人工知能学会
第10回知識流通ネットワーク研究会
SIG-KSN-012-02

ソフトウェア知識体系のメタモデルについての一考察

山本 修一郎

名古屋大学
愛知県名古屋市千種区不老町

Dimensions on Software Knowledge Bodies

Shuichiro YAMAMOTO

Nagoya University
Furo-cho, Chikusa-ku, Nagoya Aichi Japan

概要

多様なソフトウェア関連知識が体系化されてきており, 知識を効率的に習得する手段として期待されている. 一方, 多様な知識体系が出現することにより, 対象となるソフトウェアプロジェクトに対しては, 多様な知識体系を効率的に選択することが逆に困難になる状況が生まれている. 知識が専門化することによって個別化するため知識活用が複雑化するという状況がソフトウェア知識領域においても顕在化している. このため本稿では, 多様なソフトウェア知識体系を用いて大学院教育を実践した経験に基づいて, ソフトウェア知識体系のメタモデル構築と, それによる包括的なソフトウェア知識体系の理解に向けた取り組みについて述べる.

Abstract

Various bodies of knowledge on Software have been developed to promote learning practices of software development effectively. In this paper, an approach is proposed to develop a meta-model for integrating Software related bodies of knowledge. It is expected to help engineers systematically understand Software knowledge. The study is based on our lecture of a project management theory for graduate students of Nagoya University.

1 はじめに

筆者は 2011 年度から, 名古屋大学情報科学研究科で, プロジェクトマネジメントについての講義を担当している[1]. この講義では, 一般的なプロジェクトマネジメントに関する体系的な知識の枠組みと実践的な技法について解説している. 具体的には, プロジェクトマネジメント概論, プロセス知識, 開発対象知識, 品質保証知識, プロジェクト環境論, プロジェクト経営論, プロセス構成法論, エンタープライズ・アーキテクチャ論, プロセス管理論, 目標管理論, リスク管理論, コンプライアンス論, 問題解決論, コミュニケーション論などについて講述している. この内容からわかるように, 本講義は, 通常のプロジェクトマネジメントという言葉から想像される内容を越えていると思われることが多い.

しかし, 筆者が企業で 20 年以上にわたってシステム開発プロジェクトを担当した経験から, これらの知識がプロジェクトをマネジメントする上で重要な成功要因であったと考えている. もちろん, これらの知識を大学院の半期の講義で詳しく口述することができないことも明らかである. そういう意味では大学院生の皆さんには, 社会に出た後で遭遇する現実のプロジェクトとは何か, そしてそれを支えるために必要な知識領域について, 筆者の体験や現実社会の事例に基づいて解説している.

この講義経験から, これらのプロジェクト関連知識が個別的には体系化が進んでいること, 一方で相互の関係については共通性があるにもかかわらず十分に解明されているとは言えないことも認識した. 実際のプロジェクトマネジメントでも, 上述した知識領域を総合的に学習するカリキュラムなどないのであって, 経験的に習得することになる.

このため, 本稿では, ソフトウェアに関する知識体系を構成する基本要素を領域横断的に分析するための仮説を「ソフトウェア知識体系メタモデル」と仮称する. プロジェクトマネジメント知識ではなくソフトウェア知識に限定した理由は, 調査範囲を限定することで考察を効率化すること, 基本的な考え方はソフトウェア知識もプロジェクトマネジメント知識も類似すると考えたこと, したがって本検討のメタモデルをプロジェクトマネジメント知識に拡張できると考えたことの 3 点である.

本稿では, 上述した筆者による講義の内容からソフトウェアに関する知識体系を対象として, メタモデルの構築する上での課題について考察する. また, 上述した知識体系のどのひとつをとっても多数の有識者が参画して整理している訳だから, それらを横断的に統合する試みは個人だけで遂行できないこと

も明らかである．しかし，実際のプロジェクトでこれらの知識を活用する場合，統合化された知識体系がない以上，プロジェクトメンバによる適切な活用方法が必要である．

2 関連研究

以下では，ソフトウェア・アーキテクチャについての知識構築事例を紹介する．

2.1 アーキテクチャ・プリンシプルの次元

Greefhorst と Proper はエンタープライズ･アーキテクチャ・プリンシプル(Architecture Principle, AP)に対する次のような 9 個の次元を提案している[2]．すなわち，①アーキテクチャ情報が扱う話題についての次元，②アーキテクチャ情報が対象とするスコープ，③アーキテクチャ情報の一般性と④詳細性，⑤アーキテクチャ情報の対象読者，⑥アーキテクチャが置かれた状況の段階，⑦アーキテクチャ情報が達成すべき品質特性，⑧アーキテクチャ情報を記述する情報構造の抽象レベル，⑨アーキテクチャ情報の表現形式である．これらの次元は，ソフトウェア知識体系のメタモデルの候補となると考えられる．

アーキテクチャ・プリンシプルを記述する多様な次元に対して多様な選択ができることから，これらを明確にした記述が必要である．また，アーキテクチャ・プリンシプルをこれらの次元に基づいて分類することにより，その再利用を容易化できる可能性がある．たとえば Greefhorst と Proper は，59 個のアーキテクチャ・プリンシプルを情報分類と品質特性を明記して提示している．

2.2 アーキテクチャ・パターン

Buschmann ら[3]は，ソフトウェアのアーキテクチャ・パターンを体系化するために，①解決したい問題，②アーキテクチャ・パターンが適用される典型的な状況，③問題に対するアーキテクチャ・パターンによる解決方法，④アーキテクチャ・パターンの利点と欠点からなる特徴，⑤アーキテクチャ・パターンを活用するときの留意点や適用事例を記述することを提案している．

筆者らは，このアーキテクチャ・パターン記述体系に基づいて，アシュアランス・ケースの議論分解パターンを記述する方法を提案している[4]．

アーキテクチャ・パターンの記述項目は，アーキテクチャ・パターン技法を一般的に記述するために利用されているので，ソフトウェア知識体系のメタモデルの候補となると考えられる．

2.3 アーキテクチャ・メタモデル

ソフトウェア集約的システム(Software-intensive Systems)に対するアーキテクチャ記述への推奨プラクティス（IEEE Std.1471-2000[5]）では，アーキテクチャ関連用語を定義している．

たとえば，以下のような基本的な関係が用語とともに定義されている．

①関心事を展望するためにビュウポイントを使用する．②関心事をアーキテクチャ記述が識別する．③関係者が関心事を保有する．④アーキテクチャ記述がアーキテクチャを記述する．⑤ビュウポイントが関係者に提示される．⑥アーキテクチャ記述ビュウポイントを選択する．⑦ビュウがビュウポイントに適合する．⑧ビュウポイントがモデルの作成方法を確立する．⑨アーキテクチャ記述がモデルを統合する．⑩アーキテクチャ記述がビュウを構成する．⑪モデルがビュウに参加する．⑫関係者がアーキテクチャ記述を識別する．⑬システムが関係者に関係する．⑭環境がシステムに影響する．⑮システムが環境で動作する．⑯システムがアーキテクチャを保有する．⑰システムがミッションを実現する．

これらの用語とその関係に対して下図のようなアーキテクチャ・メタモデルを作成できる．

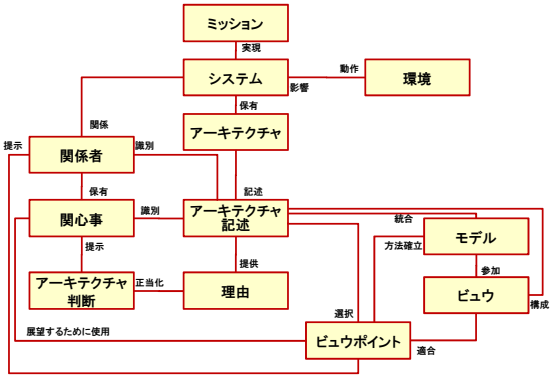

図 1　アーキテクチャ・メタモデル

アーキテクチャ・メタモデルは，ソフトウェア知識体系のメタモデルを構築する上での参考になる．

2.4 EA フレームワークの再利用

ミュンヘン工科大学の Buckl らは，状況指向型 EA マネジメント手法（A Situated Approach to Enterprise Architecture Management, SAEAM）を提案している[6]．SAEAM では，Beer が提案した VSM(Viable System Model)[7]を用いて組織の状況に応じて EA マネジメント手法を選択して統合化することができる．

このような複数の手法を選択的に統合する方法は，ソフトウェア知識体系のメタモデルを具体的なプロジェクトに適応する上で必要になる．

3 ソフトウェア知識体系の事例

本稿では，プロジェクトマネジメント論の講義で参照したソフトウェア知識体系を対象として共通的な記述項目を分析する．

3.1 SWEBOK

ソフトウェアエンジニアリング基礎知識体系 SWEBOK (Guide to the Software Engineering Body of Knowledge (SWEBOK V3)では，知識領域(Knowledge Area) ごとに，①概要(Introduction)，②知識領域の構成要素（トピックス），③トピックス参考文献対照表，④推奨参考文献，⑤関連文献一覧を記述している[8]．

SWEBOK では知識領域をトピックスに，2～3 階

層で階層的に分解している．

この階層分解は特定のアプリケーション分野，ビジネス利用，開発手法などに依存しないように構成されている．ただし，SWEBOK では，必要な知識は参考文献を直接参照することを推奨しているので知識を詳細に記述してはいない．

SWEBOK の知識領域は，①ソフトウェア要求，②ソフトウェア設計，③ソフトウェア構築，④ソフトウェアテスト，⑤ソフトウェア保守，⑥ソフトウェア構成管理，⑦ソフトウェア開発管理，⑧ソフトウェア開発プロセス，⑨ソフトウェア開発ツールと方法，⑩ソフトウェア品質である．

3.2 PMBOK[9]

PMBOK のプロセスには知識領域とプロセスグループという２つの側面がある．

PMBOK の知識領域には，①プロジェクト統合管理，②プロジェクトスコープ管理，③プロジェクト時間管理，④プロジェクト経費管理，⑤プロジェクト品質管理，⑥プロジェクト人材管理，⑦プロジェクトコミュニケーション管理，⑧プロジェクトリスク管理，⑨プロジェクト調達管理がある．プロセスグループには，①開始，②計画，③実行，④監視制御，⑤終了がある．

PMBOK では，プロセスごとに，入力，ツール・技法，出力について記述されている．あるプロセスの出力が他のプロセスの入力になるという関係がある．

3.3 CMMI[10]

CMMI (Capability Maturity Model Integration) モデルは組織が開発プロセスを改善するための実践知識を体系化している．CMMI は官民と協力して Software Engineering Institute (SEI)が中心となってまとめている．

CMMI-Dev.V1.3 では，プロセス領域について，目的，概要，関連プロセス領域，具体的ゴール，実践的活動，成果物の事例，一般的なゴール，一般的活動を記述している．CMMI では一般化されたプロセスを提示しているので, 実際のプロジェクトでは，プロセスごとに，目的，入力 開始基準，活動，役割，評価基準，検証手順，出力，完了基準を具体化する必要がある．

3.4 BABOK[11]

BABOK 2.0 では, ビジネス分析知識を①知識領域，②作業，③技法の３階層によって体系的に記述している．知識領域については，①知識領域の定義，②関連する作業，③関連する技法，④知識領域への入力，と⑤知識領域からの出力を記述している．

作業については，①作業の目的，②作業内容の記述，③関連するステークホルダ，④作業への入力，⑤作業からの出力，⑥作業の要素作業，⑦作業で活用される技法を記述している．

技法については，一般技法と作業の中でだけ記述されている個別技法がある．また入力については，作業結果による出力が後続する作業の入力になる場合と，外部の結果が入力になる場合がある．

一般技法については, ①目的, ②技法の概要記述, ③要素技法, ④留意点を記述している. 留意点では, 技法適用上の利点と欠点を明記する．

3.5 REBOK[12]

要求工学知識体系 REBOK(Requirements Engineering Body Of Knowledge) はユーザとベンダが協調した要求開発を実践するための知識体系であり，以下の特徴がある．

(1) ベンダ，ユーザを問わない共通の知識体系
(2) 要求アナリストに加えて, エンドユーザ, 経営者など，要求開発に関与するステークホルダが，必要に応じて習得すべき範囲と水準を整理した知識体系
(3) ビジネス要求, システム要求, ソフトウェア要求の 3 つのスコープに応じた知識体系
(4) エンタープライズシステム, 組込みシステムの要求開発に共通する技術の知識体系．ただし，ドメイン固有の知識は個別に定義

REBOK では，プロセスごとに，概説，目的，アクタ，手順，構成要素，技術，入力，参照，出力，関連知識領域を記述している．

3.6 ITIL[13][14]

ITIL(IT Infrastructure Library)では，目的に適した信頼性の高い安定したサービスを顧客に提供し，事業から信頼できるプロバイダとみなされるように，サービスマネジメントにおける実践的な知識を提供している．ITIL にはサービス・ライフサイクル全体を網羅するプロセスベースのフレームワークであり，戦略，設計，移行，運用，継続的改善という 5 プロセスからなる．

ITIL では，これらのサービスプロセスに対して，目標，主要原則，基本概念，役割，測定基準，課題，活動，他のプロセスとの関係を記述している．また活動に対して，達成目標，適用範囲，価値，活動手順を記述している．

3.7 TOGAF[15][16]

TOGAF V9 は 7 部から構成されている．第 1 部では，主要概念の説明と用語が定義されている．第 2 部は，エンタープライズ・アーキテクチャを開発するための段階的手法であるアーキテクチャ開発手法 ADM が説明される．ADM の説明では, 概要, 目的, 手順，入力，出力を記述している．

第 3 部では, ADM を適用するためのガイドラインと技法を説明している．

第 4 部では，アーキテクチャコンテンツ・フレームワークを説明している．アーキテクチャコンテンツ・フレームワークには，メタモデル，再利用のためのアーキテクチャ･ビルディング･ブロック, ADM の各工程の生産物が含まれる．第 5 部では，エンタープライズ・アーキテクチャ活動の生産物を格納する分類体系であるエンタープライズ・コンティニュームとツールを説明する．エンタープライズ・コンティニュームは，エンタープライズ・アーキテクチャで生産される全情報を体系的に分類して関係付けて格納できる仕組みのことだと考えればいい．第 6 部では TOGAF 参照モデルがファウンデーション・

アーキテクチャと統合情報基盤参照モデルとして説明される．エンタープライズ・コンティニュームと参照モデルによって，ビジネス・ケイパビリティからエンタープライズ・アーキテクチャの実践状況を獲得して，ビジネスの現状をビジネス・ビジョンに対して提示できるようになる．

最後に，第 7 部では，EA 活動の運営に必要な組織，プロセス，スキル，ロール，責任などが，アーキテクチャ・ケイパビリティ・フレームワークとして説明される．

3.8 SQuaRE[17]

SQuaRE (Software Product Quality Requirements and Evaluation)は，ソフトウェア製品品質要求評価のための新たなソフトウェア品質要求の標準である．SQuaRE では，スコープ，適合性，規範的参考文献，用語定義，ソフトウェア品質要求フレームワーク，品質要求の要求を記述している．

ソフトウェア品質要求フレームワークでは，目的，ソフトウェアとシステム，関係者と関係者の要求，ソフトウェアの性質，ソフトウェア品質測定モデル，ソフトウェア品質要求，システム要求の構成，品質要求ライフサイクルモデルを記述している．

品質要求の要求では，一般要求，ステークホルダ要求，ソフトウェア要求を記述している．

3.9 SQuBOK [18][19]

SQuBOK(Software Quality Body of Knowledge)は，JUSE(The Union of Japanese Scientists and Engineers)が 2007 に SQuBOK version 1 を制定している．

技術者に，品質保証知識（quality assurance）を教育するために，日本のソフトウェア品質についての暗黙知を定式化することにより，ソフトウェア品質の新しい課題を組織的に体系化している．これにより，ソフトウェア品質技術を顕在化して改善することにより，ソフトウェア品質保証プロセスを確立できるとしている．

SQuBOK の知識領域には，基本概念，マネジメント，品質技術の 3 個のカテゴリがある．各カテゴリには，知識領域，副知識領域，トピックスが階層的に整理されている．知識領域と副知識領域では，目的と下位知識へのポインターを記述する．最下位のトピックスでは，概要，関連知識，参考文献，関連文献を記述している．

4 知識体系の共通構造

上述したように，ソフトウェア関連知識体系では，知識領域，プロセス知識，プロダクト知識，技法知識について整理されていることがわかる．以下では，これらについて共通する知識構造に述べる．

4.1 知識領域

知識領域については，いずれの知識体系も下位の知識領域やトピックスによって階層的に知識が分解されている（表 1）．

4.2 プロセス知識

プロセス知識については，表 1 に示したように，すべての知識体系で説明されている．また，プロセス知識の記述項目を，目的・概要，入力と出力，役割，詳細手順，基準，関連技法の観点から比較すると表 2 のようになる．したがって，このプロセス知識の記述項目は，ソフトウェア知識体系のメタモデルの候補になる．

4.3 プロダクト知識

IEEE830 や ISO/IEC/IEEE29148 などのソフトウェア開発の工程生産物に対する知識の例には要求仕様書の目次例と要求文の制限構文がよく知られている．

アーキテクチャ要求仕様については TOGAF でも標準的な記述項目を例示している．またアーキテクチャ・リポジトリで成果物を管理することを推奨している．

5 ソフトウェア知識構築事例

以下では，上述したソフトウェア知識体系を組織内のプロジェクトで活用するために，プロジェクトの状況に応じて適応する仮想的な知識活用シナリオを説明する．

5.1 関心事に応じた知識体系の選択

プロジェクトが対象とするシステムの範囲を考える．もし組織全体のプロジェクトを扱うのであれば，TOGAF と PMBOK を選択する必要がある．

プロジェクトの進行に応じて，対象とするプロセスが変化するから，それに応じた知識体系を選択する必要がある．すなわち，ビジネス分析プロセスには BABOK，要求開発には REBOK，ソフトウェア開発プロセスには SWEBOK と CMMI－DEV，運用プロセスには ITIL，品質保証プロセスには SQuBOK/SQuaRE, プロセス改善プロセスには CMMI を選択できる．ここで，CMMI は多様なプロセスに対する改善プロセスであるから，他のソフトウェア知識体系と補完的に組合せることができる可能性がある．

5.2 技法知識の選択と適応

知識体系ごとに技法知識の記述レベルが異なる．たとえば，SWEBOK では具体的な知識を理解しようとすると参考文献を調べる必要がある．逆に，他の知識体系では技法知識が簡潔にまとめられているけれども詳細な内容を調べようとすると SWEBOK ほど詳細な文献一覧は用意されていないという課題がある．

いずれの場合でも，知識体系では一般的な技法知識が集合として整理されているので，プロジェクトの状況に応じて複数の技法知識を選択して，適用法を具体化する必要がある．

6 考察

6.1 知識体系の範囲

各ソフトウェア知識体系を比較した結果を表 3 に示す．前述したように，この表からもソフトウェア知識体系間で，共通する部分と，目的の違いから，逆に相補的な組合せの可能性があることが分かる．

今後，このようなソフトウェア知識体系の結合が期待できる．

6.2 知識体系の適用プロセス

知識体系は実践的に有効であることが確認された知識を体系的に整理していることから，そのままでは，実際のプロジェクトに適応することは難しい．すなわち，プロジェクトの状況に応じた具体化が必要になる．複数のソフトウェア知識体系が共通する技法を要素として持つ場合，その技法を利用する目的が知識体系間で異なる可能性があるので，注意が必要である．また，関連する技法が補完的に活用できればいいが，それぞれの技法が前提とする条件が対立する可能性もある．

したがって，今後，これらの点を考慮した知識体系の適用プロセスの研究を進める必要がある．たとえば，関連研究で紹介した VSM を用いて，プロジェクトの状況に応じてソフトウェア知識体系の技法知識を選択して統合化する手法を明らかにする必要がある．

6.3 知識体系の技法の比較

本稿では，各ソフトウェア知識体系が持つ技法について具体的に比較していない．このため，具体的な技法知識がどのように知識体系で整理されているかを明らかにしていく必要がある．

6.4 知識体系の範囲

本稿で対象としたソフトウェア知識体系は限定されている．たとえば，筆者が講義しているプロジェクトマネジメント論のうち，コンプライアンス論，問題解決論，コミュニケーション論などについての知識体系を取り上げることができなかった．今後，これらの知識体系についても同様の研究を進める必要がある．

7 まとめと今後の課題

本稿では，ソフトウェア知識体系を対象としてメタモデルを構築する取組みについて紹介した．ただし，調査対象とした知識体系の事例は 9 件であり，一般化するためには，他の事例についても評価する必要がある．しかし，取り上げた知識体系は広く知られていることから，本稿で紹介したソフトウェア知識体系の共通要素は，メタモデルの重要な構成要素を示していると考えられる．

また，本稿の内容はメタモデル構築のための予備検討の段階にとどまっているため，メタモデルの構築までは至っていない．今後，メタモデルの試案を取りまとめる予定である．

さらに今回の分析では，具体的な技法知識の内容にまで踏み込んだ分析をしていない．今後，異なる知識体系の技法知識がどのようにして再結合できるのかについても検討していく必要がある．

参考文献

[1] 山本修一郎, プロジェクトマネジメント論, https://acs.is.nagoya-u.ac.jp/

[2] Greefhorst, D. and Proper, E., *Architecture Principles, The Cornerstones of Enterprise Architecture*, The Enterprise Engineering Series, 2011, Springer.

[3] F.ブッシュマン，R.ムニエ，H．ローネルト，P.ゾンメルラード，M.スタル，ソフトウェアアーキテクチャ，ソフトウェア開発のためのパターン体系，Pattern-Oriented Software Architecture：A System Of Patterns, 1996, John & Wiley & Sons, Ltd.，近代科学社, 2000

[4] 山本修一郎, 松野裕, ディペンダビリティケース分解パターンについての考察，KBSE 研究会，2013.3.15

[5] IEEE Std 1471 -2000, IEEE Computer Society, Recommended practice for architectural description of software –intensive systems, 2000.

[6] Sabine Buckl, Christian M. Schweda, Florian Matthes, A Situated Approach to Enterprise Architecture Management, 2010

[7] Beer, S., The Viable System Model: its provenance, development, methodology and pathology, The Journal of the Operational Research Society, Vol.35, pp.7-26, 1984．

[8] Guide to the Software Engineering Body of Knowledge (SWEBOK V3), http://www.computer.org/portal/web/swebok/home

[9] PMBOK ガイド, http://www.pmi.org/

[10]CMMI, CMU/SEI-2010-TR-033, 2010

[11] IIBA 日本支部, ビジネスアナリシス知識体系ガイド, 2009

[12] 情報サービス産業協会 REBOK 企画 WG 編，要求工学知識体系 REBOK(Requirements Engineering Body Of Knowledge)，近代科学社，2011

[13] ITIL, itSMF japan http://www.itsmf-japan.org/itil

[14] iTSMF, ITIL V3 Foundation Handbook, 2009

[15] THE Open GROUP, TOGAF V.9 A Pocket Guide, 2008

[16] オープン・グループ・ジャパン，http://www.opengroup.or.jp/

[17] Jorgen Boegh, A New Standard for Quality Requirements, IEEE Software, pp.20-27, January/ February, 2008.

[18] ソフトウェア品質知識体系ガイド―SQuBOK Guide―, http://www.juse.or.jp/software/365/

[19] SQuBOK Guide βversion, http://www.juse.or.jp/software/pdf/squbok-beta.pdf

表 1　ソフトウェア知識体系の比較

項目	SWEBOK	PMBOK	CMMI	BABOK	REBOK	ITIL	TOGAF	SQuARE	SQuBOK
知識領域	ソフト開発	プロジェクト管理	開発，調達，サービス	ビジネス分析	要求工学	サービス	EA	ソフト品質	ソフト品質
プロセス知識	開発プロセス	管理プロセス	開発，調達，サービスプロセス	分析プロセス	要求開発プロセス	サービス開発プロセス	EA 開発プロセス	品質管理プロセス	品質管理プロセス
プロダクト知識	*1	*1	*1	*1	要求文書	管理文書	EA 文書	品質特性	品質特性
技法知識	*2	○	○	○	○	○	○	○	○
知識発展	プロセス発展	プロセス発展	成熟度発展	ビジネス価値実現	*3	サービス価値実現	ビジネス価値実現	品質向上	品質向上
知識活用	*4	*4	*4	*4	*4	*4	ADM 適応ガイド	*4	*4

注：*1)成果物の構造をとくに規定していない.

*2)知識体系の中では詳述していない.

*3)知識適用の運用監視によるフィードバックプロセスが考慮されていない.

*4)体系化された知識の適応が必要であるとしているが，具体的な適応法については明記していない.

表 2　プロセス知識の記述構造

	目的・概要	入出力	役割	詳細手順	基準	技法
PMBOK	ツール・技法	入力，出力	--	--	--	--
CMMI	目的	入力，出力	役割	活動，検証手順	開始基準, 評価基準，完了基準	--
BABOK	作業目的	作業入力, 作業出力	関連ステークホルダ	作業内容の記述，作業の要素作業	--	作業で活用される技法
REBOK	概説，目的	入力，出力	アクタ	手順，構成要素，技術，参照，関連知識領域	--	--
ITIL	目標	--	役割	主要原則，基本概念，課題，活動，他のプロセスとの関係 活動適用範囲，活動手順	活動価値	測定基準，達成目標
TOGAF	概要，目的	入力，出力	--	手順	--	--
SQuaRE	目的	--	関係者 関係者要求	ソフトウェアとシステム，ソフトウェアの性質，ソフトウェア品質要求，システム要求の構成	ソフトウェア品質測定モデル	品質要求ライフサイクルモデル

注) SQuBOK を省略した.

表 3　ソフトウェア知識体系の共通性分析

	SWEBOK	PMBOK	CMMI	BABOK	REBOK	ITIL	TOGAF	SQuX
SWEBOK	ソフト開発	開発プロジェクト	開発管理	要求工学	要求工学	--	要求管理	ソフト品質
PMBOK		プロジェクト管理	開発管理	要求管理	リスク管理	運用管理	EA 管理	品質改善プロセス
CMMI			開発，調達，サービス	プロセス改善	プロセス改善	サービス成熟度	EA 成熟度	品質改善プロセス
BABOK				ビジネス分析	ビジネス要求	ビジネス要求	ビジネスアーキテクチャ	品質保証
REBOK					要求工学	運用要求	要求管理	品質要求
ITIL						サービス	運用監視	サービス品質
TOGAF							EA	EA 品質
SQuX								ソフト品質

注) 紙幅の節約のため SQuX で，SQuaRE と SQuBOK を総称した.