

# 流用開発におけるトレーサビリティを利用したレビュー支援の方法とその支援ツールの提案

周 一凡<sup>†1</sup> 中島毅<sup>†1</sup>

**概要:** 近年、ソフトウェアの大規模化、製品機能の高度化が進み、要求仕様の漏れが IT 業界で問題となっている。この問題を解決するために、トレーサビリティが注目されている。SRTracer は開発工程、作業内容を組み込んだタグ ID から、トレーサビリティリンクを構築し、その結果を要求トレーサビリティマトリクスとクロスリファレンスマトリクスとして可視化するツールである。流用開発をする際に SRTracer 単体では以前の要求仕様書への要求の追加によって足さ下位工程にどのように影響が出るかをレビューにより確認することが難しいという問題である。その為、バージョン間の要求が実装されているかを確認するためには構成管理ツールとの連携によるレビュー支援ツールを提案する。

**キーワード:** 要求仕様, 流用開発, トレーサビリティリンク

## A method of review support using traceability in diversion development and proposal of its support tool

Yifan Zhou<sup>†1</sup> TSUYOSHI NAKAJIMA<sup>†1</sup>

**Abstract:** In recent years, the scale of software and the advancement of product functions have advanced, and leakage of required specifications is a problem in the IT industry. In order to solve this problem, traceability has attracted attention. SRTracer is a tool for constructing a traceability link from the tag ID incorporating the development process and work content and visualizing the result as the request traceability matrix and cross reference matrix. In diversion development, SRTracer alone is a problem that it is difficult to confirm by review how the previous requirement specification is influenced in addition to the request by addition of a request. For this reason, we propose a review support tool in cooperation with the configuration management tool in order to check whether requirements between versions are implemented.

**Keywords:** Requirements specification, diversion development, traceability link

### 1. はじめに

近年、ソフトウェアの大規模化・複雑化、製品機能の高度化が進み、要求仕様の実装漏れやテスト漏れの問題が社会問題となっている。こうした問題を防止するために、一つの製品に関わる要求仕様から設計仕様、ソースコード、テスト仕様に至る関係を保持し、レビューやテストにより確認することが求められる。これをトレーサビリティ管理と呼ぶ[1]。

大規模な製品では、管理すべきトレーサビリティの数は膨大になるため、リンクの生成・削除・可視化し、リンクをたどってドキュメント間の対応箇所を表示することでレビューを支援するなどの機能を持ったトレーサビリティツールが提案されている[1][2]。

しかし、近年のソフトウェア開発は、通常、1 回限りの新規開発で終わることはなく、既存の製品からの仕様の変

更、機能の追加を繰り返し製品展開していくケースが多い。しかも、こうした製品展開では、ユーザーニーズの多様化への対応の必要性や競争の激化により、製品のバリエーションも増え、製品開発サイクルも短くなっている。

流用開発は、要求変更により既存製品の一部を変える開発スタイルになる。流用開発では、既存製品の成果物間においてトレーサビリティが確立していたとしても、要求変更を実装する作業を進める中で、本来あるべきトレーサビリティの一部を喪失し、既存機能や設計要素の整合性を崩すようなミスが入りやすくなる。これにより、①変更された機能が全体として正しく実装されない、②既存の機能が新たに不具合を起こすことなどが起きてしまう。

我々は、こうした流用開発における変更要求仕様の実装ミスや、既存機能の整合性の喪失を防止するため、タグ付きトレーサビリティツールと構成管理ツールを連携し、変更要求の実装に伴うトレーサビリティの適切な保守を支援するための、レビュー支援システムを提案する。

本報告では、2 節で本研究のベースとなるトレーサビリティツールと構成管理ツールについて説明し、3 節で流用

<sup>†1</sup> 芝浦工業大学  
Shibaura Institute of Technology

開発における解決したい課題を設定し、4 節でその問題を解決するためにトレーサビリティツールと構成管理ツールを連携させたレビュー支援システムを提案し、5 節で今後の評価計画、6 節で関連研究についてまとめる。

## 2. 研究のベースとなる技術

### 2.1 トレーサビリティツール SRTracer

#### 2.1.1 SRTracer の位置づけと採用理由

トレーサビリティツールは、要求仕様書や方式設計書など成果物ドキュメントの該当箇所間を何等かの手段により関連づけ、その情報を管理する。この該当箇所間の関連付けをトレーサビリティリンクと呼ぶ。

この関連付けの方式には、大きく直接リンクとタグ付きの2つがある。直接リンク方式は、該当箇所間に直接的なリンクを張りそれをデータとして管理する。これに対してタグ付け方式は、文書にタグと呼ばれる名前付きマーカーを埋め込み、名前による識別によりリンクを間接的に生成する。直接リンク方式は専用の文書ツールとして実現される場合が多く、タグ付け方式は既存の文書ツールを利用するが多い。

構成管理ツールとの連携を考えた場合、タグ付け方式はリンク情報の履歴がタグ情報（タグ付けされた文書要素）の履歴としてビジブルであるため、有利である。そのため、本研究では、タグ付け方式のツールとして SRTracer[3]をベースとして採用する。

#### 2.1.2 タグ付け方法

SRTracer は、成果物ドキュメント文書の要件とその実装・検証事項との間の関係を保持するため、構造化タグ ID を採用している。構造化タグ ID は、要件の識別とその要件に対応する開発工程内の識別を組み合わせさせたタグ ID である。

要件の識別として、“SR-aa”（a は数字）をタグ ID として、各要件に対して定める。SR は“Software Requirement”を意味する。変更要求には CR(Change Requirement) を先頭文字として定義する。

要件識別以降の識別番号は、工程識別別に用いる。開発工程は、図 1 に示す一般的なソフトウェア開発工程の定義に従いレベルを決定する。

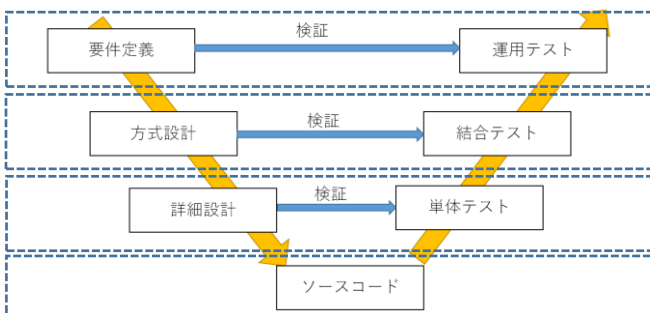


図 1 工程内識別をする工程レベル

例えば、ソフトウェア方式設計書では、要件定義書の要件“SR-aa”に対して、2 階層目なので設計箇所“SR-aa-bb”（bb が工程内識別）タグを付ける。一般的に一つの要件は複数の設計箇所に対応するので、同一工程レベルで識別できるようにする。ソフトウェア詳細設計については、“SR-aa-bb-cc”のように、もう一段識別子の階層が深くなり、直上のレベルの特定要素（“SR-aa-bb”）に対応するのであることを明確にする。

SRTracer ツールにはタグのグループ化機能がある。これは、同一箇所にある異なるタグを自動的にグループ化して扱う機能である。

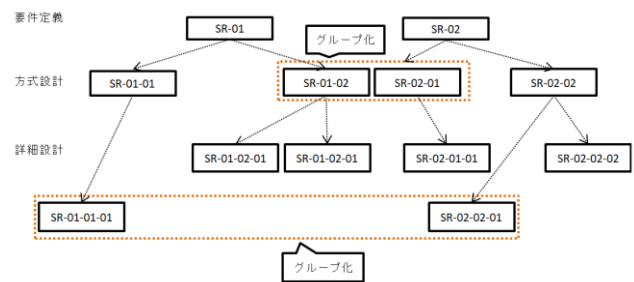


図 2 タグのグループ化

図 2 に示すように、SR-01-02 と SR-02-01 は異なる要件（SR-01 と SR-02）が同一の方式設計内容に関係づけられていることを表している。この機能により、2 つの要件を満たすために同一の設計要素が必要なことを表すことができるとともに、変更要求など後から追加された要求に対して、方式設計以降の影響範囲はどこかを識別することができる。

#### 2.1.3 トレーサビリティマトリクス作成機能

SRTracer ツールは、要件をキーとしたタグ付けるによる成果物に記載したタグの名称と位置情報から、トレーサビリティリンクを構築する。構築したトレーサビリティリンクは、2 つの成果物ドキュメント間の関連のある箇所を示すトレーサビリティマトリクスとして可視化することができる。

## 2.2 構成管理ツール Subversion

構成管理ツールとは、成果物ドキュメントやソースコードなどを管理対象として、個々の管理対象のバージョンの管理と、管理対象全体に対するリビジョンの管理を行うためのツールである。

ここで、管理対象の成果物はソースコードやテストスクリプトなどのテキストデータのみからなるファイルだけでなく、Word や Excel などの仕様書、設計書のバイナリファイルも扱う。リビジョンの管理とは、個々のバージョン管理された対象に対して、実施するものであり、テスト終了時や納入時など“ある時点”の論理的な一貫性を持った成果物群を保持・再現するための機能である。

Subversion (SVN) は、集中型のリポジトリ（構成管理サーバ）を利用するタイプの構成管理ツールである。

Subversion の特長は、テキストデータだけでなく Word や Excel で作成されたドキュメント (バイナリ) のデータも扱えること、集中型リポジトリを利用しているためプロジェクト型で一貫したソフトウェア開発を進めるのに向いていることなどである。

図 3 に示すように、構成管理のユーザは、主に commit と checkout/update により SVN リポジトリを利用する。Commit は、ユーザのローカルで作業した最新バージョンのファイル群を SVN にアップロードする。Checkout は、SVN の最新のバージョン、あるいは特定のリリースの成果物群を SVN リポジトリからユーザ環境へダウンロードする。Update とは、SVN からユーザのローカルに最新バージョンのファイルにダウンロードする。

これらのリポジトリ操作コマンドは、公開されているのでアプリケーション連携に向いている。

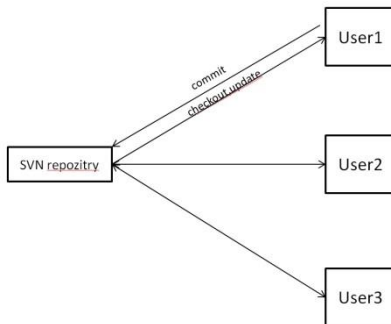


図 3 Subversion の概念図

### 3. 流用開発における解決したい課題

ここでは、流用開発におけるトレーサビリティに関する具体的な問題を図 4 の例によって説明する。

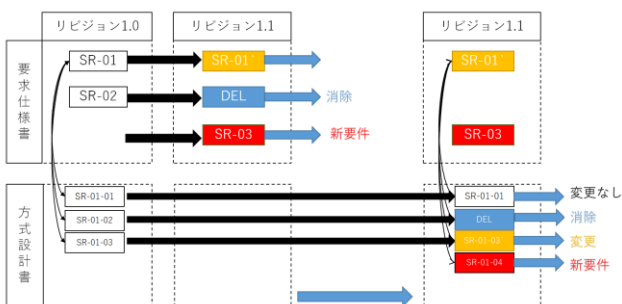


図 4 要件はリリース間で変更図

図 4 は、ある製品の出荷が終わり (リリース 1.0 の確率)、拡張開発 (リリース 1.1 になる予定) がスタートした状況において、拡張開発のための変更要求により、要求仕様書の一部に変更が起きた状況を表す (SR-01 は SR-01' に変更, SR-02 は削除, SR-03 が新要件)。

この状況で、方式設計書などにもその変更は展開される (SR-01 から SR-01' への変更は、リリース 1.0 の設計要

素 ST-01-01 にそのまま利用し, ST-01-02 は削除され, ST-01-03 は変更され ST-01-03' に, ST-01-04 は新たに追加されている)。

この状況下で, SR-01' への変更は, 方式設計書でどのように修正されたか, その変更で本当に充分であるかその差分を出力し, 比較する必要がある。しかし, このことは, リビジョン 1.1 内だけを対象に, 要求仕様書と方式設計書を見比べるだけでは十分に行うことはできず, リビジョン 1.0 からの変更が適切に行われているかどうかをみる必要がある。

つまり, リビジョン 1.1 内の成果物間に限定して, トレーサビリティリンクを使いドキュメント比較するだけではこの要求を満足することはできない。

## 4. 提案システム

### 4.1 レビュー支援ツールの提案

我々は, 3 節で示したような状況に対して, 構成管理ツールと連携することで, 流用開発においてリリースにまたがる変更が適切に行われているかどうかを確認することを支援するレビュー支援ツールを提案する。

以下に, 提案するレビュー支援ツールのシステム構成と機能について述べる。

### 4.2 システム構成と機能の流れ

図 5 は, 提案するレビュー支援ツールのシステム構成図である。

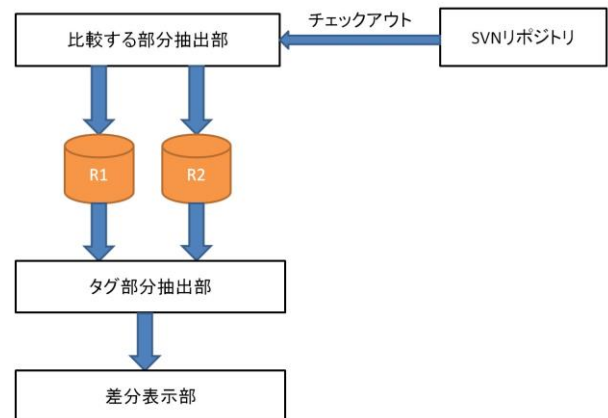


図 5 システム構成

レビュー支援ツールは, 比較する部分抽出部, タグ部分抽出部, 差分抽出部からなる。

#### (1) 比較する部分抽出部

比較する部分抽出部は, 比較対象の流用開発におけるベース製品 (R1) と, 改良部分 (R2) を SVN から抽出する。このとき, R1 はトレーサビリティツール SRT racer のタグ付けルールに沿って, SVN リポジトリにコミットされているリリースである。改良部分 R2 はその R1 への変更を順次コミットした最新版であるとする。

#### (2) タグ付け抽出部

タグ付け抽出部は、対象となる成果物ドキュメントから、タグとタグ付け対象部のテキストを抽出する。SRTracer タグ付けは、図 6 に示すように、WORD や EXCEL の表形式やコメント機能を使ってタグ付け対象にタグ付けしているの、.NET を利用してタグからタグ付け対象を抜き出すことができる。

要件定義書のドキュメント		
タグ	機能	内容
CR-0024	クロスリファレンスマトリクスの出力	指定した2つのフェーズの要件の関係を表
CR-0025	検索後 XML とマトリックス作成	フェーズ設定で指定したファイルからタグ

方式設計書のドキュメント	
③ CR-0024-0001	クロスリファレンスの作成 <ul style="list-style-type: none"> <li>収集したタグ情報から、2つの開発フェーズの要件の関係を表すマトリクス（クロスリファレンスマトリクス）を作成する。</li> <li>クロスリファレンスは、①マトリクスと同じ EXCEL ファイルの別シートに出力される。このため、作成の元となるシートをあらかじめ設定する必要がある。</li> </ul>

図 6 WORD や EXCEL の表形式やコメント機能を使ってタグ付け

(3) 差分表示部

差分表示部は、タグ付け抽出部が抽出した、R1 と R2 のそれぞれのタグ情報とタグ付け対象部のテキストを読み込み、差分を表示する部分である。

図 7 に差分レビュー表示部の出力のイメージを示す。R1 の要求仕様書のある要求に注目したときに、以下を確認することができる。

- R1 ではそれがどのように方式設計書に展開されているか(左縦)
- R2 では要求としてはどのように変更されているか(上横)
- その変更された要求がどのように R2 方式設計書に展開されているか(右縦)
- 方式設計書が R1 から R2 に実装が変化しているか(下横)

	前製品の最終リビジョン	現開発の最新リビジョン
要求仕様書	状態 ファイル(旧) UPD [SR-0001] 対象するファイルをテキストファイルとし、ト	ファイル(新) [SR-0001] 対象するファイルをテキストファイルとWor
方式設計書	状態 ファイル(旧) [SR-0001-0001] ファイルの絶対パス取得する。 UPD [SR-0001-0002] テキストファイルの読み込みを行う。 ADD [SR-0001-0003] 文字列の読み込みを行う。	ファイル(新) [SR-0001-0001] ファイルの絶対パス取得する。 [SR-0001-0002] テキストファイル、Wordファイルの [SR-0001-0004] ファイルからドキュメントを読み込む。 [SR-0001-0003] 文字列の読み込みを行う。

図 7 差分レビュー表示部の出力のイメージ

4.3 差分表示によるレビュー支援の期待効果

流用開発における製品の品質保証のカギは、成果物に対する変更機能の確実な実装と、変更による影響により不具

合が発生しないことをレビューにより確認することである。

提案するレビュー支援ツールにより、流用開発において問題となる：

- ① 変更要件の実装(縦)
- ② 変更による影響範囲(横)

を要件ごとに吟味することができる。これにより上記のレビューを確実に行うことが可能となる。

レビューの際に、変更要件のレビューで R1 と R2 のそれぞれ2つ(計4つ)の成果物ドキュメントの該当位置を探し確認する手間もなくなるため、レビュー時間を大幅に削減することも期待できる。

5. 提案ツールの評価計画

SRTracer ツールの有効性を評価するための実験計画を以下にまとめる。

5.1 実験計画

(1) 目的

提案するレビュー支援ツールが、流用開発における要求仕様の実装漏れの防止のために有効であるかどうかを評価する。

(2) 評価対象

使用する要求仕様書は、要求仕様書のサンプルはシステム文書品質研究会を発行する、「SWP1-D301 ムーブレゴ通報システムソフトウェア要求仕様書」[4]を利用する。ただし、実際に評価に利用した部分は、目次によって、「第三章入出力値の定義」の部分を利用する。

(3) 実験内容

作業者が実施するレビュー項目は3つ:(1)個々の要件の展開状況の確認,(2)変更した要件の箇所の確認(3)要求定義を変更し方式設計の変更状態の確認。

(4) 評価

SRTracer ツールと支援ツールと連携を使った場合とツール使わない場合のレビュー時間を比較して、ツールを使った場合レビュー項目を実装漏れの防止の状況と使わない場合防止の状況を比較する。

6. 関連研究

ソフトウェアが、流用開発において、要求の追加や変更を受け続けていくことで、トレーサビリティを喪失するという問題がある。

トレーサビリティの回復や派生開発におけるトレーサビリティリンクの抽出などの開発成果物間のトレーサビリティを自動復元する研究[5]がなされている。

ツールの連携の問題を解決するため、具体的な開発成果物以外に、開発に関わる情報の管理を含む、API 提供する。トレーサビリティ情報の種類が多く、トレーサビリティリンク形を設定する。版が更新されるたびにトレーサビリティ

情報が不明とならないよう,該当箇所に変更がなかった場合は,古い版と同一の構成要素とみなす機能が必要である[6].

## 7. おわりに

本研究では要求仕様の実装漏れを防ぐために,タグ付けトレーサビリティツールと構成管理ツールを連携させたレビュー支援ツールを提案と評価方法を行った.今後の予定は,レビュー支援ツールの実装と評価することを実施する.

## 参考文献

- [1] Gotel, Orlena CZ, and C. W. Finkelstein: "An analysis of the requirements traceability problem," Proceedings of the First International Conference on Requirements Engineering. IEEE, 1994.
- [2] 宇田川佳久: ウォーターフォールモデルに基づく情報システム開発における成果物の品質管理手法について, 情報処理学会論文誌 pp. 922-929, 49(2), 2008
- [3] 徳本修一,谷垣宏一,高橋洋一,中島毅: "タグ情報による要求トレーサビリティリンクの構築と可視化." 研究報告ソフトウェア工学, pp. 1-6 SE-46, 2015
- [4] SWP1-D301 ムーブレボ通報システムソフトウェア要求仕様書
- [5] 上田健之: 情報検索手法に基づくトレーサビリティリンク回復のための手法オプションについてのマイニングの提案と評価,電子情報通信学会論文誌 D,Vol.97,No.3,414-426,2014.
- [6] 渥美紀寿,小林隆志,高田広章. "ソフトウェア開発におけるトレーサビリティ確保のための開発環境の検討." 研究報告ソフトウェア工学 (SE) 2012.11 (2012): 1-7.