第３分科会 PS グループ

ソフトウェア開発の品質保証活動のポイント

Point of quality assurance of software development

主査　　：　飯泉　　紀子　（(株）日立ハイテクノロジーズ）
副主査　：　田所　　孝文　（(株）山武）
副主査　：　吉澤　　智美　（NEC エレクトロニクス（株））
リーダー：　新内　　貴弘　（オムロンソフトウェア（株））
　　長谷部　幹夫　（(株）東芝　デジタルメディアネットワーク社）
　　金子　　和成　（アンリツエンジニアリング（株））
　　豊田　　拓也　（(株）アドバンテスト）

１．研究概要

現在、多くの企業あるいは組織で CMMI®（能力成熟度モデル統合版、Capability Maturity Model® Integration）〔注１〕をモデルとしたプロセス改善が行われており、そのプロセスの元、ソフトウェア開発が行われている。また、CMMI®の導入により、独立したグループによる客観的な品質保証活動の必要性が認知されてきている。

しかし、実際の品質保証活動がモデルベースの活動から脱却できていなことへの不満や、品質保証活動の効果をうまく表現できないあるいは実感されていないという理由により、開発部門との間で意識のずれが発生している。

私たちのグループでは、複数の組織形態で利用可能な“開発者と SQA 担当者共に有意義と感じられる SQA 活動”の研究を行った。

Abstract

Recently many enterprise or organization have achieved process improvement activities based on CMMI(Capability Maturity Model Integration), and software have been developed on the process. Moreover, the necessity of an objective quality assurance by an independent group has been recognized because of Introducing CMMI.

However, the gap of consideration is caused in the quality assurance that cannot get rid of the activity of the model base or cannot specify the result or cannot actually feel the effectiveness between the development section and the quality assurance unit.

We researched the quality assurance which can solve the above-mentioned problem and can share the meaning between the development member and the person in charge of the quality assurance.

２．研究の背景と経過

本グループのメンバは、製品開発の組織形態や品質保証活動（SQA：Software Quality Assurance)の形態に違いはあるものの、組込みソフトウェア開発を対象としているという共通点があった。

このメンバで SQA 活動の現状について議論を行ったところ､「SQA 活動は実施されてはいるものの SQA 実施者と開発者との認識に乖離があり、その効果が得られているとはいいがたい状況にある」との結論に行き着いた。

このような状況を改善するため、次の手順により進めることにした。

①．複数の組込みソフトウェア開発プロセスと SQA 形態（権限と責任)の整理

②．[複数の形態で利用可能でかつ]開発者と SQA 担当者共に有意義と感じられる SQA 活動の考察

※SQA (Software Quality Assurance)

要員および管理者に対して、ソフトウェアプロセスおよび関連する作業成果物の客観的見通しを提供する活動である。

３．活動の目標

本グループでは、

「SQA 活動が開発者と SQA 担当者双方にとって意義が感じられ、プロセス改善活動においても効果ある活動となる。」

ことを目標とした。

そのために、

①．開発者と SQA 担当者共に納得できる SQA チェックリストの作成観点

②．SQA チェックリストの効果的な運用方法

を提案する。

４．活動内容

まず、複数の組込みソフトウェア開発プロセスと SQA 形態（権限と責任)を整理し、次に SQA 活動のあり方を検討した。

４．１　組込みソフトウェア開発プロセスの実態

組込みソフト開発プロセスの概要の例を表１に挙げる、それぞれの例のプロセスの概要図は、付録１を参照のこと。

表 1．組込みソフト開発プロセスの例

例 （タイプ）	分野	プロセスの特徴
例１ （Ⅰ）	ＦＡ向け制御機器用組み込みソフト	スケジュールのベースがハードのスケジュールとなっており、ハード要因の影響を大きく受けることが多い。また、予算が決まってから、実現する要件を取捨選択することが多いので要件の確定に時間を要することがおおく、プロジェクト開始後に要件が確定することが多い。
例２ （Ⅱ）	移動体通信用計測器の組込みソフト開発	製品仕様が、発展し続ける規格と多様な顧客要求から形成されているため、その要件管理プロセスは継続したものとなる。また複数要件の並行開発が行われ、日程および体制の変更をタイムリに行う必要がある。
例３ （Ⅲ）	コンシューマ向けのＡＶ機器向け組込みソフト開発	コンシューマ向けのため、顧客要件は多方面から複合的に形成され、自らも要件を生み出す必要がある。また要件間に矛盾やギャップなども含まれるので、摺り合わせが必要となる。そのため要求と要件管理のプロセスがプロジェクトの終了間近まで必要となる。

今回はここに挙げた分野を対象にして研究を進める。

表１のタイプについては、後の節で説明する組織の特徴による分類のどれに近い組織かを示したものである。

４．２　SQA 対象プロセスの定義

組込みソフトウェアの開発では、ハードウェアの開発と同時並行してソフトウェアの開発が行われることが多い（図１参照)。

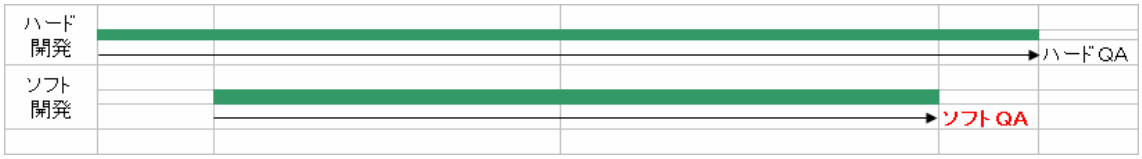


図 1．ハード QA とソフト QA

ハード QA・ソフト QA が一緒になっている場合もあるが、本研究のターゲットはソフト QA（SQA）とする。

SQAの対象プロセスを図2に示す組込みソフトウェア向け開発プロセスのＶ字モデル（下記の図、IPA/SEC による）で SQA の対象プロセスを説明する。

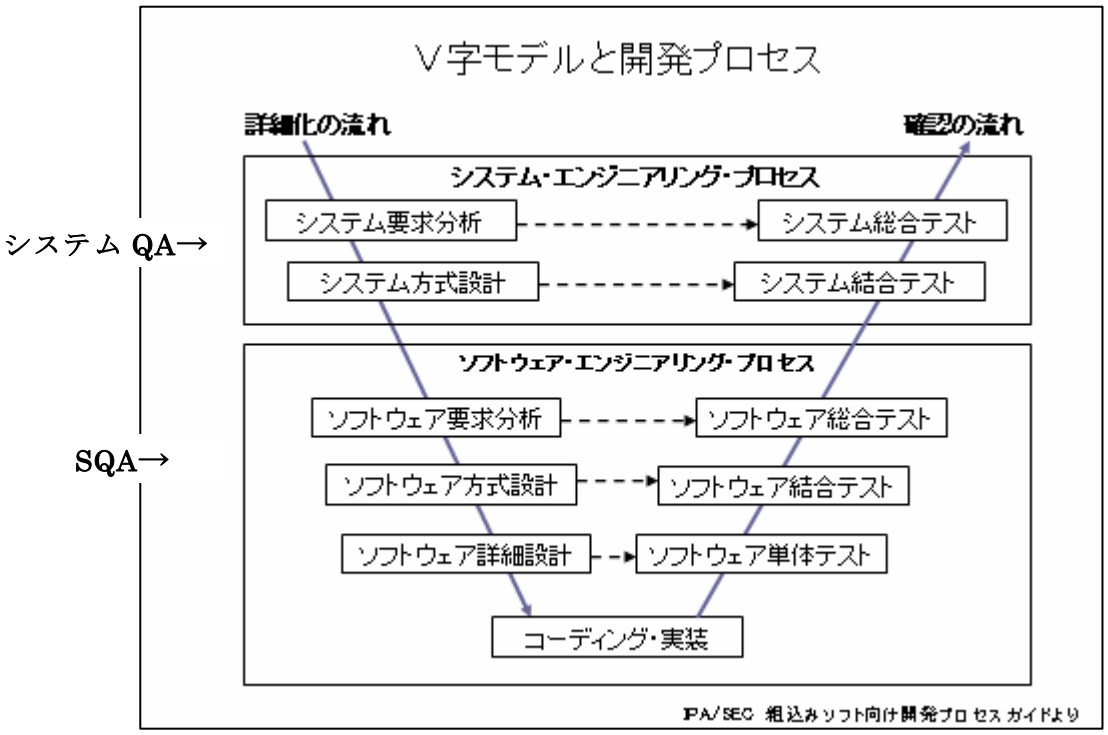


図 2. Ｖ字モデルと開発プロセス

組込みソフトウェアが組み込まれて動作するハードウェア・システム全体を対象とするシステム・エンジニアリング・プロセスに対する QA 活動（システム QA）と区別して、特にソフトウェア・エンジニアリング・プロセスのみを対象とした QA 活動（SQA）を今回の研究の対象とする。

４．３　SQA 形態（権限と責任）の分類

様々な分類観点があり得るが議論を通して、少なくとも SQA の対象となる組織の形態をハードウェア開発組織からの独立性という観点で分類しておく必要があると感じた。そこで、私たちのグループメンバーの経験と知識を元に、表 2 に示すように分類した。

表 2. 組織の特徴による分類

分類	組織の特徴	備考
Ⅰ	ソフトウェア開発組織がハードウェア開発組織から完全に独立していて、SQA を実施するチームに十分な権限と責任が与えられている組織形態。	会社または部門として独立組織となっている。
Ⅱ	ソフトウェア開発組織がある程度独立しているが、SQA を実施するチームに十分な権限と責任が与えられていない組織形態。	会社または部門として分かれてはいるが実質一体運営。
Ⅲ	ソフトウェア開発組織とハードウェア開発組織が一体化していて、システム QA と SQA を明確に区別する事が難しい組織形態。	会社または部門として分かれていない。

この分類は、組織の規模によらないものであり、様々な大きさの組織に適用できる。効果的な SQA を実施するためには、表 2 に示した組織のタイプに応じた SQA が必要である。自部門や他部門のこのような組織タイプを意識することは、SQA のノウハウ等の相互の理解にも有効と考えられる。

４．４　SQA 活動のあり方

SQA 活動を行うにあたり、よく使われるものにチェックリストがある。

「チェックリストの効用は、基本的定石事項によるスクリーニングにあるともいえる。」と言われているように、チェックリストを使用することにより、一定の効果は期待できる。しかし、機械的にチェックだけをしてしまうというデメリットもある。

私たちはチェックリスト作成と運用に工夫をくわえることにより、デメリットを克服し、以下の効果を得ようと考えた。

- チェック項目がもたらす双方の効果や目的が明確になっている
- チェック項目を基に課題事項や問題点が議論できる
- 議論の結果やチェックの結果が現場にフィードバックできる

５．研究の成果および考察

ここでは、開発者とSQA担当者共に納得できるSQAチェックリストの作成観点と、SQAチェックリストの効果的な運用方法について述べる。

５．１　SQAチェックリストの作成観点

SQA活動において、チェックリストは活動のベースとなることが多い。より効果ある活動となるよう、チェックリストの作成は以下の観点に留意して作成する。

・何を、なぜ確認したいのかを明確にする。

・その確認事項は、SQA実施者、開発者双方にとってどのような効果があるのか？

チェックリスト作成方法の一例（今回検討した作成方法）を以下に記述する。

１）CMMI®でモデル化されている確認項目（注意すべき観点）を基に確認したほうがよいと思われる項目を「チェック項目」としてピックアップする。

２）「チェック項目」を確認するための具体的な確認項目を「具体的な確認項目」としてピックアップする

３）「具体的な確認項目」を確認することによりどのような効果・意義があるのかを「確認する意義・効果」として記載する。

４）「確認する意義・効果」を得るための「具体的な確認項目」になっているか検証する。

EX.客先との交渉（要件事項）を確認するには（添付資料2参照）

① 客先との交渉事項を確認するには「要求事項を決めるまでの過程の議事録」を確認すればよい。

② 要求事項が理解できているかどうかを確認するには、実際に要求事項が表現（説明）できるかどうかを確認すればよい（理解できていなければ説明はできない)。

③ なぜ、そんなことをするのか？それは見積もりの精度に影響してくる。要件が曖昧だと見積もりも曖昧になる。逆に要件が明確になっていると見積もり精度も高くなってくる。

④ 見積もり精度を高めるためにはどうすればよいか？要件を明確にすることは見積もり精度を上げるためのひとつの条件である。

上記①～④の検討の結果「チェック項目」「具体的な確認項目」「確認する意義・効果」の項目が決定する。

５．２　SQAチェックリストの運用の工夫

SQA活動は事前に開発者の理解を得ておくことが重要であり、そのためはSQAチェックリストを開発者の教育に利用してもらうことが有効である。そこで開発者にチェック項目の目的や理由を説明し理解してもらい、実際の開発プロセスに活かしてもらうことがポイントである。

また SQA チェックリストを有効なものとして維持するには、ヒヤリングやイン

タビューなどの実際の SQA 活動からチェックリストの不備を見つけて定期的に見直すことや、対象組織の成長に合わせてチェックリストを改訂する等の改善サイクルを回す必要がある。

さらにローテーションなどが可能であれば、開発者にも SQA 活動に参加してもらい、開発者の生の声やノウハウをチェックリストに反映し易くなれば、より開発者に役に立つものとして維持することが可能になる。

SQA チェックリストに基づいた監査などの SQA 活動の結果は、開発者へ確実にフィードバックし、開発者の信頼を得ることが非常に大切である。開発者の信頼を得られなければ、SQA 活動は形骸化する。さらに続けていく中で、SQA の役割を進化させていくことが重要と思われる。SQA の役割の進化の起点となる様に SQA チェックリストを育てていく必要がある。SQA の役割の進化の例を表 3 に示す。

表 3. SQA の役割の進化の例

レベル	目的	具体的な活動・役割
Ⅰ	ルールを守らせる	・ルール（標準・規定類）遵守にフォーカス ・第三者的に淡々とチェックし報告 ・チェックリストSQA、間接的SQA
Ⅱ	与えられたQCDを守らせる	・プロジェクトへのヒヤリングを実施 ・逸脱事項を開発者と協力して解決 ・インタビューSQA、直接観察SQA
Ⅲ	顧客満足を得る	・品質の定量的なデータに基づくコンサルテーション ・継続的な改善活動を支持 ・コンサルティングSQA、プロジェクトの相談役

６．目標の達成度合いと反省点ならびに今後の課題

６．１　目標の達成度合いと反省点

チェックリスト作成の観点については当初の目的を達成できた。より効果的にするために、ベストな実施タイミングの検討、運用実績を基にした検証ができればよかったがそれに至らなかったのは残念である。

６．２　今後の課題

このチェックリスト（の項目）を基に SQA 担当者と開発者が課題の検討や改善に関して議論できるかどうかは開発現場との協力関係の構築、SQA 担当者のスキルによるところが大きい。誰でもができるような工夫が必要になってくる。

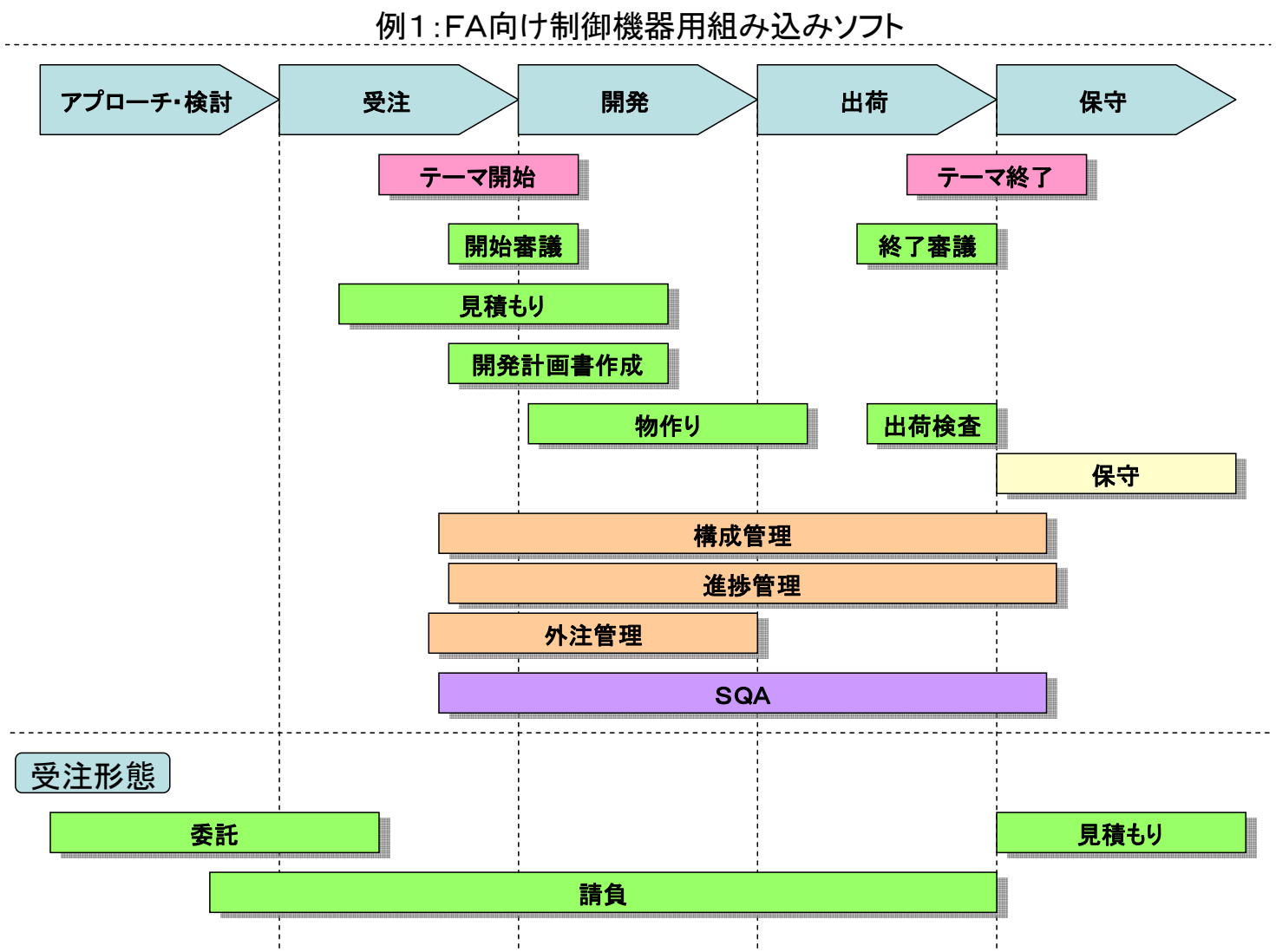
また、チェックリストの作成はあくまでも通過点であり、活動を通しての改善（フィードバック）が重要になってくる。この活動を定着させていくことが重要となる。

参考文献

- 日本的デザインレビューの実際、日科技連、菅野文友他
- CMMI®モデルガイド、日刊工業新聞社、Dennis M. Ahern
- 組込みソフトウェア向け開発プロセスガイド、翔泳社、IPA/SEC、２００６
- 要求を仕様化する技術・表現する技術、技術評論社、清水吉男、２００５
- IPA/SEC：http://sec.ipa.go.jp/index.php

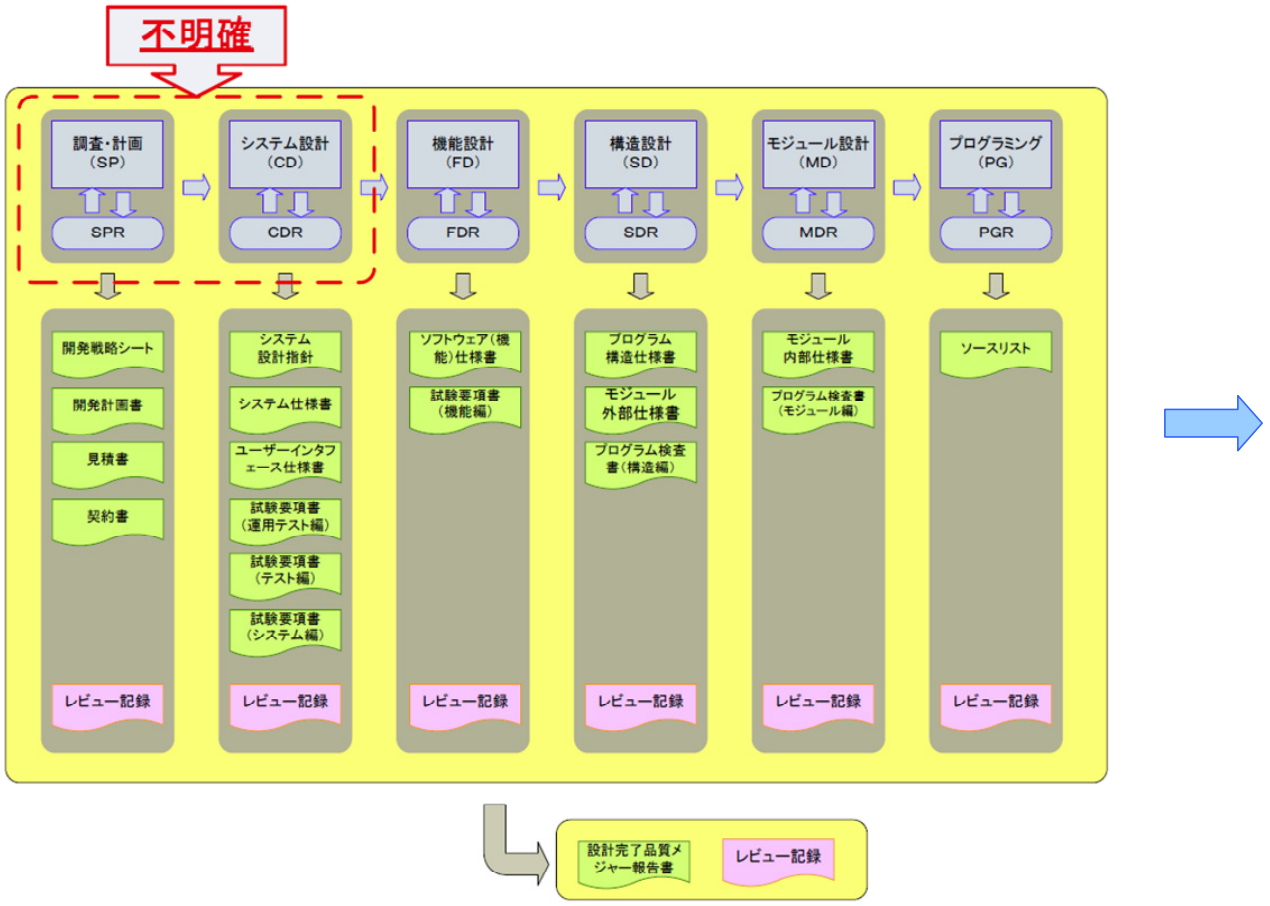
（注１）：®- CMMI および Capability Maturity Model は、米国特許商標局に登録されています。

参考資料１（例１）


例１:FA向け制御機器用組み込みソフト
アプローチ･検討
受注
開発
出荷
保守
テーマ開始
テーマ終了
開始審議
終了審議
見積もり
開発計画書作成
物作り
出荷検査
保守
構成管理
進捗管理
外注管理
SQA
受注形態
委託
見積もり
請負

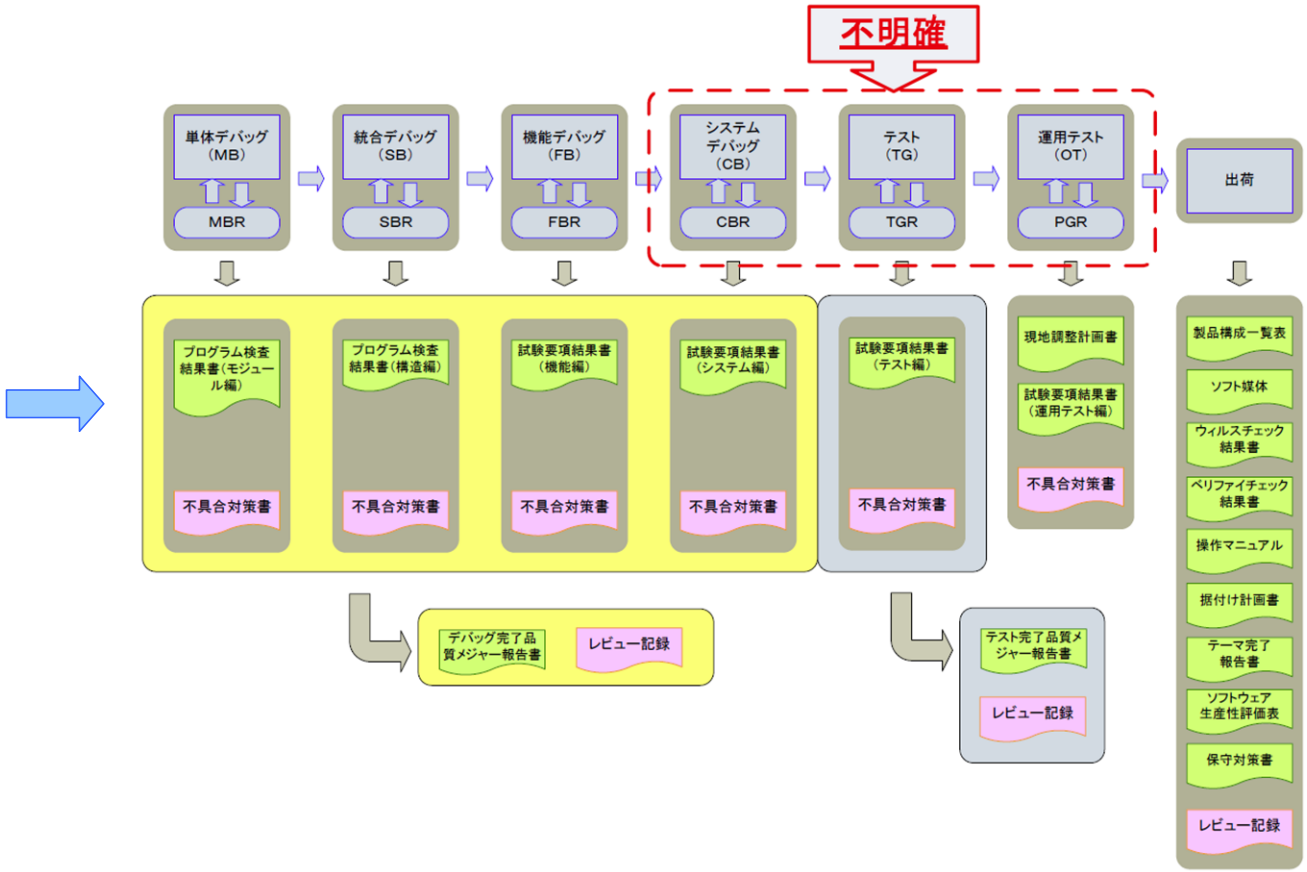
例１：FA向け制御機器用組み込みソフト（開発プロセス詳細①）

調査・計画（SP） ⇒ プログラミング（PG）



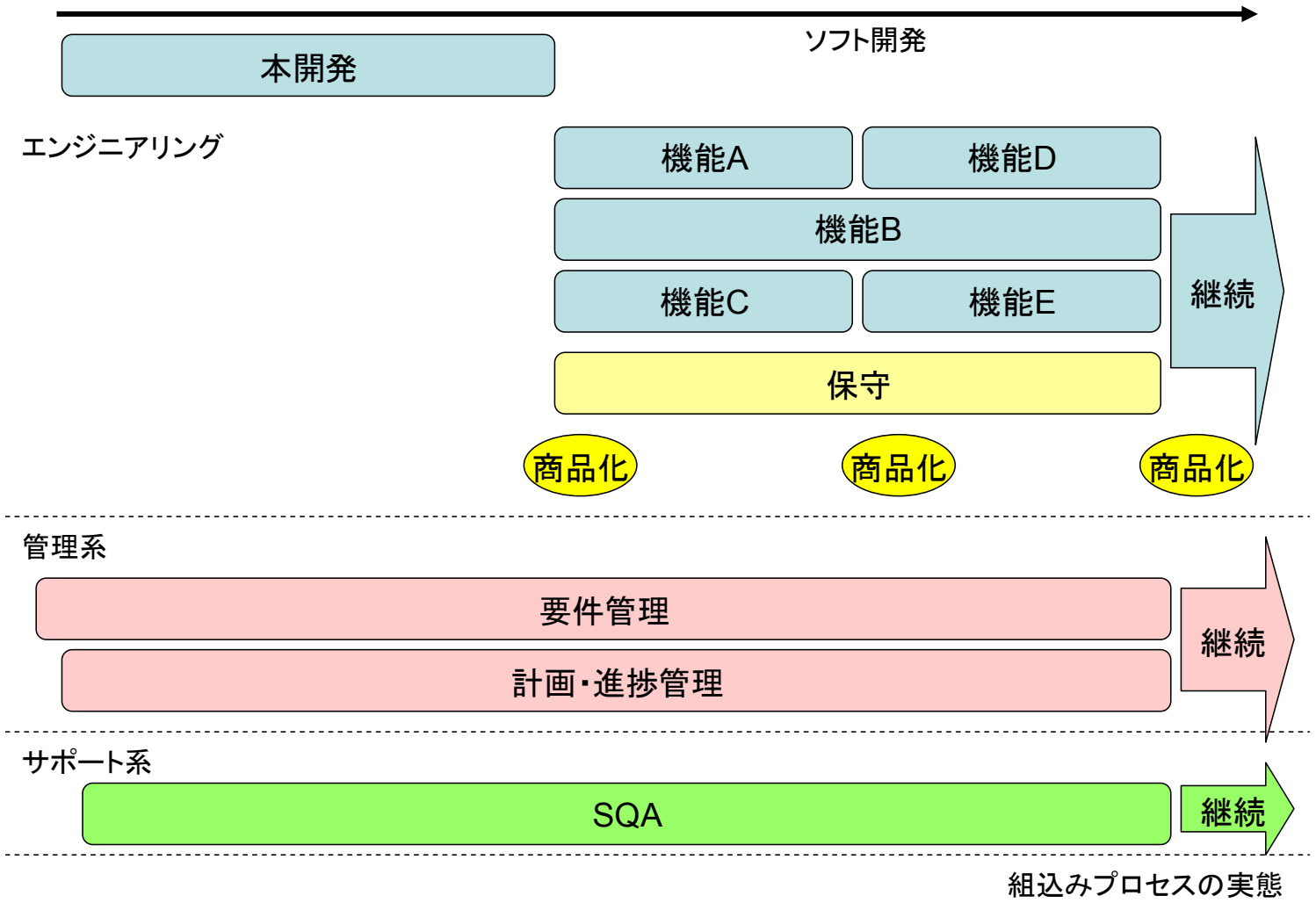
例１:FA向け制御機器用組み込みソフト(開発プロセス詳細②)

単体デバッグ(MB) ⇒ 運用テスト(OT)



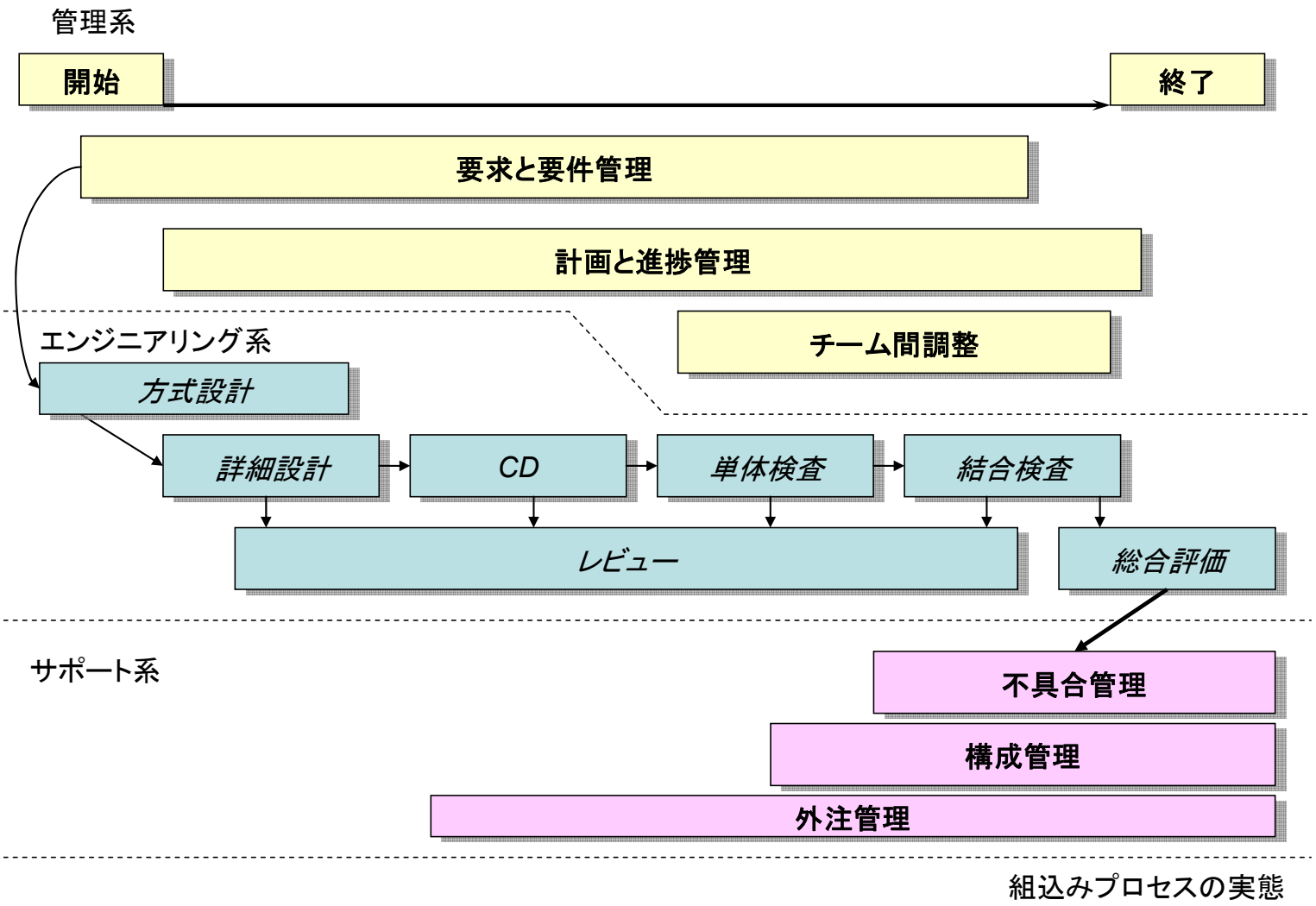
参考資料１（例２）

例2：移動体通信用計測器の組込みソフト開発



参考資料１（例３）

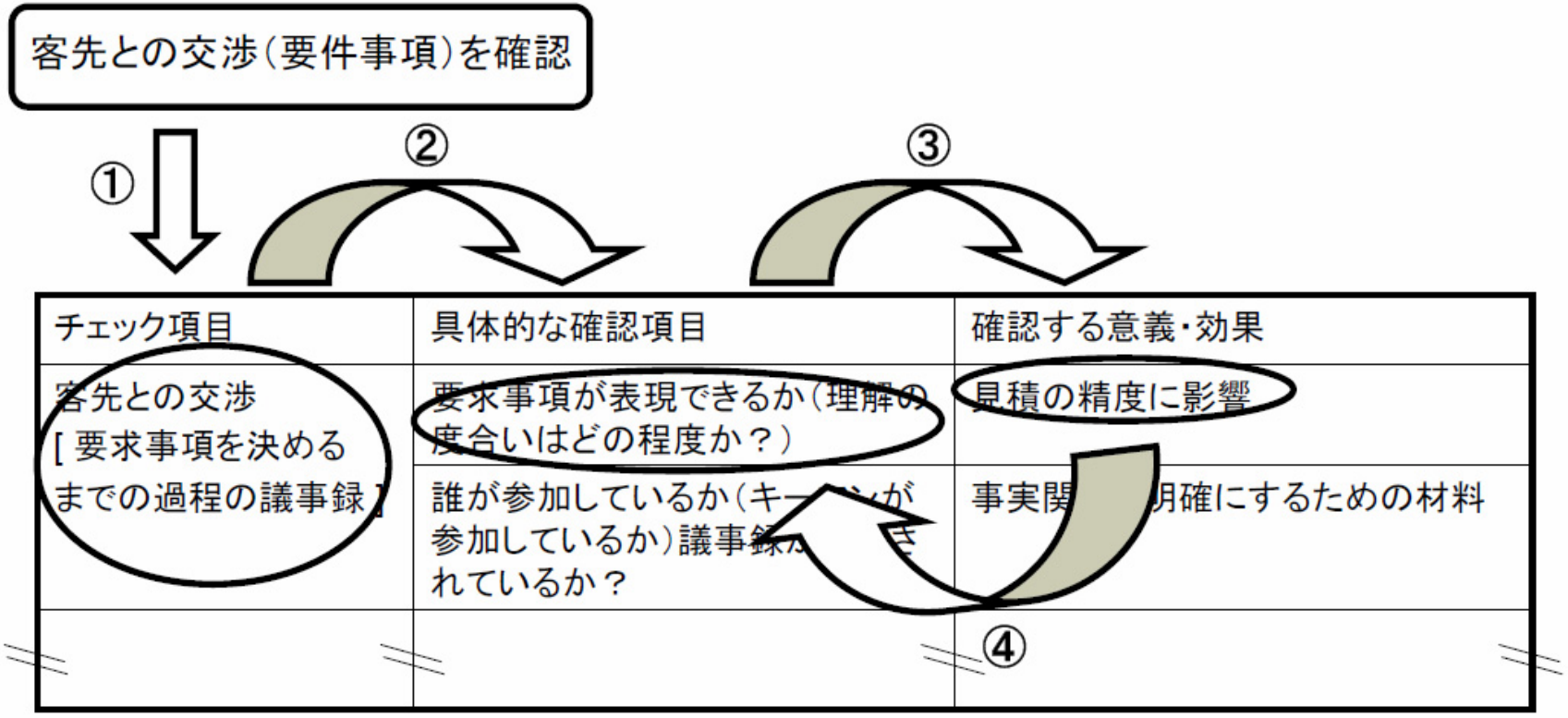
例3:コンシューマ向けのAV機器向け組込みソフト開発



組込みプロセスの実態

添付資料２

SQA チェックリストの作成例



添付資料３

SQA チェックリスト

チェック項目	具体的な確認項目	確認する意義・効果
客先との交渉 [要求事項を決めるまでの過程の議事録]	要求事項が表現できるか（理解の度合いはどの程度か？）	見積の精度に関係してくる
	誰が参加しているか（キーマンが参加しているか）議事録が保管されているか？	事実関係を明確にするための材料
	日時や時間、予算など	事実関係を明確にするための材料
	ハードとの切り分けは明確になっているか(どこまでが責任範囲か)？	事実関係を明確にするための材料
	要求事項の実現性（新規性）はどうか？	実現に関するリスク度合いに関係してくる
	リスク事項はあるのか？（あげる努力をしているか？）	実現に関するリスク度合いに関係してくる
	客先から納期の希望や重要なマイルストーン要求事項の優先度があるのか？	スケジュールの策定に関係してくる
	互換性や処理性能の希望はあるのか？	表面化されていない事項（当たり前と考えられている事項）を明確にし漏れを防ぐ（明確な基準が示されていること）
	要求事項に関して認識が一致しているか？	事実関係を明確にする
	要求事項が引き出せているか？	さらに考えるため。よりよい物にするため。
	客先との分担	責任範囲の明確化。
	ハードウェアスケジュールは確認できているか？	スケジュール管理
	デバッグ・テスト機材は確保できているか？（ハード開発とソフト開発のタイミングや代替手段ができているか？）	環境構築などをあきらかにするため。
要件の文書化	要件仕様書の記載事項は十分か？ （仕様・受注工程範囲・納期・出荷物など）	作業内容および見積範囲の明確化

チェック項目	具体的な確認項目	確認する意義・効果
	要件レビューチェックリスト（過去の経験から押えておいたほうがよいポイントをチェックリストとしたもの）の内容にNGがないか？	過去と同様のミスを防ぐ
ドキュメントとしての機能しているか	機能用件だけでなく非機能用件（中でも品質要求）は記載されているか？	品質要求が忘れられることが多い
	要件に固有の記番号がついているか？	要件の管理に必須となる
	要件の全体像が表現されているか？	要件の漏れ防止。要件範囲の明確化
	要件未定部分が明記されているか？	要件の漏れ防止。顧客との認識合わせ。 ステータス（状態）がどのような状況でこの後のストーリーは？
	性能要件が記載されているか？	性能定義が抜けることがある。
	このドキュメントへのポインタが明確か？（参照しやすい状況か？） 要件（背景、課題、環境等の条件）とその理由が明確か？ 下流ドキュメントへのポインタが明確か？ 改版履歴、執筆者、担当者へのポインタが明確か？	最上流ドキュメント決定。 理由が抜けがちなので明らかにしたい。 最低でも、人手で辿れるポインタを 確保。
要件の受け入れ（確定）	システム要件が明確になっているか？	担当範囲・環境の明確化
	テーマ担当者による要件レビューやシステム要件レビューは実施されているか？	テーマ担当者の要件の理解度
	課題やリスク事項の有無	課題やリスクの明確化と打ち手
	要件のベースラインへの登録	要件の確定
	事前レビューを実施したか？	要件の漏れや欠陥・解釈の違いを見つける
	要件確定のレビュー会を実施したか？	関係者の理解と合意を得る、漏れや矛盾を防ぐ

チェック項目	具体的な確認項目	確認する意義・効果
	代表者（責任者）のサインがあるか？	ベースラインを設定する
	受け入れレビューを実施しているか？	要件受け入れ前後では顧客との対応が変わってくる。
	責任者が確認しているか（捺印）？	組織として必須。
	リスクの洗い出しや整理をしているか？	リスク規模を見積りに反映する必要がある。
	要求解析と対応策の理由（等の重要項目）が明確か？ 要件と、それを満たす機能との対応が明確か？ 未対応要求があれば、その理由と承認が明確か？ 期日、担当、優先度などあれば、それらが明確か？	要求対応マップ（機械的な漏れチェック機構）確保。 漏れチェックの検証を確認。
要件の変更	変更の影響範囲は検討されているか？	作業内容など、影響範囲の明確化
	開発計画が更新されているか？	進捗管理の元となるため押えておく必要がある
	要件変更のプロセスは、顧客と合意されているか？	無節操な要件変更を制御する（誰が受けて誰が変更するのか明確にしておく）
	顧客との要件変更手順を決めているか？	両者の責任者が確認。記録に残す。
	影響度や優先度を検討しているか？	日程や予算など再計画を検討する。
	要件の構成管理手順が決められているか？	最新要件の明確化。チーム内の周知徹底。
	変更プロセスと、それによる影響が明確か？ 変更の理由が明確か？ 変更前後のレビジョン情報（等の重要項目）が明確か？	変更時のプロセスを明確にする。 事後のトレーサビリティを確保。
要件の追跡性	追跡マトリクスが作成されているか？	要件と成果物との関係を明確にする
	要求仕様の記番号から要件が特定できるか？	要件の実現の追跡を可能とする
	変更要求仕様の記番号から要件が特定できるか？	要件変更の追跡を可能とする
	要件の項番管理がなされているか？	要件変更の追跡性、設計工程への追跡性確保。

チェック項目	具体的な確認項目	確認する意義・効果
	保守性の高い版管理か？ 「明確か？」とされている物には常にアクセス可能か？ 「アクセス可能」へのステップ数が規定範囲内か（厳しいか？） 全ドキュメントに対して検索可能か？（ここには相応しくないが、 下流ドキュメントに上流へたどるポインタがあるか？も重要） バグトラックシステムはあるか？	ドキュメントトレーサビリティ、検索性を確保。 バグトラックはここで言うことかな？
成果物要件の開発	顧客要件が設計に落とし込まれているか？	仕様漏れを防ぐ
	顧客要件から要求仕様が展開されているか？	顧客要求を満たす
	市場要求や開発コンセプトが分析されているか？	顧客要件の妥当性検証。
	製品の利用形態や運用形態が明記されているか？	顧客要件の妥当性検証。
	提案するソリューションに対して承認を得られたか？ ユーザの作業フロー（または標準的なサンプル）確認したか？	ソリューションの確認があるか？ 状況を確保できているか？
	顧客要件から要求仕様が展開されているか？	顧客要求を満たす
	ユーザの作業フロー（または標準的なサンプル）を常時試行可能か？	常に顧客要求とのマッチングを試せるか？
要件の妥当性	要件の分析がされているか（レビュー記録）	実現の有無を含め検討されているか
	要件の妥当性が検討されているか	実現の有無を含め検討されているか
	顧客の利用環境が考慮されているか	ユーザー視点で考慮されている
	要件は適切に階層化（粒度）されているか？	重複や漏れ・矛盾を防ぐ
	要件毎に、（予想）サイズが見積もられているか？	工程管理の基本データとする
	関係者あるいは見識者のレビューを受けているか？	レビューの十分性を検証する。
	新規性およびその実現性が明確になっているか？	要件検討の十分性検証のひとつの項目。
	レビューとマイルストーンの設定基準が明確か？ レビューとマイルストーンが設定基準どおり設定されたか？	非論理的なレビュー、計画はさせない。