

ソフトウェア品質知識体系ガイド SQuBOK® 第2版 改訂内容のご紹介

2015年 2月 9日

SQuBOK®策定部会(第2版) Area Editorチーム
リーダー (日本電気(株)) 菅田 直美

目次

- 第1版～2版 世の中の動き
- 第2版 改訂内容
- 第2版の解説内容紹介
- SQuBOK®ガイドの使い方
- まとめ

第1版～2版(7年間) 世の中の動き

<新製品>

<SNS関連>

<セキュリティ>

<その他>

2007

iPhone発売

YouTube、ニコニコ動画人気
学校裏サイトが問題に

ファイル交換SW経由の情報漏えい

2008

2009

Twitterブーム

グリーンICT
クラウドコンピューティング台頭

2010

iPad発売

Facebookがアクセス数で
グーグルを超える

尖閣諸島事故がYouTube流出
コンシューマライゼイション化

2011

中東紛争でSNSを利用
DeNAが横浜ベイスターズ買収

スマホの位置情報履歴保有が問題に
(東日本大震災)

2012

Lineユーザ急増

ビッグデータへの期待

2013

3Dプリンタ
グーグルグラス
自動走行車

SNSのワルノリ投稿で逮捕者
リベンジポルノ

NSAの盗聴(スノードン氏)

2014

ゲーム実況(YouTuber)が台頭

ダークウェブ
IOT社会のセキュリティ問題
(最もハッキングされやすい車)

第2版 主な改訂内容

1. **開発技術の追加**
2. **安心、安全、快適な社会への対応**
3. **国際規格の改訂への対応**

※ほぼ全体を最新情報へ更新

※用語の統一、文献一覧を巻末に整理 など

開発技術の追加

- 第1版：開発技術は樹形図に示すのみ



- 第2版：開発技術を品質の視点から解説
 - **要求分析の技法**：要求抽出、要求分析(非機能要求分析…)、要求仕様化、要求の妥当性確認と評価
 - **設計の技法**：方式設計の技法(品質に基づくアーキテクチャ設計論…)、詳細設計の技法(テスト駆動開発…)
 - **実装の技法**：リファクタリング、IDE、CI…
 - **モデル化の技法**：離散系のモデル化技法、連続系のモデル化技法、ドメイン特化言語
 - **形式手法**：形式仕様記述の技法、形式検証の技法

★SQuBOKの全貌を提示

安心、安全、快適な社会への対応

- 「**専門的品質特性のソフトウェア品質技術**」の追加
 - 使用性
 - ユーザビリティテスト、インタラクティブシステムの人間中心設計プロセス
 - セーフティ
 - セーフティ実現のためのリスク低減技法、セーフティ・クリティカルシステムのテスト
 - セキュリティ
 - セキュリティ要求分析、セキュリティ設計、セキュリティパターン、セキュアコーディング、セキュリティテスト

★安心、安全、快適な社会を求める世の中の動きを反映

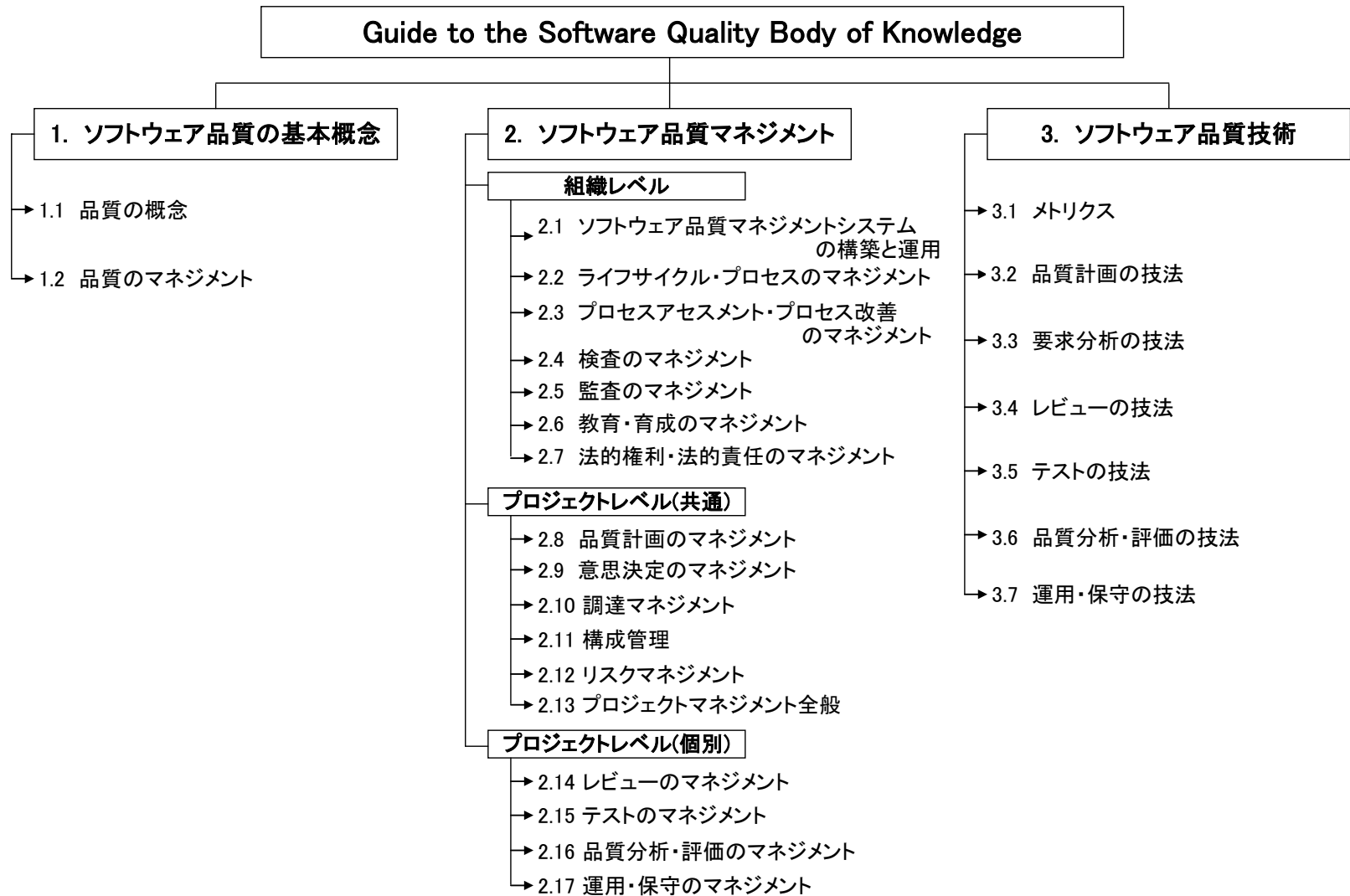
国際規格の改訂への対応

- 185件の国際規格を参照
 - 第1版は125件

特に大きな変化

- **ISO/IEC25000シリーズ(SQuaRE)の本格的な登場**
 - 製品品質モデルの6つの品質特性⇒8つの品質特性へ(機能適合性, 性能効率性, 互換性, 使用性, 信頼性, セキュリティ, 保守性, 移植性)
 - ソフトウェアからシステムへの視点の変化
(SW単品⇒システムの中核をなすコンポーネントとしてのSWへ)
- **セーフティ関連規格の充実**
 - 機能安全(IEC61508)の改訂、ISO/IEC Guide51
 - 個々の産業分野向けグループ安全規格(ISO 26262(2011年)など)

SQuBOK® V1の構造(樹形図)



SQuBOK® V2の構造(樹形図)

Guide to the Software Quality Body of Knowledge (V2)

1. ソフトウェア品質の基本概念

- 1.1 品質の概念
- 1.2 品質のマネジメントの概念
- 1.3 ソフトウェアの品質マネジメントの特徴

2. ソフトウェア品質マネジメント

組織レベル

- 2.1 ソフトウェア品質マネジメントシステムの構築と運用
- 2.2 ライフサイクル・プロセスのマネジメント
- 2.3 ソフトウェアプロセス改善のマネジメント
- 2.4 検査のマネジメント
- 2.5 監査のマネジメント
- 2.6 教育・育成のマネジメント
- 2.7 法的権利・法的責任のマネジメント

プロジェクトレベル(共通)

- 2.8 意思決定のマネジメント
- 2.9 調達のマネジメント
- 2.10 リスクマネジメント
- 2.11 構成管理
- 2.12 プロジェクトマネジメント

プロジェクトレベル(個別)

- 2.13 品質計画のマネジメント
- 2.14 要求分析のマネジメント **NEW!**
- 2.15 設計のマネジメント **NEW!**
- 2.16 実装のマネジメント **NEW!**
- 2.17 レビューのマネジメント
- 2.18 テストのマネジメント
- 2.19 品質分析・評価のマネジメント
- 2.20 リリース可否判定 **NEW!**
- 2.21 運用のマネジメント
- 2.22 保守のマネジメント

3. ソフトウェア品質技術

工程に共通な技術

- 3.1 メトリクス
- 3.2 モデル化の技法 **NEW!**
- 3.3 形式手法 **NEW!**

工程に個別な技術

- 3.4 品質計画の技法
- 3.5 要求分析の技法 **NEW!**
- 3.6 設計の技法 **NEW!**
- 3.7 実装の技法 **NEW!**
- 3.8 レビューの技法
- 3.9 テストの技法
- 3.10 品質分析・評価の技法
- 3.11 運用の技法
- 3.12 保守の技法

専門的品質特性の技術

- 3.13 使用性の技法 **NEW!**
- 3.14 セーフティの技法 **NEW!**
- 3.15 セキュリティの技法 **NEW!**

SQuBOK® V2の構造(樹形図)

Guide to the Software Quality Body of Knowledge (V2)

開発技術の追加

1. ソフトウェア品質の基本概念

- 1.1 品質の概念
- 1.2 品質のマネジメントの概念
- 1.3 ソフトウェアの品質マネジメントの特徴

「品質のマネジメント」を2つに分割

2. ソフトウェア品質マネジメント

- 組織レベル
- 2.1 ソフトウェア品質マネジメントシステムの構築と運用
 - 2.2 ライフサイクル・プロセスのマネジメント
 - 2.3 ソフトウェアプロセス改善のマネジメント
 - 2.4 検査のマネジメント
 - 2.5 監査のマネジメント
 - 2.6 教育・育成のマネジメント
 - 2.7 法的権利・法的責任のマネジメント
- プロジェクトレベル(共通)
- 2.8 意思決定のマネジメント
 - 2.9 調達のマネジメント
 - 2.10 リスクマネジメント
 - 2.11 構成管理
 - 2.12 プロジェクトマネジメント
- プロジェクトレベル(個別)
- 2.13 品質計画のマネジメント
 - 2.14 要求分析のマネジメント **NEW!**
 - 2.15 設計のマネジメント **NEW!**
 - 2.16 実装のマネジメント **NEW!**
 - 2.17 レビューのマネジメント
 - 2.18 テストのマネジメント
 - 2.19 品質分析・評価のマネジメント
 - 2.20 リリース可否判定 **NEW!**
 - 2.21 運用のマネジメント
 - 2.22 保守のマネジメント

開発技術の追加

3. ソフトウェア品質技術

- 工程に共通な技術
- 3.1 メトリクス
 - 3.2 モデル化の技法 **NEW!**
 - 3.3 形式手法 **NEW!**
- 工程に個別な技術
- 3.4 品質計画の技法
 - 3.5 要求分析の技法 **NEW!**
 - 3.6 設計の技法 **NEW!**
 - 3.7 実装の技法 **NEW!**
 - 3.8 レビューの技法
 - 3.9 テストの技法
 - 3.10 品質分析・評価の技法
 - 3.11 運用の技法
 - 3.12 保守の技法
- 専門的品質特性の技術
- 3.13 使用性の技法 **NEW!**
 - 3.14 セーフティの技法 **NEW!**
 - 3.15 セキュリティの技法 **NEW!**

安心、安全、快適な社会への対応

SQuBOK® 樹形図 ~5層構造~

追加

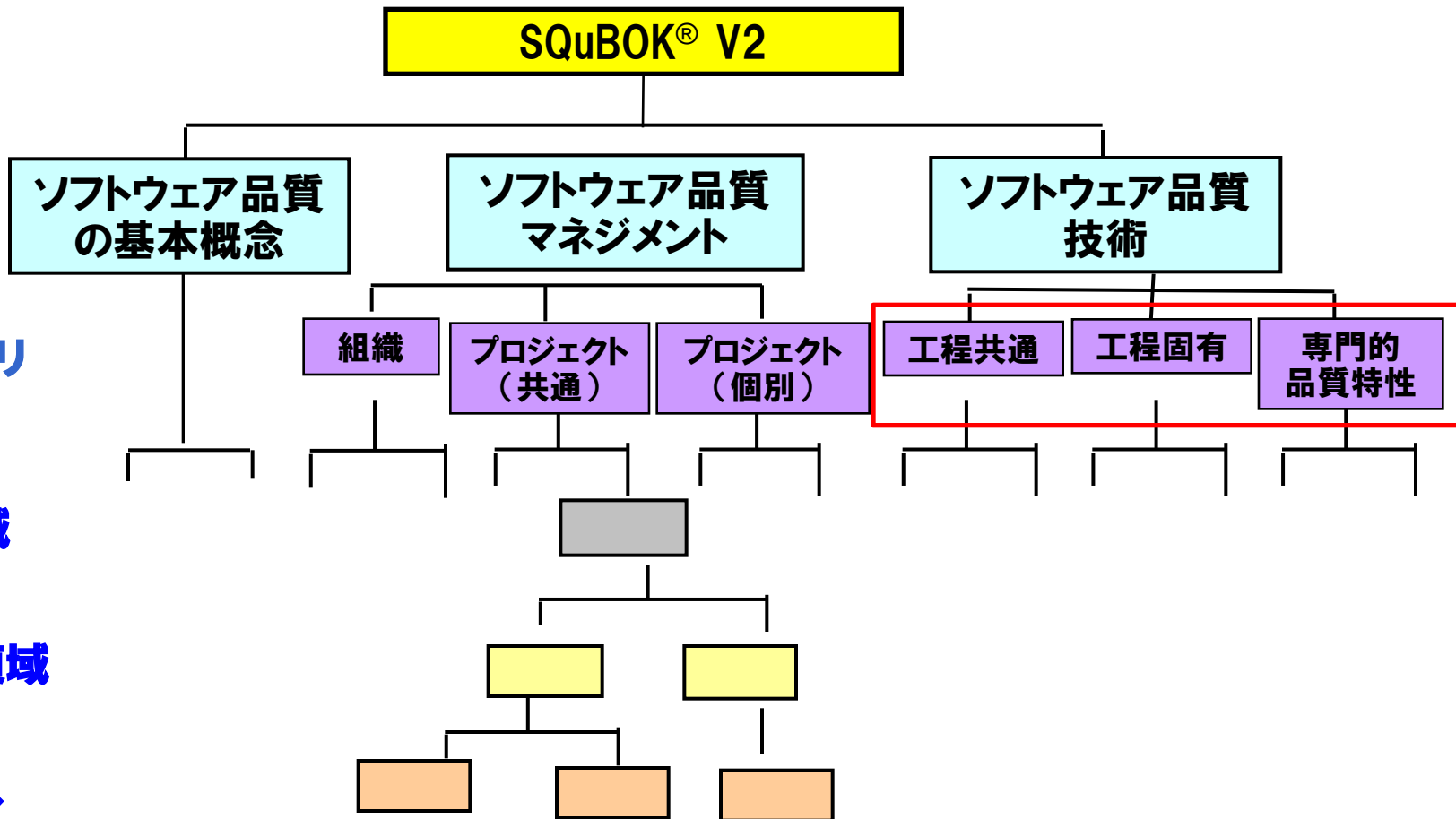
1. カテゴリ

2. 副カテゴリ

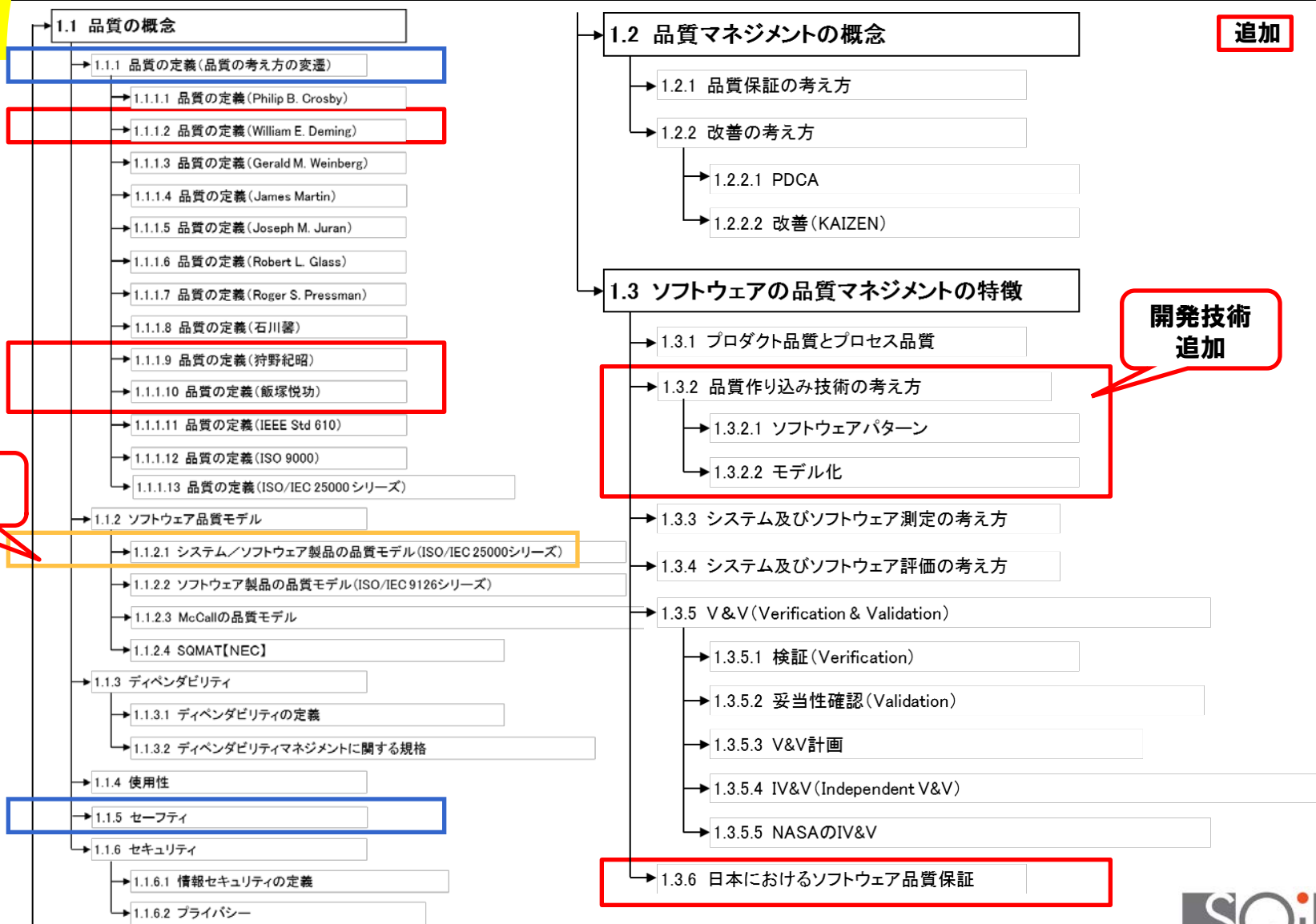
3. 知識領域

4. 副知識領域

5. トピックス



1. ソフトウェア品質の基本概念

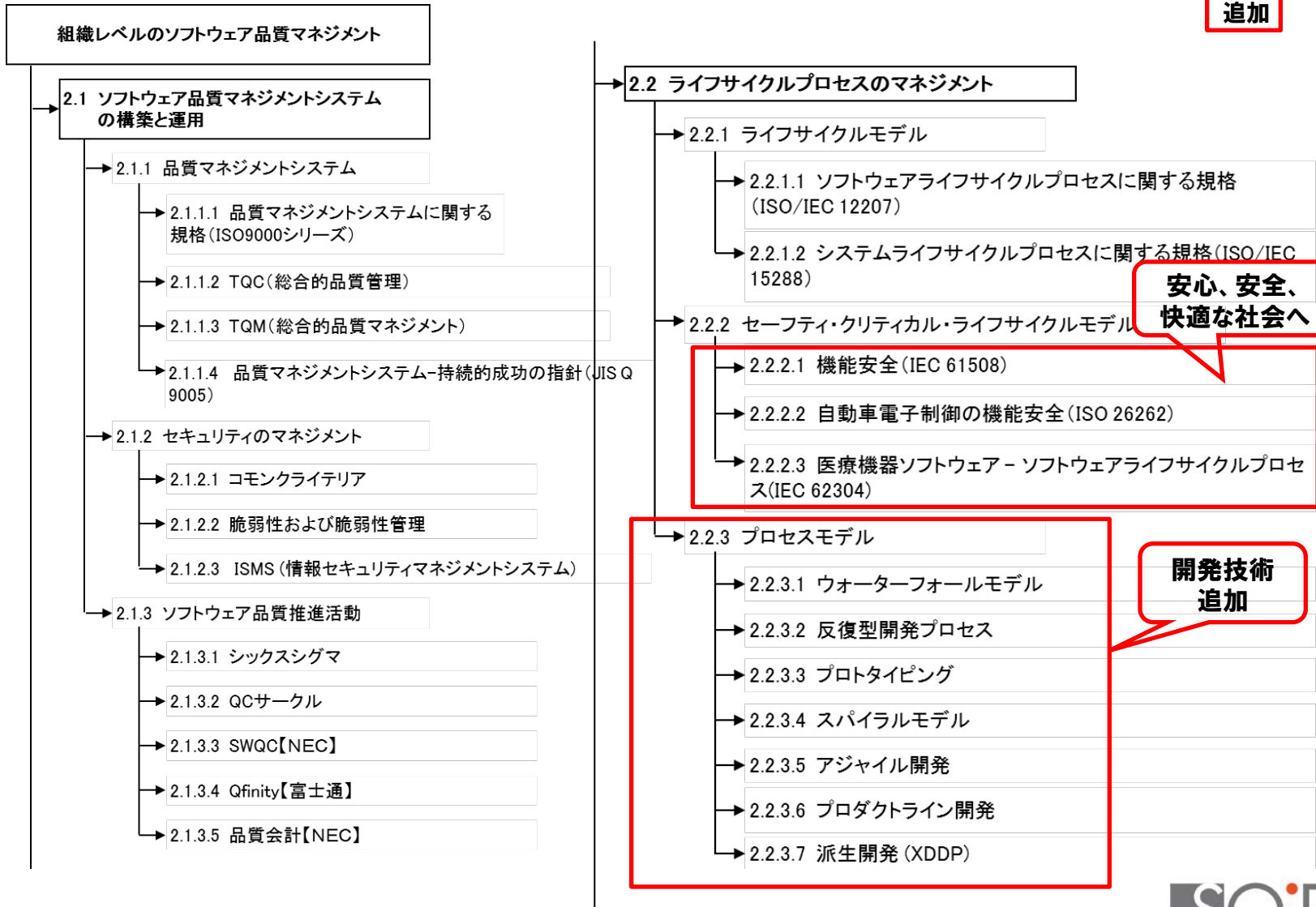


国際規格の
大幅改訂

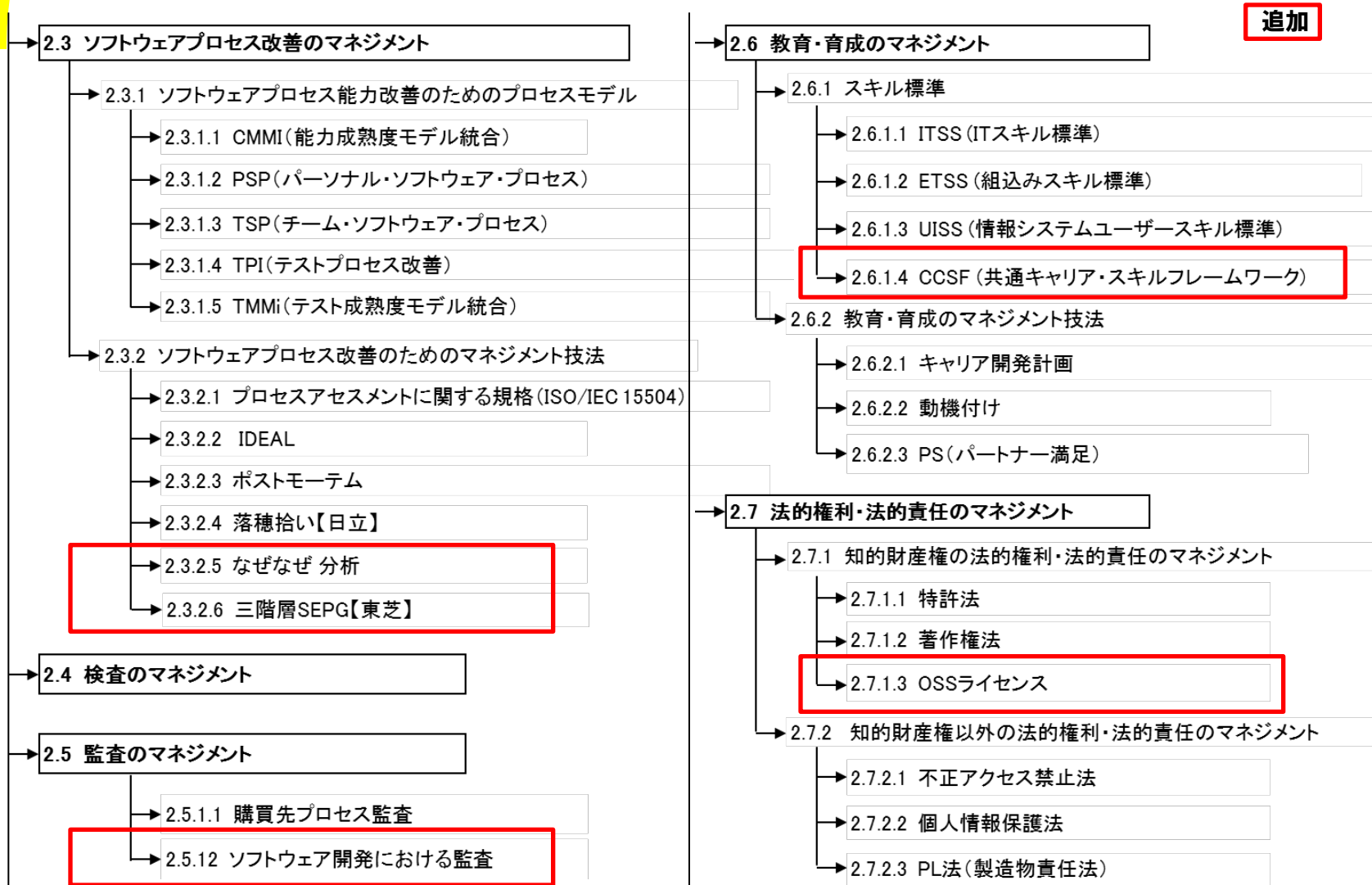
追加

開発技術
追加

2.ソフトウェア品質マネジメント①

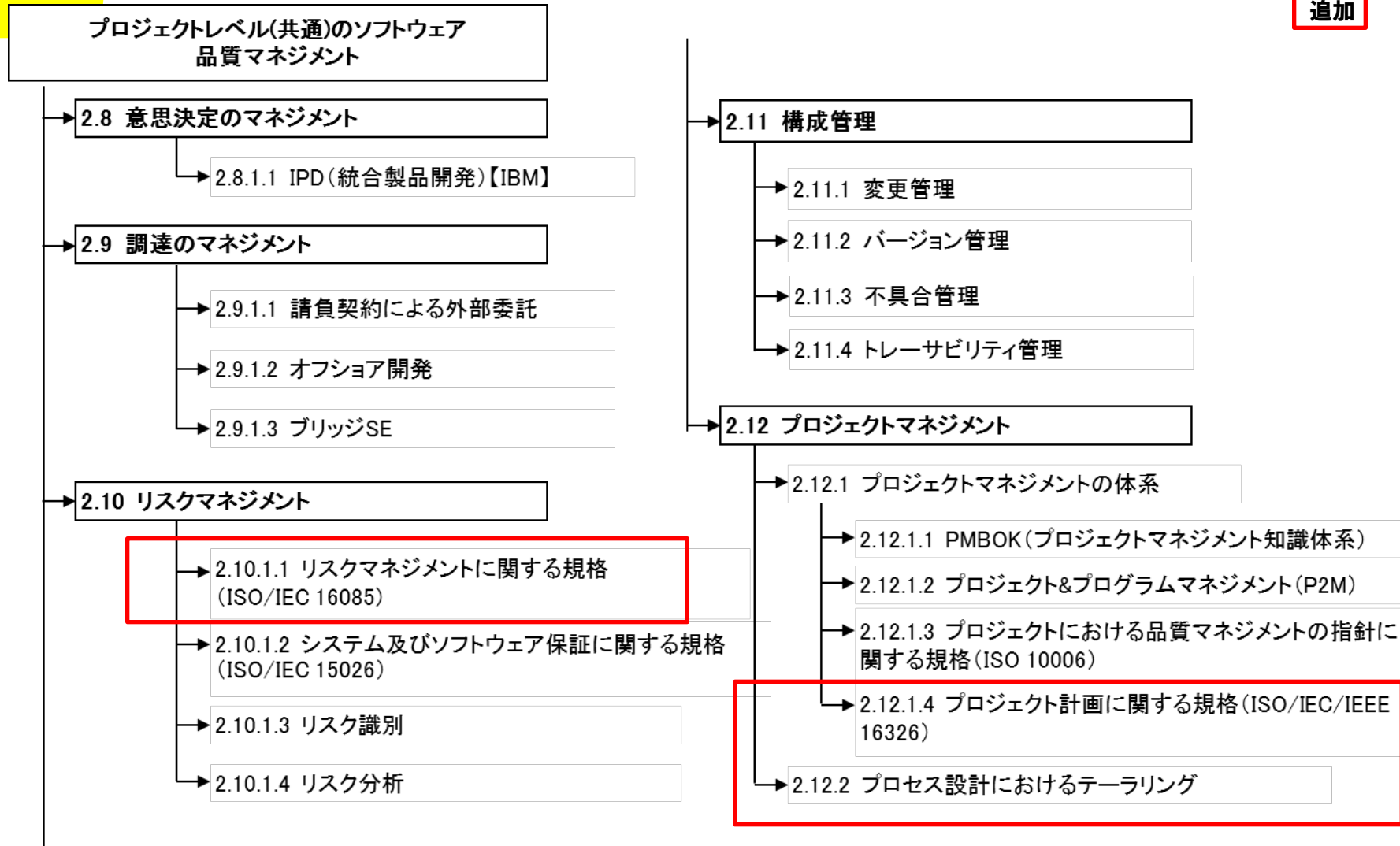


2.ソフトウェア品質マネジメント②

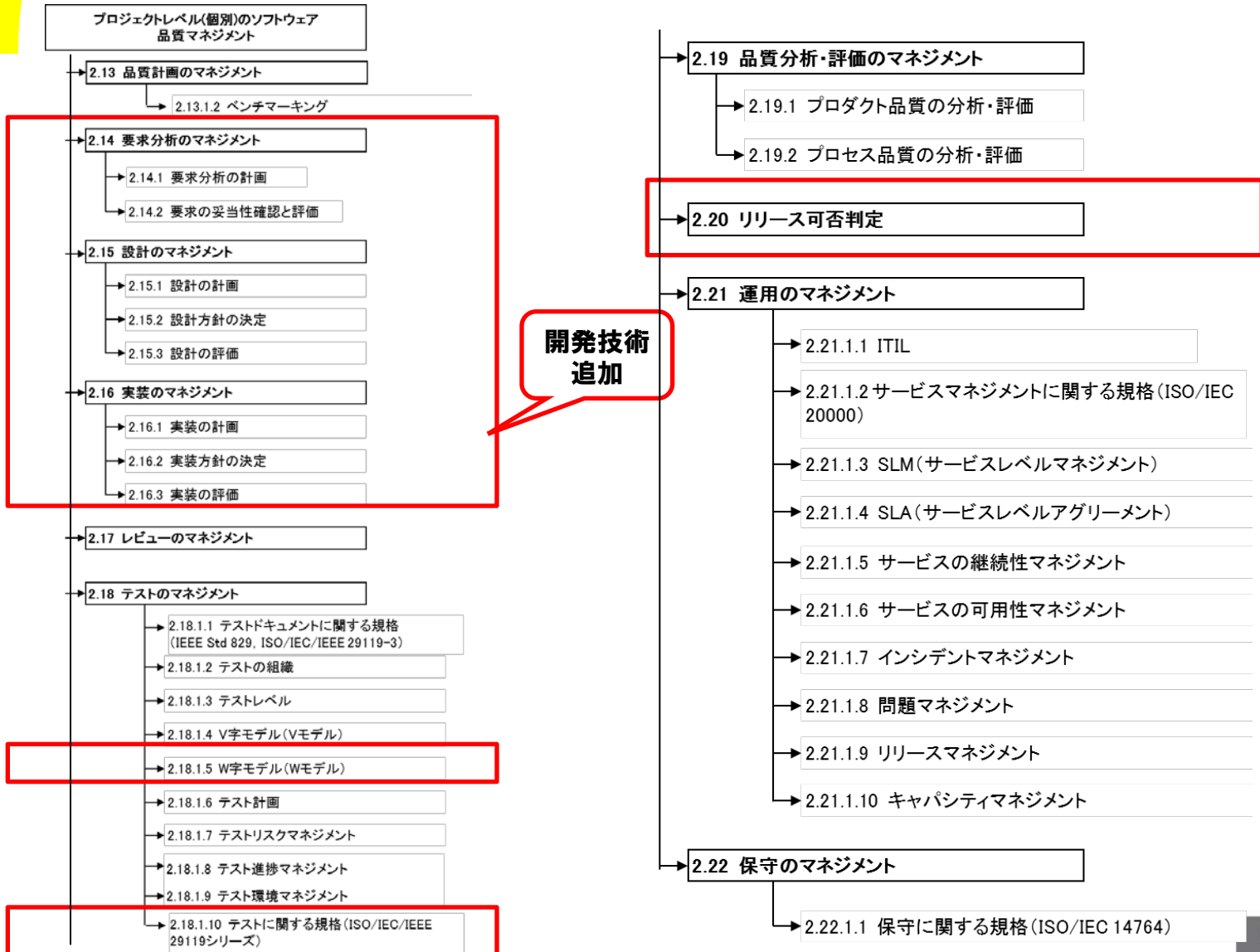


2.ソフトウェア品質マネジメント③

追加



2.ソフトウェア品質マネジメント④



追加

開発技術追加

3.ソフトウェア品質技術①

工程に共通なソフトウェア品質技術

3.1 メトリクス

3.1.1 測定理論

3.1.1.1 測定量

3.1.1.2 指標

3.1.1.3 尺度

3.1.1.4 評定水準

3.1.1.5 測定プロセス

3.1.1.6 GQM手法

3.1.2 プロダクトメトリクス

3.1.2.1 内部メトリクス

3.1.2.2 外部メトリクス

3.1.2.3 利用時の品質メトリクス

3.1.2.4 複雑度のメトリクス

3.1.2.5 LOC(ソースコード行数)

3.1.2.6 ファンクションポイント

3.1.3 プロセスメトリクス

3.2 モデル化の技法

3.2.1 離散系のモデル化技法

3.2.1.1 UML

3.2.1.2 SysML

3.2.1.3 構造化チャート(PAD)

3.2.2 連続系のモデル化技法

3.2.3 ドメイン特化言語

3.3 形式手法

3.3.1 形式仕様記述の技法

3.3.2 形式検証の技法

追加

開発技術
追加

3.ソフトウェア品質技術②

追加

工程に個別なソフトウェア品質技術

→ 3.4 品質計画の技法

- 3.4.1.1 費用便益分析
- 3.4.1.2 品質計画書

→ 3.5 要求分析の技法

- 3.5.1 要求抽出
 - 3.5.1.1 ステークホルダー識別
 - 3.5.1.2 要求開発 (Openthology)
- 3.5.2 要求分析
 - 3.5.2.1 機能要求分析
 - 3.5.2.2 非機能要求分析
 - 3.5.2.3 品質機能展開
 - 3.5.2.4 要求可変性分析
- 3.5.3 要求仕様化
 - 3.5.3.1 ソフトウェア要求仕様
 - 3.5.3.2 USDM (要求仕様記述法)
- 3.5.4 要求の妥当性確認

3.6 設計の技法

- 3.6.1 方式設計の技法
 - 3.6.1.1 部品化の技法
 - 3.6.1.2 アーキテクチャパターン
 - 3.6.1.3 品質に基づくアーキテクチャ設計・評価技法
 - 3.6.1.4 DSM (依存関係マトリクス)
 - 3.6.1.5 フレームワーク

→ 3.6.2 詳細設計の技法

- 3.6.2.1 テスト駆動開発
- 3.6.2.2 デザインパターン
- 3.6.2.3 設計原則

3.7 実装の技法

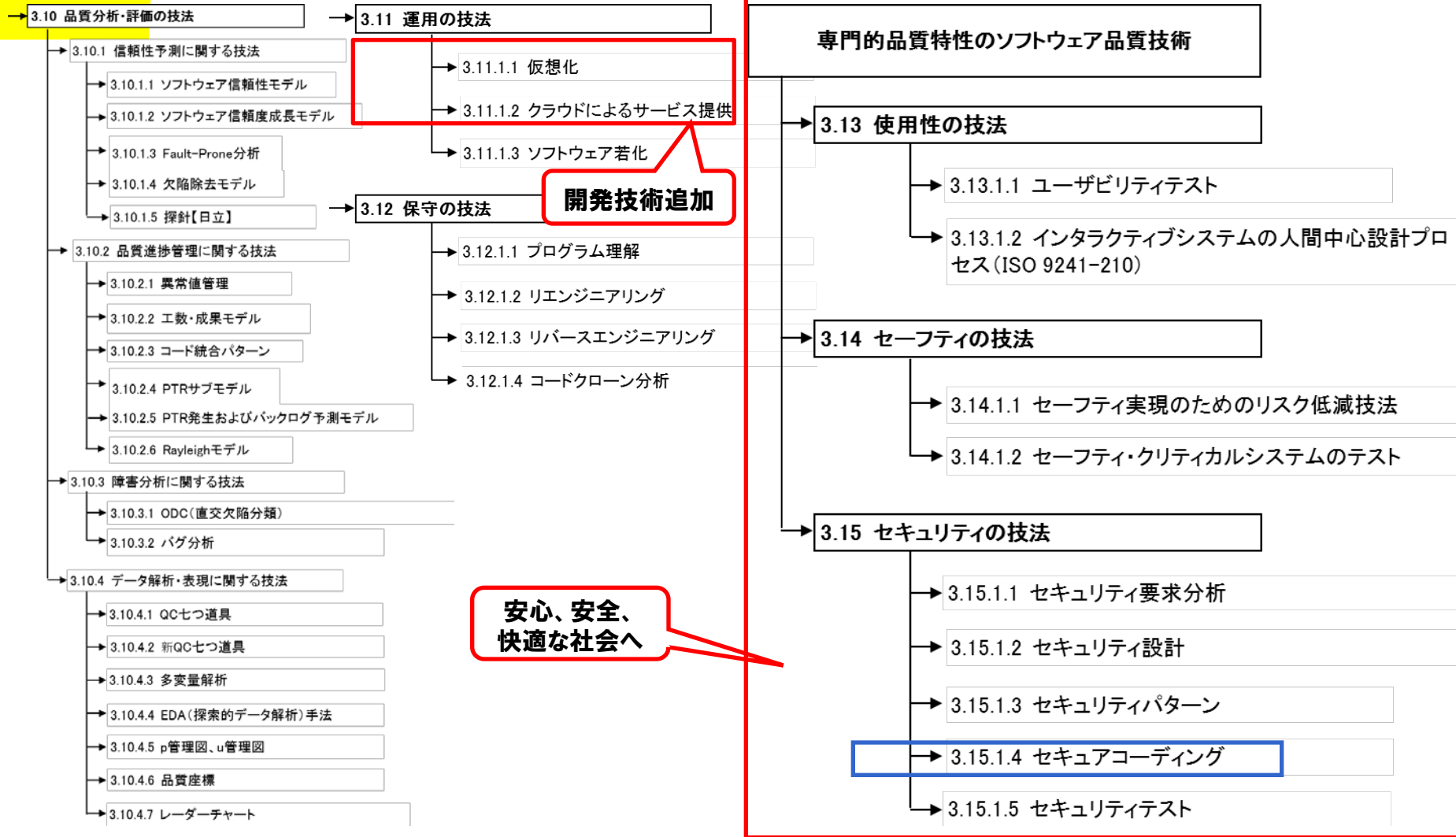
- 3.7.1.1 コーディング規約
- 3.7.1.2 リファクタリング
- 3.7.1.3 契約による設計
- 3.7.1.4 IDE (統合開発環境)
- 3.7.1.5 CI (継続的統合)

開発技術追加

3.ソフトウェア品質技術③



3.ソフトウェア品質技術④



開発技術追加

安心、安全、
快適な社会へ

追加

主な改訂内容のご紹介①

1. ソフトウェア 品質の基本概念

1.1.1 品質の定義

品質とは

- 「品質」という用語自体が本質的に深い意味を持つ
- 時代の変遷、ニーズの高度化・成熟化により多面的な意味を保有
⇒ 様々な研究者が様々な場面で品質を定義（統一見解はない）

★ 品質の理解のためには、以下の3つの点から考えるとよい

■ 顧客の要求把握

- 明示的要求だけでなく、潜在的なニーズや期待。時間の要素。
- 直接の顧客以外の要求（社会、供給者等）。要求のバランス。

■ 要求の実現

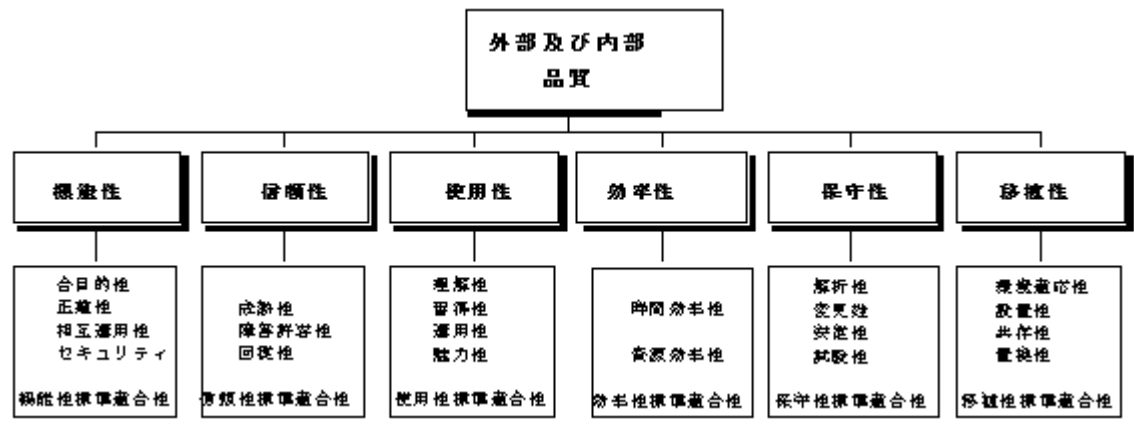
- 要求をSWの要求に置き換えて実現。ソフトウェアエンジニアリング

■ 結果として得られる顧客満足

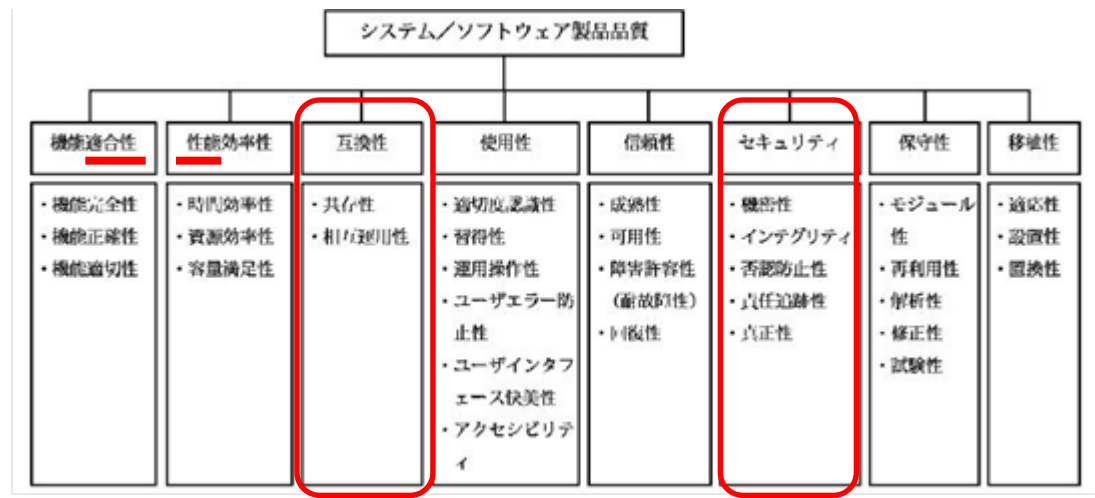
- 要求の実現は当たり前。顧客の予想を超えた価値の提供が重要
 - コトづくり（ものの提供によりコトを作り出す、行動様式の変化をもたらす⇒ものづくりに参加する人たちに夢や将来像を明示）

製品品質モデル

ISO/IEC9126



ISO/IEC25000 (SQuaRE)



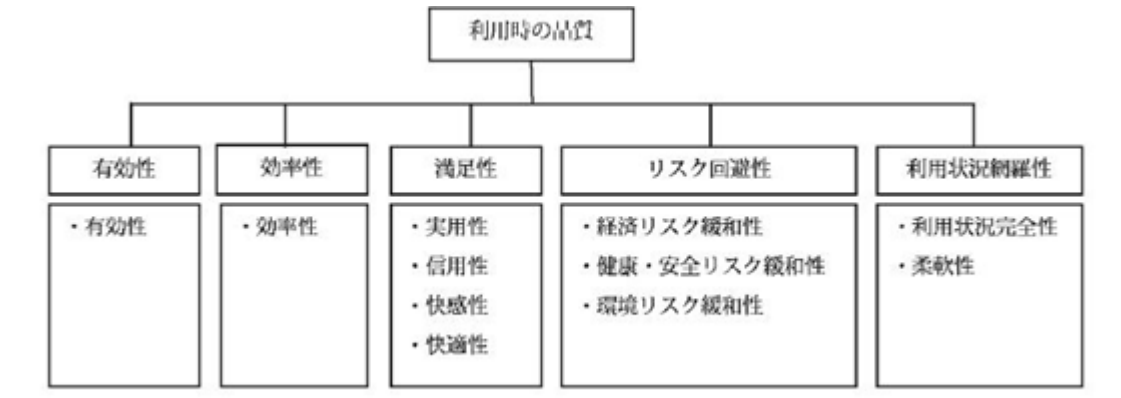
利用時の品質モデル

ISO/IEC9126



※副特性は未定義

ISO/IEC25000
(SQuaRE)



主な改訂内容のご紹介②

1.ソフトウェア品質の基本概念

1.1.5 セーフティ

セーフティとレジリエンス(Resilience)

■ セーフティ

- 『受容できないリスクがないこと』[ISO/IEC Guide 51:1999]

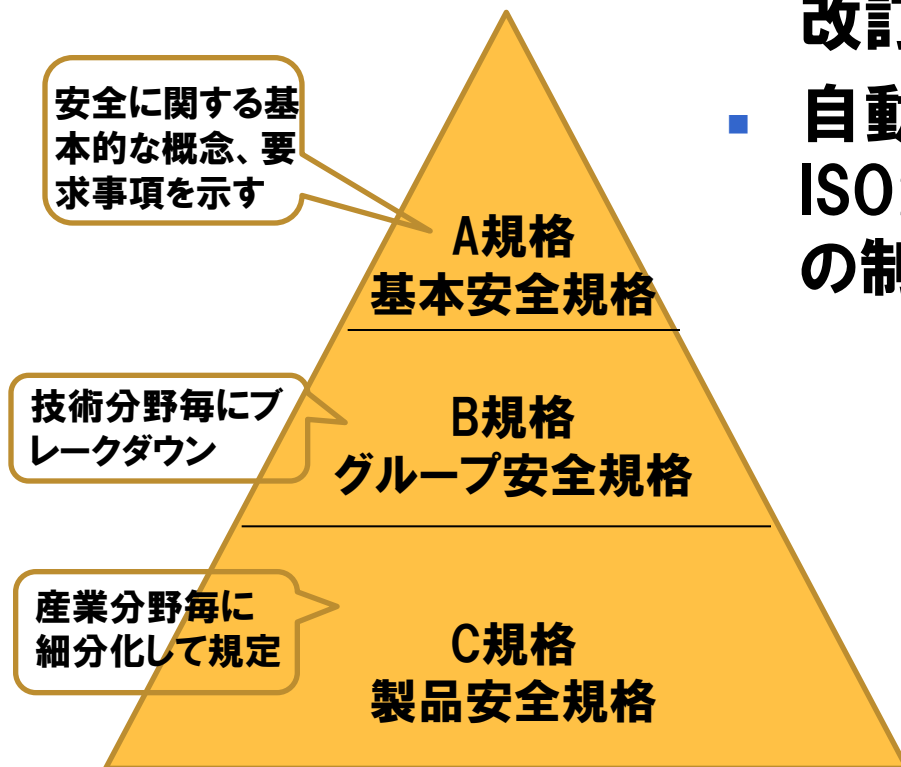
■ レジリエンス(Resilience)

- 『安全は変化する条件下で成功する能力である』[Hollnagel 2014]
- 『システムが想定された条件や想定外の条件の下に要求された動作を継続できるために、自分自身の機能を、条件変化や外乱の発生前、発生中、あるいは発生後において調整できる本質的な能力のこと』[Hollnagel 2014]
- 東日本大震災を受けて提案され、広まりつつある
 - 復元力、回復力、弾力などと訳される。心理学では「折れない心」「強くてしなやかな」という意味合いで使用される。
 - (1)事象に対処する能力, (2)進展しつつある事象を監視する能力, (3)未来の脅威と好機を予見する能力, (4)過去の失敗・成功双方から学習する能力, という4つの能力を重視

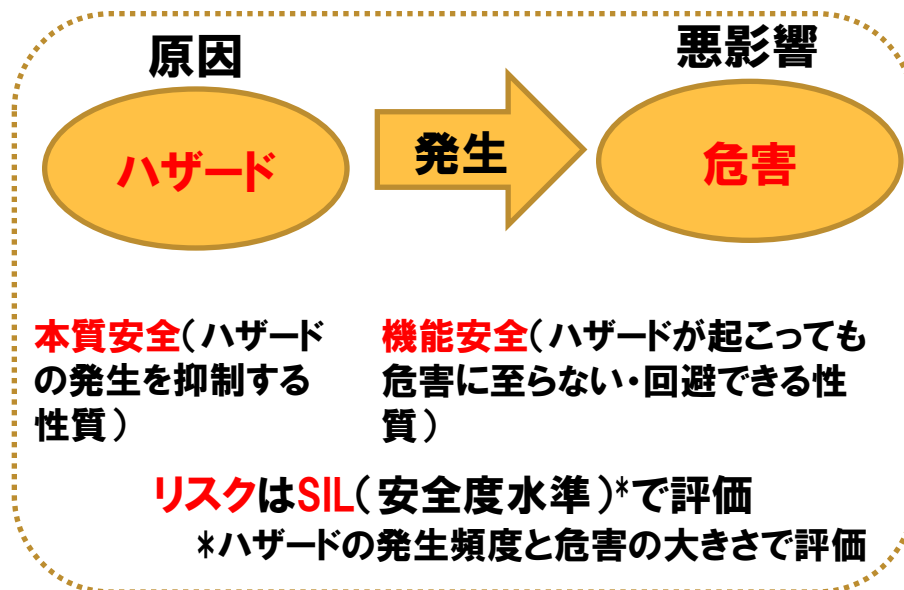
※セーフティを考えるときはレジリエンスも考慮

セーフティ関連規格

- セーフティ関連規格は、近年、頻繁に改訂されている
- 自動車向け機能安全規格(ISO26262)など、産業分野毎の規格の制定が進む



安全に関する国際規格の階層構造
(ISO/IEC Guide51より)



セーフティの考え方

主な改訂内容のご紹介③

1.ソフトウェア 品質の基本概念

1.3.6 日本における SW品質保証

日本のSW品質保証の特徴

①レビュー重視

- よりよい設計、よりよいソースコードを目指して
- レビューが技術者育成の役割も

②障害分析に基づく改善

- 障害分析⇒プロセス改善の流れ

③独立した品質保証部門の存在

- 開発部門とは独立した品質保証部門
- 開発途中の品質監査、出荷製品の評価や出荷判定、出荷後の管理などを担当

主な改訂内容のご紹介④

2.ソフトウェア 品質マネジメント

2.7.1.3 OSSライ センス

OSS

- ソースコードが入手でき、ソースコードの改変と、手を加えたソースコードの再頒布が認められているソフトウェア

OSSライセンス

- OSSを利用する際の条件を、そのOSS著作者が定めたもの
- OSSは著作権法で保護される知的財産

OSS利用時の注意点

- 特に再頒布の利用時の、OSSライセンス遵守が重要
- 遵守しなければ、著作権侵害→訴訟のリスクを負う
- 複数のOSS利用によるライセンス条件が両立しないケースなどがありうる(ソース開示できないOSSとソース開示要求しているOSSの組み合わせなど)
- 事前のOSSライセンス調査が必須

SQuBOK®ガイドの使い方

- **経営層**
 - SW品質の基本的な概念の理解、品質問題発生時の適用技術の妥当性判断
- **開発リーダーやその管理者**
 - SW品質問題にぶつかったときの解決書
- **開発者**
 - 自分の開発したSWの良し悪しの見極め方法、良いSWを開発するための技術の理解
- **品質保証に携わる技術者**
 - 全体を読破し全貌を理解、問題発生時や組織的改善での利用
- **ソフトウェアの営業に関わる方々**
 - SW品質技術の常識、自社の強み・弱みの把握とアピール
- **学生**
 - 知識として理解し、社会へ出たとき先輩に一步先んじる

まとめ

- **ソフトウェア品質技術の最新技術動向をコンパクトに整理**
 - V1からほぼ全面改訂に近い
- **知識習得や業務へ利用してください**

SQuBOK®

Guide to the
Software **Q**uality **B**ody **o**f **K**nowledge