

# ソフトウェア開発演習における問題解決情報共有システムの 適用結果分析

樫山淳雄 島田和幸 小林祐介

東京学芸大学

〒184-8501 東京都小金井市貫井北町4-1-1

**概要.** ソフトウェア開発は知識集約的な協調作業である。ソフトウェア開発では問題解決活動が繰り返し行われる。著者らはグループによるソフトウェア開発演習において発生する問題の解決とその過程を通して獲得した知識を定着させることを目指した、内省と協調による問題解決プロセスモデルとその支援システムを構築した。そして、支援システムを大学学部にて実施しているグループによるソフトウェア開発演習に2度にわたり適用した。適用の結果、2度の適用に共通の特徴として、開発工程による分類ではコーディングに関する問題解決情報が最も多かったこと、また、内容面ではプロセスに関する情報よりも技術的な情報が多いという特徴が見られた。知識流通の状況として、質問を発している学習者に対して、TAや教員が登録されている問題解決情報の所在を知らせる活動を行っている場面が見受けられた。

## Usage Analysis of a Problem Resolution Information Sharing System for a Software Engineering Project Course

ATSUO HAZEYAMA KAZUYUKI SHIMADA YUSUKE KOBAYASHI

Tokyo Gakugei University

4-1-1 Nukuikita-machi, Koganei-shi, Tokyo 184-8501 Japan

**Abstract.** Software development is a highly knowledge-intensive and collaborative activity. Problem resolution processes are performed iteratively during software development. The authors proposed a problem resolution process model that was based on reflection and collaboration for a software engineering project course. They also developed a support system and applied it to an actual university course. The results from the stored log data and the contents from two years' usage showed that similar trend on the number of registered problem resolution information, ratio of classification by phase and by contents was seen in both years (problem resolution information on coding with respect to phase was registered most, and most information was that on technical with respect to contents). We observed knowledge transfer by the teaching staff in the discussion space and in the bulletin board system of group.

### 1. はじめに

ソフトウェア開発は協調活動を伴う知識集約的な作業であり[13]、開発において問題解決活動が繰り返し行われる。ソフトウェア開発に従事する個々の開発者は開発に必要なすべての知識を有しているわけではなく、彼らは開発を行いながら、同時に必要な情報の収集を行っている。あるいは、知識を有している他者に問い合わせを行うこともある[13]。葉雲文はソフトウェア開発における知識コラボレーションの重要性を主張し、コーディング作業を対象としたサンプルプログラムの登録・検索、過去の議論の閲覧、専門家への問い合わせを可能とするソフトウェア開発環境を提案している[13]。また、文献[8]

では、産業界のソフトウェア開発における情報共有やノウハウ共有の課題を指摘している。

一方、問題解決活動を通じて獲得した知識を定着し、学習効果を向上させるプロセスとして内省(reflection)がある。また、失敗から学び、同じような失敗を繰り返さないことを目指した学問に失敗学がある[3]。失敗学では失敗について正しく記述し、その失敗を他者に伝達するために、6つの属性(事象、背景、経過、原因、対処、総括)を定義している。

我々は、グループによるソフトウェア開発演習教育に取り組んでいる[4-6]。この演習では、学習者がグループにより協調してソフトウェア開発を行う経験を通じて、ソフトウェア開発に必要な知識やスキルを習得することを目指している。そして、グループによるソフトウェア開

発演習において発生する問題を解決し、問題解決により獲得した知識を定着させることを目指した、内省と協調による問題解決プロセスモデルを提案した[7]。内省のための枠組として失敗学の考え方を導入している。そして、提案モデルに基づく支援システムを開発した[7]。このシステムは扱う情報という点で、Bugzilla, Dhruv [12, 1]のようなバグ管理システムと異なる。バグ管理システムではシステムの不具合や改善要望を管理するために使用されるのに対して、本システムではソフトウェア開発の全工程を対象にした問題解決情報を扱う。

本論文では、システムを2度の演習に適用した結果と結果に基づく分析について報告する。

本論文の構成を以下に示す。2節で問題解決プロセスモデルを概説する。3節で支援システムについて述べる。4節でシステムの適用評価について述べ、最後にまとめと今後の課題を述べる。

## 2. 問題解決プロセスモデル

我々が提案したグループによるソフトウェア開発演習における問題解決プロセスモデルを図1に示す。本プロセスモデルは、ソフトウェア開発において、開発者が直面する問題を解決する過程で、失敗学で定義された各属性情報を記述させ(表1参照)内省を促すとともに、それらを開発者間で共有することによりソフトウェア開発における問題解決に必要な知識やノウハウを習得することを目的としている。

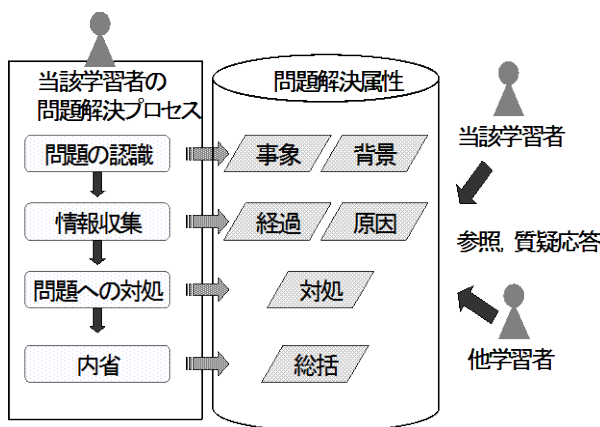


図1 問題解決プロセスモデル

本研究が想定する問題解決プロセスは、問題の認識、情報収集、問題への対処、内省のステップからなる。

### (1) 問題の認識

開発者(あるいは開発者グループ)は問題に直面したとき、その事象、背景を識別する。問題として、インスペクションやテストにより指摘された事項、プログラミングにおける不具合、開発環境構築におけるトラブル等が考えられる。

### (2) 情報収集

開発者(あるいは開発者グループ)は問題解決に必要な情報を収集し、問題の原因を検討する。情報源として、本研究で構築する支援システムあるいは外部リソースに蓄積されている事例や議論が考えられる。また、演習に参加している他開発者(グループメンバー、他グループのメンバー)や教授者(TAや教員)とのコミュニケーションにより情報を収集することも想定する。

### (3) 問題への対処

(2)で収集した情報に基づき検討した原因を解決するために対処を行う。

### (4) 内省

問題が解決できた場合には、問題解決プロセスを振り返り、学んだこと(教訓)を記録として残す。

以上の問題解決プロセスを経て蓄積された問題解決情報は他の開発者にも参照可能となるよう共有される。他開発者が情報を参照した場合にフィードバック情報(評価やコメント)を提供することが期待される。

表1 失敗学における属性の本研究における定義

属性	失敗学における定義	本研究による定義
事象	どのようなことがあったかに関する記述	発生した問題の内容に関する記述(エラーメッセージ等)と問題を含む成果物
背景	失敗の事象が生じた背景に関する記述	問題発生時の(開発・実行)環境・開発工程・ユースケース
経過	失敗の進行に関する記述	要求事項と、問題解決のために参照した情報
原因	失敗の原因に関する記述	問題の原因に関する記述
対処	失敗に対してどのようなことを行ったのかに関する記述	問題解決のために行ったことに関する記述と問題解決後の成果物
総括	失敗から学んだことに関する記述	問題解決から学んだ教訓に関する記述

## 3. 支援システム

2節で述べた問題解決プロセスモデルに基づく支援システムを構築した。本システムは分散環境においても利用できるようWebアプリケーションとして、またグループによるソフトウェア開発演習支援システム“心技体”[10]のサブ

システムとして構築した。“心技体”は、グループを単位とした成果物管理、コミュニケーション支援、レビュー等のプロセス支援を提供している。本サブシステムはグループの枠を越えて、問題解決情報の共有や問題解決に関する協調活動を可能にしている。以下に主要な機能を説明する。

■ 問題解決情報登録機能

問題解決情報の登録を行う。登録者が表 1 の属性に従って入力することにより、問題に関するコンテキストを記述することができ、また内省を喚起することが期待される。

■ 問題解決情報参照機能

登録されている問題解決情報を参照する。参照画面からは、当該問題解決情報に対する評価やコメントを登録することができる。図 2 に問題解決情報参照画面を示す。

タイトル	Eclipseのビルド・パスエラー??	環境設定	Tomcat: 5 mysql: 5 java: 1.4 eclipse: 3.2
工程	コーディング・単体テスト	担当ケース	?
背景・問題	データベースとの接続ができなくなった。Eclipseが自動的にビルドしてくれなくなった。	原因	いろいろmysqlコネクタが入ってしまっていて、ビルド・パスエラーが生じていた。
対価	いろんな人に相談 ・ネットで検索(有用な情報はなかった) ・コミュニティを利用(フォーの所在をききとめる)	考察	どこにエラーがでているのかを知らないと、エラーの解決がつかない。 コミュニティを利用したことで早い解決につながりました。ほんとにありがたうございます!
参照Webサイト	なし	キーワード	Eclipse ビルドされない
添付ファイル	なし	議論内容へのリンク	<a href="#">リンク</a>

編集する 問い合わせ/コメント付加

2人中2人が  
このノウハウは役に立つと登録しています

図 2 問題解決情報参照画面

■ 質問登録機能

自身では解決できない問題について、質問を登録するための機能である。回答者が質問内容を的確に把握し、質問・回答を効率的に行えるよう、ここでも失敗学の属性の一部を活用した。事象については、KT法[9]の問題分析手法の属性も取り入れ、現在起こっている問題と実現させたいことに項目を分け、質問内容の構造化を図った。

■ 質問に対する回答登録機能

登録されている質問を解決するための議論・回答を行うための機能である。議論の経過を閲覧することも可能である(図 3)。議論により質問内容が解決した場合に、質問投稿者は問題解決情報を登録することが期待される。問題解決情報

の登録を効率化するために、質問時の属性の内容を引き継ぐことができる。また、質問・回答を経て登録された問題解決情報からは、問題解決に至った質問・回答も閲覧することができる。

登録者	akko	タイトル	Eclipseのエラー??
背景・直面している問題 (実現したいこと)	Eclipseでファイルを作成・更新しています。更新した後、ファイルを実行してもその更新がなされていません。	背景・直面している問題 (実際の状況)	Eclipseでファイルを作成・更新しています。更新した後、ファイルを実行してもその更新がなされていません。
どんな対処を回ったか	ファイルを書き換えた後、 ・Tomcatの再起動 ・Eclipse再起動 ・パソコン自体の再起動 しましたが状況が変わりません。	考え得る原因	クラスが更新されていないので、コンパイルができていないのかも?
工程	コーディング・単体テスト	ユースケース	分類名登録など
開発環境	java: 1.4 mysql: 5 eclipse: 3.2.1 tomcat: 5	添付ファイル	なし

1. (どう?)	2007/12/21 08:37
ビルドされていないんじゃないですか?	
2. akko	2007/12/21 12:46
たぶんビルドされてないです...	
自動ビルドにチェックが入っている状態です。チェックはすべてビルドとか試してみましたが動いていないみたいです。どこを直したらいいのでしょうか...	

図 3 質問とそれに対する回答参照画面

4. 適用評価

本研究で提案した問題解決情報共有システムの有効性を検証するために、大学におけるソフトウェア開発演習に適用を行った。本節では適用対象、結果とその分析について述べる。

4.1 適用対象

システムを東京学芸大学教育学部情報教育専攻の3年次後期に開設されている「システム設計演習」の2007年度と2008年度の演習に適用した。本演習では4名から5名でグループを構成し、各グループが教員から与えられた課題を実現するWEBアプリケーションシステムを開発するというものである。演習では要求分析にはじまり設計、実装、テストという工程を経てソフトウェアを完成することが求められる。開発ではUMLやJavaなどのオブジェクト指向技術を用いて行う。上流工程の成果物に対してインスペクションが、開発されたシステムに対して受入テストが教員並びにTAによって行われている。

本演習に先立ち、著者らの専攻ではプログラミング言語の教育としてC言語のコースが計1年半開設されている。また、2年次に「オートマトンと言語理論」が開設されている。さらに本演習の直前の学期にソフトウェア工学の入門コースが開設されている。このコースでは、ソフトウェア開発ライフサイクルモデル、オブジェクト指向の概念、UMLによるモデリング、

Java 技術を用いた Web アプリケーション開発を行っている。しかしながら、開発に必要なすべての知識を講義において教授することは困難であるので、演習において直面した問題を解決するために自ら調査したり、情報を交換することが求められる。

2007 年度の受講者は 22 名で、5 つのグループが編成された。システムの適用期間は 2007 年 10 月 31 日から 2008 年 1 月 22 日までの 84 日間であった。2008 年度の受講者は 26 名で、6 つのグループが編成された。システムの適用期間は 2008 年 10 月 30 日から 2009 年 1 月 21 日までの 84 日間であった。開発スケジュール策定はグループに委ねられているので多少の違いはあるが、両年度とも 12 月中旬まではシステム分析・設計作業を行い、12 月中旬から 1 月初旬にかけてコーディング、単体テスト、システム結合を行い、1 月中旬にシステムテストとバグ修正が行われている。両年度ともにすべてのグループが開発を完了した。

受講者には演習開始時に、開発演習で得た問題解決情報を最低 1 人 1 件本システムに登録するよう指示を行った。自身の問題解決を振り返るという趣旨もあるので、他者が登録した情報と同じ内容のもの(複写を認めているわけではない)に登録することも認めた。

## 4.2 結果

適用結果について、工程による分類、内容による分類、協調的問題解決の観点から報告する。

### 4.2.1 問題解決情報の工程による分類

2007 年度の演習では 39 件の問題解決情報が登録され、2008 年度には 42 件の問題解決情報が登録された(一人あたりの登録数は 2007 年度が 1.8 件、2008 年度が 1.6 件)。図 2 は実際に登録された問題解決情報の一例を示している。図 4 は 2007 年度と 2008 年度に登録された問題解決情報の工程ごとの割合を示している。開発工程を、要求分析、設計、コーディング(開発環境構築も含む)、テスト、プロジェクト管理、その他に分類した。図が示すように、問題解決情報の工程による分類では、両年度ともに似たような傾向を示した。すなわち、コーディングに関する情報の登録が最も多く、次いで設計に関する情報が多かった。

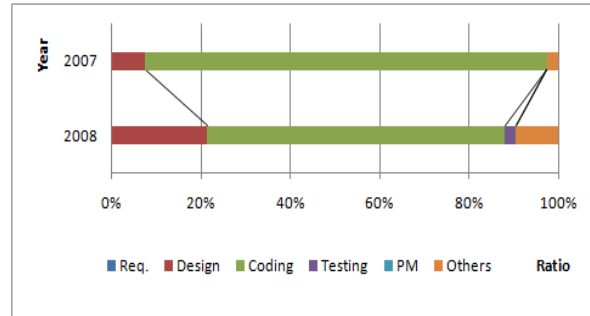


図 4 登録された問題解決情報の工程による分類

### 4.2.2 問題解決情報の内容による分類

問題解決情報を石田らによって提案されたカテゴリ [8]—「作り方」と「進め方」—に従い分類した。図 5 はその結果を示す。2007 年度では 37 件(95%)は「作り方」(ツールの使い方も含む)に関するものであり、「進め方」に関するものは 2 件(5%)であった。「進め方」に記述された 2 件は技術的問題に遭遇した時の解決方法であった。2008 年度に登録された問題解決情報も約 90%が技術的な問題であった。

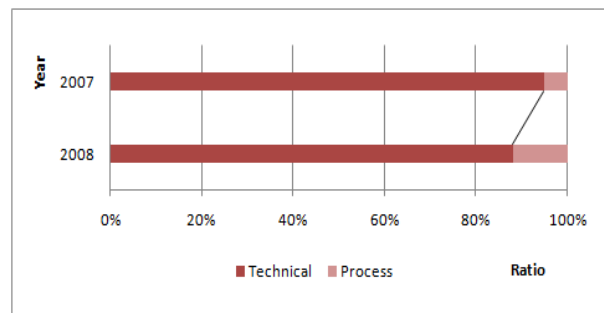


図 5 登録された問題解決情報の内容による分類

### 4.2.3 協調的問題解決(知識流通)の状況

質問登録機能により、2007 年度は 8 件、2008 年度は 1 件の質問が寄せられた。それらはすべて解決に至っている。これらのうち 6 件は 1 度の回答により解決に至っている。また、1 件は問い合わせ内容が簡潔すぎて状況の把握が困難であるという理由から、TA が質問内容に関する詳細を聞き出していた。この事例では、質問を發した受講者は詳細な説明を記述するとともに、その後の調査により問題の真の原因を見出したようである。TA は受講者が見出した原因を取り除くためのアドバイスをして終結している。

残りの 2 件の質問は質問者と回答者の何度かにわたるやり取りにより解決に至っている。

このうちの 1 つは「現在時刻を DB に書き込みたい」という要望で、1 人の TA が、質問者から現状と要求を聞きつつインターネット上で調べ

た情報を提示し、5回のやり取りで解決に導いていた。

もう1件は「ビルドに関する問題」で3人の回答者による6回のやり取りで解決に至った。この事例では最初に他のグループメンバーが原因を予想した。しかし具体的な解決策の提示には至っていない。次にTAの1人(TA1と記す)がいくつかの質問をし、その回答をもとにいくつかのアドバイスを送ったが解決に至らなかった。その後別のTA(TA2と記す)がログファイルを確認するようアドバイスした。質問者により提示されたログファイルの内容からTA1が具体的な解決策を見出し、解決に至った。質問者は「どこでエラーが出ているかを調べることが解決につながる。質問登録機能を利用することにより解決につながった」と振り返っていた。

何人かの受講者が、問題解決情報が登録されているにも関わらずそれに気づかずに、質問登録機能やグループのBBSに問い合わせを行っていた。それに対して教授者が知識の仲介役を担っており、関連すると思われる情報の所在を教えていた。以下がその例である：

#### 例 1

受講生 S1: いろいろ調べたのですが、メール送信機能をどのように実現すればよいかわかりません。

教員: 2008年1月22日 20:26 に登録された問題解決情報が参考になると思います。私は、この問題解決情報を登録した人は苦勞してこの問題を解決したことをよく覚えています。

#### 例 2

受講生 S2: 作成したプログラムをコンパイル後、ブラウザでURLを指定しました。しかし、404エラーが起きました。どうすればよいでしょうか？

TA: 2007年11月1日 9:34 に登録された問題解決情報を参照してみてください。

問題解決情報へのフィードバック情報から、少なくとも7件の問題解決情報が他の受講者の問題解決に再利用されていることがわかった。それらはDBのバックアップの方法、サブページからメインページへの制御、メール送信機能、関連クラスであった。

### 4.3 議論

#### 4.3.1 問題解決情報について

2年間のシステムの適用実験の結果、登録された問題解決情報の数、問題解決情報の工程による分布、問題解決情報の内容による分類は似た傾向を示していた。工程による分類では、コーディングに関する問題解決情報が最も多かった。本演習では、受講生はいくつかのユースケースを担当する。そのため、必要となる技術は各人異なる可能性がある。したがって、直面した問題は自身で解決しなければならず、問題が顕在化しやすい。一方、本演習では設計はグループ活動として行っている。また、本演習では分析、設計段階の成果物に対して、教授者をインスペクタとしたインスペクションを実施している。各グループはインスペクションのコメントに従い成果物を修正することが求められる。設計工程に関する問題解決情報を分析してみると、多くはツール(UMLエディタ)の使い方に関するものであった(2007年度では3件の問題解決情報のうち2件が、2008年度では6件のうち2件がツールの使い方に関するものであった)。

内容の観点から登録された問題解決情報を分析すると、多くの情報は技術的なものであった。2007年度には技術力に自信がないと自ら述べている学生が以下のような問題解決情報を登録していた:「答えを直接聞くのではなく、調べ方を聞くという態度が重要である」、「わからないことがあってもあきらめてはいけない。積極的に関わりなさい。そうすれば、困ったときに仲間が手伝ってくれるかもしれない」。これらは問題解決方法というプロセスに関する情報である。

#### 4.3.2 知識流通について

本事例では質問・回答並びにグループのBBSにおいて教授者が問題解決情報の存在を知らせるという活動を行っていたことが観察された。

Boden と Avram は小さなソフトウェア企業における地理的に分散したプロジェクトでの拠点間の知識流通の状況を調査した[2]。彼らはSkypeによる口頭コミュニケーション、出張、他拠点に滞在している技術者による知識の橋渡しの重要性を指摘している。

我々の演習は、地理的な分散という状況ではないが、受講者はそれぞれのスケジュールを有するため時間を共有することが困難なようである。従って、同期コミュニケーションは限られる状況にある。また、本演習では人(受講生とTA)の入れ替わりが非常に速い。知識流通の観点からすると、口頭コミュニケーションは揮発して



しまうので、本演習の性質を考えると適切ではない。これらの理由から問題解決情報を記述、蓄積、共有する方法を採用した。中小路らは専門知識に関するコミュニケーションのための設計ガイドラインとして 9 項目を提案している [11]。彼らは、何らかの情報を得るために文書やコードが存在するならば、コミュニケーションが発生しないようにできるだけそれらを活用すべきであることを述べている。我々のシステムはこの項目を満たしている。

## 5. おわりに

本論文では、大学におけるソフトウェア開発演習のための問題解決情報共有システムの 2 度にわたる演習での適用状況について報告した。問題解決情報の登録数、工程並びに内容による分類について 2 度にわたる適用において同じような傾向を示した。工程についてはコーディングに関するものの割合が最も多く、内容に関してはプロセスに関するものよりも技術的内容が多かった。また、システムに問題解決情報が登録されているにも関わらず受講者がその存在に気づいていない場合に、TA や教員がその存在を示唆し、知識流通における仲介役の役割を果たしている事例が見られた。

今後は、問題解決情報がどのように有効活用されているのかについて調査を進めたい。

## 謝辞

本研究の一部は、科学研究費補助金(C) 18500701 の助成のもとで行われた。記して謝意を表す。

## 参考文献

- [1] A. Ankolekar, K. Sycara, J. Herbsleb, R. Kraut, and C. Welty, Supporting Online Problem-Solving Communities with the Semantic Web, Proceedings of the 15th World Wide Web Conference (WWW2006), pp.575-584, ACM Press, 2006.
- [2] A. Boden, and G. Avram, Bridging knowledge distribution - The role of knowledge brokers in distributed software development teams, Proceedings of the ICSE 2008 Workshop on Cooperative and Human Aspects of Software

- Engineering (CHASE2009), ACM Press, 2009.
- [3] 畑村洋太郎, 失敗学のすすめ, 講談社, 2000.
- [4] 樫山淳雄, 業務ソフトウェア設計・開発教育の実践とその評価, 教育システム情報学会誌, Vol.17, No.3, pp.367-378, 2000.
- [5] 樫山淳雄, 岩崎新一, グループによるソフトウェア開発演習実践の分析, 情報処理学会ソフトウェアエンジニアリングシンポジウム 2008 (SES2008), pp.155-162, 近代科学社, 2008.
- [6] A. Hazeyama, A Case Study of Undergraduate Group-based Software Engineering Project Course for Real World Application, Proceedings of the First International Symposium on Tangible Software Engineering Education, (STANS2009), pp.39-44, 2009.
- [7] A. Hazeyama, K. Shimada, and Y. Kobayashi, A Collaborative Problem Solving Support System for Group-based Software Engineering Project Course and Its Application, Proceedings of the Seventh International Conference on Creating, Connecting and Collaborating through Computing (C5 2009), pp. 74-78, IEEE Computer Society Press, 2009.
- [8] 石田厚子他, ソフトウェア開発における情報共有の課題と効果に関する研究, [http://www.juse.or.jp/software/pdf/17\\_spc/17dep71.pdf](http://www.juse.or.jp/software/pdf/17_spc/17dep71.pdf).
- [9] C.H.ケプナー, B.B.トリゴア, 上野 一郎(訳), 新・管理者の判断力—ラショナル・マネジャー, 産能大出版部, 1985.
- [10] M. Miura, Y. Kobayashi, K. Shimada, K. Takahashi, S. Seiki, and A. Hazeyama, A Proposal of Integrating Personal and Community Support with Learning Environment for Group-based Software Engineering Course, Proceedings of the 2nd International Conference on Knowledge, Information and Creativity Support Systems (KICSS2007), pp.144-151, 2007.
- [11] 中小路久美代, 葉雲文, 山本恭裕, ソフトウェア開発における知識コミュニケーションのためのインタラクションデザイン, 人工知能学会全国大会, June, 2009.
- [12] N. Serrano, and I. Ciordia, Bugzilla, ITracker, and other bug trackers, IEEE Software, Vol. 22, No.2, pp.11-13, 2005.
- [13] Y. Ye, Socio-Technical Support for Knowledge Collaboration in Software Development Tools, Proceedings of the Workshop on Integrating Software Engineering and Usability, pp.39-51, 2005.