

トレーサビリティの定義・効果・課題 に関する考察

宇田川 佳久[†]

トレーサビリティは多くの業種で採用されている品質管理の基本である。ソフトウェアシステム開発の品質管理も例外ではなく、ウォーターフォールモデルが確立した1970年代以来、多くのトレーサビリティに関する研究が行われてきた。

本稿は、流通・小売業、物流業、製造業、ソフトウェア関連業におけるトレーサビリティの特徴を比較し、ソフトウェア関連業におけるトレーサビリティには、管理対象物の種類が多い、物理的な形状がない、変更が発生する、利用者が少ないといった特質があることを論じる。

次に、トレーサビリティに係るソフトウェアシステム開発に関する規格・標準の調査結果を示し、主な規格・標準におけるトレーサビリティの定義と背景について概観する。続いて、ソフトウェアシステム開発の各工程における利用者、効果と表現手段の観点からトレーサビリティの枠組みを提案する。最後に、今後の研究の方向性について展望する。

Considerations to definition, effectiveness and issues of traceability

Yoshihisa Udagawa[†]

Traceability has been widely accepted as the basis of quality management and has been adopted in many industries. Software system industry is no exception. Many technological advances concerning traceability have been made since the 1970s when the waterfall model dominated.

In this paper, we observe characteristics of traceability in several industries and conclude that traceability in software system industry potentially has adverse factors such as various kinds of work products, no physical shape, a small number of users, etc.

After surveying public standards concerning software system engineering, we summarize the definitions of traceability in major ones. We also propose a traceability framework focusing on users, effectiveness and representation of work products in each process of software system development. Finally, we make perspectives of the research area in the near future.

1. はじめに

トレーサビリティ (traceability: 追跡可能性) は、多くの産業分野において、品質管理の基本的な手段として採用されている。ソフトウェアシステム開発も例外ではなく、1970年代には重要性が認識され、これまでに、多くの研究、開発規格・標準の制定、および支援システムの開発が行われてきた。

大規模ソフトウェアシステムの開発モデルについては、1970年に米国の W.W.Royce氏が、ソフトウェアの作成から廃棄までのライフサイクル・プロセスの概念を提唱した論文が最初と考えられている²⁰⁾。このモデルは、後にウォーターフォールモデルと呼ばれ、今日でも国内の業務向けソフトウェアシステム開発の大多数がウォーターフォールモデルに基づいて開発されているとの報告がある²²⁾。

ウォーターフォールモデルは、テスト工程がプロジェクト後半に設定されているため、上流工程に欠陥があってもプロジェクトの終盤まで発見できないという問題があり、この問題を軽減するためにプロトタイプモデルやスパイラルモデルが開発されている。ウォーターフォールモデル、プロトタイプモデル、スパイラルモデルに共通する考え方として、ある工程の成果物を次工程の入力とすることがある。オブジェクト指向開発モデルにおいても、モデリング、分析、設計、実装、テストという工程があり、前の工程の成果物は次の工程に引き継がれる。すなわち、いずれのモデルにおいても成果物の連鎖によって、最終成果物であるシステムを構築してゆくという点において共通している。

トレーサビリティは、成果物間の関連、または関連を管理する能力と定義され、多くのソフトウェアシステム開発モデルに共通する基本的な品質管理手法であると考えられる。ただし、品質を特に重視するケースを除けば、一般のソフトウェアシステム開発プロジェクトでは、トレーサビリティの管理が十分に定着しているとは言い難い状況にある。主な理由としては、トレーサビリティの管理コストに対する効果が十分に認識されていないことがあげられる^{1), 19), 21)}。

本稿は、トレーサビリティの導入が進められている分野、システム開発の規格・標準におけるトレーサビリティの定義と効果について調査した結果を踏まえ、ソフトウェアシステム開発においてトレーサビリティに期待される効果と今後の研究の方向性について論じている。

本稿の構成は次の通り。2章では、各業種でのトレーサビリティの取組みについての調査結果に基づいて、ソフトウェアシステム開発におけるトレーサビリティの特質について論じる。3章では、トレーサビリティについて扱っている開発標準や規格について述べている。また、トレーサビリティの定義の相違について比較し、定義の背景

[†] 三菱電機インフォメーションシステムズ(株)
Mitsubishi Electric Information Systems Corp.

について考察する。4章では、トレーサビリティに関する規格・標準、研究論文を中心に調査した結果を総合し、トレーサビリティの枠組みについて論ずる。5章では、今後のトレーサビリティの研究の方向性を述べ、システム開発の全工程に渡るトレーサビリティの枠組みを提案する。

表 1. 業種別のトレーサビリティ管理の特質
Table 1. Characteristics of traceability for industries

業種	管理対象	物理実体	変更	利用者
流通・小売業	食品、商品	有り	有り	消費者、生産者、加工者、流通、小売業者
物流業	食品、商品、製品、部品	有り	なし	荷主、物流業者
製造業	製品、部品	有り	有り	製造ラインメンバー
ソフトウェア関連業	要件・設計項目、ソースプログラム、実行プログラム、テストケース	なし	有り	プロジェクトメンバー

2. 各業種でのトレーサビリティの取組み

この節では、流通・小売業、物流業、製造業、ソフトウェア関連業でのトレーサビリティに対する取組みを比較し、ソフトウェアシステム開発におけるトレーサビリティ管理の特質を論じる。

2.1 流通・小売業での取組み

食品・商品などの、いわゆる消費財の安全性が消費者から注目されており、品質管理の一環としてトレーサビリティが導入されている⁷⁾。主な消費財としては、食品、化粧品、医薬品、家電がある。

トレーサビリティ管理の効果としては消費財の生産、加工などの履歴情報を追跡、公開することにより消費者からの信頼を高めることである。また、品質に不具合があった場合でも迅速かつ網羅的な対応を可能にすることで、影響を最小限に止める効果が期待されている。

一般に、消費財は生産、加工、運送、流通に関わることから、消費財のトレーサビリティを管理するためには、関係するすべての企業間で情報を交換することが前提である。このために、企業横断的な情報の標準化が推進されている。消費財のトレーサビリティに関する情報としては、生産地、生産者名、収穫日、入荷日、出荷日、加工場所、加工者名、加工日、使用資材名、使用量、賞味期限などがある。これらの消費

財には、物理的な実体があることから、個体識別のために、上記の情報を記憶したバーコードや IC タグ (RFID: Radio Frequency ID) が一般的に使われている。

2.2 物流業での取組み

物流業においても、荷貨物を確実に受荷主に届けるため、また、輸配送情報を適時に提供することで荷主からの信頼を向上するために貨物追跡システム (Cargo/Freight tracking system) が導入されている⁸⁾。

貨物追跡システムは、トレーサビリティ管理の一形態と考えられるものである。物流は、特定の企業グループで完結する傾向があり、企業グループごとに固有の貨物追跡システムが開発されているケースが多い。

貨物追跡システムでは、物の加工を伴わないことが特徴である。管理対象となる主なデータ項目は、入荷先、入荷日、保管場所、保管者名、出荷先、出荷日、車両番号、運転手名、保存状態などである。管理対象に物理的な実体があることから、個体識別のために、上記の情報を記憶したバーコードや IC タグが一般的に使われている。

2.3 製造業での取組み

製造業では不良品・故障の原因追究などの品質管理、リコール対応、安全管理などの目的で、製品や部品の個別管理への努力が行われてきた。大量生産工業製品におけるロット管理が代表的なものである。

一方、個別生産工業製品については、生産履歴、検査履歴、部品の流通履歴などの管理を強化するためにトレーサビリティ管理の導入が進められている。データの追加・変更の容易さ、改ざん防止、一元管理などの観点から、IC タグを使った取組みも進められている²³⁾。トレーサビリティの主な機能は、下記の2つである。

トレースバック：製品がどの原材料 (部品) から構成されているかを追跡する機能

トレースフォワード：特定の原材料 (部品) を使った製品を追跡する機能

2.4 ソフトウェア関連業での取組み

ソフトウェア関連業では、開発・運用工程における成果物間の関連を追跡するためにトレーサビリティを導入している。対象とする成果物は、要求定義書、設計書、ソースプログラム、テストケース、テスト結果などである。

トレーサビリティの効果としては、変更の影響度合いの分析 (impact analysis)、ある成果物の上流工程の成果物に対する適合性検証 (requirements validation, compliance verification)、ソースコードから設計項目への追跡 (traceback of code)、回帰テスト項目の選択 (regression test selection) などがある⁷⁾。また、CMMI⁴⁾、Caltran の ITS システム構築ガイドブック³⁾をはじめとする開発標準規格では、要求と設計成果物、設計成果物とプログラム、プログラムとテストなどとの間での双方向のトレーサビリティを求めている。

2.5 各業界におけるトレーサビリティに関する考察

表 1 は、各業種におけるトレーサビリティ管理の特徴を、管理対象の種類、物理的

な実体の有無，変更の有無，利用者の観点からまとめたものである。

ソフトウェア関連業における管理対象は，要求・設計項目，ソースプログラム，実行プログラム，テストケースなどであり，他の業種と比較して管理対象が多様である。一般に，成果物は電子データとして表現されていることから，物理的な実体がないという特質がある。従って，ソフトウェア関連業では，管理対象の識別にバーコードやIC タグの適用が困難であることから，対象物の識別のために，独自の手法が開発されている。

物流業以外は，管理対象の変更が発生する。管理対象に変更が発生する場合は，管理対象の識別方法に工夫が必要であり，トレーサビリティ管理を複雑なものにする傾向がある。特に，ソフトウェア関連業における成果物は，電子データであることから，変更が容易であり，変更の痕跡が残り難いことも他の業種には見られない特質である。

利用者に関しては，流通・小売業と物流業が，一般消費者（荷主）と業界の担当者であり，不特定多数（数万人規模と推測）を対象にしていると考えられる。一方，製造業における利用者は，製造ラインのメンバーや品質管理者であり数百人規模と推測される。ソフトウェア関連業における利用者は，プロジェクトメンバーであり数十人規模と考えられる。製造業およびソフトウェア関連業では，他の業種と比較して利用者が少ない。これは，トレーサビリティ管理の効果を楽しむ人数が少なく，トレーサビリティ管理への投資規模が制約されることを意味している。

以上より，ソフトウェア関連業におけるトレーサビリティ管理は，下記の観点から，他の業種と比較して最も実現が困難なものであると結論する。

- (1) 管理対象が多様である。
- (2) 物理的な実体がない。
- (3) 管理対象の変更が容易である。
- (4) 利用者数が少ない。

3. 規格・標準におけるトレーサビリティ

3.1 規格と標準

表 2 はトレーサビリティに関連する記述があるソフトウェアシステム開発に関する規格および標準を年代とカテゴリーに分けて一覧したものである。以下，各規格および標準の概要を述べる。

- (1) IEEE Std 610.12:1990 は，ソフトウェア開発における用語を定めた規格である¹¹⁾。詳細は後述するが，この規格では，トレーサビリティを要素間の関連(Relationship)として定義している。この定義は，現在でも多くの論文や技術資料で引用されている。

1990 - 94年	1995 - 99年	2000 - 04年	2005 - 09年
IEEE Std 610.12:1990	IEEE-Std 1362:1998		IEEE Std 1220:2005
ISO 9001:1987	ISO 9000-3:1997(SW)	ISO 9001:2000 ISO/IEC 90003:2004	ISO 9001:2008
CMM V1:1991		CMMI V1:2000	CMMI DEV V1.2:2006
	ISO/IEC 12207:1995	ISO/IEC 15288:2002 ISO 12207:Amd1,Amd2	ISO/IEC 15504:2008 ISO/IEC 15288:2008 ISO/IEC 12207:2008
	NASA SE HandBook:1995	NASA SE Proc.:2003	NASA SE HandBook:2008
		CalTrans SEG ITS V1:2005	CalTrans SEG ITS V2:2007
ISO10303-1 STEP IS:1994		ISO10303 STEP AP233 CD:2000	ISO10303 STEP AP233 DIS:2009
			OMG SysML V1.1:2008
	INCOSE SE HandBook V1:1998	INCOSE SE HandBook V2:2000	INCOSE SE HandBook V3:2007

表 2. トレーサビリティに関連する主な規格と標準

Table 2. Standards and handbooks concerning traceability

- (2) IEEE-STD 1362:1998 は，ソフトウェアシステムの利用者の観点から，システムの操作ドキュメント（Concept of operations document）の記述方法を定義している規格である¹⁰⁾。詳細は後述するが，この規格では，成果物の階層とドキュメントの観点からトレーサビリティを定義している。
- (3) IEEE Std 1220:2005 は，高品質のシステム開発を目的としたライフサイクルに渡る統合化されて管理プロセスを定義している⁹⁾。これを実現するために，システム要素，設計，製造，テスト，運用，保守，トレーニング，廃棄に関する情報の統合的な管理を規格化している。
- (4) ISO 9001 は，品質マネジメントシステムに関する規格であり，1987年の制定後，改定を重ね，現時点の最新版は2008年度版である。ISO/IEC 90003:2004 は，ISO 9001:2000 をソフトウェア製品に適用するためのガイドラインとして作成されたものである¹³⁾。詳細は後述するが，ISO 9000 シリーズでは，ハードウェア製品およびソフトウェア製品の品質マネジメントを対象としていることから，トレーサビリティに関しても包括的に定義していると読み取れる。

- (5) CMMI DEV V1.2:2006 は米国カーネギーメロン大学の SEI (ソフトウェア工学研究所: Software Engineering Institute) がソフトウェアの開発能力を客観的に評価するための指標を定めている⁴⁾。
- (6) ISO/IEC 15504:2003 は、SPICE (Software Process Improvement and Capability dEtermination) と呼称されているソフトウェア開発を中心とした工程に関する指標の測定、評価、改善に関する規格である。この規格では、プロセス定義の参照モデルとして ISO/IEC 15288, ISO/IEC 12207 などを利用することができる。
- (7) ISO/IEC 15288:2008 は、ハードウェアとソフトウェアから構成されるシステム開発に関する、発注側と受注側の双方で相互に理解できるシステム・ライフサイクル・プロセスの共通枠組みと用語を規定した規格である。ソフトウェアのみを対象とする場合は、ISO/IEC 12207:2008 を適用する。
- (8) ISO/IEC 12207 は、ソフトウェア開発の発注側と受注側の双方で相互に理解できるソフトウェア・ライフサイクル・プロセスの共通枠組みと用語を規定している。JIS X 0160:1996 (ソフトウェア・ライフサイクル・プロセス) は、ISO/IEC 12207:1995 を日本語に翻訳したものである。
- (9) NASA (米国航空宇宙局: National Aeronautics and Space Administration) は、宇宙開発に関わる計画を担当する米国の連邦機関である。安全性の高いシステムを構築する必要があることから、独自のシステムエンジニアリングハンドブック (NASA Systems Engineering Handbook) を作成し、運用している¹⁵⁾。
- (10) CalTrans (米国カリフォルニア州交通局: California Department of Transportation) は、カリフォルニア州のハイウェイの管理を行う公的機関である。CalTrans は、ITS の開発に多数の企業が円滑に参画できるように、独自の ITS システム構築ガイドブック (System Engineering Guidebook for ITS Version2) を公開し、運用している³⁾。
- (11) ISO 10303 STEP は、CAD/CAM システムにおける製品データ交換の国際規格であり、STEP 全体としては数百の分冊から構成されている。AP233 (応用プロトコル 233) は、システム開発全般に関わるデータ (要求、要求トレーサビリティ、システムの構造、プロジェクト管理データ、スケジュール、リスク管理データ、問題管理データなど) を規格の対象としている⁶⁾。
- (12) SysML は、OMG (Object Management Group) が作成した、ハードウェア、ソフトウェア、情報、人などを含むシステムを定義、分析、検証するためのモデリング言語である¹⁶⁾。
- (13) INCOSE (International Council on Systems Engineering) のシステム構築ハンドブック (INCOSE Systems Engineering Handbook Version 3.1) は、IEEE Std 1220, ISO/IEC 15288, CMMI DEVなどをベースとしたシステム構築のガイドラインについて標準を定めている¹²⁾。

表 3. 規格・標準におけるトレーサビリティの定義

Table 3. Definitions of traceability in standards and handbooks

No	規格・標準	定義内容	試訳(日本語)
1	IEEE Std 610.12	The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another.	開発工程における複数の成果物間に確立された関連の程度であり、特に、前工程と後工程、または、主従関係がある成果物に関するものである
2	IEEE-Std 1362	The identification and documentation of derivation paths (upward) and allocation or flowdown paths (downward) of work products in the work product hierarchy.	成果物の階層において、成果物の導出経路(上方向)、配置または下流への経路(下方向)に関する識別と文書化したものである
3	CMMI DEV V1.2 NASA Systems Engineering Handbook	A discernable association among two or more logical entities such as requirements, system elements, verifications, or tasks.	システム要素、検証、またはタスクなど、二つ以上の論理的実体間の識別可能な関連
4	ISO10303 STEP AP233	Tracing relationships defining traceability from requirements to elements in the functional or physical architecture.	機能または物理的アーキテクチャにおいて要件から要素へのトレーサビリティを定義するトレーシング関連(をサポートする)。
5	CalTrans: System Engineering Guidebook for ITS- An Introduction	It is important to be able to relate the items in one step with those in another. The relationship between items is called traceability.	ある工程と他の工程における項目を関連付けることは重要である。項目間の関連は、トレーサビリティと呼ばれる。
6	ISO 9001	The ability to trace the history, application or location of that which is under consideration.	考慮の対象となっているものの履歴、適用又は所在を追跡できる能力

3.2 トレーサビリティ定義のバリエーション

表 3 は、トレーサビリティに関連する主な規格・標準におけるトレーサビリティの定義をまとめたものである。以下、定義の概要と背景について考察する。

IEEE Std 610 では、ソフトウェア開発の観点から、トレーサビリティを「開発工程における複数の成果物間に確立された関連 (Relationship) の程度であり、特に、前工程と後工程、または、主従関係がある成果物に関するものである」と定義している。この定義は、現在でも、多くの論文や技術資料で引用されている。

IEEE-Std 1362 は、ドキュメンテーションの観点から、トレーサビリティを「成果物の階層において、成果物の導出経路 (Path) および、成果物の配置状況の識別と文書化である」と定義している。

CMMI DEV V1.2 ではトレーサビリティを「システム要素、検証、またはタスクな

ど二つ以上の論理的実体間の識別可能な関連 (Association)」と定義している。現時点で最新である CMMI DEV V1.2 では、要求管理、プロジェクト計画策定、構成管理など 22 個のプロセス領域で構成されており、10 個のプロセス領域でトレーサビリティの管理を必須と定めている。

NASA システム構築ハンドブックではトレーサビリティを、CMMI DEV V1.2 同様の文面で定義している。CMMI の発行年が 2006 年、NASA システム構築ハンドブックの発行年が 2008 年であることから、CMMI DEV V1.2 の定義を参照しているものと推察できる。

ISO10303 STEP AP233 では、システムエンジニアリング全般に関するデータ交換の観点から「TRACING_RELATIONSHIP」というエンティティによってトレーサビリティを表現している。「TRACING_RELATIONSHIP」は 2 つの要素間の関連として定義されており、任意の分類「CLASS_OF_REQUIREMENT_TRACING」を関連付けることができる。

CalTrans の ITS システム構築ガイドブックでは、トレーサビリティによって得られる効果や、トレース可能な要素 (購入業者、要求、設計、テストケースなど) について記述しているが、トレーサビリティの直接的な定義は見当たらない³⁾。ただし、システムガイドの手引書²⁾では「(ある工程と他の工程における) 項目間の関連はトレーサビリティ呼ばれる」と定義している。

ISO 9001 では、品質マネジメントシステムの管理の観点から、トレーサビリティを「考慮の対象となっているものの履歴、適用又は所在を追跡できる能力」と定義している。この定義は包括的なものである。実際、能力を構成する要素には、トレーサビリティ情報を取得、登録、検索、報告、通知するための情報管理システムの機能 (性能) としての側面と、その情報管理システムの運用者のヒューマンスキルに関する側面が含まれる。したがって、トレーサビリティ管理能力の有無の判定には複数の側面からの評価が必要になる。

なお、ISO 9000 におけるトレーサビリティは、ソフトウェア部品が製品のどこに、どのように組み込まれているかを追跡できることを要求している。この要求は、ハードウェア製品などの部品が製品のどこに、どのように組み込まれているかを管理することと同様のレベルの要求であると解釈されている。

以上の調査結果をまとめると、表 1 における「ソフトウェア関連業」に関連するソフトウェアシステムの規格・標準では、トレーサビリティを「要素間の関連」と定義する傾向が伺える。一方、表 1 における「ソフトウェア関連業」以外の業種では、ISO 9000 の影響から「追跡できる能力」とする傾向があるものと考えられる。

用語としてのトレーサビリティの定義は、本質ではないと考えるが、定義の違いが対象や前提、期待する効果に違いをもたらすことがあるので、トレーサビリティに関しても定義にバリエーションがあることを考慮する必要があると考える。

4. トレーサビリティの分類, 工程, 利用者, 効果

表 1 に示したように、ソフトウェア関連業では、管理対象が多様である、管理対象が電子データとして表現されるために物理的な形状を有しない、変更が容易、などの特質を有する。そのため、ソフトウェア関連業では、ソフトウェアシステム開発と運用に特化したトレーサビリティの枠組みが必要になる。この章では、トレーサビリティに関する規格・標準、研究論文を中心に調査した結果を総合し、各工程でのトレーサビリティの分類、効果、タイプの観点からトレーサビリティの枠組みについて論ずる^{5), 14), 17), 19), 21)}。各工程でのトレーサビリティの分類、効果などをまとめたものを図 1 に示す。

4.1 工程内と工程間

一般に、ひとつの工程では、複数の成果物が作成される。例えば、Web アプリケーションにおける外部設計では、画面設計、機能設計、データベース論理設計、外部システムインターフェース設計などが行われる。本稿では、ひとつの工程内で作成される成果物を構成する項目間のトレーサビリティ^[*1]を垂直トレーサビリティ^[*2]と呼ぶ。工程に跨る成果物間のトレーサビリティを水平トレーサビリティ^[*2]と呼ぶ。

4.2 工程と利用者

利用者の観点からトレーサビリティを分類したものとして、要求定義前トレーサビリティ (Pre-RS: Pre-Requirements Specification) と要求定義後トレーサビリティ (Post-RS: Post-Requirements Specification) がある⁵⁾。一般に、ソフトウェアシステム開発は要求定義の前に、システム化構想、ニーズ効果分析が実施される。この作業は、ソフトウェアシステムの発注者側で実施されるもので、成果物は後続の要求定義に網羅的に引き継がれていることが前提である。一方、要求定義とその後に続く、設計、製作 (プログラミング)、テスト工程は、ソフトウェアシステムの開発者側で実施されるものである。トレーサビリティの利用者が異なる (従って期待する効果が異なる) ことから、要求定義以前のトレーサビリティと要求定義後のトレーサビリティとを区別すべきであるというのが、Gotel 氏らの主張である。

この考え方の延長として、CalTrans の ITS システム構築ガイドブック³⁾では、システムが納入されてからリプレースされるまでの運用保守工程で使う 引渡し後 (Post-Delivery) トレーサビリティを定義しており、本稿でも、この定義を採用する。

[*1] 成果物単位ではなく、成果物を構成する項目レベルでのトレーサビリティを、微細トレーサビリティ (fine-grained traceability) と呼び、成果物単位のトレーサビリティと区別することがある (文献 18))。成果物の項目レベルでのトレーサビリティは、ソフトウェア関連業が扱うシステム開発では一般的なことであり、本稿では、微細トレーサビリティを単にトレーサビリティと呼んでいる。

[*2] 水平、垂直の軸の方向については、逆方向に定義している標準や研究がある。本文では、文献 14) および 17) に従って定義した。

4.2.1 要求定義前トレーサビリティ

要求定義前の工程では、システム化構想、ニーズ効果分析などを実施する。システム化の対象は、ニーズ効果の高いものを選択することから、この工程では、主に、要求の優先度付け、代替要求案の検討が行われる。

これらの作業を確実にかつ効率的に行うために、要求が存在する理由や前提、代替要求案の関連（要求定義前トレーサビリティ）の管理が求められる。要求定義前の工程の成果物は、主に自然言語、図、表、イメージデータで表現されている。

4.2.2 要求定義後トレーサビリティ

要求定義および設計では成果物の詳細化、製作では成果物のプログラミング言語（形式言語）への変換、テストは成果物の統合化という性質を持つ。このことから、トレーサビリティに期待する効果も各工程の特徴に依存して異なる。

(1) 要求定義および設計工程

要求定義および設計工程は成果物の詳細化という性質ある。そのため、ニーズ効果分析と要求定義間、要求定義と外部設計間、外部設計と内部設計間の網羅性の管理、代替案の追跡および変更影響範囲の分析が効果として期待される。

これらの効果を実現するために、要求または設計項目間の導出管理、設計項目の代替案とそれに付随する理由や前提に関する関連（水平トレーサビリティ）の管理が必要になる。これに加えて、各工程における成果物の項目間の依存関係（垂直トレーサビリティ）の管理が求められる。この工程における成果物は、主に自然言語、図、表、イメージデータで表現されている。

(2) 製作

製作では、外部設計書および内部設計書の内容をプログラミング言語に変換する作業が行われる。設計項目が実装されたことを確実にかつ効率的に管理するために、設計項目とソースコード間の関連を管理することが求められる。この工程の成果物は、プログラミング言語、自然言語によるコメントで表現されている。

(3) テスト

テストでは、プログラミング言語によって実装された成果物が、要求定義、外部設計、内部設計における内容と整合しているか否かをテストデータによって検証する作業が行われる。テストが要求および設計項目に対し網羅的に実行されたことを管理するために、設計項目とテストケースおよびソースプログラム間の関連を管理することが求められる。障害発生時には、影響範囲を特定することもテストを確実にかつ効率的に行うために求められる機能である。回帰テストのためのテストケースの選択もトレーサビリティに求められる。この工程の成果物は、主に自然言語（テスト内容）、テストデータ、テスト結果（イメージ、表、ログ）で表現されている。

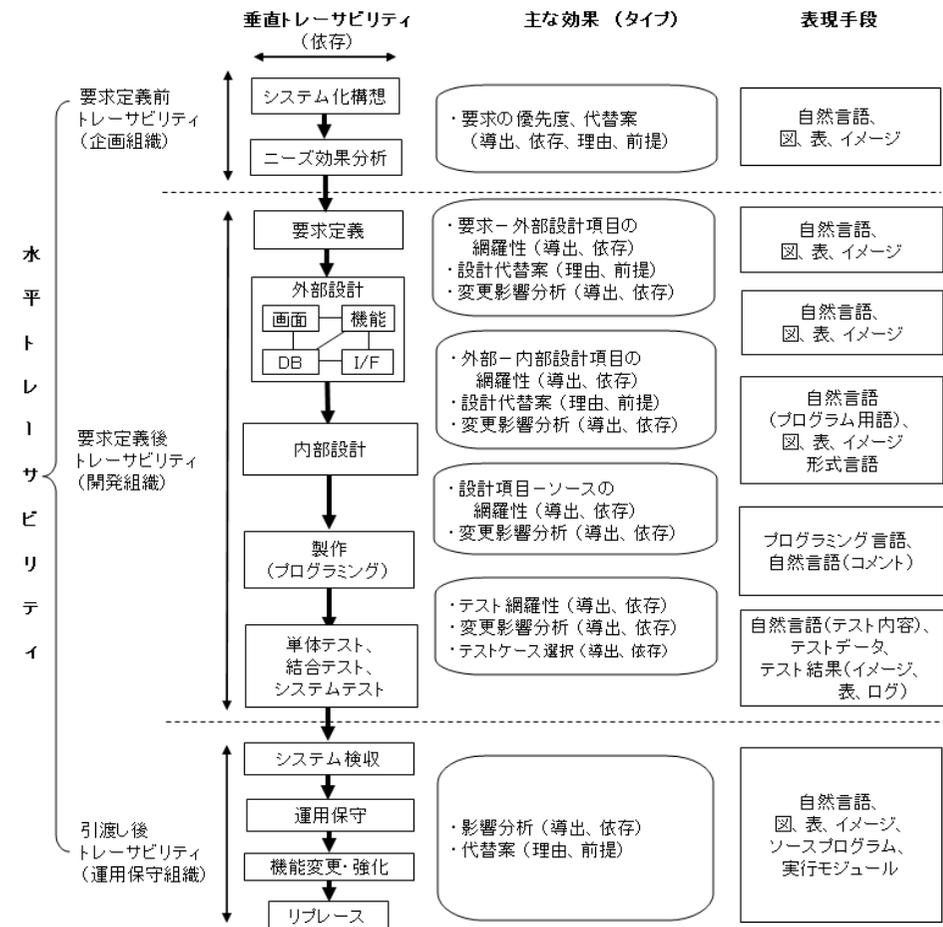


図 1. トレーサビリティの分類, 利用者, 工程, 効果
Figure 1. Classification, user, process and effectiveness of traceability

4.2.3 引渡し後トレーサビリティ

運用保守では、運用保守マニュアル、ソースプログラム、実行モジュールなどが管理対象となる。運用保守マニュアルは、企画組織、開発組織が作成した成果物へのト

レーサビリティが求められる。一般に、運用保守では、システム機能の部分的な変更が発生することから、変更の影響分析や代替案の検討などが行われる。これらの検討作業を確実にかつ効率的に行うために、設計項目の導出管理、設計項目の代替案とそれに付随する理由や前提に関する関連（トレーサビリティ）の管理が必要になる。

5. トレーサビリティ研究の方向性と枠組みの提案

5.1 研究の方向性

トレーサビリティ普及の課題として、コスト対効果が実務レベルにないこと、利用者がプロジェクトメンバー（数十人規模）であることに着目し、今後の研究の方向性について展望する。

(1) コストを低くする

現状、実務レベルでは、トレーサビリティの多くは手動で生成されている。手動による生成は、属人的な判断に依存すること、および多大なコストが発生することからトレーサビリティの普及の妨げになっている²¹⁾。トレーサビリティの半自動、自動生成に関する研究は今後も変わらない重要なものである。

(2) 効果を高める

これまでの研究、規格・標準から、トレーサビリティに期待する効果として、網羅性分析、変更影響分析、存在理由追跡、代替案と前提追跡などが知られている。これらの効果は、管理する関連情報とそれらの情報を活用する機能によって達成されるが、情報と機能の具体的な内容については更なる研究が必要である^{19), 20)}。

(3) 利用者数を実質的に多くする

ひとつのプロジェクトの利用者が数十人規模であっても、複数のプロジェクトに適用すれば実質的な利用者数を増やすことができる。ただし、トレーサビリティに関する共通の枠組み（フレームワーク）が確立していることが前提である。トレーサビリティの枠組みに関しては、アプリケーションや開発方法論などと組み合わせたものが活発に研究されている^{19), 20)}。

ソフトウェアシステム開発は、設計や開発手法の点で類似点も多い。この類似点はISO, IEC を初めとする組織によって規格化が進められている。また、規格にまでは至らなくても、システムの発注部門や開発担当企業内で標準化が進められている。ソフトウェアシステム開発の規格や標準におけるトレーサビリティの扱いは、本稿3章で述べたとおりで、トレーサビリティに関する要求は増加するものと予想する。今後とも、規格や標準におけるトレーサビリティの扱いに注目してゆきたい。

ソフトウェアシステム開発は多様化している。分散開発などに対応した環境ツールの整備と、ツール間のデータ交換方法（例えば、STEP AP233）に関する研究も重要なテーマである。

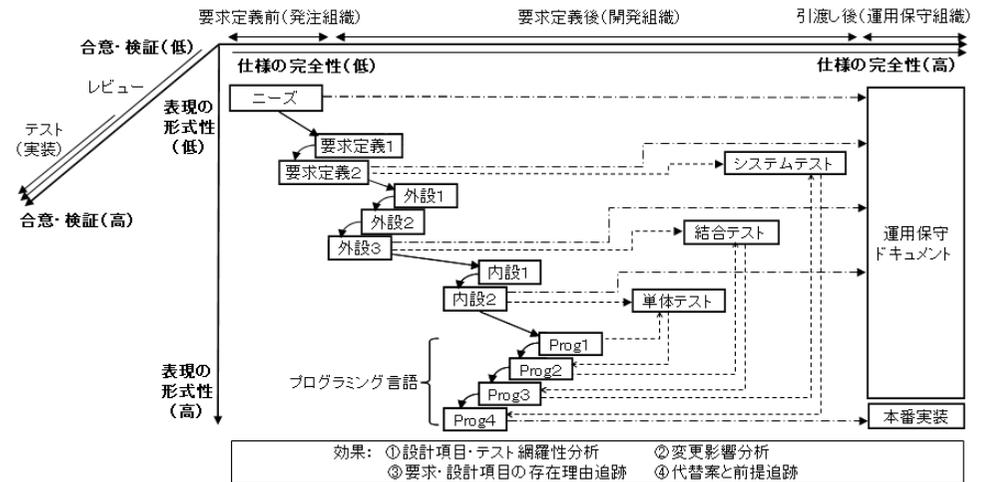


図2. トレーサビリティの枠組み
Figure 2. A traceability framework

5.2 トレーサビリティの枠組みの提案

ここでは、ウォーターフォールモデルを前提とした汎用的なトレーサビリティの枠組みを提案する。

Pohl K.氏らは、システム開発を仕様（Specification）、表現（Representation）、合意（Agreement）を軸とする3次元空間において、合意が得られていない曖昧な仕様と不完全な表現を、合意が得られた完全な仕様と形式的な表現に変換する過程としてモデル化している。その上で、この変換過程を追跡できるようにすることがトレーサビリティの役割としている¹⁸⁾。

図2は、筆者らが提案するトレーサビリティの枠組みであり、トレーサビリティを工程、合意・検証の程度、成果物の形式性の程度で表現している。Pohl K.氏らのモデルと図2の枠組みには、下記の解釈によって関係付けることができる。

- ・仕様は、要求定義、外部設計、内部設計という一連の工程を通して完全なものに変換される。
- ・表現も、一連の工程を通して、自然言語、図、表による形式性の低い表現から、

プログラミング言語による形式性の高い表現に変換される。

- ・要求定義書，設計書はレビューによって，また，プログラムはテストによって，関係者との合意が高められる。

矢印付き直線（実線）は，設計工程における水平トレーサビリティを表している。
矢印付き円弧は，同じ工程における成果物間の垂直トレーサビリティを表している。

矢印付き破線は，テスト工程のトレーサビリティを表しており，対応する設計書とテストケース，および，テストケースとプログラムの関連で構成されている。テストケースによって障害が発生した場合は，障害を修正する前と後のプログラム要素への関連を保持する。これは，単体テスト，結合テスト，システムテストからプログラム要素に対する2本の矢印付き破線で表現されている。

運用保守ドキュメントは，運用保守に必要な項目を網羅すべきものであるが，これらの項目は開発工程における成果物と無関係ではない。一点鎖線は，運用保守項目と開発工程における成果項目とのトレーサビリティを表している。

これらのトレーサビリティがサポートすべき効果（機能）としては，設計項目およびテスト網羅性分析，変更影響分析，要求および設計項目の存在理由追跡，代替案の前提追跡である。

6. おわりに

本稿では，トレーサビリティに関し，ソフトウェア関連業と主な業種との比較を行った。ソフトウェア関連業では，管理対象が多様，物理形状がない，変更が発生する，利用者が少ないといった特徴があり，トレーサビリティの管理を困難にしていることを論じた。一方，ソフトウェアシステム開発の標準・規格では，品質管理の観点からトレーサビリティの管理を義務付けるものがあるが，標準・規格の目的の違いから，トレーサビリティの定義は統一されていないという調査結果を得た。定義の違いは本質的ではないが，定義の違がトレーサビリティの対象や期待する効果に違いをもたらすことがあるので，定義にバリエーションがあることを理解することは重要であると考える。

今後の研究の方向性とトレーサビリティの枠組みについて提案したが，トレーサビリティに関する研究や標準・規格化の活動は膨大であり，本稿で対象にできたのは，その一部の活動にとどまっている。トレーサビリティの枠組みを精緻なものにしてゆくため，今後とも調査を継続する予定である。

参考文献

- 1) Arkley P. and Riddle S.: Overcoming the Traceability Benefit Problem, Proc. of the 13th IEEE International Conference on Requirements Engineering, pp.385 - 389 (2005).
- 2) California Department of Transportation: System Engineering for Intelligent Transportation Systems -

An Introduction for Transportation Professionals (2007).

- 3) California Department of Transportation: Systems Engineering Guidebook for ITS, Ver. 2.0 (2007).
- 4) CMMI Product Team: CMMI for Development, Ver. 1.2 - Improving processes for better products, CMU/SEI -2006-TR-008, Software Engineering Institute, Carnegie Mellon University (2006).
- 5) Gotel O. and Finkelstein A.: An Analysis of the Requirements Traceability Problem, Proceedings of the 1st Inter. Conference on Requirements Engineering, pp. 94-101 (1994).
- 6) Herzog E. and Torne A.: Information Modelling for System Specification Representation and Data Exchange, Proc. of the 8th IEEE Inter. Conference and Workshop on the Engineering of Computer-based Systems, pp.136-143 (2001).
- 7) <http://www.answers.com/topic/traceability> など.
- 8) <http://www.answers.com/topic/track-trace> など.
- 9) IEEE Std 1220-2005, IEEE Standard for Application and Management of the Systems Engineering Process, IEEE, New York (2005).
- 10) IEEE Std 1362-1998, IEEE Guide for Information technology, System Definition - Concept of Operations (ConOps) Document, IEEE, New York (1999).
- 11) IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, IEEE, New York (1990).
- 12) International Council on Systems Engineering: INCOSE Systems Engineering Handbook, Version 3.1 (2007).
- 13) ISO/IEC 90003:2004, Software engineering - Guidelines for the application of ISO 9001:2000 to computer software, ISO JTC1 SC7(2004).
- 14) Lindvall, M. and Sandahl, K.: Practical Implications of Traceability, Software Practice and Experience, vol. 26, no. 10, pp. 1161-1180 (1996).
- 15) National Aeronautics and Space Administration: NASA Systems Engineering Handbook, NASA/SP-2007-6105, (2007).
- 16) Object Management Group: OMG Systems Modeling Language Version 1.1 (2008).
- 17) Pfleeger S. L.: Software Engineering: Theory and Practice. Prentice-Hall, Inc., 3rd edition (2005).
- 18) Pohl K., Bockle G., and Van der Linden F.: Software Product Line Engineering: Foundations, Principles and Techniques, Berlin, Germany, Springer (2005).
- 19) Ramesh B. and Jarke M.: Toward Reference Models for Requirements Traceability, IEEE Transactions on Software Engineering Vol.27, No. 1, pp.58-93 (2001).
- 20) Royce W.W.: Managing the development of large software systems, Proceedings of IEEE WESCON, pp1-9 (1970).
- 21) Spanoudakis, G. and Zisman A.: Software Traceability: A Roadmap, Handbook of Software Engineering and Knowledge Engineering. World Scientific Publishing, pp.395-428 (2005).
- 22) (独)情報処理推進機構 ソフトウェア・エンジニアリング・センター著: ソフトウェア開発データ白書, 日経 BP 社(2007).
- 23) 高田, 細矢, 島: IC タグを活用したトレーサビリティシステムの開発, 石川島播磨技報, Vol.47 No.2, pp71-75 (2007).