

2015年5月28日

SERC フォーラム

1. プログラム
2. 保守技術者シラバス
3. SWEBOK・JIS X0161 翻訳相違一覧表
4. 懇親会会場（なりまつ 会費 3000 円）

なりまつの場所は、下記の地図を参照



SERC フォーラム
ソフトウェア保守技術者の技術力評価を考える
主催
ソフトウェア・メンテナンス研究会 (SERC)

●日時 : 2015年5月28日(木) 13:30-17:00 (受付13:00-)

●プログラム

13:00 - 13:30 受付

13:30 - 13:35 開会の挨拶

13:35 - 14:35 基調講演 「SWEBOK V3.0の解説」

新谷 ITコンサルティング 新谷 勝利氏

14:35 - 15:05 研究報告(1) 「SWEBOK ソフトウェア保守の解説とシラバス」

SERC 幹事 馬場 辰男

15:05 - 15:20 休息

15:20 - 15:50 研究報告(2) 「ソフトウェア保守技術者の技術力評価に向けて」

「仕事品質」改善教室・SERC 研究員 大島 道夫

15:50 - 16:50 研究報告(3) 「技術力評価のためのサンプル試験問題」

SERC 研究員 高橋 宏志

16:50 - 17:00 クロージング

-----講師略歴-----

・新谷 ITコンサルティング 新谷勝利氏

1969年 京都大学卒

1969年 日本 IBM にカスタマー教育インストラクターとして入社

1984年 NYC に新設された IBM のソフトウェアエンジニアリング研究所にて、
インストラクター資格を取得

2004年 IPA ソフトウェア・エンジニアリング・センター研究員。

主としてプロセス改善タスクフォースをリード。

ISO JTC1 SC7/WG10 (プロセスアセスメント) 委員、

ISO/IEC 33004 のエディターを務める。

2013年 IPA/SEC を退職。IPA/SEC 専門委員及び連携委員に就任、現在にいたる。

その他、フラウンホファ IESE との共同研究を通してゴール指向の経営企画支援
の方法の普及、及び形式手法の普及にあたっている。

ソフトウェアに関する同一用語(英語)のSWEBOK V3.0 日本語翻訳とJIS X0161の相違一覧

↓ SWEBOK V3.0 日本語版で同一英単語に異なる日本語訳がある場合、それぞれを提示した

No.	原用語(英文)	JIS X0161及び文献1)の日本語訳	SWEBOK V3.0 第5章 日本語版	SWEBOK V3.0 日本語訳出現箇所
2	adaptive maintenance	適応保守	適応のための保守	1.6
3	analyzability	解析性	分析可能性	2.4.1
4	appropriate	適切な	適当な 適切な	2.1.2 2.4.1
5	assess (assessment)	評価する	査定する	3.2.1, 3.2.3, 4.4
6	baseline	基準線	ベースライン	3.2
7	capability	能力	資質 能力	2.1.4 3.1
8	change	変更	変化 変更	序説, 2.1.4 多数
9	change request	変更要求	変更要請	2.1.3, 2.3.1, 3.2.4
10	changeability	変更性	変更容易性	2.4.1
11	completion	完了	完成	4.4
12	concept	概念	概念 構想	1, 2.1.2 3.2.3
13	control	制御	制御 コントロール	1.3, 2.1.4, 4.3 2.2.5, 3.2.1, 3.2.4
14	correction	訂正	是正	1.4, 1.6, 2.1.1, 2.1.4
15	corrective maintenance	是正保守	是正のための保守 (※1) 是正保守	1.6 3.1 派生的読み物(2番目)
16	critical	重要な	重要な 臨界的	1, 2.1.2 2.2.5, 3.2.4
17	develop	(文脈に応じて)用意する, 策定する, 作成する, 取得する, 開発するなど	(文脈に関わらず)開発する	多数
18	develop options	選択肢を用意する	選択肢を開発する	2.1.3
19	effort	活動	工数	序説, 1.2, 2.1.1, 2.3.3, 2.4, 2.4.1, 3.2.1, 4.3, 4.5
20	enhancement	改良	増補	1.3, 1.4, 1.6, 2.2.1, 3.1
21	entities	構成要素	実在	2.4
22	faults	障害	フォールト	1.3, 1.4, 1.6
23	help desk	ヘルプデスク	相談窓口 ヘルプデスク	序説, 3.2.1 他はヘルプデスク
24	identify	特定する	同定する 識別する (※2)	1.2, 2.2.4, 3.2.1, 3.2.3, 3.2.4 1.3, 2.1.3, 2.3.1, 2.4.1, 4.3 2.4
25	implementation	実装	実現	多数
26	improvement	改善	改善 改良	1.6, 3.1 2.1.4
27	infrastructure	基盤	インフラストラクチャ	2.2.5
28	integrity	完全性	完全性 保全性	1.1 3.2
29	maintainability	保守性	保守容易性	序説, 1.4, 1.6, 2.1.3, 2.1.4, 2.4, 2.4.1, 4.2

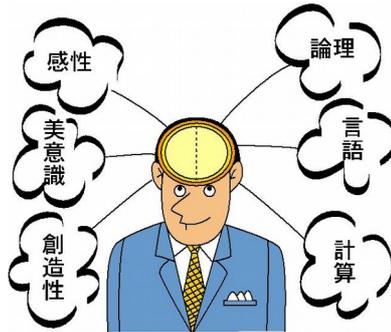
No.	原用語(英文)	JIS X0161及び文献1)の日本語訳	SWEBOK V3.0 第5章 日本語版	SWEBOK V3.0 日本語訳出現箇所
30	maintenance concept	保守概念	保守構想	3.2.3
31	maintenance review/acceptance	保守レビュー及び受入れ	保守レビュー／承認 保守レビュー／容認	1.2 3.1
32	management	管理	マネージメント 管理	2, 2.2, 2.2.1, 2.2.3, 2.4 1.4, 2.1.3, 2.2.3, 3.2.2, 3.2.4, 5
33	measurement	測定	計量	2, 2.3.3, 2.4, 5
34	measures	測定値	計量尺度	2.4, 2.4.1
35	migration	移行	移転 移転実装 移行	1.2 3.1 4.4
36	modification	修正	是正 修正	多数 2.1.2
37	Modification Implementation	修正の実施	是正の実現 修正の実現	1.2 3.1
38	modification request	修正依頼	是正要請 修正要請	1.2 2.1.3, 3.2.1, 3.2.2, 3.2.4
39	perfective maintenance	完全化保守	完全化のための保守	1.6
40	practices	慣行	実践 実践慣行	2.2.3 3.2.1
41	preventive maintenance	予防保守	予防のための保守	1.6
42	Problem and Modification Analysis	問題分析及び修正の分析	問題及び是正分析 問題および修正のための分析	1.2 3.1
43	process implementation	プロセス実装	プロセス実現	1.2, 3.1
44	progressively	徐々に	前進的に	1.2, 3.2.1
45	reactive	受け身の	認知応答的	1.6
46	released	リリース済みの	可用化された	1.2
47	retirement	廃棄	退役撤収 用済みソフトウェアの撤収	1.2, 4.5 3.1
48	size	規模	大きさ	2.4, 2.4.1, 3.2.1
49	software products	ソフトウェア製品	ソフトウェアプロダクト	序説, 1.2, 1.6, 2.1.3, 2.1.4, 2.3.1, 3.2, 3.2.4
50	specified	定めた	仕様化された 特定された	2.1.4 2.4.1
51	tailor	修整する	仕立てる	3.2
52	testability	試験性	テスト可能性	2.4.1
53	transition	引継ぎ	引渡し 移行 遷移	序説 3.2.1, 3.2.3 4.4
54	understand	理解する	理解する 解明する	1, 1.4, 1.5, 2.1.1, 2.3, 2.4, 3.2.1, 4.5 4.1
55	version	版	版改訂 版	2.1.1 1.2

文献1) 増井, 馬場他著「～ISO14764による～ ソフトウェア保守開発」2007 ソフトリサーチセンター

(※1) 第5章の最終ページ「派生的読み物」へ日本語訳が移された可能性あり。

(※2) 特定個人名がidentifiesした部分を日本語訳では割愛した可能性あり。

SERC フォーラム ソフトウェア保守技術者の技術力評価を考える



ソフトウェア保守技術者のシラバス



ソフトウェア保守技術者のシラバス (SERC-Dグループ 第23年次報告書から抜粋)

はじめに

(1) 技術レベルについて

技術者には初級者からベテランまであり、持つ技術の幅が大きいため、技術者認定試験を行う場合にはその対象によりレベル分けする必要がある。個々の知識をどの程度理解し、身につけているかの尺度である知識レベルと、その人がどの程度の知識を持っているかの尺度である技術レベルとを設定した。

表1. に知識レベルをに技術レベルを示す。

表1. 知識レベル

知識レベル	補足説明
レベル1 (知っている)	概念や用語を知っており、その概要を述べることができる。
レベル2 (知識を説明できる)	概念や用語の意味や背景を理解しており、具体的な例を挙げて説明することができる。
レベル3 (概念と使い方がわかる)	概念や技術の使い方がわかっており、それらを適切に選択して、限られた条件の下で与えられた課題を解決できる。
レベル4 (詳しく理解し応用できる)	概念や技術を詳しく理解しており、実用的な問題を解決するために、その知識を応用できる。
レベル5 (熟達している)	実社会の複雑な問題に対して、構造を明らかにして要素に分解するとともに、解決に必要な検討を加えて結論を導くことができる。

表2. に技術レベルを示す。

表2. 技術レベル

技術レベル	説明	知識レベル
基本技術者	ソフトウェア保守に携わる者には全員に知っておいて欲しい基礎知識をもつ基本技術者。一般担当者レベル	レベル1 (一部2を含む)
アドバンス技術者	保守プロジェクト運営の中核となり、推進すし、既存のプロセスの改善を提案する技術者。 サブリーダー、リーダー、管理者クラス	レベル2, レベル3
エキスパート技術者	社内外に情報を発信し、ソフトウェア保守の技術革新に貢献する技術者。	レベル4, レベル5

シラバス（基本技術者レベル）

本年度見直し改訂した基本レベルのシラバスを以下示す。

1. ソフトウェア保守の基本事項

1. 1 定義と用語

●学習目標

- ・ ISO/IEC 14764:2006 に定義された用語を理解する。
- ・ ソフトウェア保守の目的を理解する。

●学習の対象となる用語，概念

ISO/IEC 14764:2006, JIS X 0161, 適応保守, 是正保守, 緊急保守, 保守性, 改良保守, 修正依頼, 完全化保守, 予防保守, 問題報告, ソフトウェア保守, ソフトウェア引継ぎ

1. 2 保守の性質

●学習目標

- ・ ソフトウェア保守がライフサイクルを通してソフトウェア製品を支えることを理解する。
- ・ 保守者という述語が保守アクティビティを実行する個人を指す場合と，組織を指す場合とがあることを理解する。
- ・ ISO/IEC 14764:2006 で述べられているソフトウェア保守の主要アクティビティを理解する。
- ・ 保守者からみた開発者との関係を理解する。

●学習の対象となる用語，概念

ソフトウェアライフサイクル, 保守プロセス, 保守者, 開発者との連携

1. 3 保守の必要性

●学習目標

- ・ 保守は，ソフトウェアが利用者の要求を継続して満たすために必要とされることを理解する。
- ・ 保守は開発時に使われたライフサイクルモデルにかかわらず必要とされることを理解する。
- ・ 保守の目的は，障害の是正，設計の改善，エンハンスの実施，他のソフトウェアとのインタフェースの維持，異なるハード・ソフト・システム要素・通信機能を使えるようにプログラムを適合させること，レガシーソフトウェアの移行，ソフトウェアの廃棄の実施であることを理解する。
- ・ 保守者のアクティビティを構成する次の5つの重要特性を理解する。
 - 1) ソフトウェアの日々の動きの管理を維持する
 - 2) ソフトウェア修正の管理を維持する
 - 3) 既存機能を完全なものにする
 - 4) セキュリティ上の脅威を認識し，セキュリティ上の脆弱性を除去する

5) ソフトウェアの性能が、容認できないレベルまで劣化することを防止する

●学習の対象となる用語、概念

利用者要求，障害の是正，設計の改善，エンハンスの実施，他ソフトウェアとのインタフェース，動作環境の変化へのプログラム適合，レガシーソフトウェアの移行，ソフトウェアの廃棄

1. 4 保守コストの構造

●学習目標

- ・ソフトウェア保守の非是正アクションがソフトウェアライフサイクルにかかわるコストの大部分を占めることを理解する。
- ・ソフトウェア保守コストの大部分は訂正以外のものに使われることを理解する。
- ・ソフトウェア保守コストの多くが訂正に使われているというのは勘違いであり，それは改良と訂正を区別しない管理が原因であることを理解する。
- ・ソフトウェアの保守コストは，ソフトウェアの運用およびハードウェア環境に関連することを理解する。
- ・ソフトウェア保守コストが組織の理念，競合相手，保守プロセス，製品，要員等の組織環境と関係することを理解する。
- ・保守コストの構造の例を挙げられる。

●学習の対象となる用語、概念

ソフトウェア保守コストの構造，改良と訂正，非是正アクション

1. 5 ソフトウェアの進化

●学習目標

- ・レーマンの8つの進化の法則を理解する。
- ・この8つの法則に対する対策案／例を挙げられる。

●学習の対象となる用語、概念

進化型システム，複雑度，進化プロセス，自己調整的，漸進的拡張度

1. 6 保守の分類

●学習目標

- ・ソフトウェア保守の4分類を，訂正，改良およびプロアクティブ，リアクティブの4象限の分類で理解する。
- ・是正保守の定義と特徴を理解する。
- ・適応保守の定義と特徴を理解する。
- ・完全化保守の定義と特徴を理解する。
- ・予防保守の定義と特徴を理解する。

●学習の対象となる用語、概念

是正保守，緊急保守，適応保守，完全化保守，予防保守，訂正，改良，プロアクティブ（事前実施），リアクティブ（事後実施）

2. ソフトウェア保守の主要課題

2. 1 技術的課題

2. 1. 1 限られた理解

●学習目標

- ・ソフトウェア製品の保守に要する労力の半分は，ソフトウェアを理解することに費やされることを理解する。
- ・他者が開発したソフトウェアの場合は特にそうであるが，保守母体を完全に理解できることは稀であり，限られた理解のもとで修正箇所の洗い出し，修正方法を定めることを理解する。
- ・ソフトウェア・エンジニアにとって，ソフトウェア母体の理解が重要なことであることを理解する。

●学習の対象となる用語，概念

製品理解手法（ソースコード解析，仕様書解析，マニュアル，ブラックボックステスト），変更点管理，ソフトウェア構成管理，オープンソース導入時の事例

2. 1. 2 テスト

●学習目標

- ・保守者にとって回帰テストは，重要なテストであることを理解する。
- ・異なった保守チームが同時に異なった問題を対策する時には，両方を意識したテストが必要であることを理解する。
- ・保守のテストでは，オンラインを止められないことがあることを理解する。
- ・以上のソフトウェア製品のメンテナンス時のテストの重要性を理解した上で，総てのメインパスを毎回テストするコストは，時間と金銭的に難しい課題であることを理解する。

●学習の対象となる用語，概念

各種テスト手法理解（カバレッジテスト，外部仕様テスト，単体・結合テスト，ブラックボックステスト，回帰テストなど）

2. 1. 3 影響分析

●学習目標

- ・影響分析（既存のソフトウェアの変更の影響について完全な分析）をどのように行うか，その記述が必要であることを理解する。
- ・保守者は，ソフトウェアの構造と内容の詳細な知識が必要なことを理解する。
- ・保守者は，ソフトウェアの変更に対しての見積もりを実施することを理解する。
- ・変更に対してのインパクト分析やリスク分析も必要なことを理解する。

・ IEEE 24764 の影響分析のタスクを理解する。

●学習の対象となる用語，概念

ソフトウェア構成管理，修正する機能と関係する機能の洗い出し，grep 作業による洗い出し，排他資源の確認，性能影響の確認，プロトコルへの影響確認，データスキーマの影響確認，互換性，接続性，従来仕様の永続性，製品コンセプトが踏襲できているか？

2. 1. 4 保守性

●学習目標

- ・ソフトウェア製品の保守の機能と変更の定義を理解する。
- ・ソフトウェア開発活動中に保守コストを削減するためのコントロールの必要性を理解する。
- ・体系的，成熟したプロセス，テクニック，およびツールの存在が，ソフトウェアの保守性を高めるのに役立つことを理解する。

●学習の対象となる用語，概念

JIS X 0161 : 2008 (ISO/IEC 14764 : 2006) ， 保守の機能，変更の定義，機能仕様，品質特性，ソフトウェアのマニュアル整備，体系的および成熟したプロセス

2. 2 管理上の課題

2. 2. 1 組織目標との整合性

●学習目標

- ・保守組織の目標と個別保守プロジェクトの目標との整合性（予算，顧客，プロセス，人材育成の各目標視点）がとれていることの重要性を学ぶ。
- ・保守組織の上級管理職（専門職）が保守レビューに参加することが，投資収益率改善につながることを学ぶ。

●学習の対象となる用語，概念

保守組織の目標，保守活動への投資とリターン，保守プロジェクト予算，上級管理職レベルでのビュー，投資収益率，目標の整合

2. 2. 2 人員確保

●学習目標

- ・保守組織のスタッフが保守要員を確保する戦略について学ぶ。
- ・保守組織においては，優秀な保守要員が長く組織内にとどまるよう，保守作業の価値，重要性，楽しさ，やり甲斐などを継続的に説明し，保守要員のモチベーション維持・向上が大切であることを学ぶ。

●学習の対象となる用語，概念

保守要員の確保と維持，保守作業の価値，保守作業の重要性，保守作業の楽しさ，保守作業のやり甲斐，保守要員のモチベーション維持・向上

2. 2. 3 プロセス

●学習目標

- ・ソフトウェアライフサイクルプロセスの概念（ソフトウェアは開発したら終わりということではないこと）を正しく学ぶ。
- ・ソフトウェア開発プロセスとソフトウェア保守プロセスの各アクティビティを比較し、両プロセスの違いを学ぶ。
- ・ソフトウェア保守プロセスのアクティビティで管理上の課題となる点を学ぶ。

●学習の対象となる用語，概念

ソフトウェアライフサイクルプロセス，ソフトウェア保守アクティビティ，ソフトウェア開発プロセス

2. 2. 4 保守の組織的側面

●学習目標

ソフトウェア保守の責務を担当する組織，あるいは職務をどのようにして決めるかを学ぶ。

- ・専門化を実施する
- ・コミュニケーションの経路を作る
- ・エゴレス(非利己)で，学究的な雰囲気を促進する
- ・個人に対する依存（属人化）を減らす
- ・定期監査チェックを実施する
- ・チーム制を実施する

●学習の対象となる用語，概念

ソフトウェア保守の責務を担当させる組織，恒久的な保守チームの利点，割り当てと委任，属人化防止，チーム制，定期監査

2. 2. 5 外部・海外調達

●学習目標

- ・外部調達と海外でのソフトウェア保守は，IT産業の主要な構成要素となっていることを学ぶ。
- ・ソフトウェア保守も含め外部調達しつつあることを学ぶ。
- ・外部調達に際しては，ソフトウェア保守サービスの範囲，サービスレベル合意項目，契約書の詳細項目，言語障壁の改善策，双方にとっての有効初期投資額，プロセスなどの明確化が重要であることを学ぶ。

●学習の対象となる用語，概念

要求される保守サービスの範囲，サービスレベル合意の項目，契約書の詳細，遠隔サイトのヘルプデスクは母国語の話し手を配置，有意義な初期投資，自動化を要求する保守プロセスの確立

2. 3 保守コストの見積もり

2. 3. 1 コスト見積もり

●学習目標

- ・保守のコスト見積りの基本概念を理解する。
- ・保守のコスト見積りの要点と開発のコスト見積りの要点に異なる部分があることを理解する。

●学習の対象となる用語，概念

保守コスト，影響分析，影響範囲の識別，変更に要するリソース，技術的コスト要因，非技術的コスト要因，ISO/IEC 14764:2006

2. 3. 2 パラメトリックモデル

●学習目標

- ・保守要件は対応工程ごとの作業比率（負荷山積み）の形態がさまざまであることを理解する。
- ・保守のコスト見積りにおいて、適切かつ詳細に前提条件を置くことが重要であることを理解する。

●学習の対象となる用語，概念

パラメトリックモデル，数学的モデル，過去の時系列データ計測，コストドライバ（コストを大きく変化させる）属性

2. 3. 3 経験値

●学習目標

- ・保守要件は比較的短期間で対応が完了することが多いため，過去のコスト見積りとコスト実績の比較データ（経験値）が十分な数記録されていることを理解する。
- ・新たな保守コスト見積りを実施する場合，それらの経験値を参考にして見積もることで精度の高い見積りができることを理解する。

●学習の対象となる用語，概念

経験値，保守経験，保守見積もり過去データ，計測プログラム，人・時間

2. 3. 4 開発コスト見積りとの相違点

●学習目標

- ・開発コストの見積り手法をそのまま適用するとは，保守コスト見積りを精度高く見積もるには有効でないことを理解する。
- ・工程別見積り，ファンクションポイント法などを保守コスト見積りに応用する場合，開発コスト見積りにない注意点があることを理解する。

●学習の対象となる用語，概念

ひとこぶラクダモデル, ふたこぶラクダモデル, 原因調査, インパクト分析, 修正設計, テスト環境構築, 回帰テスト

2. 4 ソフトウェア保守計測

●学習目標

- ・保守の計量尺度は, その組織の状況に基づいてふさわしいものが決定されなければならないことを理解する。

●学習の対象となる用語, 概念

保守性 (または, 保守容易性) とその4つの副特性 (分析容易性, 変更容易性, 安定性, テスト容易性)

(参考) 開発と共通した計量尺度: ソフトウェアの規模, 工数, 工程, 品質

3. 保守プロセス

3. 1 保守プロセス

●学習目標

- ・ソフトウェアの代表的な保守プロセスを理解する (ISO/IEC 14764:2006)

●学習の対象となる用語, 概念

ISO/IEC 14764:2006, 変更要請, クラス分けと識別, 分析, 設計, 実現, システムテスト, 受け入れテスト, 引き渡し, プロセス実装 (プロセス実現), 問題分析および修正分析 (問題および変更分析), 修正の実施 (変更実装), 保守レビュー/受け入れ, 移行 (移転実装), 廃棄 (ソフトウェア退役撤収)

(参考) 「ソフトウェア保守開発」(P88)では, 保守プロセスを円滑に進めるためのプロセス (管理, 環境整備, 要員管理, 契約) を提示している。

3. 2 保守アクティビティ

3. 2. 1 特有のアクティビティ

●学習目標

- ・ソフトウェア保守には, 開発と共通のアクティビティおよび保守特有のアクティビティがあることを理解する。

●学習の対象となる用語, 概念

保守特有のアクティビティとして, 移行, 変更要請の受け入れ/拒絶, 変更要請および問題報告のための相談窓口, 影響分析, ソフトウェア支援, 保守契約などがある

3. 2. 2 支援アクティビティ

●学習目標

- ・保守担当が, 実施すべき支援アクティビティを理解する

●学習の対象となる用語, 概念

保守担当者が、実施すべき支援アクティビティには、ソフトウェア保守計画、ソフトウェア構成管理、検証と妥当性確認、ソフトウェア品質保証、レビュー／監査／ユーザ訓練、保守担当の訓練などが挙げられる。

3. 2. 3 保守計画アクティビティ

●学習目標

- ・保守者は計画の見地と結び付けられる多くの課題に取り組まなければならないことを説明出来る。
- ・さらにソフトウェア保守計画は新しいソフトウェア開発を行うという決定から始まり、保守のコンセプト文章がメンテナンス計画に続いて作られるべきであることを理解する。

●学習の対象となる用語、概念

事業計画、保守計画、リリース／バージョン計画、個別のソフトウェア変更要求計画、保守コンセプトの内容

3. 2. 4 ソフトウェア構成管理

●学習目標

- ・ソフトウェア構成管理（SCM）の手続きに要求されるそれぞれのステップについて規定される事項を理解する。
- ・さらにソフトウェア保守のSCMは変更ソフトウェア開発のためのSCMと違っている点があることを理解する。

●学習の対象となる用語、概念

ソフトウェア製品の識別、認可、実装、リリース、ステップの検証、妥当性確認、識別

3. 2. 5 ソフトウェア品質管理

●学習目標

- ・望ましい品質を達成するために、ソフトウェア品質保証（SQA）のためのアクティビティと技法、つまりV&V、レビューと監査は他のプロセスとの連携を全て強いなければならないことを理解する。

●学習の対象となる用語、概念

ソフトウェア保守の品質プログラム、開発プロセス、技法、配布可能物、テスト結果の改善

4. 保守のために技法

4. 1 プログラム理解

●学習目標

- ・プログラム変更に際して、プログラムソースコードを読み、理解することを学ぶ。
- ・ソースコードを分類（整理）したり、表示したりするツールで重要なものはコード・ブラウザである。

・明確で簡潔なドキュメントもプログラムの理解を助けることを理解する。

●学習の対象となる用語，概念

オープンソース，ソフトウェア，コード・ブラウザ

4. 2 リエンジニアリング

●学習目標

- ・リエンジニアリングとリファクタリングを学ぶ。
- ・リエンジニアリングは，老化したレガシーなソフトウェアを新しい形式に再構成するために行う調査と改造，そしてそれに続けて実施する新しい形式の実装までを指す。リファクタリングは，振る舞いを変更せずに，プログラムを再構成することを目的としたリエンジニアリング技術を指す。
- ・リファクタリングは，プログラムの構造とその保守性の改善を目指して行われ，マイナーチェンジの中で，その技術が利用できることを理解する。

●学習の対象となる用語，概念

リエンジニアリング，レガシーソフトウェア，リファクタリング

4. 3 リバースエンジニアリング

●学習目標

- ・リバースエンジニアリングを学ぶ。
- ・リバースエンジニアリングが，ソフトウェアのコンポーネントおよびそれらの相互関係を識別し，別の形式やより高い抽象概念でのレベルのソフトウェア表現するために行う，ソフトウェアを分析するプロセスであることを理解する。
- ・リバースエンジニアリングは，ソフトウェアを変更することでもなく，新しいソフトウェアを作成するものでもないことを理解する。
- ・リバースエンジニアリングによって作成されるソースコードからコールグラフおよび制御フローは，再文書化や設計情報の再作成に利用されることを学ぶ。
- ・物理的データベースから論理スキーマを作ろうとするデータ・リバースエンジニアリングは，ここ数年重要性が増してきたことを学ぶ。

●学習の対象となる用語，概念

リバースエンジニアリング，コールグラフ，制御フロー，再文書化，設計情報の回復，データ・リバースエンジニアリング

4. 4 移行 (Migration)

●学習目標

- ・システムの寿命が続く間，異なる環境の中で稼働させるために，変更が必要となった場合における移行に学ぶ。
- ・移行には次のタスクを全て実施することが望ましいことを理解する。

移行計画，目的の通告，運用及び教育訓練の実施，完了の通告，運用後のレビュー，データの保管

●学習の対象となる用語，概念

移行のための要求分析及び要求定義，移行のためのツールの開発，ソフトウェア製品及びデータの変換，移行の実施，移行の検証，移行後の旧環境の支援，目的の通告，旧環境での並行運用，利用者の教育支援，旧環境の保管（文書，コード，ログ），完了の通告，運用後のレビュー，データの保管

4. 5 廃棄

●学習目標

- ・ソフトウェア製品が有用でなくなった場合には，廃棄する。廃棄について学ぶ。
- ・廃棄するかどうかの判断を支援するために，分析を実施することが望ましい。次のことに費用をかける価値があるか判断することが望ましいことを理解する。
 - － 時代遅れの技術の保持
 - － 新しいソフトウェア製品を開発することによる新技術への移行
 - － モジュール性を達成するための新しいソフトウェア製品の開発
 - － 保守を容易にするための新しいソフトウェア製品の開発
 - － 標準化を達成するための新しいソフトウェア製品の開発
 - － 納入者からの独立を容易にするための新しいソフトウェア製品の開発
- ・ソフトウェア製品は，新しいソフトウェア製品に替えてもよいが，幾つかのケースでは替えられない。
- ・ソフトウェア製品を廃棄するために，保守者は，廃棄を成し遂げるための処置を決定し，廃棄を可能にするために必要なステップを策定し，文書化することが望ましい。廃棄されたソフトウェア製品によって保管されたデータの利用について考慮することが望ましい。
- ・廃棄アクティビティからのすべての成果物を，構成管理の制御下に置く。
- ・廃棄には次のタスクを全て実施することが望ましいことを理解する。

廃棄計画，目的の通告，運用及び教育訓練の実施，完了の通告，運用後のレビュー，データの保管

●学習の対象となる用語，概念

廃棄計画，全体又は一部の支援停止，ソフトウェア製品及び関連文書の保管，将来にわたって残る支援責任，新しいシステム・ソフトウェア製品への切り替え，保管したデータへのアクセス可能性，ソフトウェア製品廃棄の影響，代替ソフトウェア製品，廃棄のスケジュール，新旧製品からのデータ収集，データ整理及び分析，現場に固有な問題の文書か，旧ソフトウェア及びデータの保管，旧機器の撤去，旧ソフトウェア及びデータの保管，旧ソフトウェア及びデータの複製，安全な場所への媒体保管

5. ソフトウェア保守ツール

●学習目標

- ・ソフトウェア保守に利用するツールの種類や目的を理解する。

●学習の対象となる用語，概念

プログラム保守ツール（プログラムスライサー，静的アナライザ，動的分析アナライザ，クロスリファレンス，従属分析（依存性分析）等），リバースエンジニアリングツール，ソフトウェアのテスト，構成管理，文書化，測定ツール

6. 参考文献

- ・ ISO/IEC 14764, IEEE Std 14764-2006, Software Engineering – Software Life Cycle Processes – Maintenance
- ・ JIS X 0161 : 2008, ソフトウェア技術—ソフトウェアライフサイクル—保守
- ・ ～IS014764による～ソフトウェア保守開発，増井和也他著，株式会社ソフトウェアリサーチセンター発行，ISBN978-4-8873-249-4
- ・ ソフトウェアエンジニアリング基礎知識体系—SWEBOK 2004 ，松本吉弘訳，オーム社発行，ISBN-13: 978-4274500299
- ・ ソフトウェア品質知識体系ガイド—SQuBOK Guide, SQuBOK 策定部会編，オーム社発行，ISBN-13: 978-4274501623
- ・ 共通フレーム2013 ～経営者，業務部門とともに取組む「使える」システムの実現～，独立行政法人情報処理推進機構（IPA）技術本部 ソフトウェア・エンジニアリング・センター（SEC）編集・発行，ISBN-13 : 978-4-905318-19-4



SWEBOK V3.0をそれまでの関連版 との比較で概説

新谷勝利

新谷 ITコンサルティング

(2015年5月28日SERC Forum用に改訂)

Topics



1. ソフトウェアエンジニアリングの歴史
2. 1968年NATO会議からの報告
3. 2001年5月発行、SWEBOK Trial Version
4. 2004年2月発行、SWEBOK 2004
5. 2013年12月発行、SWEBOK V3.0
 - 5.1 SWEBOKの構造
 - 5.2 SWEBOK V3.0対2004年版
 - 5.3 V3.0における主たる追加副知識領域
 - 5.4 ソフトウェアプロフェッショナル認証
6. ソフトウェアエンジニアリングは効果をあげてきたか？

1. ソフトウェアエンジニアリングの歴史



1) 1969年1月発行、NATO会議のSoftware Engineeringという
タイトルの報告書

<http://homepages.cs.ncl.ac.uk/brian.randell/NATO/>

2) 2001年5月発行、SWEBOK Trial Version、

http://cisas.unipd.it/didactics/STS_school/Software_development/Trial_Version1_00_SWEBOK_Guide.pdf

3) 2004年2月発行、SWEBOK 2004、日本語訳版あり

<http://www.math.unipd.it/~tullio/IS-1/2007/Approfondimenti/SWEBOK.pdf>

4) 2013年12月発行、SWEBOK V3.0、日本語訳版あり

<http://www.computer.org/web/swebok/v3>

2015/05/28

新谷ITコンサルティング

3

2. 1968年NATO会議からの報告



- 1967年： 秋に、NATO科学委員会はコンピューター科学に関する作業部会を発足。
- 1967年： 暮れに、ソフトウェア開発者の開発に実践的規律が他のエンジニアリング分野と同様必要で、それを「ソフトウェアエンジニアリング」ということを決定。ソフトウェアの設計、ソフトウェアの生産、及びソフトウェアのサービスの観点で議論。
- 1969年： 報告書発行
- 2001年： オリジナル報告書をスキャンしOCR技術を用いてテキスト版作成。編集の上、ニューキャッスル大学のサーバーから広く入手可能。

2015/05/28

新谷ITコンサルティング

4

NATO会議：一般的論点



- コンピューターの導入は年率25～50%で増えており、これらの増加に対応するソフトウェアのために25万人にのぼる分析者及びプログラマーに影響。
- ほとんどのプログラムは正常に動くであろうが、そうでない分野も出てくる。
- 危機と言えない分野があるかもしれないが、特に大規模システムにおいて危惧がある。
- 大規模システムにおいては、ソフトウェアの不具合の発生をなくすことはできない。
- 他のエンジニアリング分野と比べ、ソフトウェアエンジニアリングはまだその初期段階。
- プログラミングのコスト、スケジュール管理は、依然として低い評判。
- ソフトウェア開発の工程管理の難しさの源泉は、進捗をどう測定するのがよいのかわかっていないことにある。
- ソフトウェア不具合は指数関数的に増加している。
- ソフトウェア開発への要望は現場の能力を超えてなされている。

2015/05/28

5

NATO会議：ソフトウェアエンジニアリングの性質



- ソフトウェア開発はフィードバックループであり、それを通して改善。
- 設計と実装のペアが繰り返され、再構築を経て最終の成果物。
- 設計の段階でユーザーが使用時のオプションに関与できない。
- どの段階においても、外部仕様はユーザーに入手可能なものを示し、内部仕様は外部仕様を実現するプログラム構造を示す。
- プロジェクトの初期の段階から最終段階まで外部仕様に関わるものと内部仕様に関わるものの間にはフィードバック系ができていなければならない。
- ソフトウェアシステムの構造として、一番上にアプリケーション、次いでミドルウェア、そして、サービスルーチン及び制御プログラムとなっている。
- より下位のものを変えることはより上位のものに影響を与えるので、維持に経費がかかる。

2015/05/28

新谷ITコンサルティング

6



- ソフトウェア設計の方法論は、プログラムとは何かの理解の下に、それが開発される手法、手順及び技術から構成。
- プログラミングには、アートの部分があるが故に、技術の側面を教育し、それを実践するように動機付け。
- ハードウェア、オペレーティングシステム、およびそれらのリリースレベルから独立したパッケージはない。これに対応するには、システムエンジニアリングという範疇が必要。
- プログラム設計及び生産の管理のために、以下を考察。

設計プロセス

設計プロセスを実施するための組織

プログラムの文書化

汎用コンピューター用のツールの開発



- 設計と実装が分けられているのは、担当する人が異なることによるが、例えばパフォーマンスは実装の影響が大きく、設計がパフォーマンスに責任を持つのであれば、実装者が何も判断せずに実装できるような設計がなされている必要がある。
- 製品の品質は設計による。
- 設計プロセスは繰り返されるものであり、特に大規模システムの場合、他の繰り返して問題が分かったとしても自分の仕様を変更するのは締め切りの関係で困難な場合があり、0版、1版、N版というものが作られてしまうことになる。
- 一般的に実践されているものに、先ず仕様が決められ、それに基づいて設計がなされ、更に実装、と進む。設計者が仕様の対象に関して無知の場合、何が起こる？

3. 2001年5月発行、SWEBOK Trial Version



- 第1章 ガイドへの序説
- 第2章 ソフトウェア要求
- 第3章 ソフトウェア設計
- 第4章 ソフトウェア構築
- 第5章 ソフトウェアテスト
- 第6章 ソフトウェア保守
- 第7章 ソフトウェア構成管理
- 第8章 ソフトウェアエンジニアリング・マネジメント
- 第9章 ソフトウェアエンジニアリングプロセス
- 第10章 ソフトウェアエンジニアリングのためのツール及び方法
- 第11章 ソフトウェア品質

2015/05/28

新谷ITコンサルティング

9

4. 2004年2月発行、SWEBOK 2004



第1章から第11章までは、Trial Versionと同じ知識領域をカバーし、以下を追加。

第11章 ソフトウェアエンジニアリングに関連するディシプリン

2015/05/28

新谷ITコンサルティング

10

5. 2013年12月発行、SWEBOK V3.0



新しいソフトウェアエンジニアリングの定義

「ソフトウェアの開発、運用及び保守に対して、
システムチックでよく訓練された定量化可能な
アプローチを適用すること、すなわち、**エンジニアリングをソフトウェアに適用すること**」

ISO/IEC/IEEE Systems and Software Engineering Vocabulary (SEVOCAB)

<http://www.computer.org/sevocab>

SWEBOKの構造



- 知識領域
 - 副知識領域
 - トピックス

SWEBOK V3.0対2004年版 (1/5)



V3.0	2004年版	注釈
ガイドへの序説	第1章 ガイドへの序説	
第1章 ソフトウェア要求	第2章 ソフトウェア要求	②
第2章 ソフトウェア設計	第3章 ソフトウェア設計	
第3章 ソフトウェア構築	第4章 ソフトウェア構築	
第4章 ソフトウェアテスト	第5章 ソフトウェアテスト	
第5章 ソフトウェア保守	第6章 ソフトウェア保守	
第6章 ソフトウェア構成管理	第7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ジメント	第8章 ソフトウェアエンジニアリングマネジメント	
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プロセス	
第9章 ソフトウェアエンジニアリングモデル及び手法	第10章 ソフトウェアエンジニアリングのためのツール及び手法	①
第10章 ソフトウェア品質	第11章 ソフトウェア品質	

2015/05/28

新谷ITコンサルティング

13

SWEBOK V3.0対2004年版 (2/5)



V3.0	2004年版	注釈
第11章 ソフトウェアエンジニアリングプロフェッショナルプラクティス	第12章 ソフトウェアエンジニアリングに関連するディシプリン	③④⑤⑥ ⑦
第12章 ソフトウェアエンジニアリングエコノミクス		⑦
第13章 コンピューティング基盤		⑦
第14章 数学基盤		⑦
第15章 エンジニアリング基盤		⑦
付録A: 知識領域記述のための仕様	付録A: SWEBOKのIronmanバージョンにおける知識領域記述のための仕様	
付録B: ソフトウェアエンジニアリング基礎知識体系(SWEBOK)を支援するIEEEとISO/IECの規格	付録B: SWEBOKへのガイドの進化	
付録C: 統合された参考文献	付録C: IEEE及びISOソフトウェアエンジニアリング標準のSWEBOK知識領域への割り付け	⑧
	付録D: ブルームの教育評価分類法によるトピックスのクラス分け	

2015/05/28

新谷ITコンサルティング

14

SWEBOK V3.0対2004年版(3/5)



- ① V3.0における「ツール」の扱いは2004年版とは異なり各知識領域の副知識領域化。
- ② V3.0と2004年版で知識領域名は同じだが、副知識領域が異なる。
=> 4.V3.0における主たる追加副知識領域

SWEBOK V3.0対2004年版(4/5)



③④⑤⑥⑦ ディシプリンを再構成し、幾つかは知識領域に

- ・新規: 第11章 プロフェッショナルプラクティス
- ・マネジメント
 - => 第12章 ソフトウェアエンジニアリングエコノミクス
- ・コンピュータサイエンス
 - => 第13章 コンピューティング基盤
- ・数学=> 第14章 数学基盤
- ・コンピュータエンジニアリング
 - => 第15章 エンジニアリング基盤
- ・プロジェクトマネジメント
- ・品質マネジメント
- ・ソフトウェアエルゴノミクス
- ・システムエンジニアリング



- ⑧ V3.0では、ISO/IEC規格、IEEE規格との関係、特にプロフェッショナル認証に関わるものを紹介。

V3.0における主たる追加副知識領域(1/14)



- ① 第1章：ソフトウェア要求

副知識領域は変わらないが、以下の形式手法に関係するトピックス追加。

- 1) 要求の記述
- 2) 要求の検証

V3.0における主たる追加副知識領域(2/14)



② 第2章： ソフトウェア設計

1)「ソフトウェア設計における主要な問題」副知識領域に以下のトピックス追加。

ーセキュリティ

2)新副知識領域

ーユーザーインターフェース設計

V3.0における主たる追加副知識領域(3/14)



③ 第3章： ソフトウェア構築

1)「ソフトウェア構築の基礎」副知識領域に追加
ー再利用

2)「構築の技術」副知識領域の追加
ーカバーしている新トピックス

API設計と活用、OO実行時の留意点、
パラメーター化、表明・契約による設計・防
護的プログラミング、エラー処理・例外
処理・耐障害性、実行可能モデル等16技
術



④ 第4章： ソフトウェアテストニング

1)「テスト技法」副知識領域に追加

—Model-based(モデルベース):

要求時の形式手法記述に対応するテスト技法



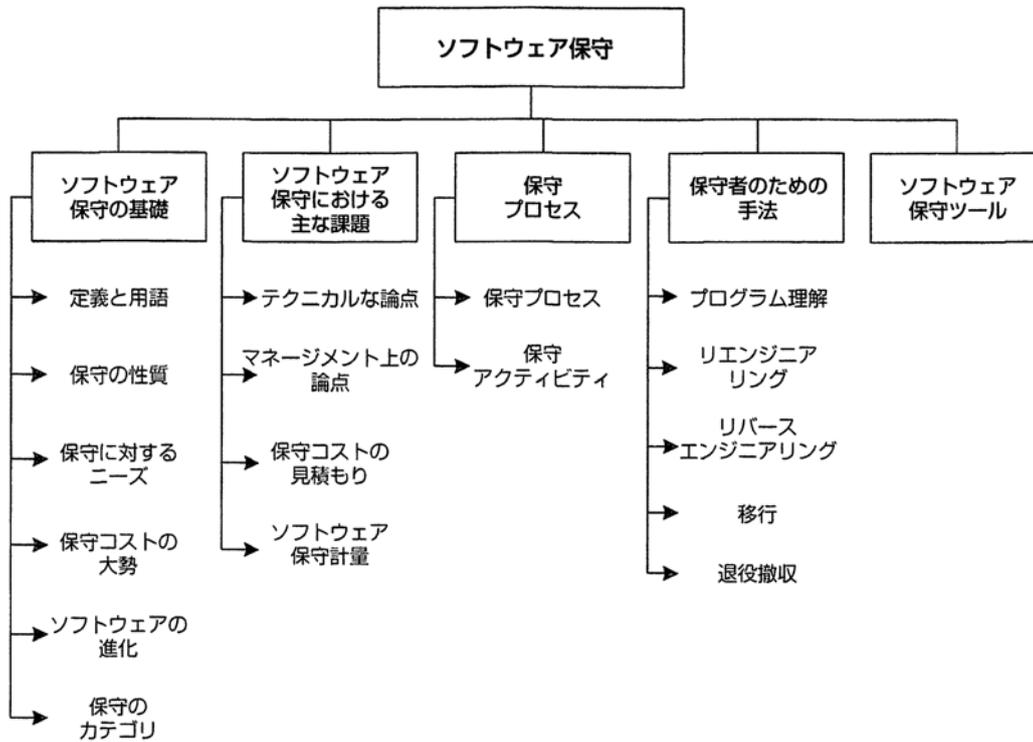
⑤ 第5章： ソフトウェア保守

1)「保守のための技法」副知識領域に追加

—マイグレーション: ソフトウェアのライフ中に異なる環境で運用される場合の考慮点

—除却: 運用しないという決定への考慮点

第5章: ソフトウェア保守(1/3)一構造図



第5章: ソフトウェア保守(2/3)一参照資料



派生的読み物

A. April and A. Abran, *Software Maintenance Management: Evaluation and Continuous Improvement* [6].

この書籍は、小規模ソフトウェア保守プロセス (S3M) の領域を探索し、組織のなかにおけるソフトウェア保守プロセスを改善するために辿るべきロードマップを提供している。この書籍は、ベンチマーク作りおよび継続的改善を可能にする水準に従って組織化された、ソフトウェア保守に限定した成熟度モデルを記述している。主要な実践慣行領域 (practice areas) それぞれに対する目標 (goals) が提供され、ここで提示されているプロセスモデルは、国際標準 ISO12207, ISO14764 および ISO15504, ならびに ITIL, CoBIT, CMMI および CM3 のような一般的に認められた成熟度モデルのアーキテクチャおよびフレームワークと、十分に整合されている。

M. Kajko-Mattsson, "Towards a Business Maintenance Model," *IEEE Int'l Conf. Software Maintenance* [7].

参照資料

[1*] IEEE Std. 14764-2006 (a.k.a. ISO/IEC 14764:2006) *Standard for Software Engineering—Software Life Cycle Processes—Maintenance*, IEEE, 2006.

[2*] P. Grubb and A.A. Takang, *Software Maintenance: Concepts and Practice*, 2nd ed., World Scientific Publishing, 2003.

[3*] H.M. Sneed, "Offering Software Maintenance as an Offshore Service," *Proc. IEEE Int'l Conf. Software Maintenance (ICSM 08)*, IEEE, 2008, pp. 1–5.

[4*] J.W. Moore, *The Road Map to Software Engineering: A Standards-Based Guide*, Wiley-IEEE Computer Society Press, 2006.

[5] *ISO/IEC/IEEE 24765:2010 Systems and Software Engineering—Vocabulary*, ISO/IEC/IEEE, 2010.

[6] A. April and A. Abran, *Software Maintenance Management: Evaluation and Continuous Improvement*, Wiley-IEEE Computer Society Press, 2008.

[7] M. Kajko-Mattsson, "Towards a Business Maintenance Model," *Proc. Int'l Conf. Software Maintenance*, IEEE, 2001, pp. 500–509.



トピックスと参照資料の
対照表

	IEEE/ISO/IEC 14764 2006 [1*]	Grubb and Takang 2003 [2*]	Sneed 2008 [3*]
1 ソフトウェア保守の基礎			
1.1 定義と用語	c3	c1s2, c2s2	
1.2 保守の性質		c1s3	
1.3 保守に対するニーズ		c1s5	
1.4 保守コストの大勢		c4s3, c5s5.2	
1.5 ソフトウェアの進化		c3s5	
1.6 保守のカテゴリ	c3, c6s2	c3s3.1, c4s3	
2 ソフトウェア保守における主な課題			
2.1 テクニカルな論点			
2.1.1 限られた理解		c6	
2.1.2 テスティング	c6s2.2.2	c9	
2.1.3 影響分析	c5s2.5	c13s3	
2.1.4 保守容易性	c6s8, c3s4	c12s5.5	
2.2 マネージメント上の論点			
2.2.1 組織目的との整合		c4	
2.2.2 要員配置		c4s5, c10s4	
2.2.3 プロセス	c5	c5	
2.2.4 保守に対する組織的側面	c7s.2.3	c10	
2.2.5 外部委託			all
2.3 保守コストの見積もり			
2.3.1 コスト見積もり	c7s4.1	c7s2.4	
2.3.2 パラメトリックモデル		c12s5.6	
2.3.3 経験		c12s5.5	
2.4 ソフトウェア保守計量	c6s5	c12, c12s3.1	
2.4.1 特化された計量尺度		c12	
3 保守プロセス			
3.1 保守プロセス	c5	c5	
3.2 保守アクティビティ	c5, c5s3.2.2, c6s8.2, c7s3.3		
3.2.1 特有のアクティビティ	c3s10, c6s9, c7s2, c7s3	c6, c7	

2015/05/

25

V3.0における主たる追加副知識領域(6/14)



⑥ 第6章: ソフトウェア構成管理

⑦ 第7章: ソフトウェアエンジニアリング・マネジメント

* : 共に構成上の改訂無し

V3.0における主たる追加副知識領域(7/14)



⑧ 第8章： ソフトウェアエンジニアリングプロセス

1) 副知識領域が、以下のように全面的に改訂。

- ープロセス定義
- ーソフトウェアライフサイクル
- ーソフトウェアプロセスアセスメント及び改善
- ーソフトウェア測定

* :現在、ISOにおいて、ソフトウェアを扱う12207、システムを扱う15288が改訂作業中。統合の方向。

V3.0における主たる追加副知識領域(8/14)



⑨ 第9章： ソフトウェアエンジニアリングモデル及び手法

1) 知識領域名が変わり、ツールは各知識領域の副知識領域に移動すると共に、本知識領域の副知識領域は以下に変更

ーモデリング

モデリングの基礎、モデルの特性と記述、シンタックス・意味と語用、事前条件・事後条件・不変条件

ーモデルの型

主としてUMLを使用する時の図紹介

ーモデルの分析

対象の有効な分析にモデルを使用することを説明

ーソフトウェアエンジニアリング手法

形式手法に関する記述拡充とアジャイルを追加



⑩ 第10章： ソフトウェア品質

- 1)「ソフトウェア品質の基礎」副知識領域に新トピックス
ーソフトウェアセーフティ

2015/05/28

新谷ITコンサルティング

29

V3.0における主たる追加副知識領域(10/14)



- ⑪-1 2004年版「第12章ソフトウェアエンジニアリングに
関連するディシプリン」は、V3.0で第11章～第15
章に拡大し新知識領域をカバー

- 1) 第11章： ソフトウェアエンジニアリングプロフェッショナルプ
ラクティス

以下の副知識領域をカバー

- ープロフェッショナリズム
- ーグループダイナミックスと心理学
- ーコミュニケーション・スキル

2015/05/28

新谷ITコンサルティング

30



⑪-2 2004年版「第12章ソフトウェアエンジニアリングに関連するディシプリン」は、V3.0で第11章～第15章に拡大し新知識領域をカバー

1) 第12章: ソフトウェアソフトウェアエンジニアリング
エコノミックス

以下の副知識領域をカバー

- ソフトウェアエンジニアリングエコノミックスの基礎
- ライフサイクルエコノミックス
- リスクと非確定性
- エコノミックス分析手法
- 実践的考慮事項

V3.0における主たる追加副知識領域(12/14)



⑪-3 2004年版「第12章ソフトウェアエンジニアリングに関連するディシプリン」は、V3.0で第11章～第15章に拡大し新知識領域をカバー

1) 第13章: コンピューティング基礎

以下の副知識領域をカバー

- 問題解決手法、抽象化、プログラミング基礎、プログラミング言語基礎、デバッグツールと手法、データ構造と表現、アルゴリズムと複雑性、システムの基本的概念、コンピューターの構造、コンパイラ基礎、オペレーティングシステムの基礎、データベースの基礎とデータ管理、ネットワークの基礎、並列及び分散コンピューティング、ユーザー人間工学の基礎、開発者人間工学の基礎、確実なソフトウェア開発と維持

V3.0における主たる追加副知識領域(13/14)



⑪-4 2004年版「第12章ソフトウェアエンジニアリングに関連するディシプリン」は、V3.0で第11章～第15章に拡大し新知識領域をカバー

1) 第14章： 数学基礎

以下の副知識領域をカバー

一集合・関係・機能、論理基礎、証明手法、集計の基礎、離散確率、有限状態機械、文法、正確性・精密性と誤差、整数論、代数構造

V3.0における主たる追加副知識領域(14/14)



⑪-5 2004年版「第12章ソフトウェアエンジニアリングに関連するディシプリン」は、V3.0で第11章～第15章に拡大し新知識領域をカバー

1) 第15章： エンジニアリング基礎

以下の副知識領域をカバー

一実験法、統計分析、測定、エンジニアリング設計、モデリング・シミュレーション・プロトタイピング、標準、根本原因分析

ソフトウェアプロフェッショナル認証(1/4)



SWEBOKの開発に当たり、以下の5つの目的あり。

1. ソフトウェアエンジニアリングに関して全世界に渡り整合性ある見方を推進
2. コンピューター科学等の他のディシプリンとの比較において、ソフトウェアエンジニアリングを明確化
3. ソフトウェアエンジニアリングディシプリンの内容を明確化
4. SWEBOKがトピックスとしてとりあげているものにアクセスできるようにする
5. 教材開発及び個人の認証とそのため資料のライセンス付与の基盤を提供する。

2015/05/28

新谷ITコンサルティング

35

ソフトウェアプロフェッショナル認証(2/4)



ISO/IEC 24773:2008 Software Engineering – Certification of Software Engineering Professionals

— 現在、SC7/WG20で改訂作業が進行中で、要件の1つはSWEBOK V3.0の内容を理解していること。

— 問題: 1) 日本の大学でSWEBOK V3.0の内容を教えているのか?

2) 日本における発注者、受注者あるいは第三者がそのような知見を持つエンジニアを期待しているか?

2015/05/28

新谷ITコンサルティング

36

ソフトウェアプロフェッショナル認証(3/4)



第11章 ソフトウェアエンジニアリングプロフェッショナルプラクティス

1.1.2 認証

認証の無いことが個人がソフトウェアエンジニアとして仕事ができないということではない。現在ソフトウェアエンジニアリングへの認証は全く任意である。実際問題として、どのようなプログラム下であれ、殆どのソフトウェアエンジニアは認証されていない。

1.7.6 プロフェッショナルとしての責任

法に基づけば、ソフトウェアエンジニアは一般的に受け入れられているプラクティスに準拠することが期待されている。(期待に反することを「ネグリジェンス＝怠慢」という。)

ソフトウェアプロフェッショナル認証(4/4)



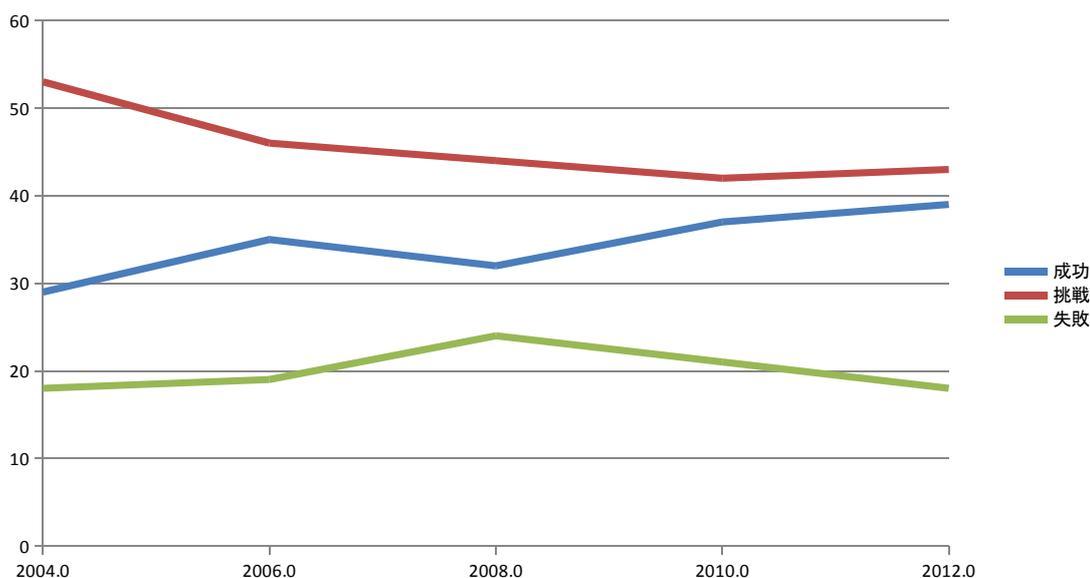
第11章 ソフトウェアエンジニアリングプロフェッショナルプラクティス

1.7.6 法的要求

ソフトウェアエンジニアは、地域、国、国際環境下の法的制約の中で活動しなければならない。よって、ソフトウェアエンジニアは、以下の法的要求を認識している必要あり。

- 登録とライセンス(試験、訓練等を含む)
- 契約条項
- ガバナンスのような非契約の責任事項
- WTOにより規制される国際的法的フレームワーク

ソフトウェアエンジニアリングは効果をあげてきたか？



Standish Chaos Manifesto 2013

2015/05/28

新谷ITコンサルティング

39

ご清聴有り難うございます



- ご質問等、以下にお寄せ下さい

katsu.shintani@k3.dion.ne.jp

2015/05/28

新谷ITコンサルティング

40

SERCフォーラム
ソフトウェア保守技術者の技術力評価を考える

研究報告(1)

SWEBOK V3.0「第5章 ソフトウェア保守」と
ソフトウェア保守技術者向けシラバスの解説



ソフトウェア・メンテナンス研究会
SERC (Software Evolution Research Consortium)

幹事 馬場 辰男

発表内容

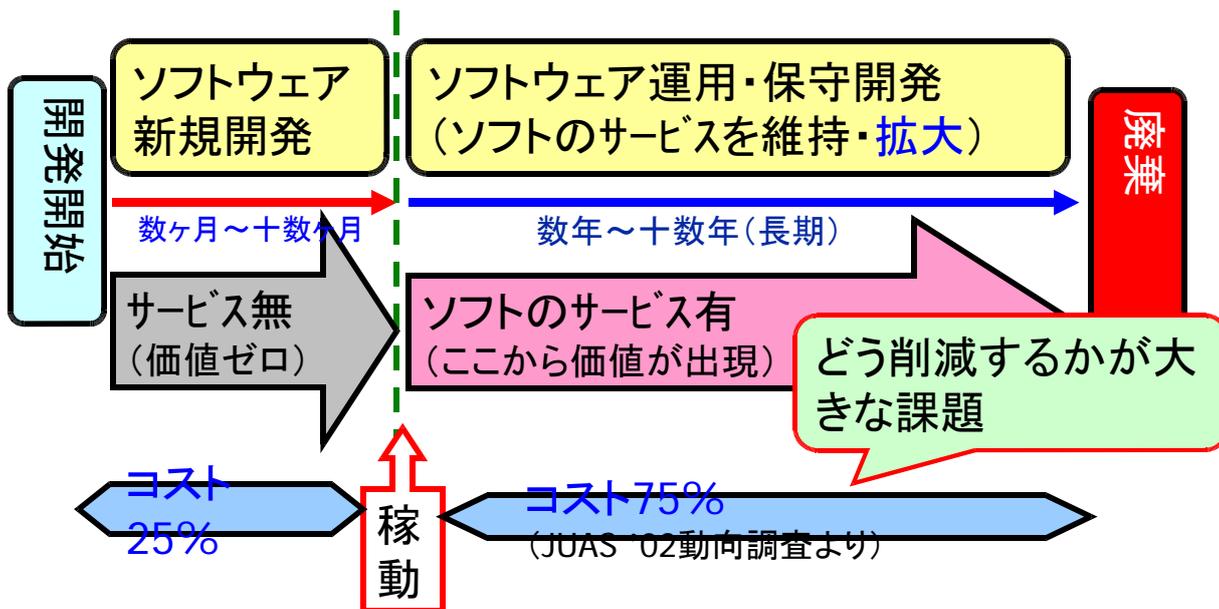
- ソフトウェア保守の認識 SWEBOK V3.0とSERC
- 「ソフトウェア保守」知識領域のトピックス分類
- ソフトウェア保守技術者向けシラバスの解説
- 参考図書と翻訳用語の相違

SWEBOK V3.0 ソフトウェア保守の認識

- ソフトウェア保守は、ソフトウェアライフサイクルのなかで**全体の統合**に携わる部分の1つ
- しかし、他のフェーズに比べて同程度の注目すら得ることがなかった。歴史的に、ほとんどの組織で、ソフトウェア開発はソフトウェア保守に比べ、ずっと高い格を持つ像として描かれてきた。
- いまや事態は変わって、組織は、ソフトウェアをできるだけ永く運用して、開発に投じた資金を少しでも多く回収しようとするようになった。

SERC見解:

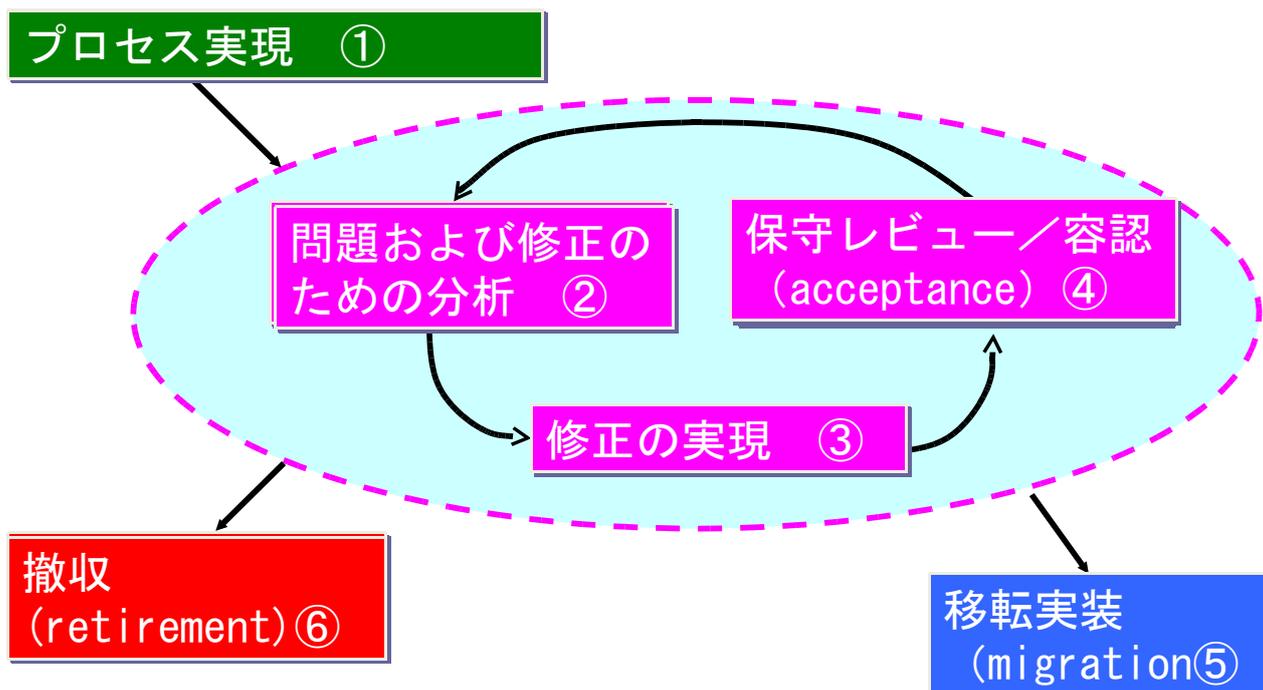
<ソフトウェアの価値は稼働後から生まれる>



SWEBOK V3.0 ソフトウェア保守の認識(続き)

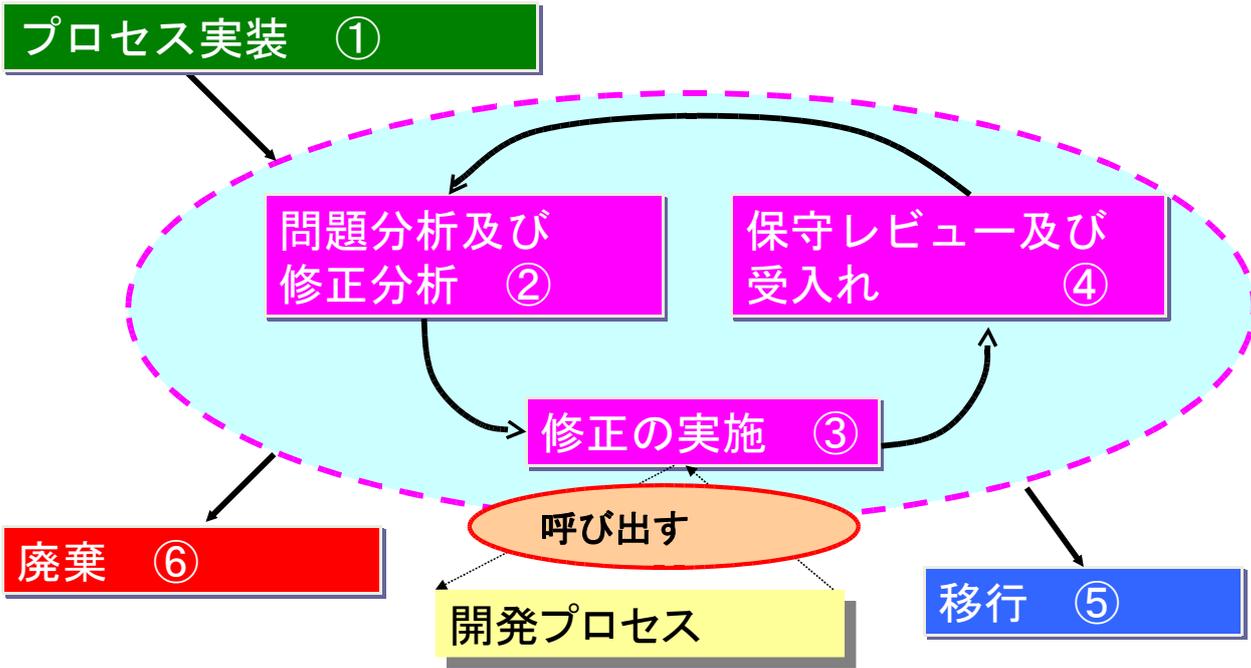
- **ソフトウェア保守**は、ソフトウェアに対する費用-効果的サポートを提供するために必要な**アクティビティ全体**であると定義されている。
- 「ソフトウェア保守」知識領域は、ソフトウェアエンジニアリングにおける他の**すべての側面と関係**を持っている。

SWEBOK V3.0 ソフトウェア保守プロセス

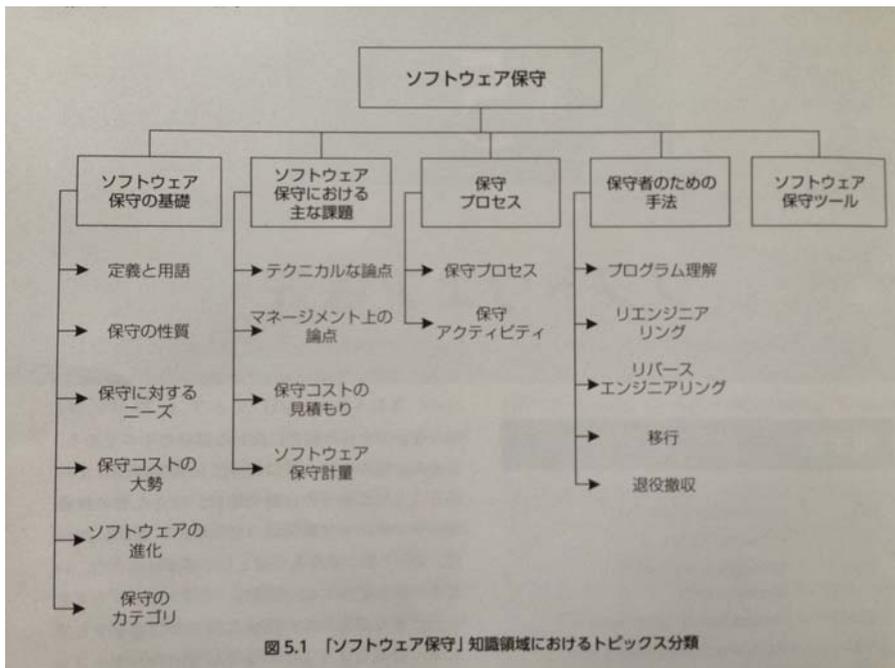


SERC見解:

ソフトウェア保守プロセス (JIS X 0161:2008)

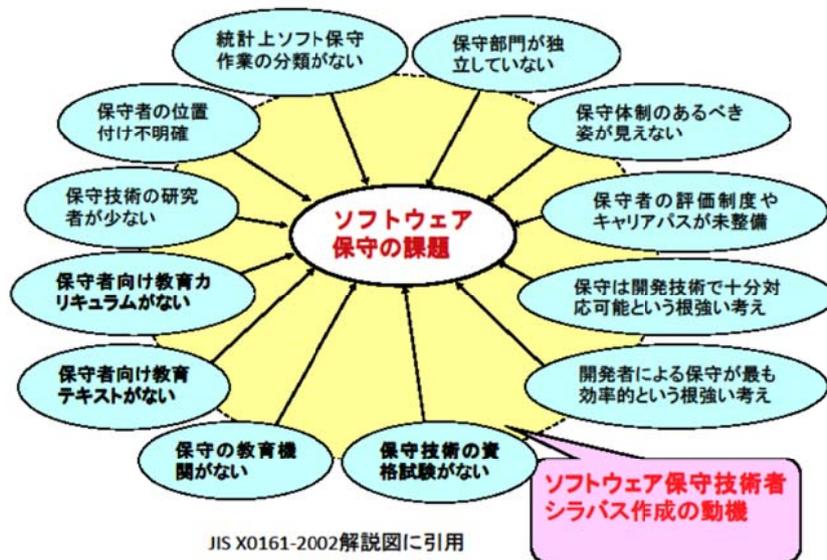


「ソフトウェア保守」知識領域のトピックス分類



ソフトウェア保守技術者向けシラバスの解説

- ソフトウェア保守技術者の教育体系の必要性



ソフトウェア保守技術者向けシラバスの解説

- ソフトウェア保守技術者の教育体系の必要性
- SWEBOK V3.0(ソフトウェアエンジニアリング基礎知識体系)をベースに昨年度シラバスを作成
- SWEBOK V3.0日本語解説本、これまでのSERCのソフトウェア保守研究の成果である以下の図書を参考文献として今年度見直し

JIS X 0161:2008 (ISO/IEC 14764:2006)

ソフトウェアライフサイクル プロセス—保守

～ISO14764による～ ソフトウェア保守開発 (SRC)

ソフトウェア保守技術者向けシラバスの例

1. ソフトウェア保守の基礎(抜粋)

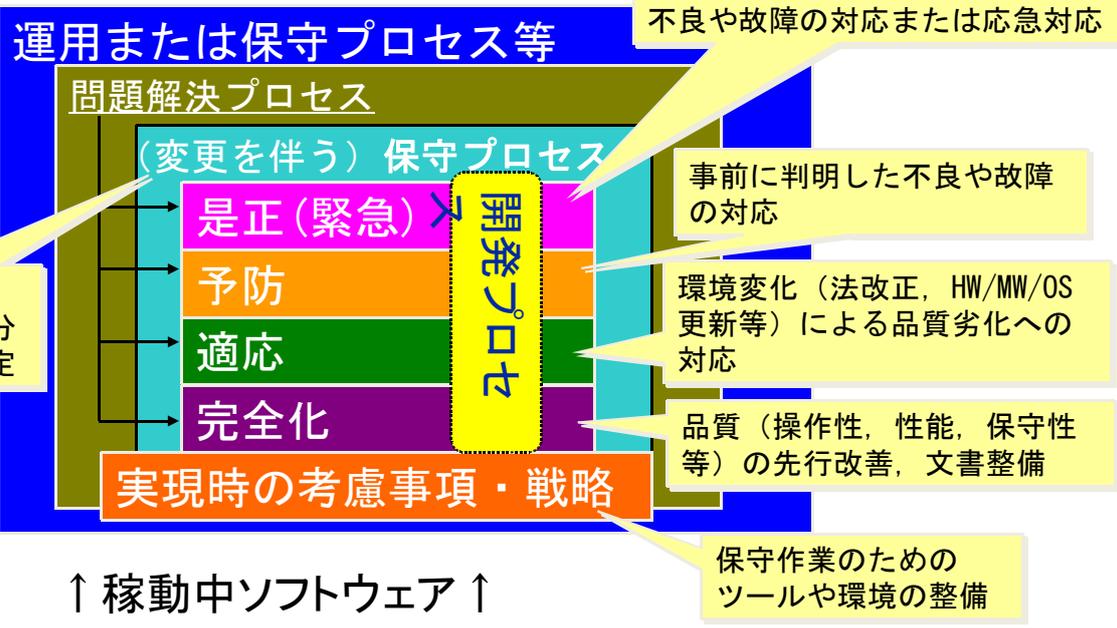
項目	学習目標	学習対象となる用語、概念	知識レベル
1. ソフトウェア保守の基礎			
1.1 定義と用語	<ul style="list-style-type: none"> ソフトウェア保守の定義 (ISO/IEC/IEEE 14764とIEEE/EIA 12207で同様)を理解する。 ソフトウェア保守の目的 (IEEE 14764)を理解する。 	<ul style="list-style-type: none"> ISO/IEC/IEEE 14764、IEEE/EIA 12207 ソフトウェアライフサイクル・プロセス、保守プロセス ソフトウェア保守、ソフトウェアの引き渡し 	L1
1.2 保守の性質	<ul style="list-style-type: none"> ソフトウェア保守が、そのライフサイクルを通してソフトウェアプロダクトを支えることを理解する。 保守者の定義 (保守アクティビティを実行する組織)を理解する。なお、SWEBOOK (V3) では、個人を指す場合がある。 ソフトウェア保守の基本的アクティビティ (IEEE 14764)を理解する。 保守者からみた開発者との関係を理解する。 	<ul style="list-style-type: none"> 保守者、開発者 保守アクティビティ 	L1
1.3 保守に対するニーズ	<ul style="list-style-type: none"> 保守の必要性 (ソフトウェアがユーザ要求を継続して満たしていることを保証する)を理解する。 保守は、開発時に使われたライフサイクルモデルにかかわらず適用されることを理解する。 保守の目的 (フォールトの是正、設計の改善、エンハンスの実施、他のソフトウェアとのインターフェースの維持、様々なハード・ソフト・システム特徴及び通信装置を使えるようにプログラムを適合させること、遺産ソフトウェアの移行、ソフトウェアの廃棄)を理解する。 保守者のアクティビティを構成する5つの重要な特徴を理解する。 	<ul style="list-style-type: none"> ユーザ要求 増補 (enhancement) と是正 (correction) システム特徴 (features) 遺産ソフトウェア (legacy software) 退役撤収 (retire) 	L1

SWEBOK V3.0 ソフトウェア保守カテゴリ

	是正カテゴリ	増補カテゴリ
プロアクティブ (先んじて実施)	予防保守	完全化保守
リアクティブ (事後応答的实施)	是正保守	適応保守

SERC: 保守の型 (タイプ)

- 是正保守・問題を訂正するための受身の修正
- 緊急保守・是正保守で緊急対応を要する修正
- 予防保守・潜在的な障害顕在化前の修正
- 適応保守・環境の変化に対して適応する修正
- 完全化保守・予防的な適応保守や将来に対する性能や品質の改良(含むリファクタリング)



ソフトウェア保守技術者向けシラバスの例 2. ソフトウェア保守における主な課題(抜粋)

項目	学習目標	学習対象となる用語、概念	知識レベル
2. ソフトウェア保守における主な課題			
2. 1 テクニカルな論点			
2. 1. 1 限られた理解	<ul style="list-style-type: none"> 限られた理解 (limited understanding) という概念を理解する。 全保守工数の半分がソフトウェア理解のための割かれているという研究者の示唆に留意する。 ソフトウェア・エンジニアにとって、ソフトウェアを理解することは重要な関心事であることを理解する。 	<ul style="list-style-type: none"> 限られた理解 (limited understanding) ソフトウェア理解 (software comprehension) 	L1
2. 1. 2 テスティング	<ul style="list-style-type: none"> 保守者は、問題報告が有効であるかを確認するために、テストを実行して問題の再現を検証することが必要であることを理解する。 保守にとって、回帰テストは重要なテストであることを理解する。 テストのための時間を見つけることは、しばしば難しく、保守チーム内の異なるメンバーがそれぞれ異なる問題の解決に同時に取り組んでいる場合には、協調という問題がさらに加わることを理解する。 重要な機能を、テストのためにオフラインにすることは出来ないことを理解する。 	<ul style="list-style-type: none"> 問題報告 回帰テスト 	L1
2. 1. 3 影響分析	<ul style="list-style-type: none"> 影響分析では、既存のソフトウェアの変更の影響についての完全な分析を、どのようにして費用-効果的に行うかが重要であることを理解する。 影響分析には、ソフトウェアの構造と内容の詳細な知識が必要であることを理解する。 保守者は、変更要請に対する影響分析を行い、必要な資源の見積もるとともに、生じるリスクを特定する必要があることを理解する。 影響分析のタスク (IEEE 14764) を理解する。 	<ul style="list-style-type: none"> 変更要請、修正要請 (MR)、問題報告 (PR) 	L1

ソフトウェア保守技術者向けシラバスの例

3. 保守プロセス(抜粋)

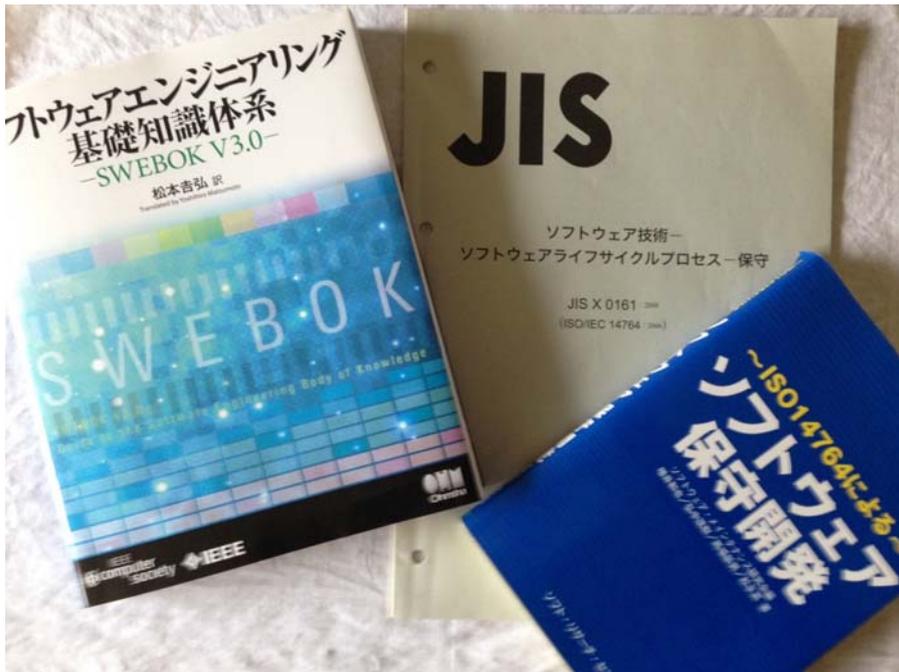
項目	学習目標	学習対象となる用語、概念	知識レベル
3. 保守プロセス			
3.1 保守プロセス	<ul style="list-style-type: none"> IEEE 14764に記載されているソフトウェアの保守プロセス(プロセス実現、問題及び修正のための分析、修正の実現、保守レビュー/承認、用済みソフトウェアの撤収、移転実装)を理解する。 その他の保守プロセスモデル(敏捷補修、螺旋状、オズボーン型、反復増補、再利用志向)を理解する。併せて、近來、アジャイル方法論が保守にも適用されるようになってきたことを理解する。 保守プロセスの改善を支持する方法(ソフトウェア保守能力成熟度モデル)について、理解する。 	<ul style="list-style-type: none"> プロセス実現、問題及び修正のための分析、修正の実現、保守レビュー/承認、用済みソフトウェアの撤収、移転実装 敏捷補修、螺旋状、オズボーン型、反復増補、再利用志向 アジャイル ソフトウェア保守能力成熟度モデル 	L1
3.2 保守アクティビティ			
3.2.1 特有のアクティビティ	<ul style="list-style-type: none"> 保守に特有のアクティビティ(プログラム理解、移行、修正要請の受け入れ/拒絶、ヘルプデスク、影響分析、保守のSLA及び保守許諾・契約)があることを理解する。 	<ul style="list-style-type: none"> 健全性 (integrity) 仕立て (tailor) プログラム理解、移行 (transition)、修正要請の受け入れ/拒絶、ヘルプデスク、影響分析、保守のSLA及び保守許諾・契約 	L1
3.2.2 支援アクティビティ	<ul style="list-style-type: none"> 保守の支援アクティビティ(文書化、構成管理、検証と妥当性確認、問題解決、品質保証レビュー及び監査、に追加訓練)を理解する。 	<ul style="list-style-type: none"> 文書化、構成管理、検証と妥当性確認、問題解決、品質保証レビュー及び監査、に追加訓練 	L1
3.2.3 保守計画アクティビティ	<ul style="list-style-type: none"> 保守の計画見直しに関する論点(ビジネス計画、保守計画、リリース/版計画、個々の変更要請計画など)を理解する。また、リリース/版計画で保守者に要求される事項を理解する。 保守計画の手順(保守構想→ソフトウェア保守計画)とその概要を理解する。 最終的に、ビジネス計画アクティビティ(予算的資源、財政的資源、人的資源の計画)を実行しなければならないことを理解する。 	<ul style="list-style-type: none"> ビジネス計画、保守計画、リリース/版計画、個々の変更要請計画 保守構想 (maintenance concept) 	L1

ソフトウェア保守技術者向けシラバスの例

4. 保守者のための手法(抜粋)

項目	学習目標	学習対象となる用語、概念	知識レベル
4. 保守者のための手法			
4.1 プログラム理解	<ul style="list-style-type: none"> プログラマが、変更の実現のために、プログラム理解(プログラムを読み、解明すること)に極めて長い時間を費やすことを理解する。 プログラム理解の主要ツール(コードブラウザ)を理解する。 	<ul style="list-style-type: none"> プログラム理解 (program comprehension) 	L2
4.2 リエンジニアリング	<ul style="list-style-type: none"> リエンジニアリングの定義(ソフトウェアを新しい形式に再構成するための調査と改造)を理解する。併せて、一般的に保守容易性の改善のためには実施されないことを理解する。 リファクタリング(プログラムの構造的変換を必要とせず再組織化すること)が、リエンジニアリング手法の一つであることを理解する。併せて、その狙いがプログラム構造及び保守容易性の改善であることを理解する。 	<ul style="list-style-type: none"> リエンジニアリング リファクタリング 調査 (examination)と改造 (alteration) 再構成、再組織化 	L2
4.3 リバースエンジニアリング	<ul style="list-style-type: none"> リバースエンジニアリングの定義(ソフトウェアのコンポーネントおよびそれらの相互関係を識別し、別の形式でより高い抽象レベルで表現するために、ソフトウェア分析プロセス)を理解する。 リバースエンジニアリングの特徴(受動的)、生成物(コールグラフ及び制御フロー)及びタイプ(再文書化、設計回復)を理解する。 データ・リバースエンジニアリング(物理的アーキテクスから論理スキーマを回復しようとするもの)が、ここ数年間で重要性を認められてきたことを理解する。 	<ul style="list-style-type: none"> リバースエンジニアリング コールグラフ、制御フローグラフ 再文書化、設計回復 データ・リバースエンジニアリング 	L2

シラバスとサンプル問題の参考図書



参考図書間の翻訳用語の相違

同一用語(英語)のSWEBOK V3.0 日本語翻訳とJIS X0161の相違一覧から抜粋

↓ SWEBOK V3.0 日本語版
で同一英単語に異なる日本語訳がある場合、それぞれを提示

No.	原用語(英文)	JIS X0161及び文献1)の日本語訳	SWEBOK V3.0 第5章 日本語版	SWEBOK V3.0 日本語訳出現箇所
8	change	変更	変化 変更	序説, 2.1.4 多数
9	change request	変更要求	変更要請	2.1.3, 2.3.1, 3.2.4
14	correction	訂正	是正	1.4, 1.6, 2.1.1, 2.1.4
20	enhancement	改良	増補	1.3, 1.4, 1.6, 2.2.1, 3.1
25	implementation	実装	実現	多数
35	migration	移行	移転	1.2
			移転実装	3.1
			移行	4.4
37	Modification Implementation	修正の実施	是正の実現	1.2
			修正の実現	3.1
38	modification request	修正依頼	是正要請 修正要請	1.2 2.1.3, 3.2.1, 3.2.2, 3.2.4
43	process implementation	プロセス実装	プロセス実現	1.2, 3.1
47	retirement	廃棄	退役撤収	1.2, 4.5
			用済みソフトウェアの撤収	3.1

文献1) 増井, 馬場他著「～ISO14764による～ ソフトウェア保守開発」2007 ソフトリサーチセンター



ソフトウェア保守技術者の 技術力評価に向けて

ソフトウェア・メンテナンス研究会

研究員 大島道夫

2015年5月28日

本日、お話ししたいこと

1. 保守の現状を見てみよう。
2. 今、私達は何をすべきか？
3. しかし、実態は・・・
4. だからソフトウェア保守技術者の技術力評価が必要なのです。
5. 今後もソフトウェア保守技術者の技術力評価活動へのご支援をよろしくお願いいたします。

1.保守の現状を見てみよう。

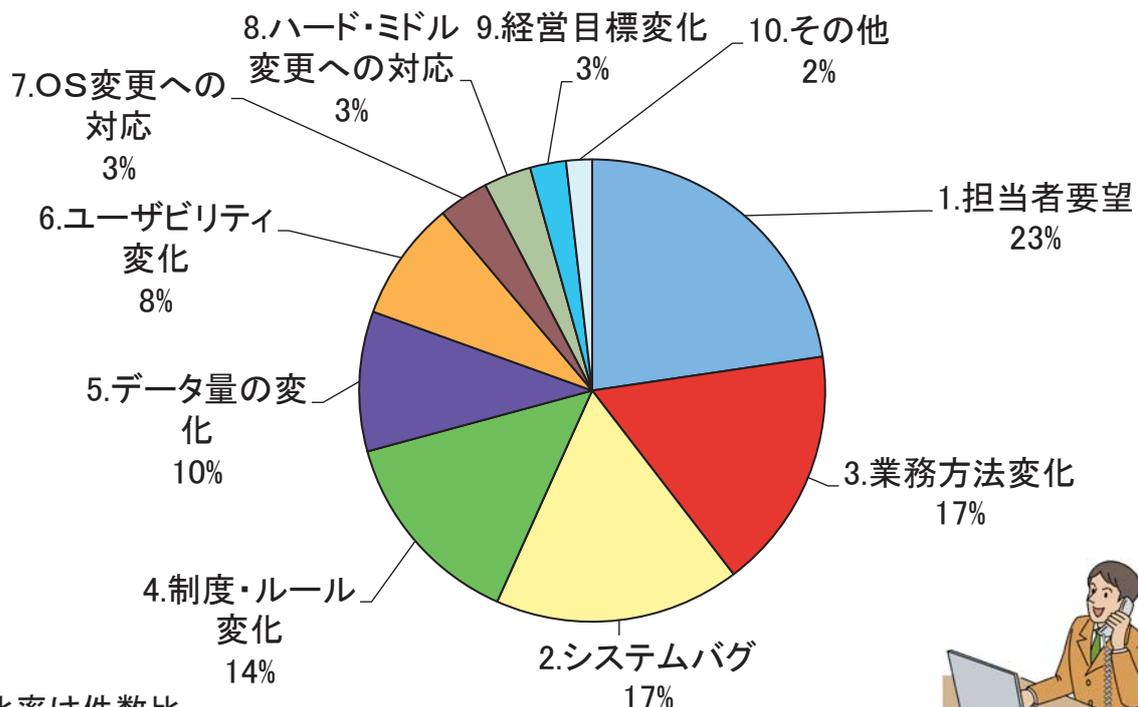
2. 今、私達は何をすべきか？
3. しかし、実態は・・・
4. だからソフトウェア保守技術者の技術力評価が必要なのです。
5. 今後もソフトウェア保守技術者の技術力評価活動へのご支援をよろしくお願いいたします。

(1)保守作業の理由(件数比)



回答数328

JUAS=社団法人日本情報システム・ユーザー協会(ソフトウェアメトリックス調査2013)



(注)比率は件数比

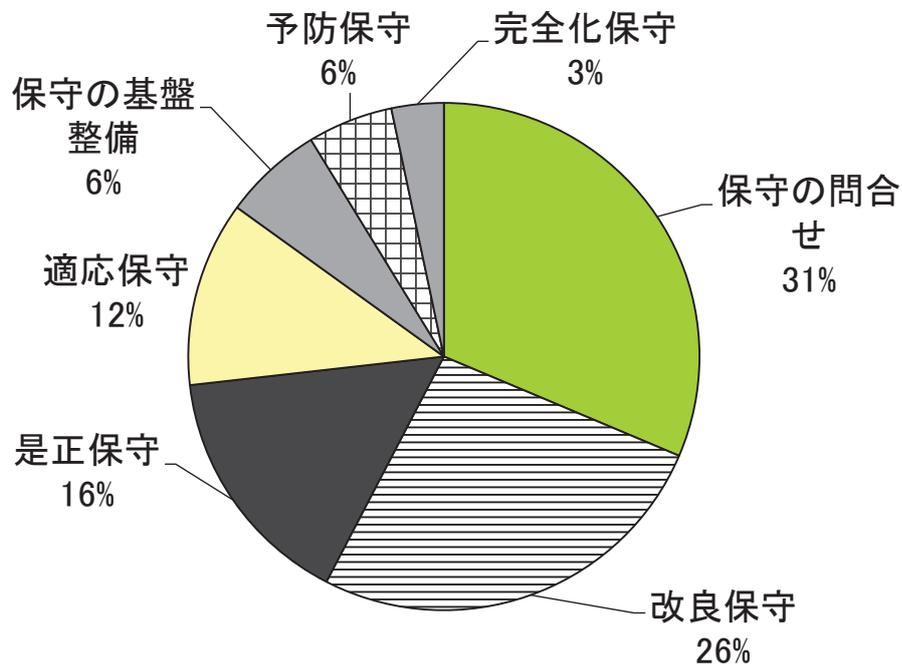


(2)保守作業の工数割合



回答数387

(出典:JUASソフトウェアメトリクス調査2014)

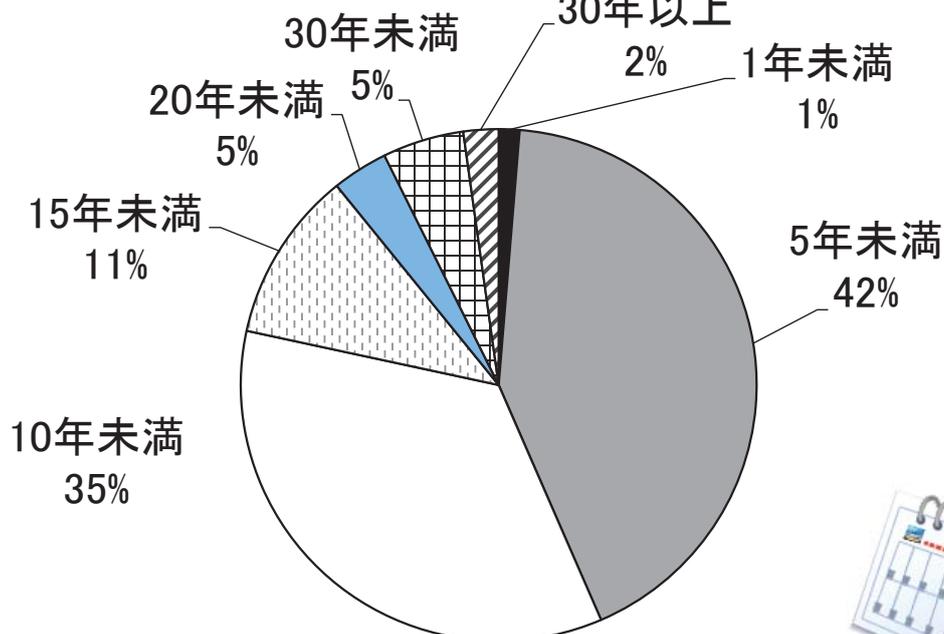


(3)保守要員の平均経験年数



回答数315

(出典:JUASソフトウェアメトリクス調査2014)



平均値7.0年、中央値5.0年
(前年調査より平均値が2年延びている)



(4)ソフトウェア保守プロセスの整備状況

選択肢	(全体)		(委託側)		(受託側)	
	組織数	比率	組織数	比率	組織数	比率
整備済みである	37	31.6%	9	39.1%	28	30.8%
整備中または整備予定である	28	23.9%	1	4.3%	27	29.7%
整備していない	48	41.0%	13	56.5%	33	36.3%
その他	4	3.4%	0	0.0%	3	3.3%
合計	117	100.0%	23	100.0%	91	100.0%

出典：財団法人経済調査会「平成24年度ソフトウェア保守に関する調査集計結果」

注：この調査は、ソフトウェアメンテナンス研究会等との共同調査である。

(5)改善の推進体制

選択肢	(全体)		(委託側)		(受託側)	
	組織数	比率	組織数	比率	組織数	比率
改善専任の担当がいる	7	10.8%	0	0.0%	7	12.7%
専任ではないが改善担当がいる	30	46.2%	5	50.0%	25	45.5%
改善担当はいない	25	38.5%	4	40.0%	21	38.2%
その他	3	4.6%	1	10.0%	2	3.6%
合計	65	100.0%	10	100.0%	55	100.0%

出典：財団法人経済調査会「平成24年度ソフトウェア保守に関する調査集計結果」

注：この調査は、ソフトウェアメンテナンス研究会等との共同調査である。

1. 保守の現状を見てみよう。

2. 今、私達は何をすべきか？

3. しかし、実態は・・・

4. だからソフトウェア保守技術者の技術力評価が必要なのです。

5. 今後もソフトウェア保守技術者の技術力評価活動へのご支援をよろしくお願いいたします。

私の提言

1 保守作業の理由は、
担当者要望、業務方法変化 40%

2 保守作業の工数割合
問い合わせ、改良保守 57%

3 保守要員の平均経験
年数 平均7年

4 保守プロセス整備済
み 30%

5 保守プロセスの専任
体制 13%

- 業務視点で積極的な改善提案が求められているのではないかと？

- 平均7年近く経験していたら、これくらいできるのではないかと？

- 会社としても新規開発同様に保守プロセス整備のため専任者設置の投資も必要ではないかと？

1. 保守の現状を見てみよう。
2. 今、私達は何をすべきか？

3.しかし、実態は・・・

4. だからソフトウェア保守技術者の技術力評価が必要なのです。
5. 今後もソフトウェア保守技術者の技術力評価活動へのご支援をよろしくお願いいたします。

皆さんの保守のイメージは・・・



ソフトウェア保守の課題

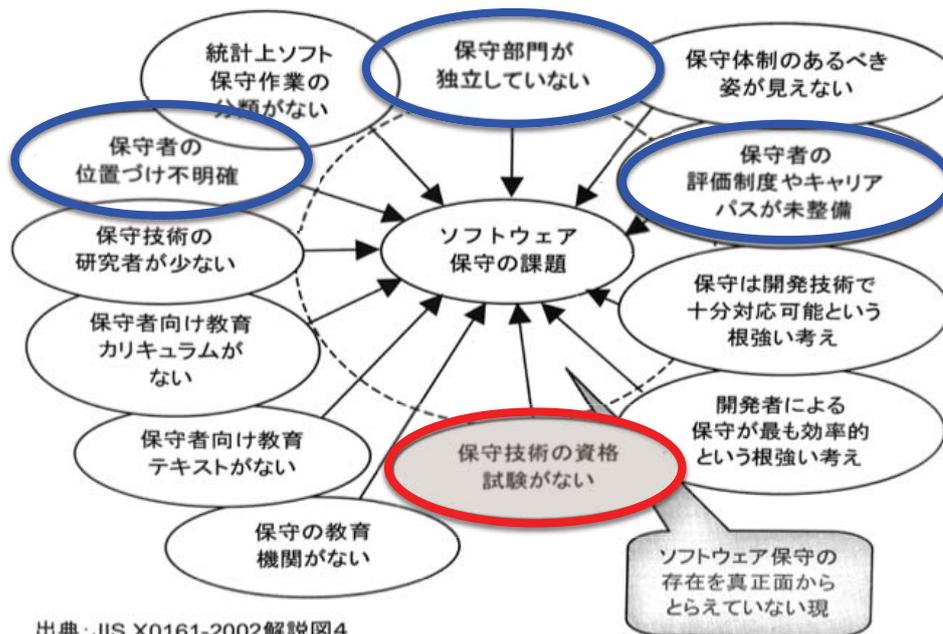


図 1.6.1 ソフトウェア保守の課題

ソフトウェア保守の未来は明るい！ ビジネスの宝庫です！

- 現在の日本全体のソフトウェア保守コストは、経済産業省の「情報処理実態調査」等を基に推計すると、約6兆円である。
- 現在の保守対象案件の保守コストが半減できれば、その金額は3兆円である。

- ユーザ企業の支払う年間ソフトウェア費用…… 約7～8兆円
- その内の保守費用の比率…… 70～80%
- 7～8兆円×70～80%…… 約5兆円～6兆円
- この2分の1は…… 約3兆円

- この3兆円を、ビジネスを支援する前向きな「システム強化」に振り向けることができれば、日本の産業の競争力強化に貢献できる。

なんと！ IT/UIスキル標準に保守の視点が無い。

ITSSの職種	UISSの職種
<ul style="list-style-type: none">• マーケティング• セールス• コンサルタント• プロジェクトマネジメント• ITアーキテクト• ITスペシャリスト• <u>アプリケーションスペシャリスト</u>• ソフトウェアデベロップメント• カスタマサービス• オペレーション• エデュケーション	<ul style="list-style-type: none">• ビジネスストラテジスト• ISストラテジスト• ISアナリスト• プログラムマネージャ• システムデザイナー• プロジェクトマネージャ• <u>アプリケーションデザイナー</u>• ISオペレーション• ISアドミニストレータ• ISアーキテクト
※アプリケーションスペシャリストに近いが、その担当業務に保守のことは記述されていない。	※アプリケーションデザイナーの担当業務はIS保守と書かれているが、内容の記述はない。

禁無断転載・複製 システム企画研修株式会社

1. 保守の現状を見てみよう。
2. 今、私達は何をすべきか？
3. しかし、実態は・・・

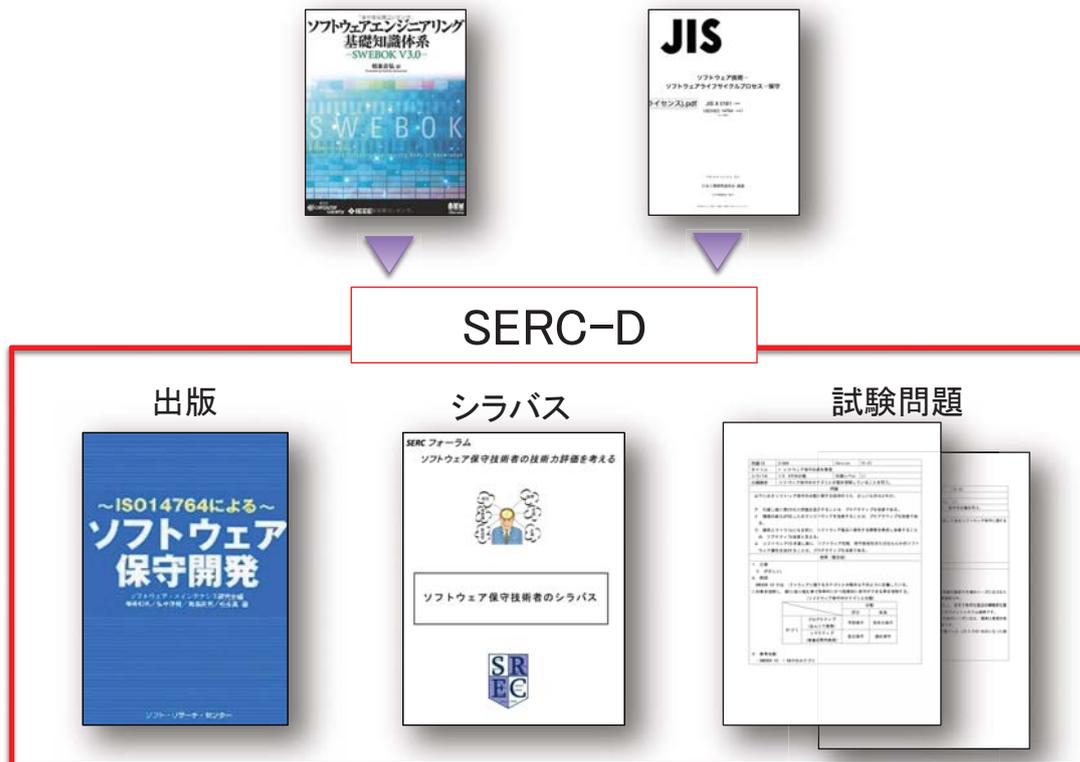
4.だからソフトウェア保守技術者の技術力評価が必要なのです。

5. 今後もソフトウェア保守技術者の技術力評価活動へのご支援をよろしくお願いいたします。

ソフトウェア保守技術者の技術力評価の実現



SERC-Dグループの成果が見えてきました。



1. 保守の現状を見てみよう。
2. 今、私達は何をすべきか？
3. しかし、実態は・・・
4. だからソフトウェア保守技術者の技術力評価が必要なのです。

**5. 今後もソフトウェア保守技術者の技術力評価活動へのご支援を
よろしくお願いいたします。**

メンテナンスからサービスへ

Enhancement
Service



Software
Maintenance

**今後もソフトウェア保守技術者の
技術力評価活動へのご支援をよろ
しくお願いいたします。**



ソフトウェア・メンテナンス研究会

研究報告(3)

技術力評価のためのサンプル試験問題



ソフトウェア・メンテナンス研究会
SERC (Software Evolution Research Consortium)

研究員 高橋 宏志

試験問題のレベル

知識レベル	補足説明
レベル1 (知っている)	概念や用語を知っており、その概要を述べることができる。
レベル2 (知識を説明できる)	概念や用語の意味や背景を理解しており、具体的な例を挙げて説明することができる。
レベル3 (概念と使い方がわかる)	概念や技術の使い方がわかっており、それらを適切に選択して、限られた条件の下で与えられた課題を解決できる。
レベル4 (詳しく理解し応用できる)	概念や技術を詳しく理解しており、実用的な問題を解決するために、その知識を応用できる。
レベル5 (熟達している)	実社会の複雑な問題に対して、構造を明らかにして要素に分解するとともに、解決に必要な検討を加えて結論を導くことができる。

サンプル試験問題と解説(7題)

3

問1 次のISO規格の内, ソフトウェア保守のプロセスが記述してあるソフトウェア保守に関する国際規格はどれか。

- ア ISO 9001
- イ ISO 13485
- ウ ISO 14001
- エ ISO/IEC 14764

4

問2 以下に示すソフトウェア保守の分類に関する説明のうち、正しいものはどれか。

- ア 引渡し後に発見された問題を是正することは、プロアクティブな改変である。
- イ 環境の変化が発生したのでソフトウェアを改変することは、プロアクティブな改変である。
- ウ 運用上でトラブルになる前に、ソフトウェア製品に潜在する障害を発見し改変することは、リアクティブな改変と言える。
- エ ソフトウェアの引き渡し後に、ソフトウェア性能、保守容易性またはなんらかのソフトウェア属性を改善することは、プロアクティブな改変である。

5

問3 保守性は、修正に対してソフトウェアプロダクトが示す資質である。次の保守性に関する説明のうち、誤っているものはどれか。

- ア 解析性、変更性、安定性及び試験性は、保守性の副特性である。
- イ 修正には、ソフトウェアの訂正だけでなく、環境、要求および機能仕様の変化に対する改良を含める。
- ウ 保守者は、保守性を仕様化し、レビューし、実装する。
- エ 保守者が、保守性を維持・向上するためには、体系的および成熟したプロセス、手法およびツールを用いることが、効果的である。

6

問4 ISO/IEC 12207:2008 (JIS X 0160:2012) で示されるソフトウェアライフサイクルプロセスにおいて定義されていないプロセスはどれか。

- ア ソフトウェア実装プロセス
- イ ソフトウェア保守プロセス
- ウ ソフトウェア運用プロセス
- エ ソフトウェア購買プロセス

7

問5 ひとつのソフトウェア保守案件の適正な工数見積もりを行うとき、注意すべき事からに関する次の記述のなかで、誤っているものはどれか。

- ア 小修正で済むと予測される案件でもプログラム修正工数が大部分を占める工数見積もりを行うことは、過少見積になる危険性が高い。
- イ 担当者の保守経験の有無による実工数が大きく異なるため、見積基準を作成する場合は保守経験やスキルに関する前提条件が必要である。
- ウ 緊急対応を要する場合の見積りは、プログラム修正工数をまず算出し、一定割合の修正設計やテストの工数を付加する簡易見積りが有効である。
- エ テスト工数見積りは、修正の影響分析により大きく異なるため、影響分析工程と修正設計工程以降は2段階に分割して見積もることを提案すべきである。

8

問6 ソフトウェア保守に特有のアクティビティおよびタスクに関する次の文章を読んで、空欄に当てはまる適切な組み合わせを選べ。

修正依頼・問題報告の分析アクティビティでは、問題報告および修正依頼の分析、解決策の勧告と文書化および承認の取得を行う。問題報告の場合には を行うこと、修正依頼 にはそのを行うことが、各々における特徴的なタスクである。

	a	b
ア	優先権の付与	受け入れ／拒絶
イ	優先権の付与	コスト算定
ウ	再現および検証	受け入れ／拒絶
エ	再現および検証	コスト算定

9

問7 次の移行（Migration）に関するアクティビティの記述で誤っているのはどれか。

- ア 目的の通告：旧環境の支援を停止する理由、新しい環境の説明及び利用開始日等を通知する。
- イ 並行運用：ユーザが新しい環境への滑らかに移行できるように、新旧環境を提供する。
- ウ 運用後のレビュー：並行運用と新しい環境に変わる影響をレビューする。
- エ データの保管：古いデータのみを保管し、ソフトウェアは保管対象にしない。

10

ご清聴ありがとうございました。

アンケートに、ご協力をお願いします。