

第5分科会（Aグループ）

現場で使える品質評価手法の研究

Quality evaluation technique that can be used in practical.

主査	保田 勝通	（つくば国際大学）
副主査	西 康晴	（電気通信大学） 高橋 寿一（ソニー株式会社）
リーダー	宇野 也寸志	（ダイキン工業株式会社）
サブリーダー	萩原 俊幸	（アンリツエンジニアリング株式会社）
研究員	町田 欣史	（株式会社NTTデータ）
	白山 新一	（富士通北陸システムズ株式会社）
	伊藤 史伸	（株式会社アルゴ21）
	相原 尚子	（サイボウズ株式会社）

（敬称略 順不同）

概要

ソフトウェア開発に従事する者の大半は「テスト漏れが多く、客先で不具合が発生する」、「仕様検討漏れが多く、手戻りが発生する」という共通の課題を抱えている。

そこで我々研究員は、ISO/IEC 9126 で定義された品質特性および品質副特性を利用してこの共通の課題を解決しようとしたが、これらの特性は難解かつ抽象的であり、現場で使うには困難であることが分かった。

研究開始当初、我々には、品質特性および品質副特性の理解と、その特性を利用したテスト技法の具体化という2つの課題があった。その課題に対して我々は、特性のインディケータ（測定指標）の理解を通して特性を推測する方法により前者の課題を解決し、チェックリストの作成という方法により後者の課題を解決した。

こうして作成したチェックリストは、V字プロセス（SLCP[ISO/IEC12207:JIS X0160]に準拠）の各工程別に成果物の評価ができること、チェックすべき観点をレビューとテスト実施時に分離していることなど、現場で活用するための工夫が施されている。さらに本研究では、チェックリストの作成に留まらず、実際のプロジェクトにおける成果物の評価にチェックリストを使用し、その有効性の評価と考察も行った。

Abstract

Many software developers have shared problems that failures occur in client's office and that after releasing repairment is required because of incomplete specifications.

To solve this issue, we researched quality characteristics and quality subcharacteristics, both of them are defined by ISO/IEC 9126. However, we realized that both of them are so complicated and abstract that it's difficult to use these characteristics at working sites.

After reviewing these facts, in early stage of our research, we set our goal of two topics: understanding both quality characteristics and quality subcharacteristics, then, with these characteristics, materializing the test method. For achieving our goal, we

dealt with the former goal by building the method how to speculate characteristics as the result of analyzing indicators of characteristics. Next, to accomplish the latter, we established the new checklist.

We improve the check-sheet: such that V-model (SLCP[ISO/IEC12207 : JIS X0160]) can be evaluated each stage of development, and the check-sheet can be separated by the testing and the reviewing point of view. Those will helps a lot toward practical engineers. In addition to creating checklist, we have evaluated and considered it how efficient when checklist is used to evaluate outputs in actual software development projects.

1. はじめに

ソフトウェアの品質は曖昧で掴み所の無いともものと考えられやすいが、その品質モデルとして1991年にISO/IEC 9126が発行され、日本では1994年にJIS X0129がJIS化された。この品質モデルでは、ソフトウェアをユーザの視点で外部から評価するための6種類の品質特性とそれを詳細化した21種類の品質副特性を定めている。さらに、日本規格協会情報技術標準化センター(INSTAC)の調査研究では、ソフトウェアを開発者の視点で内部から評価するための40種類の内部品質特性を定めている。

しかし、このような世界共通の品質モデルが存在するにも関わらず、ソフトウェア開発現場では、この品質モデルへの積極的な取り組みは見受けられないという問題がある。

この問題を解決するために、我々のグループではこの品質モデルを理解するとともに、開発現場で現実的で有効に使える品質モデルの活用方法を研究することとした。こうして「現場で使える品質評価手法の研究」というテーマを決定した。

2. 活動目標

我々のグループは“ソフトウェアの高品質を実現するエンジニアリングアプローチ”を実現するにあたり、本研究の成果を各自が自社に持ち帰って利用することができる評価手法の研究を活動目標の前提とした。次にグループ内で、現場のテストにおいてテスト項目漏れが多いという意見が多かったことに着目し、下記の課題に対してどう解決してゆくかを研究することとした。

- ・ 暗黙の了解が原因となる不具合が多い。
- ・ 仕様書と実施するテスト項目が同時にレビューできない。
- ・ 上流工程において、仕様が明確にできない。
- ・ トラブル事例のフィードバックができていない。

これらの課題を解決する方法として、世界共通の品質モデルであるISO/IEC 9126を利用することを考えたが、ISO/IEC 9126で定められている品質特性・副特性とINSTACにより定められた内部品質特性の定義は難解かつ抽象的であるため、そのまま現場で適用するのは難しいことが分かった。

そこで我々研究員は、体系的な品質モデルとして定義、設計そして管理すべき項目を明確にすることにより、テスト実施の土台を確立することを活動目標とした。今年度の活動目標は、品質評価の具体案が提案されている「ソフトウェア品質評価ガイドブック」[1]を参考に、まず品質モデルを理解し、続いて開発の現場において前記の課題を解決するため

のチェックリストを作成し、最後にそのチェックリストを実際を使ってレビューした結果を評価、考察することとした。

3. 活動内容

実施した研究活動の内容を表1に示す。

表1 活動内容

No.	作業
1	ISO/IEC 9126 で定められた品質特性・副特性、INSTAC で定められた内部品質特性にどのようなものがあるかを理解した。
2	品質特性・副特性、内部品質特性を測定するためのインディケータの種類とそれぞれの意味を、参考文献[1]を基に調査し、理解した。
3	類似するインディケータを集約することにより、品質特性・副特性、内部品質特性を、現場でわかる言葉に置き換えた。
4	品質特性・副特性、内部品質特性に関するインディケータの集約結果からチェックリストを作成した。
5	現場で使うことを意識して、チェックリストを要件定義、基本設計、詳細設計の各工程に展開した。同時にチェックリストの各項目を、各上流工程のレビューで使用するものなのか、それぞれに対応するテスト工程で実施するテスト項目のチェックに使用するものなのかの分類も行った。
6	作成したチェックリストを用いて、実プロジェクトの成果物のレビューに使用し、評価を行った。

今回作成したチェックリストは現場で使う事を前提としているため、「テスト密度」や「障害密度」など特徴的で実践的と思われるインディケータを選んで採用している。すなわち、実践的でないインディケータを省略しているため、体系的にまとめられた品質モデルの網羅性が失われている。したがって、このチェックリストだけではテストを網羅したとはいえないので注意が必要である。なお、このチェックリストはソフトウェアの品質測定が目的でなく、仕様書やテスト抜けを見つけることが目的であるため、出来上がったソフトウェアの評価に使うものではない。

4. 活動成果

4.1. チェックリストの作成

我々が作成したチェックリストはインディケータを基にしているが、チェックリストにインディケータを書くと、現場で使うときに「チェックリストに書かれたインディケータを測定すればテスト項目を網羅する」という誤解を招く可能性があることから、チェック項目と品質特性・品質副特性、内部品質特性の記載のみとした。表2にチェックリストの例を示す。なお、今回作成したすべてのチェックリストを付録1に示す。

表2 チェックリストの例

No.	チェック項目	特性	副特性
1	要求分析の内容を過不足なく展開しているか。	機能性	標準適合性
		内部特性	完全性
			追跡可能性
2	仕様書を階層的に表現しているか。	内部特性	階層性
3	仕様改定率はどのくらいか計算したか。	機能性	合目的性
4	計算処理が存在する場合、その有効桁数や、まるめ処理に関して正しく設計しているか。	機能性	正確性
		内部特性	計算正確性

4.2. チェックリストの活用成果

4.2.1. チェックリストの適用箇所

チェックリストの活用成果を評価するにあたり、評価対象のシステムのテスト時間に制約があったので、システムテスト用チェックリスト（付録1の付表3、付表4）を用いたテスト項目漏れの検出に限定して適用し、評価した。

4.2.2. 適用したシステムの説明

チェックリストを適用したシステムは、ユーザから制御機器に対して監視・制御を行うシステムである(図1)。システム中には、ユーザとのインタフェースをつかさどる画面表示(GUI)部、内部処理部、制御機器とのインタフェースをつかさどる I/F(Interface)部から構成する。

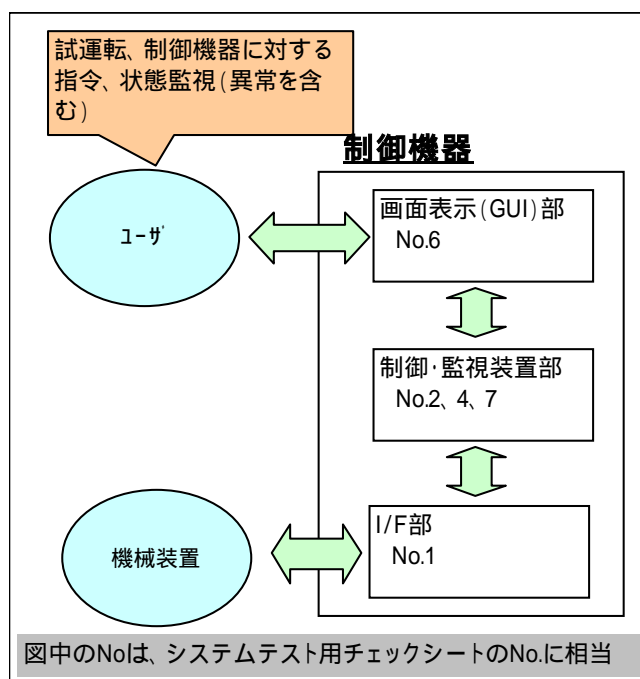


図1 システムの構成

ユーザは、GUI部から制御機器の現在状態（運転、停止、異常の有無等）の監視や制御

(運転・停止)をする。内部処理部では、GUI部を通じてユーザからの指令を判定し、どの制御機器に対して指令するかを判断処理したり、逆に制御機器の状態をGUI部へ伝達したりする。

4.2.3. チェックリストの適用手順

4.2.3.1. チェックリストの使用箇所の特定

今回は、システムテスト用のチェックリストの適用に限定することを4.2.1で述べたが、そのチェック項目の中で使用しなかった項目とその理由を下記に示す。

項目 No.4： 適用時期が性能・負荷テストの検討前であったため、性能・負荷テストに相当する項目を使用せず。

項目 No.5： 今回検討した対象システムの特性上、セキュリティテストに相当する項目を使用せず。

項目 No.8： この項目はシステムテストの完了基準に使用する項目であるため、テスト項目のチェックには使用せず。

4.2.3.2. チェックリストに対する補足説明の追記

チェックリストの項目だけではテスト項目の抽出を十分に行うことは難しいと考え、各項目に補足説明を付けた。その結果のチェックリストを、表3に示す。

表3 補足説明を加えたシステムテスト用チェックリスト(適用項目のみ)

No.	チェック項目	テスト項目抽出のための補足
1	開発対象のシステムに関連する法規、業界標準を遵守しているか。	標準化(業界標準、社内標準)されている機能について、標準化に準拠していることの検証を行う
2	計算処理について有効桁数や、まるめ処理の結果が正しいか。	<ul style="list-style-type: none"> 有効数字検証テスト(演算途中で桁落ちしていないか。回り込み処理の有無について) まるめ処理検証テスト
3	対応するハードウェア構成、ソフトウェア構成、運用形態などの必要最低限の性能を満たしているか。また、これらの前提が変更になった場合のデグレードチェック結果が正しいか。	データ形式、文字・制御コード、インタフェース・フォーマットの合致度の検証を目的としたテスト
6	ユーザインタフェースについて、メッセージ、操作、レイアウト等が標準化したものになっているか。また、誤入力、誤操作の対応ができていないか。デフォルト値によるユーザ入力省略が可能か。	<ul style="list-style-type: none"> グレーアウト制御 デフォルト表示確認テスト 省力化(反復操作)確認テスト 状態進捗表示機能テスト
7	予想される障害の発生確率を測定したか。また、障害発生時の障害検知、障害記録、原因追求、対応・復旧までのプロセスについての要件を満たしているか。	<ul style="list-style-type: none"> 異常操作(フルブーフ)テスト 適正(コマンド)エラーチェック確認テスト データバックアップ確認テスト システム異常対処確認テスト

4.2.3.3. システムテスト項目の抽出

システムテスト項目は、まず機能仕様書を基に抽出し、次に上記のチェック項目を基に検討漏れが無いことを確認し、必要であればテスト項目に追加するという順序で実施した。その際、チェックリストを用いてテスト項目をチェックした結果、機能仕様書に記載されていないがシステムテストが必要であると判断した項目については積極的にテスト項目を追加した。なお、合格基準について開発担当者から明確な回答が得られなかった項目は、合格・不合格が曖昧になり、テストする価値がないと判断したため、テスト項目を追加し

なかった。

4.3. 適用結果

図1で示したシステムに対してチェックリストを適用した結果、77のテスト項目を追加することになった(表4)。それらは、(1)機能仕様書で明記しているがテスト項目に展開していなかった項目、(2)機能仕様書上の表現が明確でない項目の2つに大きく分けられた。

表4 適用結果一覧

	項目数(件)	比率(%)
全テスト項目数	1265	-
チェックリスト利用により抽出された項目数	77	6.1
<内訳>(表3のチェック項目別)		
No.1 標準適合	0	0.0
No.2 計算処理	9	11.7
No.3 構成	28	36.4
No.6 GUI	27	35.1
No.7 障害対応	13	16.9

5. 活動評価

5.1. 実プロジェクトにおける適合の可否

テスト設計の経験が少ない者、すなわち実際にソフトウェアが動作しているイメージができない者に本チェックリストを用いてもらったため、テスト項目漏れを発見することが困難であった。したがって、現状のチェックリストで我々が期待している効果を出すためには、テスト設計の経験が豊富で、不具合事例が頭にイメージできる者である必要があることが分かった。

一方、テスト設計の経験が少ない者がテスト項目漏れチェックのヒントとして本チェックリストを利用するためには、更なるブレイクダウン(各チェック項目の意味するところの解説、適用事例等)が必要であると考えられる。

5.2. 実際に使用してみて良かった所

5.2.1. 機能仕様書の品質向上

GUIテストに関して本チェックリストを適用した結果、機能仕様書の漏れを検出でき、検出した漏れを容易にテスト項目へ展開できるという効果が得られた。なぜなら、本チェックリストを適用して見つかった機能仕様書の漏れの多くは開発担当者が認識している機能であり、明確な回答が得られるからである。

また、今回、機能仕様書を基に本チェックリストを適用しテスト項目の検討したところ、仕様書に記載していないとテスト項目が抽出できない機能が見つかり、その追記による仕様書の品質向上の効果もあった。具体的にはデフォルト表示の仕様、グレーアウトすべきタイミング等である。

5.2.2. 上流工程で仕様を検討する必要性の認識

演算処理や障害対応に関するテスト項目は、十分な展開が出来たとは言い難い。なぜなら、要求仕様が不明確であり、テスト項目の合格基準を定義出来なかったからである。この点について考察すると、開発担当者は機能要件のような意識して明確にしないと設計で

きないものについては明確にするが、非機能要件のような不明確でも設計できてしまうものについては意識しないで設計しているという問題があることがわかる。

以上のことからわかるように、現状のシステムテストでは非機能要件が不明確な状態でテストプロセスを実施することが多く、最悪の場合は要件が不明確な項目に関してテストせずに市場に出すケースもある。今後はこのような低品質なシステムを市場に出さないためにも、要求分析のような上流工程で非機能要件を明確にし、検証すべき内容についてステークホルダを巻き込んで定義しておく必要がある。

検討したチェックリストは、非機能要件を明確にする手段として活用できるであろう。

5.3. チェックリストの網羅性について

今回、品質特性を「要求分析用」など6つの工程用のチェックリストという形にまとめたが、そのチェックリストについて客観的評価が必要であると考え、評価検証した。

評価内容は、要求分析用チェックリストのまとめ方、品質特性の抜け落ち、他の工程でチェックすべき項目の混在などについてである。評価方法は、今回作成した要求分析用チェックリストと、マイヤーズの「ソフトウェア・テストの技法」(参考文献[5])に記載されているシステムテストの種類とを比較し、包括度合いを見ることとした。なお、システムテストの種類は文献により異なることをここに記載しておく。

付録2の付表11に比較した結果を示す。マイヤーズのシステムテストの種類はそれぞれ本チェックリストの項目のいずれかに該当しているため、概ねテスト範囲をカバーしていると評価できる。つまり、システムテスト用のチェックリストとしては有効な形にまとめられているのではないかと考える。

5.4. 4つの課題の評価

今回の研究において、活動目標に示した4つの課題について課題解決できたかどうかを評価した。以下にその結果を示す。

5.4.1. 課題1 暗黙の了解が原因となる不具合が多い。

システムの応答待ち時間を例に挙げると、具体的な数値を示さないだけでなく、待っている人が遅いと感じない時間など定性的な要件を意識しながら仕様書に記載を省略する開発担当者が多い。特に非機能要件については仕様への記載の省略が多く見られる。このことから、本課題の「暗黙の了解」を非機能要件と捉えると、要求分析において本チェックリストが活用できている。したがって非機能要件を機能仕様として定義するための道具として活用することで課題1を解決している。

5.4.2. 課題2 仕様書と実施するテスト項目が同時にレビューできない。

下記の手順を繰り返すことにより、要求定義完成時に機能仕様書とシステムテスト項目が同時にレビューできる。したがって課題2を解決している。

- (手順1) 要求分析時に本チェックリストを活用し機能仕様を明確に定義する。
- (手順2) 上記で定義した機能仕様書を基にシステムテスト項目を抽出する。
- (手順3) 本チェックリストを活用しシステムテスト項目の漏れをチェックする。
- (手順4) 手順3でテスト合格基準が明確でない(仕様が不明確な)場合は、機能仕様の検討をやり直す。

5.4.3. 課題3 上流工程において、仕様が明確にできない。

要求分析時に本チェックリストの活用で仕様の曖昧な箇所を明確にすることができる。したがって課題3を解決している。

5.4.4. 課題4 トラブル事例のフィードバックができていない。

本チェックリストに該当しない事例がある場合は、チェックリストを補足改版する必要がある。課題4を解決するには、それぞれの会社独自の取組みが必要である。その取り組み方法について体系付けできていないので課題4は未解決である。

6. おわりに

今回我々は、ISO/IEC 9126 品質特性・副特性および INSTAC の内部品質特性を、一度インディケータという視点に落とし、集約することによってチェックリストを完成させ、そのチェックリストを設計前のミーティング、レビュー、テストの場で使用することによって、難解な ISO/IEC 9126 を現場視点に落とすことができた。その結果、非機能要件の可視化や機能要件の抜け防止などの効果を得ることができた。

しかし、チェックリストを使用するにあたり、機能の詳細化・テストするソフトウェアの設計ポリシーの理解・要求されたシステムの実動作イメージの認識などが必要となることがわかった。したがって、手始めにこのチェックリストの使い方についてレビュワーや設計者に理解させるためのトレーニングをしたり、過去の不具合事例をチェックリストに反映したりすることなどが必要であることが分かった。

今後の課題として、本チェックリストをもっと現場の人に簡易的に使ってもらうための具体的な手段、チェックリストをもっと上流の工程において使う方法、チェックリストの網羅性をどう捉えるか、などが挙げられるが、チェックリストの使用事例が少ないため今後もっと多くの課題が発生することが予想される。また、本チェックリストはインディケータの集約という視点で作成していることから、チェックリストによる機能の組合せによる問題点の検出、状態遷移のテスト網羅など設計の脆弱性の検出については適応外としていることも課題の一つである。これらの課題解決については、次年度以降で本研究を引き継がれる方に期待する。

7. 参考文献

- [1] 東基衛：ソフトウェア品質評価ガイドブック，（財）日本規格協会，1994
- [2] 真野俊樹、福山俊一、徳山芳男、坂本英夫、富本達明、池村健一、島村陽子、矢島正人：オープン化時代に対応した開発プロセスと品質メトリクス及び評価方式の研究，第13年度ソフトウェア品質管理研究会分科会報告書，（財）日本科学技術連盟，1993，pp45-61
- [3] 真野俊樹、菊本正紀、福山俊一、坂本英夫、猪坂伸二、斉藤一雄、柴田浩、中川忠彦：組込型ソフトウェアにおける内部特性とそのインディケータの研究，第14年度ソフトウェア品質管理研究会分科会報告書，（財）日本科学技術連盟，1994，pp66-83
- [4] 君島浩、小笠原秀人、豊永正人、鈴木秀幸、亀和田里江、奈良信一、田上典子、藤原克隆、白土智志：GUI・マルチメディアの品質保証，第15年度ソフトウェア品質管理研究会分科会報告書，（財）日本科学技術連盟，1995，pp19-37
- [5] G. J. マイヤーズ：ソフトウェア・テストの技法，近代科学社，1980，pp123-129

付録

付録1 チェックリスト

今回、我々研究員が作成したチェックリストを以下に記載する。

付表1 要求分析用チェックリスト(1/2)

No.	チェック項目	特性	副特性
1	要求事項は過不足なく、要求分析結果に反映しているか。また、要求事項と、要求分析結果の対応と、その効果は明確に定義しているか。	内部特性	完全性
			追跡可能性
			注目性
2	開発対象のシステムに関連する法規、業界標準の有無の調査と、それらに対するシステムでの対応方針は検討したか。 開発にあたり、守るべき開発規約や標準の有無、それらの変更や新規作成の必要性の検討をしたか。	移植性	規格適合性
		機能性	標準適合性
		内部特性	相互運用性
			統一性
			注目性
			一貫性
			データ共通性
通信手順共通性			
伝達性			
3	計算処理が存在する場合、その有効桁数や、まるめ処理に関する要求は明確に定義したか。	機能性	正確性
		内部特性	計算正確性
4	要求分析結果は、具体的にイメージできる表現であるか。難解な内容に関しては、理解を補助する工夫をしているか。	使用性	理解性
		内部特性	比喩性
			注目性
5	対応するハードウェア構成、ソフトウェア構成、運用形態などの前提条件の定義、必要最低限の性能は明確か。 また、これらの前提が変更になった場合の対応方法や、システムの拡張性とその実現方法は検討したか。	移植性	環境適合性
		内部特性	設置性
			置換性
			資源効率性
6	性能要件は検討したか。	使用性	運用性
		効率性	時間効率性
		内部特性	適時性
7	セキュリティ要件は検討したか。 暗号化すべきデータを明確にし、不正なアクセスに対する検出・記録は検討したか。	機能性	セキュリティ
		内部特性	アクセス監査性
8	ユーザインタフェースについて、メッセージ、操作、レイアウト等の標準化は検討したか。また、誤入力、誤操作への考慮は検討したか。 デフォルト値によるユーザ入力省略を検討したか。	使用性	理解性
			運用性
			習得性
		信頼性	障害許容性
		内部特性	省力性
			説明性
			堅固性
安全性			
アクセス制御性			

付表2 要求分析用チェックリスト(2/2)

No.	チェック項目	特性	副特性
9	予想される障害の発生原因や、その発生確率は検討したか。また、障害発生時の障害検知、障害記録、原因追求、対応・復旧までのプロセスについての要件を検討しているか。	信頼性	成熟性
			障害許容性
		保守性	解析性
		内部特性	整合性
			安全性
			計測性
10	導入方法及び導入支援(インストール)に関し、所要時間や導入手順などの作業者負荷の軽減を検討したか。また、導入支援実施において、プロセス、必要な成果物とその記載内容の定義、役割分担は検討したか。	使用性	運用性
			理解性
11	導入後の定着化支援(習得)に関し、マニュアルやヘルプ機能充実の検討をしたか。また、定着化支援実施において、プロセス、必要な成果物とその記載内容の定義、役割分担は検討したか。	使用性	習得性
		内部特性	自己記述性
12	導入後の運用・保守に関し、ドキュメントの充実について検討したか。また、運用・保守業務において、プロセス、必要な成果物とその記載内容の定義、役割分担は検討したか。	保守性	変更性
13	要求分析結果フェーズ及び、その実装結果を確認する為のシステムテストの実施方法と、フェーズの終了条件は検討したか。	機能性	合目的性
			正確性
		信頼性	成熟性
		内部特性	無矛盾性

付表3 システムテスト用チェックリスト(1/2)

No.	チェック項目	特性	副特性
1	開発対象のシステムに関連する法規、業界標準を遵守しているか。	移植性	規格適合性
		機能性	標準適合性
			相互運用性
		内部特性	統一性
			注目性
			通信手順共通性
2	計算処理について有効桁数や、まるめ処理の結果が正しいか。	機能性	正確性
		内部特性	計算正確性
3	対応するハードウェア構成、ソフトウェア構成、運用形態などの必要最低限の性能を満たしているか。 また、これらの前提が変更になった場合のデグレードチェック結果が正しいか。	移植性	環境適応性
		効率性	資源効率性
4	性能・負荷要件を満たしているか。	使用性	運用性
		効率性	時間効率性
		内部特性	適時性
5	セキュリティ要件を満たしているか。 仕様通りの暗号化ができていないか。 不正なアクセスに対する検出・記録ができていないか。	機能性	セキュリティ
		内部特性	アクセス監査性

付表4 システムテスト用チェックリスト(2/2)

No.	チェック項目	特性	副特性
6	ユーザインタフェースについて、メッセージ、操作、レイアウト等が標準化したものになっているか。また、誤入力、誤操作の対応ができているか。デフォルト値によるユーザ入力省略が可能か。	使用性	運用性
		信頼性	障害許容性
		内部特性	省力性
			説明性
			堅固性
			安全性
アクセス制御性			
7	予想される障害の発生確率を測定したか。また、障害発生時の障害検知、障害記録、原因追求、対応・復旧までのプロセスについての要件を満たしているか。	信頼性	成熟性
			障害許容性
		内部特性	安全性
		計測性	
8	要求分析結果フェーズ及び、その実装結果が確認できたか。システムテストの実施方法と、フェーズの終了条件を満たしたか。	信頼性	成熟性
		機能性	正確性
		内部特性	無矛盾性

付表5 基本設計用チェックリスト(1/2)

No.	チェック項目	特性	副特性
1	要求分析の内容を過不足なく展開しているか。	機能性	標準適合性
		内部特性	完全性
			追跡可能性
2	仕様書を階層的に表現しているか。	内部特性	階層性
3	仕様改定率はどのくらいか計算したか。	機能性	合目的性
4	計算処理が存在する場合、その有効桁数や、まるめ処理に関して正しく設計しているか。	機能性	正確性
		内部特性	計算正確性
5	メッセージ名称・用語、画面操作方式の統一化については検討しているか。	使用性	運用性
		内部特性	省力性
6	データ形式、媒体フォーマット、文字・制御コード、インタフェース・フォーマット、通信手順などについて標準形式を定め、それを遵守しているか。	機能性	標準適合性
			相互運用性
		内部特性	一貫性
			データ共通性
			通信手順共通性
			アクセス可能性
			統一性
			表現性
			適量性
			簡潔性
			選択性
			資源使用性
伝達性			
7	セキュリティレベルが定められ、それに合わせた設計がなされているか。また、暗号化やパスワードの必要性について検討したか。	機能性	セキュリティ
		内部特性	アクセス制御性

付表6 基本設計用チェックリスト(2/2)

No.	チェック項目	特性	副特性
8	機能の適用範囲が明確であり、十分検討しているか。ユーザ向け機能、運用者向け機能ともに検討しているか。	機能性	セキュリティ
		使用性	理解性
			習得性
			運用性
		内部特性	自己記述性
			アクセス監査性
計測性			
9	変数の通常稼働範囲を検討し、デフォルト値設定可能な場合はその値を正しく設定しているか。また、各変数について、異常値の保護対策や暗号化、圧縮の必要性を検討しているか。	使用性	運用性
		機能性	セキュリティ
		内部特性	省力性
			資源使用性
10	異常操作の検出・取消・回避の方法が定義しているか。また、異常の原因が分かるような情報が取得できるよう考慮したか。	機能性	セキュリティ
		使用性	運用性
		保守性	解析性
		信頼性	障害許容性
		内部特性	アクセス制御性
			堅固性
整合性			
11	利用者が理解しやすく使いやすいようなガイドを準備したか。	使用性	理解性
			運用性
		内部特性	誘導性
12	多重化が必要な場合、どの処理が多重化するのか。	内部特性	動的効率性
13	処理の分散を考慮したか。	内部特性	動的効率性
14	ハードウェアやネットワークの性能を考慮して、性能要件を見積もっているか。	使用性	運用性
		効率性	時間効率性
			資源効率性
		内部特性	適時性
			動的効率性

付表7 結合テスト用チェックリスト(1/2)

No.	チェック項目	特性	副特性
1	要求分析の内容を網羅したか。	機能性	標準適合性
		内部特性	完全性
2	計算処理が存在する場合、その有効桁数や、まるめ処理に関する結果が正しいか。	機能性	正確性
		内部特性	計算正確性
3	メッセージ名称・用語、画面操作方式は統一しているか。	使用性	運用性
		内部特性	省力性

付表8 結合テスト用チェックリスト(2/2)

No.	チェック項目	特性	副特性
4	データ形式、媒体フォーマット、文字・制御コード、インタフェース・フォーマット、通信手順などが標準形式を遵守しているか。	内部特性	一貫性
			データ共通性
			通信手順共通性
			統一性
			表現性
			適量性
			簡潔性
			選択性
資源使用性			
5	セキュリティレベルが定められ、それに合わせた実動作をしているか。また、暗号化やパスワードが要求通りに動作しているか。	機能性	セキュリティ
		内部特性	アクセス制御性
6	実動作を想定したテストを実施したか。ユーザ向け機能、運用者向け機能ともにテストしたか。	機能性	セキュリティ
		使用性	理解性
			習得性
			運用性
		内部特性	自己記述性
			アクセス監査性
			計測性
安全性			
7	変数の境界テストを実施したか。デフォルト値が設定可能な場合に正しく設定されているか。また、各変数について、異常値の保護対策や暗号化、圧縮が仕様通りに動作しているか。	使用性	運用性
		機能性	セキュリティ
		内部特性	省力性
			資源使用性
8	異常系テストにおいて異常操作の検出・取消・回避が想定通りに動作しているか。 また、異常の原因が分かるような情報が取得できたか。	機能性	セキュリティ
		使用性	運用性
		保守性	解析性
		信頼性	障害許容性
		内部特性	アクセス制御性
			堅固性
			整合性
安全性			
9	利用者が理解しやすく使いやすいようなガイドが想定通りに動作しているか。	使用性	理解性
			運用性
		内部特性	誘導性
10	多重化における不具合が発生していないか。多重化における動作効率の効果を測定したか。	内部特性	動的効率性
11	処理の分散における不具合が発生していないか。処理の分散における動作効率の効果を測定したか。	内部特性	動的効率性
12	ハードウェアやネットワークの性能要件を満たしているか。	使用性	運用性
		効率性	時間効率性
			資源効率性
		内部特性	適時性
			動的効率性

付表9 詳細設計用チェックリスト

No.	チェック項目	特性	副特性
1	上位設計の内容を十分に詳細化しているか。	機能性	合目的性
		内部特性	完全性
2	上位設計の内容が漏れなく詳細化しているか。	内部特性	完全性
			追跡可能性
3	演算の端数処理を明確に定義しているか。	機能性	正確性
		内部特性	計算正確性
4	標準・規約・データ形式・通信手順など設計標準に準拠しているか。	内部特性	一貫性
			データ共通性
			通信手順共通性
5	モジュール分割が適切か。 モジュール独立性は高いか。	内部特性	モジュール性
6	想定される異常処理パターン(データ値範囲外、誤操作、システム異常など)を網羅しているか。 異常状況の把握を検討しているか。	内部特性	計測性
			アクセス制御性
			堅固性
			整合性
7	異常処理内容(異常発生の記録、診断機能装備、データバックアップ等)が適切に記述しているか。	内部特性	アクセス監査性
			診断性
			安全性
8	性能要件(動作速度、メモリ容量使用量等)が明確か。	内部特性	アクセス監査性
			適時性
			動的効率性
9	変数・データの形式、デフォルト値などを明確に定義しているか。	内部特性	データ共通性
			省力性
10	実動作環境(ネットワーク環境、ソフトウェア環境、ハードウェア環境、データ環境等)を考慮して設計しているか。 また、それは想定可能な仕様変更に対して容易に対応できることを考慮しているか。	内部特性	環境適合性
			通信手順共通性
			ソフトウェアシステム独立性
			アクセス可能性
			マシン独立性
			アクセス可能性
データ独立性			

付表10 単体テスト用チェックリスト

No.	チェック項目	特性	副特性
1	テストパターンはその機能を網羅しているか。	内部特性	完全性
2	上位設計の内容を漏れなくテストしているか。	内部特性	完全性 追跡可能性
3	演算の有効桁数について桁落ちしていないか。	機能性	正確性
4	異常系テスト(データ値範囲外、誤操作、システム異常など)で想定通りの結果が得られたか。	内部特性	計測性 アクセス制御性 堅固性 整合性
5	異常処理内容(異常発生の記録、診断機能装備、データバックアップ等)が想定通りに動作しているか。	内部特性	アクセス監査性 診断性 安全性
6	動作時間が想定範囲内で動作しているか。 メモリアオーバーフローをしていないか。	内部特性	適時性 動的効率性
7	変数・データの形式、デフォルト値などが想定通りに動作しているか。	内部特性	省力性
8	インタフェーステスト結果が、仕様通りか。	内部特性	通信手順共通性 アクセス可能性 マシン独立性 アクセス可能性

付録2 システムテスト用チェックリストの検証

我々が作成した要求分析用チェックリストとマイヤーズの「ソフトウェア・テストの技法」(参考文献[5])に記載されているシステムテストの種類との比較結果を付表11に示す。

付表11 要求分析用チェックリストとシステムテストの種類との比較

No.	チェック項目	該当するシステムテストの種類
1	要求事項は過不足なく、要求分析結果に反映しているか。また、要求事項と、要求分析結果の対応と、その効果は明確に定義しているか。	機能部分テスト
2	開発対象のシステムに関連する法規、業界標準の有無の調査と、それらに対するシステムでの対応方針は検討したか。 開発にあたり、守るべき開発規約や標準の有無、それらの変更や新規作成の必要性の検討をしたか。	
3	計算処理が存在する場合、その有効桁数や、まるめ処理に関する要求は明確に定義したか。	機能部分テスト
4	要求分析結果は、具体的にイメージできる表現か。難解な内容に関しては、理解を補助する工夫をしているか。	機能部分テスト
5	対応するハードウェア構成、ソフトウェア構成、運用形態などの前提条件の定義、必要最低限の性能は明確か。 また、これらの前提が変更になった場合の対応方法や、システムの拡張性とその実現方法は検討したか。	構成テスト 互換性 / 変換テスト 記憶域(ストレージ)テスト
6	性能要件は検討したか。	ボリュームテスト パフォーマンステスト
7	セキュリティ要件は検討したか。 暗号化すべきデータを明確にし、不正なアクセスに対する検出・記録は検討したか。	セキュリティテスト
8	ユーザインタフェースについて、メッセージ、操作、レイアウト等の標準化は検討したか。また、誤入力、誤操作への考慮は検討したか。 デフォルト値によるユーザ入力省略を検討したか。	ユーザビリティテスト
9	予想される障害の発生原因や、その発生確率は検討したか。また、障害発生時の障害検知、障害記録、原因追求、対応・復旧までのプロセスについての要件を検討しているか。	信頼性テスト 回復(リカバリ)テスト
10	導入方法及び導入支援(インストール)に関し、所要時間や導入手順などの作業員負荷の軽減を検討したか。また、導入支援実施において、プロセス、必要な成果物とその記載内容の定義、役割分担は検討したか。	インストールテスト 手続き(プロシージャ)テスト
11	導入後の定着化支援(習得)に関し、マニュアルやヘルプ機能充実の検討をしたか。また、定着化支援実施において、プロセス、必要な成果物とその記載内容の定義、役割分担は検討したか。	文書(ドキュメント)テスト
12	導入後の運用・保守に関し、ドキュメントの充実について検討したか。また、運用・保守業務において、プロセス、必要な成果物とその記載内容の定義、役割分担は検討したか。	サービス性テスト
13	要求分析結果フェーズ及び、その実装結果を確認する為のシステムテストの実施方法と、フェーズの終了条件は検討したか。	