

統合型要求工学の実証研究に向けて

堀 昭三[†] 中谷多哉子[‡] 片峯恵一[†] 鵜林尚靖[†] 橋本正明[†]

[†]九州工業大学 〒820-8502 福岡県飯塚市川津 680-4

[‡]筑波大学大学院 〒112-0012 東京都文京区大塚 3-29-1

E-mail: [†] hori@ysknet.co.jp [‡] nakatani@gssm.otsuka.tsukuba.ac.jp [†] katamine@ci.kyutech.ac.jp
[†] ubayashi@ai.kyutech.ac.jp [†] hasimoto@ai.kyutech.ac.jp

あらまし 我々は、ソフトウェア開発において、要求の追加や変更するプロセスを開発者側からコントロールする仕組みを明らかにするために、実際の開発プロジェクトにおける要求変更履歴に関するデータを解析中である。本稿では、その結果得られた要求プロセスと設計プロセスを統合した開発プロセスを提案できるのではないかと考えている。

キーワード 統合型要求工学, 実証研究

The Empirical Study of Integrated Requirements Engineering

Shozo HORI[†] Takako NAKATANI[‡] Keiichi Katamine[†]

Naoyasu UBAYASHI[†] and Masaaki HASHIMOTO[†]

[†] Kyushu Institute of Technology 680-4 Kawazu ,Iizuka-shi ,Fukuoka, 820-8502 Japan

[‡] University of Tsukuba , 3-29-1 Otsuka, Bunkyo-ku, Tokyo, 112-0012 Japan

E-mail: [†] hori@ysknet.co.jp [‡] nakatani@gssm.otsuka.tsukuba.ac.jp [†] katamine@ci.kyutech.ac.jp
[†] ubayashi@ai.kyutech.ac.jp [†] hasimoto@ai.kyutech.ac.jp

Abstract We study the mechanism of controlling the process of requirements changes of software development. We analyze the requirements change record of an actual development project. According to our study, the development process which integrated the requirements elicitation process with other processes can be proposed.

Keyword Integrated Requirements Engineering, Empirical Study

1. はじめに

多くの開発プロジェクトは当初計画とおりに行われていないと言われている。その要因の一つに要求の変更や不備が上げられている[1]。しかし、実際の実証プロジェクトでは、開発途中に要求変更が発生することは一般的であり、それによる手戻りを想定して、顧客との調整を行い、システムを構築している。統合型要求獲得、すなわち開発プロセスと要求獲得プロセスとを統合させるプロセスを進めるべきであると言われている。このような研究分野は統合型要求工学と呼ばれている[2]。しかしながら、依然として開発プロセスがウォーターフォール型で定義されていることから分かるように、要求獲得プロセスと設計以降の開発作業とは分離されて管理されている。そのため、開発途中の要求獲得は、管理者にとって、忌々しいものだと思われるのではないだろうか。

我々は、より効率的で且つ妥当なシステムを開発するために、管理プロセスと実際の実証プロジェクトとを一致させる必要があると考えている。そのために、本研究では、あるプロジェクトの実証履歴を調査して、開発の実情を明らかにすることにした。このプロジェクトは、開発を進めながら、発注者からの要求を獲得し続けた事例である。

2. ソフトウェア開発事例

本章では、実際の実証プロジェクトを研究の対象とし、そのプロジェクトの特徴と開発プロセスを紹介する。

2.1. プロジェクト概要

2.1.1. システム概要

本プロジェクトで開発するシステムは、レストラン注文管理システムのサーバシステムである。レストラン

注文管理システムは、接客係が持つ注文受付端末（ハンディターミナル）から入力された注文情報を注文受付システム（オーダーエントリーシステム：以降、OESと呼ぶ）に送信し、注文受付、調理指示、伝票発行、会計精算と一連の機能を管理している。

また、顧客テーブルから顧客手動での注文受付が可能となるようにテーブル端末群（タッチパネル式）からの注文受付が可能である。サーバシステムは、OESとの連携およびテーブル端末とも連動しており、レストラン注文管理システムの中心的な位置づけである。

2.1.2. プロジェクトの特徴

開発プロジェクトの特徴を以下に示す。

(1) 開発の背景

開発プロジェクトが担当するサーバシステムは、数年前に、A社によって開発されたものであった。しかし、現状のサーバシステムの性能向上ために再構築することになった。

(2) 開発規模、工数、開発期間

開発規模は約50Kステップ。開発工数は全体で約40人月。開発期間5ヶ月の予定であったが、実質4.5ヶ月で開発終了した。

(3) プロジェクト関係者

図1. に示すように、開発プロジェクトに係わるステークホルダは以下の通りであった。OES開発メーカーT社、N社などが関わり、そこにテーブル端末ソフト開発会社としてS社、サーバ開発会社としてY社、発注元B社が全体システムの取りまとめを行った。本研究の調査は、Y社の開発記録を元に行った。

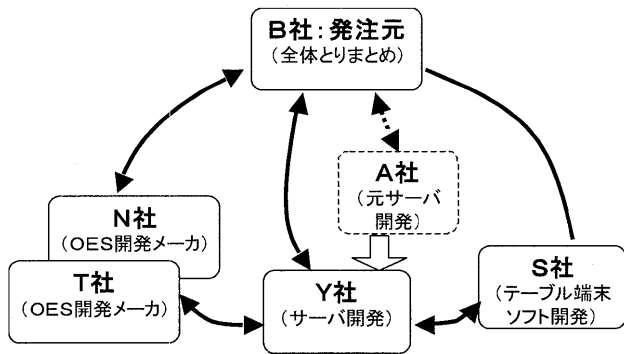


図1. プロジェクト関係者

3. 要求変更に関する調査方針と概要

3.1. 調査方針

要求の変更や追加のプロセスは、管理プロセス関連の資料であるQ&A管理表、課題管理表、および開発プロセス関連資料として要求仕様提案書と、さらに品質関連の資料である障害一覧表、故障票、要望書一覧

から、調査を行った。その結果を図2. にまとめた。

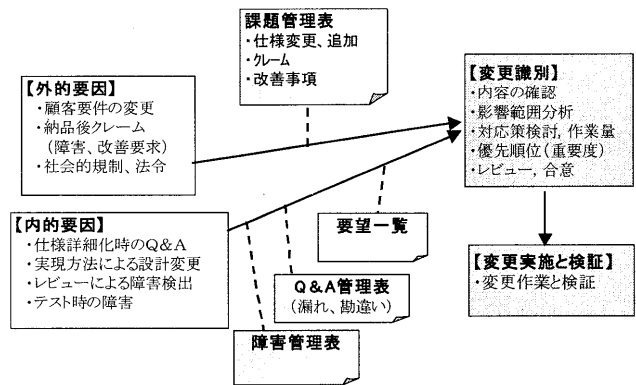


図2. 要求変更の要因と変更対応

図2. において、外的要因とは、Y社以外を発生場所とする要求を表す。また、内的要因とはY社内部を発生場所とする要求を表す。これは、設計を改善する必要に気づくことによって、仕様を変更する場合は該当する。一つの要求変更がいずれを原因とするかは、要望一覧、Q&A管理表などを参照することによって識別することができた。

実際に、発生した要求変更に対して開発に取り込むか否かは、組織間の合意、実現可能性、変更にかかるコスト、他への影響度、緊急度、優先順位付けなどを調査して決定される。これらの各要求変更に対する管理情報も、各文書に記録されていた。したがって、本研究では、これらの文書を参照し、要求変更の頻度、および開発プロセスと要求変更のタイミングとの関係を調査することにした。ただし、本稿では、調査プロジェクトが統合型要求獲得プロセスを実行することによって、どのようなプロジェクト管理上のリスクが解消されたかについて報告を行う。

3.2. プロジェクトのリスクとそれへの対応

本章では、プロジェクト開始時に予想されたリスク、それへの対処、および効果について、Y社の開発記録、仕様を調査した結果をまとめる。

リスク1：ステークホルダ間の意思疎通低下による開発効率の悪化

対処)発注元を含めたステークホルダの責任者全員参加による毎週の進捗会議、技術会議の実施や情報共有となるインフラ整備をおこなった。

効果)上記の対処を実施することで、ステークホルダ間の意思疎通が図れ、壁を取り除くことができ、協力し合う雰囲気をつくることができた。また、問題解決が忘れ去られることを防ぎ、効率的な進め方ができた。

リスク2：発注元およびA社の協力が得られない可能

性によるシステム再構築遅れ

対処) 要求を継続的に獲得するためには、発注元の協力は不可欠である。そのために、契約段階で、発注元への全面的な協力要請とA社への協力支援を発注元経由で依頼した。さらに、開発効率をアップするために、以下の成果物を開発の進捗にあわせて作成し、関係者で共有した。

- ・ 課題管理表：残作業，要求変更履歴，やるべき作業，期限，優先順位，担当を含む
- ・ Q & A 管理表：要求機能に関する質問事項，問合せ先，期限，優先順位，回答内容を含む
- ・ ペンディング事項：次の作業を実施する人への送り事項
- ・ 要求仕様提案書：発注元への要求に対する提案確認内容
- ・ 打合せ議事録：進捗会議や技術会議にて決定された事項，担当，期限含む
- ・ レビューによる障害一覧：障害内容，対処方法，担当，優先順位，期限，承認者含む

効果) 図3. の①に示されているように、2006年2月から6月にかけて、開発スケジュール中の要件定義での要件確認、システム設計での「Q & A」や設計書レビュー、システムテストや運用テストでの妥当性検証での全面的な協力が得られていた。

リスク3：経験の浅さによる品質低下

対処) 2006年2月に、要求獲得に先駆けて、システム設計工程で、DB設計、インタフェース設計を行う上で、将来的に、システム拡張、変更に対応できる設計にするために、事前に、業務サービス内容や要件ヒヤリングを実施した。

効果) これによって、Y社の開発者にも、どのような要件が定義されることになるかといった、システム開発概要を把握することができた。

リスク4：開発のボトルネックによる納期遅れ

図3. に開発プロセスの計画と実績状況を示す。図の□は計画を示し、◀▶は実績を示す。

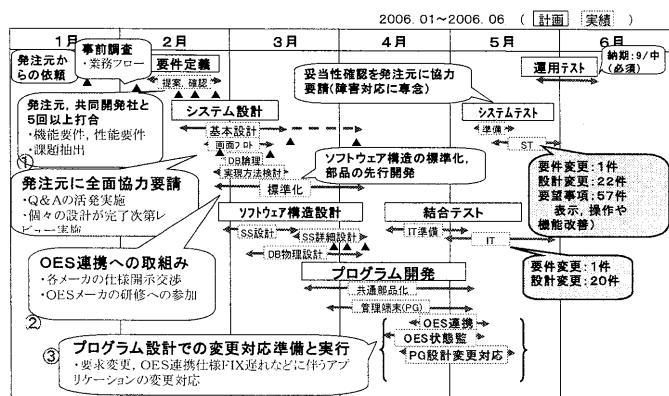


図3. 開発プロセスの計画と実績

この図3. の②を見てもわかるように、OES連携がボトルネックになってしまっている。開発スタート時からボトルネックとなりそうなものを識別されていた。対処) 特に、本システムでは、OESとのインタフェースの早期開示だった。このボトルネックが解消されない場合、全体スケジュール遅れのような問題が発生することが予想されていた。

そのために、関係者は以下の対処を行っていた。週1回の技術会議と進捗会議を行った。この会議には、図1. に示したすべてのステークホルダが加わっていた。

効果) これはOES機器開発メーカの非協力が原因でスケジュール実績が伸び伸びになって、システム設計でのOES連携の要求獲得は不十分な状態であった。

リスク5：OES機器メーカの非協力による開発遅れへの挽回対応策

対処) 開発のスタート段階から、リスクとして上げていたが、各社OES開発メーカからの協力関係が上手く行かず、対応策として以下のことを実施した。OES開発メーカのOES製品導入研修への参加によるインタフェース仕様の事前把握と、プログラム開発段階でOES実機調査と開発へのフィードバックを行った。(図3. の②)

協力が得られない場合を想定した対処：プログラム設計での要求変更へ対処するための準備と実行

開発当初から、要件定義、システム設計などの上流工程で仕様は確定せず、途中で要件変更や追加が発生すると想定し、確定したところから進める方針であった。手戻りが発生することを最初から想定しリスクとして許容し、最小限にするように努めていた。

つまり、開発途中の要件変更や追加、設計ミスによる変更、OES連携仕様確定遅れによるアプリケーションの変更対応をプログラム開発段階で吸収するために、変更対応プロセスを開発スケジュールに新たに盛り込んだ。発注元とは合意し、実行に移した。(図3. の③)

効果) 2段階への対応で、OES連携を乗り切ることができたが、プログラム開発、およびテスト工程での手戻りは多大になったが、最終納期と品質は確保できた。

4. 考察

以上の観察によって、以下の知見を得た。

4.1. 情報共有を実現した効果

情報共有を実現したことによって、開発プロジェクトへの問題や要求獲得の意識向上に繋がる。また、残作業が忘れ去られることなく、問題解決をスムーズに

且つ自律的に解決するプロジェクトチームになる。

4.2.できあがったもの順に納品効果（逐次納品）

できあがったものから逐次納品しては要求獲得，仕様化，レビュー，合意，開発（ウォーターフォール，V字）納品というプロセスにて契約，合意，顧客の協力を得た。

図3の開発プロセスのシステム設計で，「できあがったものから提出」のプロセスは通常の客では問題になる。しかし，実際には問題になっていない。何故ならば，今回の開発の課題は短納期でサーバシステムを再構築することであり，発注元，共同開発各社が協力し合い，如何に早く，既存システムを把握し，品質と納期を守るか必須であった。不明点は申し送り事項やペンディング事項といった情報を添付して発注元の担当者にレビューをかけて貰う進め方を取っており，問題解決のスピードアップが図れた。そういうルール，契約を交わし，合意しておけば，有効である。

4.3.文書による記録効果

開発，レビュー，仕様変更の履歴を文書で管理することによって，要求確認，及び回答，ペンディング事項などの検討項目が明確に文書として残り，共通認識と合意が得られたことである。また，前工程の要件定義段階での残課題や，システム設計での新たな要求や変更を始めとした以下の項目が課題管理表にて整理されていた。

- ・ 残課題
- ・ 要求変更履歴
- ・ 打合せの決定事項

課題管理表にかかれていた内容は，ステークホルダ間の共通認識ややるべきことであり，これが記録として残されていたことによって，やるべきことを取りこぼさないように管理ができ，有効であったと判断する。

つまり，開発プロセスに管理プロセスを取り込んだコンカレントな開発が現実的であると考える。

(4)ボトルネックへの対処

今回の開発でボトルネックになったのはやはりOESインタフェース仕様確定遅れであった。開発スタート時点から協力を得られないことは想定され，協力要請やOES機器導入研修への参加やOES実機検証による通信仕様の把握など，本来やらなくても良い作業に振り回されたのは事実である。プログラム開発工程にてその不明確な部分を吸収する計画は有効であったが，結合テスト，システムテスト工程ではOES連携に関連する処理で障害がかなり発生していた。OES連携がシステム設計の段階で仕様開示が出来ていれば，もう少しじっくり仕様検討ができたかと判断する。また，OES機器メーカーが複数社あり，それぞれインタフェースも違えば，扱う商品種類，コードが違うと

ころがあった。そのため，サーバシステムの商品マスタ登録，処理プログラムへの影響がでた。

整理すると，

1)ボトルネックにならないためにどうしたか。

リスクとして捉え，事前の協力要請を実施する。また，2手3手と先を考えた策を検討しておき，実施していくこと。

2)ボトルネックになってしまったらどうなるか。

要求獲得の遅れにより，開発遅れや品質不良に繋がる。

3)やるべきことをやらないと駄目

ムダかもしれないが，やるべきことはきっちりやって，最大限の努力を惜しまないこと。

5.まとめと今後の課題

本稿では，実際のソフトウェア開発プロジェクトの調査結果に基づき，プロジェクトが抱えていたリスクを回避するために，統合型要求獲得，すなわち開発プロセスと要求獲得プロセスとを統合させるプロセスを選択し，プロジェクトを成功させていたことを示した。

しかしながら，今回の執筆では，一つの実例プロジェクトを分析したのみであり，管理プロセスと開発プロセスの関連を全体に渡ってさらに深い分析が必要であると感じている。

今後は，統合型要求獲得プロセスがプロジェクトに与える効果，およびその管理プロセスを提案するために，他の実例プロジェクトに対しても分析を行う。

文 献

- [1] The Standish Group CHAOS Report '1995
- [2] IEEE SOFTWARE Published by the IEEE Computer Society, pages 16, January/February 2005 IEEE SOFTWARE