

2L-7

ソフトウェア開発における
コミュニケーション不良と品質の関係
—— コミュニケーション不良に起因するトラブルの分析 ——

井上敦子 福田由紀雄

(株式会社東芝 システム・ソフトウェア技術研究所)

1. 位置づけ

今回の分析は、IMAP(*1)の推進する『ソフトウェア開発の広域化』の一要素である『遠隔地における意思伝達・確認支援技術』を確立するためのものである。これまで開発の作業場所に着目し、『開発担当者・管理者の意識』や『コミュニケーション形態の変化』などについて調査/実験を行い、作業場所の分離を行った場合の問題点を把握した<1>。その結果、ソフトウェア開発におけるコミュニケーション不良は、ソフトウェアの品質に大きな影響を与えることがわかった。そこで今回は、コミュニケーション全般に視野を広げ、その問題点を把握、分析し、今後の指針を示す。

なお、ここで言うコミュニケーションとは、上流工程(ソフトウェア開発における仕様作成までの工程)担当と下流工程(設計からの工程)担当の間の、開発に必要な技術情報の伝達に限定する。

2. 調査概要

2.1 目的

システムのトラブルのうち、ソフトウェア開発におけるコミュニケーション不良が原因のものについて分析を行い、傾向を把握する。

2.2 対象

今回は、新規作成システムの開発プロジェクトと、システムの初版の改良の開発プロジェクト各1件ずつを調査の対象とした。

2.3 分析データ

今回の調査は、各プロジェクトで残されたトラブル記録に着目した。またそれをもとに以下の2項目について、担当者にヒアリングを行った。

(1) トラブルの起きた原因

トラブルの起きた原因を、上流工程担当と下流工程担当のそれぞれの立場から判断した。

(2) コミュニケーション不良の分析

(1)でコミュニケーション不良、もしくはその可能性がある、と判断された場合については、次のような点を調査した。まずコミュニケーションを行ったかどうかについて分類し、行わなかった場合には、コミュニケーションの

The relations between human communication
in software development and software quality.

Atsuko INOUE Yukio FUKUDA

TOSHIBA CORPORATION

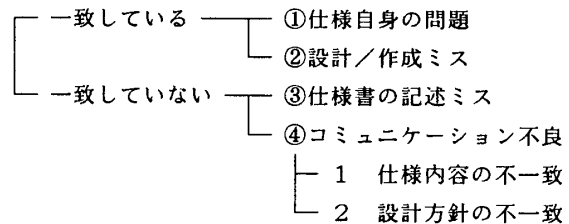
必要性を感じていたかどうかを調べ、また行った場合には、どの点が悪かったかなどについて調べた。

3. トラブル原因

3.1 トラブル原因による分類方法

今回はトラブルの分類基準として、『上流工程担当と下流工程担当の仕様理解の一致の有無』を採用した。図1のようにトラブル原因には、上流工程担当が考えている仕様内容(仕様書の内容ではない)と、下流工程担当の知っているあるいは理解している仕様内容が、一致しているものとそうでないものがある。

図1 トラブルの分類



一致しているトラブルは、上流工程担当の考えた仕様内容自体が、間違っているあるいは客観的にみて不適当である場合(①)、下流工程担当が自分の知っている(理解している)仕様内容を十分実現できなかった場合(②)が挙げられる。

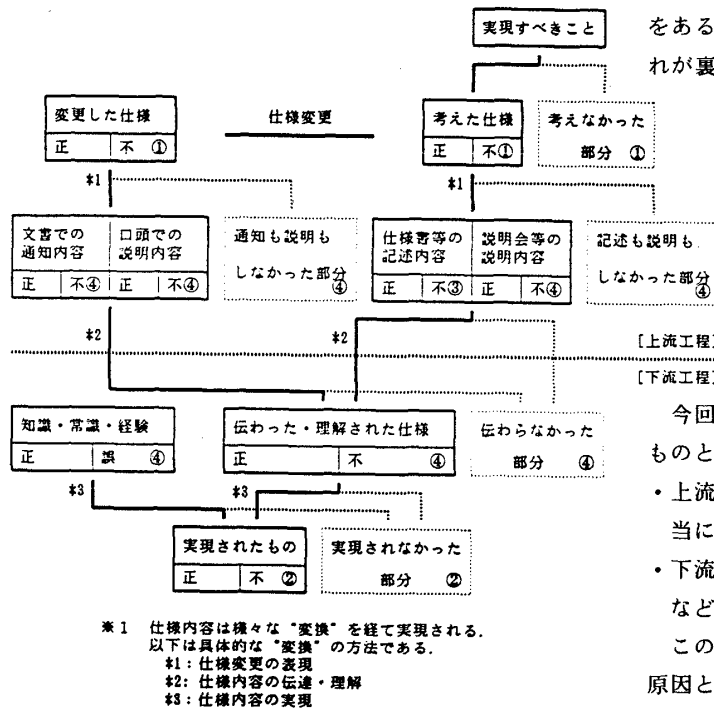
一方一致していない場合は、上流工程担当が、考えた仕様内容と異なった内容を仕様書に記述した場合(③)、また上流工程担当の考えた仕様内容が、下流工程担当に正しく漏れなく伝わっていない場合(④-1)や上流工程担当の設計思想・センス・価値観などが下流工程担当と同じではない場合(④-2)に分けられる。④はどちらも意思の疎通が悪かったために発生したトラブルで、その違いは一致していない部分が“仕様内容”か、“設計常識”かという点である。だが、トラブルの原因となった“一致していない部分”が、そのどちらに該当するのか、はっきり区別するのは難しい。

今回の分類は従来とは異なり、④-1だけでなく④-2も含めてコミュニケーション不良とする。

3.2 仕様内容の流れの中のトラブル原因

図2はソフトウェア開発を上流工程と下流工程に分けて行った場合の、仕様内容伝達の流れである。この中で各トラブル原因がどの時点で起こるのかを示した。

図2 仕様内容伝達の流れ



などで補うか、下流工程担当に任せる場合も多い。これは、下流工程担当の設計の基本方針・常識・技術力などがある程度信頼しているからなのだが、今回の事例はこれが裏目にでたものである。

今回注目すべき点は、下流工程担当のミスの内容が全員同じだったことである。つまり逆に考えると開発の初期段階で基本方針を徹底すれば、これらのトラブルは完全に予防できる、と言える。設計方針など仕様書に明示されない事項は、上流工程担当で徹底しない限り、下流工程の設計/作成に反映させることは難しい、と[上流工程]考えた方がよい。

今回の事例は、以下の2点が不明確なために起こったものと考えられる。

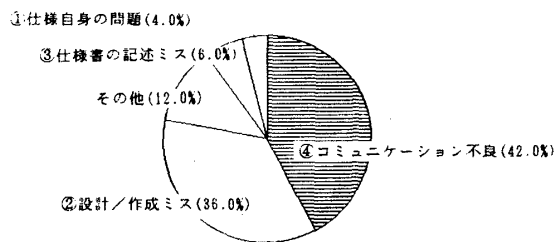
- 上流工程担当は、どこまで仕様内容として下流工程担当に伝えるべきか
- 下流工程担当は、どこから自分の設計方針、設計常識などで判断するべきか

この2点の境目付近で食い違いが発生し、トラブルの原因となる。これらを明らかにしない限り、食い違いを完全になくすことはできない。

4. データ集計結果

まず、トラブル記録を前章で述べた方法で分類した。図3はその結果である。全体的にみると、トラブルの原因は「②設計/作成ミス」と「④コミュニケーション不良」が、大きな位置を占めている。

図3 原因別のトラブル比率



5. 集計結果の考察

今回は、仕様書に記述していない事項あるいは説明を行っていない部分に関して、下流工程担当が実現した仕様内容が、上流工程担当の考えていたものと食い違っている、といった事例が目立った。

例えば、[1] 上流工程担当の考えていた画面の応答時間より長くなった、[2] ツールの応答時間が長くなるのに、それに対するメッセージ（例えば「しばらくお待ちください」）を出力していない、などである。

設計の基本方針あるいは設計常識など、仕様書に記述しないが満たさなければならない事項は、一般的に説明

6. まとめと今後の課題

ソフトウェアのトラブルの原因となるコミュニケーション不良は、当初考えていた「どのように伝達すればよいのか」よりも、「何を伝達すればよいのか」についての問題が目立った。従って、上流工程担当は下流工程担当に何を伝達すべきかを明らかにしない限り、コミュニケーション不良は減少しないと考えられる。しかし、伝達すべきか否かという判断基準は明確にすることが難しい。次回は、伝達必須項目の判断基準を確立するために、データ収集を継続して行く。より多くの（件数面・種類面）プロジェクトからトラブルデータを収集することで、コミュニケーション不良によるトラブルの傾向をつかみ、工程を分担し開発を行うプロジェクトにありがちな“落とし穴”をみつけることができると考える。

また今後の実験計画として、これまでの作業場所の違いによるコミュニケーション不良の傾向だけでなく、ソフトウェア開発における様々な条件（開発の難易度・規模、メンバーの技術力・経験など）の違いによる傾向を把握する。

※I MAP: Integrated software Management and Production support system

<参考文献>

<1> 福田・井上他：「作業場所を分離したソフト開発実験」情処学会第36回全国大会講演論文集 p1115