

類似製品の並行派生開発における共通仕様の検出と共有方法

Methodology of Detecting and Sharing Common Specifications to Stake

holders in Developing Concurrent Derivative Similar Systems

主査 : 清水 吉男 (株式会社システムクリエイツ)
副主査 : 飯泉 紀子 (株式会社日立ハイテクノロジーズ)
副主査 : 足立 久美 (株式会社デンソー)
リーダー : 内藤 史郎 (I s t ネクスピア株式会社)
研究員 : 星野 充史 (アンリツエンジニアリング株式会社)

研究概要

同じ開発母体による類似製品の並行派生開発において、共通仕様の漏れや誤った解釈は、その影響が複数製品に及ぶ可能性が高いため、品質リスクが大きい。

この問題に対し、専門組織を設置し共通仕様を取りまとめて管理する解決方法があるが、小規模な派生開発では開発コストに厳しい制限があり、適用が困難であった。

そこで筆者らは、過去の方法論でありあまり対象としていない、小規模な並行開発において専門組織の代替となる方法により共通仕様の検出と、これを並行開発される製品の関係者で共有する R²SPL という仕組みを考案した。

これにより、共通仕様の一元化ができ、仕様を正確に伝達できるようになる。この結果、共通仕様の漏れや解釈ミスによるソフトウェア不具合の半数以上を回避できた。

Abstract

In modification type software developments which develop concurrent multi-systems that have same existing code, there is high possibility that leaking or misunderstanding common specification affect other systems. So it has severe risks.

In previous research, there is a method of establishing expert division that organizes common specifications which are provided by consumers of each systems to manage concurrent multi-system development and to mitigate the risks, but in small-sized concurrent multi-systems development, there is severe cost restriction, so we could not adopt the method.

So the authors have devised a new methodology of which common specification can be detected and sharing them between stake holders and it can apply to small-sized modification type of concurrent multi-systems development instead of establishing expert division. And we named it R²SPL.

As a result, common specification can be centralized and specification can be transferred correctly. So half and higher software anomalies caused by leaking and misunderstanding of common specification have been mitigated.

1. はじめに

急速な市場の変動や IT 技術の進展にともない、顧客の経営環境も刻々と変化してきた。その結果、競合他社にない優位性を持つ機能や性能をいち早く実装し市場投入することが求められるようになり、組織が生き残るための競争がいつそう激しさを増してきている。この潮流の中で、競合他社に無い優位性を実現するために求められる顧客からの要求も高度化し、開発規模は増加傾向にあるが、開発期間は短縮される傾向にある。また、この限られた期間内に、品質も保証しなければならない。

この熾烈な生存競争に勝ち残るために、製品開発スピード向上を目的として類似機能を同じ母体で開発でき、生成される製品は複数の製品機能を含み、環境変数を変えると特定製品として動作する統合開発環境が構築されることがある。製品ごとに開発環境が独立し、類似機能であっても個々に開発されたり、その他の製品から移植し最適化したりするのが非効率なためである。筆者らの組織においても、このような統合開発を導入した¹。従来との開発環境の違いを図1に示す。

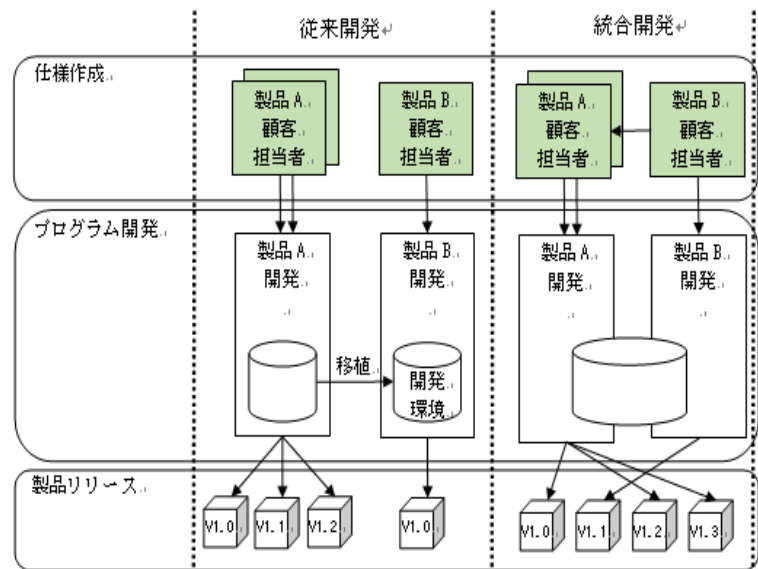


図1 従来開発と統合開発の開発環境の違い

このような状況において、ある製品の変更がその他の製品へ意図しない影響を与える新たな問題を引き起こすことがある。同じ母体の複数製品に対し、相互への影響を適切に把握しないまま変更を加えることがあるからである。これらの問題は、仕様提供者と獲得者間が多対多の複雑な構図で非同期・不定期に仕様提供が行われることと、製品開発担当者における、自身の担当製品以外の知識の欠如により共通仕様²に気付きにくいこと、顧客から提示された仕様の認識が関係者間で等しく共有できる工夫の欠如に起因することが多い。

そこで、共通仕様を一元管理でき、製品知識の欠如を補完でき、関係者間で等しい仕様の認識を共有でき、さらにはその成果を製品開発計画へフィードバックをも可能にする仕組み、R²SPL(Requirement Repository for Software Product Line)を考案した。これを適用することにより、共通仕様漏れや共通仕様伝達ミスによる品質リスクをおよそ半数にすることができた。本論では、R²SPL適用結果について述べる。

2. 複雑な関係者の構図と統合開発環境の実際

2.1 統合開発環境における開発背景

本論で対象とする開発（以下、本開発という）は、以下の特徴を持っている。

- (1) 各製品の顧客担当者から各製品に該当する開発チームへ変更依頼が提示されている
- (2) 顧客担当者は筐体制作担当者、電気制作担当者、ソフトウェア設計担当者に分かれており、開発チームも要求獲得担当者が複数存在することが多い
- (3) 通常、ソフトウェアの変更依頼はソフトウェア設計担当者から提示される
- (4) 新規のハードウェアが追加された場合は電気制作担当者から提示される場合もある

このような仕様提供者と獲得者間が多対多の複雑な構図（付図1参照）においても、製品ごとに開発環境が独立し、それぞれの開発チームが顧客から提示された仕様に対して開発していれば、変更による直接的な影響を他製品に与えない。しかしながら、統合開発環境で従来同様に仕様の受け入れ確認を各製品で行う場合、ある製品に対する顧客からの仕様が、他製品に与える影響を十分に把握しないまま実装され、変更による影響を与えることがある。また、顧客から提示された仕様が他製品へ影響することに気付いても、さまざまな方法で関係者へ伝達されることで、情報が正しく伝わらないこともある。

¹ 一元管理されたプログラムコードを含むプログラム開発環境を意味する

² 共通仕様は、複数の類似製品に存在する同一の機能を意味する。ただし、製品のハードウェア構成や扱うデータの違いから詳細仕様が異なる場合がある。

これらのことから、統合開発環境での開発時には以下が必要であるといえる。

- (1) 他の製品に与える仕様を検知する
- (2) その情報を正確に共有する

2.2 本開発における不具合事例の傾向

2.1 で述べた統合開発の問題と不具合事例との関係性を把握するため、本開発における総合テスト以降に発見された不具合事例 318 件について、不具合混入工程・仕様の分類・不具合要因別の順にふるいに掛けながら分析した。結果を表 1 に示す。

表 1 統合開発における総合テスト・評価で発見された不具合の要因別内訳

総合テスト・評価における不具合件数	不具合混入工程	仕様の分類	不具合要因別
318 件	設計以降： 224 件	--	--
	仕様受入： 94 件	製品個別仕様： 28 件	--
		共通仕様： 66 件	仕様記述漏れ ³ ： 37 件 仕様不明瞭ミス ⁴ ： 29 件

不具合が混入した開発工程を分析した結果、仕様受け入れ確認時に抽出できなかったものが 94 件あった。仕様受け入れ確認は、顧客担当者から提示された変更依頼に対し、仕様漏れの抽出や機能レベルでの変更および追加による影響範囲を特定する作業である。この 94 件を、従来開発と同じ製品個別仕様⁵によるものと、統合開発へ移行して複数製品に影響を及ぼした共通仕様に分類し、さらにこの共通仕様について不具合要因別に分類した。従来には無かった共通仕様起因の不具合が見られる一方で、設計以降で混入した不具合は、仕様解釈ミス、論理設計ミス、設計書の構成管理の問題によるインタフェースミス、プログラム言語スキル不足によるプログラミングミスなどの従来開発と類似する内容が多かった。つまり、統合開発に移行したことで、従来の仕様の受け入れ確認では共通仕様の検出に対応しきれなくなったといえる。

そこで、問題の主要因を把握するため、仕様の受け入れ確認で混入した不具合の内、統合開発の不具合 66 件に着目し、その原因の傾向を分類するとともに、何の情報不足が不足して発生したのかを整理し、その結果を表 2 にまとめた。

表 2 仕様受け入れ確認で抽出できなかった不具合の傾向

ID	不具合原因の傾向	原因分類	不足した情報
#1	変更依頼に共通仕様の記述が漏れているため、共通仕様に気付かない場合。	仕様記述漏れ	・共通仕様の記述
#2	開発時期の違いで共通仕様に気付かない場合。	仕様記述漏れ	・共通仕様の記述
#3	変更依頼の表現方法の問題による仕様の解釈ミス。	仕様不明瞭ミス	・関係者が分かり合える表現で記述した共通仕様
#4	他製品開発へ伝達する際の共通仕様の表現方法の問題による仕様の解釈ミス。	仕様不明瞭ミス	・関係者が分かり合える表現で記述した共通仕様

³ 仕様記述漏れは、複数製品で共通の機能自体が漏れる場合と、機能に含まれる一部の仕様が漏れる場合、および性能要件が漏れる場合を意味する。

⁴ 仕様不明瞭ミスは、仕様の文章表現の問題により開発者が解釈を誤る場合を意味する。

⁵ 製品個別仕様起因の不具合は、当該製品にのみ存在する機能の変更により発生したものを意味する。製品個別仕様の不具合が他製品へ影響した不具合も含む。

表 2 に示す不具合原因の具体例を付表 1 に示す。共通仕様の仕様漏れは変更依頼に記述されないことに起因するが、並行開発する複数製品において共有仕様であることにまったく気付けない場合と、一部の製品開発で気付いているが他製品と情報共有ができないため気付けない場合の 2 つ存在があった。特に前者の場合は、仕様の受け入れ確認にて、共通仕様か、または個別仕様かの 2 択で分析するため、共通仕様に気付かなければ個別仕様と判断されやすい。つまり、並行開発する製品と“共通仕様かもしれない”、または後発の他製品開発が立ち上がったときに“共通仕様になるかもしれない”という状態を初期判断時に残していないため、共通仕様の漏れが起きやすいといえる。また仕様不明瞭ミスは、顧客担当者と製品開発チームの間または各製品開発チームの間において、お互いに分かり合える仕様の表現になっていないために発生していた。

2.3 類似製品の並行開発における先行研究での提案手法

このような類似製品における並行開発に関係する手法には次の 2 つがある。

まず、専門組織を構築して要件定義・開発・現地試験まで複数の開発チームをマネジメントし、仕様やプログラムコードの一元管理を実現する手法である^[1]。この手法の効果的なプロジェクトの特徴は、「類似製品の並行開発が 2, 3 以上で、全体開発期間が 1 製品分の開発期間より若干長い程度」と論じている。この研究は、最大 10 のプロジェクトが、それぞれ 400~500 人月の類似製品を新規に並行開発する規模である。これに対し、本開発は最大 4 のプロジェクトが、3~40 人月の類似製品を並行して派生開発する規模で、全体の開発期間が 1 製品分の開発期間より大幅に長い場合が多い。また、同様の手法で専門組織を構築するには、製品知識を有する組織を新たに育成するか、または開発チームの熟練者を引き抜く必要があり、現組織において新しい組織や開発チームを柔軟に構築／解散することはコスト上難しい。このため、この手法のそのままの適用が難しいと判断した。

次に、変更依頼に対する関連システムへの影響を事前に検知する手法である^[2]。これには、顧客から当該システムに対する変更依頼の背景に、他システムの変更依頼が含まれていることに気付かない問題に対し、専門組織を構築せずに解決する手法が論じられている。この手法は相互に通信するシステムにおけるインタフェースの相互影響を検知する手法であり、独立したシステムにおける共通資産の相互影響を検知できない。このため、本開発へそのまま適用するのが難しいと判断した。しかしながら、専門組織の代替となる手法で顧客から提示された仕様の影響範囲を検知している点は利用可能であると考えた。

3. 制約条件の中で共通仕様を展開する方法論

2.1 で述べたように、共通の母体で類似製品を並行開発する場合、他製品に与える影響を検知することと、その情報を共有することが必要である。そして、2.3 で述べた制約条件の範囲内である必要もある。そこで、専門組織の代替となる新しい方法を考案した。この方法は、共通仕様に気付きやすくする仕組みとそれを共通の伝達手段で展開する仕組みから構成される（付図 3 参照）。これらの仕組みを解説する。

3.1 共通仕様に気付きやすくする仕組み

3.1.1 要求仕様の一元管理

共通仕様を検出するには、まず各製品の顧客担当者と開発チームにおける多対多の複雑な構図を一元化し、単純にする必要がある。多対多の複雑な構図で非同期に情報通信が行われると、その管理も複雑になり他の製品の適切な情報を得ることが難しくなるからである。先に述べたように、専門組織によって共通仕様の一元管理は行えるが、コスト上の制限で適用できない。そこで、これと同等の機能を持たせるため、専門組織の代わりに「要求リポジトリ」という仕組みを考案した。要求リポジトリは、不定期で非同期にコミット

されるソースコードを一元管理するよくあるリポジトリと同様に、複数製品を跨いだ共通仕様を含む変更依頼をファイルとして一元化する。また、要求リポジトリは、最初に開発を始めた開発チームが作成し、変更依頼の度に更新する。そして、途中で他製品の開発がされた際も同じファイルを共有し、すべての製品開発が完了するまで使用する。これにより、変更依頼の一元化は行えるようになると考えた。

3.1.2 共通仕様と個別仕様、共通仕様の可能性がある仕様の分類

各製品開発担当者が要求リポジトリを参照した際、記載された仕様が共通仕様であるか個別仕様であるかの判別を容易にするには、それぞれの仕様に識別子があるとよい。ただし、要求リポジトリへ仕様記述を行う担当者は、必ずしも他製品の仕様に精通しているとは限らないため、記述しようとしている仕様が個別仕様であることに確証を持っていない場合もある。このような場合に誤って個別仕様と判断しないために、共通仕様と個別仕様の他に共通仕様の可能性がある仕様を識別できる必要があると考えた。そこで、要求リポジトリへの仕様記述時に以下に該当する一意の識別子を付与することをルールとした。

(1) 共通仕様であることが明らかであるもの

(2) 「(1)」以外の個別仕様であることが明らかではないもの（以下、保留仕様という）

これにより、各製品開発担当者は要求リポジトリを参照することで、担当製品における実装工程前に共通仕様と保留仕様を識別できると考えた。

3.1.3 保留仕様から共通仕様を抽出

保留仕様には共通仕様を含むことがある。これを特定しないまま実装されると、他の製品へ意図しない影響が出る可能性がある。したがって、実装工程が開始される前に保留仕様は共通仕様と個別仕様に確実に分類されている必要がある。各製品担当者は他の製品の仕様に精通しているとは限らないため、この分類には他製品仕様の熟練者によるレビューを要する。そこで、要求リポジトリへの仕様記述後、保留仕様を共通仕様と個別仕様に分離するため、各製品の熟練者による保留仕様のレビュープロセスを導入した。これにより、各製品開発担当者が実装開始前に確実に共通仕様を識別できると考えた。

3.2 共通仕様を共通の伝達手段で展開する仕組み

前項の方法により共通仕様の検出が行われても、関係者に正しく伝達されなければ正しく実装されず、後工程で不具合を来す可能性がある。仕様がさまざまな文書⁶で任意の書式によって記載されていると、更なるそのリスクは高まる。したがって、仕様とそれに該当する要求の関連性が一別できる標準文書に記載するべきである^[3]。また、本開発は短納期の制約がある派生開発である。清水^[4]は、「派生開発の最大の制約は『部分理解』のもとで作業しなければならないこと」と述べている。このため、この担当者の思い込みや勘違いを始めとする部分理解の罠に陥らないような標準文書であることが望ましい。更に、それを作成するためのコストや手順が少ない方がよい。そこで、これらの特徴に有効とされる USDM (Universal Specification Describing Manner) を採用し、要求リポジトリのデータを入力としてこれを自動生成するツールを開発することにした。これにより、要求と仕様の表現が鮮明になり、仕様表現の均質化ができると考えた。また、USDM を自動生成し関係者がレビューすることで、担当者の負担の軽減と部分理解の罠を回避できると考えた。

筆者らは、この一連の仕組みにより、要求リポジトリに蓄積させた共通仕様を、SPL (Software Product Line) 開発^[5]におけるコア資産⁷の導出を可能にすると考え、R²SPL と命名した。R²SPL の俯瞰図を付図 4 に示す。

⁶ Word, Excel, PowerPoint, 電子メールなどの文書を意味する。

⁷ 過去の開発経験などを通じて蓄積され、再利用のために整備されたソフトウェア資産。

4. R²SPL の検証

4.1 検証方法

本研究で考案した手法により、類似製品の並行開発における共通仕様の検出と共有方法について、その効果を検証した。なお、検証は過去に実施した派生開発へ適用したものである。検証範囲やその方法について詳細を以下で説明する。

検証範囲は、過去の3つの製品開発における不具合事例について、総合テストとその後の品質保証部門における評価で発見された不具合の内、仕様の受け入れ確認で摘出できなかった共通仕様を対象とした。また検証方法は、被験者による検証結果のばらつきを抑えるため研究員の他に製品開発の熟練者と、比較的開発経験の浅い担当者で実施した。この体制を表3に示す。仮想的な並行開発環境⁸を実現するために、これら3つの製品開発をそれぞれの被験者が並行作業するようにした。なお、本検証の効果を客観的に判断するため、研究員は2.2で述べた事例を選択した。検証手順は次の通りである。

- (1) 変更依頼を共通仕様と保留仕様に分類して要求リポジトリへ登録する。
- (2) 要求リポジトリを各製品開発の熟練者らでレビューし共通仕様を共有する。この際、保留仕様がある場合、共通仕様または個別仕様に識別する。
- (3) 要求リポジトリからツールにより中間ファイルを出力し、各製品開発へ伝達する。
- (4) USDMに中間ファイルを読み込んで、共通仕様を反映する。
- (5) 変更依頼が非同期に提示される想定とするため、研究員の製品開発は2回に分けて上記(1)～(4)の手順を繰り返す。
- (6) 上記手順において、USDMに記載された要求と仕様から、過去の不具合事例が解決したかどうかを確認する。

表3 被験者とシミュレーションする製品開発の定義

被験者	製品開発経験年数	シミュレーション時の製品担当
研究員	3年	担当製品：製品A
協力者（熟練者）	15年以上	担当製品：製品B（Ver.1.0）
協力者（担当者）	2年未満	担当製品：製品B（Ver.1.1）

4.2 検証結果

それぞれの被験者に対する検証の結果、仕様の受け入れ確認で摘出できなかった共通仕様起因する不具合の約40%～68%を摘出できた。この結果を表4に示す。

表4 共通仕様の摘出件数と不具合要因別の摘出率

被験者	検証対象製品	内訳 (不具合要因)	不具合事例	手法適用	
			件数	摘出件数	摘出率
研究員	製品A	合計	66件	45件	68.1%
		(仕様記述漏れ)	(43件)	(34件)	(79.0%)
		(仕様不明瞭ミス)	(23件)	(11件)	(47.8%)
協力者 (熟練者)	製品B	合計	11件	5件	45.4%
		(仕様記述漏れ)	(8件)	(4件)	(50.0%)
		(仕様不明瞭ミス)	(3件)	(1件)	(33.3%)
協力者 (担当者)	製品B	合計	30件	12件	40.0%
		(仕様記述漏れ)	(10件)	(6件)	(60.0%)
		(仕様不明瞭ミス)	(20件)	(6件)	(30.0%)

⁸ 協力者の検証範囲は、過去の製品B開発の一部機能の不具合事例に限定し実施した。

表 4 の不具合事例における件数は、4.1 で述べたように仕様の受け入れ確認で抽出できなかった共通仕様に起因する不具合を、製品開発ごとに示したものである。その不具合要因の内訳を括弧内に示し、手法適用による抽出率を要因別に示した。この要因別の抽出率より、仕様記述漏れは約 50%~79%、仕様不明瞭ミスは約 30%~47%の範囲で抽出できることがわかった。また、各製品の抽出件数の合計 62 件について、本手法の 2 つの仕組みによる抽出件数の内訳を表 5 に示す。

表 5 本手法の 2 つの仕組みによる抽出件数内訳

本手法の仕組み	仕組みの構成要素による効果	抽出件数
共通仕様に気づきやすくする仕組み	要求リポジトリによる効果	47 件
	保留仕様を許容した効果	6 件
共通の伝達手段で展開する仕組み	USDM の表現で展開した効果	9 件

また、検証作業の協力者から本手法の効果について以下の意見が得られた。

- (1) 並行開発される他の製品仕様との関係に気付かされる場を得られた。
- (2) 共通仕様と製品固有仕様を把握できるため、それぞれ分離して変更設計できた。
- (3) 保留仕様の表現は仕様受け入れ確認時の仕様漏れの不安を低減できた。
- (4) 要求リポジトリから USDM を自動生成する仕組みにより仕様の解釈ミスが低減できた。
- (5) USDM の自動生成は過負荷時における人為的ミスを低減できる。

4.3 考察

表 4 が示すように、本手法の 2 つの仕組みは仕様記述漏れに起因する不具合に効果があるといえる。また、表 5 が示すように、特に共通仕様に気づきやすくする仕組みに効果があるといえる。さらに、共通の伝達手段で展開する仕組みについても一定の効果があるといえる。しかしながら、仕様不明瞭ミスに起因する不具合は、仕様記述漏れに起因する不具合より抽出しにくい傾向にある。この背景には以下の問題があると考えられる。

- (1) 仕様提示者の曖昧な表現により仕様提示者の意図を仕様獲得者が誤認する場合があります。この時点で正確な仕様が獲得されなければ、後工程で正確に伝達されたとしても仕様提供者の期待通り実装できない
- (2) 提示された仕様の根拠や背景が明確になることで正確な仕様が伝達されやすくなるが、仕様記述は自然言語であるため完全に共通の認識が得られない場合がある

ただし、旧シリーズ製品から存在する機能については、熟練者が仕様を旧シリーズ製品の視点で比較するため、曖昧な仕様表現であっても共通仕様か個別仕様かを判断できる場合がある。また、仕様受け入れ確認時に保留仕様の状態を許容した結果、担当者は共通仕様または個別仕様かの判断をレビュープロセスまで遅延させて、関係者が共通仕様であることに気付ける場合もある。

このように、R²SPL は仕様を要求リポジトリで獲得した仕様を一元化し、関係者が共通仕様に気付く場を与えることに特に効果があり、獲得した共通仕様を正確に伝達することにも効果がある手法といえる。

5. まとめ

5.1 組織の弱みを認識し強みに活かす仕組みが達成したこと

昨今のソフトウェア開発は、開発効率を上げるためにさまざまな取り組みがされている。その取り組みのひとつに、統合開発環境での派生開発を述べた。納期優先の派生開発は、対象領域や既存ソフトウェアの知識が必須であるにも関わらず、部分理解で開発され決められた品質も達成しなければならない。この制約の中で、品質リスクが高い共通のソフトウェア部品に影響を及ぼす共通仕様を正しく認識するには、仕様提示者と仕様獲得者の複

雑な構図による影響を排除する仕様受け入れ工程の一元化と、各製品の熟練者の知識を集結させる必要がある。また、これを実装するために仕様を正確に伝達する必要がある。

その手段として、要求リポジトリによる共通仕様と保留仕様の一元化と USDM による情報伝達を紹介した。要求リポジトリは、仕様提示者と仕様獲得者の多対多の構図の中でも共通仕様と保留仕様を一元化してくれる。また、一時的に保留仕様を許容し、チームでレビューすることで、一人の担当者では気付けなかった共通仕様に気付く機会を与えてくれる。次いで、USDM 自動生成による情報伝達方法について説明した。この方法は、担当者の思い込みや勘違いによる情報伝達ミスの低減に必ず効果が出る。自動生成による情報伝達のため、作業ミスが発生することもない。

飯泉^[6]は、ソフトウェアの品質を作り込むには、自組織の強み・弱みを設計者自身が客観的に把握することが重要だという。一つの製品の設計者では共通仕様の判別が行えない弱みを設計者自身が客観的に把握した上で保留仕様の判断を許容し、他製品の設計者が支え合う強みを持つ組織を作ること、高品質であるという付加価値を持つ競争力のある製品を世に出すことができ、組織の持続力に繋がる。

5.2 品質改善の枠組みを超える R²SPL の展望

統合開発環境を導入することで発生した課題について、2 つの仕組みを組み合わせた手法を考案し品質問題を改善できた。この手法を継続することによって要求リポジトリに共通仕様と個別仕様が蓄積され、これらの仕様と製品の機能との紐付けを行うことができれば、共通のソフトウェア部品の数量や規模が導き出せる。

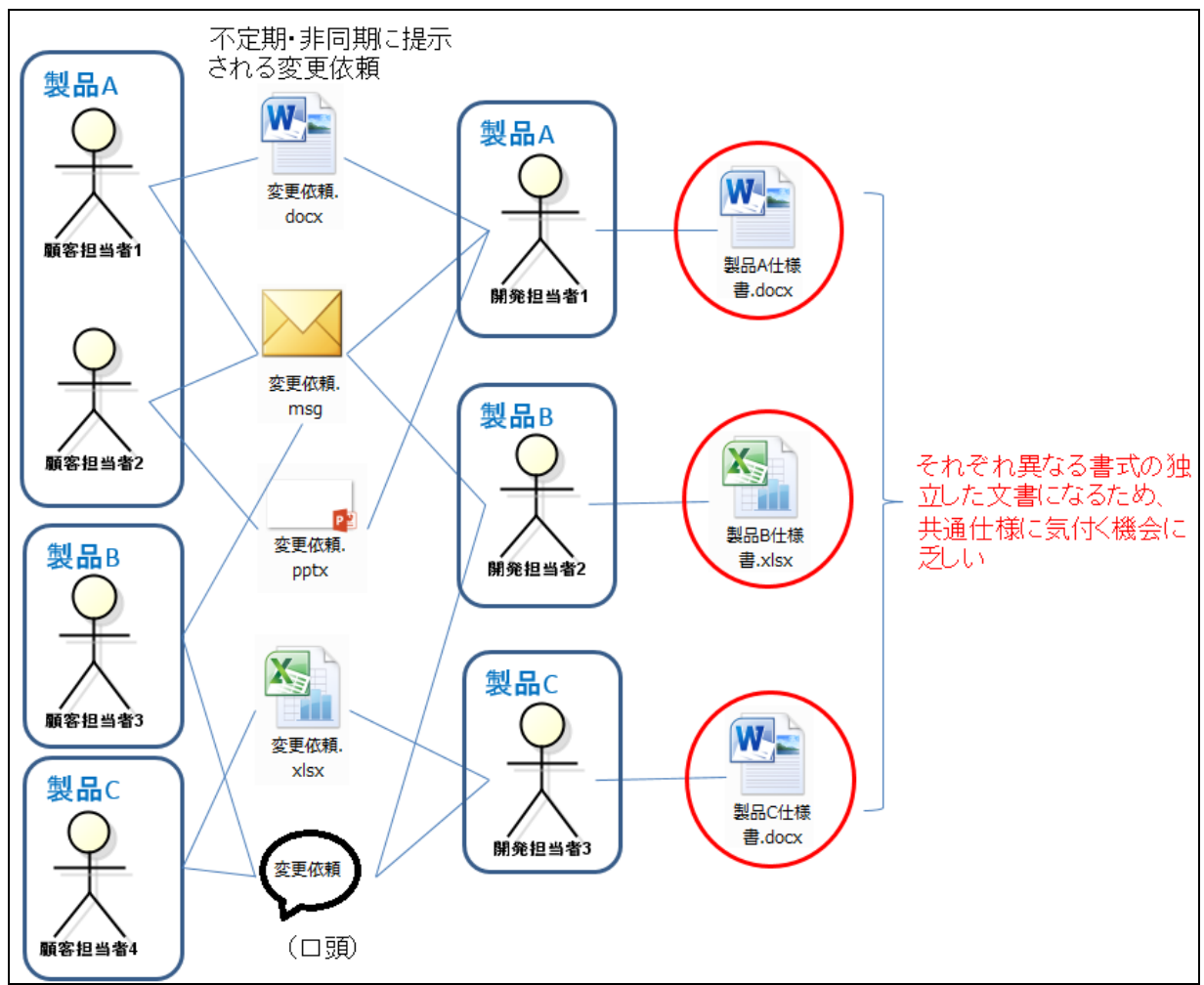
本論で述べた統合開発環境は、開発効率を上げるために導入された。その背景にはコストと納期を圧縮したいという組織からの暗黙の要請がある。共通仕様と製品の機能との紐付けを行えるよう本手法を拡張することで、SPL 開発における要求差分の実現工数を見積もる上での入力情報と成り得る。そして、類似製品におけるコア資産の導出とコア資産からの製品用資産の導出を行い、この資産を再利用する製品サイクルによって、生産性が向上されるであろう。

このように、R²SPL はソフトウェア品質改善の枠組みを超え、予算、規模、計画を始めとする価値のある情報を開発ライフサイクル全体に提供するという発展性を秘めている。そしてこの活動が、組織からの暗黙なコストと納期の要請に応えるための有効な手段になることを期待する。

参考文献

- [1]尾崎信之, 石川隆, 類似システムの並行開発に適したマネジメント手法の提案, Journal of the Society of Project Management Vol.8, No.1, 2006
- [2]変更依頼の対応箇所を検討する前に他システムへの影響を検知する方法, ソフトウェア品質管理研究会 2011 年度 分科会成果報告, 日本科学技術連盟
- [3]矢口竜太郎, 開発ドキュメント やってはいけない読み手を悩ます文書, 日経 SYSTEMS, 日経 BP 社, 227, p21-39, 2012
- [4]清水吉男, 「派生開発」を成功させるプロセス改善の技術と極意, 技術評論社, 初版 第 4 刷発行, 2012
- [5]Paul, K., Böckle, G. and van der Linden, F., Software Product Line Engineering, Springer-Verlag (2005) (林 好一, 吉村健太郎, 今関剛訳, ソフトウェアプロダクトラインエンジニアリング, エスアイビー・アクセス (2009)).
- [6]飯泉紀子, 設計者自身によるソフトウェア品質作り込みのアプローチ, 品質, 日本品質管理学会, 42 巻, 4 号, p478-485, 2012

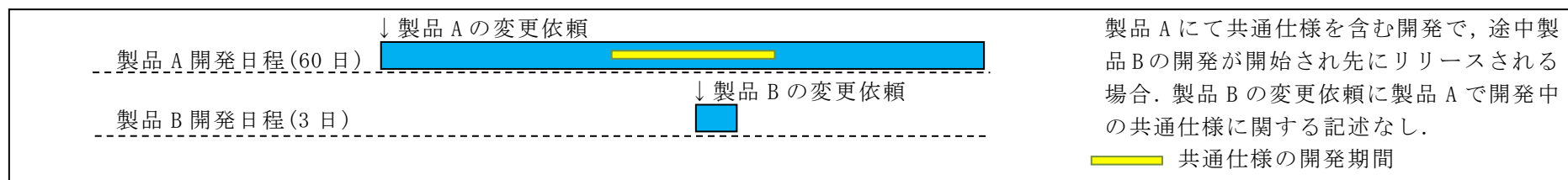
付録



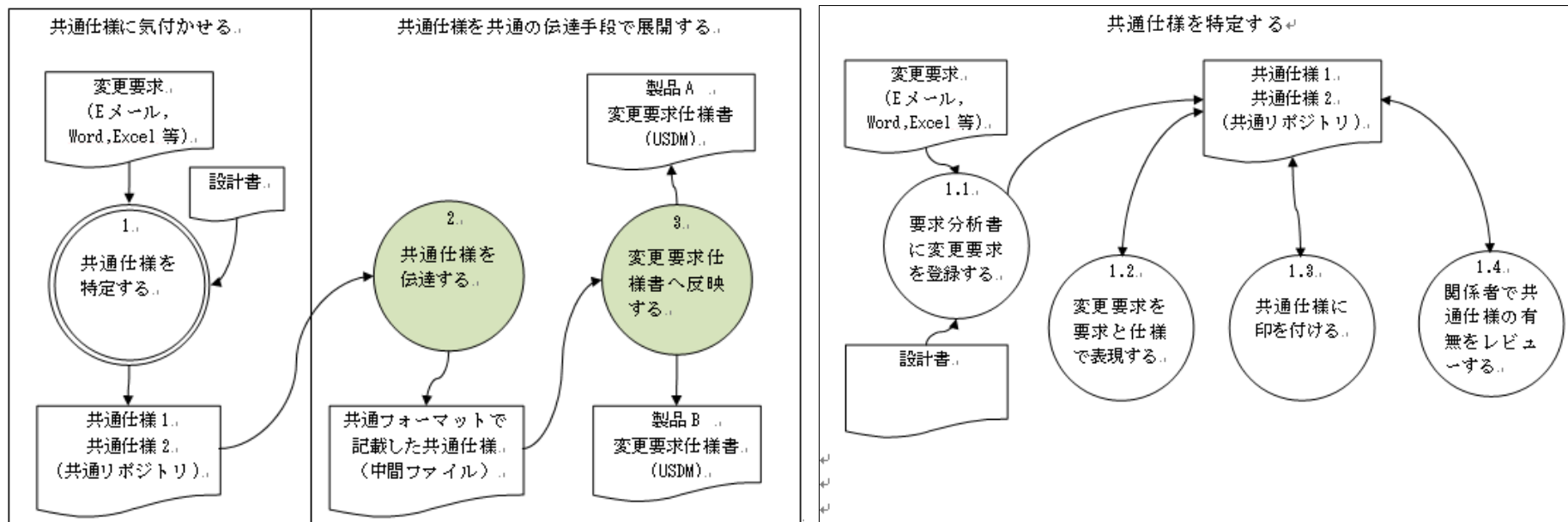
付図 1 不定期・非同期・さまざまな方法で提示される変更依頼

付表 1 仕様受け入れ確認で抽出できなかった不具合の具体例

ID	不具合原因の傾向
#1	製品 A の変更依頼に対応した際に共通仕様であることに気付かず、製品 A の個別仕様として開発した。その結果、製品 B は正しく動作しなかった。
#2	製品 A にて共通仕様を開発中に製品 B が立ち上がった。製品 B の変更依頼には製品 A で開発中の共通仕様記述がない。その結果、製品 A より先にリリースした際、開発中の共通仕様の処理が製品 B に影響を与えた。(付図 2 参照)
#3	変更依頼に記載されていた共通仕様について、各製品の詳細仕様の差異がどの製品を指しているか曖昧な表現だったので、開発担当者が解釈を誤った。
#4	開発計画の異なる複数製品に共通の新規機能を追加した際、先発リリース製品で実装した仕様を後発リリース製品の関係者へ意図した要求内容が伝わらなかったため、後発リリース製品にて仕様の解釈を誤った。



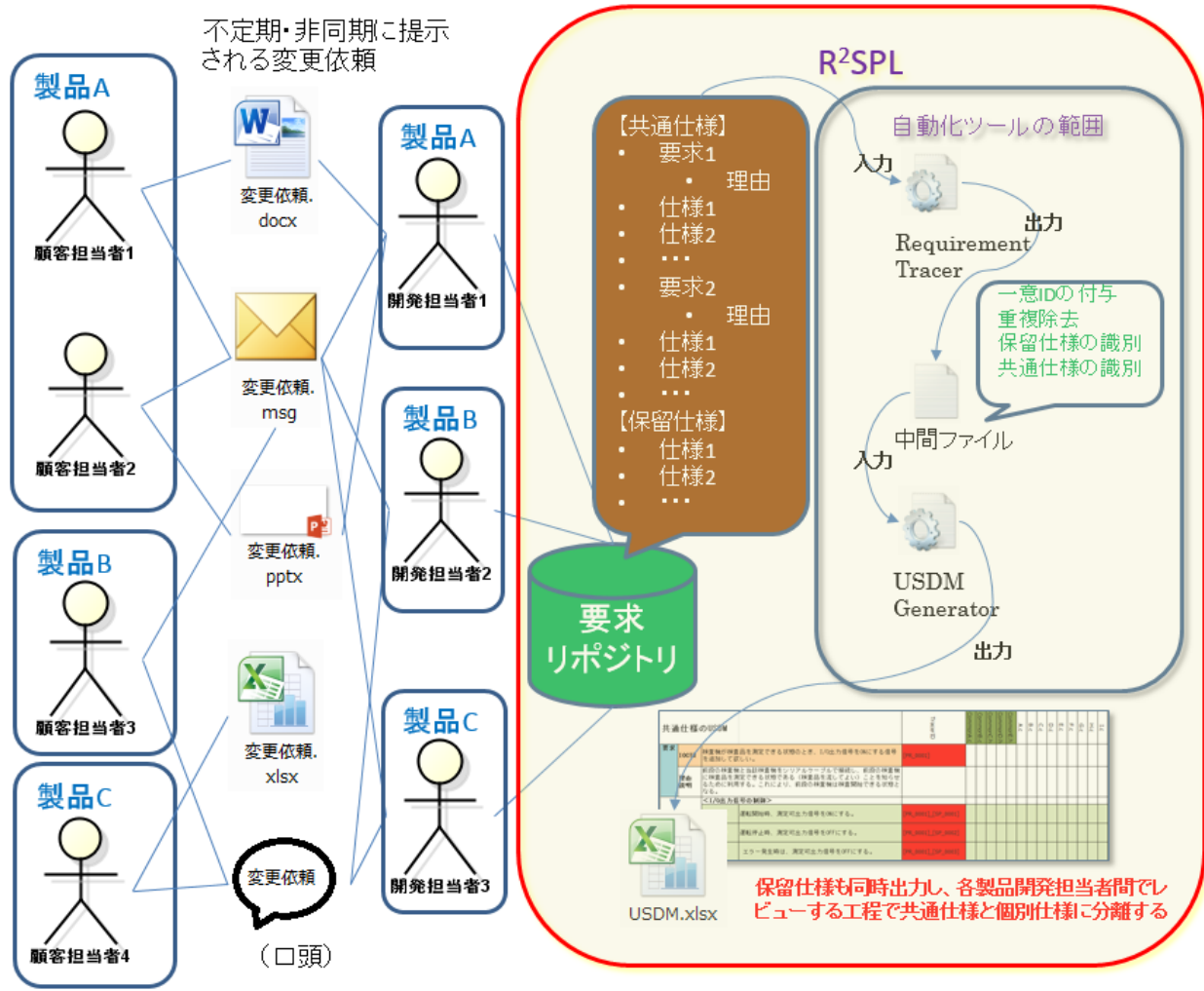
付図 2 並行開発における変更依頼の提示時期の違いにより、製品 A の共通仕様開発が製品 B に影響する例



付図 3 2つの仕組みによる課題解決の PFD

【付図 3 の補足】

網掛けしたプロセス 2 と 3 は、本研究で作成したツールによって作業を自動化するプロセスを示す。「2 共通仕様を伝達する」プロセスは要求リポジトリに埋め込んだマクロを実行し、「3 変更要求仕様書へ反映する」プロセスは USDM の Excel ファイルに埋め込んだマクロを実行する。



付図 4 R²SPL の俯瞰図

2.2 で述べた不具合事例 318 件の内、仕様の受け入れ確認で気づけなかった共通仕様に関する不具合事例 66 件の抜粋を下付表に示す。

付表 2 仕様の受け入れ確認で気づけなかった共通仕様に関する不具合事例（抜粋）

不具合分類	不具合事例	原因	要求 or 仕様	不具合の影響度
仕様漏れ	装置のパラメータを USB へ外部出力した際、現製品に関係ないパラメータも出力される	複数製品で共通の仕様であるが、製品により USB へ出力するデータに違いがあった。しかし、仕様には明記されていなかったため、すべてのパラメータを出力していた。	仕様	中
仕様漏れ	被検査品に関するパラメータを設定中、その被検査品が正しい状態で設定されていないエラーが表示されてしまう。被検査品に関するパラメータ設定中はこのエラーを表示しないこと。	複数製品で共通の仕様であるが、一部の製品についてはエラー表示条件が異なるため、このエラーが発生していた。仕様には明記されていなかったため、共通仕様として実装してしまった。	仕様	小
仕様漏れ	パラメーター一覧リストに表示するパラメータのうち、現製品で使用しないパラメータや機能を示す文字列が表示されてしまう。	パラメーター一覧リストの表示は共通仕様として実装されているが、どの製品にパラメータが依存しているかどうか仕様には記載されていない。	仕様	中
仕様漏れ	ある検査モードのとき、製品を検査する画面に遷移できない（製品 A では遷移可能だが、製品 B は遷移できない）	製品検査画面は共通仕様として実装されているが、製品によって表示条件が異なる。製品 B では表示条件が標準よりも少ないが、標準の範囲で判断していたため、仕様には記載されていない。	仕様	小
仕様漏れ	製品共通パラメータである〇〇パラメータの名称を製品 C の時だけ表示名称を変更したい	製品の下位互換を保つために同じ意味のパラメータでも表示名称を製品限定で変更したいという要求であるが、仕様には記載されていないため総合テスト工程で発見された。	仕様	小
仕様漏れ	製品共通の機能における画面遷移において、現状では製品 B の設定が不十分になってしまうため製品 B の時だけ専用画面への遷移を追加する	製品共通の機能であるが、製品 B のときだけ標準の画面遷移の途中に専用画面を挟み込む必要があった。しかし、仕様には記載されていないため総合テスト工程で発見された。	仕様	中

不具合分類	不具合事例	原因	要求 or 仕様	不具合の影響度
仕様漏れ	アクセス権限を設定する画面にて、製品 B の専用権限が製品 A でも表示されてしまう	アクセス権限設定画面は製品共通の画面であるが、製品 B のときだけ使用する権限だとは気付かずに標準として表示してしまった。仕様には記載されていない。	仕様	中
仕様漏れ	設定が必須なパラメータがすべて設定されているかを確認する機能において、製品 A は正しく動作するが製品 B は正しく動作しない	必須パラメータの設定未済み確認機能は製品共通であるが、製品 B だけは確認条件が異なる。しかし、仕様には記載されていなかったため対応が漏れてしまった。	仕様	中
仕様不明瞭	製品 B のある機能が有効なとき、パラメーター一覧画面にその機能が有効である旨のアイコンが表示されるはずだが表示されない	パラメーター一覧画面は製品共通の画面であるが、製品 B のときだけある機能の有無を参照して表示方法を変えなければならない。しかし、製品 A のみ知っている担当者が共通の母体を改造する際に、仕様条件が不明瞭だったため、要求された内容を正しく解釈できなかった。	仕様	中
仕様不明瞭	製品 B において被検査品の質量を表示するグラフがグレーにならない	被検査品の質量を表示するグラフの色使いは製品共通であるが、製品 B においてはある条件のときだけグレーで表示する必要があった。しかし、仕様に記載されている条件が曖昧だったため、設計から漏れてしまった。	仕様	小
仕様伝達漏れ	製品 B において、被検査品の判定機能が正しく動作しなくなった	製品 A 向けの仕様変更対応を実施したところ、その機能が製品共通であったため別製品に影響を与えてしまった。共通仕様であることを関係者で共有できていなかったことが原因である。	仕様	大
仕様伝達漏れ	製品 A にて動作確認画面を表示したとき、あるべきはずのボタンが表示されない。以前は表示されていた	動作確認画面は製品共通の画面であるが、製品 A のときだけ専用ボタンが表示される仕様であった。しかし、並行開発されている製品 B の仕様変更にて修正を加えた結果、製品 A に影響してしまった。	仕様	大

(1) 共通仕様の定義

【Project_COM】
測定可出力
<要求>
[PR_0001][COM]検査機が検査品を測定できる状態のとき、I/O 出力信号を ON にする信号を追加して欲しい。
<理由>
前段の検査機と当該検査機をシリアルケーブルで接続し、前段の検査機に検査品を測定できる状態である（検査品を流してよい）ことを知らせるために利用する。これにより、前段の検査機は検査開始できる状態となる。
<仕様>
[SP_0001][COM]運転開始時、測定可出力信号を ON にする。(PR_0001)
[SP_0002][COM]運転停止時、測定可出力信号を OFF にする。(PR_0001)
[SP_0003][COM]エラー発生時は、測定可出力信号を OFF にする。(PR_0001)
[SP_0004][COM]異常発生時、運転を停止するときは、測定可出力信号を OFF にする。(PR_0001)
[SP_0005][COM]エラーリセット時、「異常時の処置」の設定値が「無停止、かつ運転中」の場合は、測定可出力信号を ON にする。(PR_0001)
[SP_0006][COM]被検査品オーバーフロー検出有効時、オーバーフロー個数に達した場合は、測定可出力信号を OFF にする。(PR_0001)
[SP_0007][COM]被検査品オーバーフロー検出有効時、オーバーフロー個数にリセットした場合は、測定可出力信号を ON にする。(PR_0001)
[SP_0008][COM]不良品停止時は、測定可出力信号を OFF にする。(PR_0001)
[SP_0009][COM]運用確認中は、測定可出力信号を OFF にする。(PR_0001)
[SP_0010][COM]運用確認終了時、運転中の場合は、測定可出力信号を ON にする。(PR_0001)
[SP_0011][COM]振分テストモード中は、測定可出力信号を OFF にする。(PR_0001)
[SP_0012][COM]洗浄モード中は、測定可出力信号を OFF にする。(PR_0001)

(2) 共通仕様に対する製品 A 個別仕様の定義

【Project_製品 A】
測定可出力
<要求>
[PR_0001][COM]検査機が検査品を測定できる状態のとき、汎用 I/O 出力信号を ON にする信号を追加して欲しい。
<理由>
前段の検査機と当該検査機をシリアルケーブルで接続し、前段の検査機に検査品を測定できる状態である（検査品を流してよい）ことを知らせるために利用する。これにより、前段の検査機は検査開始できる状態となる。ただし、「○○○調整中」「○○○補正中」「振分確認調整中」「検査機点検中」は本検査機の補正運転モードのため、測定可出力信号は OFF にしなければならない。
<仕様>
[SP_0001]「自動初期値設定警告」発生時、測定可出力信号を OFF にする。(PR_0001)
[SP_0002]「被検査品検出センサ異常」発生時、測定可出力信号を OFF にする。(PR_0001)
[SP_0003]○○○調整中は、測定可出力信号を OFF にする。(PR_0001)
[SP_0004]○○○補正中は、測定可出力信号を OFF にする。(PR_0001)
[SP_0005]振分確認調整中は、測定可出力信号を OFF にする。(PR_0001)
[SP_0006]検査機点検中は、測定可出力信号を OFF にする。(PR_0001)

(3) 共通仕様に対する製品 B 個別仕様の定義

【Project_製品 B】
測定可出力
<要求>
[PR_0001][COM]検査機が検査品を測定できる状態のとき、汎用 I/O 出力信号を ON にする信号を追加して欲しい。
<理由>
前段の検査機と当該検査機をシリアルケーブルで接続し、前段の検査機に検査品を測定できる状態である（検査品を流してよい）ことを知らせるために利用する。これにより、前段の検査機は検査開始できる状態となる。ただし、「自動設定時」「○○○補正中」「○○○モードが有効」のときは本検査機の補正運転モードのため、測定可出力信号は OFF にしなければならない。
<仕様>
[SP_0001]自動設定時は、測定可出力信号を OFF にする。(PR_0001)
[SP_0002]○○○補正中は、測定可出力信号を OFF にする。(PR_0001)
[SP_0003]○○○モード有りのときは、測定可出力信号を OFF にする。(PR_0001)

【付図 5 の要求リポジトリの補足】

製品 A と製品 B で共通の仕様である測定可出力機能について、3.1 で説明した要求リポジトリを作成したものである。要求リポジトリには変更依頼を一元化した 1 つの Word ファイルで、共通仕様と各製品の製品個別仕様を【】で示すタグによりページを分割して記載される。(1)は製品 A と製品 B の共通仕様を示し、(2)は製品 A の製品個別仕様、(3)は製品 B の製品個別仕様を示す。<要求>に示すマーキングした文（黄色マーカー）は各製品で共有する要求を示し、<仕様>に示すマーキングした文（モノクロ印字の関係上、水色マーカーで示す。本来は赤色マーカー）は各製品で共有する仕様を示し、“UN_xxxx”で示されるマーキングした文（緑色マーカー）は保留仕様を示す（保留仕様の例は付図 7 を参照）。この 3 色のマーカーで示した範囲の文が USDM への入力対象になる。要求リポジトリの Word ファイルに埋め込んだマクロを実行することにより、要求と仕様にそれぞれ一意の識別番号を自動付与する。識別番号は、要求には“PR_xxxx”，仕様には“SP_xxxx”の形式で付与する。また、要求や仕様の文頭に共通仕様を示す表記として[COM]と記述すると共通仕様を意味する。

付図 5 共通仕様の定義と製品ごとの個別仕様の定義を示した共通リポジトリ（抜粋）

共通仕様のUSDM		TracerID	Common A	Common B	Common C	Common D	Common E	A.C	B.C	C.C	D.C	E.C	F.C	G.C	H.C	I.C
要求	IOCTR	検査機が検査品を測定できる状態のとき、I/O出力信号をONにする信号を追加して欲しい。	[PR_0001]													
理由	説明	前段の検査機と当該検査機をシリアルケーブルで接続し、前段の検査機に検査品を測定できる状態である（検査品を流してよい）ことを知らせるために利用する。これにより、前段の検査機は検査開始できる状態となる。														
		<I/O出力信号の制御>														
	IOCTR-1	運転開始時、測定可出力信号をONにする。	[PR_0001]_[SP_0001]													
	IOCTR-2	運転停止時、測定可出力信号をOFFにする。	[PR_0001]_[SP_0002]													
	IOCTR-3	エラー発生時は、測定可出力信号をOFFにする。	[PR_0001]_[SP_0003]													
	IOCTR-4	異常発生時、運転を停止するときは、測定可出力信号をOFFにする。	[PR_0001]_[SP_0004]													
	IOCTR-5	エラーリセット時、「異常時の処置」の設定値が「無停止」、かつ運転中の場合は、測定可出力信号をONにする。	[PR_0001]_[SP_0005]													
	IOCTR-6	被検査品オーバーフロー検出有効時、オーバーフロー個数に達した場合は、測定可出力信号をOFFにする。	[PR_0001]_[SP_0006]													
	IOCTR-7	被検査品オーバーフロー検出有効時、オーバーフロー個数にリセットした場合は、測定可出力信号をONにする。	[PR_0001]_[SP_0007]													
	IOCTR-8	不良品停止時は、測定可出力信号をOFFにする。	[PR_0001]_[SP_0008]													
	IOCTR-9	運用確認中は、測定可出力信号をOFFにする。	[PR_0001]_[SP_0009]													
	IOCTR-10	運用確認終了時、運転中の場合は、測定可出力信号をONにする。	[PR_0001]_[SP_0010]													
	IOCTR-11	振分テストモード中は、測定可出力信号をOFFにする。	[PR_0001]_[SP_0011]													
	IOCTR-12	洗浄モード中は、測定可出力信号をOFFにする。	[PR_0001]_[SP_0012]													

製品BのUSDM		TracerID	Common A	Common B	Common C	Common D	Common E	A.C	B.C	C.C	D.C	E.C	F.C	G.C	H.C	I.C
要求	IOCTR	検査機が検査品を測定できる状態のとき、汎用I/O出力信号をONにする信号を追加して欲しい。	[PR_0001]													
理由	説明	前段の検査機と当該検査機をシリアルケーブルで接続し、前段の検査機に検査品を測定できる状態である（検査品を流してよい）ことを知らせるために利用する。これにより、前段の検査機は検査開始できる状態となる。ただし、「自動設定時」「○○○補正中」「○○○モードが有効」のときは本検査機の補正運転モードのため、測定可出力信号はOFFにしなければならない。														
		<I/O出力信号の制御>														
	IOCTR-1	自動設定時は、測定可出力信号をOFFにする。	[PR_0001]_[SP_0001]													
	IOCTR-2	○○○補正中は、測定可出力信号をOFFにする。	[PR_0001]_[SP_0002]													
	IOCTR-3	○○○モード有りのときは、測定可出力信号をOFFにする。	[PR_0001]_[SP_0003]													

製品AのUSDM		TracerID	Common A	Common B	Common C	Common D	Common E	A.C	B.C	C.C	D.C	E.C	F.C	G.C	H.C	I.C
要求	IOCTR	検査機が検査品を測定できる状態のとき、汎用I/O出力信号をONにする信号を追加して欲しい。	[PR_0001]													
理由	説明	前段の検査機と当該検査機をシリアルケーブルで接続し、前段の検査機に検査品を測定できる状態である（検査品を流してよい）ことを知らせるために利用する。これにより、前段の検査機は検査開始できる状態となる。ただし、「○○○調整中」「○○○補正中」「振分確認調整中」「検査視点検中」は本検査機の補正運転モードのため、測定可出力信号はOFFにしなければならない。														
		<I/O出力信号の制御>														
	IOCTR-1	「自動初期値設定警告」発生時、測定可出力信号をOFFにする。	[PR_0001]_[SP_0001]													
	IOCTR-2	「被検査品検出センサ異常」発生時、測定可出力信号をOFFにする。	[PR_0001]_[SP_0002]													
	IOCTR-3	○○○調整中は、測定可出力信号をOFFにする。	[PR_0001]_[SP_0003]													
	IOCTR-4	○○○補正中は、測定可出力信号をOFFにする。	[PR_0001]_[SP_0004]													
	IOCTR-5	振分確認調整中は、測定可出力信号をOFFにする。	[PR_0001]_[SP_0005]													
	IOCTR-6	検査視点検中は、測定可出力信号をOFFにする。	[PR_0001]_[SP_0006]													

【付図6のUSDMの補足】

3.2で説明したツールにより出力したそれぞれのUSDMは、測定可出力機能について要求リポジトリから自動生成された要求と仕様を示している。共通仕様のUSDMには、製品Aと製品Bに共通の仕様が記載され、製品Aと製品BのUSDMには製品個別仕様が記載される。つまり、製品A開発はこの共通仕様と製品固有仕様を実装する。また、各USDMの“Tracer ID”列の識別子は、要求リポジトリと、USDMに記載された要求と仕様を紐づけるものである。ツールは、この識別子を参照して要求リポジトリから出力された中間ファイルを読み込んでUSDMを自動生成する。なお、“Tracer ID”セルが薄緑色（モノクロ印字の関係上、薄緑色で示す。本来は赤色）で示される要求や仕様は、共通仕様であることを示す。これは、各製品の固有仕様が共通仕様に関連することを気付きやすくするためである。

付図6 共通リポジトリから自動生成した中間ファイルツールで自動読み込みしたUSDM（抜粋）

(1) 共通仕様の定義

不良判定出力
<要求>
[PR_0002][COM]検査機が検査品を測定したとき、不良判定の場合、I/O 出力信号を ON にする信号を追加して欲しい。
<理由>
検査品の判定結果が不良あることを前段あるいは後段の検査機に知らせるために利用する。
<仕様>
[SP_0001][COM]検査品の判定結果が〇〇不良のとき、NG 判定出力信号を ON にする。 (PR_0002)
[SP_0002][COM]検査品の判定結果が〇〇不良のとき、NG 判定出力信号を ON にする。 (PR_0002)
[SP_0003][COM]検査品の判定結果が〇〇不良のとき、NG 判定出力信号を ON にする。 (PR_0002)
[SP_0004][COM]異常発生するとき、不良判定出力信号を ON にする。(PR_0002)
[UN_0001]検査品を判定するとき、前回の不良判定出力信号を OFF にする。(PR_0002)

(2) 共通仕様に対する製品 A 個別仕様の定義

不良判定出力
<要求>
[PR_0002][COM]検査機が検査品を測定したとき、不良判定の場合、I/O 出力信号を ON にする信号を追加して欲しい。
<理由>
検査品の判定結果が不良であることを前段あるいは後段の検査機に知らせるために利用する。
<仕様>
[SP_0001]検査品の判定結果が〇〇不良のとき、NG 判定出力信号を ON にする。 (PR_0002)
[SP_0002]検査品の判定結果が〇〇不良のとき、NG 判定出力信号を ON にする。 (PR_0002)
[SP_0003]検査品の判定結果が〇〇不良のとき、NG 判定出力信号を ON にする。 (PR_0002)

(3) 共通仕様に対する製品 B 個別仕様の定義

不良判定出力
<要求>
[PR_0002][COM]検査機が検査品を測定したとき、不良判定の場合、I/O 出力信号を ON にする信号を追加して欲しい。
<理由>
検査品の判定結果が不良であることを前段あるいは後段の検査機に知らせるために利用する。
<仕様>
[SP_0001]検査品の判定結果が〇〇不良のとき、NG 判定出力信号を ON にする。 (PR_0002)

【付図 7 の補足】

要求リポジトリに 3.1.2 で説明した保留仕様を記述したときの例である。「(1) 共通仕様の定義」に共通仕様と疑われる仕様を [UN_xxxx] (緑色マーカー) でマーキングし、保留仕様を表したものである。本要求リポジトリから中間ファイルを出力し、そのファイルを読み込んだ USDМ を付図 8 に示す。

付図 7 保留仕様が記載された要求リポジトリ (抜粋)

共通仕様の USDM

要求	IOCTR2	検査機が検査品を測定したとき、不良判定の場合、I/O出力信号をONにする信号を追加して欲しい。	[PR_0002]							
理由説明		検査品の判定結果が不良であることを前段あるいは後段の検査機に知らせるために利用する。								
		<前段/後段検査機へのI/O出力信号>								
	IOCTR2-1	検査品の判定結果が〇〇不良のとき、NG判定出力信号をONにする。	[PR_0002]_[SP_0001]							
	IOCTR2-2	検査品の判定結果が〇〇不良のとき、NG判定出力信号をONにする。	[PR_0002]_[SP_0002]							
	IOCTR2-3	検査品の判定結果が〇〇不良のとき、NG判定出力信号をONにする。	[PR_0002]_[SP_0003]							
	IOCTR2-4	異常発生するとき、不良判定出力信号をONにする。	[PR_0002]_[SP_0004]							
	IOCTR2-5	検査品を判定するとき、前回のNG判定出力信号をOFFにする。	[PR_0002]_[SP_0005]							

【Project_COM】 [UN_0001]検査品を判定するとき、前回の不良判定出力信号をOFFにする。(PR_0002)

製品 A の USDM

要求	IOCTR2	検査機が検査品を測定したとき、不良判定の場合、I/O出力信号をONにする信号を追加して欲しい。	[PR_0002]							
理由説明		検査品の判定結果が不良であることを前段あるいは後段の検査機に知らせるために利用する。								
		<前段/後段検査機へのI/O出力信号>								
	IOCTR2-1	検査品の判定結果が〇〇不良のとき、NG判定出力信号をONにする。	[PR_0002]_[SP_0001]							
	IOCTR2-2	検査品の判定結果が〇〇不良のとき、NG判定出力信号をONにする。	[PR_0002]_[SP_0002]							
	IOCTR2-3	検査品の判定結果が〇〇不良のとき、NG判定出力信号をONにする。	[PR_0002]_[SP_0003]							

【Project_COM】 [UN_0001]検査品を判定するとき、前回の不良判定出力信号をOFFにする。(PR_0002)

製品 B の USDM

要求	IOCTRL2	検査機が検査品を測定したとき、不良判定の場合、I/O出力信号をONにする信号を追加して欲しい。	[PR_0002]							
理由説明		検査品の判定結果が不良であることを前段あるいは後段の検査機に知らせるために利用する。								
		<前段/後段検査機へのI/O出力信号>								
	IOCTR2-1	検査品の判定結果が〇〇不良のとき、NG判定出力信号をONにする。	[PR_0002]_[SP_0001]							

【Project_COM】 [UN_0001]検査品を判定するとき、前回の不良判定出力信号をOFFにする。(PR_0002)

【付図 8 の補足】
 付図 7 の要求リポジトリに記録された“保留仕様”を、3.2 で説明した USDM に自動読み込みした例である。共通仕様を示す USDM と各製品の USDM に保留仕様が表外に記録される。

付図 8 保留仕様が記録された USDM (抜粋)