

# ESQR導入事例紹介

## —導入から適用と計測に至るタスクフォースの事例—

キヤノンソフトウェア株式会社 エンジニアリング事業本部第四エンジニアリング事業部  
技術士（情報工学部門）

羽部 高志

IPA/SECのトライアルプログラムを利用してコンサルティングを受けながら、「[改訂版]組み系ソフトウェア開発向け品質作り込みガイド」[ESQR Ver.1.1]<sup>※1</sup>で提唱されている品質のコントロール手法を約半年間に渡ってパイロット導入した事例を紹介する。

### 1 はじめに

ソフトウェア開発における品質の測定というテーマについて、とくにプログラム実装フェーズ以降の工程においては、単体テストのカバレッジ率、テスト密度やバグ密度、不具合収束率などの指標を用いることにより、定量的に成果物の品質計測を可能にする手法が既に確立されている。しかし、実装フェーズ以前の、要件定義フェーズから設計フェーズに至るいわゆる上流工程においては、主たる成果物がテキストベースのドキュメントであることから、品質の担保は主に有識者によるレビューに頼っているのが実情である。このため上流工程のプロセスや成果物の品質を定量的にコントロールすることが難しく、品質を作り込む上での課題となっていた。IPA/SECでは「組み系ソフトウェア開発向け品質作り込みガイド ESQR」をリリースしており、当該課題に対する一つのアプローチ方法が提唱されている。なおIPA/SECによるESQRなどの一連のESxRシリーズのリリースに当たっては、ソフトウェア開発現場への普及を図るための導入トライアル企業の募集を行っていたことから、弊社においてもこれに応募し、SECのコンサルティングを仰ぎながらとくに上流工程の定量的品質コントロールの実現を目指すこととした。具体的にはESQRの導入と検証を行うタスクフォースを立ち上げ、パイロット導入プロジェクトに対して品質指標の計測を行った。以下では、上記タスクフォースにおける導入事例を紹介する。

### 2 導入の目的

要件定義フェーズ、設計フェーズといった上流工程から品質を作り込むための定量的な品質コントロール手法を確立することを目的とする。ESQRの導入により、SECが集めたデータに基づく、妥当性のある品質目標が定まることになる。これにより、我々の組織のプロセスや成果物の品質を客観的に評価して改善活動に繋げていくことが可能になると考えるものである。

### 3 ESQRについて

まず今回トライアルの対象としたESQRについて簡単に紹介させていただく。なお詳細は、オリジナルの「[改訂版]組み系ソフトウェア開発向け品質作り込みガイド」[IPA-SEC 2012]（以下、ガイド）を参照されたい。

#### 3.1 品質指標に基づく組み系システムのソフトウェア開発

ソフトウェア開発プロジェクトは、成果物としてのシステムの特長や、プロジェクトを取り巻く外部環境、内部環境は個々に異なる。ESQRでは、まずこれらシステムの特長及びプロジェクト環境の特長を数値化して、大きく二つの観点、一つはプロジェクトを遂行するに当たっての「プロセスの品質」、もう一つはプロジェクトの成果物である「プロダクトの品質」それぞれに対して定量的な品質目標を設定することから始める。ESQRは

脚注

※1 ESQR : Embedded System development Quality Reference

あらかじめ品質目標設定の候補となる各種指標を用意しているため、プロジェクトの遂行に当たり、品質コントロールのために最適な品質指標を選択することが出来る。プロジェクト実施期間において、選択した品質指標を用いて「プロセス品質」及び「プロダクト品質」の計測を行い、2つの観点から目標との差異について傾向分析を行うことで、ソフトウェア品質のコントロールを実現することを意図するものである。

## 4 事前準備

ESQR 導入に当たっては、現場の負担を軽減するため、品質目標の決定やメトリクス of 採取は極力自動化することが望ましいのは自明である。このため、まずは ESQR 導入実施プロセスをサポートするためのツール類を整備し、これらツールの使用を前提として導入を進めることとした。

### 4.1 プロファイラの作成

まずは、ESQRに提示された各種の品質目標値の算出が簡単に出来るよう、プロファイリングツールを準備した。

最初のステップはシステムプロファイリングであるが、現場のPMにとっては、オリジナルのガイドに提唱されている「人的損失」、「経済的損失」の検証のみではシステムタイプを決定することが困難であると考えた。このため、弊社独自に「組み込み先への影響」、「組み込み処理の正確性」、「セキュリティレベル」、「リコール可能性」という4つの判断基準を新たに追加した。これにより担当PMが、システムの性質に合わせて用いる判断基準を適宜選択出来るようになる(表1)。

プロジェクトプロファイルについてもオリジナルのガイドに記載された10ファクターの他、弊社案件の特性を加味して新たに3つのファクターを追加した。更にPMが自らの判断によりプロジェクトに対して独自のファクターを追加出来るようにした(表2)。

表1 システムプロファイルにおけるシステムタイプ判断基準

システムタイプ	影響度	条件:品質問題が発生した場合、ユーザが被る損失					
		基本条件		基本条件によるプロファイルが困難な場合の代替条件			
		人的損失	経済的損失	組み込み先への影響	組み込み処理の正確性	セキュリティレベル	リコールの可能性
Type-IV Highly Critical	7:激	殆どのユーザに重大な健康被害が出る	ユーザの一部または全部が被る直接・間接の被害額が極めて大きい	品質問題が組み込み先へ影響し、ユーザが許容できないレベルで組み込み先の使用継続が不可能となることにより、社会的問題に発展する可能性あり	品質問題に起因して組み込み側の処理に誤りがあると、社会的問題に発展する可能性あり	組み込み側の品質問題に起因して、第三者への情報の流出、あるいは情報の破壊が発生すると、社会的問題に発展する可能性あり	組み込み側の品質問題に起因して製品のリコールが発生し、社会的問題に発展する可能性あり
	6:烈	多くのユーザに重大な健康被害が出る					
Type-III Critical	5:強	一部のユーザに重大な健康被害が出る	ユーザの一部または全部が被る直接・間接の被害額が¥10,000K以上	品質問題が組み込み先へ影響し、ユーザが許容できないレベルで組み込み先の使用継続が不可能となる可能性あり	品質問題に起因して組み込み側の処理に誤りがあると、一部または全部のユーザの使用に多大な影響が出る	組み込み側の品質問題に起因して、第三者への情報の流出、あるいは情報の破壊が発生すると、ユーザが被害を被る可能性あり	組み込み側の品質問題に起因して製品のリコールが発生し、一部または全部のユーザの使用に多大な影響が出る可能性あり
	4:中	一部のユーザに健康被害が出る					
Type-II Normal Quality Required	3:弱	一部のユーザに軽微な健康被害の可能性あり	ユーザの一部または全部が被る直接・間接の被害額が¥1,000K以上	品質問題が組み込み先へ影響し、代替手段により継続使用可能となるものの、不効率になる	品質問題に起因して組み込み側の処理に誤りがあると、一部または全部のユーザの使用に軽度の影響が出る		組み込み側の品質問題に起因して、一部のユーザの使用に軽度の影響が出るものの、個別の対応が可能で製品リコール発生の可能性は無い
	2:軽						
Type-I Normal	1:微		ユーザの一部または全部が被る直接・間接の被害額が¥1,000K未満	品質問題が組み込み先へ影響するものの、代替手段により継続使用が可能			
	0:無						

これらの入力情報を元にオリジナルのガイドに記載された算出式を実装し、プロファイリング結果を反映した品質目標値を提示出来るようにした。また、品質目標値の設定に当たっては、算出した値がガイドの記載に従って誤差範囲として±15%のレンジ幅を持つようにした(表3)。

#### 4.2 メトリクス収集のインフラについて

今回 ESQR のパイロット導入先を選定するに当たっては、既にインフラとしてBTS<sup>※3</sup>が導入され、これをベース

##### 脚注

- ※2 略称など、評価指標の詳細についてはガイド参照のこと
- ※3 BTS : Bug Tracking System

表2 プロジェクトプロファイル

No	ファクター	ファクター取捨	マイナス補正(-1)	補正なし(±0)	プラス補正(+1)	備考(選択根拠などを記載)
1	ソフトウェアの規模	使用しない	極めて小さい	普通	大きい	
2	ソフトウェアの複雑さ	使用しない	極めて低い	普通	高い	
3	システム制約条件の厳しさ	使用しない	極めて制約がゆるい	普通	制約が厳しい	
4	仕様の明確化度合い	使用しない	極めて明確	普通	明確になっていない	
5	再利用するソフトウェア/パッケージの品質レベル	使用しない	極めて高品質	普通	品質低い	
6	開発プロセスの整備度合	使用しない	整備できている	普通	整備できていない	
7	開発組織の分業化・階層化の度合い	使用しない	開発組織が単純	普通	開発組織が複雑	
8	開発メンバーの経験とスキル	使用しない	メンバスキルが高い	普通	メンバのスキルが低い	
9	PMの経験とスキル	使用しない	PMのスキルが高い	普通	PMのスキルが高い	
10	案件固有のファクター or システム障害時の自社の損失	使用しない	極めて小さい	普通	大きい	
11	リグレッションテストの量が大いか	使用しない	極めて小さい	普通	大きい	
12	API等により、自由度の高いインターフェイスを提供するか	使用しない	極めて小さい	普通	大きい	
13	稼働プラットフォームが多数あるか	使用しない	極めて小さい	普通	大きい	
14		使用しない	極めて小さい	普通	大きい	
15		使用しない	極めて小さい	普通	大きい	

表3<sup>※2</sup> 自動算出された各指標別目標値

プロセス品質評価指標						
ID	名称	略称	指標値	下限値	～	上限値(指標値±15%)
PR10	仕様レビュー作業充当率	RSRE	3.60 %	3.10 %	～	4.10 %
PR11	設計レビュー作業充当率	RDRE	3.60 %	3.10 %	～	4.10 %
PR12	コードレビュー作業充当率	RCRE	2.60 %	2.20 %	～	3.00 %
PR13	テストレビュー作業充当率	RTRE	2.60 %	2.20 %	～	3.00 %
PR14	テスト作業充当率	RTWE	32.00 %	27.00 %	～	37.00 %
PR15	レビュー作業充当率	RORE	5.60 %	4.80 %	～	6.40 %
PR20	仕様レビュー作業実施率	ERSR	8.16 (人時/KLOC)	6.94	～	9.38 (人時/KLOC)
PR21	設計レビュー作業実施率	ERDR	8.16 (人時/KLOC)	6.94	～	9.38 (人時/KLOC)
PR22	コードレビュー作業実施率	ERCR	4.06 (人時/KLOC)	3.45	～	4.67 (人時/KLOC)
PR23	テストレビュー作業実施率	ERTR	6.80 (人時/KLOC)	5.78	～	7.82 (人時/KLOC)
PR24	テスト作業実施率	ERTW	40.80 (人時/KLOC)	34.70	～	46.90 (人時/KLOC)
PR25	レビュー作業実施率	EROR	27.20 (人時/KLOC)	23.10	～	31.30 (人時/KLOC)

プロダクト品質評価指標						
ID	名称	略称	指標値	下限値	～	上限値(指標値±15%)
PD10	要求仕様書ボリューム率	RSDV	4.60 (Page/KLOC)	3.90	～	5.30 (Page/KLOC)
PD11	設計書ボリューム率	RDDV	13.00 (Page/KLOC)	11.00	～	15.00 (Page/KLOC)
PD12	テスト仕様書ボリューム率	RTDV	13.00 (Page/KLOC)	11.00	～	15.00 (Page/KLOC)
PD30	ファイル行数	FLOC	2.00 (KLOC)		～	2.00 (KLOC)
PD31	関数の行数	MLOC	160.00 (LOC)		～	160.00 (LOC)
PD32	制御文記述率	ROCS	33.00 %	28.00 %	～	38.00 %
PD33	コメント行記述率	ROCL	22.00 %	19.00 %	～	25.00 %
PD34	コーディングルール逸脱率	RDCR	270.00 (箇所/KLOC)	230.00	～	310.00 (箇所/KLOC)
PD40	テスト密度	DOTI	0.35 (項目/KLOC)	0.30	～	0.40 (項目/KLOC)
PD41	不具合収束率	ROFC	0.05 (簡易計算による比率)	0.04	～	0.06 (簡易計算による比率)
PD42	不具合修正率	ROFE	95.00 %	80.80 %	～	100.00 %

としたチケット駆動開発<sup>\*4</sup> (TiDD) が実践されているプロジェクトを対象とすることとした。これはBTSにより、プロジェクト内の各タスクに要した工数実績がチケットに記載され保管されていることから、容易に各種レビュー工数など選定したメトリクスを抽出することが容易であり、現場の負担も少ないこと。また、過去の実績についても調査が可能であることを期待したためである。

## 5 適用案件の事例

### 5.1 案件概要

案件：画像処理モジュールの改修

期間：2012/07 - 2013/03

工数：延べ54人月

### 5.2 前提条件

今回対象としたプロジェクトは、リファクタリング<sup>\*5</sup>を主目的とするエンハンス案件である。なお、当該プロジェクトは、過去のプロジェクトよりテスト資産が引き継がれており、また仮想環境下においてかなりテストの自動化が進んでいる。

### 5.3 選択した指標

表4のように、プロセス品質指標として6つ、プロダクト品質指標として2つを選択した。エンハンス案件であり要件定義フェーズがないことから、ドキュメントのページ数を用いた指標の採用を見送ることとした。

表4 品質評価指標

プロセス品質評価指標		プロダクト品質評価指標	
ID	名称	ID	名称
PR11	設計レビュー作業充当率	PD11	設計書ボリューム率
PR12	コードレビュー作業充当率	PD40	テスト密度
PR15	レビュー作業充当率		
PR21	設計レビュー作業実施率		
PR22	コードレビュー作業実施率		
PR25	レビュー作業実施率		

### 5.4 計測結果

#### 5.4.1 プロセス品質

##### (1) 設計フェーズ

設計フェーズのプロセス品質については、設計レビュー作業における作業充当率及び作業実施率を5カ

分の計測値から算出した結果として、図1、図2が得られた。

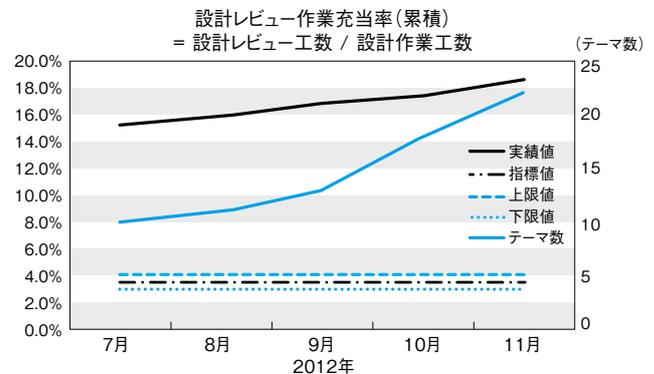


図1 【PR11】設計レビュー作業充当率

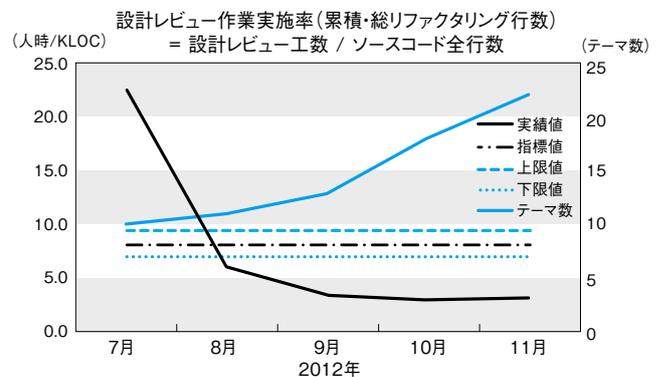


図2 【PR21】設計レビュー作業実施率

##### (2) 実装フェーズ

実装フェーズのプロセス品質については、コードレビュー作業における、作業充当率及び作業実施率を5カ月の計測値から算出した結果として、図3、図4が得られた。

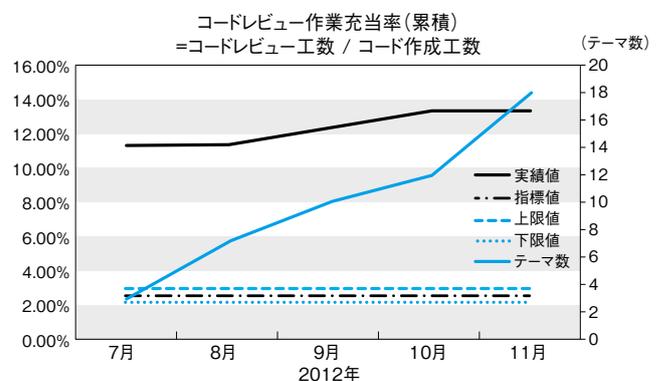


図3 【PR12】コードレビュー作業充当率

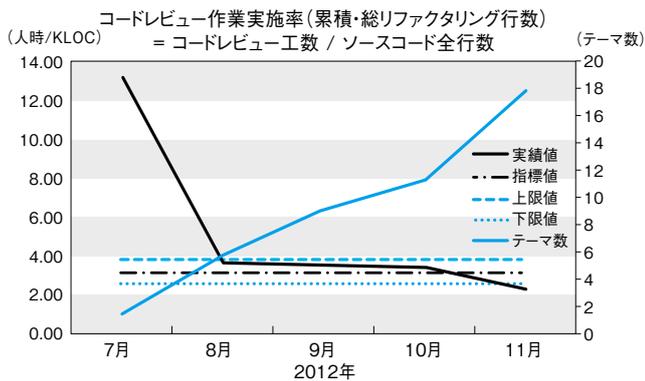


図4 【PR22】コードレビュー作業実施率

### (3) 分析と考察

本プロジェクトは、リファクタリングのテーマ別に設計フェーズ／実装フェーズ／テストフェーズの開発サイクルを短い期間で回していく開発プロセスを採用している。テーマ別に難易度や影響範囲が異なり、月別にフェーズの配分も異なることから、フェーズ毎の累積値としてデータを採取した。

結果として作業充当率ベースで見た場合は、レビューの生産性が悪く、手を掛け過ぎているというデータになっている（図1、図3）。

一方、作業実施率で見た場合は、リファクタリングしたソースコードに対するレビューの不足というデータとなっており、結果が相反する（図2、図4）。

#### ① 充当率に対する考察

エンハンス案件であることから、設計書やプログラムコードは修正が主体となる。このため、初めから成果物を作成するスクラッチ案件と比べ、設計工数や実装工数は少なく済む。一方レビュー実施に当たっては既存部分への影響度を見る分の工数が増えるため、相対的にレビュー工数の比率が高まり、充当率がよく計測される傾向にあるものとする。

#### ② 実施率に対する考察

実施率については、設計変更を伴わない障害修正等の場合、設計レビューは行わないため、設計レビュー実施率は下がる傾向にある。一方、障害修正においても修正した分のコードレビューは行われるため、コードレビューの実施率は下がる。このため、設計フェーズと実装フェーズの実施率に差異が生じているものとする。

今回実施のプロジェクトチームにおいては、直近のコードレビュー実施率が目標レンジを下回る傾向にある。チームに確認したところ、類似の修正が多く発生し、全ての類似修正箇所に対してまではレビューを実施していないことが計測結果に影響していると報告を受けた。しかし、修正箇所については修正漏れ、修正ミスが発生するリスクや既存の機能への影響度が見過されるリスクがあり、レビュー工数が不足している可能性が高い。

#### (4) プロセス品質の改善策の検討

本プロジェクトについては、充当率よりも実施率ベースでプロセス品質を評価した方がよいと判断し、いままで省略していた類似修正箇所に対するレビュー実施率を引き上げる方針とした。なお残念ながら対策実施後の効果測定については、本事例紹介後のタスクとなる。

## 5.4.2 プロダクト品質

### (1) 分析と考察

設計書ボリューム率（図5）については、前述したように設計変更を伴わない修正ソースがカウントされることと、一文でも修正した場合は、設計書修正1ページとカウントされることから、設計書ボリュームもソースコード行数も実態よりも大きい値で計算されていると考えられる。テスト密度（図6）に関しては、仮想環境を用いた自動テストが非常によく整備された案件であるため、目標値をはるかに超える密度でテストを実施して

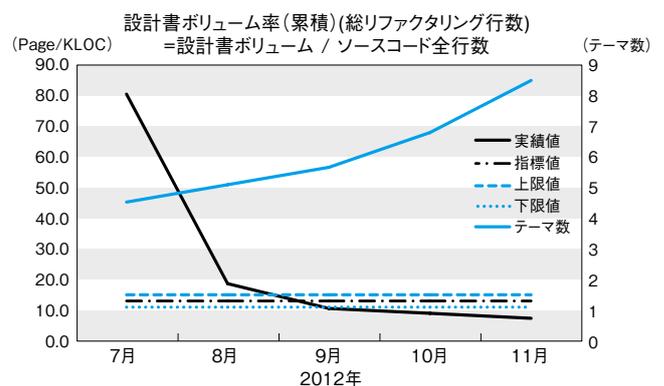


図5 【PD11】設計書ボリューム率

#### 脚注

- ※4 チケット駆動開発：ticket-driven development (TiDD)。Trac, Redmine等のトラッキングシステムを使用してプロジェクトのタスクを管理する手法。[小川 2012]
- ※5 リファクタリング：当案件ではクローンコードの削除、ライブラリの処理精度向上、パフォーマンスチューニング、障害対応を実施。

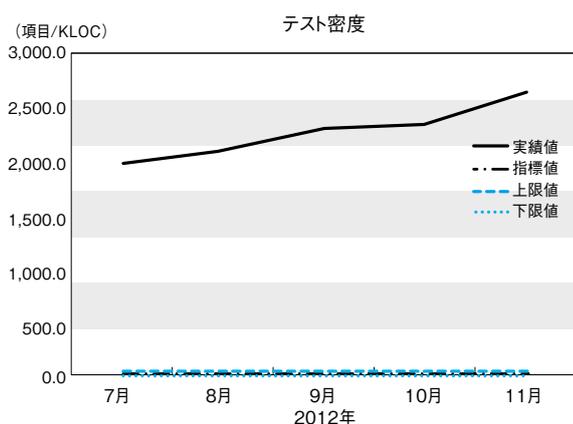


図6 【PD40】テスト密度

いる。また、テストが自動化されているため、工数はほとんど要していない。

## 6 考察と今後の課題

### 6.1 ESQRの導入効果

当初の狙い通り、成果物たるシステムとこれを開発するプロジェクト固有の特性を考慮した上流工程の品質目標を定めることが出来るようになった。これにより、以下のような効果が確認出来た。

- 上流工程の定量的な品質コントロールが出来るようになった。
- 改善の打ち手が、より早いフェーズで打てるようになった。
- 打ち手の効果を目標値に当てて評価出来るようになった。
- 複数の指標を採用することにより、複眼的に品質を評価出来るようになった。

今回のプロセス品質の計測結果のように、複数の指標を用いることにより相反する計測結果が得られるケースもあり得る。この場合、計測データにノイズが混入していないかまず検証し、必要であればデータのクレンジングを行って、プロジェクトの実態により近づけていく作業が必要となる。今回の場合、ソースコードのカウントにおいて、当初レビュー対象外のコードがカウントされていたため、実態とはかけ離れた計測結果となっていた。正しい計測結果を得たにも関わらず、矛盾が認められる場合には、メトリクスを性質をよく理解した上でプロジェクトの特性に照らして十分考察を行い、プロジェクトの実態を表す指標を取捨選択する必要があることが

ある。

しかし、定量化によって目標値との差異が数値化されたことにより、目標値との距離に見合った適切な改善コストを見込めるようになったこと、更には、定量化によって視認性のよいグラフとして計測結果を表現出来るようになったことで、経営層や顧客をレビューアとするプロジェクトレビューにおいて、直感的で説得力のある品質コントロールの成果を示すことが出来るようになったことなど、十分な導入効果を実現することが出来た。

そして、BTSとの相性がよく、適切にTiDDが運用されているプロジェクトにおいてデータ計測を行う場合は、計測コストを掛けずにデータ収集可能であることも実証された。

### 6.2 今後の課題とロードマップ

プロダクト評価指標として提唱されているドキュメントのボリューム率、バランス指標の導入に当たっては、ページ当たりのドキュメント密度を揃える必要がある。具体的には、図表の扱いをどうするのか、ドキュメントの書式を顧客が指定してきたケースではどうするのか、といった課題についてあらかじめ対応を考慮する必要がある。

今後は、これらの課題をクリアしながら、プロダクト品質指標の計測を充実させ、継続的に計測データの蓄積と改善活動を行っていく計画である。将来的には、SECが定義する目標値をそのまま採用し続けるのではなく、計測データに基づいて定期的に目標値の見直しを行うことが必要である。また、その結果をSECにフィードバックすることでESQRの改善のために貢献したいと考えている。

## 7 謝辞

本タスクフォースの実行に当たっては、多くの方々にご協力をいただいた。この場を借りて御礼申し上げる。とくに、タスクメンバとして、またトライアルプロジェクトの中心メンバとしてタスクフォース推進に多大な貢献をしてくださった賀来 茜さんに感謝申し上げます。

#### 参考文献

- [IPA2011] IPA/SEC：続 定量的品質予測のススメ，佐伯印刷，2011
- [IPA2012] IPA/SEC：[改訂版] 組込みソフトウェア開発向け品質作り込みガイド，2012
- [小川2012] 小川明彦・阪井誠：チケット駆動開発，翔泳社，2012