

数値項目一覧表を用いた設計漏れ検出方法の提案

<A Proposal to Prevent Errors by Using Work-Flow Diagram and Numerical Definition>

主査 秋山浩一（富士ゼロックス（株））、副主査 奥村有紀子、堀田文明（（有）デバッグ工学研究所）
リーダー 木原克彦（USOL東京株式会社）
研究員 大野守（ソニー株式会社）、大橋剛和（富士通株式会社）、
片桐郁代（キャノンITソリューションズ株式会社）

概要

ソフトウェア障害の原因の多くは仕様関連であり、上流工程に集中しているとする先行研究がある[1]。そこで、研究員らの企業で発生した障害 17 事例を収集して分析した結果、利用想定、限界値、入力値など、数値で表されるような事項に関して、上流工程で設計漏れ・誤りが発生した事例が 70%以上あると判明した。

本研究では、このような「上流工程での数値に関連する設計漏れ」に着目し、既存の開発プロセスを変更せずに解決する手法を検討した。その結果、業務シナリオ（業務フロー）を中心に要件定義と基本設計をもとに数値に関連する設計情報の分析・評価をおこなうことで、システム全体を横断的にみた検証ができると考え、実際のプロジェクトで検証をおこなった。

本稿では、提案する手法の詳細と、検証した結果について報告する。

Abstract

Many of software bugs are caused by errors in the specifications of the software, injected during the upper stream design stage, according to preceding reports[1]. We analyzed 17 bugs around us and found that 70% of them are caused by the incompleteness of the numerical description, such as capacity/scale and the upper-bound/lower-bound of the system limitation (in some cases they are not considered at all).

In order to detect the incompleteness of numerical definition in upper stream, we propose a new work-method that uses work-flow diagram, requirement specification, and high level designs, to investigate "numerical definition and their values". We also applied this method to actual projects to see the effectiveness.

This report introduces the method and the result of the trial.

1 はじめに

ソフトウェア開発において上流工程に起因する問題が後工程であるテスト工程で見つかる場合が後を絶たず、上流工程で早期検出ができないプロジェクトが多いと報告されている [1]。研究員らの所属する企業においてもテスト工程や納品後の市場で発生する問題があり、共通課題としてとらえている。

また、一般的に問題の 70%が仕様関連であり上流工程に集中していると記されている[2]ことと、研究員らの経験から、問題の発生工程は、要件定義や基本設計といった上流工程の方が、プログラム開発などの下流工程よりも多いと考えた。

上記より本研究グループでは、本来上流工程で決めておくべき事項に対する「設計漏れ」を検出する方法を研究の課題対象として設定した。なお本研究では、「設計漏れ」を次のように定義し、要求漏れや要件漏れは含めないこととした。

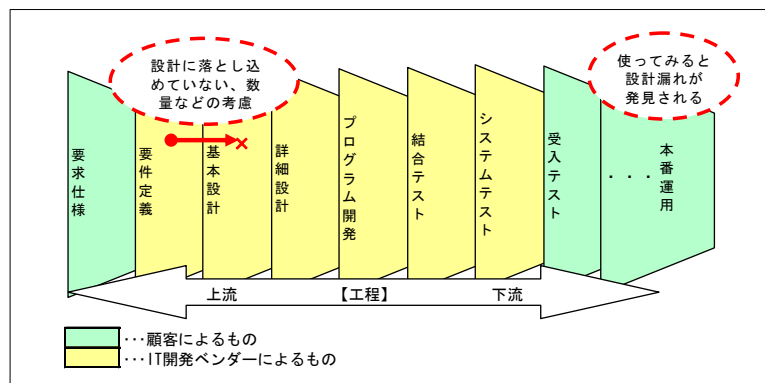
- 要件定義で明示的に定義されている事項について、機能レベルに展開して設計するときに漏れや矛盾が生じること
- 要件定義では明示していないが設計時に考慮すべき事項について、漏れや矛盾が生じること

また、研究の方向性は、既に確立された開発プロセスや品質マネジメントシステムを変更することなく即利用可能であり、開発の現場で特別な教育を必要としない方法とするために、次のように設定した。

- ① 新しい開発プロセスは追加せず、既存プロセスの中で設計品質を向上させる
- ② 実績あるレビュー方法やテスト技法の組合せか、新たな工夫の追加により効果をあげる
- ③ 実際の現場で利用しやすい方法を考える

本稿では、上記の方針にしたがって対象とすべき工程や観点を絞り込んだ上で、品質向上策を設定して部分的に検証した結果を報告する。

なお、開発工程の構成や呼称は企業ごとに異なっているため、本稿では図1のように定義する。



【図1 開発工程】

2 現状と課題

2.1 事例による現状把握（原因分類）

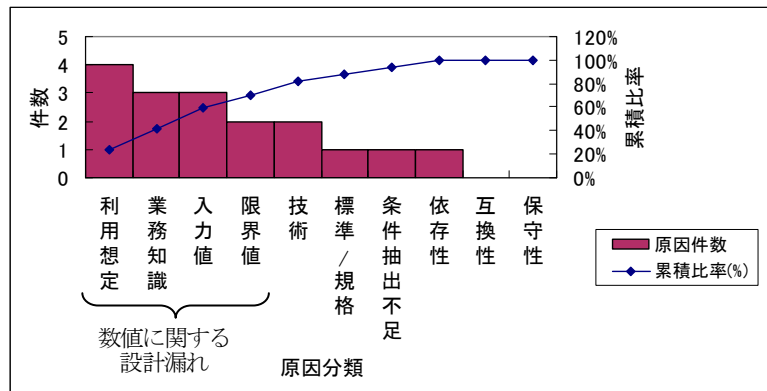
現状を把握し原因と課題を明確にするために、研究員の所属する企業で実際に発生した障害事例を収集し分析した。

まず、納品後に市場で発生した障害について 17 事例を収集し、原因分類によって整理したところ、図2のような傾向がみられた（詳細は「添付-1 障害事例の原因分類」を参照のこと）。

このうち、取扱いデータ量やアクセス数などの利用想定あいまいさ（＝利用想定）、業界標準の境界値や規定に対する考慮不足（＝業務知識）、画面入力やプログラム間の連携によるデータ入力値の範囲設定あやまり（＝入力値）、ディスクやネットワークの容量などシステム全体としての限界値の設定漏れや不適切な設定（＝限界値）など、数値に関連する設計漏れが 12 事例であった。こうした数値に関連する設計漏れは、全事例 17 に対して 70.6%を占めている。

さらに、数値に関連する設計漏れ12事例について研究グループにて問題点と真の原因を分析した結果、表1の4つの問題点と真の原因に集約できた。

さらに、数値に関連する設計漏れ12事例について研究グループにて問題点と真の原因を分析した結果、表1の4つの問題点と真の原因に集約できた。



【図2 原因分類での集計結果】

【表 1 数値に関する設計漏れの問題点と真の原因】

No.	問題点	真の原因
1	設計時、どのような数値に関して設計が必要か洗い出せていない	どのような数値に関する設計をおこなうかが不明確であり、設計担当者の知識や経験に依存しているため
2	設計をおこなっているものの、見積りの精度が低い	設計するための調査や検討が不十分（思い込みや知識不足も含む）、または設計値の妥当性を確認する方法が不足しているため
3	システム全体に関する数値的な要件や設計値から、個々の機能への展開で設計漏れが発生する	システム全体の数値的な設計がおこなわれていても、機能ごとにどう展開していくのかを明らかにする方法がないため あるいは、機能ごとの設計値が、システム全体にどう影響するかを確認する方法がないため
4	設計値に矛盾や不整合が生じる	相互に関連する機能について、設計値の矛盾や不整合を横断的に確認する方法がないため

2.2 事例による現状把握（原因工程）

2.1 項で数値に関連する設計漏れ 12 事例について問題点と真の原因を分析した（分析結果の詳細は「添付-1 障害事例の原因分類」を参照のこと）。次に分析結果をもとに、「本来どの工程で、どのような成果物で定義・設計されるべき事項か」「どの工程で、どのような作業により問題を発見するべきだったか」について研究グループ内で考察した。その結果、設計漏れ 12 事例は、すべて次の原因工程に問題があることが判明した。

- ・ 業務シナリオ（業務フロー）や利用想定などの成果物において、要件定義や基本設計など上流工程で定義・設計すべき事項である。
- ・ 要件定義や基本設計など上流工程でのレビュー、または業務シナリオやシステムに負荷が掛かった際の動作確認などのシステムテストの設計を上流工程と並行実施して検証することにより検出されるべきものである。

まとめると、障害 17 事例では、数値に関する設計漏れが要因の 70%以上を占め、かつ、これらは上流工程において設計されるべき問題であったといえる。

2.3 課題の設定

2.1 項と 2.2 項での分析結果より、解決すべき課題は次の 5 つとした。

- ① 本来、設計対象とすべき数値項目を抽出するための標準的な考え方の不足（表 1 の No.1）
- ② 個々の設計値を確認する方法の不足（表 1 の No.2）
- ③ システム全体と個々の機能間の、垂直方向に横断的に確認する方法の不足（表 1 の No.3）
- ④ 機能間での、水平方向に横断的に確認する方法の不足（表 1 の No.4）
- ⑤ 上記①～④による設計漏れを上流工程で検出する方法の不足（2.2 項）

3 品質向上策

3.1 設計漏れ検出方法

2.3 項で挙げた課題を解決するには、設計対象とすべき数値項目を抽出するための標準的な考え方と、個々の設計値を確認でき、かつ垂直方向や水平方向で横断的に確認できる方法が必要である。本研究グループでは品質向上策として、数値項目一覧表を用いて設計漏れを検出する方法を考案した。設計漏れ検

出方法は次のとおりである（作業手順の詳細、数値項目一覧表、業務シナリオのサンプルについては、「添付-2 設計漏れ検出方法の詳細」、「添付-3 作成例 1」、「添付-4 作成例 2」を参照のこと）。

【前提となる設計情報（インプット）】

- ・ 要件定義書、業務シナリオ（業務フロー）、基本設計書
- ※ 業務シナリオ（業務フロー）が要件定義にて作成済みであることが前提となる
- ※ 基本設計と並行しての実施を想定しており、画面設計書、機能設計書、データベース設計書などが順次できあがるのを受けて、インプットしていく

【作成する資料（アウトプット）】

- ・ 数値項目一覧表

本研究において提案する数値項目一覧表のフォーマットを以下に示す。

【表 2 数値項目一覧表のフォーマット】

通し番号	照合番号	振舞い	構成要素 対象の機能/処理/データ	キー ワード	数値項目	値の出元 要件/設計/想定
1	1	サイトへアクセス	ログインメニュー表示	時間	レスポンスタイム	想定
2	1	サイトへアクセス	ログインメニュー表示	回数	ページ表示回数	想定
3	2	ログイン	ログイン処理	時間	レスポンスタイム	要件
4	2	ログイン	ログイン処理	回数	ページ表示回数	想定

通し 番号	照合 番号	振舞い	構成要素 対象の機能/処理/データ	キー ワード	数値項目	値の出元 要件/設計/想定	上限	下限	最頻値	単位	値の根拠 ※想定の場合は必須	備考	検証タイミング
1	1	サイトへアクセス	ログインメニュー表示	時間	レスポンスタイム	想定	1	-		秒	情報を取得する必要がないため		システムテスト
2	1	サイトへアクセス	ログインメニュー表示	回数	ページ表示回数	想定	1	1	1	回	ログイン前に表示するのみ		
3	2	ログイン	ログイン処理	時間	レスポンスタイム	要件	5	-		秒			システムテスト
4	2	ログイン	ログイン処理	回数	ページ表示回数	想定	1	1	1	回	ログインするのは1回のみ		

上限	下限	最頻値	単位	値の根拠 ※想定の場合は必須	備考	検証タイミング
1	-		秒	情報を取得する必要がないため		システムテスト
1	1	1	回	ログイン前に表示するのみ		
5	-		秒			システムテスト
1	1	1	回	ログインするのは1回のみ		

【実施するタイミング】

- ・ 要件定義が完了しており、基本設計を実施している段階（基本設計と並行しての実施を想定）

【実施する人物像・体制】

- ・ これを実施する体制としてテスト分析・設計チームを想定している。このチームには、設計情報に基づいてシステムテストの分析と計画・設計をおこなえるスキルを有するメンバー（テストエキスパートと呼ぶ）がいることを想定している。

【設計対象とすべき数値項目を抽出するための標準的な考え方（視点・観点）】

- ・ 「時間」「容量」「回数」「頻度」等をキーワードにして、具体的な項目を決める。

【数値項目一覧表への記録方法】

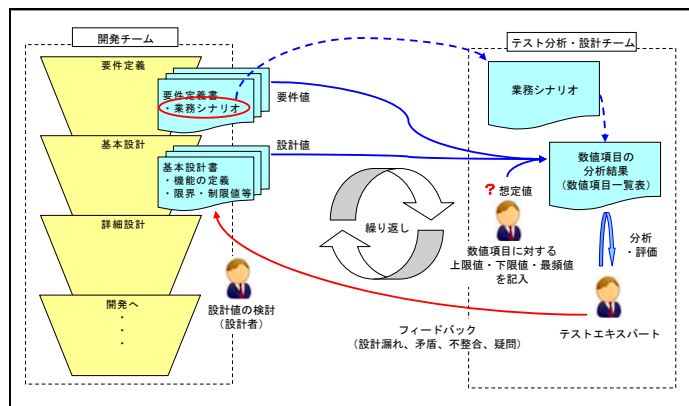
- ・ 業務シナリオ（業務フロー）をもとにして、機能・データベース・繰り返し操作などシステムの構成要素ごとに、本来設計対象とすべき数値項目を抽出する（図 3-①）。
- ・ 数値項目の上限値・下限値・最頻値は、設計情報（要件定義・基本設計）から得る（図 3-②）。
- ・ 数値項目を抽出するときに用いたキーワードを数値項目一覧表に記録する（図 3-③）。
- ・ 上限値・下限値・最頻値が設計情報から得られない場合には、システムや機能の特性、類似システムでの実績などをもとに想定した値を数値項目一覧表に記録する（図 3-①～④）。またこのとき、値の出元が想定であることと、想定の本拠もあわせて記録する（図 3-④、⑤）。

①	③	①	④	②	⑤						
通し 番号	照合 番号	振舞い	構成要素 対象の機能/処理/データ	キー ワード	数値項目	値の出元 要件/設計/想定	上限	下限	最頻値	単位	値の根拠 ※想定の場合は必須
1	1	サイトへアクセス	ログインメニュー表示	時間	レスポンスタイム	想定	1	-		秒	情報を取得する必要がないため
2	1	サイトへアクセス	ログインメニュー表示	回数	ページ表示回数	想定	1	1	1	回	ログイン前に表示するのみ
3	2	ログイン	ログイン処理	時間	レスポンスタイム	要件	5			秒	
4	2	ログイン	ログイン処理	回数	ページ表示回数	想定	1	1	1	回	ログインするのは1回のみ

【図 3 数値項目一覧表の記入例】

【設計漏れ検出方法】

- 数値項目一覧表を用いて、下記の視点で数値項目とその値に関する確認・分析をおこなう。数値項目一覧表には業務シナリオの構成要素ごとの数値に関連する設計情報および想定値がまとまっているため、個々の機能、機能間、システム全体、それぞれの視点で確認する。
 - 数値が入らない空欄部分は、設計漏れが疑われる。また、想定値が記入されている箇所は、もととなる設計情報がないことから、設計において考慮すべき事項である可能性がある。
 - 一連の処理や操作の観点でチェックをおこない、そこに含まれる数値項目の間に矛盾や不整合がないかどうかを確認する。
 - 処理や操作をまたがって、関連する数値項目間で矛盾や不整合がないかどうかを確認する。
- 設計漏れ、矛盾、不整合、疑問などの分析結果を設計に対してフィードバックする。開発チームではこれらを解消して設計書に記載するなどの対応をする。作業プロセスと各チームの作業分担のイメージを図 4 に示す。



【図 4 作業プロセスのイメージ】

3.2 期待される効果

3.1 項で述べた設計漏れ検出方法は、2.3 項であげた 5 つの課題の解決策として以下の点から有効であると考えられる。

- 課題①への策として、設計対象とすべき数値項目を抽出する標準的な考え方を定義している。「時間」「容量」「回数」「頻度」等をキーワードに数値項目を抽出するとともに、上限値、下限値、最頻値を抽出している。
- 課題②③④への策として、業務シナリオに基づいて数値項目一覧表を作成することで応じている。業務シナリオには、アクター、自システムの構成要素（機能・データベース・メディア・デバイスなど）、連携先のシステムの構成要素（前出に同じ）などが表現されている。これをもとに数値項目一覧表を作るため、個々の機能のみならず、機能間・システム全体など、横断的に俯瞰できる。
- 数値項目一覧表の内容は、要件定義と基本設計の情報をもとに作成するため、基本設計と並行して実施でき、課題⑤への対策となる。

4 実プロジェクトでの効果の検証

本研究にて考案した数値項目一覧表を使った設計漏れ検出方法の有効性および効果を検証するため、実際のプロジェクトに適用して効果を検証した。

4.1 検証に使用したプロジェクトと検証目的

品質向上策の効果確認に加え、実施する人物のスキルレベルでの相違を踏まえて、表3のとおり検証ケースを設定した。検証に適用したプロジェクトは3件で、ケース1~4のうち、ケース3と4は同一プロジェクトである。なお、ケース1と2のプロジェクトは、既に運用されているシステムであり、開発~テストまで全ての工程が完了している。ケース3と4のプロジェクトは、開発中のプロジェクトである。

【表3 実プロジェクトの概要と検証の内容】

検証ケース	適用プロジェクトおよび業務シナリオ	実施する人物像(スキルレベル)	検証目的と実施範囲
1	携帯向け Web サイト開発プロジェクト 情報検索の業務シナリオ ※開発・テストを完了し運用中のシステム	テストエキスパート	目的:3.2項で挙げた期待される効果が得られるか検証する。 実施範囲:3.1項および添付2の内容にしたがって数値項目一覧表を作成し、確認・分析までおこなう。
2	携帯向け Web サイト開発プロジェクト 簡易な情報表示の業務シナリオ ※開発・テストを完了し運用中のシステム		
3	映像機器向け組込アプリ開発プロジェクト リニア編集システムの業務シナリオ ※開発中の組込みソフト		
4	映像機器向け組込アプリ開発プロジェクト リニア編集システムの業務シナリオ ※開発中の組込みソフト	非テストエキスパート	目的:ケース3と比較し、スキルによる有効性の相違を確認する。 実施範囲:数値項目一覧表に、思いつく数値項目を挙げるまでの作業をおこなう。

4.2 検証結果

実プロジェクトを使用した検証の結果は次のとおりであった。

■ケース1~3:テストエキスパートによる検証結果

数値項目一覧表の形で設計内容を確認・分析することで、数値項目に対する設計漏れや矛盾などを検出できたことから、本研究で考案した品質向上策の有効性が確認できた。ただし今回の検証では、業務シナリオを使ってシステム全体を俯瞰したが、数値的な矛盾点の発見には至らなかった。ケースごとに検出された設計漏れなどの件数を表4に示す。

【表4 ケース1~3における検証結果】

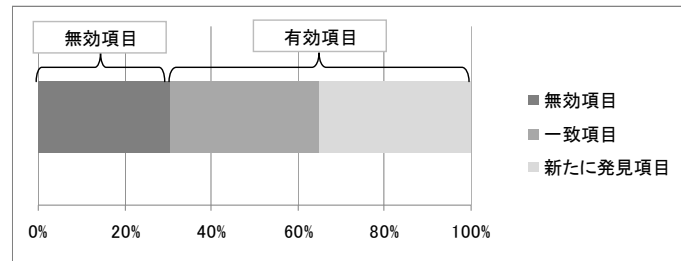
No.	検証結果	ケース1	ケース2	ケース3	説明
1	数値漏れ	0	0	2	機能ごとの数値的な記述が漏れている(単純な数値記述漏れ)
2	値の想定	5	4	4	横断的に見ることによって数値化できない項目および数値の漏れ
3	矛盾点	0	0	0	システムを横断的に見ることにより数値的矛盾が生じている項目
4	設計漏れ	1	1	1	設計書より抜けている仕様
5	気づき	0	0	2	横断的にみることにより気づいた仕様不具合
6	数値の誤記	0	0	1	設計書の数値的誤記
7	検証漏れ	2	2	0	テストフェーズで漏れた不具合。既に運用を開始しているケース1,2のみ
	発見数合計	8	7	10	

- ・ 上限値, 下限値, 最頻値など値の設計漏れの検出: 計3件 (表4のNo.1, No.6)
- ・ 数値項目そのものの設計漏れの検出: 計3件 (表4のNo.4)

- ・ システム全体を横断的に見たときの設計漏れの検出：計 15 件 (表 4 の No.2, No.5)
 - ・ 数値的な矛盾点の検出： 0 件 (表 4 の No.3)
 - ・ 検証漏れ (※) の検出：計 4 件 (表 4 の No.7)
- ※ 検証漏れ 4 件は、既に完了しているプロジェクトで検出されたものである。本来想定できたはずの数値項目の設計が設計レビューで検出されず、検証もできなかったといえる。

■ケース 4：非テストエキスパートによる検証結果

このケースでは、テスト分析・設計スキルを持たない「非テストエキスパート」が、業務シナリオをもとに想定した数値項目をテストエキスパートの作業結果 (ケース 3) と対比した。結果は、図 5 に示すように、抽出した数値項目のうち、約 30%は設計すべき項目ではなく (図 5 「無効項目」)、数値項目表に記録が不要



【図 5 スキルレベルによる有効性】

であったが、約 30%がテストエキスパートと一致 (図 5 「一致項目」)、約 40%はテストエキスパートが抽出できなかった数値項目 (図 5 「新たに発見項目」) であった。このことは、テスト分析・設計スキルをもたない場合でも、数値項目を挙げることで、設計漏れへの気づきを与えられる可能性があることを示唆している。

一方、非テストエキスパートは、「時間」「容量」「回数」「頻度」等をキーワードにすることで数値項目を挙げられたが、具体的な上限値や下限値などの数値は挙げるができなかった。また、自身が記入した数値項目一覧表を使って関連項目の不整合などの分析をおこなうことはできなかった。

上記のことから、この手法は、数値項目を想定する場合により多くの項目を発想させることについて有効であるが、具体的な値を導くにはテストエキスパートの知見が必要であることも示唆している。

5 考察と今後の課題

4 章で述べたように、業務シナリオと新しく考案した「数値項目一覧表」を用いる手法により、従来のレビューだけでは検出できなかった設計漏れを検出でき、a)~d)に示す有効性を確認できた。

- 「時間」「容量」「回数」「頻度」等のキーワードをもとに、本来設計で考慮すべき数値項目を挙げるができる
- 設計書に数値的な設計漏れや不備の可能性のある箇所を発見できる
- 数値項目一覧表中の情報を比較・相互確認することで、設計上の矛盾点を発見できる
- 要件定義の成果物である業務シナリオ (業務フロー) に基づいているため、上流工程でこれらの設計漏れ、不備、矛盾点などを検出できる

上記 a)は 2.3 項の課題①への対策、b)、c)は同項の課題②③④への対策、d)は同じく課題⑤への対策として、それぞれ有効に機能している。すなわち、3.2 項であげた期待効果を充足している。

今回の検証では、すでに完了したプロジェクトについて、本来想定できたはずの数値項目に関する設計漏れおよび検証漏れが検出された。このことから、過去に開発されたソフトウェアに対して潜在する障害リスクを分析するためにも使用できることがわかった。

考案した手法の実施者はテスト分析・設計のスキルを備えた人物を想定していたが、検証の過程において、それらのスキルが不十分であっても「数値項目一覧表」に要件定義書・基本設計書の数値項目を転記

してシステムを横断的・俯瞰的に見える状態を作ることができ、かつ設計漏れの可能性を指摘できることがわかった。これは、テストエキスパートが調達できない場合であっても、開発者がセルフレビューとしてこの手法を実施しても効果が見込めることを示唆している。

なお、本研究では以下の3点について問題点の掘り下げと課題解決策について議論ができておらず、今後の課題ととらえている。

- ① テストエキスパートの分析・評価の結果に対する検証方法
- ② 開発規模や設計の複雑さにより変化する作業量の目安・指標、および費用対効果
- ③ 数値項目一覧表の、テストプロセス（設計・実装）における具体的な活用方法

6 おわりに

本研究では、研究員らの所属する企業での障害事例の分析・考察結果をもとに、上流工程での設計漏れを防止するための課題を設定し、業務シナリオと「数値項目一覧表」を用いる手法を考案して実プロジェクトでの効果測定をおこなった。その結果、基本設計終了までの段階において、従来のレビュー方法では発見できなかった数値項目の設計漏れ・想定漏れ・考慮不足・矛盾を指摘できることがわかった。

今後の課題は、この手法を使った指摘事項に対する検証方法、作業量の指標・目安、テスト設計への具体的な活用方法の検討であると考えている。これらは、本研究成果の延長で課題となるものにとらえており、本研究成果を各社に持ち帰り実践することで、各社に適合した解決策を見出していきたいと考えている。

最後に、本稿に示す成果をあげることができたのは、第5分科会 主査 秋山浩一様、副主査 奥村有紀子様、堀田文明様のご指導の賜物です。深くお礼申し上げます。

<参考文献>

- [1] 近江久美子, 永田敦, 阿部修久, 天野佑太, 伊藤慈朗, 上田克則, 朝永糸子, 森崎一邦: 一般 3-2 要求仕様書におけるテストエンジニアの視点を活かした欠陥検出方法の提案, 第 28 回 ソフトウェア品質シンポジウム 2009
- [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
- [3] リー・コーブランド, 宗雅彦 (翻訳): ソフトウェアのテスト技法, 日経BP社, 2005
- [4] 高橋寿一, 湯本剛: ソフトウェアテスト手法, 技術評論社, 2006
- [5] ソフトウェア・テスト PRESS 編集部: ソフトウェアテスト入門, 技術評論社, 2008
- [6] ジョン・M・キャロル, 郷健太郎 (翻訳): シナリオに基づく設計, 共立出版, 2003
- [7] 森俊樹, 櫻庭紀子, 中野隆司: ソフトウェア品質技術の開発と適用, 東芝レビュー 2006 Vol.61 NO.1, 2006

添付-1 障害事例の原因分類

現状を把握するために、研究員らが所属する企業から 17 事例の障害を集めて分析をおこなった。分析と考察は次の手順で実施した。

- ① 各事例について、発生した障害の詳細情報を得て分析し、どの工程でどのようなことを実施すれば防げたか（検出できたか）を検討し、グループ内で考察した。
- ② 各事例の発生原因をグループ内で議論し、分類した。なおこのとき、発生原因は外部品質特性などを参考に決めた。分類の定義を表 5 に示す。

【表 5 原因分類の定義】

分類	説明
利用想定	顧客の具体的な利用方法について、想定していなかった使い方が発生した問題
業務知識	関連機器やメディア等の、特に新製品の規格に関する知識が不足していた
限界値	見積もったデータ容量を超える、少なすぎるなど、限界値の見積りが不適切
入力値	入力チェックの範囲設定が不適切で、想定外の値のインプットを検出できなかった
標準/規格	業界の標準的事項に則った仕様にしていなかった
互換性	旧システム(旧機種)の機能との互換性が失われた
技術(PG/その他)	プログラミング標準の認識不足、プログラミング技術の未熟による問題
依存性	複数のシステム間において、互いの仕様に依存する事項が把握されていなかった
保守性	関連システムの、非公開のデータを取得して利用したことによる問題
条件抽出不足	関連システムから発信される信号のパターン整理が不足したことによる問題

- ③ 各事例と発生原因のマトリクスを表 6 に示す。○は主たる原因分類、△は二次的な原因分類である。本文中の図 2 は、主たる原因分類により集計した結果である。

【表 6 事例と原因分類のマトリクス】

分類	組込系の事例					業務システム的事例											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
利用想定		○				△				△				○		○	○
業務知識	○	△	○	○										△			
入力値									○	○			○				
限界値				△		○				○							
技術(PG/その他)							○	○									
標準/規格					○						△		△				
条件抽出不足															○		
依存性												○					
互換性	△		△													△	
保守性												△					

添付-2 設計漏れ検出方法の詳細 (1/2)

a) 詳細手順

本文3章で記述した設計漏れ検出方法の詳細な手順は次のとおりである。

- ① 数値項目一覧表に業務シナリオの構成要素（機能・データベース・メディア・デバイスなど）ごとに、設計すべき数値項目を抽出して、その数値項目に関する「上限値」「下限値」「最頻値」を記入する。使用するフォーマットを表7に示す。
- ② 設計すべき数値項目は、「時間」「容量」「回数」「頻度」等をキーワードにして具体的な項目を決める。
- ③ 「上限値」「下限値」「最頻値」は、要件定義・基本設計から抽出して記入する（両方にあれば基本設計のものを探る）。どちらにも定義がなければ、想定値を記入する。
- ④ 繰り返し操作のような一連の操作・処理などがあれば、その単位でも①②③をおこなう。
- ⑤ 数値項目一覧表を用いて、下記の視点で数値項目とその値に関する確認・分析をおこなう。数値項目一覧表には業務シナリオの構成要素ごとの数値に関連する設計情報（および想定値）がまとまっているため、個々の機能、機能間、システム全体、それぞれの視点で確認する。
 - a) 数値が入らない空欄部分は、設計漏れが疑われる。また想定値が記入されている箇所は、もともと設計情報がないことから、設計において考慮すべき事項である可能性がある。
 - b) 一連の処理や操作の観点でチェックをおこない、そこに含まれる数値項目の間に矛盾や不整合がないかどうかを確認する。
 - c) 処理や操作をまたがって、関連する数値項目間で矛盾や不整合がないかどうかを確認する。
- ⑥ ⑤の分析結果を指摘事項としてまとめて開発チームへフィードバックする。
- ⑦ 開発チームでは、これらの指摘を解消して設計書に記載するなどの対応をする。⑤と⑥を繰り返すことで、抽出された数値項目とその値の妥当性を検証するとともに設計漏れを防止する。

b) 数値項目一覧表のフォーマット（入力例含む）

提案する数値項目一覧表は、以下のようなフォーマットである。

【表7 数値項目一覧表（例）】

通し番号	照合番号	振舞い	構成要素 対象の機能/処理/データ	キーワード	数値項目	値の出元 要件/設計/想定	上限	下限	最頻値	単位	値の規模 ※想定の場合には必須 情報を取得する必要がないため	備考	検証タイミング
1	1	サイトへアクセス	ログインメニュー表示	時間	レスポンスタイム	想定	1	-		秒			システムテスト
2	1	サイトへアクセス	ログインメニュー表示	回数	ページ表示回数	想定	1	1	1	回	ログイン前に表示するのみ		
3	2	ログイン	ログイン処理	時間	レスポンスタイム	要件	5	-		秒			システムテスト
4	2	ログイン	ログイン処理	回数	ページ表示回数	想定	1	1	1	回	ログインするのは1回のみ		
5	2	ログイン	ログイン処理	その他	アクセス数	想定	1250	1	3	数/秒	HPのアクセスが多い時間帯で135万アクセス/3h		（運用開始後に集計して確認）
6	3	サービス検索	キーワード検索	時間	レスポンスタイム	要件	20	5		秒			システムテスト
7	3	サービス検索	キーワード検索	容量	文字数	設計	30	1		byte			単体テスト
8	3	サービス検索	携帯コンテンツDB	容量	1年後の容量	設計	5.87			GB			（運用開始後に集計して確認）
9	3	サービス検索	携帯コンテンツDB	容量	3年後の容量	設計	11.26			GB			（運用開始後に集計して確認）
10	4	（繰り返し操作）		回数	繰り返し操作回数	想定	3	1	1	回	キーワードの絞込み		
11	5	サービス詳細表示	サービス詳細表示	時間	レスポンスタイム	要件	20	5		秒			システムテスト
12	6	サービス利用	携帯会員登録表示	時間	レスポンスタイム	要件	5	-		秒			システムテスト
13	7	サービス利用	電話申込表示	時間	レスポンスタイム	要件	20	5		秒			システムテスト
14	8		その他(全体)	容量	URL長	設計	300	-		小文字			結合テスト
15	8		その他(全体)	時間	セッションタイムアウト	設計	10	-		分			システムテスト
16	8		その他(全体)	時間	稼働時間	要件	24	-		時間			（運用開始後に監視して確認）
17	9	サービス検索	携帯コンテンツDB	容量	データ量	設計・想定	296			byte	会員番号(16)所属企業CD(6)契約企業CD(6)対象デバイス(1)サービス名/施設名(256)取得行数(10)取得開始行(1)		結合テスト
18	10	サービス検索	携帯コンテンツDB	容量	データ量	設計・想定	55	0		KB	1画面10件まで表示(サービス名(256)キャッシュコピー(30)都道府県名(8)カテゴリ名(256))・該当なしの場合有り		結合テスト
19	11	サービス詳細表示	携帯コンテンツDB	容量	データ量	設計・想定	50	50	50	byte	コンテンツID(50)		結合テスト
20	12	サービス詳細表示	携帯コンテンツDB	容量	データ量	設計・想定	155			KB	1件表示(サービス名(256)キャッシュコピー(750)都道府県名(8)カテゴリ名(256)サービス名(256)サービスID(20))		結合テスト
21	13	サービス利用	携帯コンテンツDB	容量	データ量	設計・想定	20			byte	サービスID(20)		結合テスト
22	14	サービス利用	携帯コンテンツDB	容量	データ量	設計・想定				byte	施設名・サービス名(256)※以下繰り返し有り(会員特典タイトル)会員特典文字列(10)会員特典文字列(20)会員特典文字列(30)会員特典備考		結合テスト

1 要件にも設計にも値がある場合は設計値を優先

添付-2 設計漏れ検出方法の詳細 (2/2)

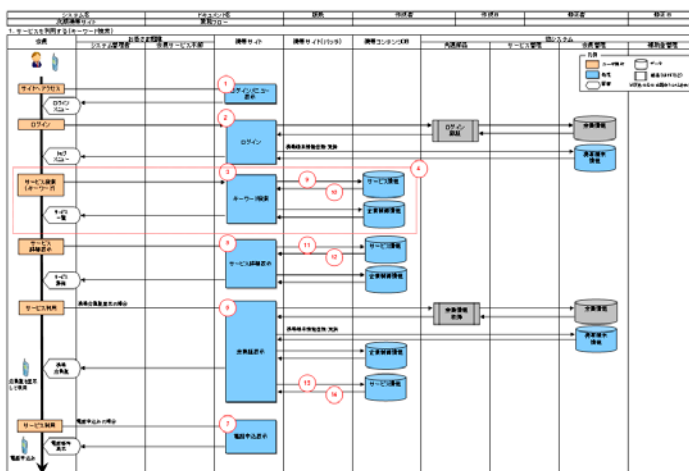
表の列の意味は、それぞれ次のとおり。

- ・ 通し番号…表の中での通し番号で、識別しやすくするために使用する。
- ・ 照合番号…業務シナリオに付番した照合番号で、業務シナリオと一覧表の内容を対応付ける番号として使用する。
- ・ 振舞い…業務シナリオにおけるアクターからみた操作で、識別しやすくするために使用する。
- ・ 構成要素…業務シナリオから抽出した構成要素（機能、処理、データベース、デバイス、メディアなど）を記入する。
- ・ キーワード…もととなったキーワードを記入する。基本的には、「時間」「容量」「回数」「頻度」から選択するが、この他にプロジェクトやシステムの特성에応じて追加してもよい。
- ・ 数値項目…キーワードをもとに抽出した、本来設計対象とすべき数値項目の名称を記入する。
- ・ 値の出元…要件値（要件定義から得たもの）、設計値（基本設計から得たもの）、想定値（他の要件や設計情報や類似するシステムの実績値を参考に、想定により得たもの）のいずれかを記入する。
- ・ 上限値・下限値・最頻値…上限値，下限値，およびもっとも発生頻度の高い値を記入する。
- ・ 単位…数値項目に対する単位を記入する。
- ・ 値の根拠…その値の根拠を記入する。想定値の場合には必須である。
- ・ 備考…備考や特記事項があれば記入する。
- ・ 検証タイミング…記入不要である（サンプルとして使用したプロジェクトが完了したものであったため、プロジェクト中に検証したものを備忘録として記録している）。

なお、値の出元は1行に1列しか設けていないが、場合によっては上限値と下限値は設計された値、最頻値は想定値といったような状況も考えられる。その場合には、上限値・下限値・最頻値のセルを色分けすることで区別するなどの工夫をすると良い。

c) 業務シナリオ（業務フロー）のフォーマット（入力例含む）

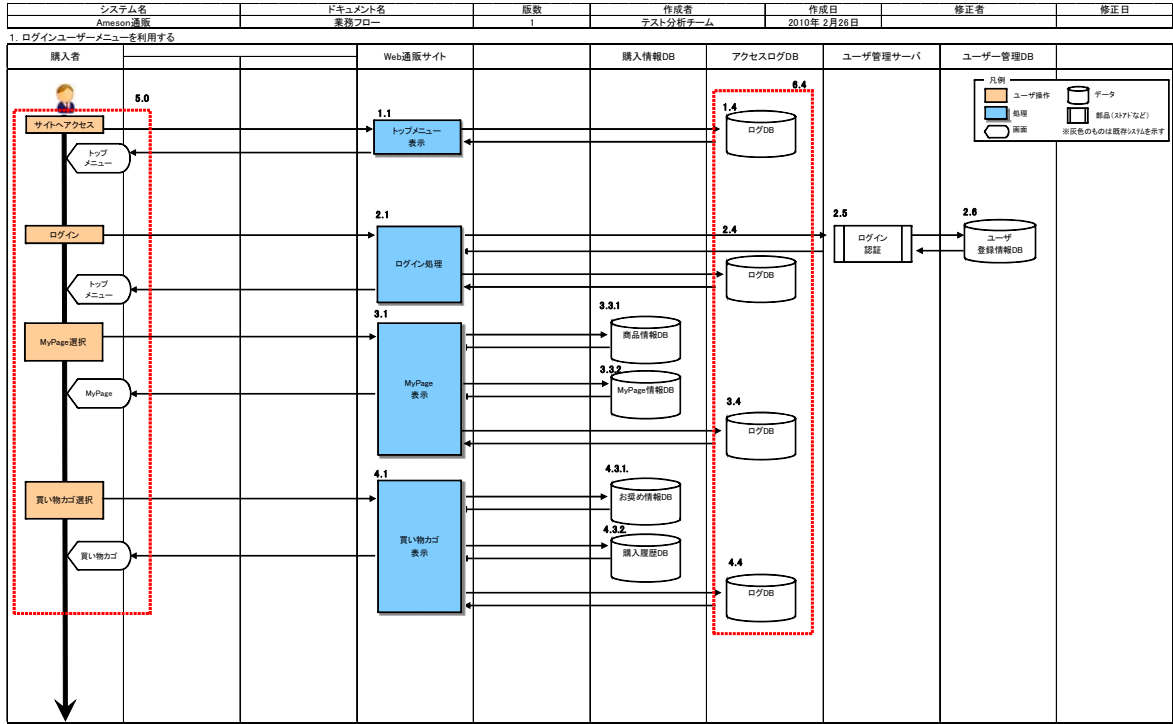
本研究で使用した業務シナリオ（業務フロー）のイメージは次のようなものである。既に構成要素に照合番号が付与されている。また、繰り返し操作の部分は枠で囲んだ上で照合番号を付与している。



【図 6 業務シナリオ (例)】

添付3 作成例 1

a) 業務シナリオ



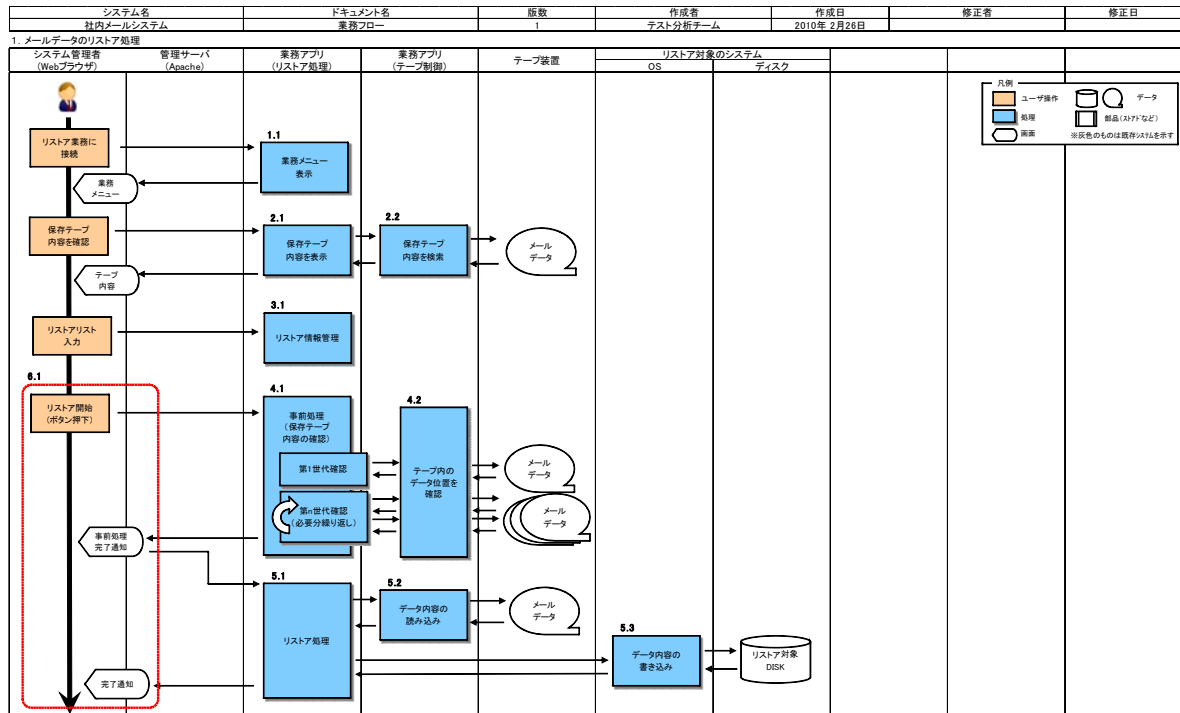
b) 数値項目一覧表

通し番号	照合番号	振替	構築要素 対象の機能/処理/データ	キーワード	数値項目	値の出元 要件/設計/想定	上限	下限	最頻値	値の根拠 ※設定の場合必須	備考
1	1.1	サイトへアクセス	トップメニュー表示	その他	アクセスユーザ数 (同時アクセス)	要件定義書	3万	0	300	規模値は上限の1%で設定	
2	1.1	サイトへアクセス	トップメニュー表示	その他	アクセスユーザ数 (1サーバ/時間)	要件定義書	10万	0	1万		
3	1.1	サイトへアクセス	トップメニュー表示	その他	サーバ/回数	要件定義書	5	1	-		
4	1.1	サイトへアクセス	トップメニュー表示	時間	レスポンス時間	要件定義書	3s	0s	1s		
5	1.4	訪問者ログ記録	ログDB	容量	ログ書込みサイズ (1メッセージ/1ユーザ)	設計書	240	80	80		
6	2.1	ログイン	ログイン処理	時間	レスポンス時間	想定	3s	0s	1s	要件定義書の範囲内で設定	
7	2.4	ログイン情報記録	ログDB	容量	ログ書込みサイズ (1メッセージ/1ユーザ)	設計書	240	80	80		
8	2.5	ログイン	ログイン認証	時間	レスポンス時間	想定	3s	0s	1s	要件定義書の範囲内で設定	
9	2.1	ログイン	ログイン認証	回数	ログイン失敗の許容回数	設計書	3	1	1		
10	2.1	ログイン	ログイン認証	その他	多重ログイン数	想定	1	-	1	多重ログインを禁止	
11	2.6	ログイン	ユーザ登録情報DB	時間	レスポンス時間 (DB参照)	想定	0s	0s	0s	性能要件として1s以内を想定	
12	2.6	ログイン	ユーザ登録情報DB	時間	レスポンス時間 (DB書込み)	想定	0s	0s	0s	性能要件として1s以内を想定	
13	2.6	ログイン	ユーザ登録情報DB	時間	DATA書込みサイズ (DATA/1ユーザ)	設計書	240	80	80		
14	2.6	ログイン	ユーザ登録情報DB	容量	DB容量	設計書	600GB	30GB	-		(*)
15	2.6	ログイン	ユーザ登録情報DB	容量	HDD容量	想定	600GB	300GB	-	一般的なDISKを想定	(*)
16	3.1	MyPage選択	MyPage表示	時間	レスポンス時間	想定	3s	0s	1s	要件定義書の範囲内で設定	
17	3.3.1	MyPage選択	商品情報DB	時間	レスポンス時間 (DB参照)	想定	0s	0s	0s	性能要件として1s以内を想定	
18	3.3.1	MyPage選択	商品情報DB	時間	レスポンス時間 (DB書込み)	想定	0s	0s	0s	性能要件として1s以内を想定	
19	3.3.1	MyPage選択	商品情報DB	容量	DATA書込みサイズ (DATA/1ユーザ)	設計書	240	80	80		
20	3.3.1	MyPage選択	商品情報DB	容量	DB容量	設計書	600GB	30GB	-		(*)
21	3.3.1	MyPage選択	商品情報DB	容量	HDD容量	想定	600GB	300GB	-	一般的なDISKを想定	(*)
22	3.3.2	MyPage選択	MyPage情報DB	時間	レスポンス時間 (DB参照)	想定	0s	0s	0s	性能要件として1s以内を想定	
23	3.3.2	MyPage選択	MyPage情報DB	時間	レスポンス時間 (DB書込み)	想定	0s	0s	0s	性能要件として1s以内を想定	
24	3.3.2	MyPage選択	MyPage情報DB	時間	DATA書込みサイズ (DATA/1ユーザ)	設計書	240	80	80		
25	3.3.2	MyPage選択	MyPage情報DB	容量	DB容量	設計書	600GB	30GB	-		(*)
26	3.3.2	MyPage選択	MyPage情報DB	容量	HDD容量	想定	600GB	300GB	-	一般的なDISKを想定	(*)
27	3.4	MyPage選択	ログDB	容量	ログ書込みサイズ (1メッセージ/1ユーザ)	設計書	240	80	80		
28	4.1	買い物カゴ選択	買い物カゴ表示	時間	レスポンス時間	設計書	3s	0s	1s		
29	4.3.1	買い物カゴ選択	お探め情報DB	時間	レスポンス時間 (DB参照)	想定	0s	0s	0s	性能要件として1s以内を想定	
30	4.3.1	買い物カゴ選択	お探め情報DB	時間	レスポンス時間 (DB書込み)	想定	0s	0s	0s	性能要件として1s以内を想定	
31	4.3.1	買い物カゴ選択	お探め情報DB	容量	DATA書込みサイズ (DATA/1ユーザ)	設計書	240	80	80		
32	4.3.1	買い物カゴ選択	お探め情報DB	容量	DB容量	設計書	600GB	30GB	-		(*)
33	4.3.1	買い物カゴ選択	お探め情報DB	容量	HDD容量	想定	600GB	300GB	-	一般的なDISKを想定	(*)
34	4.3.2	買い物カゴ選択	購入履歴DB	時間	レスポンス時間 (DB参照)	想定	0s	0s	0s	性能要件として1s以内を想定	
35	4.3.2	買い物カゴ選択	購入履歴DB	時間	レスポンス時間 (DB書込み)	想定	0s	0s	0s	性能要件として1s以内を想定	
36	4.3.2	買い物カゴ選択	購入履歴DB	容量	DATA書込みサイズ (DATA/1ユーザ)	設計書	240	80	80		
37	4.3.2	買い物カゴ選択	購入履歴DB	容量	DB容量	設計書	600GB	30GB	-		(*)
38	4.3.2	買い物カゴ選択	購入履歴DB	容量	HDD容量	想定	600GB	300GB	-	一般的なDISKを想定	(*)
39	4.4	買い物カゴ選択	ログDB	容量	ログ書込みサイズ (1メッセージ/1ユーザ)	設計書	240	80	80		
40	5.0	購入操作	購入操作	時間	レスポンス時間	要件定義書	3s	0s	1s		
41	5.0	購入操作	購入操作	時間	セッションタイムアウト時間	設計書	300s	0	180s		
42	6.4	アクセスログDB	ログDB	容量	DB容量	設計書	600GB	30GB	-		(*)
43	6.4	アクセスログDB	ログDB	容量	HDD容量	想定	600GB	300GB	-	一般的なDISKを想定	(*)
44	6.4	アクセスログDB	ログDB	その他	同時書き込み数	想定	12万	0	1200	同時アクセス数×4処理	(*)
45	6.4	アクセスログDB	ログDB	時間	書き込み時間 (1メッセージあたり)	想定	0s	0s	0s	性能要件として1s以内を想定	

(*) [指摘] 将来を想定した利用者総数、取扱商品総数が設定されていない、運用中にDB/バンクの可能性あり。
 (**) [指摘] DBサイズの根拠がないため、DBを配置するDISK容量が不足する可能性あり。
 (***) [確認] 各処理のログ採取DBが同一のため、ログの同時書き込み時に性能劣化がないか確認。

添付-4 作成例 2

a) 業務シナリオ



b) 数値項目一覧表

通し番号	照合番号	振舞い	構成要素 対象の機能/処理/データ	キーワード	数値項目	値の出元 要件/設計/想定	上限	下限	最頻値	値の根拠 ※想定の場合は必須	備考
1	1.1	業務接続	業務表示	頻度	ログイン回数	想定	1	1	1	多重ログイン禁止	(※1)
2	1.1	業務接続	業務表示	その他	同時接続数	要件定義書	10	1	1		システム管理者数(※2)
3	2.1	テープ内容確認	テープ内容表示	時間	DATアクセス時間	設計書	193m	1s	3s		
4	2.1	テープ内容確認	テープ内容表示	容量	DAT容量	設計書	80GB	1MB	100MB		
5	2.1	テープ内容確認	テープ内容表示	その他	同時アクセス数	設計書	1	1	1		(※2)
6	2.2	テープ内容確認	テープ内容検索	時間	DATアクセス時間	設計書	193m	1s	3s		
7	2.2	テープ内容確認	テープ内容検索	容量	DAT容量	設計書	80GB	1MB	100MB		
8	2.2	テープ内容確認	テープ内容検索	その他	同時アクセス数	設計書	1	1	1		(※2)
9	3.1	リストアップ情報入力	リストアップ情報管理	回数	設定回数	設計書		1	1		
10	4.1	リストアップ処理	事前処理	時間	処理時間	想定	60s	1s	20s	世代分の累計時間	
11	4.1	リストアップ処理	事前処理	容量	管理情報容量	設計書	1GB	10MB	100MB		
12	4.1	リストアップ処理	事前処理	回数	世代分の処理回数	設計書	30	1	5		
13	4.2	リストアップ処理	データ位置確認	時間	位置確認時間	想定	60s	1s	20s	世代分の累計時間	
14	4.2	リストアップ処理	データ位置確認	容量	管理情報容量	設計書	1GB	10MB	100MB		
15	4.2	リストアップ処理	データ位置確認	回数	世代分の処理回数	設計書	30	1	5		
16	5.1	リストアップ処理	リストアップ処理	時間	処理時間	設計書	193m	1s	24m	データ容量から算出	(※3)
17	5.1	リストアップ処理	リストアップ処理	時間	応答時間	想定	300s	0s	180s	Apache Timeout時間	(※3)
18	5.1	リストアップ処理	リストアップ処理	容量	データ容量	設計書	80GB	1MB	10GB		
19	5.2	リストアップ処理	データ読み込み	時間	DATアクセス時間	設計書	193m	1s	24m		(※3)
20	5.2	リストアップ処理	データ読み込み	容量	DAT容量	設計書	80GB	1MB	10GB	DAT16(記憶容量80GB)	
21	5.3	リストアップ処理	データ書き込み	時間	DATアクセス時間	設計書	193m	1s	24m		(※3)
22	5.3	リストアップ処理	データ書き込み	容量	DAT容量	設計書	80GB	1MB	10GB		
23	6.1	リストアップ処理	リストアップ処理	時間	処理時間	要件定義書	60m	1m	20m		(※4)
24	6.1	リストアップ処理	リストアップ処理	容量	DAT容量	設計書	80GB	1MB	10GB		

(※1) [指摘] Webブラウザからの多重ログインを禁止されているか。

(※2) [確認] システム管理者からの同時接続は可能でもテープアクセスは同時にできない。同時指示の場合のキュー制御方式を確認。

(※3) [指摘] リストア処理のデータ読み取り時間とWebサーバ(Apache)のデフォルトタイムアウト値に大きな差がある。

(※4) [指摘] 要件定義書のリストアップ処理時間(上限)とリストアップ時間(上限)に大きな差がある。