

学生を対象としたインスペクション支援方法の考察

栗原 紘樹[†] 松浦 佐江子[‡]

芝浦工業大学大学院 工学研究科 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

インスペクションとは、ソフトウェア開発工程において、成果物の誤りや問題点を検出するために、その作成者（依頼者）以外の他者もしくはグループ（評価者）により成果物を検査することである。

我々は、学生のインスペクションの実態を分析するためにインスペクション支援ツールを開発し、本学で実施している、UML を用いたソフトウェア開発実験に適用した。実験後に学生にアンケートを実施した結果、インスペクションの必要性は「作成者が気づかないエラーを発見できる」「メンバー間の意思の統一を図れる」というように認識しているが、その効果をあまり実感できていない[1]。その理由に、学生がインスペクションを行う際に、その成果物にのみ注目し、他の成果物との関連性を考えずにエラーの指摘や修正を行い、成果物間の整合性がとれなくなることが挙げられる。アンケートの結果では、学生の6割以上が整合性について考慮してインスペクションを行っているとは回答している。しかし、指摘内容を見ると、他の成果物と比較しての指摘はあまり見られなかった。それは、現状のツールは上述の問題点について、成果物同士を比較するための支援ができていないことが原因にあると考える。

本研究では、現状のツールの問題点を整理し、その改善案を適用した結果を報告する。

2. インスペクション支援ツール

2005 年度の実験に適用したインスペクション支援ツールでは、作業項目（フェーズ毎に与えられている必要な作業）ごとに依頼者が評価者にインスペクション依頼を出すことができる。インスペクションは、授業時間内外での対面、もしくはシステムのインスペクション支援ツールを介して非対面で行われる。

ツールには、図3にあるような入力テンプレートが依頼ごとにあり、非対面時にはテンプレートを用いてインスペクションを行う。評価者は依頼ごとに、指摘箇所（エラー発見箇所）とコメント（エラー内容）を入力する。依頼者はその指摘が採用か不採用かの判定を選択し、コメント（指摘に対する意見）を入力する。

また、ログ閲覧機能によってそれまでのインスペクションのプロセスを時系列で確認できる。

指摘箇所	評価者コメント	更新日時	依頼者の判定	依頼者コメント	更新日時
各コースケースの記述が揃っていない	可能性の向上のために、記述が揃った方が良いのではないかと。	2005/10/12 14:15:40	採用	記述を変更してあります。	2005/10/12 19:26:17
「予約日」の記述が揃っていない	使用可能な実行、使用可能な取得するときに作成された日付は、予約日と一致している必要があります。このコースケースは、予約日と一致している必要があります。	2005/10/12 14:15:59	採用	本日のプレゼンを通して、コースケースの大幅な変更が、変更日になったのでこの日に修正作業を追加してあります。	2005/10/12 19:29:29

図3 入力テンプレート画面の一部

An inspection support method for undergraduate student

[†] Hiroki Kurihara, Shibaura Institute of Technology graduate school engineering research course

[‡] Saeko Matuura, Shibaura Institute of Technology Department of Electronic Information Systems

3. インスペクション支援ツールの問題点

3.1 比較する行為につながりにくい仕様

本実験では UML を用いて成果物を作成する。インスペクションにおいて UML で定義された成果物（ユースケース記述、シーケンス図、クラス図、ステートチャート図）間の整合性を確認する必要がある。例えば、下で示す図の整合性について考慮する必要がある[2][3]。

1. ユースケース記述・シーケンス図間
2. 同じクラスが登場するシーケンス図間
3. シーケンス図・クラス図間
4. シーケンス図・ステートチャート図間

これにより、1つではなく複数の成果物を比較することで、不整合を発見したり、1つの成果物のインスペクション結果を他の成果物に反映させる必要性が生じる。

3.2 メンバー間の意思統一を図るための支援不足

複数人でインスペクションを行う場合、メンバー間において同じ考えで検査・修正されるとは限らない。例えば、あるクラスにおいてメンバー間で違う責務を考えて修正することがある。すると、そのクラスの責務が曖昧になるだけでなく、他のクラスにも影響を及ぼしかねない。その場合、グループ内でクラスの責務を決めておき、確認しながら修正することが必要になる。

また、図3上段や図4の指摘内容のように、他の成果物への影響があった場合（図3上段：「各ユースケースの記述について」とあるので他の全てのユースケースを修正する必要がある、図4：「カレンダークラスから～」とあるのでクラス図を修正する必要がある可能性がある）に、他成果物の修正が必要だという情報をグループメンバーに知らせる必要があるが、そのような支援は行われていない。

(1)	1	2	メッセージの12.5.2について	どうやって一年以上前というのを判断するのでしょうか?	2005/11/30 12:57:5	採用	カレンダークラスから現在日時を取得するなど言うように変更しました	2005/12/09 23:10:47
-----	---	---	------------------	----------------------------	--------------------	----	----------------------------------	---------------------

図4 「ログを表示する」のシーケンス図のインスペクションにおける指摘の具体例

4. 問題点の改善案

3で述べた問題に対し、提案する改善案について述べる。

4.1 他成果物への影響を通知する機能

出されているインスペクション依頼ごとに、依頼者は図5の通知書き込みテンプレートを用い、影響があると思われる成果物に対してコメントを書くことができる。

評価者は、その内容をインスペクション作業項目一覧画面で確認することができ、検査を行う前に確認し、それを考慮して検査を行うことができる（図6）。

用語集 none

No	修正点・コメント	更新日時	更新
0	属性「予約日」を「年」「月」「日」の3つに分けましょう。	none	<input type="button" value="更新"/>

シーケンス図のインスペクション(削除する) [Delete_jude](#)

シーケンス図のインスペクション(変更する) [Change_jude](#)

図5 通知書き込みテンプレートの一部

日	期	作業項目名	状態	期間	発行者
0	修正点-コメント	更新日時	通知ID	inv.ver	通知時状況
0	属性「予約日」「年」「月」「日」の3つに分けましよう。2006/10/27 4:10:28	38	0	0	作業中
1	属性「曜日」を追加しました。	2006/10/27 4:14:1	38	0	終了時

図6 通知表示画面の一部

4.2 インスペクションタスク情報表示

どんなことに注目して検査すればよいかを確認できる機能である。昨年度のツールでは、対象成果物に関するチェック項目を表示していたが、それに加えて関連成果物についてのチェック項目を提示することにより、より整合性を考慮した検査を行うことができるようになる。その一例として、システム分析フェーズのシーケンス図に対する項目を下に示す。項目ごとに確認事項と修正方針があり、例えば「基本フローのトレース」では、「ユースケース記述とシーケンス図を対比し、ユースケースを実行するために十分であるかを確認する」が確認事項、「ユースケース記述・シーケンス図の修正、クラスや操作の名前の変更」が修正方針である。

- 対象成果物に関するチェック項目
 - 形式の確認
 - 基本フローのトレース
 - 代替フローのトレース
 - 例外フローのトレース
- 関連成果物に関するチェック項目
 - 用語集
 - 対象ユースケースのユースケース記述
 - 同じオブジェクトが現れる別シーケンス図
 - クラス図

5. 評価と考察

2006年度の情報実験において、本システムから得られた学生の利用状況と学生のアンケートから分析結果を述べる。(履修者55名 内回答者53名)

5.1 追加した機能の利用状況の分析

下表1は、本ツールを介して行われたインスペクションの指摘数を集計したものである。ここでいう指摘数とは、投稿されたコメントのうち、提示した検査項目に則した内容であるコメントの数である。指摘数の割合は、全コメント中65.4%であった。

表1 グループごとの指摘数と投稿数

	A1	A2	A3	A4	B1	B2	C1
指摘数	8	129	0	33	0	0	62
投稿数	8	237	0	37	0	0	73

入力テンプレートはグループによっては使われているものの、4.1で述べた機能はほとんど使われていなかった。その原因として、次の3点が考えられる。

- 機能追加によるシステムの複雑化
- 半数のグループがツールを使用していない
- 全てのグループが、ほとんどを対面でインスペクションを行っている

特にシステムの複雑化に関しては、インスペクション以外への影響が考えられるため、改良の必要がある。

5.2 学生の整合性に対する意識の分析

「インスペクション時、どのような指摘を行ったか?」という質問において、整合性に関する項目の年度別回答率を以下表2に示す。

表2 各項目に対する年度別回答率

項目	2005	2006
シーケンス図とユースケース記述の整合性はとれているか	65.0%	88.7%
シーケンス図とクラスの整合性は取れているか(システム分析)	75.0%	77.4%
シーケンス図とクラスの整合性は取れているか(システム設計)	55.8%	75.5%

システム分析フェーズにおいて、検査項目ごとに特に多かった指摘内容を表3に示す。また、その指摘内容ごとの割合を2005年度と比較したものを以下表4に示す。

表3 各検査項目において特に多かった指摘(2006年度)

検査項目	指摘内容
形式の確認	操作の過不足・修正、クラスの過不足・修正、語彙・表記方法の統一
基本フローのトレース	操作の過不足・修正、クラスの過不足・修正、語彙・表記方法の統一
代替フローのトレース	操作の過不足・修正、代替・例外シーケンス図の不足、語彙・表記方法の統一
例外フローのトレース	操作の過不足・修正、代替・例外シーケンス図の不足、語彙・表記方法の統一

表4 特に多かった指摘の年度別割合(システム分析)

指摘内容	2005	2006
操作の過不足・修正	42.5%	36.4%
クラスの過不足・修正	16.4%	6.1%
代替・例外シーケンス図の不足	12.6%	7.6%
語彙・表記方法の統一	10.4%	27.3%

表2より、2005年度に比べ整合性に対する意識が高くなっている。また、表4からは「語彙・表記方法の統一」の割合が大幅に増加しており、意思の統一を図ることの意識の高さを見て取れることができる。これは、用語集を作ることで語彙に対する意識が高まった結果だと考えられる。用語集とは、グループ内で開発中に用いられる単語、及びその意味をまとめたもので、2006年度の本実験ではグループごとに作成することが義務付けられている。もう一つの整合性に対する意識が高まった要因として、対面でのインスペクションが多く、メンバー全員の意見を聞ける機会が多かったことが考えられる。

しかし、「語彙・表記方法」の整合性に関する指摘は増加したのに対し、「図の意味・内容」の整合性に関する指摘は2005年度同様にあまり見られなかった。

そのため、学生は、整合性を保つためには具体的に成果物のどこをどのように作成・検査・修正すればよいか分からないのではないかと考えた。その点に関してさらに支援する必要がある。

6. 参考文献

- 栗原, 松浦: 学生を対象としたソフトウェア開発におけるインスペクション効果の検証, 情報処理学会全国大会第68回, 7J-6, 2006
- 栗原, 松浦: 学生を対象としたインスペクション支援方法の考察, FIT2006 第5回情報科学技術フォーラム, N-031, 2006
- Travassos, G.H., Shull, F., Fredericks, M., Basili, V.R.: Detecting Defects in Object-Oriented Designs: Using Reading Techniques to Increase Software Quality. Proc. OOPSLA'99, p. 47-56, 1999