

学生を対象としたインスペクション支援方法の考察

An inspection support method for undergraduate student

栗原紘樹†
Hiroki Kurihara

松浦佐江子‡
Saeko Matsuura

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。

本稿では、現状のツールの問題点を整理し、その改善案について考察した結果を報告する。

2. インスペクション支援ツール

[1]におけるソフトウェア開発実験では、ソフトウェア開発プロセスのフェーズ毎に各フェーズの目標を達成するために必要な作業項目をタスクとして与え、グループはタスクを選択し、作業項目を作成することで作業計画を立案するようになっている。

タスクには「～のインスペクション」という作業項目がある。「～」の部分は、成果物の種類（ユースケース図、シーケンス図等）を指定する必要があること以外、学生が自由に設定できる。

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、その作業項目ごとに、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。依頼を出す時には、1つの成果物・評価者・期間を指定する。

依頼者	■■■■■■■■■■
成果物の種類	シーケンス図
対象成果物	user_info_registration_sequence.png
状態	進行
期間	2005/11/05~2005/11/9
評価者	■■■■■■■■■■ ■■■■■■■■■■

図2 インスペクション依頼画面の一部

†芝浦工業大学大学院, Shibaura Institute of Technology graduate school engineering research course

‡芝浦工業大学, Shibaura Institute of Technology

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インスペクション依頼ごとにより、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（エラーが発見された箇所）とコメント（そのエラーの内容）を入力する。依頼者はその指摘に対して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。

詳細一覧

依頼者ID	評価者コメント	発着日時	依頼者の判定	依頼者コメント	更新日時
■	可読性の向上のために、数行で記述した方が良いのではないかと	2005/10/12 14:15:40	採用	記述を変更してあると思います。	2005/10/12 19:25:17
■	使用は可能の発行、使用は可能と誤解する恐れがあるため、その旨を注釈に付記してください。また、その旨を注釈に付記してください。また、その旨を注釈に付記してください。	2005/10/12 14:15:50	採用	本日のレビュー結果としてユースケースの大幅な変更が予定されています。そのためこの旨に備え修正が加えられています。	2005/10/12 18:25:28

図3 入力テンプレート画面の一部

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、状態遷移図）間の整合性を確認する必要がある。例えば、下で示すような整合性について考慮する必要がある[2]。

1. ユースケース記述・シーケンス図間

- ・同じユースケースにおいて、ユースケース記述の流れをシーケンス図で追うことができるか（同じ順序で名詞が出てくるか、交換するデータの流れは同じであるか、一方にしか現れない名詞や操作はないか）

2. 同じクラスが登場するシーケンス図間

- ・クラス・メッセージの流れに矛盾はないか（同じことをしているのに違う流れで書かれていないか）

3. シーケンス図・クラス図間

- ・シーケンス図の流れをクラス図で追うことができるか
- ・シーケンス図で登場するすべてのクラスがクラス図に存在しているか
- ・シーケンス図に登場するすべてのメッセージが適切なクラスに属しているか

4. シーケンス図・状態遷移図間

- ・シーケンス図の、あるオブジェクトに注目した場合に、状態遷移図における状態遷移と合致するか

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。

この観点から、現在の支援ツールでは下の3つの問題がある。

- (1) 作業項目ごとに依頼する仕様
- (2) 1つの依頼において成果物は1つしか指定できない
- (3) 作業項目ごとにログ閲覧する仕様

(1)、(2)より、インスペクション依頼を出された成果物を検査・修正する際に、その作業項目・成果物だけに注目しがちになり、他の関連する作業項目・成果物への影響を考慮しないという問題がある。既存のツールでは、関連する成果物がわかるような支援は行われていない。

また、ログ閲覧機能も作業項目ごとであるため、ログ同士を同時に比較することができない。

3.2 メンバー間の意思の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の指摘の内容のように、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～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。

11	1	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	2006/11/30 12:51:11	指摘	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	2006/12/08 22:10:47
----	---	--------------------------------------	---------------------	----	--------------------------------------	---------------------

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

3.3 対面時のインスペクション支援不足

既存のツールは非対面でのインスペクションを支援することを目的としているため、対面時のインスペクションに対する支援は行われておらず、対面時のインスペクションの記録は学生に依存している形になっている。

アンケートの回答を見ると、「印刷物に直接書き込んでいる」「紙にメモしている」という回答がいくつか見られたが、それでは全員で見る場合などには不便である。また、ツールで行ったインスペクションの内容との比較もしづらいものになる。

4. 問題点の改善方法

3で述べた問題点に対し、提案する改善案について述べる。

4.1 インスペクション依頼作成時の成果物の複数指定

インスペクション依頼を出す際、複数の成果物を同時に指定することができるようにする(具体的には、3.1で述べた1～4の成果物の組み合わせについて)。複数指定できるようにすることで、複数の成果物を同時に比較しながら検査することにつながり、関連する成果物間の整合性を考慮しやすくなる。

4.2 関連成果物の検索機能

4.1で述べた成果物の複数指定を用いて複数の成果物を同時に検査することを促しても、関連のない成果物を同時に検査するのではあまり意味がない。そこで、ある成果物に対して、その成果物と関連のある成果物を検索する機能を考える。インスペクション依頼を出す際に、どの成果物を指定すべきかがわかりやすくなる。

検索を行う対象を全ての成果物とするのは難しいと考えたため、特に依頼が多かったユースケース記述とシーケンス図について検索機能を考えることとする(ユースケース記述は依頼全体の38.6%、シーケンス図は45.1%であった)。

3.1で述べた、整合性について考慮する必要がある項目の1、2に基づき、

- ・ユースケース記述間: 同じ名詞が使われているか
 - ・シーケンス図間: 同じクラス・メッセージが使われているか
 - ・ユースケース記述・シーケンス図間: 同一のユースケースに対する記述か
- という点を元に検索を行う。

4.3 グループ内取り決め記録機能

グループ内での決定事項を記録し、いつでも参照して確認できる機能を考える。特に、開発するシステム内で使われる語彙とその意味、システムの仕様、クラスの責務に関して取り決めたことを記録しておく。そうすることで、グループメンバー間での考えの相違を減らすことができる。

また、ここで記録した語彙を含んでいる成果物を検索できるようにすることで、4.2の機能と相まって関連する成果物を見つけやすくなる。

4.4 対面時のインスペクション支援

非対面時と同様に、システムに電子的に残すための機能を設ける必要があると考える。対面時の場合は記録するのは一人で十分なため、入力する作業量が増えすぎないように入力させる項目をある程度限定する必要がある。ただし、非対面時のインスペクションログと関連させる必要があるため、基本的な項目は同じものを用いる。

5. おわりに

本稿では、既存のインスペクション支援ツールに関して考察した。しかし、[1]で述べられているように、学生は知識不足から記述の良し悪しがわからないことがあり、検査を行うことが難しいと考えているため、インスペクションの方法自体にも問題があるのではないかと考える。そのため、検査を行いやすくするような支援が必要になる。例えば、整合性が取れているか否かを判定できる情報を提供することや、シーケンス図からクラスとそのメソッドの情報を抽出し、クラス図と照合できるようにすることなどである。その支援に加え、4で述べた改善案を考慮したツールを作成し、本学で実施しているソフトウェア開発実験に適用して評価を行うことを考えている。

6. 参考資料

- [1] 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
- [2] 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