
チケット完了	100%

チームメンバ進捗報告する際に、進捗率入力ミスを防止するため、進捗率の手動入力の代わりに進捗率自動生成するように設定した。第1イテレーションで進捗を報告する際に、図 5-10 第1イテレーション 進捗報告画面示すように、タスクの責任者が手動で進捗率を選択する。しかし、責任者が迷い、入力するミスが発生する場合があった。このミスを防止するため、設定した出来高測定基準に基づいて、図 5-11 タスクステータスの進捗率設定に示すように Redmine でのチケットのステータスに合わせて進捗率を設定した、また、図 5-12 レビューなしタスクステータスの遷移設定図 5-13 チームレビューだけのタスクステータスの遷移設定図 5-14 チームレビューと顧客レビューがあるタスクステータスの遷移設定に示すようにワークフローでタスクの種類を分けてステータス遷移を設定した、第2イテレーションで進捗報告では、タスクの責任者が進捗率を考えせずにタスクのステータスだけ選択することで、タスクの進捗率を更新できる。各ステータスの移行先を図 5-15 第2イテレーション 進捗報告画面示したように設定することで、進捗報告のフォールバックも防止できた。進捗報告のミス防止でき、曖昧性削減ができた。定量的に進捗を把握できた。

図 5-10 第1イテレーション 進捗報告画面

ステータス	進捗率	デフォルト値	終了したチケット	ソート	
未着手	0	✓		▲ ▼	削除
作業中	30			▲ ▼	削除
作業完了	60			▲ ▼	削除
作業中(レビューなし)	40			▲ ▼	削除
作業完了(レビューなし)	80			▲ ▼	削除
チームレビュー	70			▲ ▼	削除
顧客レビュー	80			▲ ▼	削除
フィードバック	90			▲ ▼	削除
却下	0		✓	▲ ▼	削除
完了	100		✓	▲ ▼	削除

図 5-11 タスクステータスの進捗率設定

現在のステータス	未着手	作業中(レビューなし)	作業完了(レビューなし)	フィードバック	却下	完了
未着手	✓					
作業中(レビューなし)		✓				
作業完了(レビューなし)			✓			
フィードバック				✓		
却下					✓	
完了						✓

図 5-12 レビューなしタスクステータスの遷移設定

ロール: Developer トリガー: Feature(No review) 編集 このトリガーで使われているステータスのみ表示する

現在のステータス

ステータスの移行先

	未着手	作業中(レビューなし)	作業完了(レビューなし)	フィードバック	却下	完了
未着手	☒	☒	☒	☒	☒	☒
作業中(レビューなし)	☐	☒	☒	☒	☒	☒
作業完了(レビューなし)	☐	☐	☒	☒	☒	☒
フィードバック	☐	☐	☐	☒	☒	☒
却下	☒	☒	☒	☒	☒	☒
完了	☐	☐	☐	☐	☒	☒

図 5-13 チームレビューだけのタスクステータスの遷移設定

ロール: Developer | トラッカー: Feature(All Review) | 編集 | このトラッカーで使われているステータスのみ表示する

現在のステータス

ステータスの移行先

	未着手	作業中	作業完了	チームレビュー	顧客レビュー	フィードバック	却下	完了
未着手	☒	☒	☒	☒	☒	☒	☒	☒
作業中	☐	☒	☒	☒	☒	☒	☒	☒
作業完了	☐	☐	☒	☒	☒	☒	☒	☒
チームレビュー	☐	☐	☐	☒	☒	☒	☒	☒
顧客レビュー	☐	☐	☐	☐	☒	☒	☒	☒
フィードバック	☐	☐	☐	☐	☐	☒	☒	☒
却下	☒	☒	☒	☒	☒	☒	☒	☒
完了	☐	☐	☐	☐	☐	☐	☒	☒

図 5-14 チームレビューと顧客レビューがあるタスクステータスの遷移設定

プロパティの変更

トランカ: Feature(Team Review)

名前: 結合テスト仕様書

説明:

ステータス: 完了

優先度: 未着手

担当者: 作業中

チームレビュー: 作業完了

顧客レビュー:

却下:

観ダクト: 398

開始日: 2014-11-16

期日: 2014-11-19

予定工数: 8.0 時間

時間記録

作業時間の記録: 時間

コメント:

活動: 進んでください

図 5-15 第 2 イテレーション 進捗報告画面

筆者は進捗報告の漏れを防止し、進捗報告の曖昧性を減少させ、定量的に進捗を把握した。進捗状況をモニタリングすること。進捗管理環境を整備することで、各タスクの進捗状況を監視し、把握できた。開発メンバの進捗報告状況を監視し、報告すべきタスクを開発チームメンバに提示することで、進捗報告の漏れを防止した。また、手動入力の代わりに進捗率を自動補完するように設定することで、開発メンバ進捗報告する時の迷いを消し、進捗報告の曖昧性を削減できた。この精度の高い進捗をモニタリングすることで、早期に進捗遅れの発見を支援した。

5.1.2 進捗遅れへの対策

進捗が遅延した際に、複数の EVM 指標[4][5][6]を掛け合わせて遅延の要因を分析して、プロジェクトに実現可能性がある対策を提案と検証することで、進捗遅れの解決を支援した。進捗が遅延した際に、遅延の原因やプロジェクトに与える影響の把握が難しい、対策の実現可能性及び実施の結果検証を把握することが難しい。本プロジェクトは少人数のメンバが開発を行うため、投入できる工数が限られている。また、開発チームメンバは本システム開発以外の活動があるため、対策を実施するタイミングについても配慮する、そして、遅れた進捗に対策を考える時、実現可能性のある対策を作成するため、定量的な対策を作成する必要がある。このため、早期的に進捗の遅れを発見し、定量的に進捗遅れの程度を把握し、実現可能性がある対策を作成する必要がある。

本プロジェクトでは、進捗遅れが複数回発生し、これらの進捗遅れを解決するために、それぞれに原因やプロジェクトに与える影響を把握した。プロジェクト進捗遅れの原因について、日経 SYSTEMS の記事で見積り精度低さ、要件定義の不備、進捗遅れを発見できないこと、プロジェクトの課題が放置されること、チームメンバが進捗遅れに対す意識が必ずしも高くないことなどを指摘された[19][21]。本プロジェクトでは、進捗遅れが複数回発生した。筆者は EVM を監視し、EVM 指標を掛け合わせて本プロジェクトの進捗遅れの原因を分析した。本プロジェクトにおける進捗遅れの原因が主に進捗把握不足と要件定義不備であった。

開発チームメンバ変更による進捗遅れ及び対策[16]

EVM を用いて、進捗が遅れた程度を把握した。本プロジェクト開発の途中では、開発チームメンバの変更が発生した。プロジェクトが 4 月から開始し、4 月末、研究開発プロジェクトテーマ調整があったため、図 5-16 開発 1 ヶ月間の EVM と図 5-17 開発 1 ヶ月間 SV と CV に示す、4 月末の EVM により、計画出来高（Planned Value 以下 PV と略称する）が高騰し、実績出来高（Earned Value 以下 EV と略称する）と実績コスト（Actual Cost 以下 AC と略称する）の記録がないという進捗遅れを発見した。スケジュール差異（Schedule Variance 以下 SV と略称する）が急に下がって、進捗には大きな遅れが発生した。その差異を計測すると 136.5 時間の作業量が遅れていると把握できた。

把握して遅れの程度をチームに共有し、開発計画調整という対策を提案した。対策の実行した結果をチームや顧客と共有した。筆者は遅れている進捗の作業量を 2 種類に分けて報告した。その中、開発メンバの変更で進めないタスクが 78 時間の遅れがあり、進んでいるタスクは書類ドキュメント化について 58.5 時間の作業量が遅れていると計算した。この定量把握した遅れ状況をチームリーダーに報告し、開発計画の調整を提案した。リーダーが顧客や開発メンバと合意し、新しい開発計画を策定した。筆者はその新しい計画に基づいて、5 月 1 日に EVM に変更した開発計画を反映することで、その時点の進捗遅れを定量的に把握でき、チームメンバや顧客と共有した。

計画変更対策で解決できない進捗遅れを発見した。図 5-16 開発 1 ヶ月間の EVM と図 5-17 開発 1 ヶ月間 SV と CV に示したように、5 月 1 日、開発計画変更した後、進捗の遅れがまだ存在するという問題がある。開発計画の変更では、転出したメンバのタスクが中止された。進んでいるタスクについては主に顧客と合意した設計内容を成果物としてまとめる作業である。例えば、機能毎にクラス図が作成したが、納品のためのシステム全体クラス図をまとめるタスクや設計書類ドキュメント化するタスクがまだ遅れている。

残った進捗遅れの程度を定量的に把握して、定量的な対策を提案した。これらの進捗遅れを解決するために、58.5 時間の工数を増やす必要がある。しかし、開発チームメンバはプロジェクト開発以外、就職や授業など活動があるため、これ以上の工数を増やすことができないと判断した。また、開発計画変化後の進捗遅れのアクティビティ順位から分析し、これらの遅れは機能実装工程に影響を与えないため、5 月 1 日からプロジェクトが機能実装工程既を開始した。これらの問題を考えた上で対策を作成して、開発チームに提案した。提案するポイント[15][17]は以下の 3 点がある。

- ① その時点の作業効率で考えると 58.5 時間の工数を増やす必要がある。増やす期間について、開発チームのメンバは 4 人がいるため、1 人平均 15 時間の工数を増やす必要がある。開発チーム週毎に作業するためのコアタイム 3 回がある、まとめると月毎に 13 回のコアタイムがある。たとえ工数を増やす期間を 0.5 月間に設定する場合は、コアタ

イムごとに 2.5 時間ぐらいの工数を増やすことになる。

- ② 工数を増やすタイミングについて、開発チームメンバ授業や就活などの活動を避けると考え、7 月中間報告完了後の時期を選定した。開発メンバの授業が春学期の AB であり、7 月に大体試験が完了する。また、7 月がプロジェクトの評価工程に入るため、チーム全体の作業量が少ないことも理由の一つである。
- ③ 提案を実行する前、定量的に進捗遅れを許容する。対策が解決する進捗遅れの程度が 58.5 時間であるので、対策を実施するまで進捗遅れが 58.5 以下になること、すなわち SV が -58500 以上であることを許容する。SV が -58500 以下になると新たな進捗遅れの警告を出す。

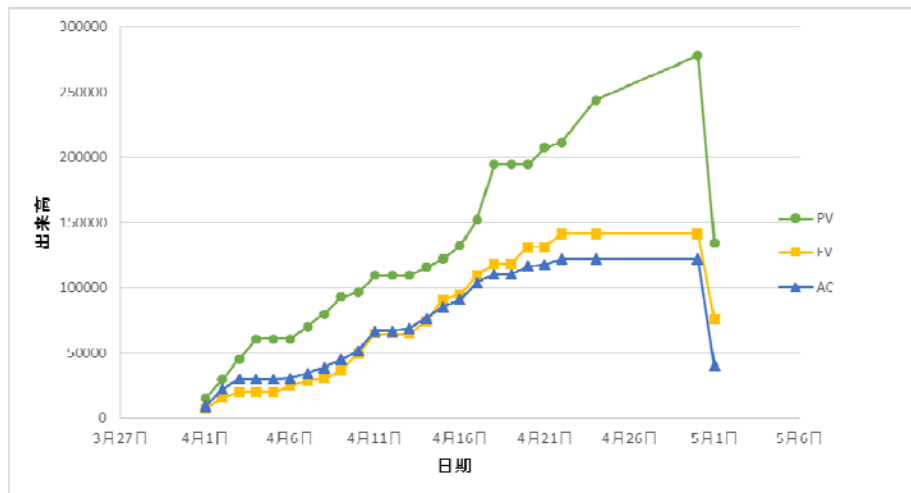


図 5-16 開発 1 ヶ月間の EVM

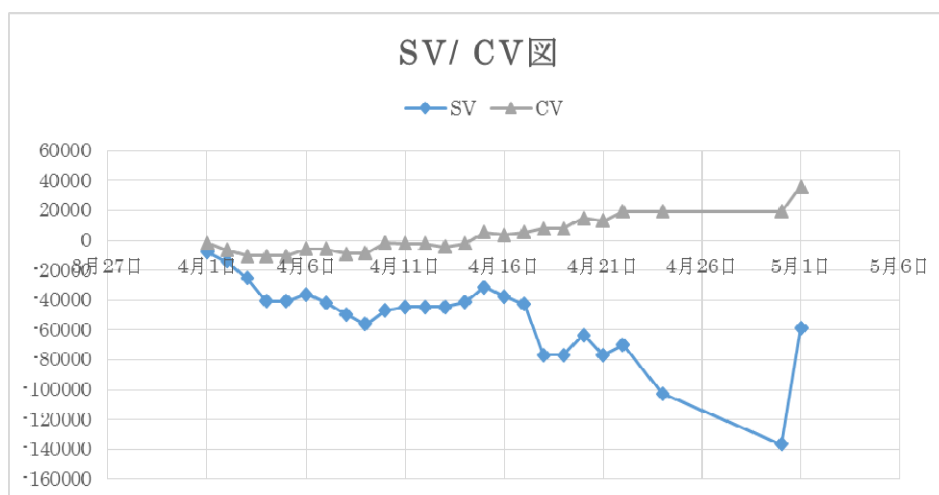


図 5-17 開発 1 ヶ月間 SV と CV

機能実装工程における EVM 上進捗遅れの原因分析及び対策

5月1日から始まった機能実装工程では、計測した EVM 指標から進捗遅れを発見した。EVM に示すように、5月1日から EV の増長が遅い、進めないところも発見された、に示すように SV が許容する範囲に超過し、-97700 に到達した。筆者が進捗には大きな遅延が発生していると判断した。

進捗報告不足という遅れの原因を判明した。遅れに対して、筆者は機能実装タスクの進捗報告状況を確認し、進捗率報告不足[20][19][21]が原因だと判断した。5.1.1 節定量的な進捗把握で説明した、タスクの粒度が大きい、開発チームメンバの進捗報告の曖昧性が存在するため、進捗の報告が不十分であり、計測データに誤差が発生したため、EVM から進捗の把握ができなくなった。

タスクの責任者を催促することにより、チームメンバの進捗報告を支援し、進捗状況を改善した。進捗遅れの原因を分析した結果、EVM 上進捗遅れが見えた。しかし、機能実装工程の作業が遅れていないと究明し、開発計画を変更する必要がないと判断した。進捗報告状況を改善する対策を提案して実施した。

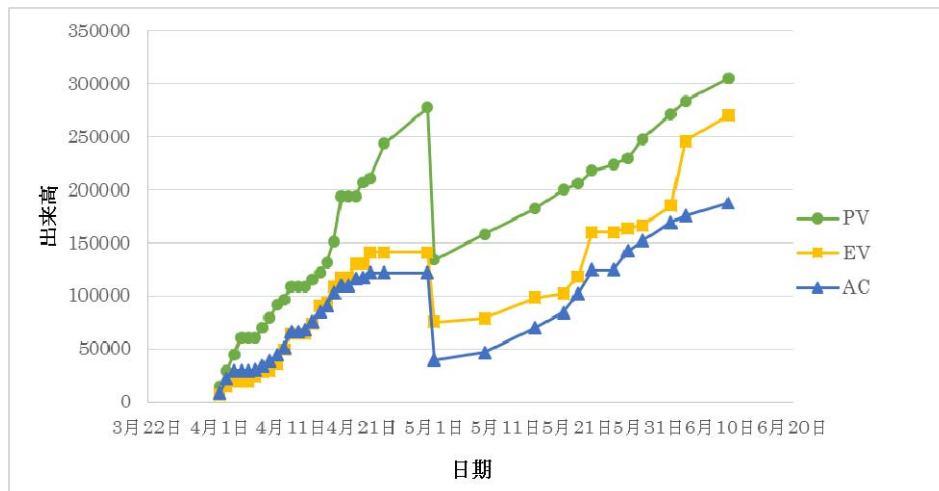


図 5-18 単体テストまでの EVM

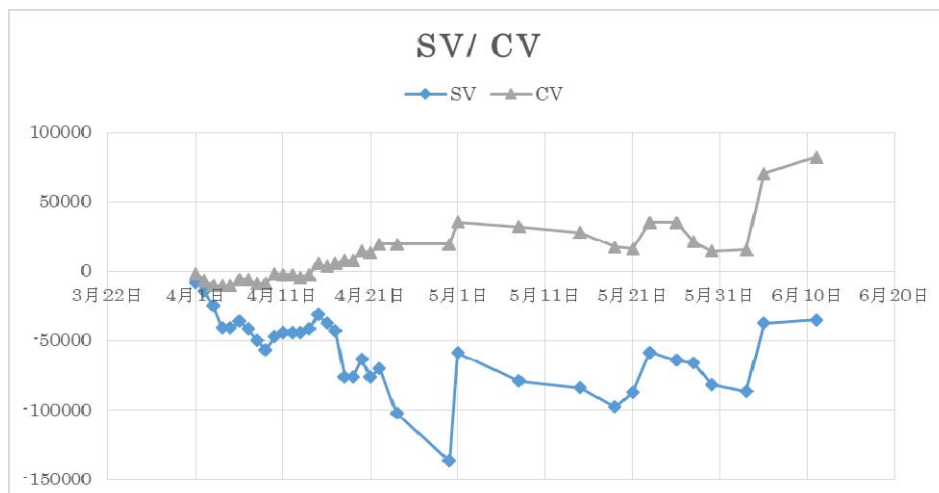


図 5-19 単体テストまでの SV と CV

進捗遅れの程度の把握及び対策

本プロジェクトの結合テスト工程で、進捗遅れが発生し、筆者はEVM指標を分析することで、この遅れがプロジェクトに与える影響を把握した。本プロジェクトの結合テスト工程では、要件定義で計画しなかった結合テストを再実施というタスクが発生し、中間報告のタスクでのACが増加した。筆者は計測したEVM指標を用い、プロジェクトに与える影響を定量的に把握し、進捗遅れを把握した。図 5-20 第1イテレーションのEVMに示すように、EVMから問題が発見できなかった。から、結合テスト後半ではSVが下がった傾向があった、しかし、許容の警告値を超えていないため、SV指標からプロジェクトに与える影響を把握できない。に示す、出来高パーセント(PC)から、結合テスト工程の後半では、完成度が67.4%から64.4%に下がったことを発見した。その原因は結合テスト再実施と中間報告のため、タスクが増やしたことである。また、完了期間予測(TEAC)指標を用いて、第1イテレーションの納期を予測し、予定4ヶ月間と比べ結合テスト工程当時の実績が4.67ヶ月間、2週間程度を遅延する傾向を把握できた。

進捗遅れの程度を把握した上で、本プロジェクトの開発工数を配慮して、開発期間延長及び投入コスト増加並行の対策をチームに提案した。プロジェクトの進捗遅れの問題を解決する方法は投入するコストを増やすと納期延長の複数の手法を提案した[19][20][21]。結果として、結合テストの期間が1週間を延長され、結合テスト工程に人員を追加され、評価フェーズの並行進めることで、納期延長をせず、開発を完了できた。

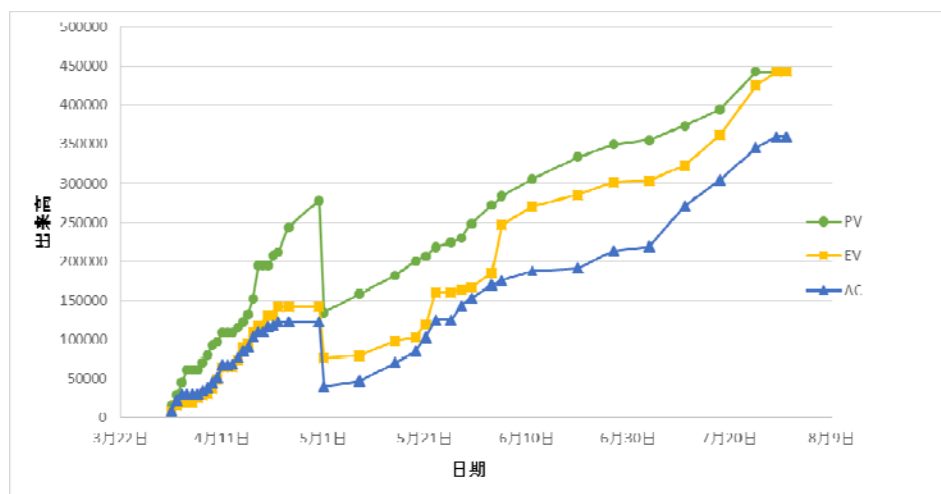


図 5-20 第1イテレーションのEVM

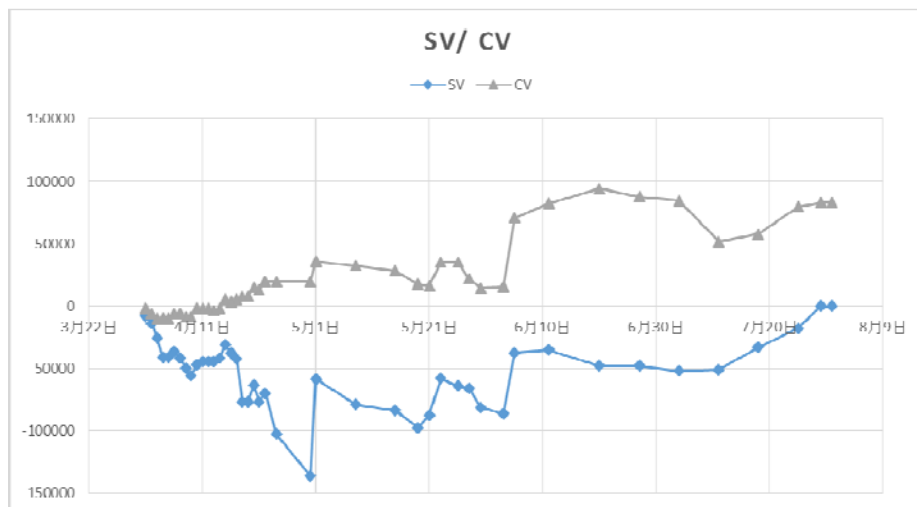


図 5-21 第1イテレーションのSVとCV

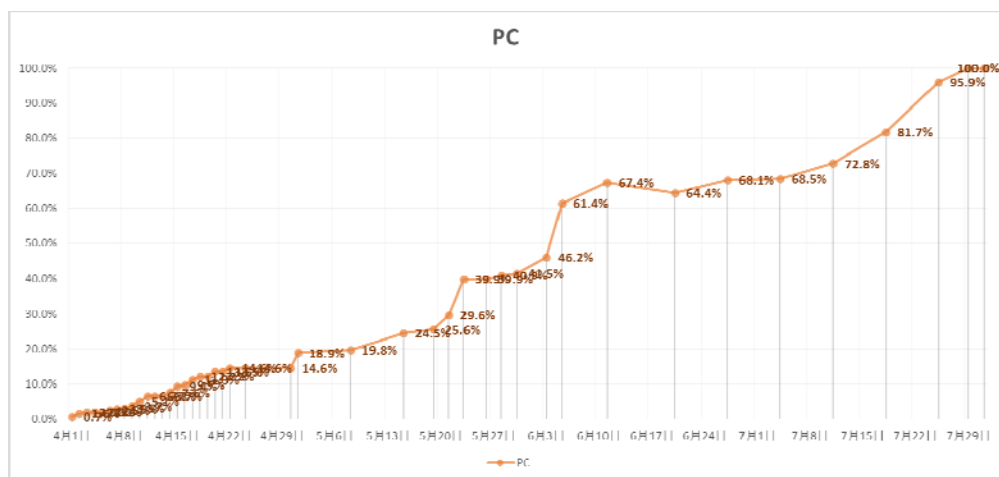


図 5-22 第1イテレーションのPC

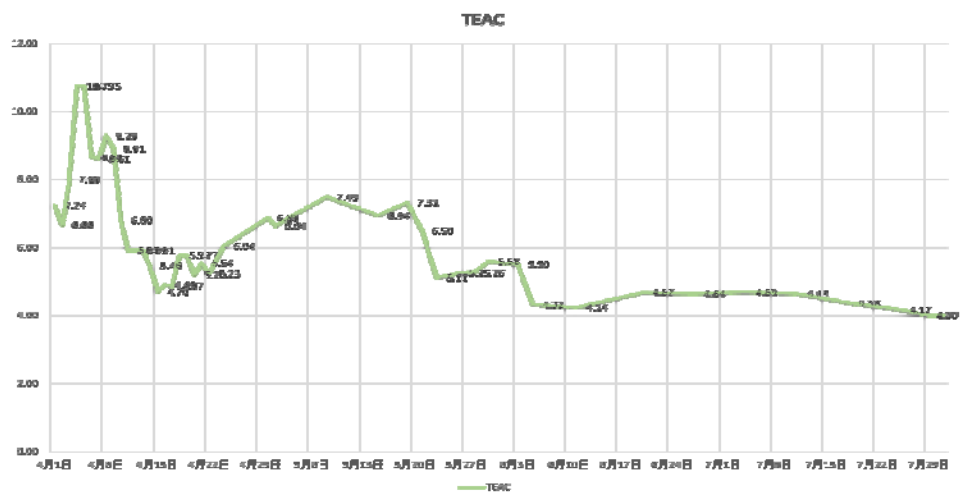


図 5-23 第1イテレーションのTEAC

5.2 ヘルプ機能の設計と開発

SWOT 分析実施者がシステムを利用する際に問い合わせの回数が多いという問題に対し、文書の代わりに理解しやすいアニメーションを用いてヘルプ機能を実装することで、問い合わせの回数を削減し、運用習熟時間を削減でき、システムの習得性が向上できた。

5.2.1 ヘルプ機能提案

第 1 イテレーションの評価では、顧客がシステム操作を学習する際に、システムの操作が覚えなくて同じ操作に対して複数回を問い合わせたという問題があった。本システムを利用して SWOT 分析を実施する際に、システムを利用するための教育時間を設けた。指導者はマニュアルを渡して、操作を演示しながらシステムの使い方を説明した。しかし、SWOT 分析参加者はシステム学習後、個別にシステムを使用する時、操作を覚えられない、マニュアルがわからないため、指導者に問い合わせた。表 5.7 第 1 イテレーション評価での操作に関する問い合わせ回数で 3 名 SWOT 分析参加者の問い合わせ回数を記録した。第 1 イテレーションの評価では、主に SWOT 分析図の編集機能や SWOT 分析図の保存と読み込み機能について評価を行うため、この二つの機能に関する操作だけ記録した。指導者がいない場合は、参加者が操作を進めない可能性がある。

表 5.7 第 1 イテレーション評価での操作に関する問い合わせ回数

操作	問い合わせ回数
付箋作成	2
付箋修正	1
色変更	3
付箋削除	0
各領域の切り替え	2
グループ化	5
ファイル保存	3

顧客の操作が気の利いたガイドが出るという要望に対し、ヘルプの運用教育や操作支援について調査を行い、分かりやすいアニメーションを用いるヘルプ機能を提案した。システムの運用教育では、SAA の利用者にシステムの操作を教える際に、マニュアルやヘルプ機能を利用することが多い。マニュアルの場合は、利用者に伝えるべき内容が多い、1 枚にまとめることができないため、マニュアルの文書が厚くなる。厚いマニュアルから利用者が読むべき箇所を探す手間がかかる問題がある[10][11]。マニュアル代わりに、システムに組み込むヘルプ機能を利用場合が多いである。

ヘルプ機能について調査を行い、分かりやすいアニメーションを用いるヘルプ機能を提案した。まず、SAA が Android アプリケーションであるため、Android でよく利用されるヘルプ機能について調査した。テキストで記述されるヘルプコンテンツ、操作を導くチュートリアル、操作方法を演示するアニメーションなどのヘルプ機能が存在する。その中に、テキストで記述されるヘルプコンテンツは単純に操作の方法を知りたいユーザにとって負担となる[9]。Android ゲームアプリケーションでよく利用される操作を導くチュートリアルは再生

途中で停止できないという問題が存在する。以上の問題を考え、操作分かりやすく操作を演示するアニメーションを用いるヘルプ機能を顧客に提案した。

5.2.2 ヘルプ機能の設計と実装

ヘルプ機能の設計では、GIF アニメーションを利用してヘルプ機能を設計し、関連技術を調査して実装した。ヘルプ機能のアニメーション形式を設計する際に、アニメーション動画を代わりに、複数のフレームを順に表示する GIF アニメーションを用いて設計した。GIF アニメーションを選択理由としては、アニメーションのフレーム数やフレームを切り替える待ち時間を自由に設定できるため、サイズが小さいアニメーションを実現できる。顧客の操作が気の利いたガイドが出るという要望に満たすために、既存 SAA のデザインに合わせ、SAA のメニューにヘルプ機能のボタンを追加した。SAA の利用者がアクセスしやすい仕様で設計した。具体的なイメージは、図 5-24 ヘルプ機能画面イメージと図 5-25 ヘルプ機能ユースケースに示す。SAA メニュー中に、ヘルプ機能のボタンを設置し、ボタンをタップすると操作名を書いた項目ボタンが表示され、項目ボタンをタップするとその項目が書いた操作の演示アニメーションが循環に再生され、SAA 利用者がアニメーションを見ることで、操作の方法を学習できる。アニメーションの再生する画面をタップすると、演示するアニメーションが閉じられ、SAA 利用者が元に作業を継続することができる。



図 5-24 ヘルプ機能画面イメージ

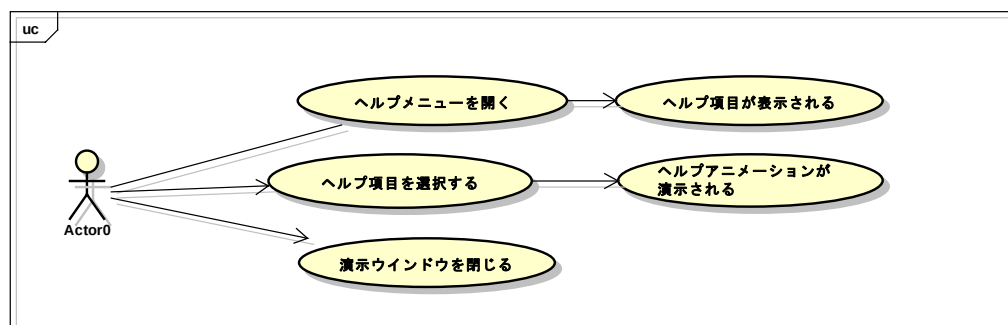


図 5-25 ヘルプ機能ユースケース

設計したヘルプ機能を実装する際に、2つの問題が発生した。ヘルプ機能が主に2つの仕組みを設計した。1つ目はGIFアニメーションを再生する仕組みである、2つ目は再生するGIFアニメーションをアプリケーション画面上で配置し、削除する仕組みである。2つの仕組みを開発する際に、問題が発生した。

まず、AndroidがGIFファイルのエンコーダができない、すなわち、GIFアニメーションを再生できないという問題があった。この問題に対して筆者はGIFアニメーションを再生できる方法を調査し、AndroidのMovieを利用してGIFファイルの再生を実現した、AndroidのMovieクラスを利用することで、動的に画像をストリームとして読み込み、画像配列として扱い、順次再生することができる。筆者はGIF再生するためのオリジナルのGIFImageViewクラスを作成し、GIFアニメーション再生することを実現した。

次に、GIFアニメーションViewを管理するクラスを実装する際に、オリジナルで定義したGIFアニメーション再生するViewが描画されないという問題が発生した。ヘルプ機能のアニメーションを再生する際に、誤操作を防止するため、画面フォーカスを実現でき、他操作に影響を与えない、独立でViewを管理できるWindowManagerクラスを用いた。

WindowManagerクラスを利用してImageViewを管理し、画面上の描画ができた、しかし、筆者はWindowManagerクラスを利用して、作成したGIFアニメーションを再生するオリジナルのGIFImageViewを管理する場合は、GIFアニメーションが表示されなかった。デバッグする際に、エラーも検出されなかった。筆者は作成したオリジナルAndroidのGIFImageViewはImageViewから継承し、ImageViewにMovieクラスをインポートし、ImageViewのOnDrawメソッドをオーバーライドして作成した。同じ仕組みで、継承先のImageViewは問題なく描画された、しかし、筆者作成したオリジナルのGIFImageViewでは問題が発生した。

筆者はAndroidの開発ドキュメントや開発記事を調査し、この問題を究明して解決し、オリジナルのGIFImageViewの描画を実現した。画像が描画できない問題に対して、筆者はAndroid画像描画に関する技術を調査した。その際に、AndroidのActivity Lifecycleについて調査し、Viewのライフサイクル[8]について調査した。図 5-26 AndroidViewのライフサイクルに示す、AndroidはViewを生成やViewの内容を更新する際に、OnDraw()メソッドを呼び出す前に、onMeasure()メソッドとonLayout()メソッドを呼び出す。この2つのメソッドについて、Androidの開発ドキュメントを調べて、onMeasure()メソッドを呼び出される際に、生成するViewがonMeasure()メソッドで子Viewのサイズを測る上で、自身のサイズを確定するプロセスがある。WindowManagerクラスを用いて直接にViewを追加、管理する際に、生成したViewは子Viewが存在しないため、自身サイズの設定が失敗したため、オリジナルのGIFImageViewクラスが描画される際に、正しいサイズで描画されていないことを究明した。この問題を解決するために、オリジナルのGIFImageViewクラスにonMeasure()メソッドをオーバーライドして、GIFImageViewのサイズを設定することで、問題を解決した。

測定する内容：

- ① 被験者が SAA 操作についての問い合わせ回数.
- ② テーマ作成開始からテーマの完成までの作業時間.
- ③ 被験者アンケート

被験者アンケートでは，グループ 1 とグループ 2 を分けて実施した.

グループ 1 の被験者に対して年齢層，性別，コンピュータ利用経験，スマートデバイス利用経験，実施したテーマのアンケートを実施した.

グループ 1 の被験者に対して年齢層，性別，コンピュータ利用経験，スマートデバイス利用経験，実施したテーマ以外，ヘルプ機能を利用した感想（役立つ，解かりやすい，利用したいか）についてアンケートで記録した.

評価項目：

- ① ヘルプ機能を利用すると運用教育の時間を節約できること.
- ② ヘルプ機能を利用すると SAA の操作に関する問い合わせ回数を削減できること.



図 5-27 筑波大学 SWOT 分析図の例

表 5.8 自由テーマでの必須作成項目

項目内容	項目数
強み	4～5項目
弱み	2～3項目
機会	3～4項目
脅威	1～3項目
グループ	1～2グループ
関連線	2～3関連線
色付け	3種類以上

実験データと結果

今回の実験で合計6名の方の実験を実施し、データ分析を行った。アンケートと計測したデータが表 5.9 グループ1のアンケートと計測したデータと

表 5.10 グループ2のアンケートと計測したデータに示す。

表 5.9 グループ1のアンケートと計測したデータ

年齢層	10代～30代	10代～30代	10代～30代
性別	男性	女性	女性
コンピュータの利用経験	5年～10年	5年～10年	10年以上
スマートデバイス利用経験 (スマートフォン, タブレットなど)	1年～3年	1年～3年	1年～3年
ヘルプ機能を利用したか	利用していない	利用していない	利用していない
ヘルプ機能が役立つと思いますか			
ヘルプ機能がわかりやすいですか			
操作がわからない時, ヘルプ機能使いたいですか			
実施したテーマは何ですか	既存例のつくば大学 SWOT 分析	既存例のつくば大学 SWOT 分析	既存例のつくば大学 SWOT 分析
教育時間	10min	10min	10min
作業時間	18m23s	20m5s	23m10s
お問い合わせ回数	5	7	8

表 5.10 グループ 2 のアンケートと計測したデータ

年齢層	10 代～30 代	40 代～60 代	10 代～30 代
性別	女性	女性	男性
コンピュータの利用経験	10 年以上	10 年以上	10 年以上
スマートデバイス利用経験 (スマートフォン, タブレットなど)	3 年以上	1 年未満	3 年以上
ヘルプ機能を利用したか	利用した	利用した	利用した
ヘルプ機能が役立つと思いますか	そう思う	そう思う	そう思う
ヘルプ機能がわかりやすいですか	そう思う	そう思う	そう思う
操作がわからない時, ヘルプ機能使いたいですか	そう思う	そう思う	そう思う
実施したテーマは何ですか	既存例のつくば大学 SWOT 分析	自由テーマ	既存例のつくば大学 SWOT 分析
教育時間	0min	0min	0min
作業時間	15m12s	30m51s	12m24s
お問い合わせ回数	2	1	1

ヘルプ機能を利用することで運行教育時間を削減できることを検証できた。以上 6 名の被験者が全員テーマを完成したため、被験者全員が SAA を習熟できたと証明できた。グループ 1 平均 10 分間の運用教育時間に対してグループ 2 は運用教育時間が 0 であるため、ヘルプ機能を利用すると、運用教育の操作説明がなくても、SAA 利用者が SAA の操作を習得できた。

ヘルプ機能を利用することで、問い合わせの回数を削減できると検証できた。グループ 1 の SAA 操作に関する問い合わせ回数が多いことに対して、グループ 2 の被験者が SAA を習得する際に、ヘルプアニメーションを見ながら操作を進むため、問い合わせの回数が少ないである、また、表 5.7 第 1 イテレーション評価での操作に関する問い合わせ回数に比べ、グループ 2 の被験者の問い合わせ回数が少なかった。

第6章 まとめ

本報告書では、「高度 IT 人材育成のための実践的ソフトウェア開発専修プログラム」で実施される特定課題研究開発プロジェクトにおいて、SWOT 分析の実施での課題を抱えている顧客に要求を分析し、要件定義から納品までのシステム開発を行い、顧客の要求を満たし、筆者がプロジェクトマネジメントの取り組み、ソフトウェア開発で得られた経験について報告した。

本プロジェクトでは、プロジェクトタイムマネジメントを担当し、進捗管理することにより、小規模ソフトウェア開発の進め方を経験でき、チームメンバと密なコミュニケーションを通して、早期にソフトウェア開発における開発計画変更、進捗遅れなどの問題の解決を支援でき、ソフトウェア開発は円滑に進行することへ貢献した。しかし、開発工程以外、日々発生している予測できないタスクの対応が不足、プロジェクトに与える影響を把握出来ないという課題が残った。また、第 2 イテレーションで、EVM 手法をアジャイル開発への適用が実現できないという課題が残った。

また、本プロジェクトでのヘルプ機能の設計開発することにより、Android アプリケーション開発のスキルを身に付けることができ、システム習得性向上へ支援することにより、システム運用フェーズでの運用教育コスト削減を実現できた。開発した機能の実証実験を行うことにより、システムの評価を実践した。

本プロジェクトを通し、チームワークの大切さを実感し、チームメンバと相談しながら、問題解決することの重要性を経験できた。今後の活動に、このようなチームワークをいかしていきたい。

謝辞

博士前期課題期間, 指導教員である筑波大学システム情報工学研究科の田中二郎先生には, 大変お世話になりました. 特定課題研究報告書の執筆において丁寧な御指導, 御助言を頂き, 深く感謝致します.

また, 課題担当教員の早瀬康裕助教, 顧客の山戸昭三研究員, 御多忙の中, 御アドバイスを多数頂き, 丁寧な御指導を頂きました. 誠にありがとうございました.

本プロジェクトのチームメンバである, 大塚優樹君, 鈴木良生君, 田中靖成君には, お世話になりました. 感謝致します.

最後に, これまで私を支えてくれた家族と友人, お世話になった方に, 心より感謝いたします.

参考文献

- [1] 大住莊四郎. 自治体への戦略マネジメントモデルの適用 SWOT 分析を中心に. ESRI DISCUSSION PAPER SERIES No.157,2,2006.
- [2] 中塚雅也,深町拓司,星野敏. SWOT 分析を応用したワークショップ手法の開発. 神戸大学大学院自然科学研究科紀要. B /神戸大学大学院自然科学研究科編,ISSN 0287-6515,3(25),pp.129-137. 2007.
- [3] Project Management Institute,プロジェクトマネジメント知識体系ガイド第 4 版,2008.
- [4] 山戸昭三,永池恒一. WBS/EVM による IT プロジェクトマネジメント. ソフトリサーチセンター,2009.
- [5] JavierPajares,AdolfoLópez-Paredes. An extension of the EVM analysis for project monitoring: the Cost Control Index and the Schedule Control Index. International journal of project management 29(5),pp.615-621. 2011.
- [6] WaltLipke,OferZwikaël,KymHenderson,FrankAnbari. Prediction of project outcome: The application of statistical methods to earned value management and earned schedule performance indexes. International journal of project management 27(4), pp.400-407. 2009.
- [7] 木崎悟,成田亮,丸山英通,中鉢欣秀. ソフトウェアプロセス・プロジェクト管理. 情報処理学会研究報告 Vol.2011-EMB-21,No.1,2010.
- [8] How Android Draws Views.
<http://developer.android.com/intl/zh-cn/guide/topics/ui/how-android-draws.html>.
- [9] 三浦元喜,田中二郎. Java アプレットののためのアニメーションヘルプシステム. 情報処理学会論文誌プログラミング(PRO),(41)SIG04 (PRO7),pp.56-64. 2000.
- [10] 柿間俊高,村松幸樹,杉本徹. 知的ヘルプシステムのための意味を考慮したテキストマッチング手法の改良と評価. 第 74 回全国大会講演論文集(1)(2012), pp.531-533.
- [11] 中谷桃子,大野健彦,片桐有理佳,中根愛,橋本周. ユーザを迷わせないマニュアルデザイン. 研究報告ヒューマンコンピュータインタラクション(HCI), (18)2010-HCI-139, pp.1-8,2010.
- [12] アイデア粘土細工ソフトウェア・マインドピース.
<http://mindp.kantetsu.com/index.html>,(参照 2014-10-31)
- [13] SWOT 分析テンプレート&フォーマット集. 2011-05-03.
<http://matome.naver.jp/odai/2130369384487847601>,(参照 2014-10-31)
- [14] Best to do list app for Windows, iOS. Plan differently. <http://www.appfluence.com/>,(参照 2014-10-31)
- [15] 福田祥久. EVM を用いた進捗マネジメントとそのポイント(<特集>EVM と進捗マネジメント). プロジェクトマネジメント学会誌,5,3,pp.22-26.2003.
- [16] 鎌田瞬,大南正裕,山内道夫,山川直樹,千種実. プロジェクト状況悪化早期検知のための一考察(<特集>プロジェクトの計画と評価). プロジェクトマネジメント学会誌 16,3-9-4,2014.
- [17] 米花幸男. システム開発における EVM 適用への実用的なアプローチ. プロジェクトマネジメント学会研究発表大会予稿集,pp.25-30.2003.

- [18] 駒谷昇一,田中二郎,北川博之. 筑波大学における高度 IT 人材育成のための実践的ソフトウェア開発専修プログラム. 工学教育,日本工学教育協会,57,4,pp.92-98. 2009.
- [19] プロジェクトの進捗遅れはなくせる.
<http://itpro.nikkeibp.co.jp/article/Watcher/20100721/350517/>(参照 2015-1-5)
- [20] 橋本勝. ユーザーのためのプロジェクト実践塾進捗マネジメントで遅延を未然に防ぐ. 日経コンピュータ,638,pp.240-243. 2005.
- [21] 進捗遅れを防ぐマネジメント 危機感共有し,遅れ見つけ,課題を解消する現場の工夫(特集 進捗遅れをなくそう--危機感高め,素早く課題を解消).日経 SYSTEMS,208,pp.22-31. 2010.