

■QCコンパス（「Quality-Characteristics」コンパス）の使い方

1. はじめに

QCコンパスは、「プロジェクト特性がSQuaRE品質特性に影響を与え、それに応じて最適化された品質管理手法を品質ボックスから選択し、テラリングに活用・適用することで、プロジェクトの成功に寄与する」という仮説をもとに作成した、プロジェクト開始前または開発プロセスのテラリング時に、QCコンパスを活用ください。

1-1. 想定する対象プロジェクト

プロジェクトの対象範囲は、開発するプロダクト（成果物）に基づく。

プロダクト全体（製品）を成果物とする場合、プロダクト全体の品質特性を考慮する。

プロジェクトの一部の機能を成果物とする場合、対象とする機能の品質特性を考慮する。

1-2. 想定する実施者

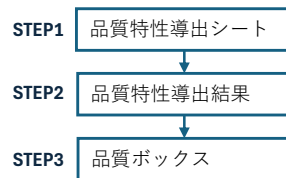
プロジェクト品質方針を把握し、全体の管理を担う方、または開発プロセスの検討や決定に関与する方

一例：プロジェクトマネージャー（PM）

ソフトウェア品質保証担当者（SQA）

2. 作業手順

「QCコンパス」は、Excelファイルの3つのワークシートを使って、3ステップで進める。



STEP1：プロジェクトの品質特性を導出するアンケート入力「品質特性導出シート」

プロジェクト開始前または開発プロセスのテラリング作業時に、プロジェクト品質方針に基づいて、プロジェクトマネージャー(PM)やソフトウェア品質保証担当者(SQA)が品質特性を導出するためのセルフアンケートを実施する。

このセルフアンケートは、品質特性を31の副特性に細分化しており、それぞれの副特性について当該プロジェクトがどの程度重要視しているかを評価する。

評価は5段階で行われ、1は「まったく重要視していない」、5は「非常に重要視している」となる。

なお、備考欄は、回答理由や判断に迷ったポイントなどを記録し、後で振り返りに活用できるメモとして利用する。

STEP2：導出された品質特性の評価結果とスコア調整を行う「品質特性導出結果」

評価はプロジェクトに求められる品質特性の度合いを示したものである。

評価の算出方法は、各設問に対する回答からスコアを計算し、品質特性ごとに総合スコア(割合)を集計する。次に、総合スコア(割合)に基づき、各品質特性の優先度を「高/中/低」の3段階で評価する。

- ・ 高評価：総合スコアが80%以上
- ・ 中評価：総合スコアが50%～80%未満
- ・ 低評価：総合スコアが50%未満

・ 結果の認識合わせ

プロジェクト品質方針が不明確な場合や、セルフアンケート者の役割や責任によって、重要とする品質特性に違いが出て、評価結果が異なる可能性がある。

そこで複数人で実施した場合、「品質特性導出結果」シートを使って、各実施者の評価結果を比較し、プロジェクト品質方針に対する認識のずれを調整する。

具体的には、品質副特性のスコアを見直し、プロジェクトで優先する品質特性を決定する。このとき、調整した品質副特性が分かるようにコメント欄に変更理由を記載する。

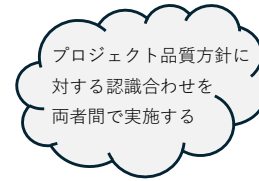
想定する使い方：

PM Aさん

品質特性	品質副特性	スコア	割合	評価
機能適合性	機能完全性	4	87%	高
	機能正確性	4		
	機能適切性	5		
性能効率性	時間効率性	1	33%	低
	資源効率性	2		
	容量満足性	2		
互換性	共存性	5	90%	高
	相互運用性	4		

SQA Bさん

品質特性	品質副特性	スコア	割合	評価
機能適合性	機能完全性	3	53%	中
	機能正確性	2		
	機能適切性	3		
性能効率性	時間効率性	1	33%	低
	資源効率性	2		
	容量満足性	2		
互換性	共存性	5	90%	高
	相互運用性	4		



調整結果

品質特性	品質副特性	スコア	割合	評価	コメント
機能適合性	機能完全性	3	53%	中	試作レベルで進めるためスコアを下げた
	機能正確性	2			
	機能適切性	3			

STEP3：品質管理手法と品質特性の対応付けを確認する「品質ボックス」

既存の概念「品質ボックス」は、縦軸に品質管理手法の知識領域を、横軸に品質特性を配置し、それらの関係性を示している。

例えば、「知識領域(要求分析の技法)：要求分析」は4つのトピックスを有している。

下記結果では、各トピックスと品質特性の関係性やプロジェクトにおける品質特性の優先度「高/中/低」を、セルの色分けで示し、視覚的に把握し易くしている。

SQxBOK			ソフトウェア品質特性							
知識領域	副知識領域	トピックス	システム/ソフトウェア製品品質							
			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
5 要求分析の技法	2 要求分析	1 機能要求分析	高					中		
		2 非機能要求分析		中		中	低	中	中	
		3 品質機能展開	高	中	低	中	低	中	中	低
		4 要求可変性分析							中	

※品質ボックスは、ソフトウェアの品質特性を体系的に整理し評価するためのモデルである。

※ソフトウェアの品質を多面的に捉え、品質特性とそれを実現するための品質技術や手法を関連付け整理している。

※品質ボックス：ソフトウェア品質技術が品質特性に与える効果の見える化

3. QCコンパスの使用例

使用例1 テーラリング結果を品質面で検証

対象：プロジェクト

実施者：1人以上

自社の標準プロセスからテーラリングした結果(プロジェクトで採用する開発プロセス)と「品質ボックス」で示された品質管理手法を見比べる。テーラリング結果の開発プロセスにおいて、品質管理手法が網羅する優先度の高い品質特性を考慮し、品質管理手法の適用を検討する。品質管理手法の具体的な説明は、SQuBOKガイドを参照する。

※SQuBOK (Software Quality Body of Knowledge) は、ソフトウェア品質に関する知識を体系的にまとめたガイドである。

使用例2 品質面でのレベル分け(コスト高、納期延伸などトレードオフに使う)

対象：プロジェクト

実施者：1人以上

プロジェクトの開発コストやスケジュールに対して強い制約があるなかで、プロジェクトの品質面を意識して開発プロセスの見直しを実施する。例えば、優先度の低い品質特性で使われている品質管理手法の適用を見送る、品質特性の網羅率の高い品質管理手法へ適用を変える。

使用例3 PMやSQAなど複数人との間での認識合わせ

対象：プロジェクト

実施者：2人以上

PMやSQAなどが個々で「品質特性導出シート」を実施し、お互いの評価結果を見比べてプロジェクト品質方針に対する認識のずれを確認する。認識合わせを行った上で、改めてテーラリング作業を実施し、開発プロセスの見直しを行う。

使用例4 複数プロジェクトの評価結果を基にした品質管理技術の標準化

対象：組織

実施者：SQA

SQAが適用を見送った品質管理手法を整理し、複数のプロジェクトで同様の事例が確認された場合は、組織として包括的に求められる品質特性および関連する品質管理手法と判断して、標準プロセスへの採用を検討する。

想定例：

下記図のトピックス「3.品質機能展開」は複数の品質特性と関係を示している。その組織では品質機能展開の実績がないため適用を見送っていたが、複数プロジェクトで「3.品質機能展開」が網羅する品質特性が常に高い評価を示したため「3.品質機能展開」を標準プロセスに組み込む判断をした。

SQuBOK			ソフトウェア品質特性								
知識領域	副知識領域	トピックス	システム/ソフトウェア製品品質								
			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	
5	要求分析の技法	2 要求分析									
		1 機能要求分析	高					中			
		2 非機能要求分析		中		中	低	中	中		
		3 品質機能展開	高	中	低	中	低	中	中	低	
		4 要求可変性分析							中		

品質特性導出シート（STEP1）

品質特性の優先度を導出するためのアンケートです。回答をお願いします。

判断根拠等ありましたら、備考欄に記入してください。

開発予定の製品/システムにおいて、PMやSQAの立場で品質面に対して、どの程度重要視しているかを回答ください。

(1:まったく重要視していない - 3:どちらでもない - 5:非常に重要視している)

No.	品質特性	品質副特性	設問	回答	備考
1	機能適合性	機能完全性	この製品/システムがユーザーに必要なすべての機能を提供することをどれくらい重要視していますか？ (例：基本機能、オプション機能、機能カバレッジ、要求実装率)	5	回答を選択した理由や判断に迷った内容など後で振り返れるようにメモとして利用する。
2		機能正確性	この製品/システムの各機能が誤りなく期待通りに動作し、計算やデータ処理の結果が正確であることをどれくらい重要視していますか？ (例：機能テストのバグの少なさ、申告不具合除去率)	5	
3		機能適切性	この製品/システムが特定の目的や状況に適した機能を提供することをどれくらい重要視していますか？ (例：利便性、ユーザーフレンドリー、シナリオ対応、試験数目標達成率)	5	
4	性能効率性	時間効率性	この製品/システムがタスクを迅速に完了することをどれくらい重要視していますか？ (例：応答速度、処理/転送速度、遅延の少なさ、高速パフォーマンス、リアルタイム処理)	4	
5		資源効率性	この製品/システムがリソースを効率よく使用することをどれくらい重要視していますか？ (例：メモリ効率、CPU使用率、バッテリー寿命、リソース最適化、低消費電力)	4	
6		容量満足性	この製品/システムが大量データや高負荷に耐えることをどれくらい重要視していますか？ (例：同時ユーザー数、大量データ対応、システムの拡張性、計画的な容量管理、負荷テスト)	3	
7	互換性	共存性	この製品/システムが他のシステムやアプリケーションと共存できることをどれくらい重要視していますか？ (例：セキュリティソフト等との複数アプリ同時実行、システム干渉の少なさ、互換性テスト、リソース共有、環境適応)	5	
8		相互運用性	この製品/システムが他の製品/システムと連携して動作できることをどれくらい重要視していますか？ (例：API連携、データ互換性、プロトコル対応、他システム接続、インテグレーション、インポート/エクスポート対応)	1	
9	使用性	適切度認識性	この製品/システムがユーザーにとって直感的で、すぐに理解できることをどれくらい重要視していますか？ (例：分かりやすいメニュー、操作ガイド、ヘルプ機能、直感的なナビゲーション)	3	
10		習得性	この製品/システムがユーザーにとって学びやすいことをどれくらい重要視していますか？ (例：チュートリアル、使いやすいガイド、学習コストの低さ、トレーニング資料、直感的操作)	3	
11		運用操作性	この製品/システムがユーザーにとって操作しやすいことをどれくらい重要視していますか？ (例：シンプルな操作、エラーメッセージの分かりやすさ、カスタマイズ可能な設定、音声操作、タッチ操作、元に戻せるUndo機能)	3	
12		ユーザーエラー防止性	この製品/システムがユーザーの誤操作を防止することをどれくらい重要視していますか？ (例：確認メッセージ、自動保存機能、誤操作防止設計、エラー/警告メッセージ、復元機能)	3	
13		ユーザーインターフェース快美性	この製品/システムのユーザーインターフェースが美しく魅力的であることをどれくらい重要視していますか？ (例：デザインの一貫性、視覚的魅力、UIの使いやすさ、インターフェースのシンプルさ、ユーザー満足度)	3	
14		アクセシビリティ	この製品/システムがハンディキャップを持つユーザーにとっても使いやすいことをどれくらい重要視していますか？ (例：音声読み上げ対応、キーボード操作、スクリーンリーダー対応、色覚バリアフリー、多言語対応、ユーザーフレンドリーなデザイン)	3	
15	信頼性	成熟性	この製品/システム全体が長期間に渡り安定して信頼性高く動作し続けることをどれくらい重要視していますか？ (例：製品バグの少なさ、長期安定稼働、信頼できるパフォーマンス発揮)	5	
16		可用性	この製品/システムが常に利用可能であることをどれくらい重要視していますか？ (例：24時間365日利用、ダウンタイムの少なさ、保守計画、冗長化、障害対応)	5	
17		障害許容性（耐故障性）	この製品/システムが部分的に故障しても、システム全体が機能し続けることをどれくらい重要視していますか？ (例：故障部分を即座にフェイルオーバー、冗長設計、バックアップ、エラーハンドリング、自動復旧)	3	
18		回復性	この製品/システムが故障後、迅速に元の状態に回復することをどれくらい重要視していますか？ (例：データ復元、システムリカバリ、バックアップシステム整備、復旧時間、効率的な復旧プロセス)	4	

品質特性導出シート（STEP1）

品質特性の優先度を導出するためのアンケートです。回答をお願いします。

判断根拠等ありましたら、備考欄に記入してください。

開発予定の製品/システムにおいて、PMやSQAの立場で品質面に対して、どの程度重要視しているかを回答ください。

(1:まったく重要視していない → 3:どちらでもない → 5:非常に重要視している)

No.	品質特性	品質副特性	設問	回答	備考
19	セキュリティ	機密性	この製品/システムがユーザーのデータや個人情報を安全に守ることをどれくらい重要視していますか？ (例: データ暗号化, アクセス制限, ユーザー認証, セキュリティポリシー策定, データ保護の仕組み)	1	
20		インテグリティ	この製品/システムがデータの一貫性と正確性を保ち、改ざんや不正アクセスから守ることをどれくらい重要視していますか？ (例: データ整合性チェック, トランザクション管理, エラーチェック, データ改ざん防止, 監査ログ)	1	
21		否認防止性	この製品/システムが操作履歴を正確に記録し、不正行為を防ぐことをどれくらい重要視していますか？ (例: 操作ログ記録, 監査証跡の保存, ユーザー認証システム, アクセス履歴管理, データ改ざん防止)	1	
22		責任追跡性	この製品/システムがユーザーの身元を確認し、アクセスを適切に管理することをどれくらい重要視していますか？ (例: 多要素認証, パスワードポリシー, アクセス制御, ユーザー管理, ログイン履歴)	1	
23		真正性 (データや通信が改ざんされていないことの確認)	この製品/システムがデータ・通信の信頼性を確保することをどれくらい重要視していますか？ (例: デジタル署名, 証明書提供, 認証プロトコル実装, セキュリティトークン活用, データ検証の仕組み導入)	1	
24	保守性	モジュール性	この製品/システムが部品ごとに分かれ、それぞれが独立して動作することをどれくらい重要視しますか？ (例: 独立したコンポーネント, モジュール間結合の少なさ, モジュラー設計, 部品の交換容易さ, 機能の独立性)	2	
25		再利用性	この製品/システムの部品やコードが他のプロジェクトで再利用できることをどれくらい重要視しますか？ (例: モジュール化, ライブラリ化, API提供, 共通コンポーネント, コード再利用)	1	
26		解析性	この製品/システムが問題を発見し、原因を簡単に特定できることをどれくらい重要視しますか？ (例: ログ分析, エラーメッセージ表示, デバッグツール利用, 監視システム導入, トレース機能)	4	
27		修正性	この製品/システムが簡単に変更やアップデートできることをどれくらい重要視しますか？ (例: モジュラー設計, 設定ファイルで変更, パラメータで変更, カスタマイズ機能充実, バージョン管理)	1	
28		試験性	この製品/システムがテストしやすいことをどれくらい重要視しますか？ (例: テストケース設計, テスト自動化, テスト環境整備, テストカバレッジ, リグレーションテスト)	3	
29	移植性	適応性	この製品/システムが様々な環境や条件に柔軟に対応できることをどれくらい重要視していますか？ (例: 設定管理のしやすさ, カスタマイズの簡単さ, 設定変更の柔軟さ, 異なる環境への迅速な適応, 様々な運用条件での安定性)	3	
30		設置性	この製品/システムが新しい環境やシステムに容易に設置できることをどれくらい重要視していますか？ (例: インストール手順の簡単さ, 設定の柔軟性, ドキュメントの整備, インストール時のエラーチェック, 初期設定のサポート)	1	
31		置換性	この製品/システムが特定の部分や機能を他のものと簡単に置き換えられることをどれくらい重要視していますか？ (例: 交換可能なコンポーネント, インターフェースの標準化, 部品交換の容易さ, 互換性のある部品, システムの柔軟性)	1	

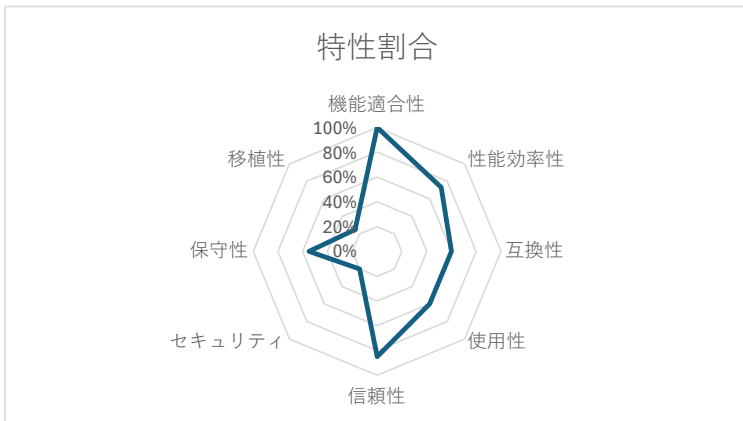
セルフアンケートは以上です。お疲れ様でした。

【付録1. QCコンパス】 品質特性導出結果（STEP2）

品質特性導出結果（STEP2）

あなたのプロジェクトの品質特性の優先度は以下になりました。

品質特性	品質副特性	スコア	割合	評価	コメント
機能適合性	機能完全性	5	100%	高	スコア調整時の変更理由を記載する。
	機能正確性	5			
	機能適切性	5			
性能効率性	時間効率性	4	73%	中	
	資源効率性	4			
	容量満足性	3			
互換性	共存性	5	60%	中	
	相互運用性	1			
使用性	適切度認識性	3	60%	中	
	習得性	3			
	運用操作性	3			
	ユーザーエラー防止性	3			
	ユーザーインターフェース	3			
	快美性	3			
信頼性	成熟性	5	85%	高	
	可用性	5			
	障害許容性（耐故障性）	3			
	回復性	4			
セキュリティ	機密性	1	20%	低	
	インテグリティ	1			
	否認防止性	1			
	責任追跡性	1			
	真正性	1			
保守性	モジュール性	2	55%	中	
	再利用性	1			
	解析性	4			
	修正性	1			
	試験性	3			
移植性	適応性	3	25%	低	
	設置性	1			
	置換性	1			



SQuBOK Guide (第2版) ベース						システム/ソフトウェア製品品質									
カテゴリ	副カテゴリ	知識領域	副知識領域	トピックス	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			
専門的品質特性のソフトウェア品質技術	10	品質分析・評価の技法		5 探針【日立】	高				高						
				2 品質進捗管理に関する技法	1 異常値管理										
					2 工数・成果モデル										
					3 コード統合モデル										
					4 PTRサブモデル										
					5 PTR発生及びバックログ予測モデル										
					6 Rayleighモデル	高				高					
				3 障害分析に関する技法	1 ODC (直交欠陥分類)	高		中	高						
					2 バグ分析	高			高						
				4 データ解析・表現に関する技法	1 QC七つ道具										
					2 新QC七つ道具										
					3 多変量解析										
					4 EDA (探索的データ解析) 手法										
					5 p管理図, u管理図										
					6 品質座標										
	7 レーダーチャート	高	中			中	高				中	低			
	11 運用の技法	1-			1 仮想化			中		中			中		
					2 クラウドによるサービス提供			中			高	低	中		
					3 ソフトウェア若化			中			高				
					1 プログラム理解									中	
	12 保守の技法	2-			2 リエンジニアリング			中				中			
					3 リバースエンジニアリング				中			中			
					4 コードクローン分析							中			
					1 ユーザビリティテスト						中				
	13 使用性の技法	1-			2 インタラクティブシステムの人間中心設計プロセス (ISO9241-210)					中					
					1 セーフティ実現のためのリスク低減技法			中		中	高	低			
	14 セーフティの技法	1-			2 セーフティ・クリティカルシステムのテスト					高					
					1 セキュリティ要求分析			中		中	高	低			
	15 セキュリティの技法	1-			2 セキュリティ設計					高	低				
					3 セキュリティパターン					高	低				
					4 セキュアコーディング						低				
					5 セキュリティテスト						低				

1. 有効性確認のアンケート内容

・観点A：品質特性の導出がテラリングの最適化に有効か

問	アンケート内容
Q1	過去のプロジェクトで今回の結果(QCコンパスの品質管理手法)を使用していたら、品質とコストの両面で改善/最適化されたと思いますか？
Q1-1	Q1にて「4.とても」もしくは「3.やや」を回答された方への質問です 今後、QCコンパスを使いたいと思いますか？ また、D列にその回答の理由を記載してください。
Q1-2	Q1にて「2.あまり」もしくは「1.まったく」を回答された方への質問です D列にその回答の理由を記載してください。
Q2	QCコンパスをより有用にするためには、どのような改善が必要だと思われますか？
Q3	QCコンパスは、プロジェクト開始前のテラリングだけでなく、プロジェクト進行中の品質やコスト見直しにも活用できそうですか？
Q4	過去のプロジェクトで今回の結果(QCコンパスの品質管理手法)を使用することで防ぐことができた可能性のある不具合はありますか？ 例：「デザインパターン」を取り入れていたら防げた不具合があった 防げた可能性のある不具合があれば事例をD列に記載してください

・観点B：ステークホルダー間の合意形成の促進に有効か

問	アンケート内容
Q5	1プロジェクト2名以上で実施いただいた方への質問です 「品質特性導出結果」シートの「評価」欄に品質特性の重要度が表示されています。実施者間で導出された重要度に差がありましたか？ 差があった場合、D列に差異の内容とその理由を記載してください。
Q6	1プロジェクト2名以上で実施いただいた方への質問です 「QCコンパス」を使用することで、品質特性の重要度に対する認識の一致が図れそうですか？

2. アンケート回答者

・観点A

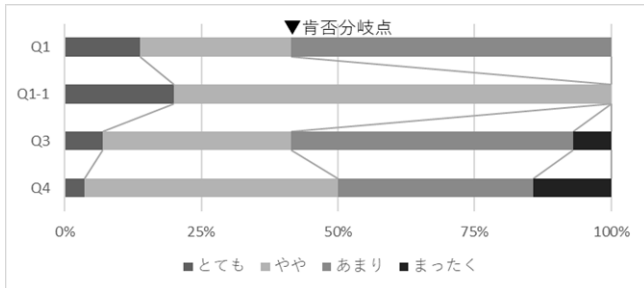
企業数：7社
実施人数：30名
プロジェクト側 (PM, PL)：22名
品質側 (SQA)：8名
プロジェクト数：21件

・観点B

企業数：5社
実施人数：15名
プロジェクト側 (PM等)：11名
品質側 (SQA等)：4名
プロジェクト数：11件

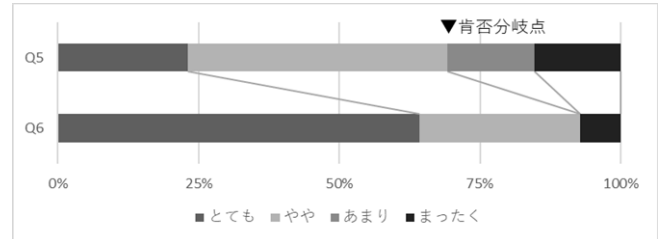
3. 有効性確認のアンケート結果 (4段階評価)

・観点A



※Q1-1は回答10名に対する割合
※Q2は文章回答のみのため除外

・観点B



4. 有効性確認のアンケート結果（自由記述）

アンケートの自由記述欄をもとに、回答者の「期待する意見」、「懸念する意見」を要約

・観点A：品質特性の導出がテラリングの最適化に有効か

問	期待する意見	懸念する意見
Q1	<p>不具合の教訓活用</p> <ul style="list-style-type: none"> ・現プロジェクトでの不具合を次回に活かすことで、類似不具合の防止が期待できる。 <p>SQAとの認識統一</p> <ul style="list-style-type: none"> ・プロジェクト内でのSQAとの認識が合うことで、プロセス改善が促進される。 <p>品質と技法への新たな気づき</p> <ul style="list-style-type: none"> ・品質ボックスの利用で新たな技法の発見に繋がり、改善が見込める。 	<p>効果の限界</p> <ul style="list-style-type: none"> ・過去のプロジェクトの知見を活かしており、新たな技法の導入による大きな効果は期待できない。 <p>顧客要求の不確実性</p> <ul style="list-style-type: none"> ・顧客要求が曖昧で開発中に変わるため、効果が限られる。 <p>選択と集中の難しさ</p> <ul style="list-style-type: none"> ・多数の技法の中で効果的なものを見極めることが難しく、チャレンジ精神が必要。
Q1-1	<p>プロジェクトの初期での合意形成</p> <ul style="list-style-type: none"> ・QCコンパスを使用することで、プロジェクト開始時にメンバーやステークホルダーとの合意形成が可能になる。 <p>品質目標の明確化</p> <ul style="list-style-type: none"> ・プロジェクト開始時に品質ゴールや品質特性が明確になり、指標として活用できる。 <p>エラー削減や時間短縮</p> <ul style="list-style-type: none"> ・作業の自動化や効率化が進み、品質向上やコスト削減に繋がり、プロジェクト全体のパフォーマンス向上が期待できる。 	<p>外部仕様書との整合性</p> <ul style="list-style-type: none"> ・QCコンパスは、既存の外部仕様書に記載済みの情報と重複することが考えられる。 <p>項目の適用性の再確認</p> <ul style="list-style-type: none"> ・対象システムやプロジェクトに応じて、シートの項目を見直す必要がある。 <p>特定の課題やリスクの適応性</p> <ul style="list-style-type: none"> ・チェックシートとしてのみ受け取られ、柔軟な問題管理やリスク管理への適応が課題として指摘される可能性がある。
Q1-2	<p>プロジェクト初期での共通認識形成</p> <ul style="list-style-type: none"> ・新規プロジェクトやメンバー変更時に共通意識の形成に効果があると考えられる。 <p>テラリング作業の補助</p> <ul style="list-style-type: none"> ・品質特性とSQuBOKの関係が理解でき、テラリング作業を迅速に行う助けとなる。 <p>プロジェクト特性の可視化</p> <ul style="list-style-type: none"> ・適切な回答でプロジェクト特性を視覚化でき、新規プロジェクトでの参考として有用。 	<p>品質特性の抽象性と具体性欠如</p> <ul style="list-style-type: none"> ・品質ボックスの内容が抽象的で、具体的にどのように対応すべきかが不明瞭。 <p>コストと効果のバランス</p> <ul style="list-style-type: none"> ・新たに理解と教育、実施プロセスの策定に多大なコストがかかるため、現行プロセスへ簡単に導入しづらい。 <p>プロダクト全体での品質特性の差異</p> <ul style="list-style-type: none"> ・部品ごとの品質特性の重要度に差が出るため、全体での適用に困難がある。
Q2	<p>プロジェクト特性の可視化と不具合防止の効率化</p> <ul style="list-style-type: none"> ・初期特性に基づく類似不具合事例の紹介が、テラリングの効率化と不具合防止に役立つ。 <p>認識を共有する仕組み</p> <ul style="list-style-type: none"> ・複数人で品質特性導出結果を比較しやすくする仕組みの導入が、認識を合わせる場面で有用。 <p>組織の重要視品質特性と強化方針の明確化</p> <ul style="list-style-type: none"> ・組織として重要視すべき品質特性を決定し、それを強化する方針があれば、コスト確保が容易。 	<p>専門用語の多用と理解の難しさ</p> <ul style="list-style-type: none"> ・設問の専門用語が多く理解が難しいため、より平易な表記が必要である。また、社内で統一された用語理解が必要となる。 <p>具体的な対応策の提示の難しさ</p> <ul style="list-style-type: none"> ・「高」や「低」に対する具体的な対応策が明示されないため、その部分がイメージしづらく、改善に繋がりにくい。 <p>設問の汎用性と視点の制限</p> <ul style="list-style-type: none"> ・設問が汎用的過ぎて、開発プロジェクト特性に応じて具体性に欠けるため、視点を強化する改善が必要。
Q3	<p>プロジェクトの方針提示や計画の軸としての活用</p> <ul style="list-style-type: none"> ・プロジェクト開始前から利用すれば、見直し時に変更の判断軸として有用。 <p>長期プロジェクトや特定のタイミングでの活用</p> <ul style="list-style-type: none"> ・方針が変わるタイミングなどでの活用が可能であり、プロジェクトの軸をぶらさない形で見直しができる。 <p>テスト設計の方針変更の参考</p> <ul style="list-style-type: none"> ・具体的な改善分析として、特にテストに重点を置く決定に活用できる可能性。 	<p>プロジェクト進行中の効果期待が低い</p> <ul style="list-style-type: none"> ・変更が少ないプロジェクトでは進行中の利用に効果を見込めないことが多いと考えられる。 <p>プロジェクト計画での検討事項との重複</p> <ul style="list-style-type: none"> ・プロジェクト進行中の見直しでは使いどころが難しくなる可能性が高い。 <p>見直し時にゴールのぶれリスク</p> <ul style="list-style-type: none"> ・PJ進行中に見直すとゴールがぶれる可能性があり、見直しのタイミングに注意が必要。
Q4	<p>ユーザ環境シミュレーションテストの導入</p> <ul style="list-style-type: none"> ・ユーザ環境シミュレーションテストを取り入れれば、非ユースケースパターンのエラーを事前に検出できた可能性がある。 <p>状態遷移図に基づくテストの重要性</p> <ul style="list-style-type: none"> ・状態遷移図やテストを活用すれば、エラー系の曖昧さを減らし、特定の不具合を防げた場面があったと思われる。 <p>品質特性の認識と事前合意</p> <ul style="list-style-type: none"> ・使用性やユーザビリティに関する前提条件を明確化し、事前に関係者間で合意を取ることで、品質とコストのバランスをより効果的に管理できる。 	<p>非機能要件の曖昧さからの影響</p> <ul style="list-style-type: none"> ・非機能要件の曖昧さが原因でQCコンパスの品質管理手法では対策が足りないと感じるが、顧客要求に応じた特性分析が促進されればより効果的。 <p>具体的な成果や効果の不確実性</p> <ul style="list-style-type: none"> ・「QCコンパスの品質管理手法」がどのように具体的な不具合作業に直結するか、ははっきりしない部分がある。 <p>設計方針やメンバーの不变による効果期待の難しさ</p> <ul style="list-style-type: none"> ・前バージョンから設計方針やチームメンバーが変わらない場合は、導入による効果が期待できない。

・観点B：ステークホルダー間の合意形成の促進に有効か

問	期待する意見	懸念する意見
Q5	<p>認識一致の確認に有効</p> <ul style="list-style-type: none"> ・多くのケースで、数字の差異はあるものの、評価の考え方に大きな違いはなく、概ね合意できていることが確認できた。 ・性能効率性の重要度に小さな差があったが、その違いは参加者間で大きくなかった。 <p>認識ずれの解消が可能</p> <ul style="list-style-type: none"> ・評価の考え方を説明することで、参加者間で概ね合意に至ることができた。 	<p>顧客と組織方針の違いによる差異</p> <ul style="list-style-type: none"> ・顧客と組織が掲げる品質基準の違いから、評価に極端な差が生じたケースが存在。 <p>要求仕様書に明記されていない要素</p> <ul style="list-style-type: none"> ・要求仕様書に明記されていない要素があり、SQAがそれを汲み取れないため、差が生じる。 <p>定性的な要素の判断の困難さ</p> <ul style="list-style-type: none"> ・使用性などの定性的な要素に関して、明確な基準がなく、評価に差異が発生。
Q6	<p>認識の一致のための基盤</p> <ul style="list-style-type: none"> ・品質特性がスコア化・明示化されることで、認識の一致を図ることが可能。 <p>ディスカッションの促進</p> <ul style="list-style-type: none"> ・プロジェクト特性についての議論の機会を提供し、認識一致の議論を促進する。 <p>品質特性の理解と認識一致が可能</p> <ul style="list-style-type: none"> ・開発や企画、営業などの異なる立場間で、品質特性の重要度に対する認識を合わせるのに有効。 	<p>品質特性の知識前提の必要性</p> <ul style="list-style-type: none"> ・品質特性についてきちんと知識があることが前提となるため、そのための教育が必要。 <p>プロジェクトマネージャーのイニシアチブに依存</p> <ul style="list-style-type: none"> ・PMがイニシアチブを取るため、他のステークホルダーが関与しづらい可能性。 <p>1人実施の制約</p> <ul style="list-style-type: none"> ・2名以上での実施がない場合、認識の一致についての効果が測れない。