

ソフトウェア開発の工程管理に関する研究

日高 智充 (沼田 一道 助教授)

1 はじめに

コンピュータを使ったシステムを多くの企業などが通常業務に導入を始めてから数十年が経過している。その間にハードウェアはメインフレームと言われる大型汎用機からワークステーションとして位置づけられた unix 機, さらに個人ひとりひとりが常時使用できるパソコンへと変遷してきた。それとともにネットワークもシステム専用のものから, 誰もが利用できるインターネットへと発展し, 現在では金融や物品販売業などを筆頭に, 企業と消費者が直接取引を行えるシステムが広く一般社会へ浸透している。

一方, それらを機能させるために必要なソフトウェアの開発はどのように変わってきたのか。実現したい機能をコンピュータが理解(実行)できる水準に分解し, それらをひとつひとつ記述してシステムを構築してゆくというソフトウェア開発の本質はこの間何も変わっていないといえる。開発手法, 管理手法の標準化が進められソフトウェア開発工程に適用されるようになってきているが [1], 進捗遅延, 工数超過の発生, 機能齟齬, バグ残存でのカットオーバーなどが多くのソフトウェア開発の現場で見受けられるのが現状である。これらの問題を実態に即して把握し, 問題解決につながる管理方法をソフトウェア開発工程に即して検討する。

2 研究目的

システムの開発・管理方法やシステム開発において考慮すべき品質・性能等に関しては多くの研究, 提言が存在する。しかし, それらの有効性に関して客観的立場から検証したものは極めて少ない。また, プロセスの管理やプロダクトの構築の面から提言されてきた手法, 理論 [2] はプロセス, もしくは, ソフトウェアそのものを対象としたもので, それを担う人についてはあまり関心が向けられていないように見える。システム開発が成功するかどうかは「それを実施する人の力量でほぼ決まる」というのは経験的に広く認識された事実でもある。本研究では, システム開発を担当する人に焦点をあてたプロセス管理の問題点を議論し, 工程管理において各開発実施者のもつ情報を把握するための方法を提案する。

3 ソフトウェア開発における問題について

ソフトウェア開発工程上で「進捗の遅れ」, 「工数の増大」, 「品質不良」につながる問題として以下が上げられる [3]。

3.1 作業時間不足, 無理な日程

企画段階では予算, 期間は確定しているが作成すべきソフトウェアは何も定義されていない。基本から詳細設計にかけて作りこむべきソフトウェアは明確になってくるが, その前段階で決められる日程に特に根拠はない。また, 現在の経済状況では常にスピード, 低コストが求められるため, それぞれ要求はきびしくなる。根拠のない数字でも一旦これらを基に見積りが行われ開発計画が策定されると, その計画が絶対的なものとなる。

3.2 仕様の不備

仕様の確定は、顧客と開発業者がそれぞれの得意とする知識、情報を持ち寄って目的を達成するために必要となる機能の要求を摺り合せながら決定していくが、あらかじめ導入後の事務処理を体験することなく限られた時間ですべての要求を明示することはできない。また、確定した仕様についても要求を詳細化、実装していくなかで新たな設計項目が発生してくる。新たに発生する設計項目のほとんどは要求された機能を実現するために必要なものであるが、発生自体が開発実施者に依存するため見付けられずに通過してしまうことも多い。発生した設計項目も、そのすべてを顧客と開発業者間で合意・確認していくことは難しく、すべてが仕様書にも記述されるわけではない。

3.3 非快適な作業環境, 作業者の状態

仕様調整に伴ない工数が増大し、これらを消化するために再日程化、要員追加や割当変更が発生する。この場合、増大した工数を消化するために時間的余裕をなくす方向で変更を行なうことになる。具体的には、残業、休日出勤を前提とした日程、作業効率のわるいところからよいところへの割当変更などの対処が実施される。程度にもよるが、いずれも各担当者を消耗させ、コンディション、モチベーションの低下をもたらす。また、仕様調整は避けられないものであるが、再スケジュールによる作業へのしわ寄せを出来るだけ明示的に扱い、作業の質の低下を防ぐ手立てが必要である。

4 現状の対策と不十分な点

ソフトウェア開発は図1のように進められる [4]。

開発作業の多くは「現状を調べる、仕様を理解する、新たに組み立てる」というようなことを頭の中で行なうことに費やされる。その結果は、仕様書、設計書、ソースコードに成果物として記述することで他の人に伝えたり、コンピュータに読ませたりする形に表現していく。前節で指摘した事項が現実問題として表面化してくるのはテスト工程が開始されてからで、上流工程では、それらは潜在している。

こうした開発作業を計画通り進めていくために、筆者が管理者として通常行なっている対策は以下の通りである。

成果物の論理の整合を確認していく方法として、レビューを実施している。参加者はレビュー対象物の作成者と他の開発メンバである。開発計画または画面、帳票など外部インターフェースに係るものがレビュー対象となる場合はエンドユーザーも参加する。レビューで決定した事項や未解決の問題は、結論、検討の経緯を議事録に記載し結論が出なかった問題は懸案事項とし記録、管理している。

担当者の作業状況、進捗、これに係わる問題を把握するために各担当者から週報の提出と各成果物毎の予定実績管理票の記入とこれを確認する場として週1回の進捗会議を開催している。集めた情報を積み上げてプロジェクトの進捗状況を把握することになる。

このような形で進めても前節で述べた問題は発生する。各担当者が「問題を提議・検討する場はレ

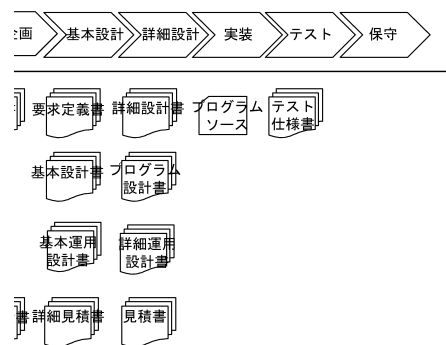


図 1: 現状

ビューと進捗会議である」と捉えてしまうため、ここに多くの問題が集中してしまう。その場で解決できないものが先送りになってしまい、発生から提議、解決までに日数を要してしまう。また、問題の内容も口頭でのやりとりが多くなるため、抽象的でポイントがずれやすく議論の中で有耶無耶となってしまうがちで問題を埋もれさせている。結果として、日程の余裕がなくなり成果物の作成に影響する。これが残業の増加につながり作業環境を悪化させていくことになる。3~5人位の少人数のプロジェクトひとつならば問題が発生しても開発工程毎のマイルストーン迄の解決を計ることで工程全体に影響しないように対応することは可能であるが、10人を越えるような場合は対象の規模が大きく、レビュー、進捗会議に取られる時間が増加することもあり無駄に工数を消化してしまうことになる。

5 改良案

5.1 着眼点

問題が潜り込んだ工程から表面化した工程の間が空くほどその解決には時間を要することになる。問題を早期に摘出し解決までの日数を短縮することができれば開発工程全体のリスク要因を減らすことになるといえる。

このために各担当者が抱える問題を管理者へ報告するための補助ツールを工程管理に導入し、管理者が迅速に情報を把握し適切な対策を講じることにつなげる。但し、これが負荷となり報告が滞ることがないように報告の簡易化を工夫する。

5.2 補助ソフトウェアの設計

5.2.1 目的

補助ツールを使って各担当者から管理者へ報告を随時(日次報告を原則とする)行なえるようにし発生に近い時点で問題を把握する。

5.2.2 概要

問題の指摘、原因、対策についてその内容が明確に報告できる定型フォームとする。報告はデータベースに登録し電子媒体として管理する。

5.2.3 報告フォーム設計

フォームは図2を参照。問題、その理由、対策をそれぞれ対象、内容を明確にできるように項目を分割した。また、計数を取得するための工夫として各項目に対し内容を分類するためのコードを設定できるようにした。

6 考察

ソフトウェア開発の工程管理に前節で述べた問題報告のしくみを追加することによって管理者にとって次のことが期待される。

- 報告の機会が増えることで各担当が抱える問題を毎日確認できる。
- 問題、理由、対策の内容を分類コード化することで数値として管理しやすくなる。

報告番号	プロジェクトID	年	報告者ID	連番	枝番	分類
報告日	報告者名					⑭
問題	気付いた日	②				⑭
	何が	③				
	どうなのか	④				
理由	どうして	⑤				
対策						
1	対策を講じた日	⑥				
	何を	⑦				
	どのように	⑧				
	実施した日	⑨				
	何を	⑩				
	どのように	⑪				
	どうなった	⑫				
2	…繰り返し	⑬				

(a) イメージ

(b) 項目説明

図 2: 報告フォーム

報告が増えるということで、無効な提議も多くなるという可能性はある。しかし、最も避けるべきことは情報が上がらないことであり、間口は広くしなければならない。それに、問題提議としては無効であっても管理者が各担当者にきめ細かく対応していくための基礎資料として役立つことができる。

担当者にとっては、問題を抱え込むことが少なくなり解決までの時間が短縮される。それが、生産性の向上により結果をもたらす作業時間のゆとりにつながる可能性がある。また、報告するさいに「何が、何を」という対象を明確にする必要があるため、これまで漠然と感じていた問題についても、それを具体的に捉える契機となる。

7 まとめ

本研究では、ソフトウェア開発の工程管理の改善を検討し、問題報告のための補助ソフトウェアの導入を提案した。管理手法はいろいろ存在しその定義通りに実施可能であるならば、納期遅れとなるような進捗遅延もシステム障害につながるような不良も少なくなるのかもしれない。しかし、実施していくための現場に馴染む手段は試行錯誤の状態である。本研究での提案は工程を運営していく手段のうちのひとつである。補助ソフトウェア導入の目的である問題発生から解決までの日数短縮については、その実装、試用ができていないため検証はできないが、各担当者の問題を積極的に工程管理に活かす上で有効になると考える。

参考文献

- [1] 松本啓之亮, ソフトウェア工学, 森北出版株式会社, 東京, 2005.
- [2] 玉井哲雄, ソフトウェア工学の基礎, 株式会社岩波書店, 東京, 2004.
- [3] ロバート・L・グラス/山浦恒央訳, ソフトウェア開発 55 の真実と 10 のウソ, 日経 BP 社, 東京, 2004.
- [4] 梅田弘之, 実践! プロジェクト管理入門, 株式会社翔泳社, 2003.