

品質レベルの最適化を実現する品質要件管理手法の提案

水野 高宏† 前田 繁章† 三輪 功† 和田 義毅†

内藤 孝一† 齋藤 毅† 高橋 成文†

株式会社 NTTデータ†

1. はじめに

システム開発では、限られた予算で必要十分な品質レベルを実現することが要求される。しかし、品質に大きな影響を与えるアーキテクチャ設計では、ITアーキテクトのスキルへの依存度が高く、設計妥当性の検証手法が確立されていない。このため、ITアーキテクトのスキルが低い場合には、品質が不十分であったり、過度な品質が作りこまれる恐れがある。そこで本稿では、品質要件に基づいたアーキテクチャの設計及び検証に必要な手順を定義し、ITアーキテクトのスキル不足を補完して品質レベルを最適化するための品質要件管理手法を提案する。

2. 品質要件管理における課題

アーキテクチャ設計を IT アーキテクト主導で実施するシステム開発においては、企画工程で顧客の品質要件を明確化し、アーキテクトがアーキテクチャの各構成技術を持つ品質特性を見極めながら、顧客の品質要件を満足できるアーキテクチャを設計する。しかし、このような開発手法には、以下に示す問題がある。

(1) スキルへの依存

品質要件に基づく設計（品質設計）では、信頼性、性能、コストなど、相反する要素を総合的に検討する必要があり、適切な設計には幅広い知識と豊富な経験が求められる。従って、ITアーキテクトのスキルが品質設計の質を決定付けることが多く、スキルの高い IT アーキテクトが不足している状況では、品質レベルが最適化されたシステムを開発することが困難である。

(2) 品質の妥当性

本来システムが必要としている品質と、設計したアーキテクチャが実現する品質の間にギャップが生じた場合でも、ギャップを解消するための手順が確立されていないため、品質レベルが不適切なままシステムが作りこまれる恐れがある。例えば、品質が不足している場合には、総合試験や総合運転試験、あるいは実運用開始後になって初めて問題が浮き彫りになる。

これらの問題を解決するために、本稿では品質レベルの最適化を実現する以下の 2 つの手法を提案する。

- アーキテクトの経験やスキル不足を補完するための品質設計手法

品質設計手法

- 設計の妥当性検証により要件と実現レベルのギャップを解消するための品質設計の検証手法

3. 提案手法

本提案では、品質を決定付けるアーキテクチャを最適化するために、アーキテクチャ要件定義の工程に新たに品質要件管理手順を定義する。図 1 に品質要件管理手順の全体像を示す。本手法では、アーキテクチャを個別案件ごとに構築するのではなく、リポジトリに蓄積されたアーキテクチャ事例を出発点に設計を行う。リポジトリには、典型的なアーキテクチャ事例と、整合性や品質を検証するためのノウハウが蓄積されており、これらを参照しながら設計と検証を進める。

なお、アーキテクチャの選定では、属性情報である品質特性やコストを参照する。品質特性には ISO/IEC 9126^[1]や JIS X0129^[2]等の標準的な規定があるが、企業ごとに独自の品質規定を採用しているケースも多い。従って、本稿では、品質特性自体の定義は行わず、与えられた品質特性が存在することを前提に、品質管理の実施手順を規定する。

3.1. 品質設計手法

アーキテクトのスキル不足を補完し、適切な品質設計を行うために、以下の 3 つの手順を規定する。

(1) システムアーキテクチャ選定

リポジトリには、アーキテクチャ事例とそれが実現している品質レベルやコスト等の属性情報が「シート A. システムアーキテクチャ選択シート」として管理されている。アーキテクトはシートを参照し、品質要件に最も近いアーキテクチャを選定する。この手順では、ユーザの品質要件を詳細レベルで把握したり、品質要件を厳密に満たすようアーキテクチャを熟考する必要は無く、最小限の作業でベースとなるアーキテクチャの当たりを付けることが目的である。図 1 では、リポジトリから「Web 系小規模公開サービス」というアーキテクチャを選択し、サーバやネットワーク等のベース構成を決定している。

(2) 個別アーキテクチャカスタマイズ

(1)での選定結果に基づき、要件に適合した品質を実現できるようにアーキテクチャのカスタマイズを行う。リポジトリには、アーキテクチャの構成要素ごとに属性情報が「シート B. 個別アーキテクチャ選択シート」として管理されており、構成要素単位でカスタマイズを行う。

図 1 では、「Web 系小規模公開サービス」のアーキテクチャの完全性やストレージの利用効率が要件を満たさないことから、サーバとストレージについてそれぞれカスタマイズを行っている。

(3) 組合せ整合性チェック

Proposal of quality requirement management process for optimization of quality level

† Takahiro MIZUNO (mizunotk@nttdata.co.jp)

Shigeaki MAEDA (macdasg@nttdata.co.jp)

Isao MIWA (miwai@nttdata.co.jp)

Yoshinori WADA (wadaysn@nttdata.co.jp)

Kouichi NAITOU (naitouk@nttdata.co.jp)

Takeshi SAITOU (saitoutk@nttdata.co.jp)

Shigefumi TAKAHASHI (takahashig@nttdata.co.jp)

NTT DATA CORPORATION

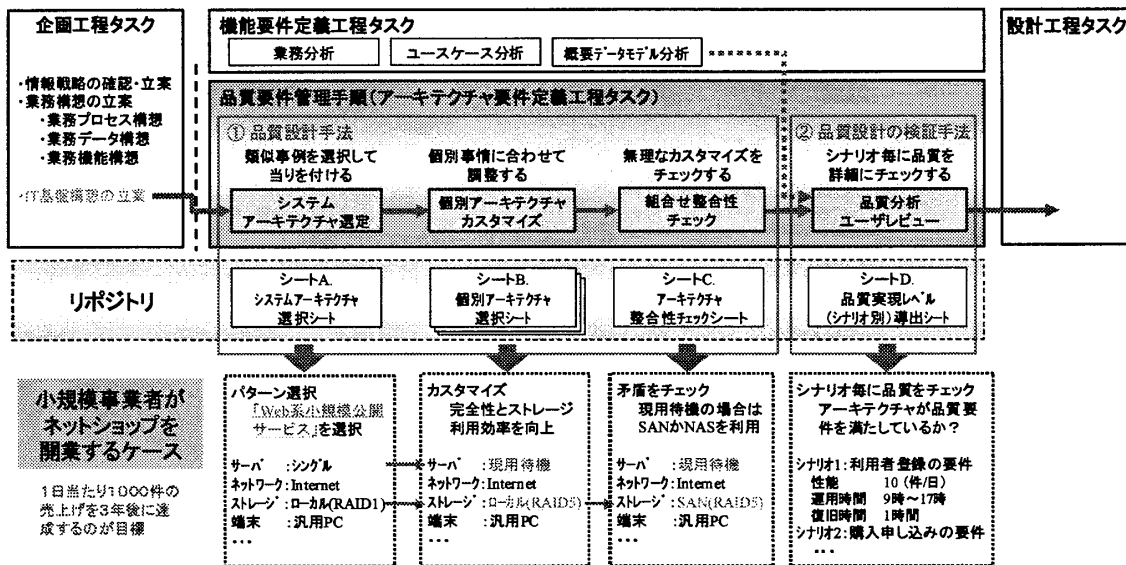


図 1 品質要件管理手順

「シートC.アーキテクチャ整合性チェックシート」を参照し、論理的矛盾や不適切な組合せのチェックを行う。ここで問題があれば、(2)のカスタマイズを再度実施する。図1では、「現用待機」と「ローカル(RAID5)」の組合せが適切でないため、ストレージを「SAN(RAID5)」に修正している。

3.2. 品質設計の検証手法

3.1 では、企画工程の段階で把握可能な品質要件に基づいてアーキテクチャの選定を行う。しかし、企画工程で大きな品質要件しか把握していない場合や、そもそもユーザが認識していない要件がある場合には、本来求められている品質が実現できている保証は無い。そこで、具体的な利用シナリオに対応する形でユーザの品質要件を詳細化し、ユーザの視点で品質設計の妥当性を検証するために、以下の2つの手順を規定する。

(1) 品質分析

一般的に、システムには複数の使われ方(シナリオ)があるため、本手順では品質要件をシナリオ単位で詳細化し、品質設計の検証を行う。図2に手順を示す。

まず、機能要件定義にて実施するユースケース分析の結果をシナリオとして定義し(①)、各シナリオの実行に関係するシステムの構成要素を洗い出す(②)。ここでは、「1.利用者登録」「2.購入申し込み」「3.運用管理」の3シナリオを定義し、それぞれのシナリオが関係するシステム構成要素を明確化している。

次に、ユーザへのヒアリングにより、シナリオ単位で品質要件の確認を行う(③)。これをシナリオ品質要件と呼ぶ。また、「シートD.品質実現レベル(シナリオ別)導出シート」を参照して、3.1で選択したアーキテクチャの各構成要素が持つ品質特性に基づき、シナリオごとに品質特性の導出を行う(④)。これを、シナリオ品質特性と呼ぶ。ここでは、シナリオ品質特性として障害頻度レベル別の復旧時間を算出している。

最後に、シナリオ品質要件とシナリオ品質特性を比較し、過度な品質特性及び不十分な品質特性をギャップとして洗い出す(⑤)。ここでは、「2.購入申し込み」シナ

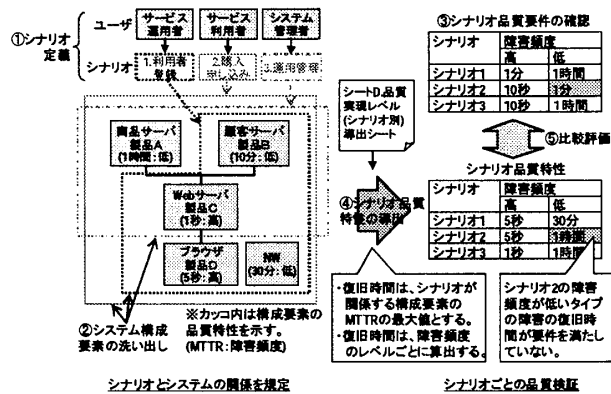


図 2 シナリオごとの品質検証

リオの障害頻度が低いタイプの障害の復旧時間が要件を満たしていないことがわかる。

(2) ユーザレビュー

(1)での分析結果についてユーザとのレビューを行う。ここで、シナリオ品質特性は、ITアーキテクトの視点ではなく、ユーザが理解できる形で提示する必要がある。ギャップのある部分については、ユーザと協議の上で、シナリオ品質要件の見直し、あるいはアーキテクチャを再度カスタマイズすることで対応する。ユーザの視点でアーキテクチャをレビューできる機会を設けることで、開発の初期工程でギャップを解消することが可能となる。

4. まとめ

本稿では、ITアーキテクトのスキルへの依存度を軽減するための品質設計手法、及び要件と実現レベルのギャップを解消するための品質設計の検証手法を、品質要件管理手順として定義し、システムの品質レベルを最適化するための開発手順を提案した。

参考文献

- [1] http://ja.wikipedia.org/wiki/ISO_9126
- [2] <http://www.jisc.go.jp/app/JPS/JPSO0020.html>