オープントレーサビリティツールプラットフォーム TERAS

キャッツ株式会社 グループマネジャ
一般社団法人 TERAS プロジェクトマネジャ
宮本　貴之

キャッツ株式会社 取締役 副社長
一般社団法人 TERAS 開発委員長
渡辺　政彦

名古屋大学大学院情報科学研究科情報システム学 教授
一般社団法人 TERAS 技術委員長
高田　広章

学校法人専門学校 HAL 東京 校長
一般社団法人 TERAS 理事長
鶴保　征城

オープントレーサビリティツールプラットフォーム TERAS※1 は、2013 年 6 月にバージョン 2 がリリースされた。2014 年にバージョン 3 のリリースに向けて開発中である。トレーサビリティとは何かをコモンクライテリアを例に具体的に示し、その後、TERAS のアーキテクチャ、機能、そして課題について紹介する。

1 はじめに

近年、安全・安心な社会に向けて機能安全やセキュリティが重要なテーマである。機能安全規格やセキュリティ評価基準ではソフトウェアの「トレーサビリティ」を要求する。ソフトウェアの「トレーサビリティ」とは、『ソフトウェア開発の成果物である文書間において追跡が可能である』ことである。良好な安全性や高いセキュリティの指標として「トレーサビリティ」を示すことが説明責任を果たすことになる。

多種多様なツールから生成される多種多様なソフトウェアの成果物間のトレーサビリティにオープンなプラットフォームを提供するのが TERAS である。

2 トレーサビリティとは

前述したようにソフトウェア開発におけるトレーサビリティとは、『ソフトウェア開発の成果物である文書間において追跡が可能である』ことである。具体的にソフトウェア開発におけるトレーサビリティを示すために、セキュリティ評価基準や機能安全規格にあるトレーサビリティを解説する。

情報技術セキュリティ評価基準 ISO/IEC15408（コモンクライテリア）[1] には、保証クラス、ファミリ、コンポーネント、EAL（Evaluation Assurance Level）がある。

保証要件の最も抽象的なセットはクラスと呼ばれる。各クラスには、保証ファミリが含まれ、保証ファミリには、保証コンポーネントが含まれ、保証コンポーネントには保証エレメントが含まれる。クラスとファミリは、保証要件を分類するために使われ、コンポーネントは PP（Protection Profile）/ST（Security Target）に保証要件を特定するために使われる。コモンクライテリアでトレーサビリティは ADV（開発）クラスの RCR（Representation CoRrespondence）ファミリに属する概念となる。コンポーネントと EAL との関係を表 1 に示す。

表 1　ISO/IEC15408 開発者向けセキュリティ評価に関する要件

保証クラス	保証ファミリ	評価保証レベルに基づく保証コンポーネント						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
開発	ADV_FSP	1	1	1	2	3	3	4
	ADV_HLD		1	2	2	3	4	5
	ADV_IMP				1	2	3	3
	ADV_INT					1	2	3
	ADV_LLD				1	1	2	2
	ADV_RCR	1	1	1	1	2	2	3
	ADV_SPM				1	3	3	3

【脚注】

※1　Tool Environment for Reliable and Accountable Software の略で、2011 年 4 月に設立された一般社団法人 TERAS が提供するオープントレーサビリティツールプラットフォームである

表２　トレーサビリティに関するアクションエレメント

レベル	開発者アクションエレメント	証拠の内容・提示エレメント	評価者アクションエレメント
ADV_RCR.1 非形式的対応の実証	開発者は、提供する TSF 表現の隣接するすべての組の間の対応の分析を提供しなければならない。	提供された TSF 表現の隣接する各々の組に対し、分析は、より抽象度の高い TSF 表現のすべての関連するセキュリティ機能性が、抽象度の低い TSF 表現に、正確かつ完全に詳細化されていることを実証しなければならない。	評価者は、提供された情報が、証拠の内容・提示に対するすべての要件を満たしていることを確認しなければならない。
ADV_RCR.2 準形式的対応の実証	同上	＋ 提供された TSF 表現の隣接する各々の組に対し、どちらの表現も最低限、準形式的である部分に対しは、表現のそれらの部分の間の対応の実証は、準形式的でなければならない。	同上
ADV_RCR.3 形式的対応の実証	＋ 対応する表現がともに形式的である部分については、開発者は、対応を証明しなければならない。	＋ 提供された TSF 表現の隣接する各々の組に対し、どちらの表現も形式的である部分で、それらの部分の間での対応の証明は、形式的でなければならない。	評価者は、形式的な分析を選択的に検証することによって、対応の証明の正確さを決定しなければならない。

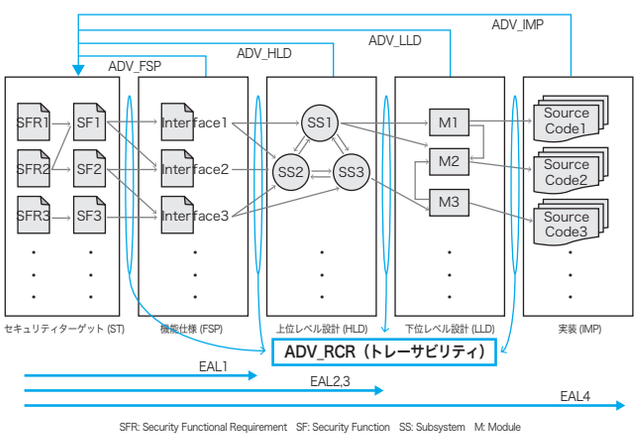

図１　セキュリティとトレーサビリティ

保証ファミリ※2は、レベル付けが行われ、コンポーネントにレベルを付ける方法の根拠が示される。この根拠は、適用範囲、深さ、及び／または厳格性の観点からである。トレーサビリティに関係する ADV_RCR は３レベルである（表２）。

開発者のアクションは、レベル１では非形式的で、レベル２では準形式的で、レベル３では形式的な TSF※3 に関する成果物間の対応関係を提供しなければならない。ADV_RCR と保証ファミリおよび EAL の関係を図１に示す [2]。

機能安全規格である ISO20262、車載ソフトウェア開発プロセスモデルである AutomotiveSPICE、IEEE Standard for Software Verification and Validation、そして IPA/SEC の ESPR（Embedded Software Process Reference）におけるトレーサビリティに関する記述を表３に示す。

表３　規格・プロセスとトレーサビリティ

文書	記述
ISO 26262：2011(E) Functional safety Part2 Management of functional safety Annex B:Examples for evaluating a safety culture	- 貧弱な安全文化の指標例：説明責任（アカウンタビリティ）がトレーサブルではない。 - 良好な安全文化の指標例：機能安全に関わる意思決定の責任がトレーサブルであることを保証するプロセスである。
AutomotiveSPICE ENG.4 ソフトウェア要件分析　Level 1	参照元の要件とソフトウエア要件との間でトレーサビリティを作成しているか。
IEEE Standard for Software Verification and Validation（IEEE Std 1012-2004）5.4.1 アクティビティ：コンセプト V&V（プロセス：開発）	・獲得要求とシステム要件とのトレーサビリティを検証する。 ・システム要件とソフトウエア要求とのトレーサビリティを開始する。
ESPR SYP2 システム・アーキテクチャ設計 SYP2.2 システム・アーキテクチャ設計の確認 2.2.1 システム・アーキテクチャ設計書の内部確認	・システムを構成する機能ブロックの分割が適切であり、システム要求で求められる事項が現実可能かどうか（トレーサビリティの確認）。 ・システム要求やテスト仕様との対応（トレーサビリティ）が取れているか。

【脚注】

※２　ADV クラスの保証ファミリ構成：
ADV_FSP：機能仕様に関する要件
ADV_HLD：上位レベル設計に関する要件
ADV_LLD：下位レベル設計に関する要件
ADV_IMP：実装表現（ソースコード）に関する要件
ADV_RCR：追跡性（トレーサビリティ）に関する要件
ADV_SPM：セキュリティ方針モデリングに関する要件
ADV_INT：TSF 内部構造に関する要件

※３　TSF：TOE（Target Of Evaluation）Security Functionality の略で、セキュリティ機能要件（SFR）が正しき動作するために必要なすべてのサブシステムやモジュールである

3 TERAS とは

トレーサビリティ対象となるソフトウェアの成果物は多種多様な文書から構成される。「要求定義書」、「機能仕様書」、「基本設計書」、「詳細設計書」、「ソースコード」、「テスト仕様書」、「テストケース」、そして「テスト成績書」などがある。さらに、「要求定義書」、「機能仕様書」、「基本設計書」、「詳細設計書」などでは、「ユースケース図」、「クラス図」、「シーケンス図」、「状態遷移図表」、そして「ブロック図」などのモデルが成果物の一部になる。

多種多様なオーサリングツールから生成される多種多様なソフトウェアの成果物間のトレーサビリティにオープンなプラットフォームを提供するのが TERAS である。

3.1. TERAS アーキテクチャ

TERAS はオーサリングツールとトレースリポジトリを分離した構造をもつ（図 2）。

分離することで、ソフトウェア開発における成果物の表現を損なうことなく、トレーサビリティを管理することができる。

ソフトウェア開発の成果物は膨大な量であり、多種多様な文書から構成される。さらにソフトウェア開発は差分／派生開発が多く、複数のバージョンが存在する。こうしたソースコードを含めた成果物の構成管理に構成管理ツール Subversion が広く使われている。こうした従来の既存環境をそのまま活用できるように TERAS では OSLC（Open Services for Lifecycle Collaboration）に対応する（図 4）。

OSLC は、開発ツールの相互運用性を向上するという共通の目標を持つ企業・組織・個人から成るオープンコミュニティである。OSLC では、共通の REST（Representational State Transfer）プロトコルと共通の開発ライフサイクルデータの表現を定義し、異なるツールを相互運用するための仕様を定義している。ソフトウェア開発における開発ライフサイクルを支えるツールの例としては、要件管理ツール、構成管理ツール、そして課題管理ツールなどが含まれる。

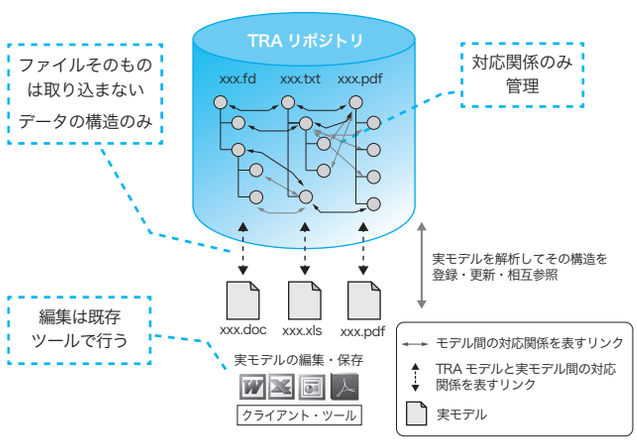

図 2　TERAS トレースリポジトリ

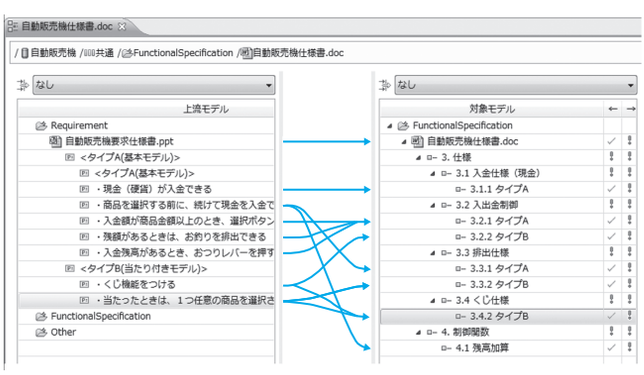

図 3　リンクエディタ

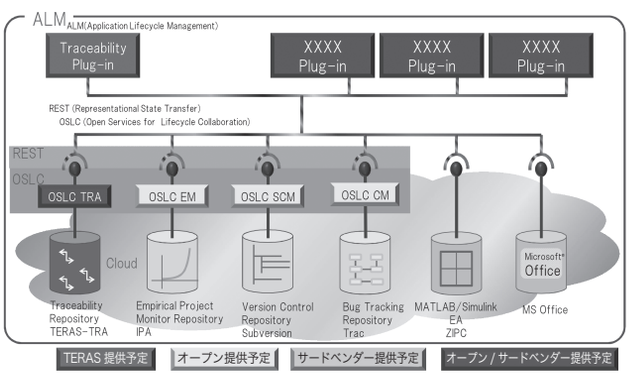

図 4　TERAS プラットフォームアーキテクチャ

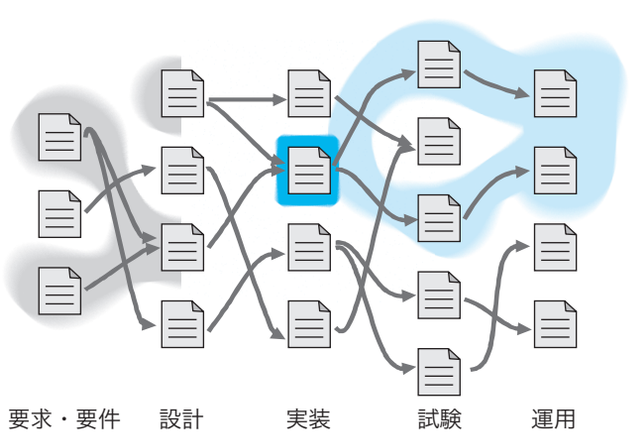

図 5　影響範囲検索

3.2. TERAS 機能

ここからは、TERAS の機能を紹介する。

(1) リンクエディタ

TERAS は、Word や Excel 等のファイルそのものを取り込まず、データ構造のみを取り込み、トレーサビリティを管理する（図 2）。リンクエディタは、取り込んだデータ構造のリンクを編集する（図 3）。

バージョン 3 では、Word、Excel、PowerPoint、PDF、MATLAB/Simulink、Enterprise Architect 、ZIPC、 テキストファイルであれば、ファイルの内部構造を抽出可能である。その他の成果物は、ファイル単位で管理可能である（図 3）。

(2) 影響範囲検索

派生製品の開発や不具合対策時の大きな関心は修正の影響範囲である。あるモジュールを修正した際に影響を受ける範囲はどこまでで、影響範囲に対してどのような修正を行い、どのような試験を行えば良いか。ある不具合によって見直しが必要な製品はどれなのか。TERAS を活用することで、影響範囲をしっかり管理でき、影響範囲検索機能でわかりやすく確認できる（図 5）。

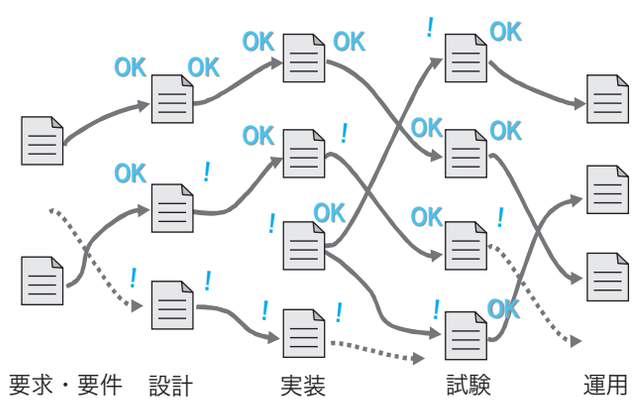

図 6　カバレッジ確認

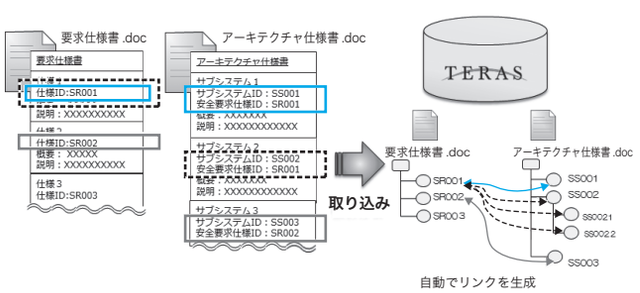

図 7　タグベースリンク

(3) カバレッジ確認

開発プロセスで作成される複数の成果物間の対応関係にモレ・ヌケがないことを確認するのがカバレッジ確認である。

例えば、要件から仕様、設計、実装にトレーサビリティを作成してあれば、要件に対してモレ・ヌケなく検討・実現されているかどうかの確認が容易にできる（図 6）。

(4) タグベースリンク（自動リンク）

トレーサビリティ対象文書にあらかじめ“トレースタグ（ID）”が記載されていれば、そのトレースタグを解析し、対応する項目間を自動でリンクする機能である（図 7）。

(5) バージョン管理ツール連携（Subversion）

TERAS はトレーサビリティ管理機能とバージョン管理ツール（Subversion）を連携させ、バージョン管理されたファイルでトレーサビリティ管理ができる。

日々の細かな試行錯誤中はバージョン管理ツールだけで成果物を管理し、ある程度成果物の内容が固まってきた段階（例えば、レビュー時やリリース毎）で、TERAS を活用してトレーサビリティ情報を管理する（図 8）。

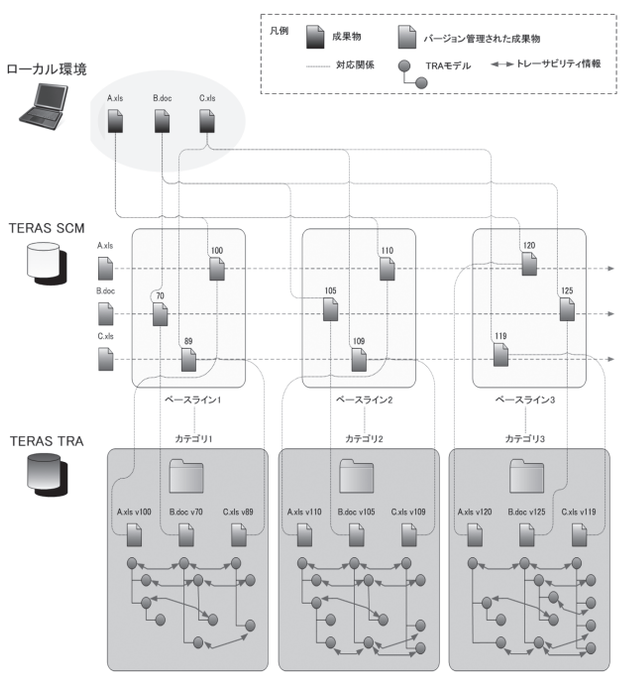

図 8 バージョン管理ツール連携

(6) Trac、Redmine 連携

バージョン 3 では、新たに Trac、Redmine と連携する。これらのツールと連携することで、Trac、Redmine が管理しているチケットの情報（タスク、要求、バグ等）をトレーサビリティ対象として取り込むことができるようになる。

TERAS のカバレッジ確認結果や影響範囲検索の結果は、実施すべきタスクや変更要求となるため、チケットとして起票して管理しておくとモレ・ヌケなく作業が実施できるようになる。この支援機能として、TERAS からチケットを起票できる機能も用意している（図 9）。

4 TERAS 課題

バリエーション管理とトレーサビリティリンクに関する課題がある。トレーサビリティリンクとは成果物間の対応を示すものである。しかしながら、現在のトレーサビリティリンクではリンクの AND 条件と OR 条件がないため、バリエーションを考慮した場合にトレーサビリティリンクが十分条件か必要条件かを明らかにすることができない課題がある。

具体的な例を図 10 に示す。図 10 の左側のシステム 1 における要件と設計の関係において、要件 A を満足するために、設計 P と Q を採用している。また、設計 Q は、要件 B に依存している。図 10 の右側のシステム 2 では、システム 1 から、要件 B が要件 C に代わったため、設計 Q の代わりに設計 R を採用している。自動車を例にすると、要件 A は自動変速（AT）で共通であるが、要件 B は前輪駆動（FF）で、要件 C は後輪駆動（FR）となる。

システム 1 とシステム 2 の要件と設計の関係を単純に重ね合わせると、要件 A を満足するために、設計 P、Q、R をどの組み合わせで採用すればよいかがわからない（図 11）。

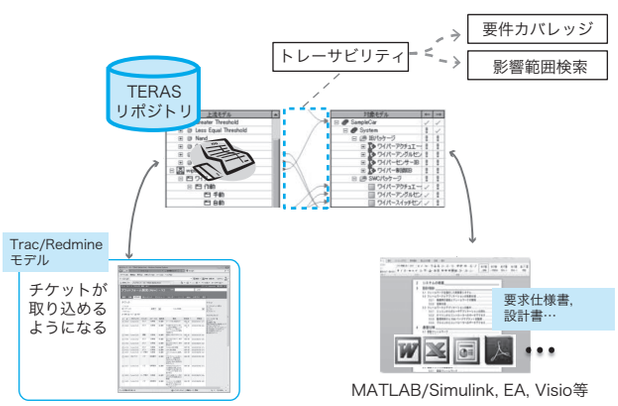

図 9　Trac、Redmine 連携

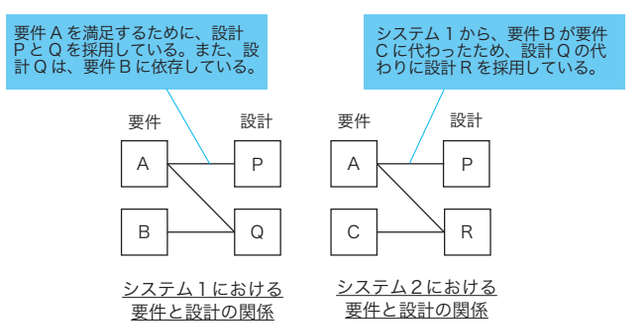

図 10　バリエーションとトレーサビリティリンク課題（1）

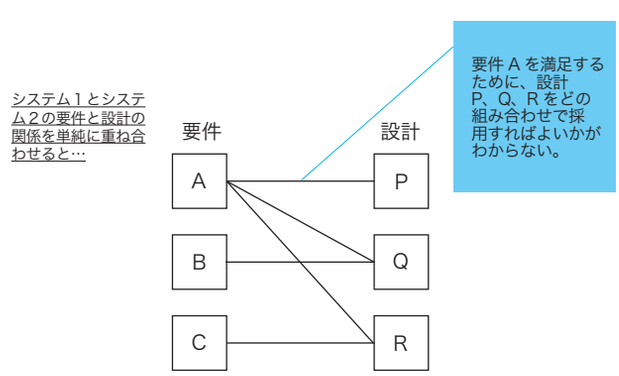

図 11　バリエーションとトレーサビリティリンク課題（2）

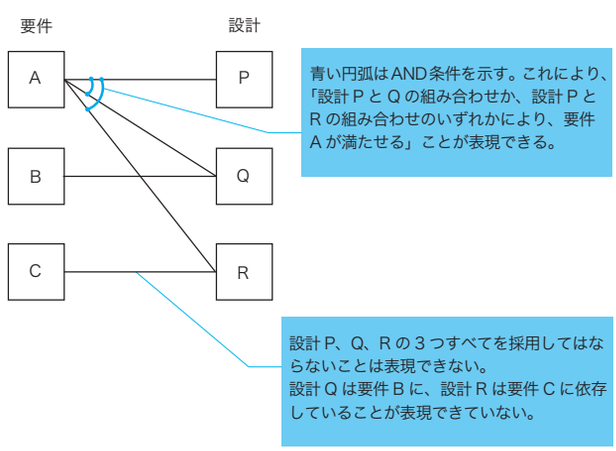

図 12　バリエーションとトレーサビリティリンク課題（3）

そこで、図 12 に示す円弧の AND 条件記号をトレーサビリティリンクに付与する。これにより、「設計 P と Q の組み合わせか、設計 P と R の組み合わせのいずれかにより、要件 A が満たせる」ことが表現できる。しかしながら、設計 P、Q、R の 3 つすべてを採用してはならないことは表現できていない。設計 Q は要件 B に、設計 R は要件 C に依存していることが表現できていない。

このようにシステム 1 とシステム 2 の要件と設計の関係を同時に管理する場合にどうするかといった点が課題である。

バリエーション管理をトレーサビリティリンク上で表現するのではなく、フィーチャモデル上で表現する、または、成果物側に C 言語の #if のように表現し、トレーサビリティエディタがこのようなバリエーション情報を読み取り、トレーサビリティリンクを表示する方法も考えられる。

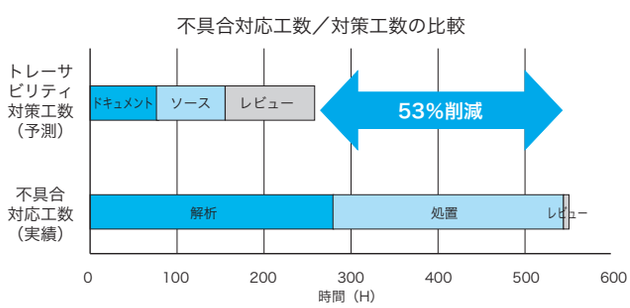

図 13　トレーサビリティによる工数削減効果

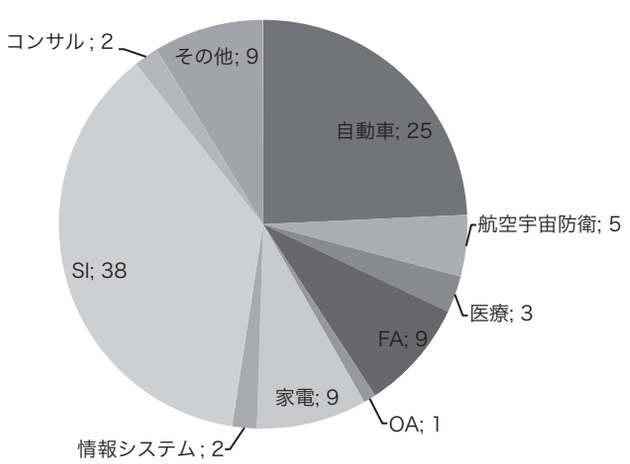

図 14　TERAS 実証評価企業分類

5 ロードマップ・活動報告

また、ソフトウェア開発におけるコスト問題対策でも「トレーサビリティ」が期待されている。エンタプライズ系ソフトウェアでは、新規開発は約 26% に対し、差分／派生開発は約 73% である [3]。組込み系ソフトウェアでは、新規開発は約 42% に対し、差分／派生開発は約 54% である [3]。差分／派生開発では仕様変更の正確な影響分析ができないと、品質、コスト、そして納期が悪化する。正確な影響分析を行うためには「トレーサビリティ」が重要である。

以下の 2 つのソフトウェア開発における不具合データを調査対象としたトレーサビリティ確保におけるソフト開発データからの効果検証が行われた [4]。

- 一般組込み機器製品に搭載される通信ソフトウェア
- 車両搭載用通信プロトコルスタックソフトウェア

不具合対応工数／対策工数の比較では 10% から 53% の工数削減効果があると報告されている（図 13）。

一般社団法人 TERAS では、オープントレーサビリティツールプラットフォーム TERAS バージョン 2 の実証評価の参加を呼び掛けています。2013 年 12 月時点での参加企業は 103 で、図 14 のような分類となっています。

6 終わりに

2011 年から開発を開始し、実証評価会員からの要望を取り入れながら進めてきたオープントレーサビリティツールプラットフォーム TERAS のバージョン 3 を、2014 年 4 月にリリースする。

また、2014 年 3 月 12 日に平成 25 年度 TERAS 成果報告会が開催される予定である。

【参考文献】

[1]「情報技術セキュリティ評価のためのコモンクライテリアパート 3 : セキュリティ保証要件」平成 17 年 12 月　翻訳第 1.0 版 IPA/ セキュリティセンター

[2]「日立認証局システム Enterprise Certificate Server Set による ISO/IEC 15408 認証取得について」2004 年 10 月 29 日　株式会社日立製作所 ソフトウェア事業部 セキュリティ対応センタ 栗田 博司

[3]「ソフトウェア産業の実態把握に関する調査　調査報告書―速報版―」2012 年 4 月 27 日　IPA/SEC

[4]「トレーサビリティ確保におけるソフト開発データからの効果検証実施報告書」2013 年 2 月　IPA/SEC