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

# テニススクール経営改革のための IT ソリューション提供

## －アジャイル型システム開発における品質マネジメントの実践－

有田正信

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

２０１３年　３月

# 概要

本報告書は，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 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いて，筆者を含む学生 5 人が実施した研究開発プロジェクトの成果をまとめたものである．プロジェクトチームは学生メンバ 5 人と課題担当教員の山戸昭三教授の頭文字を並べ，チーム SANITY と名付けた．

本プロジェクトは，つくば市内のテニススクール「T-1 インドアテニススクール」の経営者であられる末満裕之様と奥様の末満弘江様を顧客として，超上流工程から下流工程までの一貫した IT ソリューションの提供を目指した．

本プロジェクトの方針として「顧客との密なコミュニケーション」を掲げ，経営戦略の立案から，ビジネスプロセスの改善，及び IT ソリューションの納品・評価まで全 19 回の顧客ミーティングを開催した．

本報告書では，実際に顧客に対して行った取り組みをフェーズ毎に記した．また，各章の終わりには，筆者が各フェーズで担当したタスクを紹介し，実施内容と考察を述べた．

筆者が担当した技術分野として，本プロジェクトの開発フェーズで導入したテスト自動化の取り組みについて報告する．特にテスト自動化の方針や用いたツール，及び効果について報告する．

最後に筆者が担当したマネジメント領域である，品質マネジメントの取り組みを報告した．PMBOK[1]を参考に，品質計画・品質保証・品質管理の方針と実施内容を述べる．品質計画では，本マネジメントの基本方針として特に重要視する品質特性を，品質保証ではチームとして品質を担保するための品質保証プロセスを，品質管理ではテスト及びレビューで抽出したバグの実績とコントロールについて報告する．

# 目次

本書の構成について ……0
第 1 章　プロジェクトの概要 ……1
1.1　プロジェクトの背景 ……1
1.2　プロジェクトの目的 ……1
1.3　プロジェクトの基本方針 ……1
1.4　顧客の基本情報 ……2
1.5　プロジェクト体制 ……2
1.5.1　学生チームの役割分担 ……2
1.5.2　プロジェクト推進における学生チームと顧客の関わり ……3
1.6　プロジェクトのスケジュール ……4
第 2 章　戦略立案フェーズ ……5
2.1　本フェーズの目的 ……5
2.2　本フェーズのスケジュール ……5
2.3　戦略フレームワークを用いた情報収集・分析 ……6
2.3.1　マインド・マップを用いた顧客の想いとビジョンの共有 ……6
2.3.2　外部環境の調査 ……7
2.3.3　内部環境の調査 ……9
2.3.4　SWOT 分析を用いた情報分析 ……10
2.3.5　クロス SWOT 分析による経営戦略案の抽出 ……11
2.3.6　戦略の決定 ……13
2.3.7　ロードマップによる中長期経営計画の策定 ……13
2.4　本フェーズでの筆者の担当部分と振り返り ……14
第 3 章　要件抽出フェーズ ……16
3.1　本フェーズの目的 ……16
3.2　本フェーズのスケジュール ……16
3.3　本プロジェクトが支援する顧客の経営課題について ……17
3.3.1　顧客の経営課題の確認 ……17
3.4　コーチ代行管理業務について ……18
3.4.1　経営課題改善のための施策 ……18
3.4.2　現状の業務プロセスの確認 ……19
3.4.3　コーチ代行管理業務の実績データ ……20
3.4.4　コーチ代行管理業務の問題点 ……20
3.4.5　システム導入後の業務プロセスの作成 ……21
3.5　コーチの勤務評価管理業務について ……22
3.6　開発機能の分析 ……24
3.6.1　ユーザストーリについて ……24
3.6.2　ユーザストーリの抽出 ……24
3.6.3　ユーザストーリポイントの重み付け ……25
3.6.4　ユーザストーリの優先度決定 ……25

3.6.5 作成したユーザストーリ一覧 ……26
第 4 章 開発フェーズ ……27
4.1 本フェーズの目的 ……27
4.2 本フェーズのスケジュール ……27
4.3 本フェーズの方針 ……28
4.3.1 考慮したリスク ……28
4.3.2 開発方針 ……28
4.3.3 開発手法の検討 ……29
4.4 イテレーションの流れ ……30
4.4.1 開発するユーザストーリの選定と確認 ……30
4.4.2 チームタスクの抽出 ……30
4.4.3 プロジェクトメンバへのタスクの割り振り ……31
4.4.4 タスクの遂行 ……31
4.4.5 開発したシステムのデモと顧客レビュー ……31
4.4.6 KPT によるイテレーションの振り返り ……32
4.5 開発環境について ……33
4.5.1 開発を行うために用いたツール ……33
4.5.2 開発用サーバのソフトウェア構成 ……33
4.5.3 開発したシステムの構成 ……33
4.6 開発の実績 ……34
4.6.1 ユーザストーリの開発実績 ……34
4.6.2 開発したシステムの規模 ……35
4.7 筆者の担当タスクについて ……36
4.7.1 イテレーション 0 ……36
4.7.2 イテレーション 1 ……36
4.7.3 イテレーション 2 ……37
4.7.4 イテレーション 3 ……37
4.7.5 イテレーション 4 ……38
4.8 筆者の担当技術領域ついてーテスト自動化 ……39
4.8.1 テスト自動化の目的 ……39
4.8.2 テストの方針について ……39
4.8.3 本プロジェクトで用いるテストツールについて ……40
第 5 章 評価フェーズ ……42
5.1 本フェーズの目的 ……42
5.2 本フェーズのスケジュール ……42
5.3 本フェーズの方針 ……43
5.3.1 アンケートについて ……43
5.3.2 KPI/KGI について ……43
5.4 実績 ……45
5.4.1 システムトラブルについて ……45
5.4.2 アンケート回収について ……46
5.4.3 アンケート結果 ……47

第 6 章　品質マネジメント ……57
6.1　品質とは ……57
6.2　品質マネジメントとは ……59
6.3　品質計画 ……59
6.3.1　基本方針 ……59
6.3.2　考慮した点 ……60
6.4　品質保証 ……61
6.4.1　品質マネジメントフローの設定 ……61
6.4.2　品質保証プロセスの設定 ……62
6.4.3　品質分析について ……65
6.5　品質管理 ……66
6.5.1　バグの定義 ……66
6.5.2　目標値の算出 ……66
6.5.3　見積もりと実績 ……67
6.6　各イテレーションの品質判定 ……68
第 7 章　プロジェクトの総括 ……72
7.1　考察 ……72
7.1.1　本プロジェクトの取り組みについて ……72
7.1.2　今後の展望について ……73
謝辞 ……74
参考文献 ……75

# 図目次

図 1-1　顧客とのミーティング風景 ……1
図 1-2　学生チームと顧客 ……3
図 1-3　スケジュール計画と実績 ……4
図 2-1　戦略立案フェーズのスケジュール ……5
図 2-2　顧客と作成したマインド・マップ ……6
図 2-3　外部環境調査結果 ……7
図 2-4　内部環境調査の結果 ……9
図 2-5　T1 の SWOT 分析 ……10
図 2-6 SWOT 分析とクロス SWOT 分析の関係 ……11
図 2-7 SWOT クロス分析によって導き出された戦略案 ……12
図 2-8　経営戦略ロードマップ ……13
図 2-9　電子カルテシステムのマインド・マップ ……14
図 2-10　マッピングした顧客の要望と該当する IT ソリュ―ション案のまとめ ……15
図 3-1　要件抽出フェーズのスケジュール ……16
図 3-2　現状業務フロー図（コーチ代行管理） ……19
図 3-3　システム導入後の業務フロー図（コーチ代行管理業務） ……21
図 3-4　システム導入後の業務フロー図（コーチ勤務評価管理業務） ……23
図 3-5　作成したユーザストーリ ……24
図 4-1　開発フェーズのスケジュール ……27
図 4-2　反復型開発のイメージ ……29
図 4-3　Redmine を用いたかんばん（チームタスク抽出時） ……30
図 4-4　Redmine を用いたかんばん（プロジェクトメンバへのタスク割り振り時） …31
図 4-5　KPT による振り返りシート ……32
図 4-6　開発したシステムの構成図 ……34
図 4-7　RSpec のコードの一部抜粋 ……41
図 4-8　RSpec 実行結果の例 ……41
図 5-1　評価フェーズのスケジュール ……42
図 5-2　運用マニュアルに対する評価のレーダーチャート ……48
図 5-3　戦略立案フェーズの取り組みに対する評価のレーダーチャート ……50
図 5-4　要件抽出フェーズの取り組みに対する評価のレーダーチャート ……52
図 5-5　開発フェーズの取り組みに対する評価のレーダーチャート ……54
図 5-6　プロジェクト全体の取り組みに対する評価のレーダーチャート ……56
図 6-1　品質の種類 ……57
図 6-2　品質マネジメントフロー ……61
図 6-3　品質保証プロセス ……62
図 6-4　テスト項目表 ……63
図 6-5　テスト項目レビュー表 ……63

# 表目次

表 1-1　T1 の基本情報（2012 年 12 月時点） ..... 2
表 1-2　RAM 表（役割分担表） ..... 2
表 2-1　外部環境調査の観点 ..... 7
表 2-2　内部環境調査の観点 ..... 9
表 2-3　SWOT クロス分析による戦略案 ..... 11
表 3-1　経営課題と経営改革ロードマップとの対応表 ..... 18
表 3-2　コーチ代行管理業務の実績データ ..... 20
表 3-3　評価制度の概要と目的 ..... 22
表 3-4　ユーザストーリの説明 ..... 24
表 3-5　抽出したユーザストーリの機能群 ..... 25
表 3-6　コア機能およびサブ機能のユーザストーリ数と総ユーザストーリポイント数 25
表 3-7　US 一覧表（イテレーション 4 完了時点） ..... 26
表 4-1　開発方針 ..... 28
表 4-2　KPT の用語説明 ..... 32
表 4-3　開発に用いたツール ..... 33
表 4-4 開発用サーバのソフトウェア構成 ..... 33
表 4-5 開発したユーザストーリ数 ..... 34
表 4-6　開発したユーザストーリポイント数 ..... 35
表 4-7　開発したアプリケーションの規模 ..... 35
表 4-8　担当したイテレーション 0 のタスク ..... 36
表 4-9　担当したイテレーション 1 のタスク ..... 36
表 4-10　担当したイテレーション 2 のタスク ..... 37
表 4-11　担当したイテレーション 3 のタスク ..... 37
表 4-12　担当したイテレーション 4 のタスク ..... 38
表 4-13　テスト・ファーストを用いる基準 ..... 40
表 5-1　評価フェーズで扱う測定データ ..... 43
表 5-2　実施したアンケートの概要 ..... 43
表 5-3　測定した KPI の項目 ..... 44
表 5-4　評価フェーズで発生したシステムトラブルの内訳 ..... 45
表 5-5　運用マニュアルについてのアンケート結果 ..... 47
表 5-6　戦略立案フェーズの取り組みについてのアンケート結果 ..... 49
表 5-7　要件抽出フェーズの取り組みについてのアンケート結果 ..... 51
表 5-8　開発フェーズの取り組みについてのアンケート結果 ..... 53
表 5-9 プロジェクト全体の取り組みについてのアンケート結果 ..... 55
表 6-1　品質の種類の概要 ..... 57
表 6-2　ソフトウェア品質特性 ..... 58
表 6-3　品質マネジメントプロセスの概要 ..... 59
表 6-4　工程品質判定表 ..... 65
表 6-5　バグの定義 ..... 66

表 6-6　イテレーション 0 の実績値 .....67
表 6-7　イテレーション毎の見積もり値と実績との比較 .....67
表 6-8　イテレーション 1 の品質分析該当箇所 .....68
表 6-9　イテレーション 2 の品質分析該当箇所 .....69
表 6-10　イテレーション 3 の品質分析該当箇所.....70
表 6-11　再検討したバグ数見積もり .....70
表 6-12　イテレーション 4 の品質分析該当箇所.....71

# 本書の構成について

第 1 章では，本プロジェクトの目的や方針，本プロジェクトで行う各フェーズの目的を述べ，プロジェクトの全体象を説明する．

第 2 章では，戦略立案フェーズの概要と，実際に行った分析結果について述べ，本プロジェクトが推進する戦略目標を説明する．

第 3 章では，要件抽出フェーズの概要と，顧客から抽出した要件を述べ，実際に開発するシステムの概要について説明する．

第 4 章では，開発フェーズの概要と実績，及び筆者が担当したタスクとテスト自動化についての取り組みについて説明する．

第 5 章では，評価フェーズの概要と，本フェーズによって得られたデータの分析結果を述べ，本プロジェクトの目的と成果の妥当性について説明する．

第 6 章では，筆者が担当した品質マネジメントの概要と実績，及びそこから得られた知見について述べる．

第 7 章では，本プロジェクトを総括する．また，第 2 章～第 6 章では，各フェーズの目的と，筆者の担当部分について必ず記述している．

巻末には，本プロジェクトの成果ドキュメントを付録する．

# 第1章　プロジェクトの概要

## 1.1　プロジェクトの背景

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 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（以下，高度 IT コースと呼ぶ）における，2011 年度の授業科目「PBL 型システム開発」で，つくば市内にある T1 インドアテニススクール（以下，T1 と呼ぶ）を顧客として，「テニススクールレッスン予約システム」を開発するプロジェクトが立ち上げられた．しかし，企画提案段階まで行った後，学生チームの解散により終了した．本年度は顧客である T1 より，もう一度最後まで改革の推進をして欲しいとの要望を受け，特定課題研究開発プロジェクトとしてプロジェクトメンバの再編成を行い，経営改革を推進することになった．

## 1.2　プロジェクトの目的

本プロジェクトの目的は，顧客の抱える経営課題を共有し，戦略立案工程から経営改革を推進すること，及び顧客の IT ガバナンスに適合した IT ソリューションを提供し，経営基盤の確立を支援することである．

学生チームは本プロジェクトを遂行する事で，経営戦略の立案という超上流工程からシステムの開発・納品・評価まで，一連のソフトウェア開発プロセスと技術を実践的に習得することができる．

## 1.3　プロジェクトの基本方針

本プロジェクトの顧客は，IT ベンダと共同による戦略立案や IT 調達の経験が十分ではなかった．そのため，経験不足によるリスクの発生を抑えるために，顧客とのコミュニケーションを重視することにした．これによって，顧客の真のニーズを抽出し，納品後も使って貰えるシステム開発を行う事を目指した．

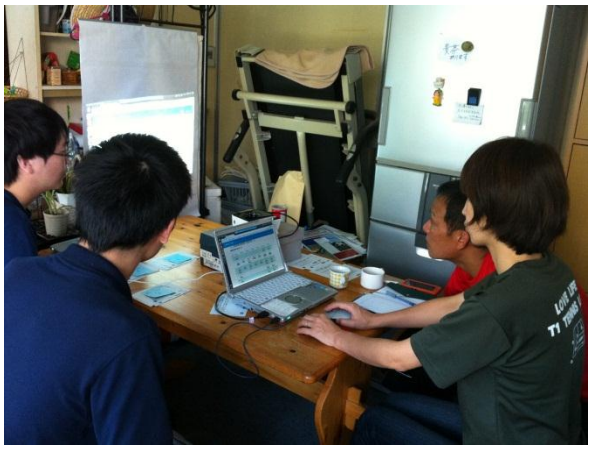

**図 1-1　顧客とのミーティング風景**

## 1.4　顧客の基本情報

T1 の基本情報を表 1-1 に示す．T1 は 2003 年 4 月につくば市梅園で開業したテニススクールである．つくば市初のインドアテニスコートを持ったスクールとして，地域に根差した経営を行っている．T-1 は大学生のアルバイトコーチが多く，若くて明るい雰囲気を醸成している一方で，学業や部活動等不定期な休みが多い，及び卒業によって退職するため，コーチの長期雇用の点で課題を抱えている．

**表 1-1　T1 の基本情報（2012 年 12 月時点）**

| | | |
|---|---|---|
| 社名 | T-1 インドアテニススクール | |
| 開業 | 2003 年 4 月 | |
| 経営陣 | 代表取締役　兼コーチ | 末満裕之　様 |
| | マネージャ | 末満弘江　様 |
| 従業員数 | コーチ | 18 名 |
| | スタッフ | 4 名 |
| 生徒数 | 約 300 人 | |
| 事業内容 | ○テニスレッスン | ビギナー・初級・初中級・中級・中上級・上級・キッズ・ジュニア１・ジュニア２・中高生 |
| | ○コート貸出し | インドアコート<br>アウトドアコート |
| 設備 | インドアコート | 1.5 面 |
| | アウトドアコート | 2 面 |
| | その他 | 託児室，駐車場，クラブハウス |

## 1.5　プロジェクト体制

### 1.5.1　学生チームの役割分担

本プロジェクトは，高度 IT コースに在籍する学生 5 人が推進することになった．プロジェクトメンバそれぞれがシステム開発，技術調査，マネジメントを分担し進める．筆者はテスト自動化の技術調査と品質マネジメントの計画・実施を担当する．学生チームの RAM 表（Responsibility Assignment Matrix）[1]を表 1-2 に示す．

**表 1-2　RAM 表（役割分担表）**

| メンバ | 技術調査 | マネジメント分野 |
|---|---|---|
| 白田良太 | コーチ評価体制と管理機能 | コミュニケーション |
| 有田正信 | テスト自動化 | 品質 |
| 永井達也 | セキュリティ対策 | 進捗 |
| 井原淳平 | 開発環境・運用環境の構築 | スコープ |
| 杜セイ雨 | ユーザビリティ | リスク |

1.5.2　　プロジェクト推進における学生チームと顧客の関わり

本プロジェクトは T1 の経営者である末満様を顧客に迎え，顧客と共に T1 の経営改革の推進と，経営基盤の確立に向けた IT ソリューションの開発を行った．現状業務に混乱を招かないように顧客との緻密な連携を取りながら，従業員に対しても適時ヒアリングを行った．

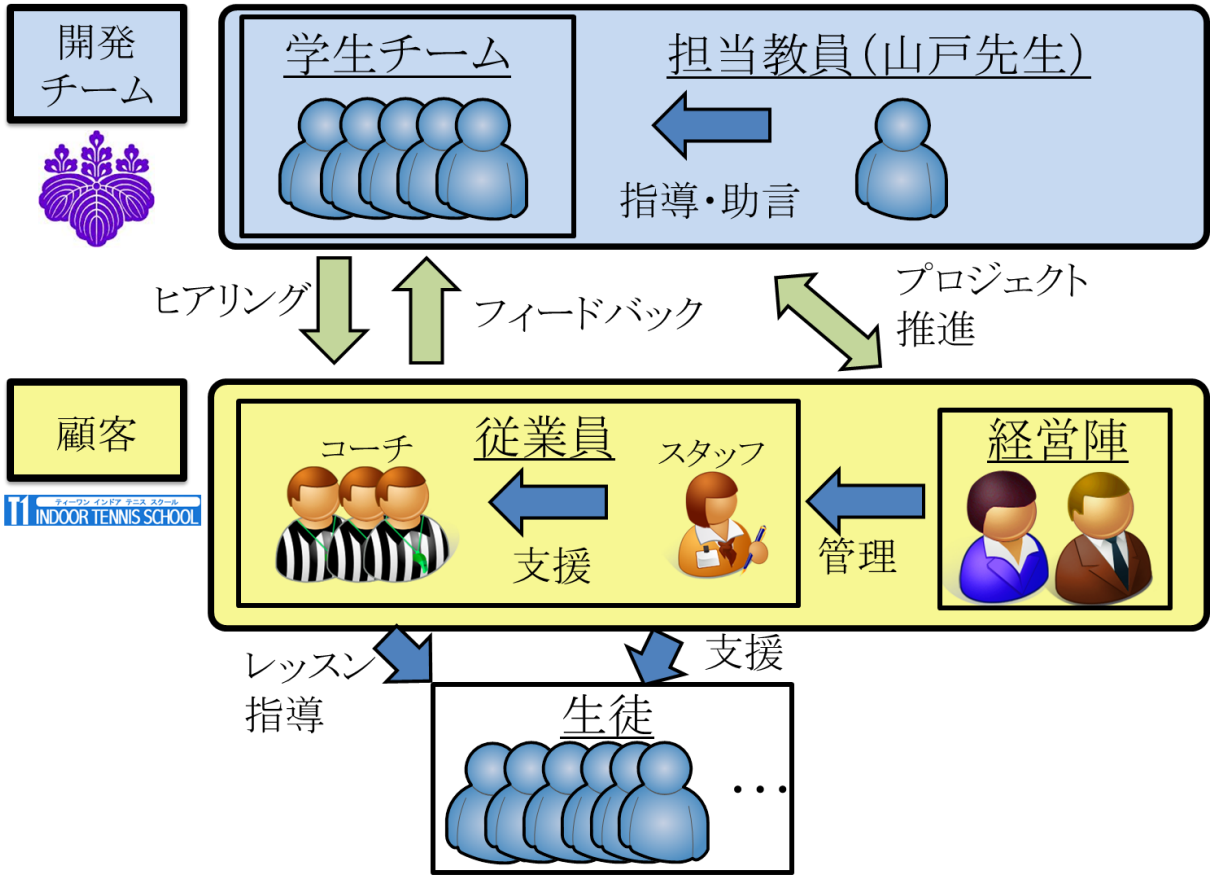


**図 1-2　学生チームと顧客**

## 1.6　プロジェクトのスケジュール

本プロジェクトは 2012 年 5 月～2013 年 1 月の約 9 か月間で行った．スケジュールの計画と実績を図 1-3 に示す．

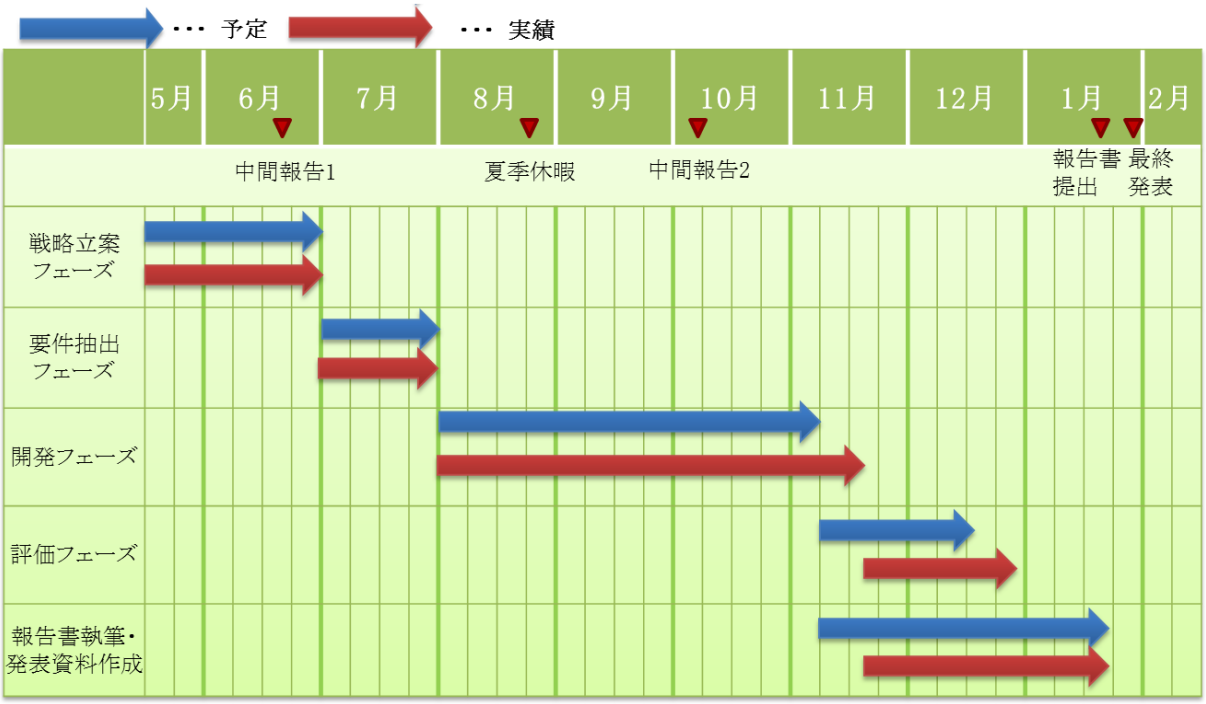


**図 1-3　スケジュール計画と実績**

本プロジェクトは 4 つのフェーズ（報告書執筆・発表資料作成は除く）に分けて進めていった．各フェーズの概要について以下に述べる．

戦略立案フェーズの目的は，顧客の真のニーズを引き出し，課題解決のための戦略とアクションプランの立案を推進することである．実施内容としては，顧客を取り巻く環境を調査・分析し，現状課題の抽出を行った．そして，各課題に対する戦略案の抽出と決定を行い，経営改革状況を定量的に測定する指標の策定を行った．

要件抽出フェーズでは，戦略立案フェーズで決定した戦略を基に，システム化の対象となる業務プロセスの構築と，実際に開発するシステムの要件を抽出した．

開発フェーズでは，要件抽出フェーズで抽出したシステムの要件を基に開発を行う．開発期間を 2 週間の短い期間で区切り，区切られた期間毎に設計・実装・テスト・リリースを行った．

評価フェーズでは，顧客にシステムを運用するための準備として，マニュアルの整備と講習会を行った．また，システムを納入し，実際に顧客にシステムを運用して頂き，従業員・経営陣へアンケートを実施した．アンケートの結果や，システムのログから得られたデータを基に，経営改革の目標達成度合い，システムの品質，及びプロジェクト活動へ対する評価を行った．

# 第2章　　戦略立案フェーズ

## 2.1　　本フェーズの目的

本フェーズの目的は，顧客の経営課題解決のための戦略とアクションプランの立案を推進することである．そのために顧客の持つ想いとビジョン，現状の課題を抽出し，課題解決に向かってどのような戦略をとるべきかを決定する．

## 2.2　　本フェーズのスケジュール

本フェーズのスケジュールを図 2-1 に示す．

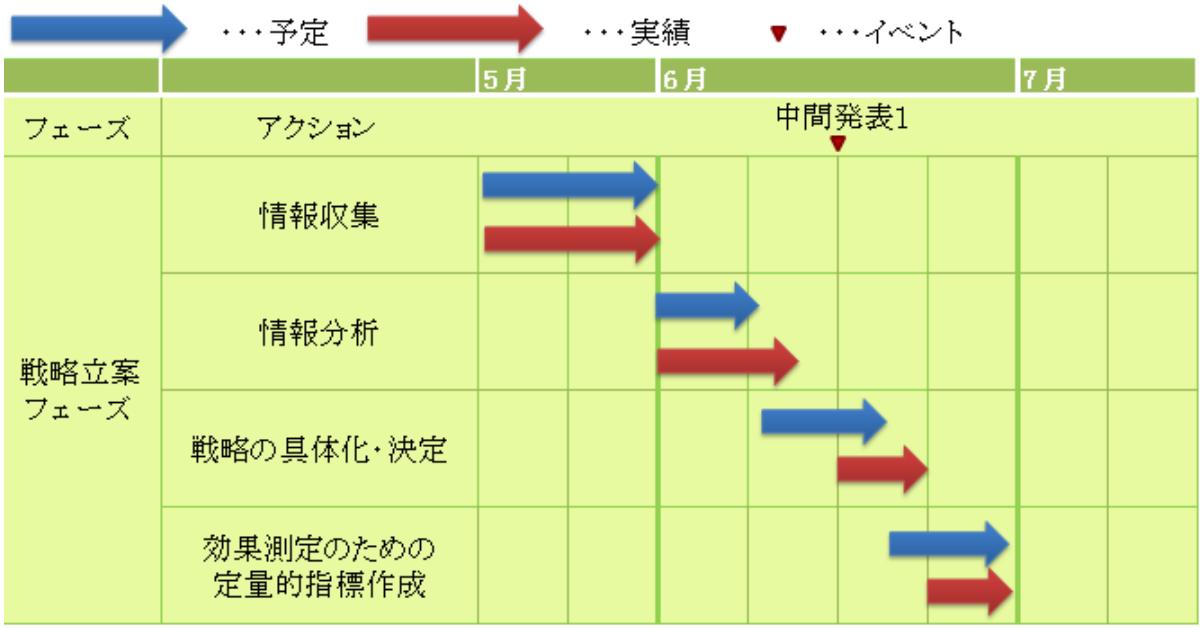


**図 2-1　戦略立案フェーズのスケジュール**

本フェーズは，5 月 17 日から 6 月 29 日までの約 6 週間の期間で行った．顧客とのミーティング回数は全部で 7 回，週に一度のペースで行った．

それぞれのアクションの概要について以下に述べる．

情報収集では，経営者にヒアリングを行い，将来のビジョンについて洗い出しを行った．

情報分析では，戦略フレームワーク[2]を用いて，T1 の内外の情報を分析し，顧客が取るべき戦略案の抽出を行った．

戦略の具体化・決定では，情報分析で抽出した戦略案の中から，最も効果的な戦略を決定し中長期的な戦略をロードマップ[2]にまとめた．

効果測定のための定量的指標作成では，本年度の施策の効果を定量的に測定するための指標を作成した．

## 2.3　戦略フレームワークを用いた情報収集・分析

当初，本プロジェクトの顧客及び学生チームは戦略立案に関する知識や経験が乏しかったため，情報を網羅的に収拾できないリスクや，収集した情報を効果的にまとめられないリスクが考えられた．そのリスクを最小限に抑えるために，ＩＴコーディネータプロセスガイドライン[2]に沿って戦略立案を行う事にした．

次章より実際に行ったワークショップの内容と成果物を示す

### 2.3.1　マインド・マップを用いた顧客の想いとビジョンの共有

ブレイン・ストーミングのように，参加者が様々な意見を自由に出し合うようなミーティングにおいては，マインド・マップ[3]を用いて会話内容の図示化を行った．これにより参加者は「何をどこまで話して何を話してないのか」という内容の整理や，聞き間違いやニュアンスの差異による言葉の誤解が無くなり，効果的に話し合いを行う事ができた．

図 2-2 に，「顧客の想いとビジョン」を話し合った際に作成したマインド・マップを示す．

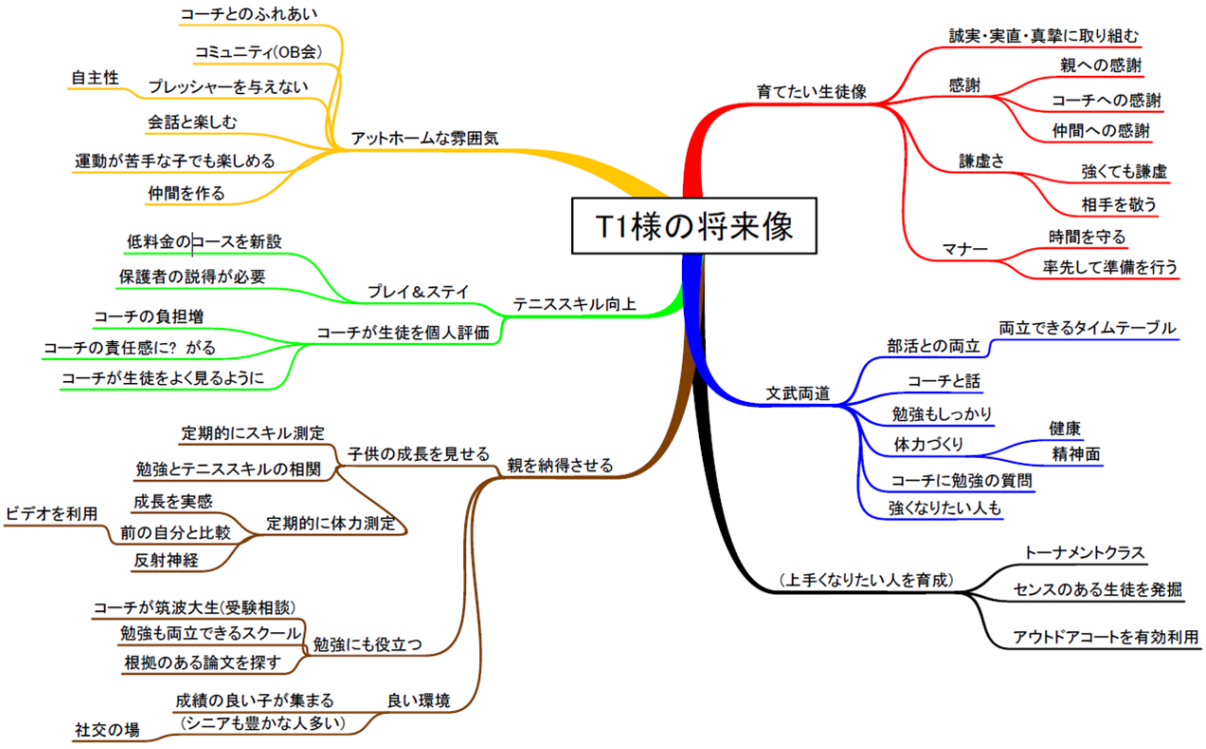


図 2-2　顧客と作成したマインド・マップ

### 2.3.2 外部環境の調査

まず，IT プロセスガイドライン[2]を参考に外部環境調査行った．T 1 を取り巻く外部環境について，表 2-1 に示す 6 つ観点から抽出した．

**表 2-1　外部環境調査の観点**

| 観点 | 概要 |
|---|---|
| 競争者 | 業界内の競争の激しさや競争相手の動向はどうか，など．今回は競争相手となる企業や脅威となる存在をまとめた． |
| 新規参入者 | 2010 年から 2012 年までの期間に台頭してきた競合企業をまとめた．「競争者」と範囲が被っている部分はこちらにまとめた． |
| 代替品・代替サービス | 代替品・代替サービスにどのようなものがあるかをまとめた． |
| 供給業者 | どのように資源を集めているかをまとめた．ここでは，T1 の人的資源の供給元をまとめた． |
| 社会動向 | T1 に影響する社会動向として，テニスやスポーツに関する動向をまとめた．経営陣へのヒアリング結果を参考にした． |
| 顧客ニーズ | T1 の生徒がスクールに求めているものをまとめた．T1 が 2005 年に生徒に向けて行ったアンケート結果を参考にした． |

実際に調査を行った結果を図 2-3 にまとめた．

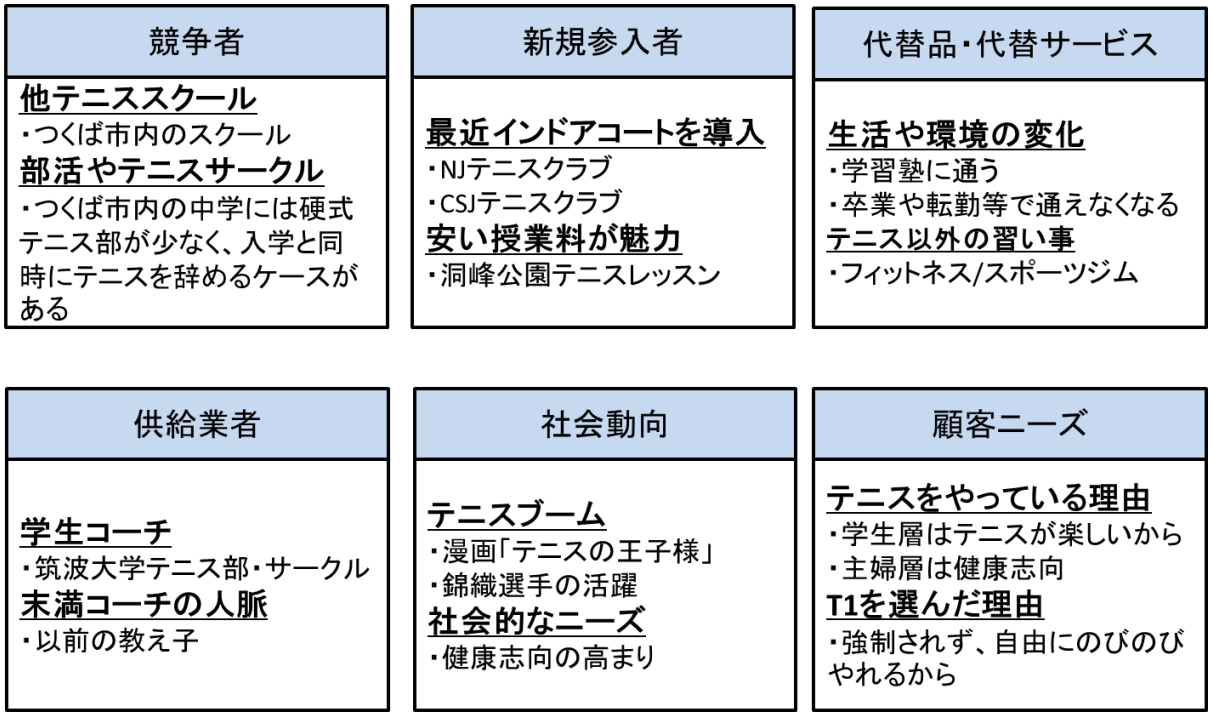


**図 2-3　外部環境調査結果**

それぞれの項目について説明する．

- 競争者

T1 と同じ経営競争力を持った競争相手はつくば市内にある他テニススクールである．また，生徒の視点で考えると，学校の部活や運動系サークル等も選択肢にあり，それらと T1 は競合関係にある．また，つくば市市内の中学校では硬式テニス部が無い学校もあり，進学を機に違うスポーツを始め，同時に T1 も辞めてしまう生徒も存在している．

- 新規参入者

T1 は 2003 年のオープン以来，つくば市で唯一のインドアテニスコートを持ったテニススクールとして競争力を持っていた．しかし，近年他スクールもインドアコートを導入したため，新たな差別化を測る必要がある．

- 代替品・代替サービス

小学生～大学生の生徒は，時期によって受験勉強や就職活動が発生する．その後進学や就職によって住む場所や生活時間帯が変わってしまうと，T1 に通う事が出来なくなる．

また，社会人・主婦の生徒は近くにスポーツジムやフィットネスクラブがある場合に，そちらを選択する事があり得る．

- 供給業者

T1 のコーチは大部分が学生コーチであり，同じ部活やサークルの人脈の紹介によって T１で働いている．また，T1 開業以前の末満コーチの人脈で働いているコーチもいる．

- 社会動向

T1 が開業した 2003 年は，人気漫画「テニスの王子様」による効果でテニスに対する人気が高かった．最近ではプロテニスプレイヤー・錦織圭選手の活躍もあり，テニスの人気は開業当初と変わらず高い状況である．

- 顧客ニーズ

T1 が 2005 年に生徒に対して行ったアンケートによると，生徒が他スクールではなく T1 を選んだ理由は，伸び伸びとした雰囲気や良好な人間関係にある事が分かった．

### 2.3.3 内部環境の調査

次に，IT プロセスガイドライン[2]を参考に内部環境調査を行った．T1 の内部環境について，表 2-2 に示す 4 つの観点から抽出した．

**表 2-2　内部環境調査の観点**

| 観点 | 概要 |
|---|---|
| 施設 | T1 が保有する施設についてまとめた． |
| 情報マネジメント力 | T1 内で行われている情報管理についてまとめた． |
| 保有人材 | T1 の人的資源についてまとめた． |
| 顧客・市場適応力 | T1 が生徒を確保するための競争力として，宣伝及び立地についてまとめた． |

実際に調査を行った結果を図 2-4 にまとめた．

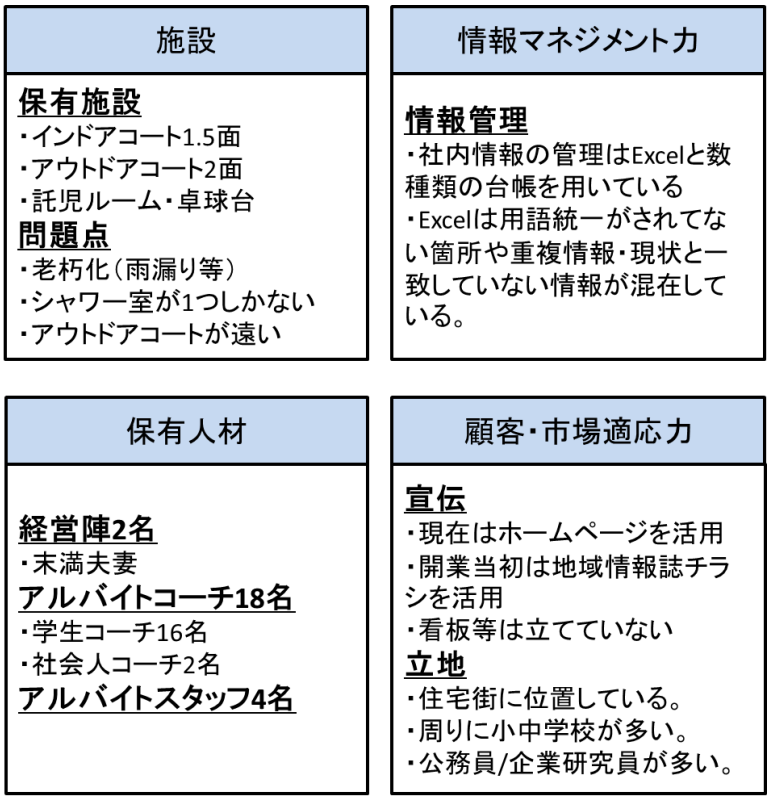


**図 2-4　内部環境調査の結果**

### 2.3.4 SWOT 分析を用いた情報分析

顧客の内部環境調査，外部環境調査の結果について SWOT 分析[2]を行い，それぞれ強み，弱み，機会，脅威に分類した．図 2-5 に SWOT 分析で作成したマトリクス図を示す．

| | | | |
|---|---|---|---|
| 内部環境 | 強み(Strengths)<br>施設<br>・インドアコートがあること<br>・半面コートがある<br>・託児ルーム・卓球台<br>人材<br>・学生コーチ(フレッシュ＆人件費が安い)<br>・末満さん(コアなファンが多い)<br>立地<br>・住宅街に立地し子供が多い | 機会(Opportunities)<br>・錦織圭効果<br>・健康志向<br>・公務員/企業研究員が多い<br>・社会人サークル<br>・茨城県公立中で硬式テニス部を作る動き<br>(遠征時の親の負担が減る)<br>・TXが出来たことによる商圏の広がり | 外部環境 |
| | 弱み(Weaknesses)<br>施設<br>・コートが狭い・老朽化してきた<br>・アウトドアコートが遠い<br>・シャワー室が１つしかない<br>宣伝<br>・ホームページ(自信がない)<br>・看板(初めての方にもわかりづらい)<br>人材<br>・雇用期間が短い＆スキルにバラつきがある<br>立地<br>・地価が高い<br>システム<br>・台帳での管理がメイン<br>・エクセルと紙の使い分けが曖昧 | 脅威(Threats)<br>同業他社<br>・インドアコートがあるスクール<br>(NJ、CSJ)<br>・安いスクール(洞峰公園公園)<br>・他の選択（フィットネス・仕事・社会人サークル)<br>・中学生の部活<br>社会的要因<br>・子供人口が減ってきている | |

**図 2-5　T1 の SWOT 分析**

T1 の SWOT 分析の結果を以下にまとめる．

強みとしては豊富な設備が整っているため，生徒に充実したレッスンを行うことが出来る点，弱みとしては，社内での人材確保や情報管理が効率的に行えていない点があげられる．

また，機会としては，テニスブームや茨城県の公立中学で硬式テニス部が作られる動きが見られるため，新規顧客獲得のチャンスが生まれている点，脅威としては競合他社や代替サービスの台頭している点があげられる．

2.3.5　　クロス SWOT 分析による経営戦略案の抽出

次に SWOT 分析の結果からクロス SWOT 分析[2]を行った．

クロス SWOT 分析は，SWOT 分析で抽出した内部環境（強み・弱み）と外部環境（機会・脅威）を図 2-6 のように組み合わせる事で表 2-3 に示す 4 つの戦略を導き出す手法である．

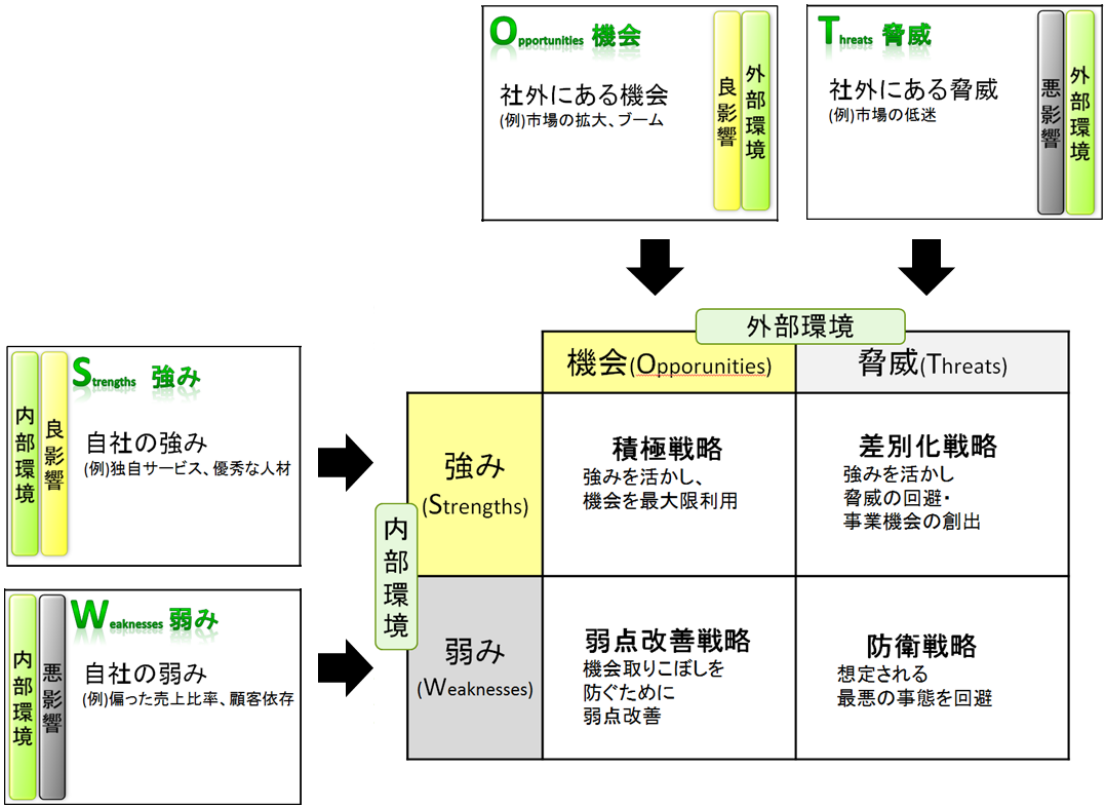


**図 2-6 SWOT 分析とクロス SWOT 分析の関係**

**表 2-3　SWOT クロス分析による戦略案**

| 戦略名（組み合わせ） | 戦略の説明 |
|---|---|
| 積極戦略（強み×機会） | 強みを活かし，機会を最大限利用 |
| 弱点改善戦略（弱み×機会） | 機会取りこぼしを防ぐために弱点を改善 |
| 差別化戦略（強み×脅威） | 強みを活かし，脅威の回避・事業機会の創出 |
| 防衛戦略（弱み×脅威） | 想定される最悪の事態を回避 |

T1 で行った SWOT 分析の結果を図 2-7 に示す.

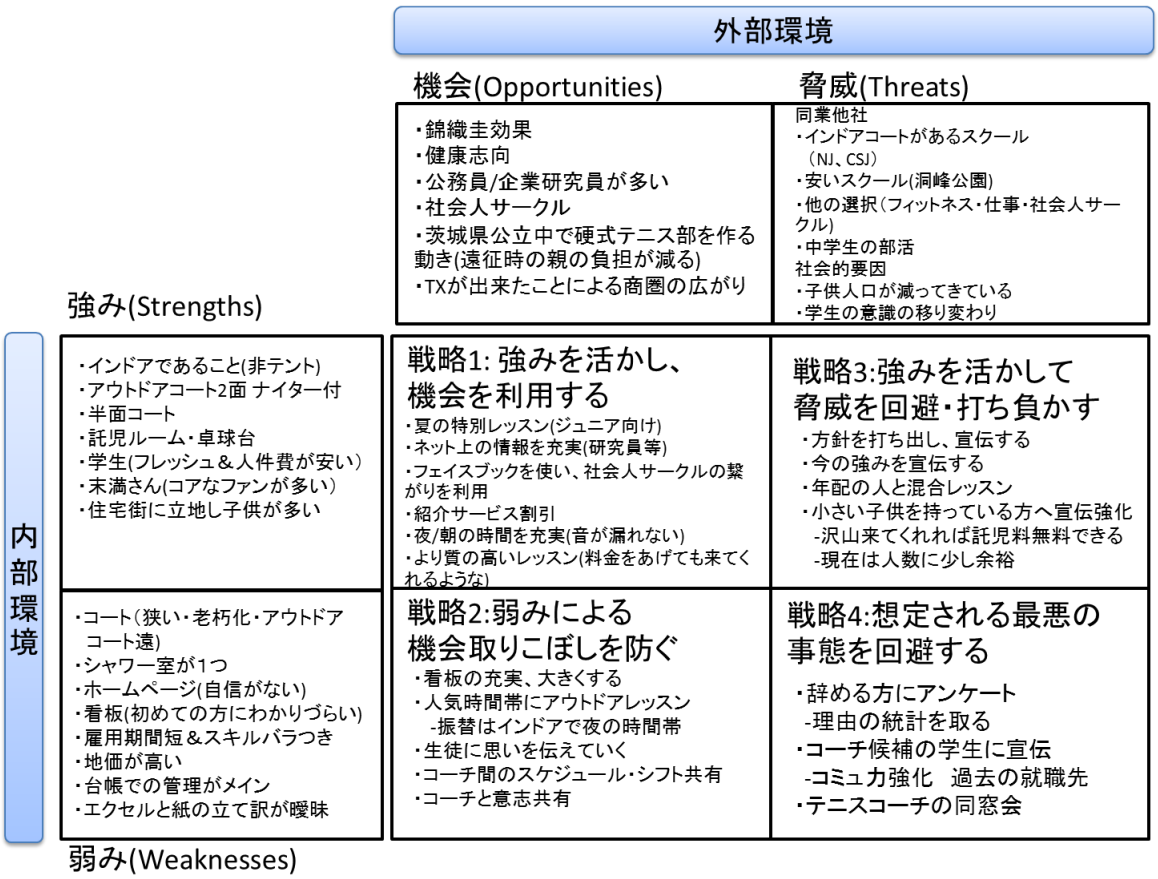


図 **2-7 SWOT** クロス分析によって導き出された戦略案

### 2.3.6 戦略の決定

クロス SWOT 分析の結果，積極戦略案，特に生徒に対するサービスの向上が比較的多く抽出された．これは【2.3.1 マインド・マップを用いた顧客の想いとビジョンの共有】でもあげられていたため，想像がし易かったからであると考えられる．

しかし，顧客との話し合いの結果，以下 2 点の問題点があげられた．

- 現状の業務で手一杯なのに，新サービスに充てる人材と時間は足りないのではないか
- 新サービスを始めて生徒数が増えた場合，コーチ数を増やす必要があるが，そうすると更に業務が煩雑になってしまうのではないか

以上の理由により，本年度は弱点改善戦略である「内部基盤の強化」に取り組み，生徒の満足度向上のための取り組みは来年度以降に行うことを顧客と合意した．

### 2.3.7 ロードマップによる中長期経営計画の策定

これまでの情報分析の結果から，T1 の中長期経営計画をまとめた経営改革ロードマップ[2]のように立案した．

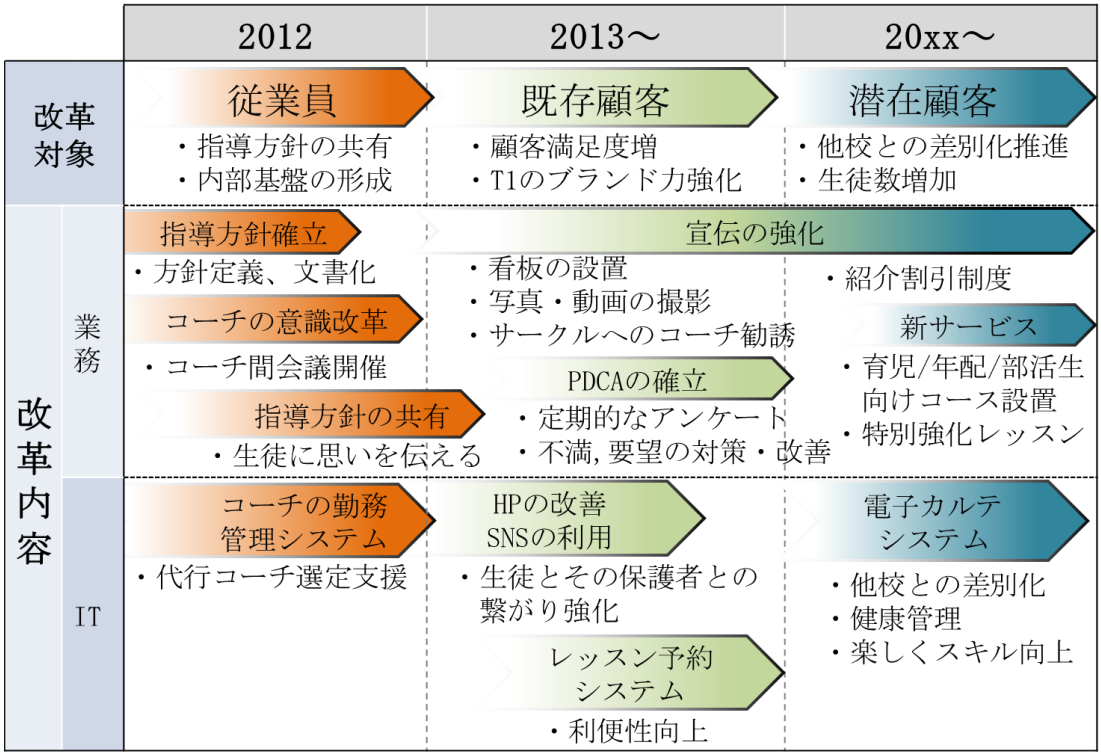


**図 2-8 経営戦略ロードマップ**

経営改革ロードマップは改革の対象を年次で分け，段階的に改革を行っていくことを表している．本年度は指導方針の共有と内部基盤の形成を目指し，経営者とコーチとスタッフが一体となって経営改善を図っていける組織体制を作っていく．来年度以降は T1 経営陣が他

スクールとの差別化を図るため顧客サービス強化を行い，更にインターネットを活用した宣伝を行い新規顧客の獲得を目指していく方針とし，IT ソリューション案にあげたシステムは来年度以降の PBL に引き継ぐ事を考えている．

## 2.4　本フェーズでの筆者の担当部分と振り返り

本フェーズでは，筆者は顧客とのコミュニケーションマネジメントを担当した．主なタスクとしては，顧客ミーティングの準備や，学生チームと顧客との折衝を行った．ここでは，顧客とのコミュニケーションにおいて実際に発生した問題を分析し，反省と学びを述べる．

まず，本フェーズにおいて顧客ミーティングは 7 回実施したが，序盤で大幅に遅延が発生した．遅延が発生した原因は次の通りである．

- 1～3 回目の顧客ミーティング（5 月 25 日，5 月 30 日，6 月 6 日）を経て，次の要望を引き出した．
  - 生徒の満足度をあげたい
  - ジュニアに特化した新サービスを始めたい
  - ジュニアの親にも喜んでもらえるサービスを取り入れる事で，生徒数を増やしたい
- チーム内でこれまでの要望をまとめ，上記の要望を満たすソリューション案として「電子カルテシステム（図 2-9）」を 4 回目の顧客ミーティング（6 月 13 日）で提案した．
- しかし，合意の直前の段階で，システムの運用に関して現状の T 1 の IT ガバナンスが低い，また運用者となる人材がいないという点から本年度のソリューション案としては不採用になった．

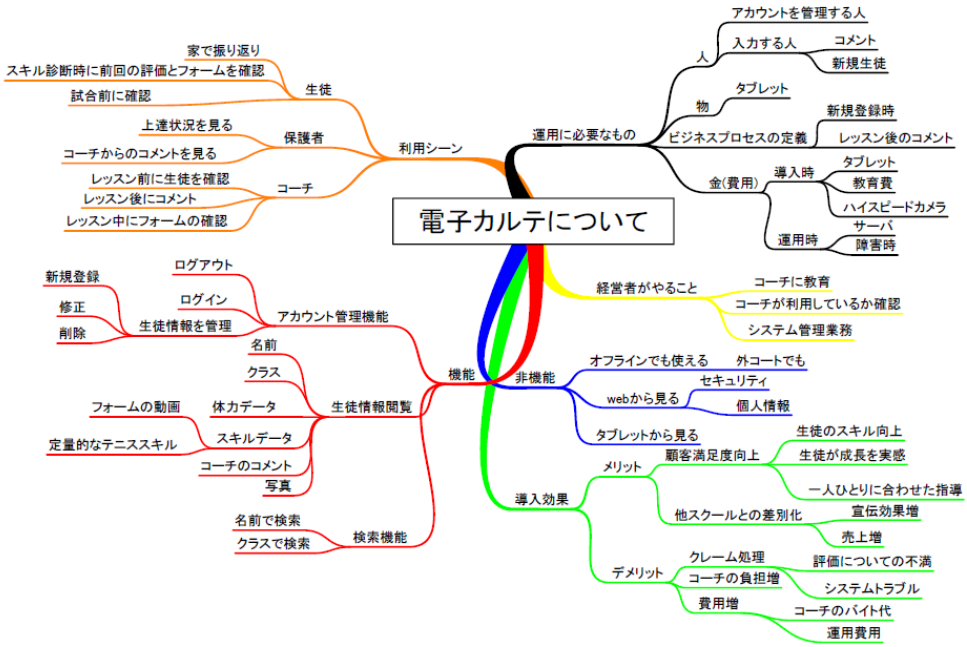


図 2-9　電子カルテシステムのマインド・マップ

上記のように，ヒアリングを重ねてきたにもかかわらず，合意の直前で問題が発生してしまった．後に【2.3 戦略フレームワークを用いた情報収集・分析】で示した一連の情報分析によって電子カルテシステムの導入は 3 年後以降が望ましいと決定したが，情報分析を行う前のチームの反省として以下の点が挙げられる．

- 全ての要件は達成できるものとして，実現性を考慮していなかった．
- 顧客の現状を分析しないまま将来のビジョンだけを見てしまっていた．そのため早い段階から運用時のプロセスを想定することが出来なかった．
- マインド・マップ等のツールをただ用いることに意義を感じており，「そのツールを用いる事で何を解決したいのか」という視点が不足していた．

上記の反省点を裏付けるものとして，第 4 回顧客ミーティングの直前に，チームで作成した顧客の要望とチームが提案したソリューションをまとめたものを図 2-10 に示す．

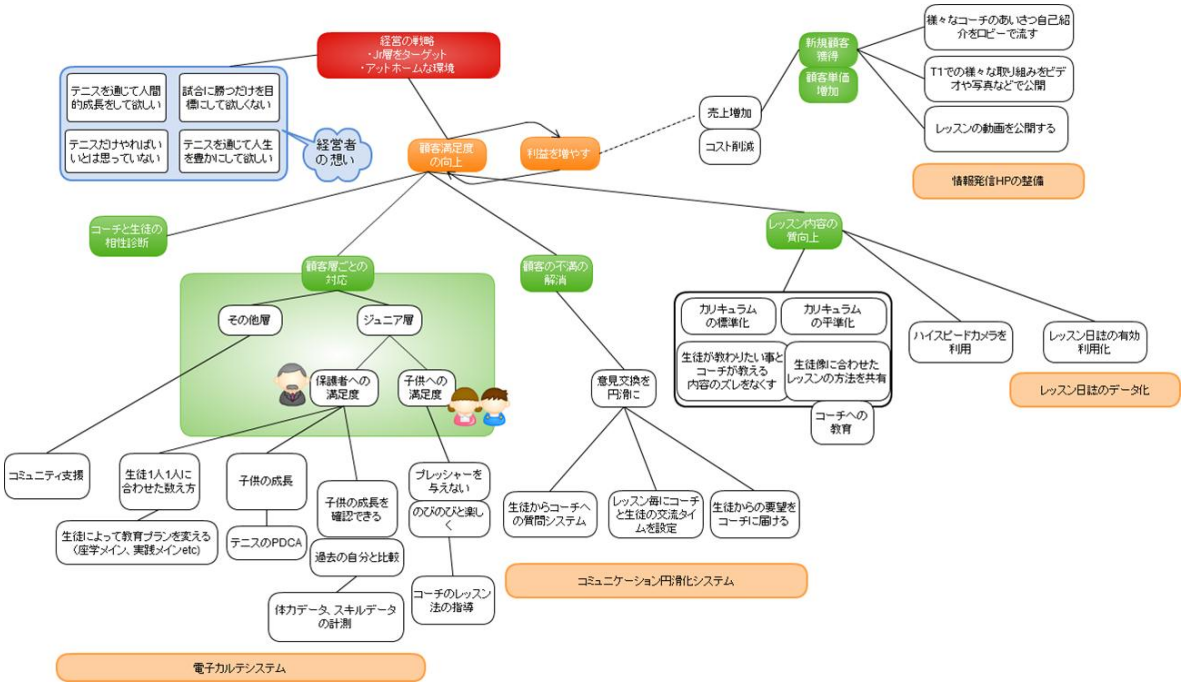


**図 2-10　マッピングした顧客の要望と該当する IT ソリュ―ション案のまとめ**

この図には「この要望を達成するためには，こんな取り組みを行えば良い」という事柄しか載っていないため，当時の学生チームは，実際に障壁となる事柄や現状の問題点が考慮できていないことが分かる．

これらのことから以下の学びが得られた．

- 学生チームは「顧客が欲しいという要件が，必ずしも顧客にとって最適な要件とは限らない．」という事を念頭に置いて，ヒアリングを行うべきであった．
- 戦略立案はビジョンを描くだけでなく，現状の分析を行い，現状と描いたビジョンの差を，誰がいつどうやって達成していくかというプロセスまで定義しなくてはならない．

# 第3章　　要件抽出フェーズ

## 3.1　　本フェーズの目的

本フェーズの目的は，戦略フェーズで決定した戦略案に基づき，経営課題を解決するためのアクションプランを決定する事である．具体的には，プロジェクトで開発する IT システムの要件の抽出及び，本年度 T1 が推進する業務改善案の提案を行った．

## 3.2　　本フェーズのスケジュール

本フェーズでは，図 3-1 に示す 4 つのアクション行った．

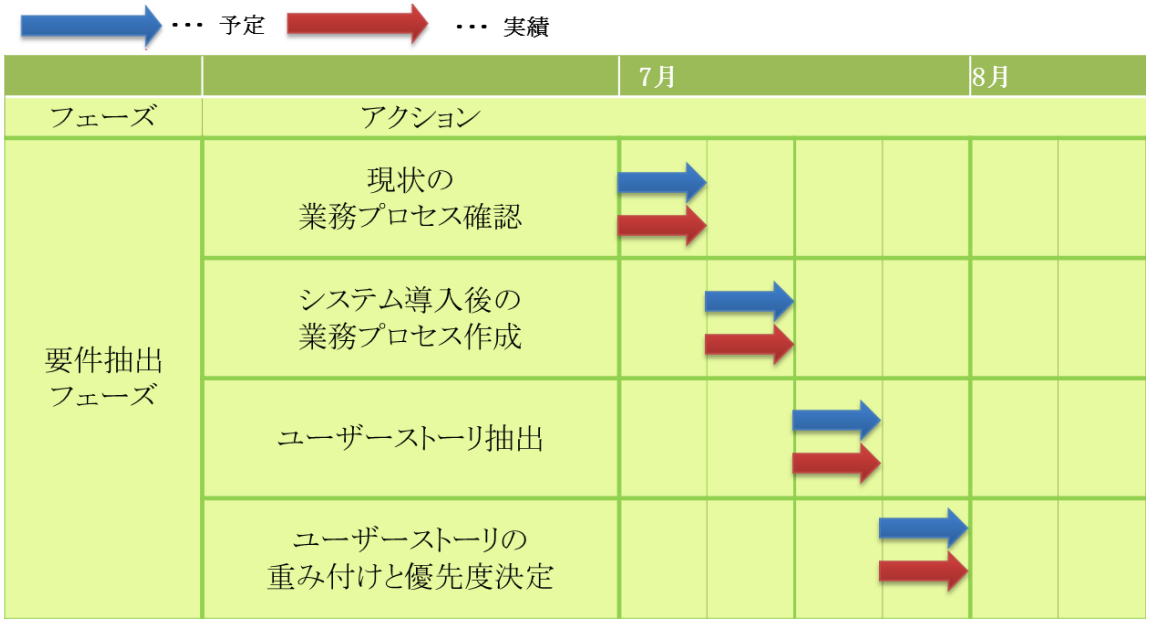


**図 3-1　要件抽出フェーズのスケジュール**

本フェーズは，2012 年 6 月 30 日から 7 月 30 日までの約 1 か月間で顧客とのミーティング回数は全部で 5 回，週に一度のペースで行った．それぞれのアクションの概要について以下に述べる．

現状の業務プロセス確認では，ヒアリングを通して現状の業務フロー図[3]を作成し，プロジェクトメンバと顧客で現状業務の問題点を共有し，改善点の模索を行った．

システム導入後の業務プロセス作成では，経営陣・コーチ・スタッフとの話し合いを通じて，システムを導入した際の新しい業務フロー図を作成した．

ユーザストーリ抽出では，顧客がシステムに求める要件をユーザストーリという単位で抽出し，開発するシステムの機能を洗い出した．

ユーザストーリの重み付けと優先度決定では，抽出したユーザストーリの規模と顧客の要望を照らし合わせ，開発を行う順序を決定した．

## 3.3　本プロジェクトが支援する顧客の経営課題について

### 3.3.1　顧客の経営課題の確認

本年度は，【2.3.5 クロス SWOT 分析による経営戦略案の抽出】で示した通り，弱点改善戦略として顧客の内部基盤の強化を行うことになった．現在顧客が抱えている内部基盤の問題について以下に述べる．

- 経営者・コーチ間で指導方針・理念が共有されていない．

経営者のスクールやテニスに対する理念・方針がコーチに共有されておらず，レッスンはコーチの自己基準によって指導を行われている．そのためコーチによってレッスンの品質にバラツキが発生している．

- 効果的な勤務評価制度が行われていない．

現在 T1 では，明確なコーチの勤務評価制度が存在していない．代行レッスンを請け負うコーチは，通常のレッスンを行う場合に比べ負担が大きいにも関わらず，現状では特に評価されない．また，レッスンのメニューを組み立てるメインコーチと，そのサポートを行うサブコーチとで，1 レッスンにおける評価が変わらないなど，業務の負荷に合わせた適切な評価が行われていない事が，コーチの不公平感を生む原因となっている．

- コーチ勤務管理の業務プロセスが属人的かつ場当たり的に行われている．

コーチの欠勤が発生した際に

## 3.4 コーチ代行管理業務について

現状のコーチ代行管理業務では，コーチがレッスンを欠勤する度に経営者が中継役となっており，その時々の判断で代行コーチを探している．そのため経営者の負担が大きい，且つ属人性が高い業務となっており，今後 T1 の規模が拡大していく際に対応できない恐れがある．

### 3.4.1 経営課題改善のための施策

前節で述べたそれぞれの経営課題に対し，【2.3.7 ロードマップによる中長期経営計画の策定】で作成したロードマップで示した，本年度取り組む 4 つの施策との関連を表 3-1 にまとめる．

**表 3-1 経営課題と経営改革ロードマップとの対応表**

| 施策 | 目的 | 課題 |
|---|---|---|
| 指導方針確立 | コーチ間の指導方針のバラつきを抑え，生徒に均一な品質のレッスンを提供する | ・経営理念，指導方針の明文化 |
| コーチの意識改革 | コーチに責任感と達成感を持たせ，自発的な成長を促す | ・公平感のある効果的な評価制度の確立 |
| 指導方針の共有 | 経営者とコーチが一体となり，経営理念に従った企業風土を醸成する | ・共有の機会の創出 |
| コーチ勤務管理システムの導入 | 属人性を排し，業務の負荷を下げる． | ・新しい業務プロセスの確立<br>・顧客の IT ガバナンスに適したシステムの開発 |

### 3.4.2 現状の業務プロセスの確認

コーチ代行管理業務の現状の業務フローを図 3-2 に示す.

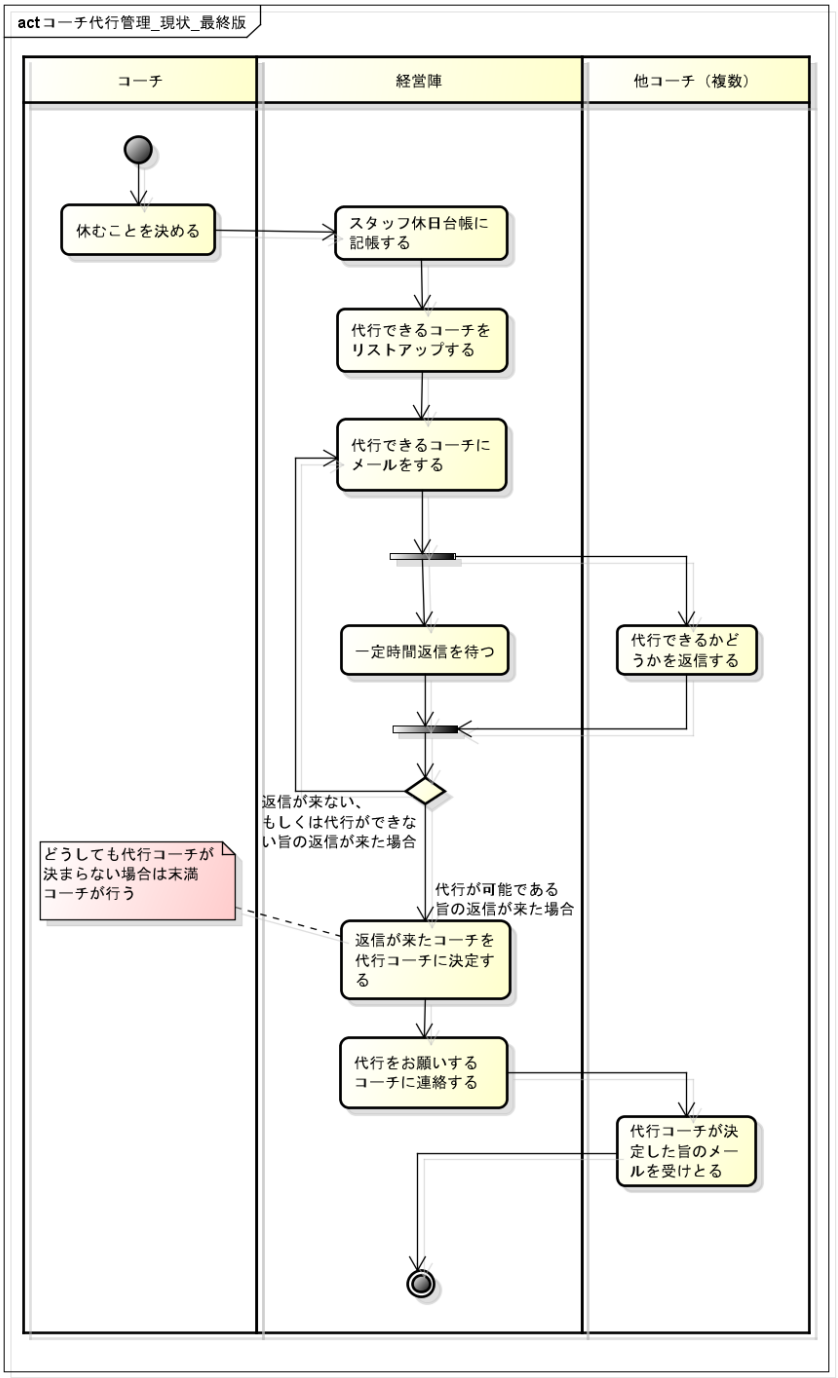


図 3-2　現状業務フロー図（コーチ代行管理）

### 3.4.3 コーチ代行管理業務の実績データ

コーチ代行が月にどのくらいの頻度で発生しているか実績データをまとめたものを表 3-2 に示す．T1 では，月に約 200 回のレッスンが行われているがあるうちの 30 回のレッスンで代行が発生している．

**表 3-2 コーチ代行管理業務の実績データ**

| レッスン回数/月 | 代行発生回数/月 | 代行 1 件が完了するまでの時間 | 備考 |
|---|---|---|---|
| 約 200 回 | 約 30 回 | 1 時間～数日 | ・日曜に集中して発生している<br>・直前まで決まらない事もある |

※T1 で記帳しているレッスン台帳の 2012 年 4 月～2012 年 6 月までのデータから算出

### 3.4.4 コーチ代行管理業務の問題点

- **経営者がコーチ間のハブ（中継役）になっており，業務負荷が大きい**

コーチ代行管理業務の業務フローに注目すると，経営陣が連絡のハブとなっていることが分かる（図 3-2）．また，代行コーチがすぐに見つからない事もあり，数日に渡って他の業務と並行して代行管理業務を行う事になってしまい，経営者にかかる負荷が大きいことが分かる．また，スクール全体の代行件数に注目すると，月に約 200 回のレッスンの内，約 30 件の代行が発生しており，平均すると 1 日に 1 件の代行が発生していることが分かる．また，実際には日曜にまとまって代行が発生する事が多く，集中的に業務負荷が高くなっていることが分かる．

- **経営者の個人的なメールでやりとりしているため，属人性の高い業務となっている**

現状業務では，代行が発生する度に，経営者がコーチに一人ずつメールを送り代行コーチを探している．そのため経営者以外の従業員は「誰がいつ休むのか」といった情報を共有できていなかった．また，代行コーチをお願いする際のプロセスは経験則によって行われていたため，経営者以外の従業員はどのコーチにお願いすればいいのかが分からず，コーチ代行業務が属人性の高い業務であることがわかる．

### 3.4.5 システム導入後の業務プロセスの作成

システム導入後の業務フローを図 3-3 に示す.

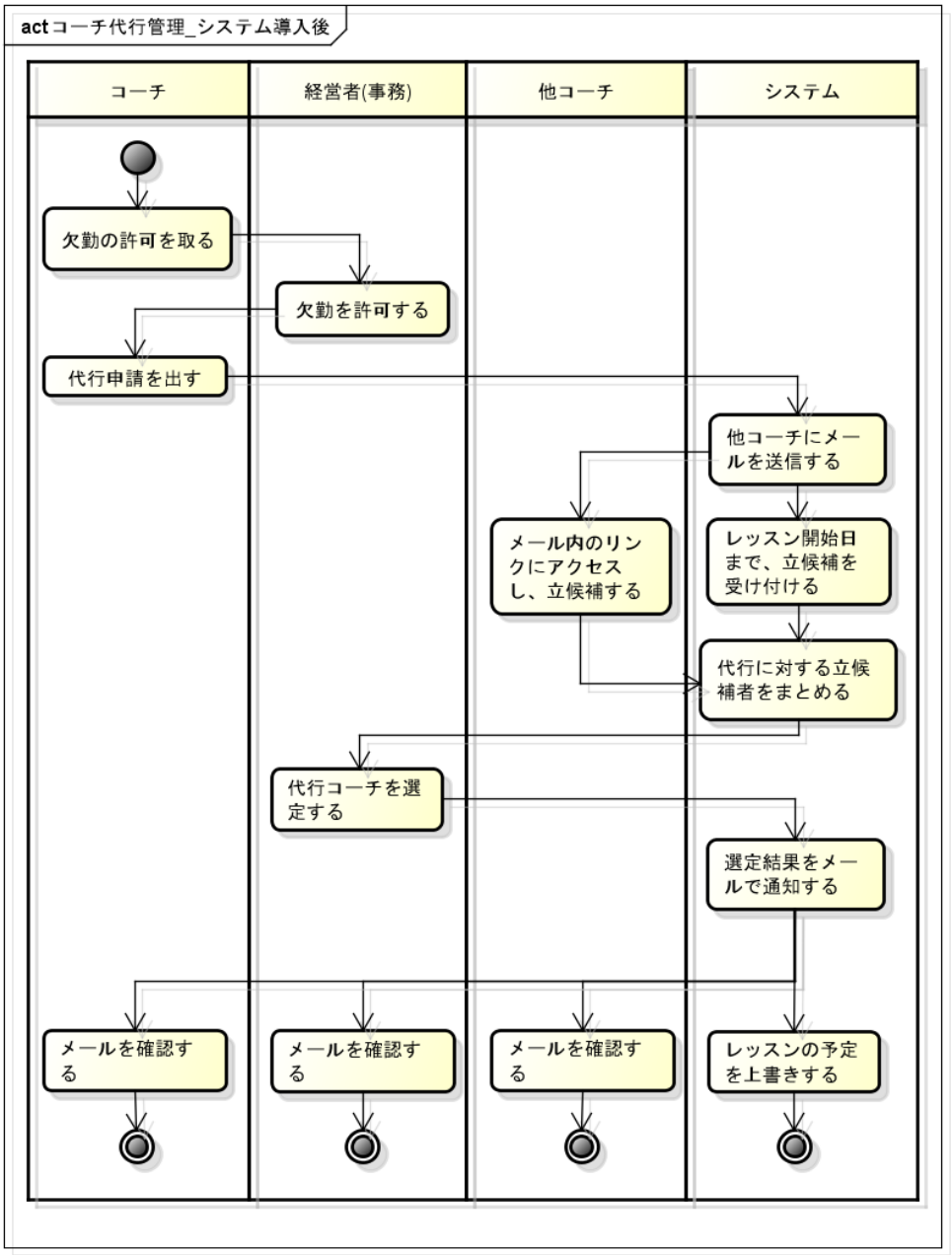


**図 3-3 システム導入後の業務フロー図（コーチ代行管理業務）**

システム導入前は，コーチへの連絡を全て経営者が行っていたが，システム導入後はシステムが代わりにコーチへ連絡を行うため，経営者は「コーチの欠勤を許可する」と「代行コーチを選定する」という 2 つの業務を行うだけで良くなる．システム導入前の業務フローでは 6 つの業務を行っていたことと比べると,経営者の業務負担を大幅に減らしたことになる.

## 3.5　コーチの勤務評価管理業務について

コーチ勤務評価管理業務は，【2.3.7 ロードマップによる中長期経営計画の策定】で作成したロードマップで示した，本年度の取り組みである「コーチの意識改革」を達成するためにプロジェクトメンバが T1 に提案した業務である．

現在 T1 では，明確なコーチの勤務評価制度が存在していない．特に積極的に T1 の業務に取り組んでいるコーチからは，T1 への貢献に比例した適切な評価が行われていない事が，不公平であるという不満が上がっている．

本プロジェクトでは，コーチ勤務評価管理業務を通してコーチ自身を正しく評価し，適正なフィードバックを与える事で，コーチ教育の PDCA サイクルを回していけるような業務プロセスを提案し，システムを提供する事を目的としている．

コーチの評価は「定量的評価」,「定性的評価」「面談によるフィードバック」の 3 つの軸を基に行っていく事にした．それぞれの用語の概要と目的について表 3-3 にまとめる．

**表 3-3　評価制度の概要と目的**

| 用語 | 概要 | 目的 |
|---|---|---|
| 定量的評価 | 「出勤日数」「代行回数」等勤務実績として数値化できるデータを基に評価を行う | ・勤務実績データを用いて評価を行うことで，主観に左右されない公平な評価を行うことができる． |
| 定性的評価 | 「生徒に人気がある」「積極的に雑用をこなす」等<br>直接勤務実績には反映されないが，T1 にとって好ましい貢献に対して評価を行う | ・データには記録されないが，T1 にとって好ましい行動を行うコーチを評価する事で，更なる T1 への貢献を促す．<br>・また，日が浅くても積極的に働くコーチを評価できるメリットがある． |
| 面談によるフィードバック | 定量的評価＋定性的評価を基に経営者とコーチで 1 対 1 の面談による評価を行う | ・コーチにとっては，経営者が自分を理解してくれていることや，フィードバックを貰うことでモチベーションの向上につながる<br>・経営者にとっては，自分の理念や思いをコーチと共有する絶好の場になる．また，コーチの意見を吸い上げる場としても活用できる． |

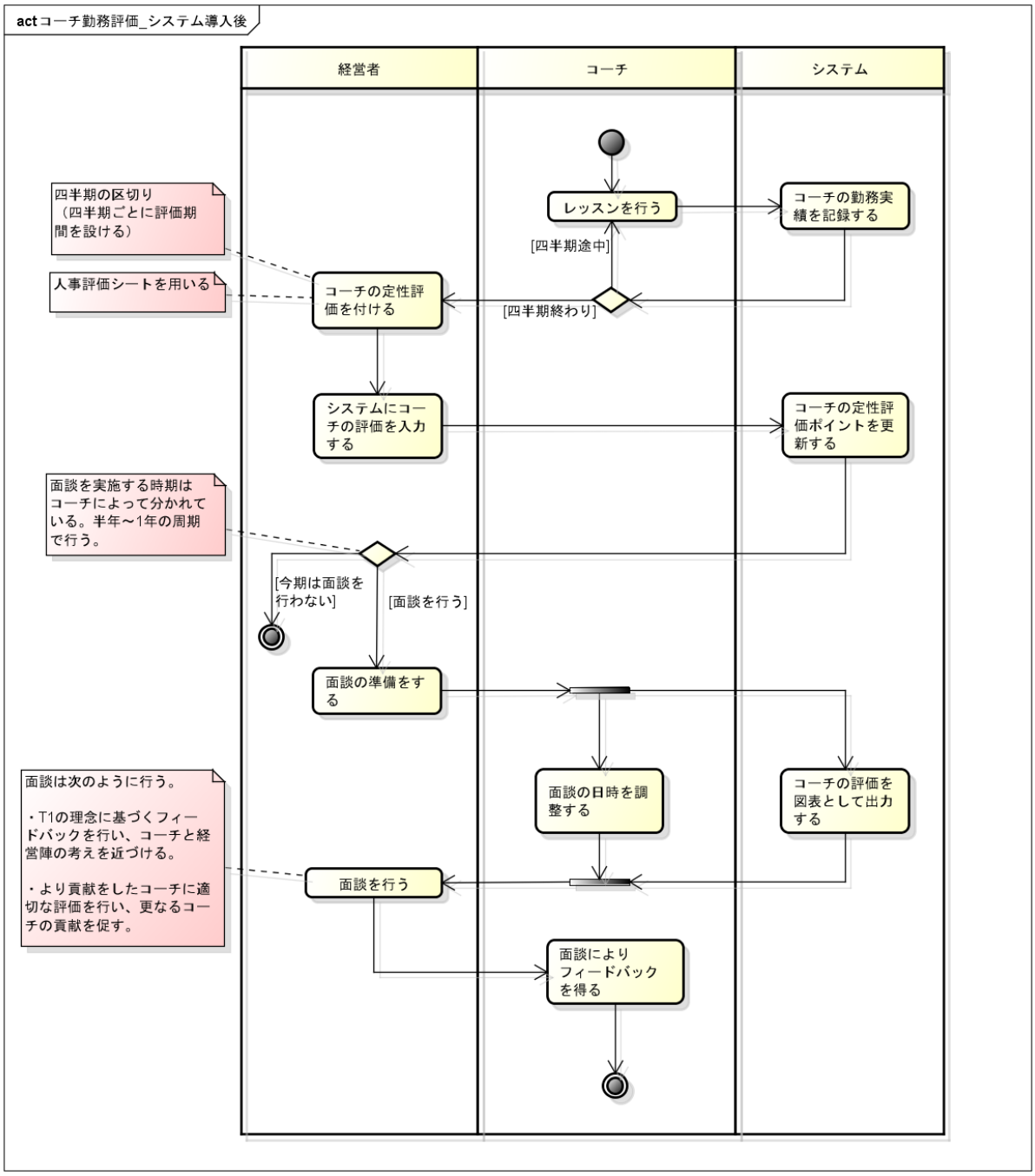


**図 3-4　システム導入後の業務フロー図（コーチ勤務評価管理業務）**

勤務評価管理業務は経営者が入力したレッスンデータに応じてシステムが記録を行う．経営者は四半期に一度各コーチの定性的評価を記入する．そして年に 1～2 度これまでの定量的評価と定性的評価を基に各コーチと面談を行う．

面談の目的は【表 3-3　評価制度の概要と目的】に示した通りである．

## 3.6 開発機能の分析

### 3.6.1 ユーザストーリについて

ユーザストーリとは，顧客がソフトウェアで実現したいと考えている機能を，顧客にとっての分かりやすい言葉で簡潔に記述したもの[5]である．技術的な用語は用いず，顧客にとっての価値，つまりシステムをどう使うかに焦点を当て作成する．

実際にプロジェクトで作成したユーザストーリの一部抜粋を図 3-5 に，その説明を表 3-4 に示す．

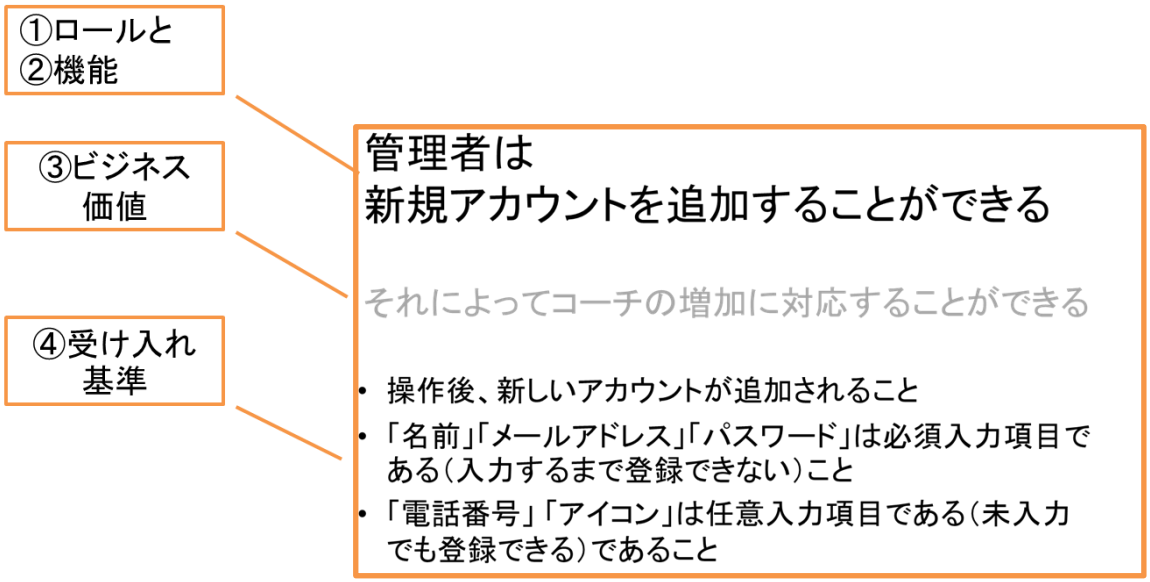


**図 3-5　作成したユーザストーリ**

**表 3-4　ユーザストーリの説明**

| No. | 用語 | 説明 |
|---|---|---|
| ① | ロール | 「誰が行うのか」を記述する． |
| ② | 機能 | 「システムの振る舞いや機能」を記述する． |
| ③ | ビジネス価値 | 「それによってどのような価値が生まれるのか」を記述する． |
| ④ | 受け入れ基準 | ユーザストーリが達成したと判断するための条件を記述する．<br>受け入れテストで用いる． |

### 3.6.2 ユーザストーリの抽出

ユーザストーリの抽出は，次の手順で行った．

1. プロジェクトメンバでブレイン・ストーミングを行う．
2. 顧客とのミーティングでユーザストーリを一つずつ見せ，顧客にとってそのストーリが必要か不必要かを決める．

抽出したユーザストーリは表 3-5 で示すように，大きく 4 つの機能群に分類できる．

表 3-5　抽出したユーザストーリの機能群

| 機能群名 | 説明 |
|---|---|
| アカウント管理機能群 | システムの利用者の管理を行うための基本機能 |
| スケジュール管理機能群 | T1 で行うレッスンの情報（日時・クラス・コーチ等）を管理・閲覧する機能 |
| 代行管理機能群 | コーチ代行管理業務を支援する機能代行の依頼・立候補・代行コーチの選定等を支援する |
| 評価管理機能群 | コーチの勤務評価業務を支援する機能 |

### 3.6.3　ユーザストーリポイントの重み付け

次に，ユーザストーリポイント[5]の重み付けを行った．
ユーザストーリポイントとは，ユーザストーリを見積もるためにチームが使う架空の単位であり，チームがユーザストーリを実装するのに必要な作業量を表している．また，ユーザストーリポイントはユーザストーリ間の相対的な大きさであり，実際の開発にかかる時間は開発メンバによって異なる[6]

### 3.6.4　ユーザストーリの優先度決定

ユーザストーリの洗い出し,及びユーザストーリポイントの重み付けを行った事によって,システムの全体量を把握する事が出来た．しかし，実際に開発が始まるまで本プロジェクトメンバの開発力は不明であり,全てのユーザストーリを実装できるかは分からない.そこで,顧客とのミーティングでユーザストーリをコア機能とサブ機能に分類し，開発を行う優先度を決定した．
表 3-6 にコア機能とサブ機能のユーザストーリ数と総ユーザストーリポイント数の内訳をまとめる．

表 3-6　コア機能およびサブ機能のユーザストーリ数と総ユーザストーリポイント数

| | US 数 | 総 USP |
|---|---|---|
| コア機能 | 22 | 81 |
| サブ機能 | 5 | 20 |
| 計 | 27 | 101 |

※US＝ユーザストーリ　USP＝ユーザストーリポイント

### 3.6.5 作成したユーザストーリ一覧

本節で抽出したユーザストーリについて表 3-7 にまとめた.

**表 3-7 US 一覧表（イテレーション 4 完了時点）**

| 機能群 | US | 分類 | USP |
|---|---|---|---|
| アカウント管理機能 | 利用者がログイン ID とパスワードを用いることでシステムへのログインを制御することができる. | コア | 8 |
| | 管理者は新規アカウントを追加することができる. | コア | 2 |
| | 管理者はアカウント情報を変更することができる. | コア | 2 |
| | 管理者はアカウントを削除することができる. | コア | 2 |
| | コーチは自分のアカウント情報の変更ができる. | サブ | 2 |
| スケジュール管理機能 | 管理者は基本スケジュールの設定ができる. | コア | 8 |
| | 管理者は基本スケジュールの変更開始日が設定できる. | コア | 3 |
| | 管理者は最新スケジュールを週カレンダで閲覧できる. | コア | 8 |
| | 管理者は週カレンダから担当コーチ名を変更することができる. | コア | 3 |
| | コーチは最新スケジュールを週カレンダで閲覧できる. | コア | 2 |
| | 管理者は最新スケジュールを月カレンダで閲覧できる. | サブ | 5 |
| | コーチは最新スケジュールを月カレンダで閲覧できる. | サブ | 5 |
| 代行管理機能 | コーチは自分の週表示カレンダから代行依頼の申請ができる. | コア | 5 |
| | コーチは代行依頼一覧とそれぞれの立候補人数を閲覧できる. | コア | 5 |
| | コーチはシステム上で代行の立候補をすることができる. | コア | 5 |
| | コーチは代行依頼のメール内のリンクをクリックすることで代行への立候補ができる. | コア | 3 |
| | 管理者は代行依頼の一覧を閲覧できる. | コア | 2 |
| | 管理者は代行依頼に対して代行コーチを選択できる | コア | 5 |
| 評価管理機能 | 経営者はコーチの評価基準を設定できる. | コア | 1 |
| | 経営者はコーチの定性評価ポイントを個別に加算することができる. | コア | 3 |
| | 経営者はコーチ毎のポイント数とレベルをグラフで閲覧できる. | コア | 3 |
| | 経営者は月毎の各コーチ勤務内訳と出勤率をグラフで見ることができる. | コア | 5 |
| | 経営者は四半期毎のコーチ全員のポイント伸び数ランキングを見ることができる. | コア | 2 |
| | 管理者は，全期間のコーチ全員のポイント数ランキング表を閲覧出来る | コア | 2 |
| | 経営者は四半期毎のコーチ全員の勤務回数とその内訳，欠勤率を比較することができる. | サブ | 5 |
| | 経営者は全期間のコーチ全員のポイントをブロックチャートで見ることができる. | サブ | 3 |
| | 経営者は HTTP ベーシック認証でコーチ評価ページにアクセスできる. | コア | 2 |

# 第4章　開発フェーズ

## 4.1　本フェーズの目的

本フェーズの目的は，顧客にとって価値のあるシステムを開発することである．実施内容としては，開発期間を 2 週間毎の期間（イテレーション）で区切り，設計・実装・テストをそれぞれ行った．また，次節で示す開発方針に則り，各イテレーションの終わりには顧客に実際にシステムを見せることで顧客からのフィードバックを得ていた．

## 4.2　本フェーズのスケジュール

本フェーズのスケジュールを図 4-1 に示す．

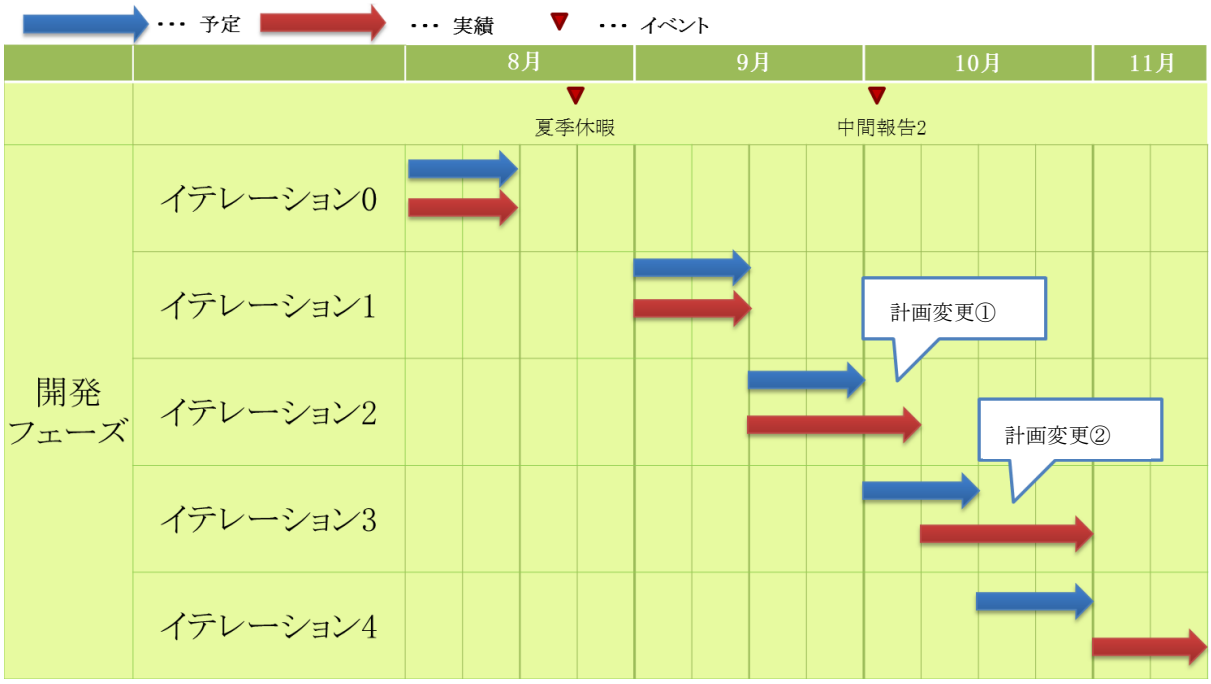


**図 4-1　開発フェーズのスケジュール**

本フェーズは，7 月 31 日から 11 月 9 日までの約 14 週間の期間で行った．

また，本プロジェクトでは反復型開発手法を取り入れ，2 週間毎の 5 つのイテレーションで開発を行った．顧客とのミーティング回数は全部で 6 回，イテレーションの前後に行った．ミーティングの主な議論内容としては進捗報告と実装箇所のレビュー，要件の見直し，及び次イテレーションで取り組むユーザストーリの決定等であった．

イテレーションについては【4.4 イテレーションの流れ】で説明する．

**計画変更について**

当初の予定では，イテレーション 2 を 2 週間で終了する予定であったが，イテレーション 2 の作業内容を見積もる際に，10 月 3 日の中間報告書提出日，及び 10 月 12 日の中間報告発表日への作業時間を見積もっていなかった. そこで, イテレーション期間を延長することで, 顧客と合意を取り，それぞれ一週間の作業時間を確保した．2 週間分の遅れについては評価フェーズに充てていた予定作業時間を切り崩す事で対応した.

## 4.3 本フェーズの方針

本フェーズを進める上の方針，及び開発の進め方について説明する.

### 4.3.1 考慮したリスク

システムを開発する上で以下 2 点のリスクを想定した.

- 顧客はシステム発注経験が無いため，要件抽出フェーズの段階では全ての要件を把握できない可能性がある.
- 顧客は IT に詳しくないため, システムにどのような機能を盛り込むべきか分からない可能性がある．本当は必要（不必要）な機能でも不必要（必要）だと感じてしまう恐れがある.

これらのリスクが顕在化すると，開発したシステムが顧客にとって価値のない，もしくは使って貰えないものになってしまうと考えた.

### 4.3.2 開発方針

前節であげたリスクを顕在化させないために，以下の 3 点の開発方針を考えた.

**表 4-1 開発方針**

| 方針 | 概要 | 備考 |
|---|---|---|
| **①コミュニケーションの機会を創出する** | 開発フェーズでも顧客とコミュニケーションを図り, 要望の変化を的確に把握する | 場合に応じて，経営者だけでなくスタッフやコーチにもミーティングに参加して頂く |
| **②顧客の要求を引き出す努力をする** | システムのレビューを開発中に実施し, イメージや期待との乖離を早期に発見する | システムのデモを見せるだけでなく，実際に操作するレビューを実施し，顧客の気づきを促進する |
| **③要求に対して随時対応する** | 引き出した要求に対し, 要件の変更や実装物へのフィードバックに対応していく | 全ての要件を受け入れるのではなく，事前に必要かどうかの議論を行う |

### 4.3.3 開発手法の検討

前節で述べたように，本プロジェクトの顧客は，IT ベンダと共同によるシステム開発の経験が十分ではない．そのため従来のウォーターフォール型開発[7]のように上流工程で要件を確定した場合，後工程で要件漏れや仕様変更が多発してしまう可能性が高いと考えられた．本プロジェクトでは，開発途中でも顧客からの要求変更や，スコープの変更に対応できること好ましいと考え，今回は反復型開発手法を採用することとした．反復型開発手法について以下に説明する．

#### 反復型開発手法について

反復型開発の流れを図 4-2[8]に示す．

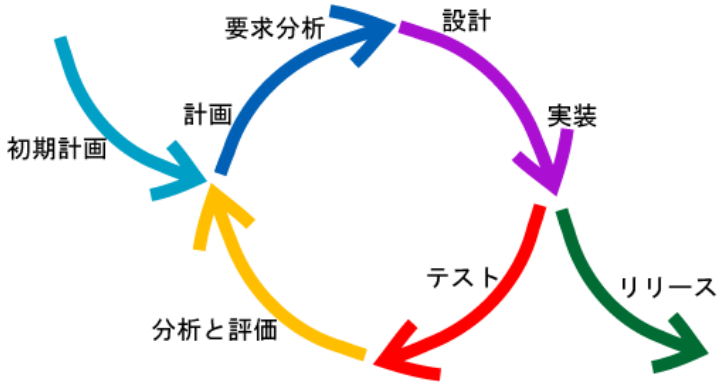


図 4-2　反復型開発のイメージ

反復型開発では，開発フェーズをイテレーションと呼ばれる短い期間に分ける．各イテレーションでは，分析・設計・実装・テストを行い，イテレーションを繰り返しながらシステムの規模を拡大させていく．

反復型開発のメリットは，短いスパンでアウトプットとフィードバックを行うことで，開発中の不測の事態にも柔軟に対応しながらプロジェクト運営を行えることである．

更に本プロジェクトでは，反復型開発手法を踏襲したアジャイル型開発手法を採用する事にした．

#### アジャイル型開発手法について

アジャイル型開発手法は動くソフトウェアを素早く顧客に届けるということに価値を置いた開発手法である．アジャイルマニフェスト[9]と呼ばれる宣言書が存在し，以下の価値観に従い，ソフトウェア開発を進める．

- プロセスやツールよりも，人と人との交流を，
- 包括的なドキュメントよりも，動くソフトウェアを，
- 契約上の交渉よりも，顧客との協調を，
- 計画に従うことより，変化に適応することを重視する．

また，アジャイル開発手法では週単位と非常に短いスパンである事が多い．　[2]

## 4.4　イテレーションの流れ

本プロジェクトで扱うイテレーションの流れを以下に述べる．

### 4.4.1　開発するユーザストーリの選定と確認

イテレーションの初めに，顧客とミーティングを行い，次のイテレーションで取り組むユーザストーリについての選定を行う．選定時には，プロジェクトの開発の進み具合や，プロジェクトが進行する中で生まれた要求の変化等を考慮し，取り組むユーザストーリの変更や優先度の決定を行う．

### 4.4.2　チームタスクの抽出

次に，決定したユーザストーリを基に，ユーザストーリを実現するために必要なチームタスクの抽出を行う．抽出は Redmine のかんばん機能（4.5.1 節参照）を用いて行う．タスク抽出時の画面例を図 4-3 に示す．黄色の長方形で示されている部分がユーザストーリで，一つのユーザストーリに対してテストケースや設計・実装等のタスクを抽出する．抽出したタスクは灰色の正方形で表示される．この時点では全てのタスクを抽出しようとせず，開発途中で新たなタスクが生まれた場合は，その都度追加を行った．

**図 4-3　Redmine を用いたかんばん（チームタスク抽出時）**

### 4.4.3　　プロジェクトメンバへのタスクの割り振り

チームタスクを抽出した後は，プロジェクトメンバでタスクを割り振る．タスク割り振り時の画面例を図 4-4 に示す．担当者が割り振られたタスクは，「新規」から「進行中」に移動し，プロジェクトメンバ毎の色が付けられることで，一目で誰のタスクかが分かるようになっている．

**図 4-4　Redmine を用いたかんばん（プロジェクトメンバへのタスク割り振り時）**

### 4.4.4　　タスクの遂行

次に，プロジェクトメンバは割り振られたタスクを各自遂行する．タスクの進行状態は「進行中」・「レビュー」・「終了」で分かれており，タスクが完了すると「レビュー」に移動させる．レビューは毎日行うデイリーミーティングで行い，そこでレビューした結果問題が無ければ「終了」に移動させる．その後プロジェクトメンバは次の新規タスクに着手する．

また，コアタイムの終わりにプロジェクトメンバは，それぞれ自分の担当したタスクの作業時間を入力する．

### 4.4.5　　開発したシステムのデモと顧客レビュー

イテレーションの終わりに，顧客とミーティングを行う．ミーティングではイテレーションで着手したユーザストーリの確認，開発したシステムのレビューを行い，フィードバックを貰った．レビュー時は，実際に顧客にシステムを操作してもらう事で，見た目だけでなく，実際の使用感や使いやすさ等についてもフィードバックを得た．

### 4.4.6 KPT によるイテレーションの振り返り

顧客へのレビューが終わった後，イテレーション全体の振り返りを行った．振り返りには KPT[10]を用いた．KPT では，表 4-2 に示すように，Keep（続けたいこと），Problem（問題），Try（工夫してみたいこと）の 3 つの項目についてチームの意見を出し合い，次イテレーションでのチームの取り組みに活用する．

**表 4-2　KPT の用語説明**

| 用語 | 説明 |
|---|---|
| Keep（続けたいこと） | ・上手くいって続けたいこと．<br>・イテレーションの中で取り組んで良かった行いについて記入する． |
| Problem（問題） | ・問題だと認識していること<br>・将来的に発生しそうだと考えられる問題や未来のリスク等について記入する |
| Try（工夫してみたいこと） | 問題に対する改善策や，取り入れてみたら良くなりそうな取り組みを記入する |

KPT を行う目的は，チームがもっと効率を高めることが出来ないかを定期的に振り返り，自分たちのやり方を最適に調整することである．また，互いの考えている問題点を共有する事で，コミュニケーションの潤滑化にも繋げようと考えた．

**図 4-5　KPT による振り返りシート**

実際に KPT によって取り組んだ改善策として，

- 毎日コアタイムの前に短時間のミーティングを行う
- コアタイム終了時に必ず Redmine へ作業時間を記録する
- 顧客ミーティング前日に事前デモを行うことを徹底する

等がある．

## 4.5 開発環境について

### 4.5.1 開発を行うために用いたツール

本プロジェクトでは，Redmine を用いてチームタスク及びプロジェクトメンバのタスクを管理し，Git を用いてソースコードの管理を行った．開発に用いたツールの一覧を表 4-3 にまとめた．

**表 4-3 開発に用いたツール**

| 種類 | 名称 | バージョン |
|---|---|---|
| プロジェクト管理ツール | Redmine | 2.0.0 |
| ソースコードバージョン管理システム | Git | 1.2.12.1 |
| 開発ドキュメント共有ツール | Dropbox | 1.6.11 |
| チーム内メーリングリスト | Google Group | |

### 4.5.2 開発用サーバのソフトウェア構成

開発用サーバに用いたソフトウェアの構成を表 4-4 にまとめた．

**表 4-4 開発用サーバのソフトウェア構成**

| 種類 | 名称 | バージョン |
|---|---|---|
| OS | Ubuntu | 12.04 |
| WEB サーバ | SQLite | 3.7.14 |
| WEB アプリケーションフレームワーク | Ruby on Rails | 3.2.6 |
| プログラミング言語 | Ruby | 1.9.3 |
| DB | SQLite | 3.7.14 |
| JavaScript フレームワーク | jQuery | 1.8.1 |
| CSS フレームワーク | Twitter Bootstrap | 2.1.0 |
| テストツール | Rspec | 2.10.1 |

### 4.5.3 開発したシステムの構成

開発したシステムの構成図を図 4-6 に示す．

Ruby on Rails を対象とした，Amazon Web Services（AWS）の IaaS 上に構築された PaaS である Heroku[11][12]上に Web サーバと DB サーバを置いた．また，クラウドメールサービスである Mailgun[13]を用いた．

システムの利用者は PC 上でシステムを利用する事が可能である．また，システムを通じて届いたお知らせに対して，利用者は PC もしくは携帯電話・スマートフォン上でレスポンスを行う事が可能である．

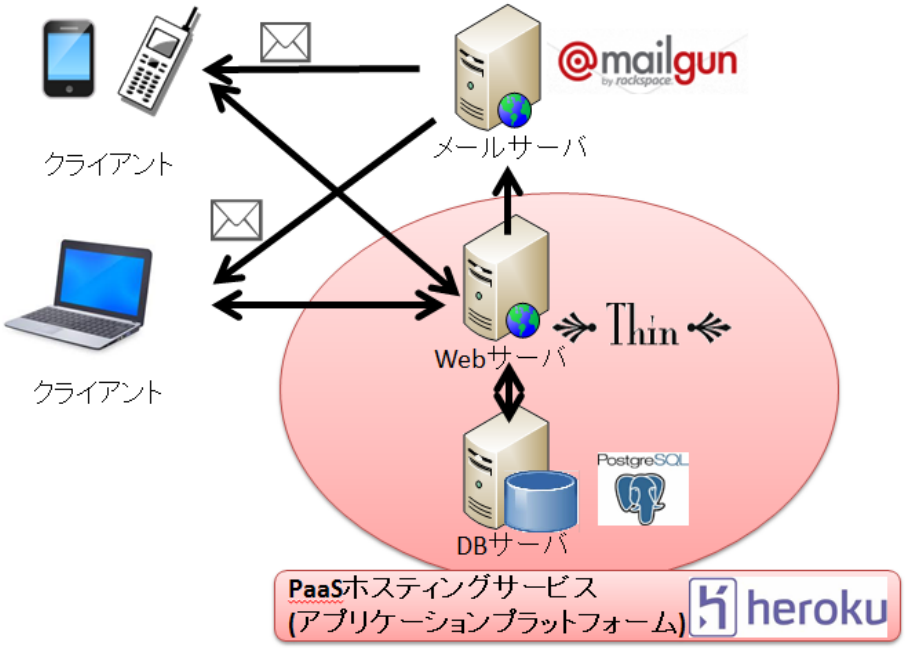


図 4-6　開発したシステムの構成図

## 4.6　開発の実績

### 4.6.1　ユーザストーリの開発実績

各イテレーションにおける開発実績を表 4-5，表 4-6 に示す．

表 4-5 開発したユーザストーリ数

※単位は（個）

| | コア機能 | サブ機能 | 計 |
|---|---|---|---|
| イテレーション 0 | 4 | 1 | 5 |
| イテレーション 1 | 5 | 0 | 5 |
| イテレーション 2 | 4 | 1 | 5 |
| イテレーション 3 | 5 | 1 | 6 |
| イテレーション 4 | 3 | 2 | 5 |
| 計 | 22 | 5 | 27 |

表 4-6　開発したユーザストーリポイント数

※単位は（ポイント）

| | コア機能 | サブ機能 | 計 |
|---|---|---|---|
| イテレーション 0 | 14 | 2 | 16 |
| イテレーション 1 | 26 | 0 | 26 |
| イテレーション 2 | 23 | 0 | 23 |
| イテレーション 3 | 14 | 10 | 24 |
| イテレーション 4 | 4 | 8 | 12 |
| 計 | 76 | 20 | 101 |

イテレーション 0 は，プロジェクトの活動時間の多くを，ルールや仕組みづくり，及び学習コストに充てたため，ユーザストーリポイントの実績は少なめになった．

イテレーション 1，2 は，スケジュール機能や代行機能といったシステムの核となる機能を実装したため，ユーザストーリポイントの実績は他のイテレーションに比べて高くなった．

イテレーション 3 では，評価機能の実装と前イテレーションで間に合わなかった機能の実装を行った．評価機能は開発したユーザストーリ数は最も多いものの，ユーザストーリポイントを低く見積もっていたため，このような結果が出た．

イテレーション 4 では，開発した機能の大半がサブ機能であったことに加え，筆者を含む一部のメンバは活動時間を評価フェーズへの準備に充てていたため，実績値は低めに出たと考えられる．

### 4.6.2　開発したシステムの規模

本プロジェクトで開発したシステムの規模は以下の通りである．

表 4-7　開発したアプリケーションの規模

| 項目 | 実績値 |
|---|---|
| 画面数（ヘッダー・フッター等の画面要素も 1 画面として集計） | 76 枚 |
| データベーステーブル数 | 9 つ |
| Ruby コード有効行数 | 1502 行 |
| HTML コード有効行数（CSS，JavaScript 含む） | 2587 行 |

## 4.7 筆者の担当タスクについて

本節では，本フェーズで筆者が担当したタスクの実績を説明する．

イテレーション毎の筆者が担当したタスクをそれぞれ表 4-8，表 4-9，表 4-10，表 4-11，表 4-12 に示す．

### 4.7.1 イテレーション 0

**表 4-8 担当したイテレーション 0 のタスク**

| イテレーション | ユーザストーリ No. | タスク名 |
|---|---|---|
| **イテレーション 0** | #65 | テスト（項目抽出） |
| | #65 | テスト（コード実装） |

イテレーション 0 では，テストに関わるタスクのみを担当した．これは，筆者に Ruby on Rails の経験が無く，通常のコーディング作業ではチームに迷惑をかけると考えたために，プロジェクトメンバが誰も経験した事の無かったテスト・テストコードの担当を，筆者自身の勉強を兼ねて申し出たためである．

また，テスト・ファースト[14]と呼ばれる手法を参考にし，開発をテスト主導で行うプロセスをプロジェクトに取り入れた（第 6 章品質マネジメント参照）．

### 4.7.2 イテレーション 1

**表 4-9 担当したイテレーション 1 のタスク**

| イテレーション | ユーザストーリ No. | タスク名 |
|---|---|---|
| **イテレーション 1** | #69 | 実装（コントローラ） |
| | #103 | テスト（項目抽出） |
| | #103 | 実装（UI） |
| | #103 | 実装（モデル） |

イテレーション 1 では，実装も担当した．メンバ毎に前イテレーションで固有の担当を設けなかったのは，プロジェクトの方針として，早期にプロジェクトメンバが様々なタスクを担当出来るようにすることで，後の工程での開発力が促進することを狙ったためである．

筆者は UI，モデル，コントローラの実装を経験したことで，Ruby on Rails の知識やチームメンバのコードが読めるようになった．

### 4.7.3 イテレーション 2

**表 4-10 担当したイテレーション 2 のタスク**

| イテレーション | ユーザストーリ No. | タスク名 |
|---|---|---|
| イテレーション 2 | #75 | 設計（クラス） |
| | #75 | 実装（モデル） |
| | #76 | テスト（項目抽出） |
| | #76 | テスト（コードの実装） |
| | #77 | 設計（クラス） |
| | #78 | 設計（クラス） |
| | #78 | テスト（項目抽出） |
| | #99 | 設計（クラス） |

イテレーション 2 では，新たに設計を担当した．設計・実装・テストを全て担当する事で，より効率的に開発が行えるようになった．

### 4.7.4 イテレーション 3

**表 4-11 担当したイテレーション 3 のタスク**

| イテレーション | ユーザストーリ No. | タスク名 |
|---|---|---|
| イテレーション 3 | #79 | 設計（クラス） |
| | #80 | 設計（クラス） |
| | #80 | 設計（モデル） |
| | #80 | 実装（モデル） |
| | #80 | テスト（コードの実装） |
| | #81 | 実装（バッチ） |
| | #00 | 実装（バッチ） |
| | #00 | 実装（コントローラ） |
| | #81 | リファクタリング |
| | #00 | リファクタリング |

イテレーション 3 では，新たにバッチ処理の実装とリファクタリングを行った．特にリファクタリングは 400 行強のコードを 120 行程度に圧縮する事になったが，事前にテストコードを完成させてからリファクタリングを行ったため，他のコードに悪影響を与えることなく修正することが出来た．テスト自動化の恩恵が強く実感できた部分である．

### 4.7.5 イテレーション 4

**表 4-12 担当したイテレーション 4 のタスク**

| イテレーション | ユーザストーリ No. | タスク名 |
|---|---|---|
| **イテレーション 4** | #79 | 実装（コントローラ） |
| | #80 | 実装（UI） |

イテレーション 4 では，ほとんどの機能が実装を完了していたために，筆者は評価フェーズの準備に関わるタスクを担当した．

## 4.8 筆者の担当技術領域について―テスト自動化

本プロジェクトにおいて，筆者はプロジェクトへの「テスト自動化」の導入を担当した．テスト自動化について，目的と方針，筆者の学びについて述べる．なお，テスト実施による実績値については第 6 章 品質マネジメントで説明するため，本章では省略する．

### 4.8.1 テスト自動化の目的

アジャイル型システム開発手法では，短い期間で頻繁にリリースを行う必要がある．一般的にテストはコストがかかるものであり，リリースのたびに手動テストを行うのは非効率的である[15]．そこで筆者は，本プロジェクトにおいて，テスト自動化の導入を試みた．テスト自動化の目的は次の通りである．それぞれの用語はアジャイルサムライ[5]を参考にした．

- 素早いフィードバックを得られる
  コードに変更を加えてテストが失敗したら「どこかを壊してしまった」ということがすぐに分かる．（リリース後ではなく）

- デバック時間を大幅に減らせる．
  リリースのたびに手作業ですべてを再テストする必要が無くなる．簡単なテストを自動化することで，より複雑なテストのためにコストを割くことができる．

- 極めて低コストにリグレッションテストを実行できる．
  リグレッションテストとは，あるコードを変更した時に，他の既存のコードに影響が出ていないかを確かめるテストの事である．自動化されたテストでは，どこを壊してしまったかが正確にわかるため，ソースコードの検索や，デバッガの起動等のコストが削減できる．

- 自信をもってデプロイできる．
  一般的な開発では，開発者はテスト工程まで自分のコードの品質を確認できないが，自動化されたテストは，開発しながら実行を行えるため，開発者は自身を持って自分の開発を行うことが出来る．

### 4.8.2 テストの方針について

筆者はテスト・ファースト[14]と呼ばれる手法を参考にし，プロジェクトに取り入れた

テスト・ファーストとは，実装よりも先にテストを書き，そのテストが動作する最低限な実装を行い，テストが成功したら次のテストを書く，というように開発をテスト主導で行うプロセスである．テスト・ファーストのメリットとして，プロダクトコードの内部構造に依存しないコードが書ける，異常系の処理を意識した設計が出来るという点がある．

反対にテスト・ファーストが向かない条件もある．それは，詳細な仕様が決まっていない機能の場合，動かして試行錯誤してみないとイメージが湧かない機能の場合などである．

筆者は，テスト・ファーストのデメリットをなるべく排除するために，表 4-13 のような基準を設けた．

表 4-13　テスト・ファーストを用いる基準

| | 条件 | 具体的箇所 |
|---|---|---|
| テスト・ファーストを用いない箇所 | 見た目や見せ方が重要で，実際に顧客に見せるまで仕様が確定しない機能 | UI 全般（スケジュールの見せ方など） |
| | 現状行われていない業務のために，仕様の変更が多くあるであろう機能 | 評価機能の全般 |
| テスト・ファーストを用いる箇所 | 上記以外の機能 | アカウント管理機能，スケジュール管理機能，代行管理機能の全般 |

### 4.8.3　本プロジェクトで用いるテストツールについて

テストツールには，Ruby on Rails をサポートしている RSpec を用いた．筆者が本プロジェクトで RSpec を用いた理由として次の 2 点がある．

①テストコードが自然言語的に記述可能なため可読性が高いから．

②テスト項目数が増えても起動時間が高速であるから

まず，「①テストコードが自然言語的に記述可能なため可読性が高い」について説明する．

本プロジェクトでは，テスト・ファーストの方針に則り，全員がテストを書いてから，テストを成功するコードを実装していくという手順を踏む．そのため，コード実装者は全てのテストコードの意図を読み取らねばならない．これを言い換えると，「テストコードは，全てのメンバが分かるよう読みやすいものでなければならない」ということになる．したがって本プロジェクトではテストコードの可読性が重要となる．

図 4-7 に実際に作成した RSpec のコードの一部抜粋を示した．RSpec では describe に「誰が」，context に「何をすると」，it に「どういう結果になる」というように意味のある文章として，テストを記述する事が出来る．

例えば図 4-7 では，次の 3 つのテストが記述されている．

- 管理者ユーザが，評価ページ（evaluation/index）にアクセスすると，HTTP ステータス 200 が返ってくること
- 管理者ユーザが，評価ページ（evaluation/index）にアクセスすると，すべてのコーチのインスタンスが生成されること
- 管理者ユーザが，評価ページ（evaluation/index）にアクセスすると，実績ページが表示されること

このように同じ条件の場合はまとめて記述する事ができ，コードの冗長化を避けることが出来る．また，describe，context，it と上から順番に読むことで日本語として意味のある文章でテストの意味を理解する事ができ，可読性が高いコードとなる．

```
16  #管理者として
17  describe "管理者として　" do
18    before { set_auth_token admin}
19    context "evaluation/indexにアクセスした場合" do
20      before {  get "index" }
21      it "HTTPステータス 200を返すこと" do
22        response.should be_success
23      end
24      it "すべてのコーチのインスタンスが生成されること" do
25        assigns[:users].instance_of?(Array).should be_true
26      end
27      it "実績ページが表示されること" do
28        response.should render_template("evaluations/index")
29      end
30    end
```

**図 4-7　RSpec のコードの一部抜粋**

次に「②テスト項目数が増えても起動時間が高速であるから」について説明する．

本プロジェクトの開発では．テストを実行する機会が非常に多く，新しく機能を実装する・もしくは修正する度に必ずテストを行う．もしテストの実行にかかる時間が長いと，その分だけ開発のボトルネックとなってしまう．また，テストコードを実装するためのコストもかかっているため，テスト自動化のメリットを受けるためには，

$$a < (b - c) \times d$$

※a = テストコードの実装コスト，

b = 手動テストの実行コスト，

c = 自動テストの実行コスト，

d = テストを実行した回数

でなければならない．そのため本プロジェクトにおいては，実行速度の速さ（特にテストケースが増えても処理が速いこと）が重要となる．

図 4-8 に Rspec の実行結果を示す．ここでは 38 個のテストケース（図 4-7 の it 部分）を実行した結果，全て成功しており，実行時間に 0.330682 秒かかっていることが分かる．
最終的にプロジェクトのテストケースは 520 個となったが，実行時間は 4 秒~6 秒弱であり，手動でテストを行う場合と比べて，非常に高速である事が分かる．

```
.....................................

Finished in 0.330682 seconds

38 examples, 0 failures
```

**図 4-8　RSpec 実行結果の例**

# 第5章　　評価フェーズ

## 5.1　　本フェーズの目的

本フェーズの目的は，本プロジェクトで行った業務改善支援，及びシステムの提供によって，顧客の経営改革にどれだけ貢献できたのか，その目的の達成度合いを測定することである．

実施内容としては，実際に顧客にシステムを運用してもらい，アンケートの実施と目標達成指数の計測を行った．

## 5.2　　本フェーズのスケジュール

本フェーズのスケジュールを図 5-1 に示す．

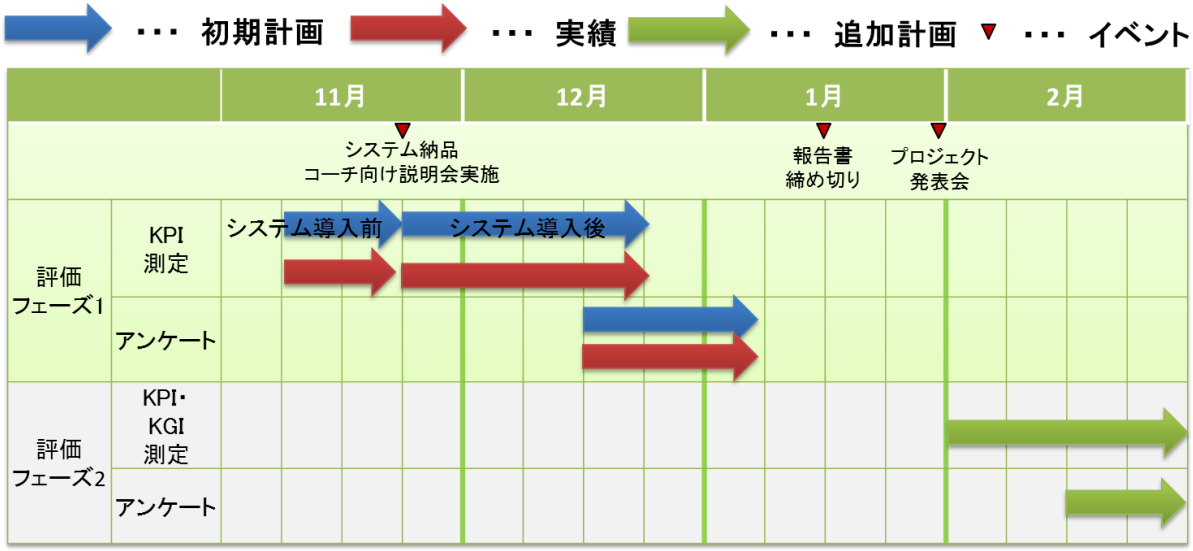


図 5-1　評価フェーズのスケジュール

本フェーズは，フェーズ 1 とフェーズ 2 に分けて実施する予定であり，現在はフェーズ 1 を終えた段階である．

フェーズ 1 は，11 月 9 日から 1 月 9 日までの約 9 週間の期間で行った．11 月 9 日から業務指標である KPI の測定を行い，11 月 23 日には T1 の従業員に対してシステムの使い方や新しい業務プロセスについての説明会を実施し，同日にシステムとマニュアルを納品した．12 月 14 日からは実際に運用した上での評価を得るために，アンケートを実施した．アンケートの詳細については【5.3.1 アンケートについて】，及び【5.4.2 アンケート】で述べる．

フェーズ 2 は，フェーズ 1 において収集しきれなかったデータや従業員へのシステムに対する周知不足を補うため，2 月 1 日から 2 月末までに引き続き評価を実施する予定である．

## 5.3　本フェーズの方針

評価の測定は表 5-1 で示した 2 種類の方法で行う．それぞれの測定方法について説明する．

**表 5-1　評価フェーズで扱う測定データ**

| データ | 対象 | 目的 | 測定方法 |
|---|---|---|---|
| **アンケート** | 経営者<br>スタッフ<br>コーチ | システムを導入した効果や，システムそのものの使いやすさについて定性的データを測定する | アンケート |
| **KPI/KGI[16]** | 業務 | システム導入後の業務改善度合いを，定量的な評価項目の値の推移によって測定する． | 業務時間記録票 |
| | IT | システムの活用と運用の水準が目標に達成したか測定する． | システム DB と業務台帳の比較 |

### 5.3.1　アンケートについて

本フェーズで配布したアンケートの種類を表 5-2 に示す．スタッフ・コーチには，システムの使いやすさや実感効果についてのアンケートを実施した．経営陣には，システムの他にマニュアル，プロジェクト活動についてのアンケートも実施した．

**表 5-2　実施したアンケートの概要**

| アンケート名 | 対象者 | 目的 |
|---|---|---|
| **システムについてのアンケート** | 経営陣<br>スタッフ<br>コーチ | システム導入によって従業員の業務が改善されたかどうか，または，システムの使いやすさ等について評価してもらう． |
| **運用マニュアルについてのアンケート** | 経営陣 | 作成した運用マニュアルの内容が，顧客がシステムを運用する上で役に立つものであったかどうか，分かりやすさや内容の充実度等について評価してもらう． |
| **プロジェクトの取り組みについてのアンケート** | 経営陣 | 戦略立案フェーズから評価フェーズにかけて行ってきたプロジェクト活動について，顧客にとって価値のあるものであったかどうか顧客の視点から評価してもらう． |

### 5.3.2　KPI/KGI について

本フェーズで測定した KPI の項目を表 5-3 に示す．評価対象は業務プロセス・IT ソリューション・IT ガバナンスの 3 つに分類した．それぞれシステム導入後に 3 回の測定日を設け，測定した KPI の推移から，システム活用による効果を評価する．

表 5-3　測定した KPI の項目

| 分類 | 評価観点 | 測定項目 | 測定時期 |
|---|---|---|---|
| **業務プロセス** | IT 成果の<br>モニタリングと評価 | ・代行 1 件あたりの事務作業時間<br>・代行 1 件あたりのメール送信回数 | ・システム導入前<br>・導入後初期（導入～2 週間）<br>・導入後後期（2 週間～4 週間） |
| **IT ソリューション** | 業務情報のデータ化 | ・DB 化（コーチ情報）<br>・DB 化（レッスン情報） | ・システム導入直後 |
| **IT ガバナンス** | システム活用の促進 | ・コーチ代行管理機能の活用度<br>・勤務評価管理機能の活用度 | ・導入後中期（10 日後時点）<br>・導入後後期（20 日後時点） |
| | サービスデスクとインシデント管理 | ・コーチ利用時のトラブル回数<br>・管理者利用時のトラブル回数 | |
| | セキュリティの保証 | 不正アクセストラブル回数 | |

## 5.4 実績

### 5.4.1 システムトラブルについて

はじめに，本フェーズで発生したシステムトラブルについて説明する．
本フェーズで発生したトラブルの内訳を表 5-4 に示す．

**表 5-4 評価フェーズで発生したシステムトラブルの内訳**

| 報告日 | 分類 | 内容 | 原因 | 解決方法 | 解決日 |
|---|---|---|---|---|---|
| **12/04** | 管理者トラブル | 基本スケジュールを変更すると，最新スケジュールの曜日がずれて登録される，元のデータが消去される． | 開発環境（SQLite）と本番環境（PostgreSQL）の違いによる，SQL 実行結果の相違． | SQL 文修正 | 12/04 |
| **12/11** | 管理者トラブル | アウトドアコートの基本スケジュール設定ページにアクセス出来ない． | データベース初期値の入力忘れ． | データベース初期値入力 | 12/12 |
| **12/11** | コーチトラブル | システムから送られるメールが 18 人中 11 人のコーチに届いていない． | RFC に準拠していないメールアドレスを使用しているユーザにより，Rails の処理が途中で停止した． | エラーが起きても処理が続行するよう設定．また，RFC に準拠していないメールアドレスの登録を拒否． | 12/12 |

表に示す通り，12 月 4 日に 1 件，12 月 11 日に 2 件のトラブルが発生した．これにより評価フェーズ 1 期間において，T1 の従業員が実際にシステムを使用した期間は 1 週間程度となってしまい，十分な測定結果を得る事が出来なかった．しかし，経営陣へのヒアリングによると，トラブル解決後は問題なくシステムを運用出来ている事が確認できた．このことから評価フェーズ 2 においては，十分な測定結果を得られると予測している．

### 5.4.2 アンケート回収について

評価フェーズ 1 に行ったアンケートの集計結果を示す．アンケートの実施は以下の通りに行った．また，各アンケートの名称及び目的は【5.3.1 アンケートについて】の通りである．

- コーチへのアンケート
  - 実施アンケート
    - システムについてのアンケート
  - 回答人数
    - 9 名
- スタッフへのアンケート
  - 実施アンケート
    - システムについてのアンケート
  - 回答人数
    - 4 名
- 経営陣へのアンケート
  - 実施アンケート
    - システムについてのアンケート
    - 運用マニュアルについてのアンケート
    - プロジェクトの取り組みについてのアンケート
      - 戦略立案フェーズの取り組みについて
      - 要件抽出フェーズの取り組みについて
      - 開発フェーズの取り組みについて
      - プロジェクト全体の取り組みについて
  - 回答人数
    - 2 名

これらのうち，【5.4.1 システムトラブルについて】の影響により，システムについての十分なアンケートの結果を得る事が出来なかった．次節では，経営陣に行った運用マニュアル及びプロジェクト活動についてのアンケートの結果について記す．

5.4.3　　アンケート結果

**運用マニュアルについてのアンケート結果**

作成した運用マニュアル【付録 D1-2】について，アンケートの結果を表 5-5 と図 5-2 に記す.

**表 5-5　運用マニュアルについてのアンケート結果**

| 運用マニュアルについて | 評価方式 | 経営者 1 | 経営者 2 |
|---|---|---|---|
| **Q 1．このマニュアルがあれば, 今後システムを運用できそうだと感じますか** | 5 段階 | 4 | 4 |
| | 理由（自由記述） | わかりやすくて丁寧にまとめてくれているから | さほど難しくないから |
| **Q2. マニュアルの内容は容易に理解できたと思いますか** | 5 段階 | 5 | 4 |
| | 理由（自由記述） | 構成が見やすい | 簡単だから |
| **Q3. マニュアルから求める情報を素早く得る事が出来ると思いますか** | 5 段階 | 4 | 3 |
| | 理由（自由記述） | 目次や見出しが分かりやすい | |
| **Q4. 誤字・脱字が無く, 表記や表現に一貫性があったと思いますか** | 5 段階 | 5 | 3 |
| | 理由（自由記述） | 非常にまとめてくれている | |
| **Q5. 問い合わせ先など困った時のための情報が明らかになっていると感じましたか** | 5 段階 | 5 | 3 |
| | 理由（自由記述） | 困った時の解決方法という項目でまとめてくれているので | |

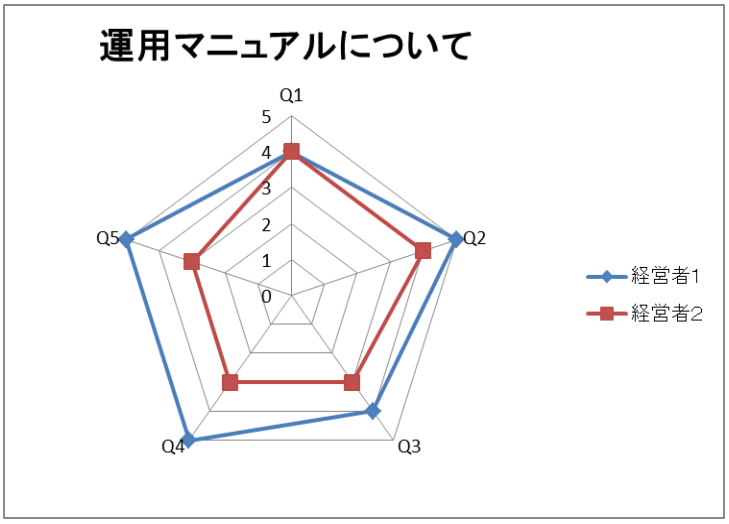


**図 5-2　運用マニュアルに対する評価のレーダーチャート**

本プロジェクトでは，コーチ・スタッフに向けた操作方法を簡易的に示したクイックマニュアルと，経営者に向けたシステムの運用法を示した運用マニュアルをそれぞれ作成した．運用マニュアルでは，システムに詳しくない顧客でもシステムを運用していく事が出来るよう，表現や構成，情報の網羅度に注意した．

アンケート結果によると，回答者の意見が割れているが，実際にシステムの管理者として運用を行っているのが経営者 1 の末満弘江様であり，全ての項目で 4 点以上を付けられているため，運用時の問題点は無いといえる．

**戦略立案フェーズの取り組みについてのアンケート結果**

戦略立案フェーズについてのアンケート結果，及びレーダーチャートをそれぞれ表 5-6，図 5-3 に示す．

**表 5-6　戦略立案フェーズの取り組みについてのアンケート結果**

| 戦略立案フェーズの取り組みについて | 評価方式 | 経営者１ | 経営者２ |
|---|---|---|---|
| **Q１. 理念の明確化と経営戦略の策定を通じ，プロジェクトが取り組むべき課題を見つけ出した今回のアプローチは適切だったと思いますか.** | 5 段階 | 5 | 5 |
| | 自由記述 | 自分たちの原点に戻って経営理念を明確化していく事ができた．アプローチの仕方も新鮮で，親身になって考えてくれたことに非常に感謝している． | 話し合いの中で T-1，あるいは自分の思いが明確になったため． |
| **Q2. 経営戦略として，初期段階では弱点改善戦略として内部の経営基盤を整え，その後新規顧客や既存顧客への満足度向上に向けた積極戦略を取るという方針を定義しました．この判断は適切だったと思いますか.** | 5 段階 | 4 | 4 |
| | 自由記述 | 全てを一度に解決していく事ができないので，方針を決めてくれてよかったと思う． | |
| **Q3. 弱点改善戦略として「コーチ代行管理業務」や「コーチ勤務評価業務」の見直しを具体的なアクションプランとして活動を行ってきましたが，このアクションプランの選定は適切だったと思いますか.** | 5 段階 | 4 | 5 |
| | 自由記述 | SANITY が関わってくれなかったらずっと変わらなかっただろう．改善に役立つシステムを選んでくれたと思う． | 各コーチと密度の濃いコミュニケーションが取れると思われるため． |
| **Q4. Q３が適切であった場合，本年度の私たちプロジェクトの活動は戦略の実現に貢献したと感じますか.** | 5 段階 | 4 | 4 |
| | 自由記述 | 後は自分たちが活用していくことだろう． | |
| **Q5. その他，戦略立案フェーズを通じての感想や意見などがありましたらご記入をお願いします.** | 自由記述 | T-1 インドアテニススクールを細かく分析して頂き，発展のプロセスを考えて頂き，私たちにとって非常に有意義なミーティングでした． | |

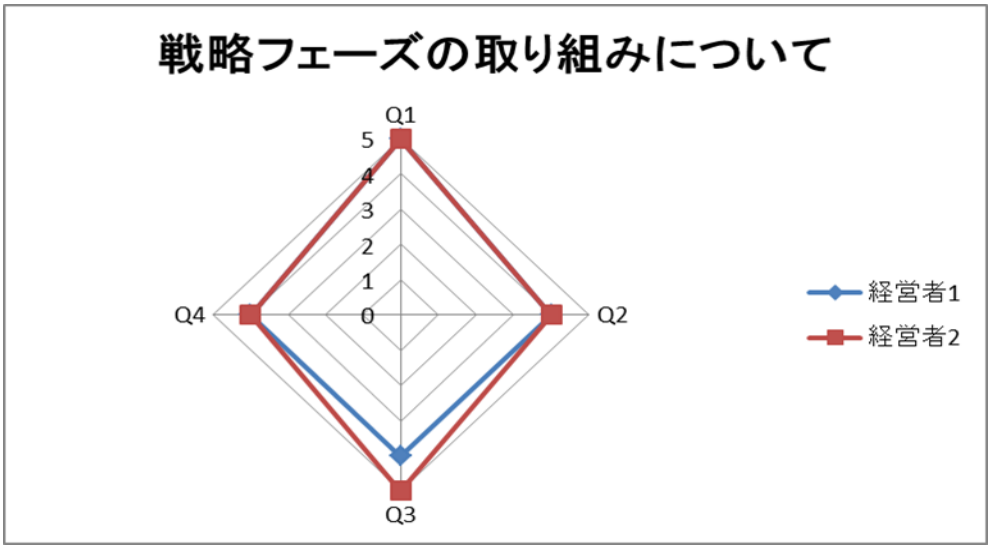


図 5-3　戦略立案フェーズの取り組みに対する評価のレーダーチャート

本プロジェクトでは戦略立案フェーズとして 2012 年 5 月 27 日から 6 月 29 日までの間に，T1 の経営分析を経て経営戦略の方針と具体的なアクションプランを定義した. 戦略フェーズでは, 現場で導入できないというリスクを避けるため, アイデアベースのシステム案を割け，顧客の持つ経営理念を原点とし，その理念を実現するための具体的な道筋として戦略の立案及び，その為に支援できることを模索するというトップダウンのアプローチに切り替えた.

アンケート結果によると，全項目で 5 点中 4 点以上を獲得することができた．特に Q1 では両者から 5 点の評価が付いた．これは，プロジェクト開始直後に，顧客にいきなり戦略の話をするのではなく，顧客の想いを中心に一つずつ進めていったことが顧客をプロジェクトに引き込む良いきっかけになったのである.

**要件抽出フェーズの取り組みについてのアンケート結果**

要件抽出フェーズについてのアンケート結果，及びレーダーチャートをそれぞれ表 5-7，図 5-4 に示す．

**表 5-7　要件抽出フェーズの取り組みについてのアンケート結果**

| 要件抽出フェーズの取り組みについて | 評価方式 | 経営者 1 | 経営者 2 |
|---|---|---|---|
| **Q1.1.** システム構想段階でのモックアップによってシステムのイメージを掴み，議論がやりやすくなったと感じじましたか | 5 段階 | 5 | 5 |
| | 自由記述 | イメージが掴みやすい | とても具体的かつ視覚的で分かりやすかった |
| **Q1.2.** 要件抽出フェーズ段階で想像していたイメージと実際に作成したシステムの間に，想定外のギャップを感じましたか | 5 段階 | ※5(1) | ※4(2) |
| | 自由記述 | | きっちり作動できればとても有意義 |
| **Q2.1.** ユーザストーリとしてシステムの機能を整理しましたが分かりやすいと感じましたか | 5 段階 | 4 | 4 |
| | 自由記述 | 優先順位をつけることでより明確化される． | 優先順位など話し合う中で決められたこと |

※※本質問は，数値が低いほど評価が良くなる質問形式であった．
よって 1→5，2→4 として集計した．

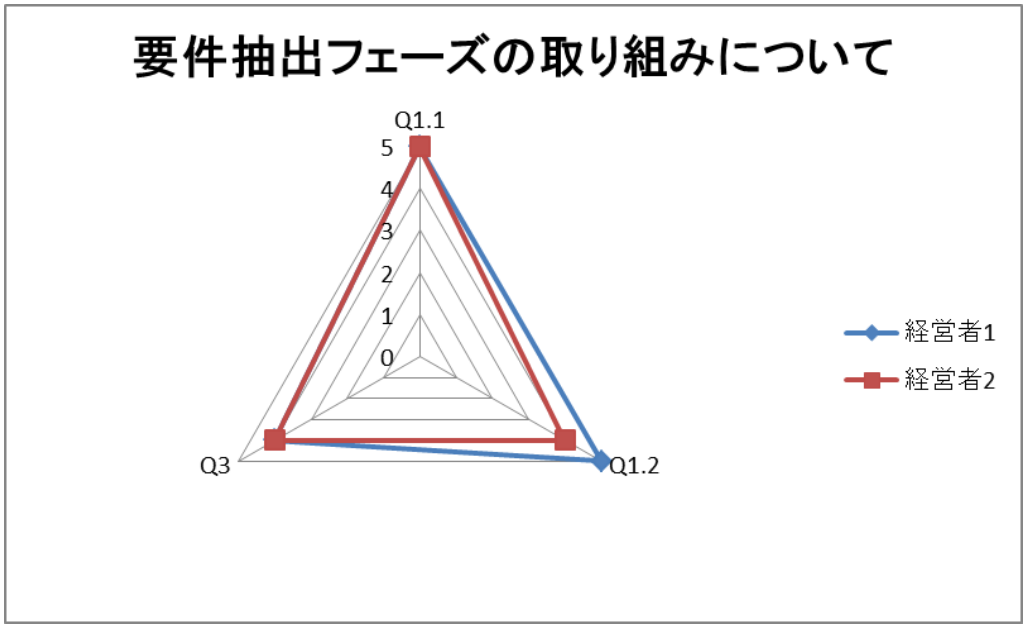


図 5-4　要件抽出フェーズの取り組みに対する評価のレーダーチャート

本プロジェクトでは要件抽出フェーズとして，2012 年 6 月 30 日から 7 月 30 日までの間に, 開発するシステム案についての議論や, 具体的な機能要件についての取り決めを行った. IT システムの発注が初めてである顧客に配慮し，出来る限り分かりやすく議論できるように工夫をした．例えば，システム案について議論する際，具体的なイメージとしてモックアップを用いて早期に具体的なイメージを共有できるように工夫した．また，ユーザストーリを用いて利用者目線で機能を整理する事でシステム開発に関する知識が無い顧客にとってもわかりやすくなるよう努めた.

アンケート結果によると全項目で 5 点中 4 点以上を獲得することができた. 特に Q1.1 では両者から 5 点の評価が付いた．これは，一つひとつの機能を細分化し，実際の使用イメージをモックアップ等で表現したことで顧客の理解を高めようとする取り組みが，顧客にとっても分かりやすかった，議論が行いやすかったとういうことであり，本フェーズの狙いが成功した.

開発フェーズの取り組みについてのアンケート結果

要件抽出フェーズについてのアンケート結果，及びレーダーチャートをそれぞれ表 5-8，図 5-5 に示す.

表 5-8　開発フェーズの取り組みについてのアンケート結果

| 開発フェーズの取り組みについて | 評価方式 | 経営者 1 | 経営者 2 |
|---|---|---|---|
| Q 1．本プロジェクトでは開発期間をイテレーションという短期間に分割し，その節目に私たちが開発したシステムの実物を T 1 様に確認して頂き，フィードバックを頂く機会を複数設けました．この取り組みは良かったと思いますか． | 5 段階 | 4 | 4 |
| | 自由記述 | 非常に丁寧な取り組みであった．私たちがあまりフィードバックできなかったことが申し訳ない | まさにイメージ通りであったため |
| Q2．本プロジェクトでは，一回のイテレーションを 2 週間として取り決めました．この 2 週間という期間は適切だったと思いますか | 5 段階 | 4 | 4 |
| | 自由記述 | 間があきすぎず，近すぎず，適切な期間の取り方であったと思う． | １週間では短すぎ，３週間では空きすぎて忘れてしまう恐れがあると感じるため |
| Q3．1．実際のシステムを見て，触ってみる事で，開発前に思いつかなかった要求に気付くことができたと思いますか． | 5 段階 | 4 | 4 |
| | 自由記述 | 操作のしやすさ，不便な所など，気づくことが出来る | 不必要なものを削除できたと思う |
| Q3．2．システムについての具体定期な意見を出しやすいミーティングであったと思いますか | 5 段階 | 5 | 4 |
| | 自由記述 | 毎回，SANITY の皆さんの準備が完璧で，相互に意見を出しやすいミーティングが出来た． | 細かいところまで話し合えたから |
| Q3．3．実際に利用者に開発段階から操作して使用して頂くという取り組みは，一般的にはあまりない特徴的な取り組みです．実際にシステムを操作する事で，何らかの気付きを得る事は出来たと感じますか（操作してみる事に価値があったか） | 5 段階 | 4 | 4 |
| | 自由記述 | | |
| Q4．Q3 のようなミーティングの場で伝えた自分の要望に対して，学生チームシステムに反映するなどの対応をしていたと思いますか | 5 段階 | 5 | 4 |
| | 自由記述 | 細かいところまで，迅速に対応してくれた． | |

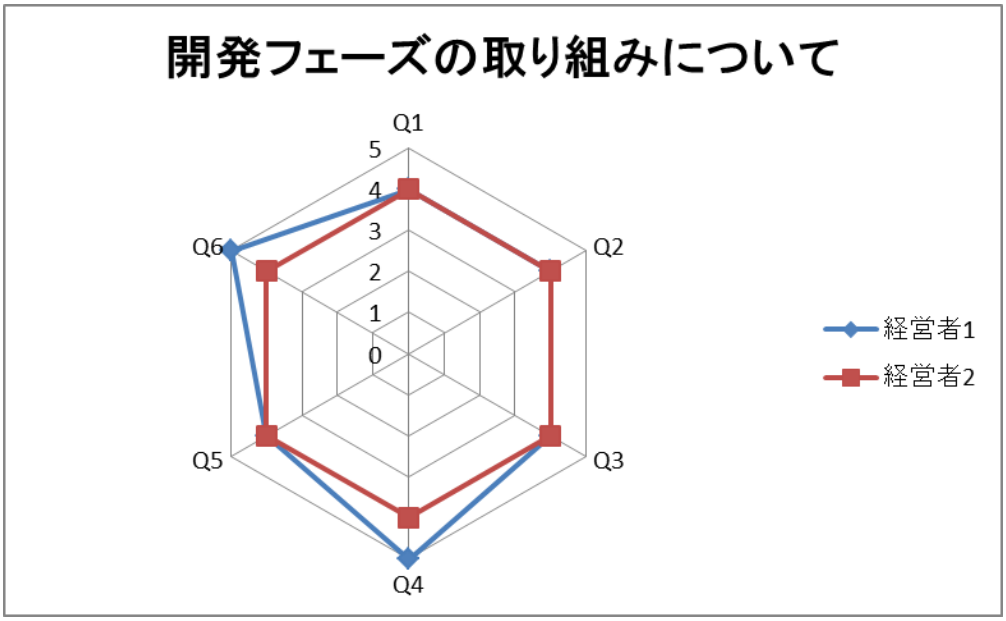


図 5-5　開発フェーズの取り組みに対する評価のレーダーチャート

本プロジェクトでは開発フェーズとして，2012 年 7 月 30 日から 11 月 9 日までの間に，要件抽出フェーズで定義したシステムの開発を行った．特にアジャイル開発を取り入れ，作成したシステムを開発の早い段階から 2 週間に一度，顧客に確認して貰った．これは，早期にイメージとの乖離を修正し，また当初洗い出し切れなかった細かい要望を積極的に取り入れていく為である．

アンケート結果によると，全項目で 5 点中 4 点以上を獲得することができた．特に Q3.2, Q4 はミーティングについての設問であるが, 経営陣のうち一人からは 5 点満点を獲得することができた．これは，毎回のミーティング前の事前デモやユーザストーリの整理を行った事によって，当日のミーティングをスムーズに行う事が出来たためであると考えられる．しかしミーティングに関する評価は 5 点である一方で，同経営者が Q1，Q3.1，Q3.3 で 4 点をつけている．この 3 つの質問はどれも，フィードバックや顧客の要求の吸出しに関する設問である．この 4 点が何か満点にするには足りていない要素があったため付いた評価であったと仮定すると，以下の事象が考えらえる．

- プロジェクトとして要求を引き出す場は用意されていたが，顧客が要求を伝えることが出来なかった．（要求を言語化する事が出来なかった.）
- 学生チームに伝えた要求が，顧客の想いとは違う形で受けとられてしまい，反映されてしまった．（学生チームは顧客要求の達成と捉えたが，顧客としては達成されていなかった.）

反省として，学生チームは顧客の要求を受け入れる場を設けるだけでなく，顧客が持つ IT に対する知識不足・認識不足からくる不安や要求を受け入れ，合意を取る更なる工夫が必要であったと考えられる．これは今後の反省とする．

プロジェクト全体の取り組みについてのアンケート結果

最後に，本プロジェクト全体を通しての設問を設けた．本プロジェクトの基本方針が「顧客と密なコミュニケーションを図る」であることを踏まえた上で，プロジェクト全体の取り組みについてのアンケート結果，及びレーダーチャートをそれぞれ表 5-9，図 5-6 に示す．

表 5-9 プロジェクト全体の取り組みについてのアンケート結果

| プロジェクト全体の取り組みについて | 評価方式 | 経営者 1 | 経営者 2 |
|---|---|---|---|
| **Q１．私たちは Face to Face のコミュニケーションを重視し，実際に顔を合わせたミーティングを 19 回実施しましたが，本プロジェクトにおいてコミュニケーションの機会は十分であったと思いますか？** | 5 段階 | 5 | 4 |
| | 自由記述 | よい雰囲気の中でミーティングを行うことができた． | |
| **Q1．2 情報共有の手段として，メールや Dropbox を用いたファイル共有手段を用いてきましたが，本プロジェクトにおいて情報共有の手段は十分であったと思いますか．** | 5 段階 | 5 | 4 |
| | 自由記述 | | |
| **Q3．システム開発に限らず，今回のプロジェクトに対する私達 SANITY の活動は T1 にとって有益であったと感じますか** | 5 段階 | 5 | 5 |
| | 自由記述 | 非常に有意義でした．当初はシステムが本当に完成するのかと思いましたが，きっちり仕上げて頂き，感心するばかりです．経営分析から関わって頂いたことで，私達自身の経営理念，目指すべき方向を考え，見直すきっかけになりました．ありがとうございました．後は SANITY の努力を活かすべく頑張ります | システムを完成できたことはいうまでもなく，その中で T-1 の理念，自分の思い等をカウンセリング的な場面を設けて頂き非常に感謝している． |

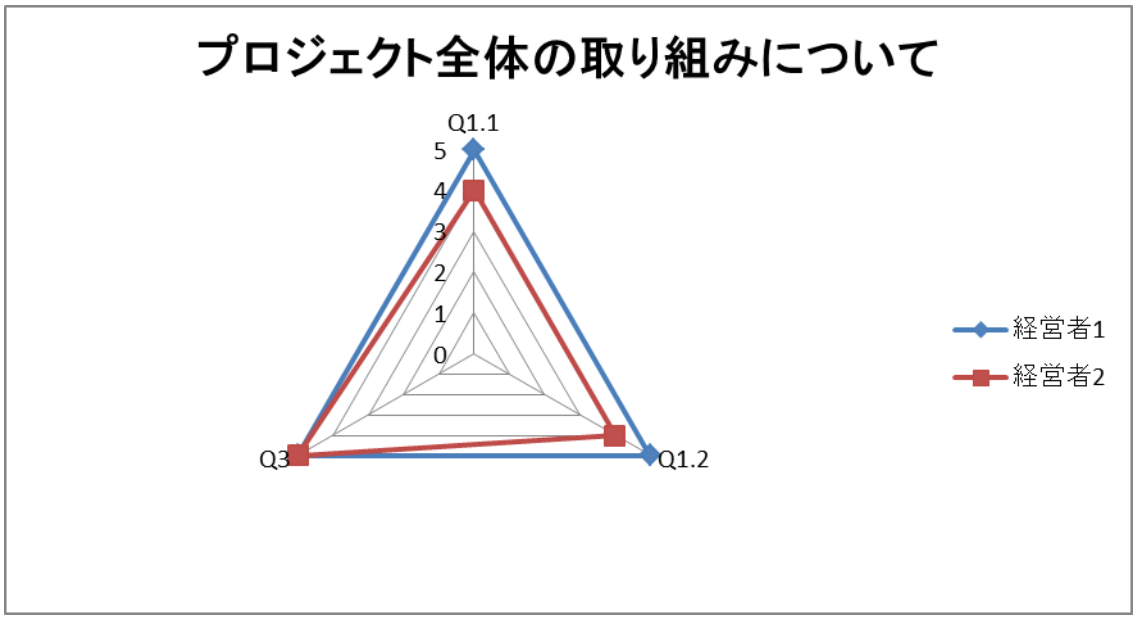


図 5-6　プロジェクト全体の取り組みに対する評価のレーダーチャート

プロジェクト全体についても，全項目で 5 点中 4 点以上を獲得することができた．特に Q3 はプロジェクトを総括した評価であり，顧客にとって本プロジェクトが有意義であったことが伺える．本設問の自由記述欄に注目すると，両者とも経営戦略フェーズで行った経営理念の抽出や分析に非常に価値を感じており，本プロジェクトが顧客とのコミュニケーションを重視し，超上流から一貫した開発工程を進めてきたという特徴が評価されていると考えられる．

# 第6章　品質マネジメント

本プロジェクトにおいて筆者は品質マネジメントを担当した．本章では筆者が実施した品質マネジメントの計画，実施内容，及び実績と，そこから得られた知見やチームに与えたコントロールについて述べる．

## 6.1　品質とは

ソフトウェア品質会計[17]によると品質とは「指定された特定の条件で利用する場合の，明示的または暗示的なニーズを満たすソフトウェア製品の能力」と定義されている．

品質は図 6-1 に示す 4 つの種類[18]に分類される．また，それぞれの品質の概要について表 6-1 に示す．

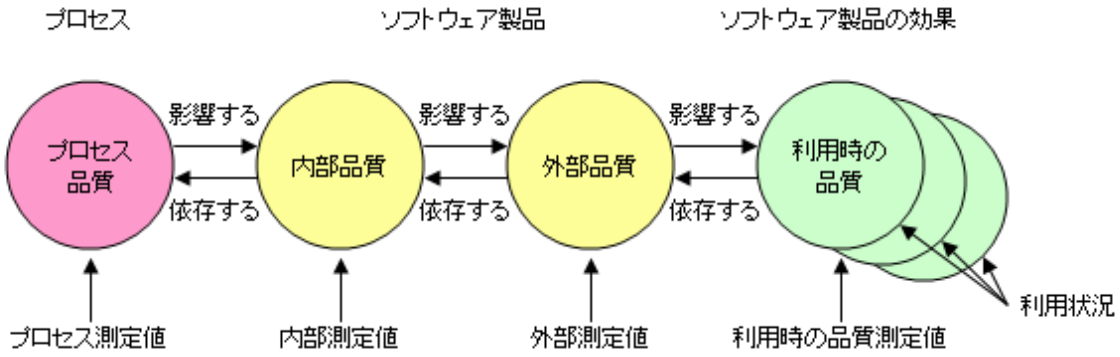


**図 6-1　品質の種類**

**表 6-1　品質の種類の概要**

| 品質の種類 | 概要 | 詳細 |
|---|---|---|
| 利用時の品質 | 利用者の視点でのソフトウェア製品の品質 | ユーザのニーズや目的をどれだけ満たしているか |
| 外部品質 | 設計仕様とプログラムの合致度合い | ソフトウェアの設計と，実際に作成されたソフトウェアがどれだけ適合しているか |
| 内部品質 | ユーザの要求と設計仕様の合致度合い | ユーザのニーズとソフトウェアの設計がどれだけ適合しているか |
| プロセス品質 | 設計及び開発のやり方や手順の品質 | 設計及び開発の品質をあげるための手順がどれだけ洗練されているか |

ソフトウェア品質を測定するための世界標準として定められた ISO/IEC25000[18]に基づき，本プロジェクトの品質マネジメントで重視する品質特性を決定した．

ISO9126 で示されているソフトウェア品質特性を表 6-2 に示す．

**表 6-2　ソフトウェア品質特性**

| 品質特性 | 英語名 | 副品質特性 | 説明 |
|---|---|---|---|
| 機能性 | Functionality | 合目的性 | ソフトウェアが，指定された条件の下で利用されるときに，明示的及び暗示的必要性に合致する機能を提供するソフトウェア製品の能力 |
| | | 正確性 | |
| | | 相互運用性 | |
| | | セキュリティ | |
| | | 機能性標準適合性 | |
| 信頼性 | Reliability | 成熟性 | 指定された条件の下で利用するとき，指定された達成水準を維持するソフトウェア製品の能力 |
| | | 障害許容性 | |
| | | 回復性 | |
| | | 信頼性標準適合性 | |
| 使用性 | Usability | 理解性 | 指定された条件の下で利用するとき，理解，習得，利用でき，利用者にとって魅力的であるソフトウェア製品の能力 |
| | | 習得性 | |
| | | 運用性 | |
| | | 魅力性 | |
| | | 使用性標準適合性 | |
| 効率性 | Efficiency | 時間効率性 | 明示的な条件の下で，使用する資源の量に対比して適切な性能を提供するソフトウェア製品の能力 |
| | | 資源効率性 | |
| | | 効率性標準適合性 | |
| 保守性 | Maintainability | 解析性 | 修正のしやすさに関するソフトウェア製品の能力．修正は，是正若しは向上，又は環境の変化，要求仕様の変更及び機能仕様の変更にソフトウェアを適応させることを含めてもよい． |
| | | 変更性 | |
| | | 安定性 | |
| | | 試験性 | |
| | | 保守性標準適合性 | |
| 移植性 | Portability | 環境適応性 | ある環境から他の環境に移すためのソフトウェア製品の能力．<br>備考：環境には組織，ハードウェア又はソフトウェアの環境を含めてもよい． |
| | | 設置性 | |
| | | 共存性 | |
| | | 置換性 | |
| | | 移植性標準適合性 | |

## 6.2　品質マネジメントとは

品質マネジメントは，プロジェクトが取り組むべきニーズを満足するために，品質方針，品質目標，品質に対する責任などを決定するプロセスと活動である[1]．本プロジェクトではPMBOK（プロジェクトマネジメント知識体系ガイド）[1]に基づき，品質マネジメントプロセスを実施した．

PMBOK で示されているプロジェクト品質マネジメントプロセスの概要を表 6-3 に示す．

**表 6-3　品質マネジメントプロセスの概要**

| マネジメントプロセス | 概要 |
|---|---|
| 品質計画 | プロジェクトおよび成果物の品質要求事項または品質標準，あるいはその両方を定め，プロジェクトでそれを順守するための方法を文書化するプロセス |
| 品質保証 | 適切な品質標準と運用基準の適用を確実に行うために，品質の要求事項と品質管理測定の結果を監査するプロセス |
| 品質管理 | パフォーマンスを査定し，必要な変更を提案するために，品質活動の実行結果を監視し，記録するプロセス |

## 6.3　品質計画

本プロジェクトにおける品質マネジメントの方針，及び考慮すべき点を以下に述べる．

### 6.3.1　基本方針

本プロジェクトの基本方針として表 6-2 で示した 6 特性のうち，本プロジェクトでは以下の特性を重視することにした．

- 機能性

本プロジェクトは経営改革のための IT ソリューションを提供する事が目的である．したがって，「顧客の経営改革に必要な機能が実装されているか」を担保する機能性を重要視する．

- 使用性

導入後に実際にシステムを利用するのは経営陣だけでなく学生コーチも含まれる．したがって，「ユーザにとって理解しやすく，使いやすいかどうか」を担保する使用性を重視する

- 保守性

本プロジェクトでは短期リリースに対応するため，デバッグやテストにかかる時間を減らす必要がある．したがって「要求変更に対応するための修正のしやすさ」を担保する保守性を重視する．

これらの品質特性を担保するために，本プロジェクトでは以下の活動を重視した．

- 顧客との積極的な関わり

本プロジェクトでは，時間の経過とともに変化する顧客のニーズ[9]に対応するために，ミーティングや顧客業務の観察，及び講習会の実施等，積極的に顧客と関わる機会を設けた．顧客と話し合いを重ねる事によって，顧客から潜在的な要求を抽出することを促進し，機能性の副特性である合目的性の向上を目指した．[19]

- 顧客に向けた十分な操作体験と対話しながらのフィードバック

顧客とのミーティングでは，開発途中のソフトウェアの動作を確認してもらうことを重視した．当日までに作成したソフトウェアを顧客に直接触ってもらい，顧客がより使いやすいソフトウェアにするためのフィードバックを出してもらった．これらの活動によって，機能性の副特性である合目的性や，使用性の副特性である理解性，習得性，運用性及び魅力性の向上を目指した．[19]

- 定期的ビルドとテスト自動化

単体テスト，及び統合テストが自動化することにより，ソフトウェアの変更があってもテストに要する労力を少なく抑えることが出来るため，保守性の副特性である試験性の向上に繋がる．[19]

### 6.3.2 考慮した点

本プロジェクトでは，【 4.3.3 開発手法の検討】で述べた通りアジャイル型システム開発を取り入れている．プロジェクト発足当初，筆者はソフトウェア開発データ白書[20]等，一般的なアジャイル型システム開発プロジェクトの基準値を参考にして品質の見積もり・測定を行うつもりでいた．しかし，現状のソフトウェア開発の現場では，アジャイル型システム開発プロジェクトの的確な定量的評価法は確立されておらず[21]，経験則・暗黙知により品質・信頼性を判断し，定量的な評価を実施することなくリリースされているケースが多い[22]という事が判明した．

そのため，本プロジェクトにおいては，イテレーション 0 では品質の見積もりを行わず，イテレーション 1 ではイテレーション 0 の実績値を目標値として，その後のイテレーションでも，これまでの実績値を目標値として見積もりを行うことで，品質マネジメントの PDCA サイクルを回していくことにした．先行研究[23]においても，アジャイル型システム開発でリリースごとに品質評価を行い，数値を洗練していく品質管理を行う手法の有用性は認められている．

## 6.4　品質保証

### 6.4.1　品質マネジメントフローの設定

本プロジェクトで行う品質マネジメントフローについて，図 6-2 のように設定した．フローの手順は以下の通りである．

(1) 1 つのユーザストーリに対して品質保証プロセス（【6.4.2 品質保証プロセスの設定】）を実施する．

(2) 工程品質判定表（【6.4.3 品質分析について】）を用いて品質分析を行う．

(ア) 計画通りの品質だった場合は終了し，次のユーザストーリの開発に着手する．

(イ) 計画値と実績値が乖離していた場合，バグの目標値を見直し再び品質保証プロセスを行う．

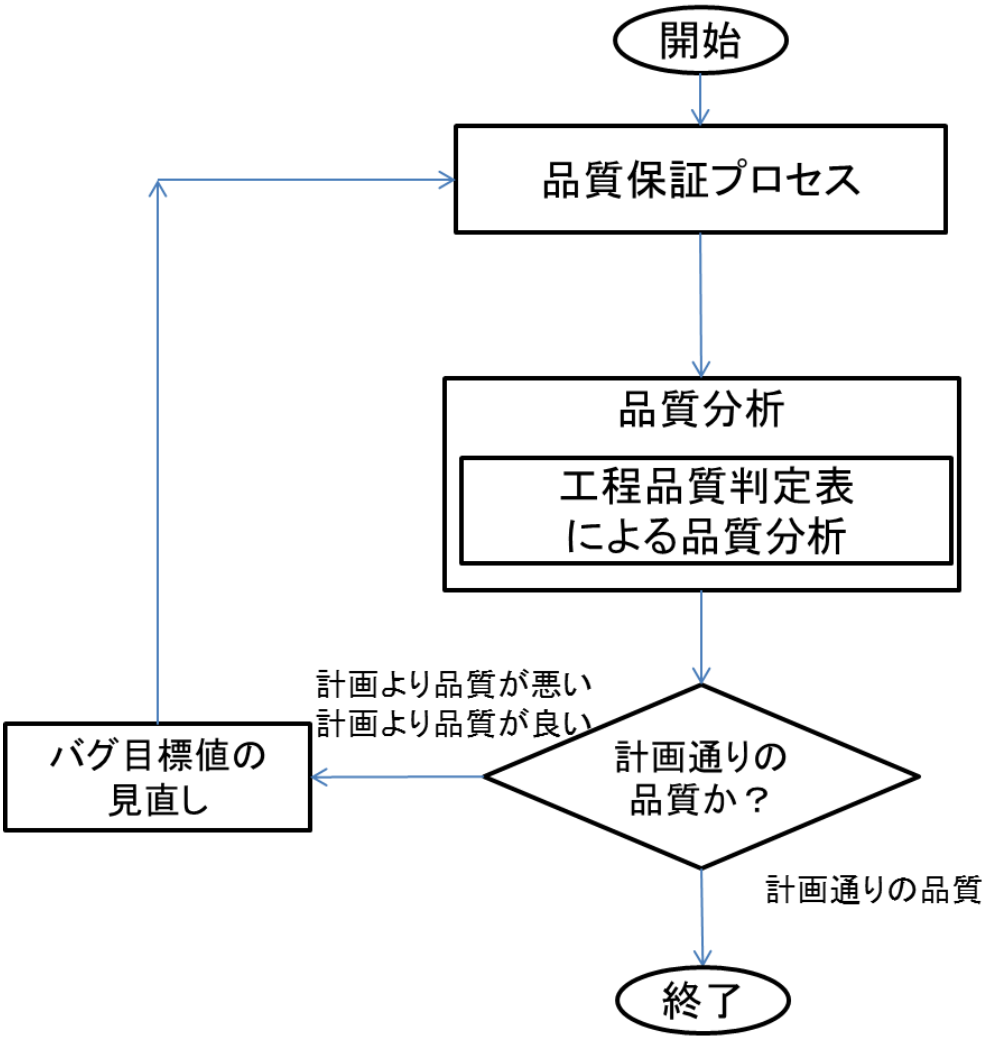


図 6-2　品質マネジメントフロー

6.4.2　　品質保証プロセスの設定

チームで行うレビューや成果物の作成等の手順をまとめた品質保証プロセスを図 6-3 のように設定した.

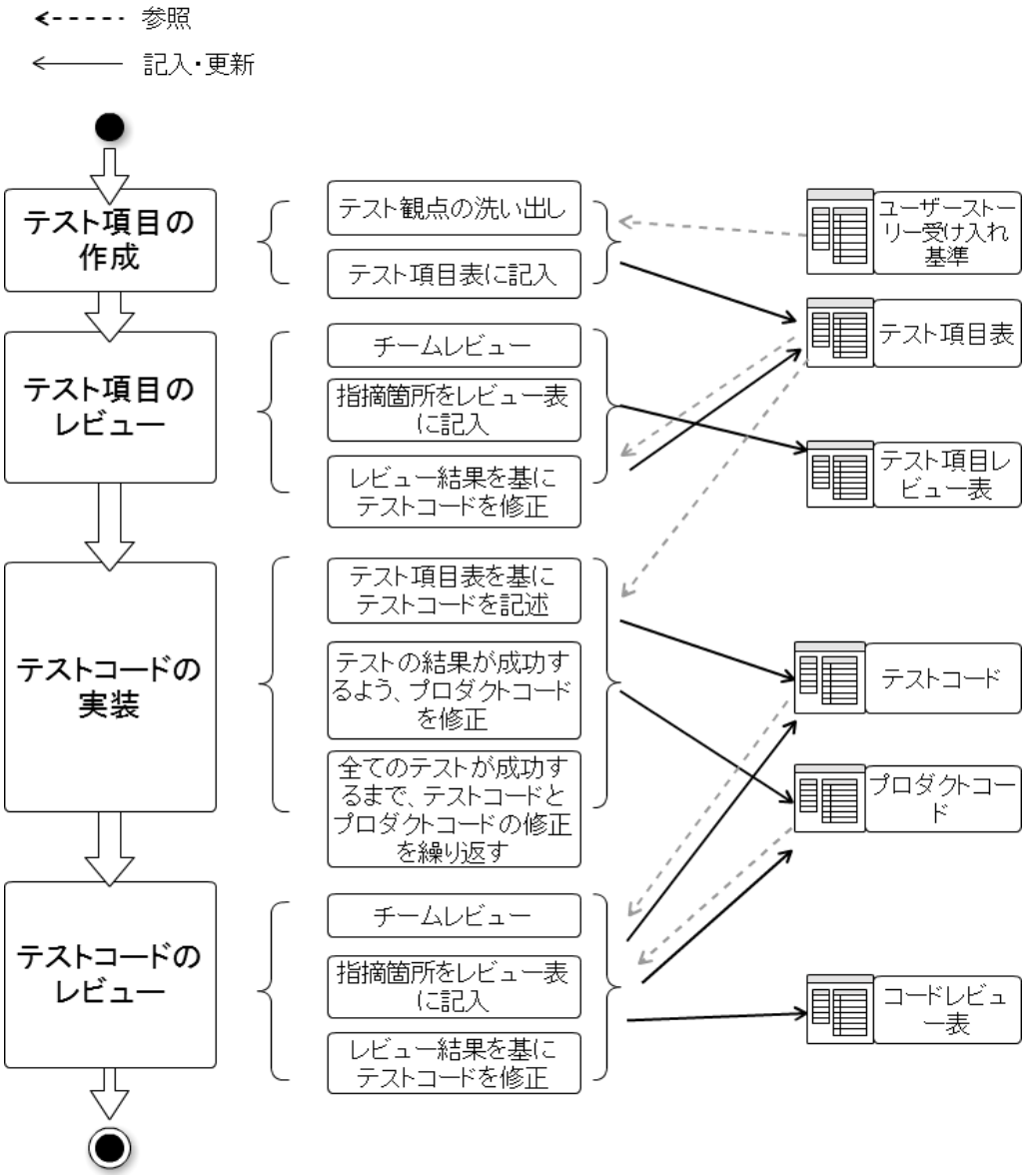


**図 6-3　品質保証プロセス**

それぞれの項目について説明する．

● テスト項目の作成

まず，要件抽出フェーズで作成したユーザストーリ受け入れ基準を参考に，テスト項目表を作成する．作成したテスト項目表の一部抜粋を図 6-4 に示す．どのような観点でテストを行うかを左から大分類・中分類・小分類で分けて記入し，テストを行う事で期待される結果を右側に記入するようにした．

| | | | |
|---|---|---|---|
| ユーザーストーリ | コア9 | | |
| ユーザーストーリ | 「コーチは自分の週カレンダーから代行依頼の申請が | | |
| | | | テストコード網羅チェック |
| 大分類 | 中分類 | 小分類 | 内容 |
| Controller | | | |
| update_coach | 管理者として | 失敗 | 管理者の最新スケジュールページへリダイレクトすること |
| | コーチとして | 過去のコマをパラメータとして渡した場合 | 失敗すること。(Should not be success ) |
| | | 更新前、且つ 自分のクラスが入ってないコマの場合 | 失敗すること。(Should not be success ) |
| | | 更新前、且つ 自分のクラスが入ってるコマの場合 | 選択したレッスンに対応したedit画面をレンダリングすること |
| | | 更新後、且つ 自分のクラスが入ってるコマの場合 | ステータスコードが200で、最新スケジュールのページにリダイレクトすること |
| | | | Flash[:success]に成功時のメッセージが格納されていること |
| | | | コーチの状態が代行申請中になっていること |
| | | | メールが送られること |
| | 略 | メールの中身 | コーチ名、休みまでの日数、代行依頼の申請レッスン(日付、曜日、開始時実施コート)、システムにアクセスするリンク、代行可能のリンクが含むこと |

**図 6-4 テスト項目表**

● テスト項目のレビュー

作成したテスト項目表に対しチームでレビューを行う．レビュー時の指摘は指摘事項，種類，事由をそれぞれ記録し，その後テスト項目作成者が修正を行う．テスト項目レビュー表の一部抜粋を図 6-5 に示す．

レビュー結果記入表

参加者

| | | |
|---|---|---|
| レビュー参加者 | ○有田･井原･永井 | ※レビューする成果物の作成者に○つける |

対象

| | | | |
|---|---|---|---|
| ユーザーストーリー | コア1「利用者はログインIDとパスワードを用いることで、システムへのログインを制御することが出来る」 | レビュー回数 | 1 回目 |

会議体

| | | | | | |
|---|---|---|---|---|---|
| 実施年月日 | 2012/7/30 | 開始・終了時刻 | 14:00 | ～ | 14:20 |

| | | |
|---|---|---|
| レビュー担当者所見 | | □レビュー済<br>□要再レビュー |

| ID | 指摘事項 | 種類 | 事由 *2 | 修正内容、検討結果 | 完了日 |
|---|---|---|---|---|---|
| 1 | 「ログインID」→「メールアドレス１」 | B.用語の統一 | Aエラー | 修正 | 7月30日 |
| 2 | 「管理者権限を持ったユーザー」→「管理者」 | B.用語の統一 | Aエラー | 修正 | 7月30日 |
| 3 | 「それ以外」ではなく「コーチ」と記載した方が良い | B.用語の統一 | Aエラー | 修正 | 7月30日 |

**図 6-5 テスト項目レビュー表**

- テストコードの実装

次に，レビューによって修正したテスト項目表を参考に，テストコードを実装する．テストコードは【4.8.2 テストの方針について】で述べた概要に沿って実装する．

- テストコードのレビュー

作成したテストコードに対して，チームでレビューを行う．ここでは，実際のテストコードがテスト項目表に記した観点に沿っているか，抜けが無いかをチェックする．

### 6.4.3 品質分析について

品質分析は表 6-4 に示す工程品質判定表[22]を用いて行う．工程品質判定表は，予め見積もったテスト項目数や摘出バグ数と実績値を比較することによって，品質を判定しようというものである．判定した結果，バグ目標値を見直す場合は， 数式 6-1， ・・・数式 6-2 を用いて見直す．品質が計画通りであった場合はそのユーザストーリを終了し，次のユーザストーリの開発に着手する．

**表 6-4 工程品質判定表**

| | | テスト項目数 | | |
|---|---|---|---|---|
| | | 実績＜計画-20% | 計画-20%≦実績≦計画+20% | 計画+20%＜実績 |
| 摘出バグ数 | 実績＜計画-20% | 品質を判断する時期ではない<br>⇒レビュー継続 | 品質は計画よりも良い<br>⇒数式 6-1 で見直し | 品質は計画よりも良い<br>⇒数式 6-2 で見直し |
| | 計画-20%≦実績≦計画+20% | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 品質は計画通り | 品質は計画通り |
| | 計画+20%＜実績 | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 計画より品質が悪い<br>⇒数式 6-2 で見直し |

$$\mathbf{NBMP} = \mathbf{OBMP} \times \frac{\frac{BAP}{TAP}}{\frac{BEP}{TEP}}$$ **・・・数式 6-1[22]**

$$\mathbf{NBMP} = \mathbf{OBMP} \times \frac{BAP}{BEP}$$ **・・・数式 6-2[22]**

NBMP＝新バグ目標値，OBMP＝旧バグ目標値，BAP＝バグ実績値

TAP＝テスト項目数実績値，BEP＝バグ数見積値，TEP＝テスト項目数見積値

## 6.5 品質管理

### 6.5.1 バグの定義

本プロジェクトで定めたバグの定義を以下に表す.

**表 6-5 バグの定義**

| | | |
|---|---|---|
| バグの定義 | 設計のバグ | ユーザストーリ受け入れ基準に沿っていないものはバグとする. |
| | | 定義されていない用語を用いた場合バグとする．レビュー時に新たに用語定義された場合も含む. |
| | | 記述が分かりにくく誤解を招く場合はバグとする．誤解を招かないと判断した場合や誤字脱字はバグに含まない |
| | | 1 回のレビュー内で指摘された重複バグはまとめて 1 件として扱う |
| | コーディングのバグ | テスト項目表に沿ってないものはバグとする |
| | | テスト項目表にバグがあった場合は設計のバグとして集計し，コーディングのバグには含めない |
| | | レビュー時にテストコードを通過していない場合はバグとする |

### 6.5.2 目標値の算出

品質を測定するためには，発生すると予測されるバグ数の見積もりを行い，その値と実績値を比較する必要がある.

一般的に，バグの見積もりを行う場合は，過去のプロジェクトのデータやソフトウェア開発データ白書[20]などの値を参考にする．しかし，本プロジェクトが取り入れる反復型開発の場合は，プロジェクト毎に規模やプロジェクトメンバの経験・技量等で数値に大きなばらつきがあり，本プロジェクトが直接参考値として扱う開発のデータを得られなかった．そこで，本プロジェクトではイテレーション 0 で計測した実績値から，1USP（ユーザストーリポイント）毎のバグ発生率を算出し，以降のイテレーションのバグ数の見積りを行う際の基準とする.

イテレーション 0 の実績値を表 6-6 に示す.

**表 6-6　イテレーション 0 の実績値**

| | US 総数 | US 総ポイント数 | テスト項目数 | バグ数 |
|---|---|---|---|---|
| イテレーション0 | 5 | 16 | 81 | 14 |

※US・・・ユーザストーリ

表 6-6 の値からイテレーション 0 のバグ基準値を算出した.
バグ数＝バグ発生確率×規模とすると，イテレーション 0 におけるバグ発生確率は

14（バグ数）＝α（バグ発生確率）×16（US 総ポイント数）

α＝0.875

となり，1 ユーザストーリポイントに対し 0.875 件のバグが発生する事が分かった.
また，1USP に対し作成されるテスト項目数は

81（テスト項目数）÷16（US 総ポイント数）＝β（適正テスト項目数）

β≒5.06

となり，1USP 毎に 5.06 件のテスト項目が作られる事が分かった.
これらの値を当面の目標値としてイテレーション 1 以降の見積もりを行った.

### 6.5.3　見積もりと実績

各イテレーションの見積もり値と実績値の比較結果を表 6-7 に示す.

**表 6-7　イテレーション毎の見積もり値と実績との比較**

| イテレーション | US 総数 | US 総ポイント数 | テスト項目数 | | バグ数 | |
|---|---|---|---|---|---|---|
| | | | 見積もり | 実績 | 見積もり | 実績 |
| イテレーション 0 | 5 | 16 | | 81 | | 14 |
| イテレーション 1 | 6 | 26 | 131.56 | 72 | 22.75 | 6 |
| イテレーション 2 | 6 | 23 | 116.38 | 121 | 20.13 | 20 |
| イテレーション 3 | 7 | 24 | 121.50 | 98 | 21.00 | 10 |
| イテレーション 4 | 4 | 12 | 60.75 | 16 | 10.50 | 5 |

## 6.6　各イテレーションの品質判定

### イテレーション 1 について

**表 6-8　イテレーション 1 の品質分析該当箇所**

| | | テスト項目数 | | |
|---|---|---|---|---|
| | | 実績＜計画-20% | 計画-20%≦実績≦計画+20% | 計画+20%＜実績 |
| 摘出バグ数 | 実績＜計画-20% | 品質を判断する時期ではない<br>⇒レビュー継続 | 品質は計画よりも良い<br>⇒数式 6-1 で見直し | 品質は計画よりも良い<br>⇒数式 6-2 で見直し |
| | 計画-20%≦実績≦計画+20% | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 品質は計画通り | 品質は計画通り |
| | 計画+20%＜実績 | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 計画より品質が悪い<br>⇒数式 6-2 で見直し |

テスト項目数，バグ数ともに見積もり値より実績値が低くなっている．テスト項目数の実績値が低い理由としては，テスト項目表の記述方法に「どのような観点でどのような部分をテストするか」といった一定の基準を設けていなかったことがあげられる．このため，テスト作成者によってテスト項目表の質・量にバラつきが生まれてしまった．また，バグ数の実績値が低い理由としては，もともとのテスト項目が少数であった事が主な理由であると考える．

以上の反省を踏まえ，筆者がテスト項目表のフォーマットを作成し，改善を図った．

イテレーション 2 について

表 6-9　イテレーション 2 の品質分析該当箇所

| | | テスト項目数 | | |
|---|---|---|---|---|
| | | 実績＜計画-20% | 計画-20%≦実績≦計画+20% | 計画+20%＜実績 |
| 摘出バグ数 | 実績＜計画-20% | 品質を判断する時期ではない<br>⇒レビュー継続 | 品質は計画よりも良い<br>⇒数式 6-1 で見直し | 品質は計画よりも良い<br>⇒数式 6-2 で見直し |
| | 計画-20%≦実績≦計画+20% | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 品質は計画通り | 品質は計画通り |
| | 計画+20%＜実績 | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 計画より品質が悪い<br>⇒数式 6-2 で見直し |

イテレーション 1 に比べてテスト項目数，バグ数ともに見積もりの精度が向上している．見積もりの精度が向上した理由として，テスト項目表のフォーマットを改善したこと，毎日作業前に行うデイリーミーティングにおいてレビューの実施を徹底したことがあげられる．

イテレーション 3 について

**表 6-10　イテレーション 3 の品質分析該当箇所**

| | | テスト項目数 | | |
|---|---|---|---|---|
| | | 実績＜計画-20% | 計画-20%≦実績≦計画+20% | 計画+20%＜実績 |
| 摘出バグ数 | 実績＜計画-20% | 品質を判断する時期ではない<br>⇒レビュー継続 | 品質は計画よりも良い<br>⇒数式 6-1 で見直し | 品質は計画よりも良い<br>⇒数式 6-2 で見直し |
| | 計画-20%≦実績≦計画+20% | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 品質は計画通り | 品質は計画通り |
| | 計画+20%＜実績 | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 計画より品質が悪い<br>⇒数式 6-2 で見直し |

テスト項目数は計画通りだが，摘出バグ数が計画より大きく下回ってしまった．この原因として，イテレーション 3 で取り組んだユーザストーリのほとんどが画面描写・UI に関する部分であったため，レビューを行う前に開発者自身がバグに気付き，各自で修正していた事があげられる．

数式 6-1 を用いてバグ数を再度検討した結果，表 6-11 のようになった．適切なバグ抽出数であると考えられる．

**表 6-11　再検討したバグ数見積もり**

| イテレーション | US 総数 | US 総ポイント数 | バグ数 | | |
|---|---|---|---|---|---|
| | | | 見積もり | 新見積もり | 実績 |
| イテレーション 3 | 7 | 24 | 21 | 12 | 10 |

イテレーション 4 について

表 6-12　イテレーション 4 の品質分析該当箇所

| | | テスト項目数 | | |
|---|---|---|---|---|
| | | 実績＜計画-20% | 計画-20%≦実績≦計画+20% | 計画+20%＜実績 |
| 摘出バグ数 | 実績＜計画-20% | 品質を判断する時期ではない<br>⇒レビュー継続 | 品質は計画よりも良い<br>⇒数式 6-1 で見直し | 品質は計画よりも良い<br>⇒数式 6-2 で見直し |
| | 計画-20%≦実績≦計画+20% | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 品質は計画通り | 品質は計画通り |
| | 計画+20%＜実績 | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 計画より品質が悪い<br>⇒数式 6-1 で見直し | 計画より品質が悪い<br>⇒数式 6-2 で見直し |

イテレーション 1 と同様にテスト項目数と摘出バグ数の実績が大きく計画を下回ってしまった．この原因として，ユーザストーリポイントの見積もりが誤っていたと考えられる．イテレーション 4 で着手したユーザストーリは，未調査の技術を用いるものだったため見積もり時には各機能につき 3～5 という高ポイントを付けていた. しかし実際には外部プラグインを用いる事で簡単に実装が出来る事が判明した．そのため大きく計画と実績が乖離してしまったと考えられる.

# 第7章　プロジェクトの総括

## 7.1　考察

### 7.1.1　本プロジェクトの取り組みについて

**良かった点**

1 点目に，当初の目標である「顧客との密なコミュニケーションを取る」の通り，毎回欠かさずに顧客と 19 回のミーティングを敢行できた点，2 点目に，メンバ全員が未経験であったアジャイル型システム開発手法を取り入れ，試行錯誤しながらも期日までにシステムを納品する事が出来た点があげられる．

1 点目については，プロジェクト結成当初からメンバ全員の共通理解として「顧客に使って貰えるシステムを作りたい」という思いがあり，その方針からぶれること無くプロジェクトを進めることが出来たためであると考える．

2 点目については，アジャイル型システム開発のリーダーのような役を特に設けない代わりに，プロジェクトメンバ全員がアジャイル開発に関する書籍[5]を熟読した上で開発フェーズに取り組んだことが出来たためであると考える．これにより，ウォーターフォール型システム開発には無い特徴的な取り組みについても，メンバ間で目的や意思を共有して進めることが出来た．

**反省点**

1点目に実際にシステムの利用者となるT1のコーチやスタッフの方の意見を取り入れる時期が遅く，コーチやスタッフの意見をシステムの機能に取り入れきれなかった点，2 点目に【5.4.1 システムトラブルについて】で起こったトラブルによって評価フェーズ 1 の期間に十分な検証を行えなかった点である．

1 点目については，スタッフの方からは「メールの開封確認機能」が欲しいという声が多く発生した．現状のシステムの場合，代行依頼が発生した際に「代行可能なコーチは代行の立候補をする」「代行不可能なコーチは立候補しない」事で，判別できると考えていた．しかし，スタッフのニーズとしては，「連絡を読んでいないため立候補しない」のか「連絡は読んだが，予定があり立候補しない」のかが判別できないため，確認作業を行う必要があり業務にかかる時間が遅くなってしまう事が分かった．そのため，「メールの開封確認機能」によってコーチが連絡を見たかどうかが分かる機能の要望があった．

2 点目については，本プロジェクトメンバが，システムのエラーやバグに対しての認識が甘かったことが原因であると考える．本プロジェクトが取り入れているアジャイル型システム開発では「バグが無いシステムを，時間をかけて提供する」よりも，「動くシステムを素早く提供する」事を優先する[9]という考え方がある．そのため，システムのバグなどがあった場合は次のイテレーションで解決すれば良いという認識があった．そのため，納品時においても不具合が起これば修正対応すれば問題が無いと考えてしまった事が原因である．

学び

1 点目について，実は「メールの開封確認機能」については一度，経営陣とのミーティングで要件に上がったものの，優先度が低いとして受け入れなかった機能である．もし，要件抽出フェーズ等の早い段階からコーチやスタッフを交えて要求を吸い上げる事が出来ていれば，システムに組み込めていたと考えられる．今後の学びとしては，最終的なシステムの利用者を事前にしっかり想定し，意見を吸い上げる重要性を学んだ.

2 点目について，アジャイルの理念である「バグが無いシステムより動くシステムを素早く提供する」というのは，バグがあってもいいという意味ではなく,「顧客にとって価値のあるシステムを提供するべきである」という意味であることを理解すべきであった．今回の場合，システムが使えなくなる程の重要なバグは，事前に厳重にテストを行い,「システムが止まらないようにすることが顧客にとっての価値」であった．プロジェクトの方針だからという理由でただ盲目的に理念に従うのではなく，目的や用途を考え，実際に用いるかどうかの最終判断は自分で考えて下す必要がある事を学んだ.

### 7.1.2 今後の展望について

まずは，本年度の取り組みを成功させるために，評価フェーズ 2 において，正確な評価の計測に努める．特に評価フェーズ 1 での失敗を活かし，システムに対しては十分なテストと事前デモを重ねること，T1 の従業員に対しては，十分な教育を設けることを徹底する.

また, T1 の課題としては, 正社員を雇い, 重要な業務を任せられる体制を築く必要がある. 特に IT リテラシーが低い従業員が多いため, システムの運用及び従業員への周知を任せられる人材の確保は急務であると考える.

本年度の目標である「内部基盤を達成する」が完了した後は，既存顧客の満足度増加を目指すべく，T1 のブランド力強化戦略にシフトする．現在は立地等や社会的なテニス人気に恵まれていたが，これからは競合他社に打ち勝つべく T1 独自の強みを押し出していかねばならない．T1 としては，これまでの「アットホームさ」を持ち味として，生徒へのアンケートや生徒の保護者との繋がり強化を図り，他スクールにはない親しみやすさ・楽しさを生徒や生徒の保護者へアピールしていくべきであろう.

「既存顧客への満足度増加」及び「T1 のブランド力強化」が完了した後は新規顧客の獲得を目指していく. 生徒のニーズに合わせた多彩なサービスときめ細やかな指導をアピールし, 様々な年代の様々な目的を持った生徒を抱え込める土台を作っていく．初めは T1 の【2.3.1 マインド・マップを用いた顧客の想いとビジョンの共有】で抽出したように幼稚園生から小学生までのジュニアに特化したサービスを取り入れ，ジュニアの生徒本人とその保護者からの支持を得る事から始める.

# 謝辞

筆者の指導教官である田中二郎教授には，筆者が本学へ入学する以前から本コースの内容や受験に関しての助言や応援のメールを頂き，大変お気遣い頂きました．入学後も事あるごとにご指導を頂き，筆者の学業や進路において大きな励みになりました．また，本論文執筆に関して多大なご尽力を頂いたことを心より感謝申し上げます.

本コース常任教員であり，本プロジェクト担当教員である山戸昭三教授には，昨年のプロジェクトに引き続き，プロジェクト活動の円滑化のために時間をかけてご指導していただき本当にお世話になりました．また，公私ともに親身になってお気遣いを頂いたこと，そして本論文執筆にあたり多大なご指導を頂いたことを心より感謝申し上げます.

本プロジェクトの顧客であり，プロジェクトに快く協力をして頂いた T-1 インドアテニススクールの末満裕之様, 末満弘江様には経営資料の提供やヒアリング, レッスン業務の見学, システムの試用, 及び全 19 回にも及ぶミーティングにお付き合いいただき本当にありがとうございました．おかげ様でプロジェクトを完了することが出来ました．心より感謝申し上げます．また，同スクールのスタッフの皆様，コーチの皆様には，講習会，アンケート等，貴重なお時間を本プロジェクトのために割いていただきました．そして，貴重なご意見を伺う事が出来ました．心より感謝申し上げます.

筑波大学大学院「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」協賛企業の皆様，高度情報通信人材育成支援センター（CeFIL）の講師の皆様，関連職員の皆様からは，学びの機会・環境を数多く提供して頂きました．心より感謝申し上げます.

筆者の同級生であり，T-1 インドアテニスクールの現役コーチである松山拓矢君には，多大な時間を本プロジェクトに割いていただきました．ミーティングへの参加や，システムのテストなど，本プロジェクトの 6 人目のメンバとしてプロジェクトを支えてくれました．本当にありがとうございました.

昨年に引き続き同じプロジェクトメンバとして頑張ってきた井原淳平君，白田良太君，杜セイ雨さん，そして本年度よりチームメンバとして加わった永井達也君，皆からは多くの知識や考え方を吸収させて頂き，そしてとても貴重な経験を積む事が出来ました．皆と同じチームで一つのプロジェクトを完了できたことを誇りに思います．本当にありがとうございました.

最後に，様々な面でご支援いただきました家族，友人，大学生活でお世話になった全ての方方々に心より感謝申し上げます.

# 参考文献


[1] 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」, Project Management Institute, 2008

[2] IT コーディネータ協会:「IT コーディネータプロセスガイドライン Ver2.0」，IT 経営研究所，2011

[3] Tony Buzan:「ザ・マインドマップ」，ダイヤモンド社，2005

[4] 浅井麻衣，重田正俊，橋本大輔，浜口弘志，藤井啓詞:「現場の UML モデルベース開発のすべて」，ソーテック社，2006

[5] Jonathan Rasmusson，西村直人（監）:「アジャイルサムライ」，オーム社，2011

[6] Mike Cohn，安井力，角谷新太郎（訳）:「アジャイルな見積もりと計画づくり 価値あるソフトウェアを育てる概念と技法」，マイナ，2009

[7] Mint（経営情報研究会）:「図解でわかるソフトウェア開発のすべて」，日本実業出版社，2006

[8] 津田　義史:「実践 反復型ソフトウェア開発」，オーム社，2012

[9] Kent Beck，“Manifesto for Agile Software Development”,

http://www.agilemanifesto.org

[10] オブジェクトクラブ，“プロジェクトファシリテーション実践編 ふりかえりガイド“

http://objectclub.jp/download/files/pf/RetrospectiveMeetingGuide.pdf

[11] Amazon，“クラウドプラットフォーム活用を支援する amazon ウェブサービス | Amazon の Iaas，Paas 型クラウド（AWS 日本語）”

http://aws.amazon.com/jp

[12] Heroku，“Heroku | Cloud Application Platoform”

http://www.heroku.com/

[13] Mailgun “Mailgun: Programmable Mail Servers”

http://www.mailgun.com/

[14] Stteve Freeman，Nat Pryce，和智右桂（訳），高木正弘（訳）:「実践テスト駆動開発 テストに導かれてオブジェクト指向ソフトウェアを育てる」，翔泳社，2012

[15] Jez Humble，David Farley，和智右桂（訳），高木正弘（訳）:「継続的デリバリー 信頼できるソフトウェアリリースのためのビルド・テスト・デプロイメントの自動化」，アスキー・メディアワークス，2012

[16] 高橋義郎:「使える! バランス・スコアカード」，PHP 研究所，2007

[17] 誉田直美:「ソフトウェア品質会計～NEC の高品質ソフトウェア開発を支える品質保証技術～」，日科技連，2010

[18] 国際標準化機構，“ISO/IEC 25000:2005 Software Engineering -- Software product Quality Requirements and Evaluation （SQuaRE）”， Guide to SquaRE，2005

[19] 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター,“非ウォーターフォール型開発に関する調査”,2010

[20] 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター,“ソフトウェア開発データ白書 2012-2013”, 2012

[21] 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター:「品質予測のススメ IT システム開発における品質予測の実践的アプローチ」，オーム社，2008

[22] 榛葉 悟, 室溝 誠治, 内山 裕文，“アジャイル開発製品のテストに対する品質工学の適用品質工学 19（5）”, 584-591, 品質工学会 2011-10

[23] 足立直，平野健一，疋田久子，山中敦，田中恵美，“アジャイル開発における品質管理の取り組みと評価”, プロジェクトマネジメント学会誌 13（2）, 24-29, 2011-04-15