

# 「T型マトリクス」を用いたXDDPとテストプロセスの接続

## — T 4 研究会活動報告 —

The concatenation of XDDP and test process using "T-type matrix"

—The activity report of AFFORDD T4 study group—

厚田 鳴海	(パナソニック ファクトリーソリューションズ株式会社)
名野 響	(ルネサスマイクロシステム株式会社)
矢野 恵生	(株式会社デンソー)
井芹 洋輝	
大山 相達	
長友 優治	(株式会社ベリサーブ)
永田 敦	(ソニー株式会社)

**概要** 近年、ソフトウェア開発の多くが派生開発によって行われるようになってきている。XDDP (eXtream Derivative Development Process) [1]による派生開発アプローチは、派生開発特有の問題に対して効果をあげてきた。XDDPでは、派生開発における要求を合理的に実装する方法について述べている。しかし、テストについては言及されておらず、XDDP適用後も適切なテスト範囲を設定できないため、納期遅延やテスト漏れによる不具合流出が発生している。派生開発におけるテストでは、ソフトウェア変更に対する影響範囲を効果的に特定し、要求される品質レベルに応じた適切なテスト範囲を設定することが重要になる。

本研究では、XDDPの成果物である変更要求仕様書（トレーサビリティマトリクス（以下、TM）付き）を活用し、「T型マトリクス」を用いてテストプロセスと接続し、効果的にテスト分析を行い、テスト方針に従ったテスト範囲を設定する方法を提案する。本手法を用いることにより、変更要求と既存テスト項目との関連を可視化し、品質レベルに応じたテスト項目の選定を行うことができる。また、「T型マトリクス」と変更要求仕様との整合性をとることによって、ソフトウェア開発者は、変更に対する影響範囲の気づきを得ることができる。

### Abstract

Most of the software development would be the derivation developments. XDDP (eXtream Derivative Development Process) is a dedicated process to the derivative development, solving the specific problem and improves its quality, delivery and cost.. However, XDDP does not describe the test process therefore we have some problem in the test design process and missed the test cases. It would make - delivery delay and failures in the customer sites, because we could not identify the influenced test scope. For the test in the derivation development, it is important to identify the range of the influence of software modification effectively, and to specify the appropriate test range according to the required quality level.

In this study, we propose a method of concatenation between XDDP and testing process using "T-type matrix" We can perform a test analysis process effectively and can specify a test range according to a test policy by utilizing modification requirements specification (with TM). We could not only visualize the relationship between the modification requirements and the existing test item, but also we could choose the appropriate test items according to the quality level. In addition, the software developer could specify the influence of the software modification by ensuring the consistency between "T-type matrix" and changed specifications.

### 1. はじめに

派生開発とは、ベースとなるソフトウェアをもとに機能追加や仕様変更を行うことによって、リリースを行う開発をいう。近年、ソフトウェア開発の多くが派生開発によって行われるようになってきている。その特徴とは、一般に新規開発に

比べて期間は短く、納期、コストともに厳しくなる。また、ドキュメント等が揃っていることはまれで、他人の書いたソースコードに対して全体を理解しないまま、思い込みと勘違いの中で修正が行われる。

そのような中、清水吉男氏は派生開発に特化し

たプロセス「XDDP (eXtream Derivative Development Process)」[1]を提唱している。XDDPの変更プロセスでは、差分情報をもとに「3点セット」と呼ばれる成果物を連鎖させ、担当者の思い込みや勘違いを防止する。この手法により、派生開発における設計からコーディングに至るまでの課題は改善されてきているが、テストについては言及されていない。

派生開発におけるテストは、ベースとなるソフトウェアは、正しく動作していたという前提の中で実施される。短い開発期間では、システム全体をテストするという事は難しく、変更箇所および影響範囲を想定したテストが中心になる。派生開発によって、ベースソフトウェアの大規模・複雑化やシリーズ展開による開発機種の増加が進むと、影響範囲を見通すことが難しくなってくる。影響範囲を考慮した適切なテスト範囲が設定できないと、過大なテストによる納期遅延や、テスト不足による不具合流出を発生させる。

研究の結果、差分情報に特化したXDDPの成果物だけでは、既存のテスト項目との関連性や影響範囲の特定が難しく、かつ、設定したテスト範囲に対して、要求された品質レベルを定めたテスト方針に対する妥当性を確認しづらいことがわかった。そこで我々は、XDDPにおいて、テスト方針に従った適切なテスト範囲を設定し、テストプロセスへ接続することを目標とした。

XDDPにおける変更要求から、適切なテスト範囲を設定するために、下記2つの内容を組み合わせ、「T型マトリクス」として用いる手法を考案した。

- ・変更要求—テスト項目マトリクス：今回の派生開発における要求と既存のテスト項目との関連性を示したマトリクス

- ・モジュール—テスト項目のマトリクス：ベースとなるソフトウェアの構造とテスト項目の関連性を示したマトリクス

本手法を用いることにより、テスト方針に従いテスト項目を選択し、テスト範囲として設定することが可能になった。また、変更要求仕様書(TM付)と組み合わせることで整合性をとることにより、テスト範囲の妥当性の確認、更には、ソフトウェア開発者に対して変更要求仕様に対する影響範囲の気づきを与えることができた。

## 2. 現状分析と課題

### 2.1. テストに関するアンケート

XDDPとテストプロセスを接続するための

課題を抽出するために、派生開発に対してXDDPを実施している組織に対して、テストに関するアンケートを行った。対象は、担当したソフトウェアに対して単体テスト/結合テストを行う開発者と、開発は行わず、システムテストを専任しているテスト技術者である。その結果を次に示す。

表1 テストに関するアンケート結果

	開発者	テスト技術者
担当テスト	単体テスト／結合テスト	システムテスト
主なテスト範囲	<ul style="list-style-type: none"> <li>・変更/追加したモジュール</li> <li>・変更/追加した機能</li> <li>・変更/追加したインターフェース</li> </ul>	<ul style="list-style-type: none"> <li>・変更/追加した機能</li> <li>・影響を受ける機能</li> <li>・回帰テスト</li> </ul>
XDDPとの親和性	変更点が明確でテスト設計しやすい	差分情報しかないため、テスト範囲の設定が難しい
テストにおける課題	<ul style="list-style-type: none"> <li>・変更箇所以外の項目に対するチェックが漏れる。</li> <li>・テスト項目に重複、漏れが発生しやすい。</li> <li>・第三者による妥当性確認が難しい。</li> </ul>	<ul style="list-style-type: none"> <li>・影響範囲の特定が難しい。</li> <li>・今回変更内容と既存の機能との関連がわかりにくい。</li> <li>・テスト範囲の妥当性をレビューしづらい</li> </ul>

### 2.2. 開発者の実施するテストに関する課題

アンケートの結果から、開発者はXDDPを行うようになって、修正した内容が明確になるため、テスト設計しやすい、といった回答が得られた。しかし、下記課題が従来プロセスと同様に存在することが分かった。

#### ①影響範囲を含めたテストが漏れやすい

結合テストでは、担当した範囲内で、変更した内容を実現できているか、といった局所的な内容が中心になるため、影響範囲を含めたテスト検討は実施されず、チェックが漏れやすい。

#### ②テスト項目に重複や漏れが発生する

作成したテスト仕様書は、次回以降も流用することになるが、他のメンバーが作成した複数のテスト仕様書が存在する場合、テスト項目に重複や漏れがあっても気づくことはまれである。

#### ③第三者による妥当性確認が難しい

検討した結果をドキュメントに残さず、直接テ

スト仕様書を変更しているため、第3者によるテスト範囲の妥当性確認が難しい。

### 2.3. テスト技術者の実施するテストに関する課題

システムテストを担当するテスト技術者のアンケート結果から、XDDPを採用することによって、テスト設計が難しくなっており、XDDPのメリットを得られていない、ということがわかった。この内容について、開発者とテスト技術者の視点の違い、従来開発とXDDPの違いに着目して分析を行った。

#### 2.3.1. 開発者とテスト技術者の違い

開発者とテスト技術者の比較を行った。

表2 開発者とテスト技術者の違い

	開発者	テスト技術者
最終成果物	ソースコード	テスト仕様書/成績書
開発における関心事	変更要求を実現するための方法 ・いかに効率よく作るか ・いかに品質良く作るか	システムに与える影響を検証 ・変更は正しく実施されているか ・システムに対する影響ある/なし
視点	深堀 (局所的)	全体俯瞰 (大局的)

開発者の場合、最終成果物はソースコードであり、その関心事は、具体的にソースコードのどこを変更すれば、効率よく、品質よく、変更要求を実現できるかということになる。その作業は、修正箇所をピンポイントで探し当てるための深堀をして行くことになる。XDDPでは、変更要求仕様をソースコードレベルで詳細に記述し、余分な情報を省いた「差分」情報のみを扱うので開発者にとってはマッチしていると思われる。

一方、テスト技術者の場合、最終成果物はテスト仕様書/成績書になる。その関心事は、変更点だけでなく、全体を俯瞰してシステムとして正しく動作しているのかを検証する。よって、システムの振る舞いとして変更による副作用がないことの確認も実施する。このような視点で見たとき、XDDPの成果物では、あまりにも詳細（実装レベル）なため、テスト技術者の場合、システム全体に対する影響がみえない、といった課題が生じている。

### 2.3.2. テスト技術者から見た従来開発とXDDPの違い

テスト技術者視点で従来開発とXDDPの比較を行った。図1は、従来開発におけるテスト仕様書作成の流れ、図2は、XDDPにおけるテスト仕様書作成の流れを示している。

図1より、従来開発の場合は、機能仕様書を先に作成する。そこには、変更部分だけでなく、その周辺の内容も記載されているため、振る舞いの変化やシステムに与える影響、既存のテスト仕様書との関連も見通しやすくなっている。しかし、機能仕様書には、afterの記述しかなく、コーディングにあたって抜け・漏れが生じやすくなっている。

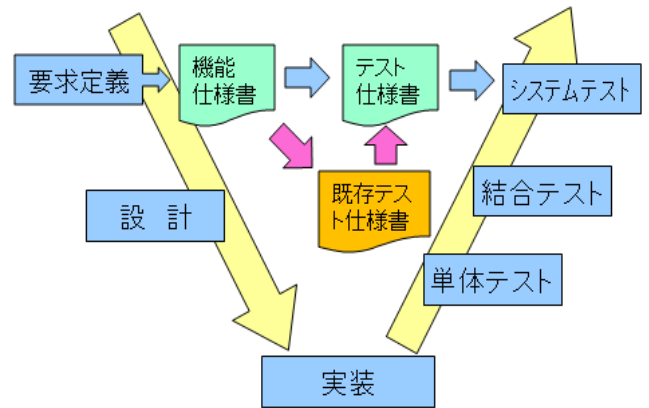


図1 従来開発におけるテスト仕様書作成の流れ

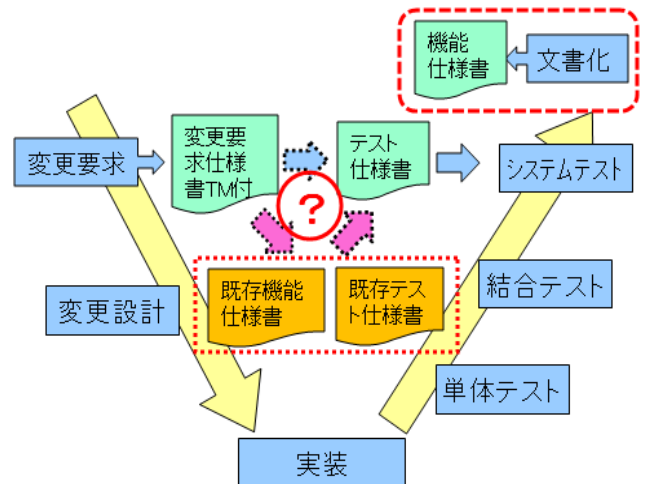


図2 XDDPにおけるテスト仕様書作成の流れ

図2より、XDDPの場合は、詳細なレベルかつ既存システムとの差分を扱う。そのため、変更要求からコーディングまでは、抜け・漏れが生じにくくなっている。一方、機能仕様書は、テスト後にベースとなる既存の機能仕様書とマージされるので、テスト仕様書を作成する段階ではまだ更新されていない。したがって、既存機能との

対応が取りづらく、変更要求仕様書に記載されている情報をもとに、テスト項目を選定することが困難になっている。

### 2.3.3. テスト技術者の実施するテストに関する課題の整理

2.3.1 および 2.3.2 より、テスト技術者が実施するテストの課題を整理すると下記になる。

#### ① システムに対する影響範囲特定が難しい

テスト技術者が XDDP 成果物を参照してテスト設計を行おうとすると、成果物の記述内容が、詳細（実装レベル）であるため、システムに対する影響範囲の特定が難しい。

#### ② 既存機能との対応がとりづらい

従来プロセスでは、修正された機能仕様書をもとにテスト仕様を変更していたが、XDDP では、機能仕様書の更新がテスト後になるため、今回、今回変更内容と既存の機能との対応がとりづらく、テスト項目の選定が困難になっている。

#### ③ テスト範囲の妥当性確認が難しい

①、②の状況より、テスト範囲の選定は、XDDP 成果物をもとにテスト設計を行うテスト技術者のスキルに依存しやすくなる。よって、テスト方針に従ったテスト範囲を設定できているか、その妥当性確認が難しい。

## 3. 施策

### 3.1. 課題解決の考え方

2.2、2.3 で述べた課題を解決するためには、テスト項目と今回の変更要求との関連を把握し、テスト全体を俯瞰することが重要である。

その実現方法として、変更要求とテスト項目の

マトリクスを使用する方法があるが、これだけでは、変更要求に対して関連するテスト項目にマークする技術者のスキルに依存してしまう可能性がある。

そこで、テスト項目とベースとなるソフトウェアのソフト構造（モジュール）との関連を明確にし、XDDP 成果物である TM を活用することができれば、変更要求とテスト項目の関連を記載する際、技術者のスキル依存を軽減し、妥当性確認の根拠とすることができると考えた。

上記内容を踏まえて、我々は、「変更要求ーテスト項目ーモジュール」の関連を示した「T型マトリクス」を提案し、XDDP とテストプロセスを接続することとする。

「T型マトリクス」によって、下記効果が期待できる。

- ① テスト全体を俯瞰し、影響範囲を変更要求とソフト構造（モジュール）の双方から確認できる。
- ② 結合テストにおいては、テスト項目とモジュールの関連を表現することにより、テスト項目と担当しているモジュールを明確にできるので、テスト項目の重複や漏れを見つけやすい。
- ③ システムテストにおいては、T型マトリクスのテスト項目に機能を記述することで、既存機能との対応をとることができる。
- ④ 変更要求仕様書（TM 付）と組み合わせて整合性をとることにより、第三者が見て、その妥当性を検証しやすい。また、文書として検討結果を残すことができる。

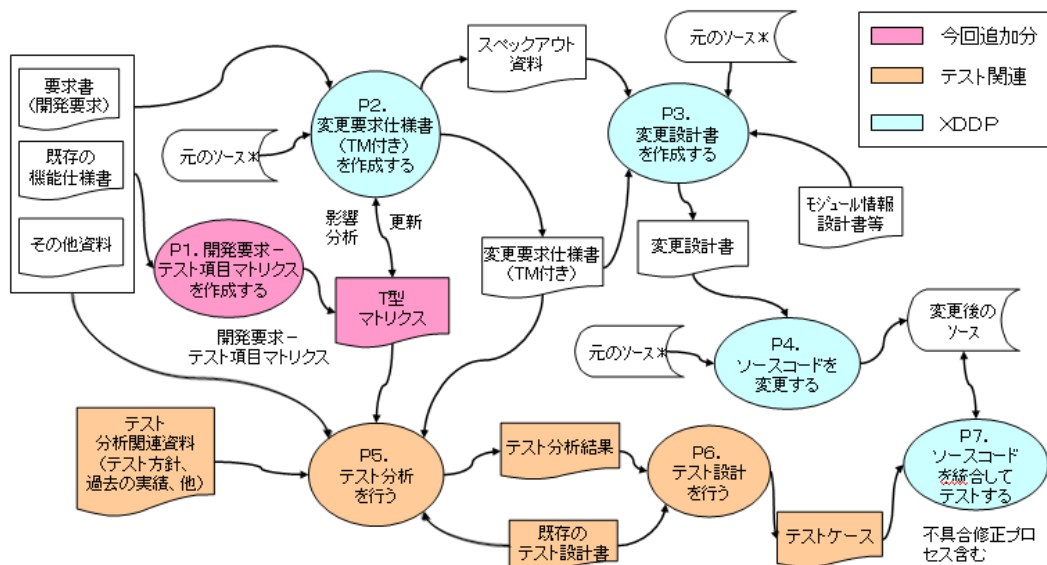


図3 XDDPにT型マトリクスを組み込んだ開発プロセス

### 3.2. T型マトリクスを組み込んだ開発プロセス

図3にXDDPの変更プロセス（PFD）に、T型マトリクスを組み込み、更にテストプロセスを追加した開発プロセスを示す。

P1は、今回提案したT型マトリクスを作成するプロセスである。

P2、P3、P4、P7は、XDDPの変更プロセスである。

P5は、テスト分析を行うプロセスであり、T型マトリクスと変更要求仕様書及び過去のテスト実績、不具合情報、機能の重要度等からテスト方針に基づき、テスト範囲を決定する。

P5の結果を受けてP6にてテスト設計を行い、P7にてテストを実施する。

### 3.3. T型マトリクス詳細

T型マトリクス（図4）は、下記2つのマトリクスの組み合わせにより構成される。

- ・変更要求ーテスト項目マトリクス（左側）
  - ・モジュールーテスト項目のマトリクス（右側）
- 以下、それぞれの構成要素について説明する

変更要求1	変更要求2	変更要求3	変更要求4	..	変更要求 テスト項目	モジュール テスト項目A	モジュール1	モジュール2	モジュール3	モジュール4	モジュール5	..
						テスト項目B						
						テスト項目C						
						テスト項目D						
						テスト項目E						
						...						

図4 T型マトリクス

#### 3.3.1. 変更要求ーテスト項目マトリクス

変更要求ーテスト項目マトリクスは、派生開発における変更要求とテスト項目との関連を表すことで、今回の変更がテスト項目に与える影響を表現する。（図5）

ここでのテスト項目とは、OK/NGを記載する個々のテストケースではなく、テスト仕様書にある大項目、中項目レベルの内容を記載する。

派生開発の場合、変更要求と既存のテスト項目の関係は、1：多になり、テスト漏れにつながりやすい。本マトリクスで可視化することによりテスト漏れを防止する。

- (1) 変更要求ーテスト項目マトリクスの列項目  
変更要求仕様書に記載された変更要求を記載する。

変更要求の粒度は、テストレベルによって変えることができる。一般に結合テストでは、

変更要求仕様書の変更要求と同じ内容を記載し、システムテストでは、今回の開発の最上位の要求を記載する。

- (2) 変更要求ーテスト項目マトリクスの行項目  
既存のテスト項目を記載する。結合テストの場合は、ソフト担当者間で認識できる内容の項目を記載する。システムテストの場合、テスト技術者と開発者がレビューして合意できる内容の項目を記載する。一般には、機能レベルの内容が記載される。

変更要求1	変更要求2	変更要求3	変更要求4	..	
					テスト項目A
					テスト項目B
					テスト項目C
					...

図5 変更要求ーテスト項目マトリクス

#### 3.3.2. モジュールーテスト項目マトリクス

モジュールーテスト項目マトリクスは、ベースとなるソフトウェア構造とテスト項目との関連を示す。（図6）

T型マトリクスにおいては、テスト項目を介して変更要求とソフト構造を結びその影響を明らかにし、かつ変更要求仕様書（TM付）と組み合わせて整合性をとることにより、妥当性を検証する要の役割を果たす。

	モジュール1	モジュール2	モジュール3	モジュール4	..
テスト項目A					
テスト項目B					
テスト項目C					
...					

図6 モジュールーテスト項目マトリクス

- (1) モジュールーテスト項目マトリクスの列項目  
XDDP成果物の1つであるTMの列要素を記載する

- (2) モジュールーテスト項目マトリクスの行項目  
変更要求ーテスト項目の行要素を記載する。

結合テストにおいては、テストで検証するモジュールの担当範囲を示し、テスト項目間の重複や漏れを防ぐために使用することができる。また、



システムテストにおいては、テスト項目に機能レベルの内容をおくことにより、機能とモジュールの関連を示し、ソフト構造の変更が機能に与える影響を把握することができる。

本マトリクスは、開発ごとに毎回作成するものではなく、ベースとなるシステムに対して一度作成すれば、差分情報をマージすることで、次回以降も使うことができる。

### 3.3.3. T型マトリクスの運用

T型マトリクスの運用手順を示す。

(1)テスト項目－モジュールマトリクスの作成  
行方向に列挙するテスト項目を、テストレベルに対応した既存のテスト項目より決定する。右列のモジュールには、XDDP成果物、TMの列方向に並べられているモジュールと同じものを列挙する。

次に、テスト項目に列挙した既存のテスト仕様書およびソースコード等の調査を行い、調査対象のテスト項目において、モジュールとの関連が認められたら「●」印をつける。

テスト項目－モジュールマトリクスは、特定のシステムにおいて事前に作成し、次回以降も更新を重ねながら使用する。

(2)変更要求－テスト項目マトリクスの作成

左列変更要求に、XDDP成果物、変更要求仕様書より、変更要求を列挙する。列挙する変更要求は、テストレベルに応じて調整する。

次に、変更要求と既存テスト仕様書の内容を照らし合わせながら、関連する項目に「○」印を記載する。

(3)T型マトリクスと変更要求仕様書の整合性チェック

変更要求仕様書(TM付)のモジュールとT型マトリクス右列のモジュールを連携させて、T型左列の開発要求とテスト項目の関連および変更要求仕様書の仕様とTMの関連に矛盾がないことを双方向に確認する。矛盾がある場合は、再度調査、検討し、変更要求仕様書(TM付)またはT型マトリクスのいずれかを修正する。

(4)テスト方針に従って、テスト項目を選択

(3)までで、今回の変更に対して影響を受けるテスト項目が特定できたので、テスト方針とあわせて、テスト項目を選択する。

テスト方針は、例えば、変更に影響あるテスト項目のみ実施する、変更部分+過去不具合の多い箇所を実施する、変更部分+基本機能を実施する等がある。

テスト項目の選択結果は、T型マトリクスに組み込むことが可能である。

その場合の例として、選択されたテスト項目に対し、下記マーキングを行う方法を示す。

- ・「○」: テスト実施  
テストケースの変更(又は新規作成)が必要
- ・「△」: テスト実施  
テストケースの変更が不要(回帰テスト)
- ・「×」: テスト不要  
整合性チェックで関連が認められたが、詳細な調査の結果、今回の変更箇所に関係等で、テストの必要なし。と判断したもの

### (5)T型マトリクスを用いたレビューの実施

T型マトリクスおよび変更要求仕様書をもとに、関係者とともにレビューをおこない、選択されたテスト項目の妥当性をチェックする。

T型マトリクスを活用することにより、結合テストでは、テスト項目を選択した根拠を残すことができる。システムテストでは、開発者とテスト技術者と異なる視点を持つ関係者が双方の視点でテスト範囲について合意し、テストを実装することができる。また、機能と開発要求の関連が明確になるため、この情報をもとに組み合わせテスト等を検討することができる。

### 3.3.4. 変更要求仕様書とT型マトリクスの連携例

図7は、変更要求仕様書とT型マトリクスを連携して整合性チェックを行った例を示している。

システムテストを想定し、T型マトリクス左列変更要求には、変更要求仕様書の最上位の要求である開発要求を記載している。(例では、開発要求1のみ記載)。

◆変更要求仕様書(TM付)→T型マトリクスの整合性確認

- ①開発要求1はモジュールAとCを変更する
- ②モジュールAとCはテスト項目AとGでテストする
- ③①と②より、開発要求1はテスト項目AとGに影響があると予想される
- ④開発要求1とテスト項目Aには、関連「○」が記入されているが、テスト項目Gには関連が記入されていない。
- ⑤検討の結果、テスト項目Gと開発要求1の関連に漏れが判明したため、「○」を追記する。

◆T型マトリクス→変更要求仕様書(TM付)の

整合性確認

- ⑥開発要求1はテスト項目Cを変更する（「○」を記入済み）
- ⑦項目Cは、モジュールEの変更をテストする。
- ⑧⑥と⑦より、モジュールEへの影響があると予想されが、変更要求仕様書（TM付）には、モジュールEへの変更が記載されていないので、モジュールEに対する影響漏れまたは変更仕様漏れがないか確認する。

4. 施策の検証

4.1. 検証方法

T型マトリクスを用いた本提案の効果を確認するために、テストレベルの異なる2事例に対して検証を行った。

【事例1】結合テストに対する適用

- (1) 適用対象
  - 情報装置の組み込みソフトウェア
- (2) T型マトリクス
  - テスト項目には、コンポーネントの結合テスト項目を配置、変更要求には、変更要求仕様書に要求を配置してT型マトリクスを作成した。
- (3) 開発者とテスト担当者の関係
  - 変更要求仕様書作成からソースコード修正、および結合テスト仕様書作成は、開発者が担当した。

【事例2】システムテストに対する適用

- (1) 適用対象
  - F A設備組み込み制御ソフトウェア
- (2) T型マトリクス
  - テスト項目にシステムテスト項目（＝システムの機能）を配置、変更要求には、変更要求仕様書の最上位の要求（＝開発要件）を配置して、T型マトリクスを作成した。
- (3) 開発者とテスト担当者の関係
  - 変更要求仕様書作成からソースコード修正を担当する開発者とは別にテスト技術者がテスト仕様書を作成している。

【事例1】は、実プロジェクトへ適用し評価をおこなったが、【事例2】に関しては、時間、評価環境の関係で適切なプロジェクトを調整できなかったため、過去事例において、XDDPを採用したプロジェクトメンバー（開発者、テスト技術者）に依頼し、シミュレーションを行った。

4.2. 検証結果

検証結果を次に示す。【事例2】に関しては、シミュレーションに参加した開発者、テスト技術者の評価を示す。

【事例1】結合テストに対する適用

- ① 従来は、変更要求仕様書の内容から直接テストケースを変更しており、多数の修正漏れが発生していた。今回、T型マトリクス

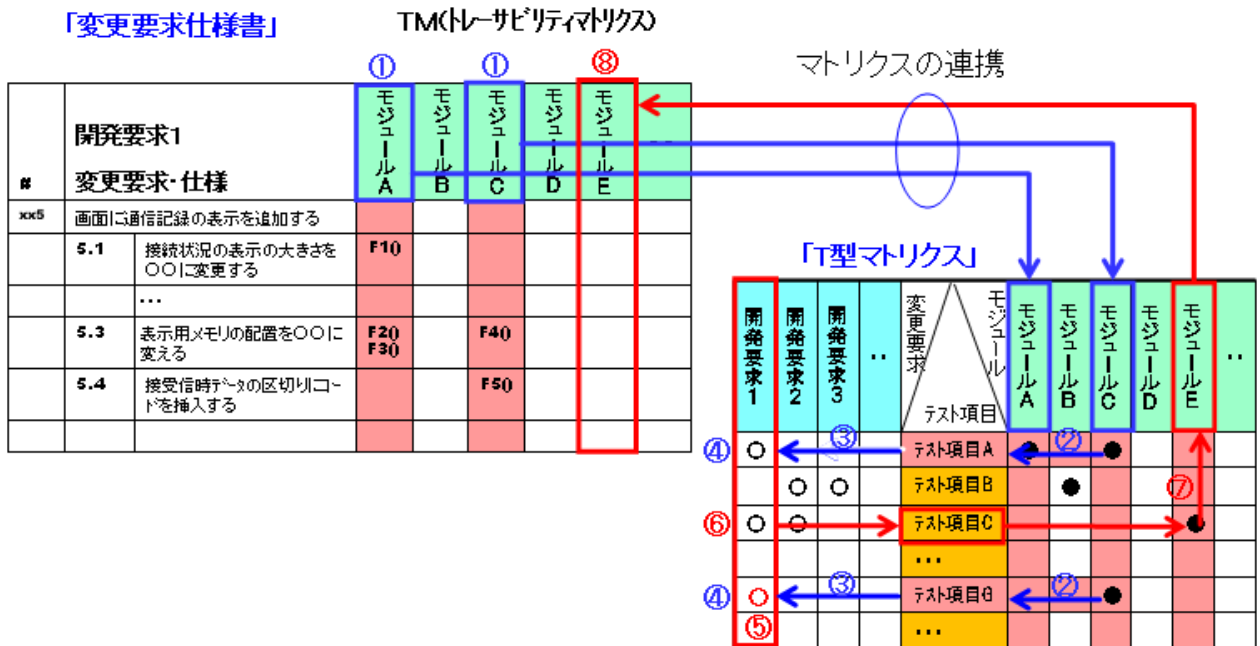


図7 変更要求仕様書TMとT型マトリクスの連携例

により、テスト仕様書修正前の変更箇所の特定と具体的な変更点の整理、第3者レビューが可能となり、テストケースの修正漏れをなくすことができた。(テストケース修正漏れ：10件→1件に改善)

(テストケース修正漏れの1件は、T型マトリクスの行に記載したテスト項目の内容を誤認し、T型マトリクスの交点に初めから現れなかったことが原因だった。)

- ② T型マトリクスを用いて変更要求仕様に影響するテスト項目を整理することで、変更箇所だけでなく、回帰テストが必要な項目も特定することができ、第3者レビューによって、テスト方針との妥当性を判断することができた。

### 【事例2】システムテストに対する適用

#### A. 開発者

- ① 過去実施したプロジェクトでは、システムテストで2件不具合が発生した。T型マトリクスがあれば、機能に対する要求仕様の漏れをチェックできるため、テスト前に不具合の混入を防げた可能性がある。
- ② T型マトリクスを用いて、同時に実施している別のプロジェクトとの情報をリンクさせることにより、並行開発にも活用でき、お互いに及ぼす影響を開発途中で検証できる。

#### B. テスト技術者

- ① T型マトリクスによって、早い段階で影響範囲が見えるので、計画を立てやすく、見積もり精度を高めることができる。
- ② 機能と開発要求の対応が明確になり、仕様を理解するための余分な調査をしなくてもよい。
- ③ 開発者とのレビューでしかテスト内容の妥当性をチェックできなかったが、T型マトリクスによって、テストの抜け・漏れをチェックしやすくなる。

## 5. 考察

【事例1】の結果より、結合テストのテストケース修正漏れに対して効果が得られた。結合テストにおいては、担当者が修正した内容を担当者自身がテストする機会が多いため、テスト項目を検討した記録がほとんど残っていない。今回、T型マトリクスによって、検討した結果が残るため、第3者によるレビューが効果を発揮したものと、考える。

しかし、ベースとなるソフトウェアの規模に応じてテスト項目が多くなると、テストの内容について関連した開発を実施した担当者しか把握していない可能性が高くなる。そのため、効果的なレビューを実施するためには、該当する機能およびモジュール構成に詳しいレビュアーを選定するか、または、テスト項目に気づきの得られる名称を設定する等、工夫する必要がある。

【事例2】の結果より、開発者、テスト技術者ともにメリットを得られることがわかった。開発者にとって、テストを意識した開発は必要とは思ってもおろそかになりがちであるが、T型マトリクスを使用して、影響範囲のチェックができるのであれば、自然とテストを意識した開発ができることになり、後工程での後戻りに効果を発揮することができると考えられる。

T型マトリクス作成に当たって、モジュールとテスト項目の関連をマトリクスで表現する必要がある。現状、人手を介して作成することになるため、大規模なシステムでは、本マトリクスを作成することがT型マトリクス導入の障壁になる可能性がある。最初から完璧を目指さず、関連が漏れやすい内容等に絞って徐々に完成させていく方法についても検討する必要がある。

## 6. まとめ

### 6.1. 取り組みと結果

テスト項目と変更要求の関連を可視化する「T型マトリクス」を用いて、XDDPとテストプロセスを接続し、派生開発におけるテスト分析を効率的に行う方法について提案した。結合テストにおける実プロジェクトでの検証とシステムテストにおける過去プロジェクトのシミュレーション検証により、その効果を確認した。

結果として、T型マトリクスによって、テスト項目全体を俯瞰しながら、変更要求との関連を明確にすることによって、抜け・漏れのないテスト範囲の設定に効果があることがわかった。

T型マトリクスは、既存のテスト仕様書をもとに作成可能である。大規模システムの場合、モジュールとテスト項目の関連作成に時間がかかる可能性があるが、関連が漏れやすい項目から徐々に充実させて行く方法をとれば、導入のための手間も最小限で抑えられると考えられる。

### 6.2. 今後の課題

本研究では、派生開発において、XDDPとテ



ストプロセスを接続する方法について提案した。しかしながら、派生開発においてテストを効果的に行うためには、膨大なテスト項目のなかからテスト範囲を絞り込む方法や、既存のテスト仕様書から変更分を抜け・漏れなくテストケースに落とし込む方法が必要であり、今後の課題としたい。

## 謝辞

本研究を進めるに当たり、貴重なご意見を頂きましたパナソニック株式会社 古山寿樹氏、効果の検証に協力頂き、有益なご討論、ご助言を頂いたパナソニック ファクトリーソリューションズ株式会社 開発センター各位に感謝の意を表する。

## 参考文献

- [1] 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、技術評論者、2007
- [2] 清水吉男：要求を仕様化する技術・表現する技術、技術評論社、2005