

SWEBOK V3.0 日本語訳版^{※1}の 連続紹介 —3の2

新谷 IT コンサルティング

新谷 勝利

1. はじめに

前回の SWEBOK V3.0 連続紹介の 3 の 1 では以下について紹介した。

- 1) 1968 年の NATO 会議の Software Engineering というタイトルの報告書^{※2}
- 2) 2001 年 5 月発行の SWEBOK Trial Version^{※3}
- 3) 2005 年 6 月発行のソフトウェアエンジニアリング基礎知識体系—SWEBOK 2004 —日本語訳版^{※4}

今回は、SWEBOK V3.0 日本語訳版を紹介してゆく。この紹介にあたり、原文の発行元の IEEE 及び日本語訳の発行元のオーム社から引用をこころよく承諾していただいたことに感謝する。

2. V3.0 の構造

V3.0 は、以下の章から構成される。章名の右端の*印は SWEBOK 2004 に存在していたことを示す。ただし、第 1 章から第 8 章及び第 10 章は、章名が同じであるからといって内容まで全く同じではない。違いは後で示すが、副知識、及びトピックスを見るとはっきりする。第 9 章は、前版では、「ソフトウェアエンジニアリングのためのツールおよび方法」と称していた。V3.0 においては、ツールは各知識領域に副知識領域として集約されている。第 11 章以降は、第 12 章はトピックスとしても新規に追加され新しい章として、その他の章は大幅に内容が拡張され章として独立している。

- 第 1 章 ソフトウェア要求*
- 第 2 章 ソフトウェア設計*
- 第 3 章 ソフトウェア構築*
- 第 4 章 ソフトウェアテスト*
- 第 5 章 ソフトウェア保守*
- 第 6 章 ソフトウェア構成管理*
- 第 7 章 ソフトウェアエンジニアリング・マネージメント*

- 第 8 章 ソフトウェアエンジニアリングプロセス*
- 第 9 章 ソフトウェアエンジニアリングモデルおよび方法
- 第 10 章 ソフトウェア品質*
- 第 11 章 ソフトウェアエンジニアリング専門技術者実践規律
- 第 12 章 ソフトウェアエンジニアリング経済学
- 第 13 章 計算基礎
- 第 14 章 数学基礎
- 第 15 章 エンジニアリング基礎

SWEBOK V3.0 では、数学を始めとする基礎知識の重要性を強調している。これは近年ソフトウェアが社会及び企業・組織に与える影響が大きくなっていることから、その開発に関与するエンジニアが実践するべき規律に関する認識を高めるのみならず、従来定義されていた基礎知識と合わせ以下を達成することを目的としている。

- 1) 全世界に、ソフトウェア・エンジニアリングに対する一貫した大局観を普及・促進する。
- 2) 他の体系化された実践規律、例えばコンピューターサイエンス、プロジェクト管理、コンピュータエンジニアリング、及び数学に対して、ソフトウェア・エンジニアリングの範囲と位置付けを詳細に示す。
- 3) ソフトウェア・エンジニアリング実践規律の内容を、特徴づけして示す。
- 4) ソフトウェア・エンジニアリング知識体系に対して、トピックスを通じたアクセスを提供する。

【脚注】

- ※1 松本吉弘訳、ソフトウェアエンジニアリング基礎知識体系—SWEBOK V3.0 —、オーム社、2014 年 11 月 25 日、ISBN978-4-274-50521-8
- ※2 Software Engineering, Report on a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th Oct., 1968、<http://homepages.cs.ncl.ac.uk/brian.randell/NATO/>
- ※3 http://cis.unipd.it/didactics/STS_school/Software_development/Trial_Version1_00_SWEBOK_Guide.pdf
- ※4 <http://www.computer.org/portal/web/swebok/2004guide;jsessionid=dba3ca44da150499851e19bd752a>

5) カリキュラム開発、個人の認証、及び免許取得に必要な基礎を提供する。

これらの目的達成のために、基礎知識体系として、以下の階層構造で定義されている。

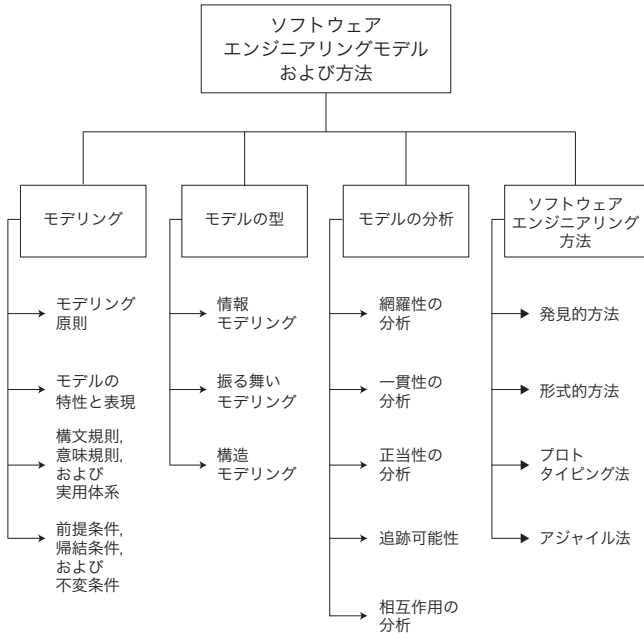
- 1) 知識領域—前述の章で示されるもの
- 2) 副知識領域
- 3) トピックス

今回の連続紹介3の2においては、先ず追加された知識領域の階層構造を紹介することにより、何が従来に増してソフトウェアエンジニアに知識獲得が期待されているかを紹介する。

3. 第9章 ソフトウェアエンジニアリングモデル及び方法

当知識領域の副知識領域は、以下の構造図における第一レベルの分割されたものである。

- 1) モデリング
- 2) モデルの型
- 3) モデルの分析
- 4) ソフトウェア・エンジニアリング方法



各副知識領域は、関連する参考文献において紹介されているトピックスをカバーしている。各章の最後に、「トピックスと参照資料の対照表」（構造図に次いで示す）及び「参照資料」のリストが準備されており、読者は更なる学習ができるようになっている。第9章の参考資料は7つ引用されているが、残念ながら日本語に翻訳されているのは2015年5月現在 Sommerville の「ソフトウェ

トピックスと参照資料の対照表

	Budgen 2003 [1*]	Mellor and Balcer 2002 [2*]	Sommerville 2011 [3*]	Page-Jones 1999 [4*]	Wing 1990 [5*]	Brookshear 2008 [6*]	Boehm and Turner 2003 [7*]
1 モデリング							
1.1 モデリング原則	c2s2, c5s1, c5s2	c2s2	c5s0				
1.2 モデルの特性と表現	c5s2, c5s3		c4s1.1p7, c4s6p3, c5s0p3				
1.3 構文規則、意味規則、および実用体系		c2s2.2 p6	c5s0				
1.4 前提条件、帰結条件、および不変条件		c4s4		c10s4p2, c10s5 p2p4			
2 モデルの型							
2.1 情報モデリング	c7s2.2		c8s1				
2.2 振る舞いモデリング	c7s2.1, c7s2.3, c7s2.4	c9s2	c5s4				
2.3 構造モデリング	c7s2.5, c7s3.1, c7s3.2		c5s3	c4			
3 モデルの分析							
3.1 網羅性の分析			c4s1.1p7, c4s6		pp8-11		
3.2 一貫性の分析			c4s1.1p7, c4s6		pp8-11		
3.3 正当性の分析					pp8-11		
3.4 追跡可能性			c4s7.1, c4s7.2				
3.5 相互作用の分析		c10.c11	c29s1.1, c29s5	c5			
4 ソフトウェアエンジニアリング方法							
4.1 発見的方法	c13, c15, c16		c2s2.2, c7s1, c5s4.1				
4.2 形式的方法	c18		c27		pp8-24		
4.3 プロトタイプング法	c12s2		c2s3.1			c7s3p5	
4.4 アジャイル法			c3			c7s3p7	c6.app. A

参照資料

of Object-Oriented Design in UML, 1st ed., Addison-Wesley, 1999.

[1*] D. Budgen, Software Design, 2nd ed., Addison-Wesley, 2003.

[2*] S.J. Mellor and M.J. Balcer, Executable UML: A Foundation for Model-Driven Architecture, 1st ed., Addison-Wesley, 2002.

[3*] I. Sommerville, Software Engineering, 9th ed., Addison-Wesley, 2011.

[4*] M. Page-Jones, Fundamentals of Object-Oriented Design in UML, 1st ed., Addison-Wesley, 1999.

[5*] J.M. Wing, "A Specifier's Introduction to Formal Method Science," Computer, vol. 23, no. 9, 1990, pp. 8, 10-23.

[6*] J.G. Brookshear, Computer Science: An Overview, 10th ed., Addison-Wesley, 2008.

[7*] B. Boehm and R. Turner, Balancing Agility and Discipline: A Guide for the Perplexed, Addison-Wesley, 2003.

「ソフトウェアエンジニアリング」が事例としてあるが、SWEBOK V3.0 に引用されている 2011 年版ではなく、1993 年版であり、これでは参照資料にはならない。以降、各章における参照資料はほとんど翻訳されていない。

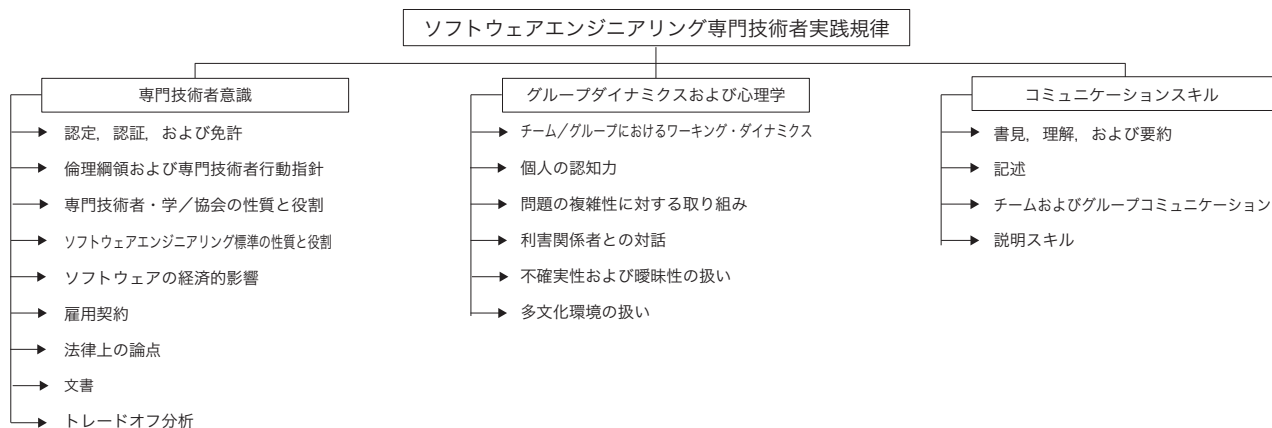
4. 第11章ソフトウェアエンジニアリング 専門技術者実践規律から第15章エンジニアリング基礎の知識領域構造図

以降、新しい第11章から第15章までの5つの章の構造図を紹介する。

1) 第11章 ソフトウェアエンジニアリング専門技術者実践規律

トピックスの一つとして留意されなければならないも

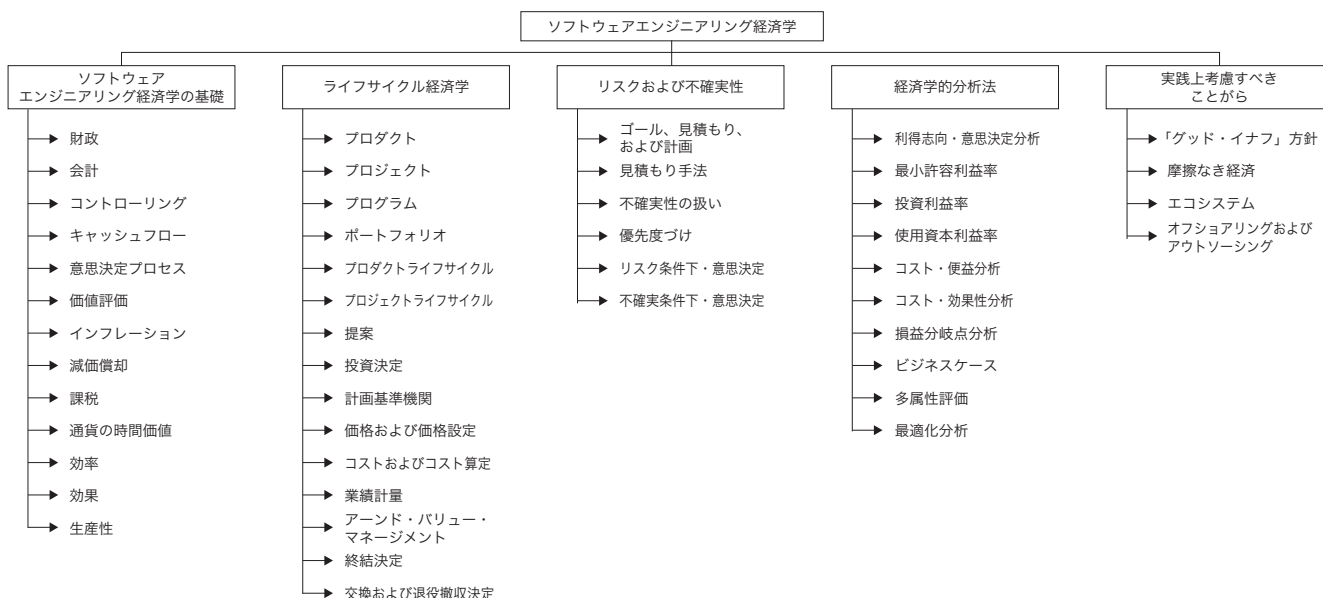
のとして、「倫理要綱及び専門技術者行動要綱」がある。エンジニアが専門職として社会的責任を担うためには本トピックスは必須事項であろう。ソフトウェア開発には多くのステークホルダが関与し、「コミュニケーションスキル」というものが副知識領域として特出している。その他、各トピックスにはものにより副トピックスも説明されている。例えば、「法律上の論点」というトピックスには、標準、登録商標、特許、著作権、企業秘密、専門職責任、法定要求、通商規則遵守及びサイバー犯罪、の副トピックスがある。



2) 第12章 ソフトウェアエンジニアリング経済学

とくに日本におけるソフトウェアビジネスにおける利益率は他国に比して高いとは言えない。健全な利益無しには業界としての発展は期待できず、エンジニアはともすれば技術論に関心が向きがちであるが、経済学の基礎知識を持つことも期待されている。ソフトウェアが出荷された後のみならず、出荷前までの投入労力に対して収入が発生するが、開発期間は言うに及ばず出荷後当該製品あるいはサービスが使用されなくなるまでのライフサ

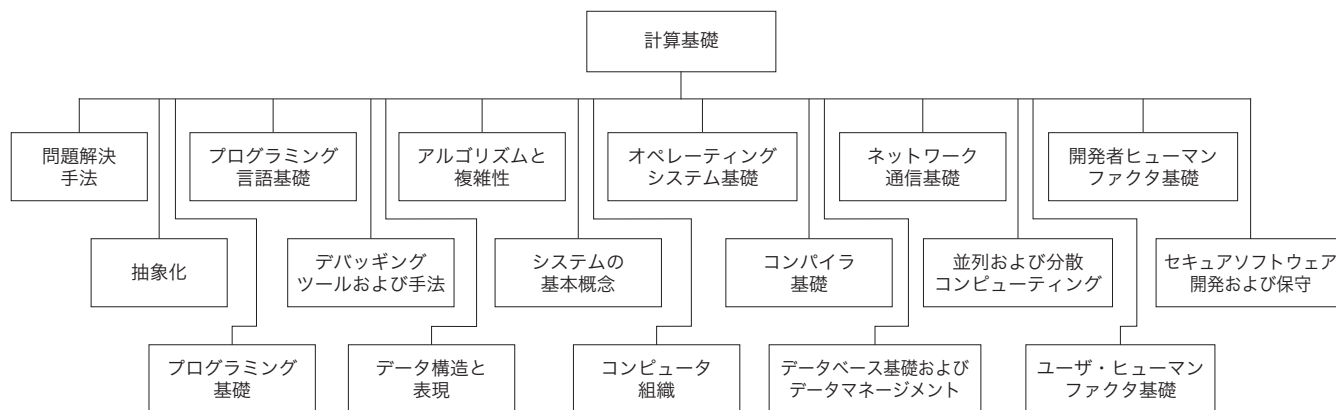
イクルにおける経済学の知識が必要である。利益率が低い時にはなおさら、ライフサイクルを通じたコスト推定と実際コストのモニタリング、リスクの事前予測及びその回避等に関わる副知識領域で述べられるトピックスが必要であり、更に「実践上考慮すべきことから」という副知識領域では、「グッド・イナフ」の方針、摩擦なき経済、エコシステム、オフショアリング及びアウトソーシングというトピックスがカバーされている。



3) 第13章 計算基礎

この章は、コンピュータサイエンスという領域にかかわる知識領域をカバーしている。ソフトウェアはコンピュータで動くものであるが故に、効率的なソフトウェア開発にはコンピュータサイエンスの背景が必要であ

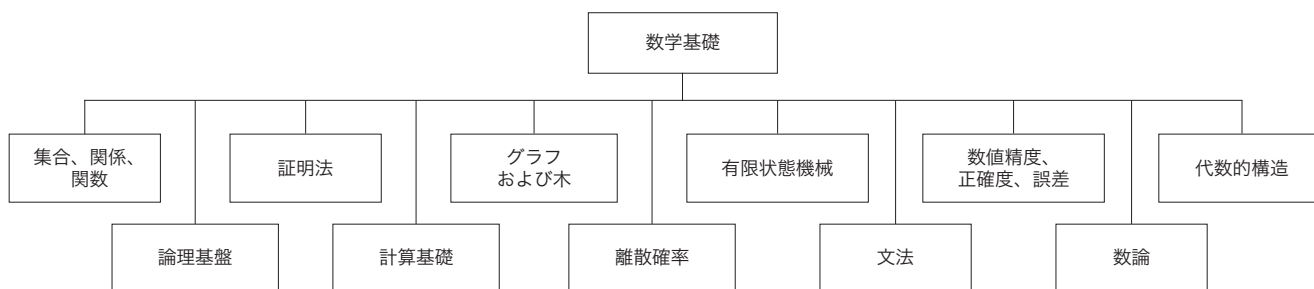
る。例えば、抽象化という副知識領域は、抽象化水準、カプセル化、階層及び代替抽象というトピックスをカバーしている。これらトピックスは McConnell の Code Complete という大書により説明されており、幸いにして第2版が上下二巻で日本語に翻訳されており、良い参考文献となっている。



4) 第14章 数学基礎

今までのソフトウェアエンジニア育成において、比較的手を抜かれていたものが、数学という知識領域であろう。それもあってか、この章は比較的丁寧に各副知識領域を説明している。数学は形式手法と結びつけられて語

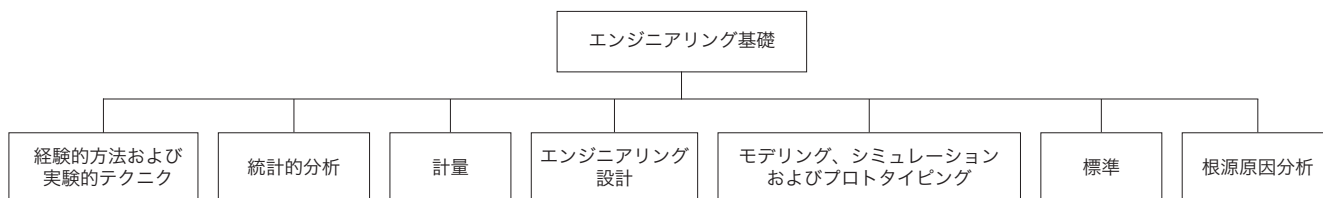
られることがあるが、形式手法を採用しない場合でも、ソフトウェアエンジニアは精密な手段を持つことが期待されている。プログラミング言語は最終的には、極めて数学的なものであるからである。



5) 第15章 エンジニアリング基礎

ソフトウェア開発がエンジニアリング的に実施されるためには、対象とするシステムに体系的にアプローチし、

実践規律に基づき定量可能な手法を用いることが期待されている。これは、エンジニアリングの定義そのものである。



5. 終わりに

連載2回目の今回は、SWEBOK V3.0 になって新しく紹介された知識領域をそれらの全体像を見る形で説明した。説明は全体を鳥瞰することを目的としており、各知

識領域については「ソフトウェアエンジニアリング基礎知識体系—SWEBOK V3.0—」を本解説の読者の皆さんが読まれることを期待している。次回の連載最終回においては、従来からカバーされていた知識領域に関し、改定された部分を中心に紹介する。