

学生のソフトウェア開発実習における問題共有方法の考察

栗原 紘樹[†] 伊藤 邦彦[†] 澤島 義人[†] 清水 誠[†] 宮原 佑也[†] 小形 真平[†] 松浦 佐江子[†]芝浦工業大学大学院 電気電子情報工学専攻[‡]芝浦工業大学 システム工学部 電子情報システム学科[†]

1. はじめに

ソフトウェア開発において問題の共有は、意思の統一や同じミスを繰り返すことを防ぐ上で重要である。

本学では、グループワーク支援システムを用いたソフトウェア開発実習を実施している[1]。本実習では、週に1回以上の支援システムからの作業報告が義務付けられており、その報告時に、作業で出た問題を問題点項目に記録している。しかし、2006年度のログを見ると、半数近くの問題が対処されていない。その理由の一つが、現システムの問題共有方法にあると考えた。

本研究では、問題点及びその解決案を分析することにより、学生によるソフトウェア開発において発生する問題の実態を明らかにし、問題のタイプに応じた問題共有方法を考察する。問題のタイプに応じた問題共有方法を提供することで、問題への対応・対処がしやすくなり、それが問題の解決に繋がると考える。

2. 本実習における問題の共有

2. 1. 2006年度における問題共有方法

2006年度における問題の共有に関しては特に明確な機能はなく、作業報告書を直接確認するか、作業項目を一覧できる機能での確認が可能であった。しかし、経験上、作業報告書に書かれた問題が議論されることはほとんどなかった。

そこで、2006年度のログから、作業報告書の「問題点」項目に書かれた内容が次の話し合いの場で議論もしくは解決された記録があるかを集計した。授業が1週間ごとなので、「次の話し合いの場」までの期間を1週間以内とし、作業報告書・掲示板・議事録・インスペクション機能のいずれかに記録があるか否かで判断した。

以下はその結果である。半数近くの問題が未議論・未解決のままになっているということが確認できる。

表 1 グループ別問題・未議論・未解決数

グループ	A1	A2	A3	A4	B1	B2	C1
問題数	48	121	71	69	60	37	95
未議論	22	57	27	31	21	19	41
未解決							

Problem Awareness Support for Software Development Experiments

†Hiroki Kurihara †Kunihiko Ito †Yoshito Sawahata

†Makoto Shimizu †Yuya Miyahara †Shinpei Ogata

†Saeko Matsuura

†Shibaura Institute of Technology Department of Electronic Information Systems

‡Graduate School of Engineering, Shibaura Institute of Technology Department Electrical Engineering and Computer Science

半数近くの問題が未議論・未解決のままになっている理由として、支援システムでの問題把握がしづらいことがあると考えた。議論されない原因として考えられる点は以下4つある。

(i)作業項目を一覧できるページで作業項目ごとの問題を確認することができる。しかし、複数の作業項目にまたがる問題やあるメンバーの問題を把握することは難しい。

例えば、「確定・未確定事項が曖昧である」という問題は、複数のユースケース記述間に関わる問題であるが、ユースケースごとに作業項目が設定されるために、ページ遷移が必要になり把握しづらい。

(ii)様々な種類の問題が混在しているが、(i)で述べた機能では作業項目ごとに分類されるために把握しづらい。

(iii)(i)で述べた機能では問題点が報告された日時を確認することができず、作業報告書で確認する必要がある。

(iv)その問題が解決したかどうかを記録できないために、解決・未解決の把握ができない。

2. 2. 今年度における問題共有方法

今年度の実習では、上述の問題を解決するために、各作業報告書に記録された問題を集め、一覧できる「問題点リスト」機能を新たに設けた。

本機能は、作業報告時に設定したカテゴリ(未分類・成果物・グループ方針・スケジュール)ごとに、問題・対象作業項目・対象作業・投稿者・報告日時を一画面に一覧表示する。これにより、複数の問題を同時に確認することができ、閲覧したい場合のページ遷移や選択の手間が省かれる。

また、各問題に対し解決案を設定することができ、解決案が設定された問題は一覧から表示されなくなることで、問題の解決・未解決の把握ができるようになる。

未分類

日	時	状態	問題点	作業項目	対象作業・成果物	投稿者	投稿日	操作
日	10	予	多重置が中途半端に書かれた	概念モデルの作成	多重置の確認	Aさん	07/10/15	削除
			解決済	数値	数値日			
			残って欲しい、変更しなおした	Bさん	07/10/16	予	削除	
日	10	未	○○○○の問題	○○のユースケース記述の作成	○○の記述	Cさん	07/10/16	削除

図 1 問題一覧表示画面

今年度のあるグループのデータについて、2006年度と同様の集計を行った結果、問題数 96 中、未議論・未解決数は 44 であった。

機能の提供が遅れたことも原因であるが、半数近くの問題が未議論・未解決のままになっており、2006年度と変化はなかった。

3. 問題の分類

問題を共有する場を提供するだけでは目立った効果は得られなかった。

本機能では全ての問題を一様に扱っていることに原因があると考え、共有する場を提供することに加えて、問題のタイプに対応したさらなる支援が必要になると考えた。

そこで、どのようなタイプの問題があり、未議論になりやすいのかを調べるため、問題をいくつかのテーマに沿って分類し、分析する。

ここでは3つのテーマにおいて分析を行った。以下はその結果である。

(表の項目にある「割合」は、問題数に対する未議論の数の割合を示す)

i) 問題点カテゴリ別

作業報告書において問題を記録するときに選択する、カテゴリごとの集計結果である。

表2 問題点カテゴリ別集計結果

カテゴリ	問題数	未議論	割合 (%)
未分類	26	9	34.61
成果物	46	16	34.78
グループ方針	16	12	75.00
スケジュール	8	7	87.50

ii) フェーズ別

本実習における各フェーズ別の集計結果である。

表3 フェーズ別集計結果

フェーズ	問題数	未議論	割合 (%)
要求分析	35	16	45.71
システム分析	26	11	42.30
システム設計	16	6	37.50
第1期実装・テスト	13	8	61.53
第2期実装・テスト	6	3	50.00

iii) 緊急性

問題の緊急性において、下記の基準において分類した集計結果である。

高：次の作業を行うために解決すべきもの

中：解決しなくても次の作業に移れるが、解決

したほうが好ましいもの

低：影響の少ない個人的なもの・個人の感想

終：すでに終了した事に関するもの

表4 緊急性別集計結果

緊急性	問題数	未議論	割合 (%)
高	54	13	24.07
中	9	7	77.78
低	20	11	55.00
終	13	13	100.0

4. 考察

i) 問題点カテゴリ別

問題数は成果物が多いが、未議論の割合が高いのはグループ方針とスケジュールであった。

ソフトウェア開発は成果物を作成して進めていくものなので、それに応じて問題も多くなるものである。それに対して未議論の割合があまり高くはないのは、成果物ができなければ次の作業に進めないため、問題の確認を行っているからであると考えられる。

しかし、成果物に比べてグループ方針やスケジュールに関する問題は、曖昧なままでも次の作業に移ることができるために、未議論の割合が高いのだと考える。この2点に関しては、問題共有に関する支援では補い難いものが多く、別視点での支援が必要になると考える。

ii) フェーズ別

上流工程と実装工程を比べると、上流工程のほうが問題数は多いが未議論の割合は低いという結果になっている。これは、実装工程における問題がプログラムのバグを指すことが多くなるため、その一つ一つを問題として記録することがなくなるためだと思われる。そのため、プログラムのバグに関する共有が必要になると考える。

iii) 緊急性

緊急性の高い問題は、特に成果物に関するものが多い。そのため、i)の結果でわかるように、問題数が多いという結果になっている。成果物に関する問題の共有は特に重要であるとわかる。

5. 今後の展望

さらに異なったテーマでの分類も分析し、2007年度の他グループ、及び過去のデータにおいても同様の分析・共有方法の考察を行う。そして、その結果に適した支援方法を考える。

6. 参考文献

[1] 松浦：実践的ソフトウェア開発実習によるソフトウェア工学教育，情報処理学会論文誌，Vol.48, No.8, pp.2578-2595, 2007.