

テスト実施状況の可視化方法の提案と実践

Visualization of the test result and progress in the software testing

株式会社 東芝 ソフトウェア技術センター

Corporate Software Engineering Center, TOSHIBA Corporation

○久連石 圭 佐々木 愛美 河村 透 小笠原 秀人

○Kei Kureishi Manami Sasaki Toru Kawamura Hideto Ogasawara

Abstract The increasing complexity and sophistication of software in recent years has given rise to a need for more effective testing. In testing process, there are many viewpoints to see the test result and test progress. We proposed the data structure that shows relations of test items and test results. And we developed test management system using proposal data structure. This test management system can show various viewpoints of the test results and test progress.

1. はじめに

近年ソフトウェアが大規模化・複雑化してきている。一方で、開発期間は短縮される傾向にあり、短い期間で品質の高いソフトウェアを開発することが望まれてきている。その中で、ソフトウェア開発の最終工程であるテストの重要性もますます高くなってきている^{[1][2]}。大規模なソフトウェア開発の場合、数万を超えるテスト項目を扱い、数十万回のテストを実施することもある。その中で、テストの実施を管理し、テストの可否を明確にしておくことが、ソフトウェアの品質を保証する上でも必要となる。

ソフトウェアの品質問題や進捗の遅延を防ぐためにテストを管理する上で重要なことは、(1)テストの実施状況を確認し、(2)テストの実施状況から進捗や品質に関する問題を特定する。そして(3)特定した問題に対して適切な対策をおこなうことである。これら(1)～(3)はすべて必要な要素であるが、(2)、(3)には(1)の実施状況の確認が必須である。そのために、本稿ではテストの実施状況を把握するための方法に着目し、テスト実施状況を可視化できる方法を提案する。

2. テスト管理における課題

2.1. テスト実施状況を確認する際の課題

テスト工程において、ソフトウェアの品質やテストの進捗に関する問題を特定するためには、単に個々のテストの可否だけでなく、ソフトウェア全体の不具合情報や、機能やモジュールごとの品質傾向、時系列の不合格の発生傾向など複数の視点でテストの実施状況を早期に把握することが必要である。しかしながら、実際のテストではテストの実施状況を正確に把握することが難しいことがある。原因としては以下が考えられる。

(1) 最新のテスト結果が収集できない

テスト実施者が多い場合、迅速にテスト結果がテスト管理者に報告されないことや、報告されてもテストの結果が古く最新の状況が確認できないことがある。

(2) テスト結果からテスト実施状況がすぐにわからない

テスト実施状況を把握するには、収集したテスト結果を集計する必要がある。集計作業に時間がかかると、実施状況の確認に時間がかかってしまい、迅速な判断ができなくなる。

株式会社 東芝 ソフトウェア技術センター

Corporate Software Engineering Center, TOSHIBA Corporation

神奈川県川崎市幸区小向東芝町 1 Tel: 044-549-2439 e-mail:kei.kureishi@toshiba.co.jp
1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki-shi, Kanagawa Japan

(3) 収集したテスト結果から、意図する視点で実施状況を確認できない

開発しているソフトウェアのドメインや規模、担当する役割に応じて、必要なテスト実施状況の視点が異なることが多い。そのため、テスト結果を複数の視点で集計する必要があるが、すべての視点で集計することは難しい。

これらの課題のうち、(1)や(2)についてはテスト管理システム^{[3][4]}などを使って試みている例がある。一方、(3)については既存のテスト管理システムでは十分に可視化・分析できないことがあるため、本稿では(3)に着目する。様々な視点でテスト状況を確認できるようにするため、まずテストの視点を整理し、実施状況を可視化するために必要な要素を明らかにする。そして可視化に必要なテスト項目とテスト結果のデータ構造を提案し、そのデータ構造を用いたテスト管理システムを提案する。

2.2. テスト管理の視点

複数のテストの実施状況を確認する必要があると述べたが、どのような視点が考えられるか整理する。テストの実施状況を確認する視点として、テストの規模や、製品ドメイン、テスト工程、担当者の役割などが考えられる。すべての視点を列挙すると書き切れないため、ここでは担当者の役割について、テストの実施状況を確認する視点を明らかにする。

担当者の役割別に確認する視点をあげる前に、本稿で扱う役割を(a)テスト管理者、(b)テスト実施者、(c)ソフトウェア開発者の三者として定義する。実際の開発では他の役割がある場合や、役割を重複して持つ場合もあるが、本稿では単純化して以上の三者のみを対象とする。簡単に三者の役割を整理しておく。

(a) テスト管理者はテスト全体の進捗や不具合の発生状況を確認し、問題発生箇所を特定し、各テスト実施者に実施すべきテストを指示する。

(b) テスト実施者は、割り当てられたテストを実施し、可否を記録する。また潜在する不具合を見つけ出すために、テスト項目によらないテスト(探索的テスト)を実施することもある。

(c) ソフトウェア開発者は、自分が担当するモジュールや機能の開発を行う。テスト工程では、検出された不具合に対して不具合の発生条件を確認し、必要に応じて修正を行う。

上記の役割ごとに必要な代表的な可視化の視点を表1に記述する。表では、各役割について、テスト実施中に確認すべきポイントと、具体的に必要な情報についてまとめた。

表1 担当者ごとの可視化の視点

	可視化の目的	確認すべきポイント	具体的に必要な要素
テスト管理者	リリースの可否を判断する	すべてのテストの可否を確認する	すべてのテストに対する可否の集計
		不合格の収束状況を確認する	累積不合格数や信頼度成長曲線
	問題箇所を把握し、注力する部分を決定する	遅れが発生している部分を確認する	機能別や担当者別による予実比較
		不合格が多発している部分を確認する	不合格が多発しているテスト項目や機能の集計
	管理層に報告する内容をまとめる	全体の進捗状況を確認する	全テストの集計や進捗の時系列変化
		残存している不合格を確認する	残存不合格の一覧
テスト実施担当者	担当するテストを期日までにもれなく実施する	担当部分のテスト進捗状況を確認する	担当部分のテストの予実比較
		担当部分の未実施のテストを確認する	未実施テストの数と一覧
	探索的テストを実施するための注力箇所を決定する	担当部分の不合格検出箇所を確認する	担当部分の検出不合格一覧や多発不合格一覧
		他の担当者の不合格多発箇所を確認する	他の担当者が実施している部分の多発不合格一覧
ソフトウェア開発者	担当部分の可否を確認し、不具合を修正する	担当部分のテスト実施状況を確認する	開発部分のテストの実施件数
		担当部分の可否を確認する	開発部分のテストの可否集計
		修正されていない不具合を確認する	未修正不合格の一覧
	自分に関連するモジュールの不具合の影響を確認する	関連しているモジュールの不具合を一覧で確認する	関連モジュールの不合格検出一覧

上記のようにまとめると、可視化の目的や確認すべきポイントが各々異なるため、複数の観点でテスト結果を集計・可視化しなければならないことが明確になる。

2.3. 表形式のテスト仕様書を用いたテスト管理の課題

これまでのテストでは、表計算ソフトでテスト仕様書を記述し、そこにテスト結果を記録することが多い^[5]。この場合、2.2 節で述べた様々な視点を網羅的に確認するテスト仕様書を作ることが難しい。たとえば、Excel でテスト項目を作成すると図 1 のようなテスト項目の一覧表となることが多い。ここでテスト項目とはテストを実行する際のテストの手順と期待値を組み合わせたものとし、各テスト項目を実施することでテスト結果（合否）を記録する。多くの場合、テストは 1 度きりではなく、ソフトウェアが更新されるたび再度テストを行う。その場合、図 2 のようにテスト結果を右の列に追加していくことで、ソフトウェアの更新に対するテスト結果の変化を確認することができる。更に、機種や実施担当者ごとのテスト結果を確認し、テストの実施状況を細かく確認したい場合もある。表形式にテスト結果を記入しているときは、例えば各機種のテスト結果は列を追加していくことで表現し、各実施担当者のテスト結果は表を分けることで表現できる（図 3）。しかしながら、この表では複数の視点からテストの実施状況を簡単に確認するのは困難である。例えば「最新の Ver1.1 について、すべての機種の結果を確認したい」とテスト管理者が考えた場合、各担当者の表から Ver1.1 の列のテスト結果をすべて取り出して集計するといった作業が必要になる。2.2 節で取り上げたそれぞれの状況を確認する場合も、容易にテストの実施状況が確認できるとは言いがたい。

大項目 入出力機能

1回目(Ver1.0)					
番号	中項目	小項目	テスト項目	実施手順	結果 実施日
1	入力	個別出力	合 6/10
2			合 6/10
3			合 6/10
4		一括出力	否 6/12
5			否 6/12
6			合 6/12
7	出力	個別出力	合 6/11
7			否 6/11
9		一括出力	合 6/11
10			合 6/11

図 1 テスト仕様書の例

大項目 入出力機能

1回目(Ver1.0)						2回目(Ver1.1)	
番号	中項目	小項目	テスト項目	実施手順	結果 実施日	結果 実施日	結果 実施日
1	入力	個別出力	合 6/10	合 6/25	合 6/25
2			合 6/10	合 6/25	合 6/25
3			合 6/10	否 6/22	合 6/22
4		一括出力	否 6/12	合 6/27	合 6/27
5			否 6/12	合 6/25	合 6/25
6			合 6/12	合 6/25	合 6/25
7	出力	個別出力	合 6/11	合 6/25	合 6/25
7			否 6/11	否 6/25	合 6/25
9		一括出力	合 6/11	合 6/25	合 6/25
10			合 6/11	合 6/25	合 6/25

図 2 テスト結果を複数記入したテスト仕様書の例

大項目 入出力機能

実施担当 Aさん

					機種 A		機種 B		
					1回目(Ver1.0)	2回目(Ver1.1)	1回目(Ver1.0)	2回目(Ver1.1)	
番号	中項目	小項目	テスト項目	実施手順	結果 実施日	結果 実施日	結果 実施日	結果 実施日	
1	入力	個別出力	合 6/10	合 6/25	否 6/10	合 6/25	
2			合 6/10	合 6/25	合 6/10	合 6/25	
3			合 6/10	否 6/22	合 6/10	否 6/22	
7	出力	個別出力	合 6/11	合 6/25	合 6/11	合 6/25	
8			否 6/11	否 6/25	合 6/11	合 6/25	

実施担当 Bさん

					機種 A		機種 B		
					1回目(Ver1.0)	2回目(Ver1.1)	1回目(Ver1.0)	2回目(Ver1.1)	
番号	中項目	小項目	テスト項目	実施手順	結果 実施日	結果 実施日	結果 実施日	結果 実施日	
4	入力	一括出力	否 6/12	合 6/27	否 6/12	合 6/27	
5			否 6/12	合 6/25	否 6/12	合 6/25	
6			合 6/12	合 6/25	合 6/12	合 6/25	
9	出力	一括出力	合 6/11	合 6/25	否 6/11	合 6/25	
10			合 6/11	合 6/25	合 6/11	合 6/25	

図 3 機種やテスト実施者の情報を追加し、複雑になったテスト仕様書の例

上記の例では紙面上、テスト項目が 10 と少なく、機種やテスト実施者もそれぞれ 2 種類のみなので実施状況が確認できないとまではいえないが、実際には数十の実施担当者や機種があること、数万のテスト項目があることを考えると、表形式のテスト仕様書を用いた可視化方法では、テストの実施状況の把握は難しいといえる。

そこで次章では、表形式のテスト仕様書では集計が困難である複数の観点での実施状況の可視化をできるようにする方法を提案する。

3. テスト項目とテスト結果のデータ構造

前章では様々な視点でテスト結果を見る必要性を述べ、これまで広く使われてきている表形式のテスト仕様書では様々な視点でのテスト実施状況の確認が難しいことを述べた。本節では 2 章で明らかにした様々な視点でテスト結果を可視化するために必要なテスト項目とテスト結果のデータ構造を提案する。

3.1 テスト項目とテスト結果に必要な要素

2.2 節で列举したテスト管理の視点から、以下の要素が必要であることが明確になった。本稿で提案するデータ構造を用いることで、以下の要素を使った可視化を可能とする。

- ・対象となる機能：集計・可視化の対象となる機能を分類でき、その分類ごとにテスト実施状況が集計できること
- ・テスト実施者：テストの実施担当者ごとのテスト実施状況が集計できること
- ・対象バージョン：ソフトウェアの更新に応じたテスト実施状況が集計できること
- ・テストの予定：テストの実施予定に対する予定と実績の差を集計できること
- ・機種情報：複数の機種をテストする際に機種ごとのテスト実施状況を集計できること
- ・実施日：テストの実施状況が日ごとに確認できること
- ・テスト実施工数：テストに要した工数を集計できること

3.2 テスト項目及びテスト結果を管理するためのデータ構造

本稿では、図 4 で示すとおり、テスト項目及びテスト結果を関連づけたデータ構造を提案する。以後の項で、各要素について説明する。

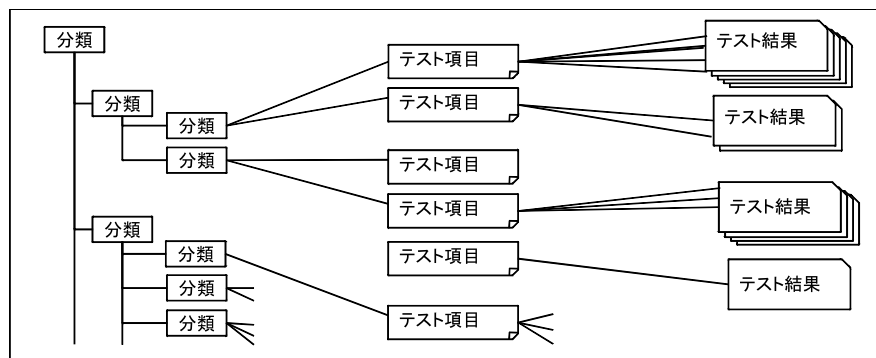


図 4 テスト項目の管理方法

(1) 木構造として構成するテスト項目の分類

テスト項目を意味的に分類できるように、木構造で構成する「分類」を用意する。分類では、テスト仕様書に大項目・中項目・小項目として記述されることが多いソフトウェアの機能やテストの観点などを記載する。また、木構造の利点として、深さに制限をなくすることができ、3 階層に収めなければならないという制限をなくすることができる。加えて、集計する際に全体的な状況を集計する場合や個々の状況を集計する場合など、選択する範囲に応じて集計する対象を変更しやすくなる。

(2) テスト項目とテスト結果の関連

テスト項目とテスト結果は一对多の関係で保管する。テストでは、不具合が検出されなくなるまで繰り返しテストをすることが多い。そのため、テスト項目 1 件に対してテスト結果は複数件存在する。これらのテスト結果は不合格が発生した日時やバージョンを知るためにすべて保管しておく必要がある。提案手法では、テスト項目 1 件に対して、複数件のテスト結果を関連づけられるようにする。同一テストを異なった機種などで実施する場合もあるが、機種などの情報はテスト結果に保管しておくことで、テスト対象は特定することができる。

(3) テスト項目の属性

テスト項目にはテストとして必要な操作手順や判定条件などの情報が必要である。加えて提案手法では、「テストの種別」を付加する(図5)。テストの種別は、テスト項目の検索・集計に用いるテスト項目の情報を追加しておくものである。例えば、リグレーションテスト時に毎回実施すべき基本的なテストについて「リグレーションテスト」と付与しておく。これらの情報は、集計する際の絞り込み条件や集計の対象として利用する。

(4) テスト結果の属性

テスト結果には合否だけでなく、どのような状態でテストを実施したかを特定するための情報が必要である。提案手法では、実施日、対象バージョン、実施担当者、対象機種、実施工数を保管できるようにしている(図6)。可視化するには、これらの情報を絞り込み条件や集計の対象として利用する。

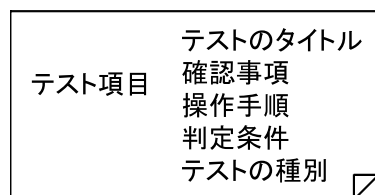


図5 テスト項目に必要な属性

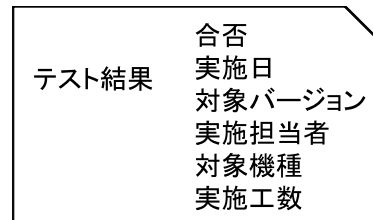


図6 テスト結果に必要な属性

(5) テストの実施予定

テスト工程における予定と実績の差を確認するため、テストの予定を計画できるようにする。図7で示すように、実施を予定している任意のテスト項目を選択し、実施期間と実施担当者、実施予定バージョンを記入し、テスト予定を作成する。テストの実施状況を確認する際は、テスト予定ごとにテストの結果を集計することで、テスト進捗の予実管理や実施担当者ごとのテスト実施状況を確認できる。

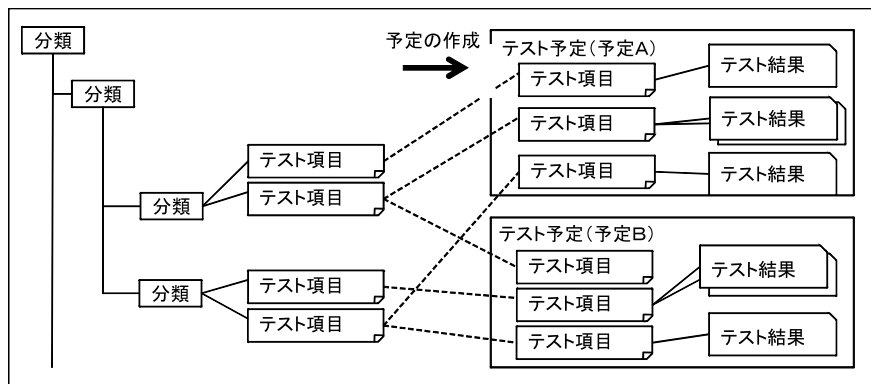


図7 テスト予定の計画

4. テスト管理システムの実現

我々は、前章であげたデータ構造を利用したテスト管理システムを実装し運用している。本章では我々が開発しているテスト管理システムを簡単に紹介する。

4.1. テスト管理システムの構成

テスト管理システムは Web アプリケーションとして動作するツール(図8)であり、サーバにテスト項目やテスト結果など提案したデータ構造で表現したすべての情報を格納している。ツールを Web アプリケーションとして実現することで、大人数で利用する



図8 テスト管理システム

際に、テスト結果を一元管理できる点と最新状況が把握できる点で大きな効果がある。

4.2. テスト項目の分類

テストの分類は「フォルダ」として木構造で構成している。テスト管理システム内では、画面の左にフォルダの構成を常に表示し、選択範囲を素早く切り替えられるようにしている。一般的には、テストの対象となる機能や、テストの工程(単体・結合・総合)、確認する観点(機能・性能・使用性)などをフォルダに対応付けて構成する。

4.3. テスト項目とテスト結果の登録

本システムは Web アプリケーションのため Web フォームを使ったテスト項目・テスト結果の登録を用意している。一方、テスト工程では大量のテスト項目やテスト結果を一括して登録することも多いため、ファイル(CSV ファイルや Excel ファイル)からの登録方法も用意している。

4.4. テスト項目及びテスト結果の集計・可視化

本システムでは、2 章であげた多様な視点をテスト工程で確認できるようにするため、テストの実施状況をリアルタイムでグラフ化できるようにしている。3 章であげたテスト項目及びテスト結果のデータ構造を用いた可視化の流れを図 9 に示す。

テストの実施状況を可視化する際は、まず本システムで保管しているテストのデータから、可視化の対象とするデータを絞り込む。ここでは、階層的に保管している分類情報や、テスト項目・テスト結果の属性情報を利用する。次に、テストのデータを任意の方法で集計する。例では不合格数と修正数の時系列グラフ描画のために、日別の不合格数と修正数を集計したが、バージョン別の不合格数や担当者別の工数なども集計できる。最後に、集計した値をグラフに描画する。

本システムでは、任意のグラフや表を一つの画面に同時に表示できるダッシュボード機能がある。ダッシュボードは利用者ごとに作成することができるため、各役割の確認すべき視点に適した応じたグラフ・表を各自設定し、描画することができる。

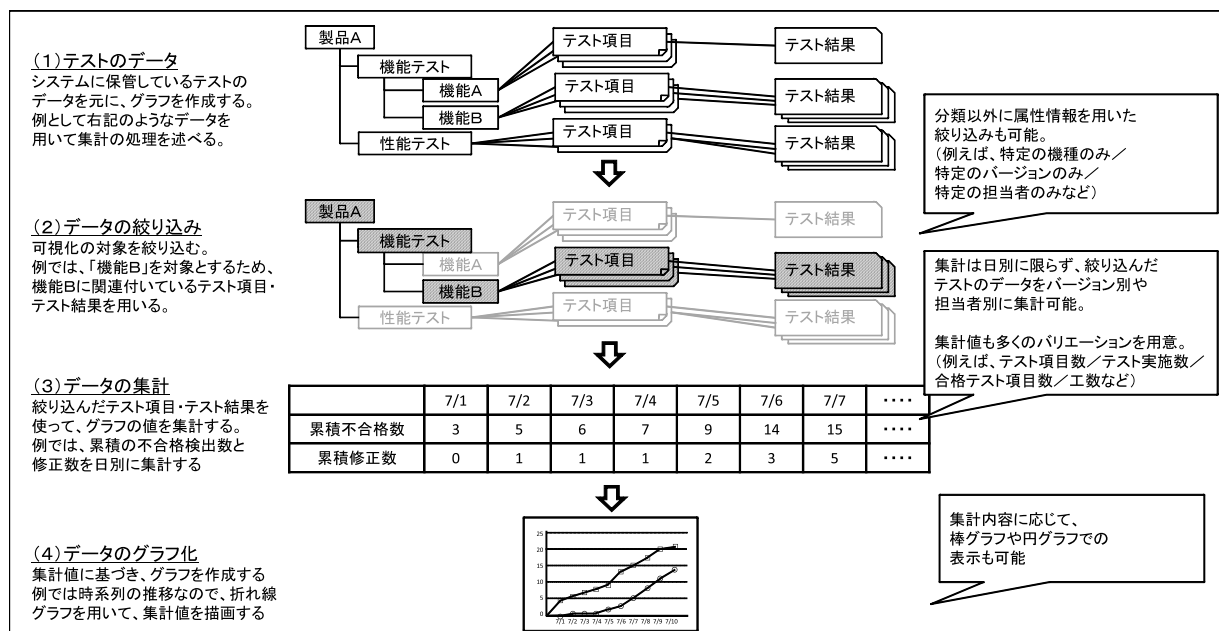


図 9 テストのデータを元にしたグラフ描画の流れ

5. テストにおける活用事例

前述したテスト管理システムを製品開発に適用した場合の可視化方法の実例を紹介する。本章では 2 章でまとめた各担当者の役割に対する可視化の目的を満たすために必要な情報を示し、各担当者の確認すべきポイントが可視化できることを示す。

5.1. テスト管理者に向けたテスト実施状況の可視化

(1) ソフトウェアの問題発生箇所の特典

テストの結果を機能（分類）別や機種別に集計し、表やグラフで提示する（図 10）。各機能や各機種で不合格が突出している場合、当該機能や機種に品質問題が潜んでいることが推定されるため、対策をとる必要がある。また、累積の不合格数だけでなく、直近 1 週間など、最近発生している状況を確認することで、今何が問題となっているのかなど、迅速な状況判断ができるようになる。



図 10 機能別や機種別の不合格数分布

(2) 上位管理層向けに報告書の作成

上位管理層には全体をとらえた状況の報告が必要であるため、全体を集計して提示する（図 11）。テストの実施傾向を知るためには、全体の可否の集計だけでなく、開始からのテスト実施状況や不合格の検出傾向を時系列のグラフで示すことが必要である。また、何らかの品質問題が発生している箇所に関しては、特に報告が必要な箇所である。このような箇所は、リスト化して報告できるようにしている。

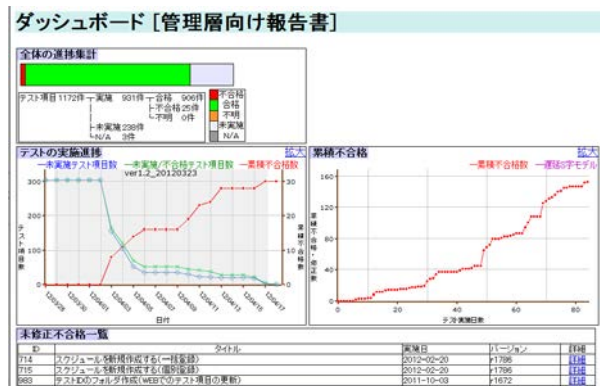


図 11 上位管理層向け報告資料

5.2. テスト実施者に向けたテスト実施状況の可視化

(1) 担当分の実施予定テスト一覧

各テスト実施者にテストを割り当てて実施する場合に、各自に割り当てられているテストの実施進捗や、未実施となっているテスト項目、不合格が発生し再テストが必要なテスト項目をテスト実施者ごとに集計表示している（図 12）。実施者ごとに実施状況を提示することで、自分で進捗を管理しやすくなり、遅れが発生しそうな状況を未然に察知しやすくなる。

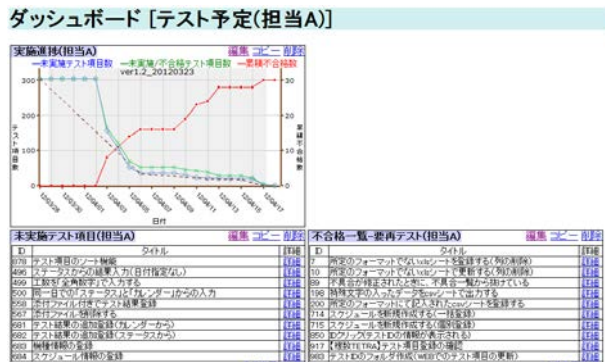


図 12 担当分のテスト実施状況及び残テスト一覧

(2) 探索的テストのための不合格多発箇所の提示

テスト実施者は与えられたテストだけでなく、テストで不具合を見つけられそうな部分に対して探索的にテストを実施する場合がある。これまで不合格が発生した箇所では再度不合格が発生する可能性が高いといった考えから、過去の不合格情報を参考にして探索的テストを実施する場合が多い。そのため、過去に失敗したテストやその発生箇所を一覧で表示することで（図 13）、過去の不合格検出状況からテストの優先順位を決めることができる。

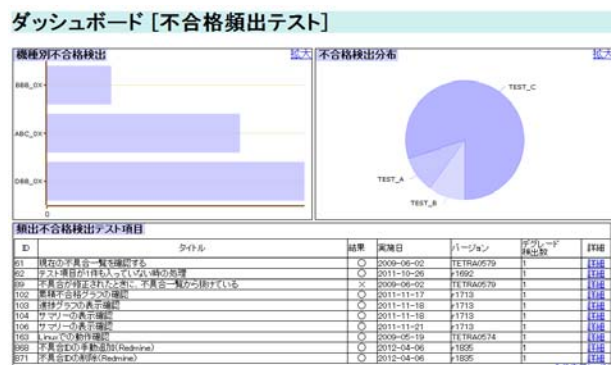


図 13 不合格頻出テスト項目の一覧

5.3 ソフトウェア開発者に向けたテスト実施状況の可視化

(1) 自分が担当している機能のテスト実施状況

自分が開発をしている機能の品質を確認するために、テストの実施状況や不合格の数、不合格の内容について一覧表示する（図 14）。これらの情報から、ソフトウェア開発者は修正の要否が判断できるようになり、効率的な修正につなげることができる。

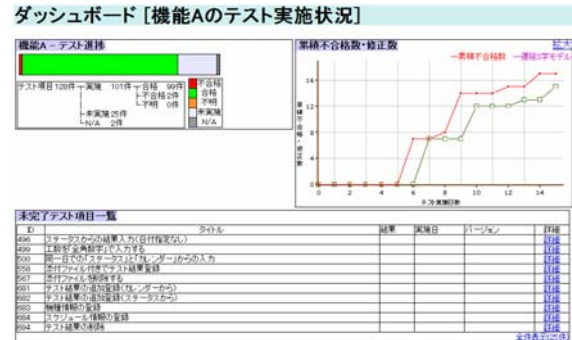


図 14 開発担当箇所のテスト結果一覧

(2) 自分に関連するモジュールのテスト実施状況

ソフトウェアを開発する際、自分が担当する機能に他の機能やモジュールが関係することがある。自分の機能が動作しない場合、自分の開発部分に問題があるのか、関係しているモジュールに問題があるのか把握しづらい。問題箇所を特定するためには、関連している機能やモジュールのテスト結果を確認し、問題の有無を知る必要がある。そのため、関連するモジュールをピックアップし、並べて表示することで、どこに問題があるのかを把握する（図 15）。



図 15 関連するモジュールの不合格一覧

6. 結びに

本稿ではテストの実施状況を確認することの難しさの一つに、多様な視点があることを課題として述べ、多様な視点で可視化するために必要なテスト項目とテスト結果のデータ構造を提案した。また、提案したデータ構造を用いたテスト管理システムを開発し、そのテスト管理システムを使ったテスト実施状況の可視化方法を紹介した。テストの実施状況を迅速に様々な視点で見られるようになると、問題の早期発見や問題箇所の特定が容易になる。その結果、素早い改善策に移ることができ、短いテスト期間でも品質を向上しやすくなる。

本稿で紹介したテスト管理システムは、東芝グループ内の複数の部門に展開し、活用されている。開発している製品や、規模や期間、対象としている工程が異なる中、本システムが利用され続けている理由は、本稿で提案したテスト項目を管理するためのデータ構造が多く部門のテスト管理に適用しやすいことが理由の一つとして考えられる。

今後は提案した可視化方法を用いて、テスト実施状況の認識方法と、とるべき対策の体系作りをしていきたい。具体的には、開発規模や開発プロセス、製品のドメインなどが異なる中で、各場面で有効な可視化方法や、実施状況に応じた対応の取り方などを、開発部門の事例を用いて検証し、まとめていきたい。

7. 参考文献

- [1] 岡崎毅久、ソフトウェアテストと品質保証の実際、日本テクノセンター、1999
- [2] 情報処理推進機構、ソフトウェア開発データ白書 2010-2011 第2版、情報処理推進機構、2011
- [3] 河村透、小笠原秀人、ソフトウェアのテスト管理システム、東芝レビュー2011年1月号、2011
- [4] 小川明彦、阪井誠、Redmineによるタスクマネジメント実践技法、翔泳社、2010
- [5] TEF プリンタ部会、テスト仕様書をどう書くか テストケースに何を挙げるか、ソフトウェアテスト PRESS Vol.6 pp.6-37、技術評論社、2008