

非機能要求グレード2018 利用ガイド

活 用 編



独立行政法人情報処理推進機構
社会基盤センター

はじめに 《情報システムの企画開発・運用に関わる全ての方々へ》

社会・企業の情報システム依存度が増している一方で、情報システムやネットワークの想定外トラブルや、活用されていない情報システムの存在なども散見されます。これらの多くは「ビジネス上の要求がうまく情報システムに反映されていない」あるいは「ビジネス変化とともに情報システムへの要求も変化した」といったことに起因しています。

本書は、情報システムの企画・開発・運用に携わる全ての方々に、「要求」の重要性を認識いただき、特に敬遠されがちな「非機能要求」について、その検討に有効なツールである「非機能要求グレード」を活用した事例を交えて分かりやすく解説することで、利用者が安心して効率的に使える情報システムを開発・維持していく一助となることを目的としています。

【要求の重要性】

現代のビジネス環境においては、経営のビジネス戦略～情報化企画～システム化～運営のライフサイクルの中で、「経営からの要求」「業務部門からの要求」「利用現場からの要求」「開発部門の制約」「運用部門の制約と改善要求」などの観点から、いかに要求を整理し経営に貢献できる情報システムを開発・維持していくかが重要ではないでしょうか。

【非機能要求の必然性】

ビジネス上の要求は「こういうことを実現したい」という戦略です。

これを「業務プロセス」化し、「人」と「システム」に役割分担し、具体的にシステム面からの業務運用を検討し、既存システムがあれば移行や連携を踏まえてシステムを開発します。ひとことで言えば、

- ・「情報システム」で何を管理・処理するか
- ・「人」が「情報システム」をどう使うか

が要求であり、前者を「機能要求」、後者を「非機能要求」と呼びます。

非機能要求は、ハードウェア・OS・ミドルウェア・制御アプリケーション等に対するものが中心と思われがちですが、実際には、業務プロセスや業務アプリケーションに依存するものが多く存在します。例えば、可用性や運用・保守性では、

- ・運用時間帯や運用日、業務影響を考慮したサービス切替時間
- ・システムトラブルからの業務復旧容易性や災害対策範囲

が重要ですし、性能・拡張性では、

- ・取引量やデータ量の伸び、システム使用年数
- ・業務アプリケーションのデータアクセス回数や画面表示項目数と伝送データサイズ

等が重要になります。また、これらの要求を満たすためには、現場業務を円滑にこなせるレスポンスや操作性を確保するアプリケーション処理方式が重要になります。

【非機能要求に基づく実装の検討】

セキュリティ強度と性能や操作性等、非機能要求にはトレードオフの関係が多く存在します。どこまで運用を自動化するかは、初期コストとランニングコストのトレードオフになります。このような関係の中で「どの程度」のバランスを決めるのが非機能要求であり、実装方法は規定しません。その時点で使える技術の組合せで実現方式を設計し、製品・サービス動作環境の相互関連設定や機能不足部分の独自開発によって、非機能要求の情報システムへの実装が完成していきます。

【情報システムの要求に携わるステークホルダー】

「非機能要求グレード」は、情報システムを新規開発する場合を想定して作成されました。

しかし、今では稼働状態やクラウドサービス仕様のアセスメント、環境や使われ方の変化に伴う要求の見直し、次期システムに向けた現状可視化や共通基盤のサービスメニューパターン化など、多くのシーンで利用が拡大しています。

ステークホルダーには、ICT 投資判断に関わる経営層、業務運営を担う業務部門・事業部門、情報システムを担う開発部門・運用部門、現場の利用者などの様々な立場があり、多くの要求の整合性を取るための共通尺度として「非機能要求グレード」の有効性が評価されているといえます。

【変化する要求への対応】

「非機能要求グレード 2018」では、仮想化技術の普及に伴うメトリクス表現の見直しや、よりプロアクティブな対応が求められるようになったセキュリティ項目の改訂を実施しました。とはいえ、システム基盤に対する非機能要求の本質は変わりません。モバイル、クラウド、AI、IoT 等の技術進化は、情報システムの実装やリソース調達形態に関わるものであり、機能要求が不明確あるいは変化に柔軟に対応するためのアジャイル開発やクラウドサービス利用においても【非機能要求の必然性】で述べた重要性に変わりはないのです。例えば、

- ・ アジャイル開発においても、性能・拡張性や可用性、セキュリティの確保は重要
(確認テストを随時実施し、当初の要求を越えるレベルが想定される場合は再構築も考慮)
- ・ クラウドサービス仕様を非機能要求レベルと比較し、不明確な項目はリスクとして見込む
(事前検証を実施し、時にはサービス仕様を制約と捉えて要求を見直すことでコストを抑える)
- ・ 複数のサービスやシステムと連携する場合は、相手方の非機能要求実現レベルも考慮する
(機能を縮退してサービス継続する場合は、アプリケーション側の対応が必須)
- ・ PoC (実現性検証) であっても、セキュリティ事故等を招かないように非機能要求を明確にする
(運用を含めて、PoC としての要求と本稼働に必要な要求を明確に区別する)

といった事がポイントになります。

既存システムの再構築では要求を「現行通り」で済ませがちですが、使われ方が変化している場合やアーキテクチャー変更に備えて非機能要求の可視化が必須です。保守開発の場合でも、変化の織り込みや要求の可視化が不十分なままの不用意な改修は、性能劣化やセキュリティ強度の低下を招きます。

【本書を効果的に活用いただくために】

情報システムの企画・開発・運用は、情報システム部門だけの責任ではありません。

必要な業務機能を実現するだけでなく、業務プロセスにおけるクリティカル性や利用者層、セキュリティ強度やデータ量の拡張限界等を見極めて合意することは、情報システムに関わる全ての方々の責任です。

本書の改訂にあたっては、これまでの実績を踏まえた様々な「非機能要求グレード」活用事例を追加するとともに、参考にして頂きたい事例を情報システムのライフサイクル別に示すことで、各工程に関係するステークホルダーの方々が有益な情報に素早くたどり着けるようにしています。情報システムが社会基盤化していく中で、多くのステークホルダーが情報システムに関わるようになり、システム基盤としての要求を明確にする「非機能要求グレード」は強力な合意形成のツールとしてますます価値を発揮するはずです。

目 次

本書で紹介する活用シーンを俯瞰するために、共通フレーム 2013 の企画プロセスから運用プロセスのどのプロセスでどのシーンが活用できるかを、以下のように整理しました。
各プロセスに対応した活用シーンを選択するときの参考にしてください。
(プロセスの定義は表の下を参照ください)

活用シーン	プロセス					ページ
	企画	要件定義	システム開発	保守	運用	
1章 新規構築						1
1.1 不明確なユーザー要求をクリアにする		○				2
1.2 ヒアリングシートを投げ出さずにすり合わせる		○				4
1.3 衝突する要求をさばく		○				6
1.4 事業のビジョンを決め手にジャッジする	○	○				8
1.5 アーキテクチャーの選択肢を持つ			○			10
1.6 より短いサイクルを繰り返して要求の成熟を狙う		○	○			12
1.7 スモールスタートを前提に要求をまとめる	○	○				14
1.8 トレードオフ要素に優先順位をつける		○	○			16
1.9 クラウドサービスへの要求を整理する (クラウド)	○	○				18
1.10 スピーディに概算で見積もる		○				20
【TIPS】段階的詳細化のすすめ						22
2章 保守・更改						23
2.1 引継ぎを機に現状を把握し課題を洗い出す				○	○	24
2.2 更改時に漏れなく問題を抽出する				○		26
2.3 保守対象システムの現状を点検する	○			○	○	28
2.4 保守開発による性能低下を防ぐ				○	○	30
2.5 法律や制度の改正に伴いシステムの改修を行う				○		32
2.6 事業変革に伴う要求変化に対応する	○	○		○		34
2.7 売り文句による一面的な判断を防ぐ (クラウド)	○			○		36
2.8 手段にとらわれずニーズを明確化する (クラウド)		○		○		38
【TIPS】システム拡張開発における非機能要求の考え方						40
3章 運用						41
3.1 システム運用時間とバックアップのトレードオフを考える	○	○			○	42
3.2 障害時の運用について業務毎の優先度を考える	○				○	44
3.3 システム開発時に運用コストを検討する					○	46
3.4 システム毎に BCP を評価する	○				○	48
3.5 機器増設要求の妥当性を評価する	○				○	50
3.6 データセンター移設の影響を洗い出す		○	○		○	52
3.7 情報セキュリティに関する新たな脅威に備える		○	○		○	54
3.8 ビジネス環境の変化に合わせて要求を見直す (クラウド)	○			○	○	56
【TIPS】開発コストと運用コストのトレードオフ						58
4章 標準化・アセスメント・展開						59
4.1 社内標準を作る	○	○				60
4.2 プロジェクトのレビューに使う		○				62
4.3 システム統合時に現状を比較する	○					64
4.4 新技術の導入を検討する	○	○				66
4.5 更改前に現行システムを評価する	○				○	68
4.6 多数のシステムに対して利用状況を点検する	○				○	70
4.7 事業継続計画はビジネス視点で検討する	○					72
4.8 サービス提供者として SLA を検討する (クラウド)		○				74
4.9 要員育成に活用する		○	○			76
【TIPS】非機能要求のカテゴリ別の考え方						78

＜プロセスの定義＞

- (1) 企画 : システムが関与するシステム化構想の立案、システム化計画の立案などを行う組織のプロセス
- (2) 要件定義 : システムが実現する仕組みに係る要件を定義する組織のプロセス
- (3) システム開発 : システム開発を行う組織のプロセス
- (4) 保守 : ソフトウェアを中心としたシステムの現状を、業務ならびに環境に適合するように維持、変更、管理する組織のプロセス
- (5) 運用 : 利用者の環境でシステムを運用する組織のプロセス

1 章

新規構築

システムの新規構築は少なくなってきましたが、システム構築の基本であることは言うまでもありません。

既存システムがある場合はそのシステムの現状を把握した上で要求を整理することができますが、全くの新システムの場合は要求を具体的に決めることが困難なため、どうしても漠然としたあいまいな要求になりがちです。中でも非機能要求と呼ばれている可用性・信頼性などは、定義も分かりにくく具体的な数値で定義されることが少ないようです。このような場合、『非機能要求グレード』が役に立ちます。システム構築でよく検討される項目が体系的に整理されており、さらに、それぞれの項目が持つべき値の例をレベルとして示しています。これらの具体的な値を参考にして、システムの要求値を具体的に決めることができます。

非機能要求グレードの利用ガイド【利用編】には要求を段階的に詳細化していく例が記載されており、コツさえつかめばそれほど時間をかけずに非機能要求項目を決定することができます。

「非機能要求グレード」はシステムの非機能要求を決める時に役立つツールですが、注意すべき点として、相手がシステムに詳しくない場合は、各項目の説明や、決定した値の根拠の確認などを丁寧に行い、合意する必要があります。間違っても、ダウンロードしたシートをそのまま送りつけてはいけません。どんな場合であっても、相手の立場に立った配慮が必要です。直接ヒアリングする場合も、具体的な数値を提示した上で意見を聞かないと、あいまいな要求をされて困ることになるので気を付けましょう。

全ての項目を細かく決めなくても一部分をクローズアップさせることで、ステークホルダー間の合意形成に活用することもできます。例えば、特に大きく費用に影響しそうな箇所をピックアップし、大きな方針を決めるために使ってみてはどうでしょうか。24 時間 365 日の連続運用が必要なシステムの場合、本当に必要であれば、コストをかけてでも多重化などが必要となります。しかし、「普段は連続運転して欲しいが、障害時や定期的なメンテナンスで事前に連絡があれば 1 日程度は止められる。」という場合には、多重化をせずにバックアップ機ですませることで構築コストを削減することが可能です。

非機能要求項目を決めていく過程において、項目間のトレードオフに気がきます。例えば、セキュリティは強化したいが、全てのデータを暗号化したり、画面の操作に頻繁にパスワードを要求したりすると、性能の低下やユーザビリティの低下を引き起こします。むやみに高いレベルの要求を主張するだけでなく、他の要求とのバランスを考慮する必要があります。

非機能要求の各項目が明確になった後で、実際のシステム設計に着手します。もちろん全てがきちんと決まらなくても、ある程度の範囲が決まれば設計は進められますが、せっかく決めた設計内容に手戻りが発生すると無駄なコストがかかりますので、できる限りシステム設計に間に合うように決めておきたいところです。設計が進めば、現在の技術でどのようなシステム構成になるか、その結果として非機能要求が実現可能であるかを検証することができます。ここがシステム担当者の腕の見せ所です。

本章では、新規構築案件への非機能要求グレードの使い方を見ていくことにします。

1.1 不明確なユーザー要求をクリアにする

千石物産の太田事業部長は、いかにも叩き上げらしい実直そうな初老の紳士だった。「今日は、新しい受注管理システムの非機能要求についてヒアリングをさせていただくためにお集まりいただきました。」

プロジェクトを請け負っているハクサン情報のSE、小原がそう切り出すと、太田事業部長は眉間にしわを寄せた。

「そう言われても、あたしらコンピューターは素人だからねえ。どんなことを聞きたいの？」

小原は少し考えた。とりあえず、要望を聞き出すために行って来いとプロジェクトリーダーの倉田に言われてやってきただけで、詳しいシナリオを考えているわけではない。

「例えば、現行システムの応答時間にご不満があると伺っていますが、どうでしょうか？」

太田事業部長の横には、ベテランの販売課事務員、水口佳代子が控えていた。彼女は、待つてましたとばかりに口を出した。

「そうなんです。月締めの時期になると、伝票受付画面が固まって、まるで壊れちゃったみたいになるんですね。実行ボタンをバンバン押しているとシステムが落ちちゃうし、何とかしていただかないと困ります。」

何の話かわかったと言うように、太田事業部長の顔がぱっと輝いた。

「そうそう。新しいシステムの画面は、キビキビとパッパッパという感じで切り替わって欲しいね。」

「パッパッパ、ですか・・・」

あまり定量的とは言えない相手の表現に、小原は少し戸惑った。追い打ちをかけるように水口が言う。

「それから、今度のシステムは24時間動かしてもらえないですかね。今のは夜8時で止まっちゃうんで、締日は追い込みが大変なんです。先月みたいに、ディスクの故障で丸一日止まるのも困りますね。平常時だったからよかったものの、キャンペーンの時なんかだと致命的です。それから・・・」

喋りはじめると、水口は止まらなかった。

ヒアリング結果について小原の報告を聞き、倉田は頭を抱えた。

「パッパッパじゃあ、要件定義にならないなあ。24時間稼働とRAID構成となると予算内では収まらないし、注文処理系とは言っても、関連会社からの物品注文がメインだから、それほどの可用性はいらんはずなんだよ。バックアップとセキュリティについては話ができなかったって？ 発注期限があるから、機器構成とファイヤーウォールの設定は急いで決めなきゃならないんだよ。こりゃあ、改めてヒアリングに行くしかないかなあ。」

「すみません。水口さんの勢いにすっかり押されてしまって・・・」

小原は頭を下げるしかなかった。



SEの小原さん、早期に非機能要求のヒアリングに臨んだところまではよかったものの、いささか準備不足だったようです。

結果から見ると、このヒアリングには三つの大きな問題があります。

一つ目は、「パッパッパ」というあいまいなレスポンス要求しか聞き出せなかったこと。

二つ目は、過大かもしれない「24時間ノンストップ」という水口さんの要求について、費用対効果の面から調整することができなかったこと。

そして三つ目は、バックアップ運用やセキュリティなど、コストに影響する確認点を漏らしてしまったことですね。

一つ目のポイントについて注意すべきことは、ユーザーは最初から「レスポンス n 秒」といったはっきりした要求を出してくれるわけではないということです。「現状困っている」のであれば、どの処理にどれだけの時間がかかっていて、どれくらい困っているのか、かかる時間が半分かになれば業務上の問題は解消されるのか…といった業務実態からアプローチしないと、なかなか真意を聞き出すことはできません。



二つ目については、業務の性質やコストとの兼ね合いを意識する必要があります。対象業務が対外顧客向けの基幹系システムなのか、社内用途のシステムなのかによっても、要求は違ってくるでしょう。機能提供に問題が生じた時の業務への影響や事業へのリスクが、業務によって異なるからです。

三つ目の点については、急いで確認したい点は何なのか、あらかじめ網羅的にまとめておく必要があります。頭の中で考えているだけでは、どうしても漏れが出てきますから、チェックリストや項目の一覧表を準備して、ユーザーとすり合わせる必要が出てきます。特に非機能要求については、ユーザーも日常の仕事で意識していないだけに、どうしても漏れが発生しやすくなるのです。

このようなすり合わせのシーンを考えると、「非機能要求グレード」のようなツールが有効です。非機能要求項目を網羅しているので、すり合わせの漏れをなくすることができますし、モデルベースで要求レベルの「グレード」を考えるように作られているため、業務特性を無視した過大な要求を調整することができます。

要求レベルの具体化についても、「例えばこんな性質のシステムではこの程度の応答」といった目安が示されているので、すり合わせの糸口になるでしょう。

業務の性質や要求レベルのグレードについては、組織毎に基準を作っておくのがベストです。そこまで時間の余裕がない場合でも、最低限、手元にツールを置いてヒアリングに臨むことで、丸腰でユーザーに立ち向かって、不意打ちをくらってしまった小原さんのような失敗を防ぐことができるでしょう。

1.2 ヒアリングシートを投げ出さずにすり合わせる

施設予約システムのサービスレベル定義について、上司の宮田次長に確認された菅沼は、自信ありげに答えた。

「今回はきっちり詰めるつもりですよ、秘密兵器がありますから。非機能要求グレードですよ。うちに合わせてちょっと加工して、ユーザーに事前送付してあるんです。」

宮田次長は笑みを浮かべながら頷いた。

「いや、経理から、またシステム化予算の見直しを言ってきたんでね。データベースサーバーのスペックを落とすことになるかもしれないが、それで大丈夫かちょっと気になってね。」

経理部によるシステム化予算の削減要求は厄介な問題だ。半期に一度はコスト削減の要求があるが、ユーザー部門は予算が削減されたからと言って我慢はしてくれない。

「そろそろ、来期の購買計画をあげる時期だ。その秘密兵器を使って、うまくユーザーの合意を取っておいてくれ、頼むよ。ユーザー要求が明確なら、スペックは下げられないと突っぱねることもできるしね。」

「わかりました。早速、今日のうちに催促しておきますよ。」

菅沼は、力強く答えた。

菅沼がかけた電話の相手、事務統括課の村上副長は、不機嫌な声を出した。

「ああ、あの一覧表のこと？ 月曜日に送ってきた。」

「それです、それです。何とかよろしくお願いします。」

菅沼は当然の要求をしたつもりだったが、なぜか怒ったような声が耳に反響する。

「200項目もあるものをいきなり送りつけられたってわかんないよ。こっちはコンピューターは素人なんだよ。ちゃんと時間取って説明してくれなきゃ、わかるわけないでしょう？」

菅沼の頭の中に疑問符が並んだ。

「ええと、例えば、あの、どのあたりが説明不足だったでしょうか??？」

少し間が空いてから、押し殺した声で答えが返ってきた。

「例えば、オンラインレスポンスって言ってもさ、応答にも色々あるじゃない。画面に結果が返るまでなのか、予約受付表が印刷されるまでなのか。成績統計みたいにまとめて結果が出てくるやつもあれば、お客様がじかにスポーツ施設なんかを Web 予約してくるケースもあるわけでしょ。一律で答えろって言われても困っちゃうよ。そうじゃない？」

「はあ。それはそうなんです、要求レベルを定義しないと、スペックが決められないので・・・」



「こっちは素人なんだから、そっちで考えてくれればいいじゃない。目標復旧時間とかさ、意味わかんないよ。止まんないようにするのが先でしょ？お客様が予約できなかったら、それだけ売り上げが減るんだから。冗談じゃないよ、全く。」

「し、しかし・・・」

「とにかく、一度説明してよ。悪いけど、今週はちょっと余裕ないから、また日を改めて。それじゃ。」

菅沼は、茫然として、切れた受話器を見つめていた。

解説

すり合わせしておくべき非機能要求項目に漏れないように、「非機能要求グレード」の項目を加工して確認シートを作成し、活用しようとしたところまでは、菅沼さん、冴えていたと思います。「非機能要求グレード」では、このような目的に利用できるように、自由に編集できる「非機能要求グレード活用シート」というものを用意しています。

ただし、それをいきなりユーザー（ここでは、利用部門や企画部門などを指します。P 7 参照）に送付して埋めてもらおうとしたのは、いささか乱暴だったかもしれません。

村上副長の言うとおりの、ユーザーは日々、非機能要求を意識して仕事をしているわけではありません。そもそも、新しいシステムを開発したり、そのために要求を定義したりすることは、ユーザーにとって、長い会社生活の中でも一度あるかないかという珍しい機会です。項目の意義や、項目間のトレードオフについて、一方的に判断を求めるのは酷と言うべきでしょう。いきなり 238 項目の活用シートを突き付けられても、戸惑うのが当然です。

「非機能要求グレード」は、業務に詳しく決定権を持っているユーザーと、システム開発の専門家である開発者が、協力してすり合わせるためのツールだと考えてください。必要なレベルについて即答できないようなユーザーでも考えやすいように、様々な工夫が施してあります。

システムの特徴 16 項目からモデルシステムを活用して、まずは全体の方向性を確認する、次にコストに大きな影響を及ぼす重要項目を固める、このあたりをキックオフまでに進めておいて、実際に開発が始まるまでに、レベル値を頼りに残りの項目をすり合わせていく。これが、標準的に想定される使い方です。結論を記載するシートというより、すり合わせ、詰めていくプロセスで利用するといいいでしょう。

そういう意味で、いきなり送付して記入を促すというより、対面で相互に質疑応答を繰り返しながら、必要なレベルを合意していくというやり方がお勧めです。とにかくユーザーは、なるべくなら安全でハイレベルなシステムを要求したくなるものですが、レベルを上げようとすれば工期が延び、コストもかさみます。また、当然経理部などから予算削減のプレッシャーもかかります。しかしユーザーには、ある要求がコストやアーキテクチャー、運用などにどういう影響を及ぼすのか想像が付きません。最適なレベルに調整するためには、「そこまでやるとコストが倍になりますよ」とか、「運用が難しくなってしまうのですが」「他のシステムではこの程度で我慢していますよ」「こういう代替策も考えられます」といった、開発の専門家の助言が不可欠なのです。

ツールをうまく活用して、お互いに納得のいく合意に至れるようにしましょう。



1.3 衝突する要求をさばく

仕事のピークが過ぎたせい、今日は、事務統括課の村上副長もいづれか落ち着いているように見えた。営業企画部の友永次長も同席しているからかもしれない。『湯の町リゾート別館はなやぎ』の黒田支配人も出席してくれていた。合意形成が難航しているため、システム担当の菅沼が事前に関係者に声をかけて、今日のミーティングの主旨やお願いしたい役割を説明していたのである。

出席者ひとりひとりに視線を配りながら、菅沼は口を切った。

「本日のミーティングの目的は、施設予約システムの使われ方をはっきりさせることです。」

村上副長が、早速口を尖らせる。

「例の、オンラインレスポンスがどうのこうのってやつでしょう？とりあえず、2秒ルールなら2秒ルールで決めちゃえばいいんじゃないの？」

村上副長の質問を予測していた菅沼は、落ち着いて応えた。

「一律2秒でいいかどうかは、これから検討したいと思っています。それから、テーマはレスポンスだけじゃありません。セキュリティもありますし、リカバリやコンティンジェンシーもテーマになります。そのために、黒田支配人にもご出席いただいています。システムがどういうふうに使われるべきかは、結局業務の現状とあるべき姿の鏡ですから。例えば、レスポンスにしても、お客様を前にフロントがどんなことをしているか、どういうサービスをしたいからどこまでお待ちいただけるか、といったことにかかっているはずですよ。」

菅沼にとって意外なことに、いかつい黒田支配人の顔がほころんだ。

「そのとおりだね。必要な性能は一律ではない。ご到着時、ご出発時、ご利用提案時で色合いが違う。例えば、予約簿なんかは朝一番に印刷してしまうから、ご到着時には端末を叩かなかったりするんでね。」

村上副長が、それは知らなかったという顔をする。友永次長も口を添えた。

「販売管理としても、締日かそうでないかによってニーズは変わります。Web サービスでは、キャンペーンなしの初見予約の時を重視しています。この時に何かあると、お客様はやり直してくれず、他の会社へ流れてしまいますから。」

菅沼は頷いた。

「主な処理について、現行のレスポンス統計を準備してきましたから、のちほど現状で十分なところと、ネックになっているところなど教えていただければと存じます。」

菅沼はさらに、穏やかな声で付け加えた。

「話が佳境に入る前に、わたしから一つ言いにくい話をしておきます。ご承知のとおり、システム投資抑制の要請もきています。今日のミーティングでは、ある程度コスト感にも言及できるように準備しております。やりたいが、やると余計にお金がかかる、という部分については、やるか我慢するか、友永次長にご裁定いただければと存じます。」

「嫌な役回りだなあ。」

友永次長は苦笑いをしたが、事前に聞いていたらしく、異を唱えることはなかった。

「まあ、どうせ経理に説明しなくちゃならないんだからね。わかりました。始めましょう。」

これなら・・・と、菅沼は先の見通しに希望を持った。うまく話がまとまるかもしれない。

解説

システム開発には、様々なステークホルダー、つまり決定に影響力を持つ関係者が登場します。商品や制度を企画する企画部門、実務を担当する事務部門、内部統制を所管する監査部門や総務部門、情報システム部門や事務統括部門。また、予算の立案と執行に関わる経理部門や財務部門に、顧客と直接接する営業やサービスの部門。会社の舵取りをする役目の経営層。一言で「ユーザー」などと称しますが、組織は決して一枚岩ではなく、ステークホルダーによって責任範囲が異なり、関心事も変わってきます。したがって、相互に矛盾した要求が衝突（コンフリクト）を起こすこともよくあります。事務担当者は懇切なシステムで自分の仕事の労力を削減したいと思っていても、経理部門は、システム化に費用をかけるくらいなら残業コストをかける方がいいと考えるかも知れません。多くのステークホルダーの要求を調整して、全体最適の結論を出すのは、容易なことではありません。

ことに非機能要求については、サービスレベルとコストの間に密接な関係があります。しかも、必ずしもわかりやすい比例関係に従いません。

1日23時間だった稼働を24時間停止なしのレベルに引き上げれば、コストは4～5%増しどころではなく、2倍、3倍に膨れ上がるかもしれません。そうすることで、夜間メンテナンスやサーバーのシャットダウンなしで稼働する設計にしなければなりませんし、常時バックアップ、常時運行監視の仕組みも必要になってくるからです。

どこまでのサービスレベルが必要か、という判断には、ビジネス（事業）という側面からの視点が不可欠です。際限なくコストをかけることはできませんから、「このビジネスはどういう性質のものか」「顧客にとって、会社にとっての重要度をどう評価するか」「障害や不正があった場合の影響やリスクはどの程度か」「コンピューターが使えないような非常の場合の事業継続はどうあるべきか、代替の継続策はどんなものが考えられるか」といった判断が必要になるのです。

したがって、非機能要求を固めるためには、「日常業務に詳しい人」だけではなく、「ビジネスの観点から重要度を判断できる人」「顧客の視点で見られる人」「予算の策定や執行に責任を持てる人」「必要な内部統制やセキュリティのレベルを判断できる人」など、様々なステークホルダーの参画が必要になります。

このケースで「関係者が一堂に会した」ミーティングを企画したのはこのためです。個別にそれぞれのステークホルダーからヒアリングしても、それぞれの責任・関心事からの意見が出るだけで、全体最適の方向に話が向きません。言い分をぶつけ合い、制約事項やトレードオフ事項を確認し、諦めるべきことは諦めながら、合意を形成する場が必要です。

非機能要求グレードは、開発担当者や事務担当者だけのためのものではありません。経営層を含めて、事業の性質を決めるためのツールなのです。

1.4 事業のビジョンを決め手にジャッジする

ある企業が新規ビジネスを立ち上げようとしている。同社はニュース配信に始まり、現在では多種多様なネットコンテンツを会員に提供し、スマートフォンやタブレットといったモバイルギアの普及とともに急成長を遂げた企業である。先日、同社は家庭向け超人気ゲームのオンライン化についてゲームメーカーと合意した。そのシステム化検討会には、太田システム部長、プロジェクトオーナーの高橋コンテンツ事業部長、松田財務部長、ゲームメーカーから来ている小川課長が顔を揃えている。後ほど石田社長も参加予定だが、会議は踊るが進まずの状況のようだ。

「誰でも知っているビッグネームですよ、我が社のキラーコンテンツになることは間違いない。ゲームの内容や操作性といった機能面については家庭用ゲーム機をそのまま移植したいと考えていますので、是非、十分なシステム基盤が欲しいんです。」

と高橋事業部長がシステム基盤の増強を主張すると、松田部長が

「それじゃコストがかかりすぎるんじゃないか。この手のゲームのファン層は一人でやり込むことを好むんだから、家庭用ゲーム機版と併売となる今回は、オンライン版にそれほど大きなニーズがあるとは思えないよ。」

とコストの圧縮を主張した。小川課長などからは揚げ足を取ったような発言もあった。

「そうです、やり込むユーザーが多いので、是非、ゲームデータの保証は万全にお願いします。」

このようなやり取りが散発的に行われる状況で、太田部長が切り出した。

「今回のプロジェクトは、当社始まって以来の大プロジェクトになります。ただ、一方で財務部長のご心配ももっともだと思います。皆さんの要望や問題意識は理解できますが、具体的な実装を決めるためには網羅性とバランスが重要です。今回は、『非機能要求グレード』というツールを用いてシステム基盤のあり方を議論し、合意形成を図りたいと思います。」

太田部長は、16項目からなるモデルシステムシートを配布した。

「最終的にはもう少し細かい粒度で非機能要求を定め、開発ベンダーと共有する必要がありますが、それはシートにあるような主要な非機能項目を詳細化しただけです。まずは、お配りしたシートを用いて各非機能要求について大きな方向性を合意させてもらえますか。」

その提案で議論に流れが出てきた。

「新規構築なので、移行性に議論はないね。」

「ゲームデータの保証は大事。セカンダリーサイトに同期バックアップを取ろう。」

議論が性能・拡張性に及んだ頃、石田社長が会議室に現れた。

「今回のプロジェクトはゲームコンテンツを専用ハードの縛りから解放するものなんだ。それだけでも膨大なアクセスを想定するべきだし、今後はこのゲームの過去のシリーズも手掛けてゆきたいと考えているんだ。システム基盤は十分なものを用意してくれ。」



財務部とコンテンツ事業部で折り合いのつかない部分だったが、石田社長のビジョンが示されたことで、システム基盤の方向性が固まりつつあった。

解説

非機能要求グレードが提供するツール、特にシステム基盤の非機能要求に関する項目一覧などを見ると、要件定義として情報システム部門の専門家が開発ベンダーとひざ詰めで一つずつ決めていくようなものに見えるかもしれません。しかし、非機能要求グレードでは、企画プロセスから開発プロセスの各段階に応じて利用可能なツールを提供しています。

利用ガイド[利用編]では、情報システムのシステム基盤の非機能要求を段階的に詳細化することを基本的な利用法として示しています。

グレード表の先頭にあるモデルシステムシートでは、自社で実現したいシステムのイメージ（大まかな非機能要求の状態）に最も近いモデルシステムを選択し、それ以降の詳細化のベース値とすることを想定しています。ここで、それぞれのモデルシステムの名称として「社会的影響がほとんどないシステム」「社会的影響が限定されるシステム」「社会的影響が極めて大きいシステム」を用いていますが、実務での利用の際には、定義の名称にこだわる必要はありません。社会に存在する多くのシステムを網羅でき、三つのサンプルを提供していると捉えていただき、自らの組織に合わせて、新しいモデルを作成していただいても構いません。

見ていただいたケースでも、移行性については「社会的影響がほとんどないシステム」の特徴を、バックアップでは「社会的影響が極めて大きいシステム」の特徴を持つシステムとしてコンセプトが形成されています。

システムの新規構築にあたっては、このようにコンセプトを定めたところで、グレード表を利用し、コンセプト化したモデルシステムのベース値を参考に 92 項目のコストへの影響や、途中変更の手戻りリスクが多いと想定される重要項目のレベルを決定し、最終的には、項目一覧を用いて非機能要求の 238 項目からなる全項目について要求レベルを決定してゆくという段階の詳細化を基本的な利用法として想定しています。

この詳細化の過程でもう一つ留意していただきたいことは、詳細化の段階で決定に関与するステークホルダーについてです。非機能要求を検討する、もしくは決めることは、社内のインフラ担当や開発ベンダーだけの仕事であると考えていないでしょうか。非機能要求グレードはシステム構築プロセスの企画プロセスから要件定義プロセスを主な守備範囲として策定されました。企画プロセスにおける検討には、社長を含めた経営層、関連部門を含めた業務部門が積極的に関わってゆくことが肝要です。モデルシステムを選定することは非機能要求における企画プロセスに該当すると言えます。

情報システム部門はこれらのステークホルダーを巻き込んで、経営層の経営理念、事業展開のビジョンを反映させて非機能要求を決めていくことが重要になります。

1.5 アーキテクチャーの選択肢を持つ

SEの安藤が眉間にしわを寄せ、難しい顔をしている。おかしい、先日、非機能要求グレードを紹介した時は、我が意を得たりと言わんばかりの自信に満ちあふれた顔をしていたのに。気になった上司の伊藤が様子を見に行くと、

「すみません、聞いてもいいですか。」

と待ち構えていたように話し始めた。案の定、何か悩んでいたようだ。

「非機能要求グレードを使ってメトリクスを決めたとして、それによって具体的にどうやってシステム基盤のアーキテクチャー（実現方式）を決めるのでしょうか。なんというか、こう最後のマシンスペックなんかにつながるところがわからないんですよ。レスポンスタイムの順守率やユーザー数が決まってもそれでCPUやメモリがいくつになるかわかりませんよね。ラストワンマイルのところがつながらない気がして、結局意味がないんじゃないかな、と思っているんですよ。」

安藤は腑に落ちないという顔をしている。

「なるほど、ラストワンマイルか。うまいことを言うな。確かにつながっていないように見えるがこれには理由があるんだ。実は、決めたメトリクスというのは、言わば業務アプリで言う要求、要件定義なんだ。だからそこからいきなりアーキテクチャーが決まることはない。業務アプリの要件定義も、その後設計作業があるだろう？同じようにここからインフラ設計をする必要がある。決めた要求を基にして設計を行って、初めてアーキテクチャーが決まるというわけだ。」

なるほど、そう言えばそうだ。非機能“要求”、最初から要求と書いてある。

「そこから先はハードベンダーやインフラ担当の設計能力がものを言うということですね。確かに業務アプリの要求と設計の関係と同じですね。」

次の日、安藤はインフラ担当の鈴木と打合せをすることにした。

「今回のシステムの非機能要求をグレード表にまとめてみたんだけどどうだろう？」

安藤が切り出すと、鈴木はこれまでと全然違っていた。

「うん、ここまで詳細に要求がまとめられていると次のシステム方式設計もやりやすくなるよ。」

とかなり上機嫌だ。

「実はインフラグループでは、最近のハードからいくつかのパターンを洗い出して標準構成を作っているんだ。後は細かいスペックを決めてやれば大体のハード構成は決まるよ。」

「えっ、そんなお手軽で大丈夫なの？」

いつもシステム方式設計では苦勞していたはずだが、その後の打合せで、あっという間にアーキテクチャーが決まっていた。

「せっかくグレード表で要求をまとめたんだから、これをシステム方式設計の網羅性確認やシステムテストのチェックリストにも使うと良いよ。」

鈴木は自信に満ちた顔で話してくれた。



標準構成を使ってハードがうまく決まったように見えるけど、やはり当初の要求がきちんと実現できるかが気になります。安藤は、既に一覧表形式にまとめた要求表があるので、それぞれの要求が設計につながっているか、要求のレベルがきちんと実現されているかを確認できるように、活用シートを加工して、要求をトレースするための設計充実度確認シートと、システムテストで要求が実現されているかを確認するためのチェックリストを作った。

安藤はもうしっかりと「非機能要求グレード」を使いこなしたようだ。

解説

非機能要求グレードに限らず、非機能要求というものを初めて知った時、「これは役に立ちそうだ。」と安藤さんのように飛びつくケースがあります。そして、一生懸命非機能要求を決めた後に、どうしたらいいか途方に暮れることがあります。それは、非機能要求を決めれば、アーキテクチャーが自然と決定する、と勘違いしてしまっているからです。まるで魔法のつえや銀の弾丸のように過度な期待をしまっているのです。当然そんな便利なものはこの世にはありません。

非機能要求を決定したということは、あくまで要求を整理できたということです。それ以上でもそれ以下でもありません。

そこを理解しないで飛びつくと、悩んでしまうことになります。その先を考えずに非機能要求をまとめた資料を作成しても、結局活かすことができずに、非機能要求は役に立たないと勘違いし、その資料は有効に使われなくなってしまいます。

しかし、本来、非機能要求の情報は特に、システム基盤にとって大元となる大事な情報です。業務アプリで言えば、まさに要件定義書と同等です。これがなければ設計を始めることはできないのです。

インフラ設計者はこの要求を基に、数あるアーキテクチャーの中から最善の組合せを選び、非機能要求グレードで決定した要求を満たすシステムを設計、構築します。工夫次第で、設計、構築、テストのそれぞれの工程で、いつでも本来の要求に振り返って確認することもできます。

あくまで非機能要求とは要求、要求だということを忘れないでください。そして非機能要求グレードは、その作業を行う際に役に立つ強力なツールなのです。

また、後半に出てきたような標準構成は、たくさんのシステムの企画を担当している場合には是非作ってみてください。きっとその効果を実感できると思います。

標準化については4章に詳しく載っていますのでこちらも参照してください。



1.6 より短いサイクルを繰り返して要求の成熟を狙う

サイバーステッチは、老舗アパレルメーカーが4年前に立ち上げた、オンラインでオーダーメイドできるシャツのブランドである。実店舗はわずか2店舗であり、しかもその場での販売は行わない。あくまで採寸や注文のみを受けるにとどまり、商品見本を体感してもらう位置づけとなっている。実店舗にかかるコストを高品質な素材の確保に振り向けてきたことにより、そのコストパフォーマンスの良さからメディアでも繰り返し紹介され、一気に人気ブランドとなった。特にドレスシャツのオンラインカスタマイズは急激な伸びを見せており、工場生産が追いつかないため、3か月待ちの状態となっている。

そのような状況の中で、注文用の Web サイトの開発と運用を担当している情報システム部では矢継ぎ早のシステム改良への対応に追われていた。部長の若岡は、いつものように浮かない顔で担当の三沢に声をかけた。

「三沢君、少し時間いいかな。」

不意を突かれた三沢は編集集中の文書を誤って削除してしまうところだったが何とかセーブしながら振り向いた。

「これでよしと。部長、なんでしょうか？ また無理難題ですか？」

決して本気で怒っている訳ではないが、いつになく予防線を張ってきた三沢に、多少たじろぎながらも若岡が言葉を継いだ。

「いや、まいったな。僕だってそんないつも無茶を言っているつもりもないんだがねえ。そうそう、やっと縫製工場の増強の目途も立ってきて、受託の拡大ができそうになったらしい。」

「ではようやく縫製の方もオーダーのペースに追いつけそうだということですね。季節が変わるまでお客様を待たせることもなくなるかな。」

三沢は、自分の努力も含めて報われそうな話に目を輝かせて答えた。若岡はつい先ほどとは正反対の三沢の表情に安心しつつも、指示を出すなら今だ！と思った。

「そうなんだ、ただ我々にもまだ課題があってなあ。」

「我々のほうにでしょうか？ レスポンスタイムの課題にも対応しましたし、お客様が過去の発注履歴を参照できるように改造も済ませたばかりですが、トラブルでも・・・」

「いや、いつも三沢君の対応は完璧で申し分ないよ。」

「それでは、どんなところが・・・」

三沢の顔が一気に曇った。若岡が続ける。

「実は、ドレスシャツの件なんだ。一般のビジネスシャツ以上にカスタマイズ箇所が多いだろう？ 同時に指定できないオプションがあったりもするし、色の指定もかなりバリエーションが出てくるから、注文時のエラーの意味が分からないというお客様が増えているというんだ。」

「確かに、ドレスシャツの FAQ の占める割合は圧倒的に多いですね。ほかの商品の 30 倍以上の件数はあったかと記憶しています。」

「そうなんだ。FAQ そのものが多くなってきたこともあって、ユーザーが見つけにくくなっているんだよ。メールや電話での問い合わせも急増していてね。これではシステム化したのに本末転倒だという人もいる。ついては今回 AI を活用したチャットボットを試験的に導入したいと思っているんだ。」

三沢は恐れていた事態の到来に天を仰いだ。

「わあ、いずれ来るんじゃないかとは思っていましたが、こんなに早く来ましたか……。次の…2年後のシステム更改のタイミングでの導入と考えていいですか？」

三沢の妥協案を若岡が一蹴する。

「いやそんな悠長なことは言ってられないよ、まずは現行システムに並行してサブシステムの形でトライアルしたい。まずは最初の試験を3か月後に実施する。」

「3か月後に試験をしてうまくいったら本稼働ですか？」

「いや、今の段階ではどのくらいのシステムリソースが必要なのか、どのくらいのデータがないと実用の精度に達するのかよくわからない。3か月ごとに仮説・検証を繰り返す形で要件定義を詰めていき、途中でも開発中と断ったうえで多くのお客様にも使ってもらうつもりなんだ。」

外堀は埋まっている。三沢はそう判断した。

「わかりました。まずはチャットボットの動作環境の調査からかかります。」

解説

ビジネスの速度や、ビジネスそのものがおかれている環境の変化が激しくなっています。これまでは3年から5年をシステムのライフサイクルとしてきたものが、1、2年となったり、正式なシステム実装の前に試作せざるを得ないケースも出てきたりしています。

こうした状況に対処するため、期日を決めて段階的に要求を詳細化したり、要求のレベルに確証が持てない場合に、一旦仮決めしてしまい、プロトタイプを作成して妥当性を検証するといったことも増えてきています。

ただしこのような手法は行き当たりばったりに要件定義を進めるということではありません。どうしてもわからない、あるいは確証をもって決められないポイントに限ってレベルを仮決めし、検証と仮定を繰り返すことですみやかに要件定義を収束させていく手法なのです。

このような手法で重要なことは3つあります。

まず第一に、一貫して変わることはない要求を洗い出し、その要求をフィックスしておくことです。

第二には、次のフェーズで変更となる可能性のある要求であっても、今回においては要求としてきちんと策定してしまうことです。非機能要求グレードを用いる場合は、変化しない要求と、次では変化する可能性のある要求をマークしておくといいでしょう。

第三には、次世代のシステムに移行するなど、時間の経過とともに要求が変化した場合は、どのメトリクスにおいてどのような背景で、今回のような要求の変化が発生したのかを記録しておくことです。

このような手順を経ることで、常にどの世代の要求も明らかになっていることになります。

短いサイクルで刷新を繰り返す手法を用いる開発においても、ライフサイクルが変化しただけであり、非機能要求定義の手法は変わらないと言えます。

1.7 スモールスタートを前提に要求をまとめる

全国でレストラン事業を展開しているカナリーフーズは、時代の要請に合わせてアイビスデリというブランドで宅配事業、つまりは仕出しの事業に参入することを決定した。この日、本社 24 階の大会議室は朝から緊張した空気につつまれていた。各組織からこの事業に関連する面々が集められたのだ。担当の執行役員、事業企画の担当者、営業に加え、システム部門からは設計開発や運用管理の担当者が顔を揃えていた。さらには社外からシステムアーキテクトも参加している様子だ。体格のいい男性が立ち上がると低い声で口火を切った。

「企画の中山です。まずは中村執行役員から、一言頂戴したいと思います。」

指名された中村は参加者の顔を一通り見回したあと、静かに話し始めた。

「今日はみなさんの意見を伺う場で、中村の方針をお話しする場ではありません。そうは言っても、いよいよこの事業の成否を左右しかねない会議ですから。みなさん存分に議論いただきたいと思います。経理からはね、こんなご時勢だからコストをかけないようにとか言われていますが・・・あ、余計なことを言ったかな。そういうわけだから、みなさん頼みますよ。」

誰もが、企画の中山の次の一言を待っていた。ところが、待ちかねたように一番若い男が話し始めた、システム運用管理を担当している青木だ。

「システム運用管理の青木です。最初に言っておきますが、今回は、ちゃんと最初から運用管理もキッチリ予算付けてくださいよ。そうでないともう回りませんから。24 時間 365 日とか一方的に無茶言うのも勘弁してください。ちゃんとやりましょ。ちゃんと。はい、以上。」

長年、営業の現場で汗をかいてきた馬場は青木の一方的な言い分に怒る様子もなくニコニコしていたが、明らかに目は笑っていない。

「レスポンスが悪ければお客様はすぐ他へ行ってしまうからねえ、性能は重要です。代金引換も一般的ではありますが、イタズラばかり発生するようでは困りますからクレジットカードも積極的に活用したいです。会員制にしてカードはレストラン事業でも使えるようにしたいですねえ。まあ今回は実証実験ですから、事業化するかどうかは、そう 6 か月後になりますか、その後で改めて決めるということで。そうは言っても、事業化する際にまたシステム構築とか言っているとタイムラグが出てしまいますから、最初から事業化を前提とした作りにはしておいていただきたいですねえ。ははは。」

まくし立てる馬場に、中山が必死に割り込んだ。

「馬場さん、ちょっと待ってください。クレジットカードとなるとセキュリティ対策も相当必要になりますよね。それに万一事業化しないという決定になったらシステムはどうします？」

馬場の目が丸くなった。何を言っているのか、とでも言いたげな目だ。

「そんなもの、レストラン事業でシステムは使えばいいんじゃないの？」

「はあ？」それまで大人しく聞いていたシステム開発担当の吉田も、さすがにたまりかねた様子で声を荒げた。



「法的な制約もあるので高いセキュリティは必須ですよ。セキュリティと性能とはトレードオフな部分もありますし、相当なコストもかかります。そこまで作らせといて、もし使わなくなったら他に転用してくださいなんて・・・無茶ですよ。何だと思ってるんですか！」

さらに運用の青木も畳みかける。

「さっきから聞いていれば、レスポンスは重要だ、システムは止めてくれるな、セキュリティも最高レベルでやってくれ。言いたい放題ですね。今回は実験レベルじゃないんですか？」

「わかっているのかなあ。社運がかかってるんですよ、これは。」

馬場もさすがに抑えきれない様子だ。頬が紅潮している。険悪な空気を払うように新参者のアーキテクト横井が口を開いた。

「いやいや、みなさんちょっと待ってください。ええと資料を拝見すると実証実験段階では見込みユーザー数は5,000人、配達のエリアも限られています。まずセキュリティ面ですが、ユーザー登録の時点で住所と電話番号と氏名をいただく事になっていますから、最低限の抑止というリスクヘッジにはなりそうですよね。実験段階ではクレジットカードの扱いはちょっと待ってもらうことにして、事業化の時に検討しませんか？」

それから、最初から全国展開を前提にシステムを作るとなると、投資もそうですが、設計・構築の規模も膨大になります。まずは身の丈に合った要求でスタートしましょう。ただし馬場さんの事業化にかける熱意はよくわかりましたから、システムが成長していけるよう『拡張性』に十分配慮するということではいかがでしょうか？」

会議室に冷静な空気が戻った。馬場も吉田も青木も、すっかり毒気を抜かれたようだ。

「拡張性ですか。なるほど、会員数やオーダーが増えてきたらシステムを增強できる仕掛けにしておこうということですか。それならリスクが少ないですね。営業としては文句ありません。」

「運用としても賛成です。今一度、中山さんにも協力してもらって、ほら非機能要求グレードとそのレベルも使いながらみんなできちんとした意識合わせをしたいです。」

会議を主催した中山は、ほっとしたように額の汗をぬぐった。

解説

非機能要求グレードについて、ユーザー（発注者）とベンダー（受注者）というステークホルダー間での利用がよくとりあげられますが、実は同じユーザーの中でも立場によって要求が大きく異なることがわかりただけたのではないのでしょうか。

上記の事例では同じ業務を実現するためのシステムに関わるユーザーの中でも、それぞれの立場から要求を発していました。しかしながら、カットオーバーまでの時間やコストの制約などにより、ステークホルダーの要求をすべて実現するのは現実的ではありません。プロジェクトの制約条件に合わせて要求事項に優先度を設定し、優先度の高いものからスモールスタートで実現していくことが重要です。

このようなケースでも非機能要求グレードを活用することができます。各ステークホルダーの要求内容を非機能要求グレードの各項目の要求レベルとして可視化し、お互いに認識合わせをした上で項目に優先度を設定します。優先度の低い項目については要求レベルを下げるなどの見直しを行い、スモールスタートに適した要求内容に集約していくことができます。

1.8 トレードオフ要素に優先順位をつける

小橋川トレーディングは、先代の社長が海外で良質な商品を安価で買い付け、国内で販売することで大きく成長してきた中堅商社である。現在も同様のビジネスを行っているが、あつかう品目の拡大やロット購入にも対応できるよう流通倉庫の拡大をしながら業績を伸ばしている。

9年前にオンライン販売に事業を広げたのは、バイヤー経験もあり、その後は販売部門の長を務めたこともある松井事業部長である。20年ほど前、商品管理のコンピューター導入を主導した経験からシステムにも明るいと自負している。

今日はその松井事業部長のもとに情報システム部の飯野部長が新システムの要件についてヒアリングに来ている。

「無論、運用時間は24時間無停止を考えているよ。」「それから、ネット上に公開するシステムなので常に最新のセキュリティも確保しておきたいね。」

という事業部長の発言に対して、飯野部長が確認した。

「24時間無停止はかなりハードルが高くなります。それなりの設計・実装が必要になりますから。最新のセキュリティを確保するためには、全てのサーバを二重化して、交互にパッチを適用する運用を考えるか、セキュリティパッチなどを適用するためのメンテナンス時間を確保しておくなどの検討が必要です。コスト面から考えると全てのサーバを二重化するのは現実的ではないと考えています。」

「ちなみに24時間無停止と最新のセキュリティ確保とどちらの重要性が高いですか？」

「どちらも重要なファクターだから死守したいね。」

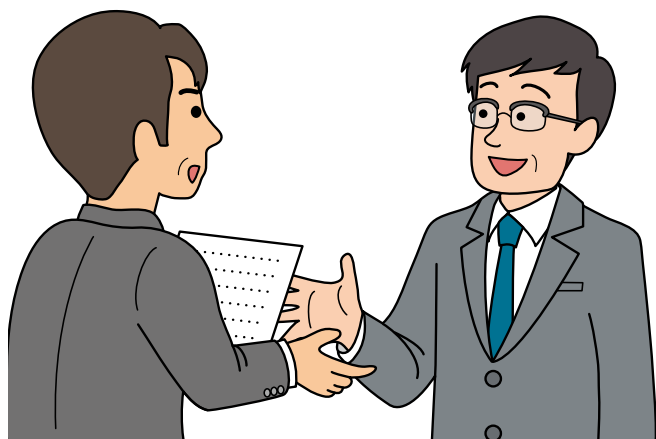
「おっしゃることはわかりますが、メンテナンスのための時間を確保して適切なセキュリティパッチを計画的に適用することがセキュリティを保つためには重要なことなのです。」

松井事業部長は渋面を浮かべた。

「しかしユーザーが何か買おうと思ってアクセスしたらシステムが止まっていたでは商機を逸してしまう。」

飯野部長が考えながら言う。

「例えば、毎月何日の早朝は3時間だけメンテナンスのためサービスを中断するということがわかっていると、停止する場合はあらかじめ1週間前にアナウンスをすればお客様にも納得いただけるのではありませんか？ 24時間無停止にしようと思えば様々な方策を行わなければなりません。設計・実装のコストも増加しますし、運用にも気を遣います。」



「そういわれてもいつでも欲しいものが手に入るというのはうちのオンラインの最大のメリットであり、アドバンテージだからね。」

「なるほど、こうお考えになったことはありませんか？ システムを停止しないことでかかるコストは莫大になりますが、毎月一定時間のメンテナンスタイムを設けることでシステムへの投資はかなり削減ができます。つまりその分商品をより安価にお客様に提供できるわけです。逆に24時間無停止にするためにかかったコストは商品に転嫁することになりませんか？ お客様はどちらが優先だと考えられるでしょうか。」

松井事業部長は目を丸くした。

「君、すごい発想をするね。そんなふうには考えもしなかったよ。ユーザーにより安価な商品をより安定したサービスとして提供するためにシステムはきちんとメンテナンスされるべきだというんだね。」

「おっしゃるとおりです。さらに一点お伺いします。早朝4時にログインされて購入をされているお客様はどのくらいおられますか？」

松井事業部長は苦笑いした。

「いや、わかったわかった、飯野さんにはかなわんね。通常は早朝に若干の停止を許容することにしよう。計画停止も7日以上前にアナウンスをしたうえで許容することにするよ。」

「かしこまりました。こうした時間はバックアップにあてられたり、システムにパッチを当てたりといったメンテナンス作業にスケジュールしていきます。後日運用スケジュールについても当部のシステム運用管理担当と一緒にヒアリングさせていただきます。」

解説

非機能要求についてはコストの制限があったり、トレードオフ関係にある条件を同時に満たすことができなかったり、システムに対する要求の重要性に基づいて優先順位を決めなければならない場合があります。

上記の例では限られたコストの中で「運用時間」と「セキュリティパッチ適用」のどちらかを諦めなくてはならない状況となっています。24時間無停止の要求については、システムの安定した稼働やコストへの影響を考え、計画的なシステムの停止を許容するなどの施策が必要であることを理解した上で、優先度を下げようとしています。

「とにかく止めるな」などのような硬直的・一面的な要求表現に固執すると、最適解が見えなくなる場合もあります。

例えば、「このサービス時間帯にはできるだけ止まらないようにしたいが、万が一止まっても復旧時間は何時間以内にとどめたい」「より安定した稼働環境の整備のため、定期的にメンテナンスができる時間を設けたい」といった、より現実的で実態に合った表現を選んで、本来の要求レベルをとらえていくことも重要です。

本当にその要求が、提示された表現通りに必要なのか、他の要求に優先すべき事項なのか、費用対効果を考えた上で、要求を見直し調整していく必要があります。

1.9 クラウドサービスへの要求を整理する（クラウド）

最近、鈴木食品では、情報システム部だけでなく、業務部門を含め全社的にクラウドシステムの導入を検討している。情報システム部の高橋部長は、マーケティング部の山下部長から最新の販売分析クラウドシステム導入の検討を依頼され、数か月後には利用開始したいと急かされており、部下の山田係長に相談した。

高橋部長が口を切る。

「山田君、マーケティング部が勧めてきた販売分析 SaaS 型のクラウドサービスだが、ユーザーインターフェースが非常に使いやすいようだ。また、多少のプログラム修正は、画面のドラッグ&ドロップ操作だけで実施できるようでメンテナンス性も優れている。クラウドサービスなので、契約後すぐに開発に着手できて導入期間を短縮化できる。しかも、システム運用も全て任せることができるので、当社の運用の手間、工数も削減できる。採用する方向で検討しようと思うが、どうかな？」

「なるほど、当部門としてもシステム運用負荷が軽減されるのであれば、嬉しい限りです。マーケティング部の要求を満たせるのか早速仕様を確認し、問題なければ採用する方針で検討を進めたいと思います。それにしても最近は本当にクラウドのサービスが増えてきましたね。」

同社では、導入スピード、運用負荷軽減といったクラウドのメリットに注目し、クラウドの採用検討を進め、まずは、検証してみることにしたようだ。ただ、検証中に色々と問題が発生しているらしい。

山田係長が高橋部長に声をかけている。

「部長、先日検証を開始した SaaS 型のクラウドサービスですが、少し問題が発生しています。」
「どういう問題かな。」

「このクラウドサービスは、画面操作だけで開発できるのですが、項目を加えると、特定のブラウザでは非常にレスポンスが悪化するようです。クラウド事業者に仕様を確認したのですが、現時点では、その仕様は改善できないとのことでした。当社としてもクラウド内で実装されているプログラムには踏み込めなくて困っています。」

「そのあたりは、事前に確認していなかったのかね？」

「はい、スピード優先で検討を進めていたため、事前の確認が足りませんでした。実は、ほかにもいくつか問題が発生しています。保存できる



データの容量は無制限とのことでしたが、ある一定の容量を超えると、パフォーマンスが劣化するようです。しかも、データ保存容量の監視を行っていないようで、事前に検知することも難しいと言われており、困っています。」

「そうなのか、運用をすべて請け負ってくれるといっても、どの程度のレベルで運用しているのか、事前に確認していないとまずかったということかな。では、非機能要求グレードを活用して、早速見直しにかかってくれ。場合によっては、別のサービスを検討した方が良いかもしれないな。」

「ただ、すぐに利用を中断しやすいのも、クラウドのメリットかもしれないですね。早速、非機能要求の見直しに取り掛かります。」

解説

昨今、様々なクラウドのサービスが提供されています。サービス形態も、IaaS、PaaS、SaaS など様々です。今回のような、運用を含めたシステム全体をサービスとして提供するような形態もあれば、API のみを提供するような形態もあります。

いずれのサービスを利用するにしても、クラウドサービスの場合には、システム運用の形態や仕様があらかじめ決まってしまう可能性もあり、自社の要求に合わせて機能および非機能要求を実現できない場合もあります。一方で、導入スピードの短縮化、運用工数・費用の削減などのメリットも考えられます。

クラウドサービスの場合には、その仕様が自社の要件に合っているのか非機能要求グレードなどを利用し確認した上で、各要求レベル、コスト等を総合的に評価して導入可否を判断する必要があります。

今回の事例では、パフォーマンスや運用監視の要求を取り上げましたが、基幹系システムにおいては、可用性、セキュリティなどもより重要な確認項目になるのではないのでしょうか。特に昨今セキュリティ対策については、各企業とも重要な経営課題の一つとしてとらえています。そのため、クラウド事業者が、どの程度の要求レベルを目指してサービス提供をしているのかは、採用可否を判断するための非常に重要なポイントになります。

このようなビジネス上の重要ポイントも、非機能要求グレードを活用することで、確認することが可能です。クラウド事業者のサービス内容を評価し、必要な追加対策を講じておくためにも利用すると良いでしょう。



1.10 スピーディに概算で見積もる

株式会社ナレッジシステムの開発部門の部長である斉藤の席に、営業部長である林がやってきた。「斉藤さん、念願だった千石商事から、昨日、次期会計支援システムの RFP をいただくことができたんですよ。概算費用の見積り提示が2週間後なので、あまり時間がありませんが、基盤の見積りをお願いできないですかね？」

「もちろん、協力させてもらうよ。」

忙しい時期ではあったが、斉藤はふたつ返事でこれを了解した。これまでの営業の苦勞を知っているからだ。

「ところで、今回はハードやソフトの物品も見積り対象なのかな？」

「そうなんです。ある程度、システム構成の想定を作らないといけません。たしか、非機能要求グレードでしたっけ？ 概算見積りを提示する前に、お客様に質問票を送ることができるので、あれを埋めて欲しいとお願いしようかと思っています。」

おっと危ない、と斉藤は思った。お客様のことを考えると、林の考えるやり方はいささか乱暴すぎる。

「いやいや林さん、非機能要求グレードの全項目をお客様に埋めてもらうのはダメだよ」

「え、ダメなんですか？ ご自身で埋めていただくのが一番確実だと思ったんですが。非機能要求グレードは既に広く使われているって聞いているので、お渡ししても問題ないんじゃないかと。」

林は合点がいかない表情で、斉藤を見た。斉藤は頷いて答えた。

「確かに記入してもらうことで、非機能要求を明確にすることはできる。ただ、非機能要求グレードには 238 の項目があって、これを提案段階で全部埋めてくれって言われたら、お客様もつらいと思わないかい？ 要件定義工程で決めれば良いと考えているかもしれないしね。」

「ということは、概算見積りを提示する場合、全部の項目を明確にする必要ないってことですか？」

思ったより飲み込みが早い林に、斉藤は安心した。

「そうそう。非機能要求グレードではコストインパクトが特に大きい項目を重要項目としてマークされているんだけど、それでも 92 項目くらいあるので、まだ多いよね。そこでだ……」

斉藤は机の引き出しの中から概算見積り向け必須項目一覧表を取り出した。A3 版一枚もののシートだ。

「以前に社内で検討したことがあるのだけど、概算見積りレベルであれば、システムの大まかな特徴が分かれば、あとはベンダーの我々がシステム構成を提案することができるんだよ。例えば、業務のピーク時のトラフィック量は秒間あたり、数件ですか？ 数百件ですか？ と聞けば、サーバー数台なのか、数百台のエンタープライズレベルなのか想定できるでしょ。他にも『個人情報扱いますか？』と確認すれば、個人情報保護法を順守するセキュリティ対策が必要になってくる。ということで、最低限この 14 項目くらいを聞けば良いのでは？ というのがこのシートなんだ。これらを聞くことによって、重要項目 92 項目のレベルは想定できるよ。」

社内で議論を尽くして、厳選した 14 つのヒアリング項目をお披露目できて、斉藤は内心嬉しかった。

「なるほど。では、これを記入してもらうようお願いすれば良いってことですね。」

意気込む林に、斉藤は釘を刺した。

「林さん、慌てない慌てない。RFP に書いてあるかもしれないので、まずは RFP の確認が最初だよ。」

解説

活用シートを見て、項目数に圧倒されることはないでしょうか。突然、「この全てを決めてくれ」と迫られたら、どんなユーザーだって引いてしまうでしょう。

非機能要求グレードの 238 項目のうち、品質やコストへの影響が大きいと考えられる項目は、重要項目とされており、全部で 92 項目あります。この事例のように概算見積りを行う場合においては、少なくともこれらの 92 項目については決めておくことが望ましいです。重要項目は、モデルシステムごとの目安になるベース値がグレード表や活用シートに示されています。そのため、3つのモデルシステム「社会的影響がほとんどないシステム」「社会的影響が限定されるシステム」「社会的影響が極めて大きいシステム」のいずれかを選択することで、92 項目の大まかな目安は仮定することができます。

一方で、「性能は二の次で、まずは可用性、それもデータの保全を第一優先に考える」といった、そのシステム特有の方針によって、各項目のレベルを決めることもあるでしょう。あるいは、セキュリティのように、システム個別で考えるのではなく、組織横断的な方針によって決められる項目もあると思います。

そのため、今回のケースのように、できるだけ精緻な概算見積りを提示したいという目的においては、社会的影響の大小だけでなく、もう少し詳細にユーザーから要件を聞き出したいところです。しかし、重要項目である 92 項目をひとつひとつユーザーに問い合わせても、まだ数が多いと感じられるでしょう。そこで「そのようなシステムであれば、だいたいこれくらいのレベルなのではないか」と推論できるようなシステムのアウトラインや全体方針を、ユーザーから端的に聞き出すことができれば、精度の高い概算見積りが提示できるのではないのでしょうか。

表 概算見積り向け必須項目一覧表

項番	ヒアリング項目	ヒアリング結果から想定できる重要項目	補足
1	システムの利用形態は？ ・社内システム（組織内の特定部署の利用に閉じたシステム） ・BtoB システム（企業間の基幹系業務を構成するシステム） ・BtoC システム（インターネットを通じて不特定多数の人が利用）	B.1.1 ユーザ数 B.1.2 同時アクセス数 C.1.2 外部データの利用可否 C.4.5.1 外部システムとの接続有無 F.2.1 ユーザ数 F.2.2.1 クライアント数	システムの利用形態をヒアリングすることで、システム停止時の影響範囲や、求められる性能や実施すべきセキュリティ対策がどの程度か、非機能要求の大まかな方針を想定できる。
2	システムのサービス提供時間は？ ・日中帯のみ ・24 時間 365 日（ただし月 1 程度の計画停止日は許容） ・24 時間 365 日（計画停止も許容せず、完全無停止）	A.1.1.1 運用時間（通常） A.1.1.2 運用時間（特定日） A.1.1.3 計画停止の有無 C.1.1.1 運用時間（通常） C.1.1.2 運用時間（特定日） C.2.1.1 計画停止の有無 D.1.1.2 システム停止可能日時	サービス提供時間をヒアリングすることで、運用スケジュールや計画停止の有無について想定できる。
3	どれくらい止まってもよいシステムですか？ ・年間でトータル数日なら止まってよい ・年間でトータル数時間なら止まってよい ・基本無停止。年間でトータル 1 分以下の停止時間に抑えたい。	A.1.2.1 対象業務範囲 A.1.2.2 サービス代替時間 A.1.2.3 業務継続の要求度 A.1.3.1 RPO（目標復旧地点） A.1.3.2 RTO（目標復旧時間） A.1.3.3 RLO（目標復旧レベル） A.1.5.1 稼働率 C.4.4.1 リモート監視地点 C.4.4.2 リモート操作の範囲 C.5.1.1 保守契約（ハードウェア）の範囲 C.5.2.1 保守契約（ソフトウェア）の範囲	システム停止の許容度合をヒアリングすることで、可用性に関する方針だけでなく、その可用性を実現するための対策（リモート監視、リモート操作、保守契約等）についても、その実現レベルについて想定できる。
4	大規模災害に備えた DR サイトは用意しますか？ ・用意する。 ・用意しないが、遠隔地へのデータ退避は行う。 ・用意しない。大規模災害は想定しない。	A.1.4.1 システム再開目標	DR サイトについてヒアリングすることで、コスト影響大きい DR サイトの要否や、バックアップデータの遠隔地保管の要否について想定できる。
5	オンライン業務はピーク時、秒間どれくらいの業務量を処理することになるでしょうか？ ・10tps 未満 ・10tps ～ 100tps 程度 ・100tps 以上	B.1.1.4 オンラインリクエスト件数	必要スループットやレスポンス時間をヒアリングすることで、おおむねのシステム規模（サーバ台数や各種ハードウェアスペック）が想定できる。また、極端に難易度が高い場合、PoC（Proof of Concept）のような事前検証の要否を判断できる。
6	オンライン業務のレスポンス時間は、どれくらいでしょうか？ ・5 秒～ 8 秒程度 ・1 秒～ 5 秒程度 ・1 秒未満	B.2.1.1 通常時レスポンス順守率 B.2.1.2 ピーク時レスポンス順守率	
12	順守すべき法令・ポリシーはありますか？	D.1.1 適用期間 C.6.1.1 内部統制対応の実施有無 E.1.1.1 順守すべき社内規程、ルール、法令、ガイドライン等の有無 F.1.1.1 構築時の制約条件 F.1.2.1 運用時の制約条件	法令やポリシー、政府の各種ガイドラインや FISC や PCIDSS 等をヒアリングし、その中身を確認することで、システムの大きな方針を想定できる。
13	データセンタや運用拠点について、ロケーション指定や制約はありますか？	F.2.3.1 拠点数 F.2.4.1 地域的広がり	システム開発の前提としてロケーションはヒアリングしておく必要がある。
14	クラウド利用や製品指定の希望はありますか？	F.2.5.1 特定製品の採用有無	お客様からの指定がないかヒアリングすることで、検討漏れを防ぐ。

TIPS 段階的詳細化のすすめ

非機能要求グレードは全 238 項目で構成されていますが、最初から全ての項目を決定する必要はありません。大まかな非機能要求しか決まっていない状態であれば、下図のように要求・合意事項を段階的に詳細化しましょう。

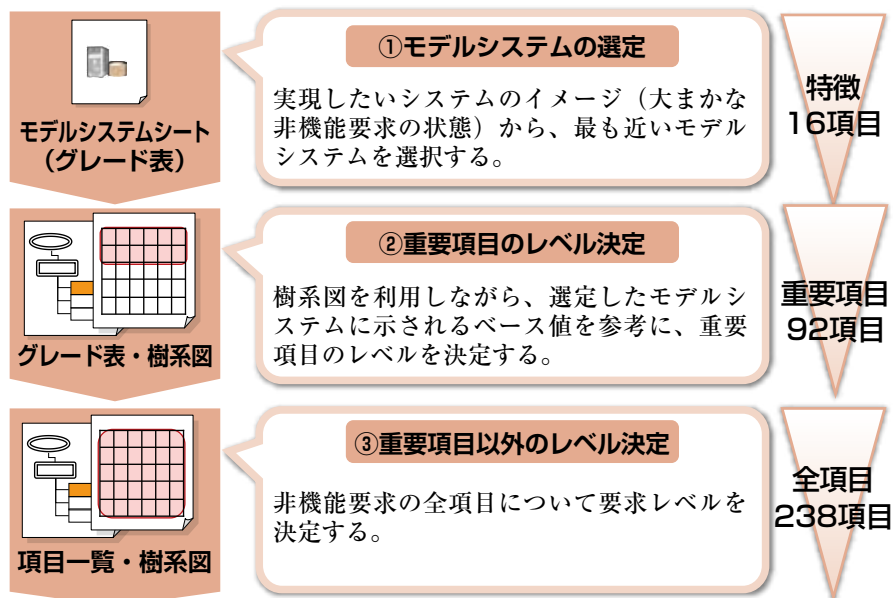


図 非機能要求グレードの段階的な詳細化

（非機能要求グレード 利用ガイド [利用編] 1 章の図を元に作成）

下図では、要件定義工程で上記 3 段階の詳細化を実施する場合の利用イメージを例示しています。非機能要求が段階的に明確になるにつれて、利用するツールも詳細なものになります。

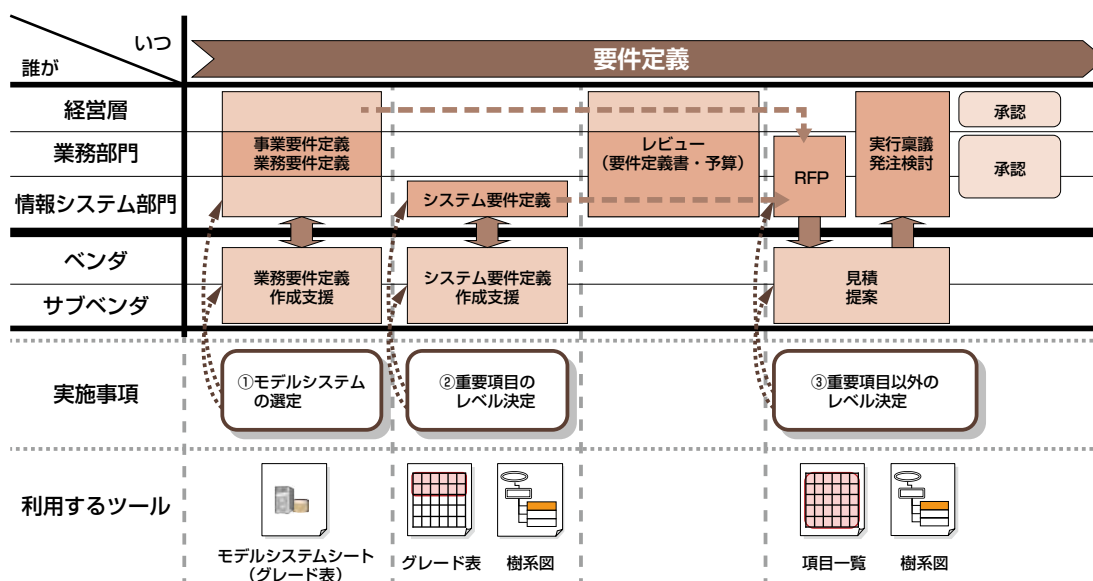


図 要件定義工程での非機能要求グレードの利用イメージ

（非機能要求グレード 利用ガイド [利用編] 2 章より転載）

2章

保守・更改

システムは、開発に要した時間よりも長い間稼働します。4～5年毎に再構築されるシステムがある一方で、10年、20年と稼働し続けるシステムもあります。機能要求に変化のないシステムであっても、サーバー等の機器やソフトウェアのサポート期間の終了によって、システム更改というシステム基盤だけが更新されるケースも増えてきました。全くの新規事業を興す場合を別にすれば、現在のシステム開発は、保守開発、システム更改が大半を占めているのではないのでしょうか。

保守・更改と一口に言っても様々な形態があります。業務機能の追加、オペレーティングシステムのバージョンアップ、ストレージ等のシステム資源の拡張、サーバーやネットワークの刷新、業務も含めた全面再構築等です。

さらに、保守・更改が行われることになった背景も様々です。ビジネスに合わなくなった、開発当初の予想を大きく上回って利用者が爆発的に増大した、メーカーやベンダーのサポート期間の終了が告げられた、予期しない障害を引き起こしてしまった、等々です。

ただし、共通して言えることは、「稼働したその日からシステムは変化し続けている」ということです。開発を担当したベテランのエンジニアが、保守を担当している現在の若手と対話した時に決まって交わされるのが「え？今そんな風になっているの？」という言葉です。この変化を定期的に捉え、常に直近の要求を反映できているシステムは、それほど多くはないでしょう。

現行システムを更改する際には、まず現行システムの機能や業務の現状（AsIs）を分析し、さらにあるべき姿（ToBe）を検討します。非機能要求も全く同様です。長年の運用によって変化している可用性要求、性能要求などを正確に把握する必要があります。非機能要求であっても「現行通り」という一言ではすみません。現状を少しでも正確に把握するために、非機能要求グレードの活用は極めて有効です。

他にも注意点があります。度重なる機能追加によって、いつの間にか稼働当初の非機能要求からレベルダウンしてしまうケースや、あるいは、機能追加によって非機能要求そのものが変わってしまうようなケースです。具体的には、機能追加によるオンラインレスポンスの劣化、暗号化など特別なセキュリティ対策が必要なデータ項目の追加などです。小規模な保守開発であっても、非機能要求に変化はないか、これまでの要求レベルを維持できるのか、といった視点で再点検することは極めて重要です。そのためにも、非機能要求の変化が定常的に把握できている必要があります。保守開発が決まってから非機能要求の現状の調査を始めても稼働に間に合わないことがあるからです。

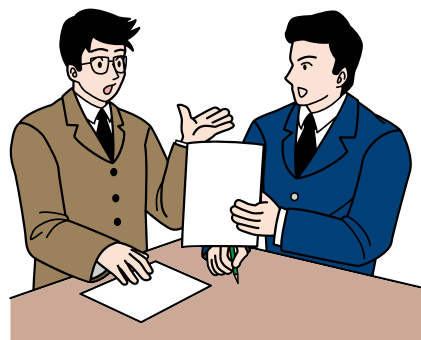
本章では、変化する要求と変化しない要求の両方を抱えたシステムの引継ぎ、網羅的に AsIs 分析を行うことによる様々な課題の洗い出し、再構築における可用性要求や運用・保守要求の変化への対応、機能要求の変化による性能劣化などのケースを取り上げます。

2.1 引継ぎを機に現状を把握し課題を洗い出す

大手金融機関で既存システムの保守を中心にキャリアを磨いてきた大宮君。ある日上司の浦和課長に呼ばれた。

「大宮君。今まで川越君に見てもらっていた市場系のシステムだが、そろそろ君に引き継いでもらいたい。」

「わかりました。川越さんも忙しいですからね。ただ市場系システムは近々更改する予定だと聞いています。」



「そうなんだ。ただ更改とは言っても業務機能に不足があるわけではない。安定して稼働しているし、余計なコストをかけて再構築するのはかえって危険だと思っている。パフォーマンスにも問題はないしね。ただ、先日の大震災の時には復旧に時間がかかった。災害対策の側面では見直しが必要だと思っている。」

「運悪くディスクがクラッシュした上に、バックアップデータが前日分しかなかったんですね。」

「それも問題だが、そもそも災害時だけでなく通常時でも障害時の目標復旧レベル、目標復旧ポイントがあいまいというか、ユーザーとの取決めもなかったんだよ。」

「文書化されてない以上、“何で元に戻らないのだ。”と言われるのは当然ですね。」

「そうなんだよ。市場系のシステムだから、日中はクラッシュ前の取引が戻らなければ大変なことになる。正直、復旧レベルは高くてもよいと思っているのだが、相応のコストをかけなければ保証はできない。それをユーザーには納得してもらって予算も確保してもらう必要がある。」

「わかりました。ところで文書化されていないのは、そのあたりだけでしょうか？」

「いい質問だ。大宮君、はっきり言おう。設計書はある。稼働当時のね。」

「……メンテナンスはされていない。直近はわからない、ということですね。わかりました。整理すると、性能面は現状を維持すればよい。可用性については、障害発生時と災害対策時の対応レベルを明確にする。バックアップ等の運用は見直す。こんなところでしょうか？」

「そうだな。大宮君、お願いするよ。」

「任せてください。非機能要求グレードを使って、まずは現状（AsIs）を整理してみます。性能は現状通りで良いとは言っても、現状値がわかっていませんからね。現行システムの稼働からは5年以上経過していますから、サーバーだけの更改ならコストダウンが可能なはずだと思います。逆に運用については強化が必要なようです。目標復旧レベル、目標復旧ポイント、目標復旧時間、どれをとっても決まっていはいないようですから、これは今一度ユーザーと話し合ってみます。ただし、その前にオンラインバックアップにした場合のコストを調べておく必要がありますね。」

「いいところに気がついたね。大宮君。市場系システムはそれなりに重要なシステムではあるが、機能追加がない更改そのものにはコストだけかかってしまうことになるから極力抑えるものは抑えておきたい。その方向で頼むよ。」

既存システムの保守を任されたのは良いが、現状が全くわからない。ドキュメントもないといったことは決して珍しいことではないのではないのでしょうか。その上、機器やソフトウェアのサポート期間の終了などによって、システム基盤だけを更改しなければならないとしたら・・・そんな時も「非機能要求グレード」が役に立ちます。

「現行通りでよい。でも、現行がよくわからない」こんな時にゼロベースであるべき姿を検討することは王道かもしれませんが、大変な手間になってしまいます。抜けや漏れをなくすためにも、まずは現状をしっかりと把握しましょう。

このケースには三つの特徴があります。

- (1) 性能については現状通りでよい。
- (2) 災害対策については強化が必要。特に復旧対策は検討を要する。
- (3) 運用については、バックアップに課題がある。

これらのそれぞれについて、非機能要求グレードには該当するメトリクスがあります。メトリクス毎のレベルを参考にしながら、まずは現状を確認し、一つ一つあるべき姿（ToBe）を検討していくのは大変ではありますが、何もないところから検討するよりは遥かに早く計画化が可能になります。

「更改は行う、ただし現行通りでよい」という場合、サーバーやストレージ等の機器については現行の後継機種で検討を進めることになるでしょう。しかし、ハードウェアの性能は日々進化しており、コストパフォーマンスは上がっているはずです。一つ間違えば過剰投資になってしまうこともあります。一方で、バックアップなど本当は対策が必要な課題があるのに見逃されているとしたら大変なことです。

このケースでは、上記三つの視点に絞っていますが、大宮君の仕事は当然これだけでは終わりません。セキュリティや移行上の問題は本当にないのか、視野を広げて点検していくことが求められます。非機能要求グレードには238項目ものメトリクスがありますが、コストに影響する重要項目はこのうち92項目です。重要項目だけでも最低限点検し、現状を正確に把握しておけば、システム基盤の更改にあたってのリスクを確実に低減することができるとでしょう。

さらに、文書化された要求を基にステークホルダーのレビューを受け、しっかりと確認を取っておけば万全です。新たな要求が出てきたり、変更を要求されたりすることもあるかもしれませんが、ステークホルダーの視点でなければ出てこない非機能要求の変更をいち早く取り込むことができるのですから、大切なプロセスと言えます。



2.2 更改時に漏れなく問題を抽出する

インターネットポータル大手のT-BA社で、人気ゲームのオンライン化プロジェクトが動き始めた頃、情報システム部では別の問題が浮上していた。同社の主力コンテンツの一つであるオークションサイトの再構築である。同社は5年前からオークションサイトを運営しているが、ハードの老朽化対応に加え、競合サイトの新機能に対抗するために、早急なシステム刷新が必要と判断されたのだ。情報システム部長の船橋、PMの成田、開発ベンダーの佐倉の3人は早速検討を開始した。



船橋部長「我が社の最古参システムも、とうとう全面更改だな。」

成田 PM「そうですね、私が引き継いでからだけでも、23時のオークション終了直前のピークの対応は綱渡りでしたし、あらゆる改善要望への対応も頻繁にやりましたからね。ハードもソフトも継ぎはぎだらけの怪物のようなものですよ。」

佐倉「機能面は良いとして、非機能については一度整理した方が良いと思います。言葉は悪いですが、場当たりの能力増強や機器の追加を行っていますので、非効率な部分があるかもしれません。」

船橋「実は、現状分析を行っていてね。成田君、説明してくれないか。」

成田 PM「今回は非機能要求グレードの項目一覧に照らして、現状把握を行い、問題点の抽出を行ってみました。時間の関係もありましたので、まず92の重要項目についてまとめたものがこちらです。」

238項目からなる項目一覧をベースに多少カスタマイズされた調査資料が配布された。調査内容によると、現状のオークションシステムの問題点は、可用性、性能・拡張性、運用・保守性の三つの大項目に集中していた。代表的なものとしては、可用性の計画停止に関する項目で、オークションシステムの性質上、本来無停止とすべきところを、毎週末に1時間程度のサービス停止が発生していた。オークション機能を導入する際に増設した専用サーバー群と既存サーバー群の同期処理のために再起動が必要となったからだ。サービスを継続したまま、新機能を追加するためには止むを得ない方式ではあった。また、別セグメントに属する各機能サーバー群を接続するルーターやスイッチといった通信経路上に単一障害点（SPOF）が存在することも判明した。単純なネットワーク設計のミスが原因である。さらに、拡張性に関しては、サーバーによってメモリの空きスロットの数やCPU性能が著しく異なっており、一部のサーバー群の拡張余力が極端に小さいことも判明した。

成田 PM「今後、項目一覧の残りの項目についても調査を行い、システム基盤全体の問題点を抽出したいと思います。今回はそれを要件定義書に反映させるプロセスを踏みます。」

佐倉「その作業に私共も参加させてください。これまでオークションシステムに関わったSEを中心に体制を組みます。正直、ここまで根深い問題があるとは思っていませんでした。万全の体制を組んであたりたいと思います。」

解説

現行システムを次期システムにリプレースする場合、必要な項目に関しては、現行システムの非機能要求を継承する必要がありますし、問題が生じている項目については、現状の把握と問題の所在、その解決策を検討する必要があります。システム更改時における非機能要求の検討は、最初に現行システムの非機能要求を確認し、次に現行システムの問題点を抽出し、次期システムに必要な非機能要求を最終的に決定するという流れになります。

最初に実施する現行システムの非機能要求の確認では、現行システムの非機能要求が明確に規定されている場合と、結果的に暗黙知の状態で実現されている場合があります。システム開発の現状を見ると、現行システムの構築メンバーが配置転換等でリプレースプロジェクトに参画していないなど、現行システムに関する理解・知識がままならない状況となっている、暗黙知のケースが多いのではないのでしょうか。この場合、非機能要求グレードの項目一覧を使用して現行システムの非機能要求を網羅的に確認することができます。

次に、現行システムの非機能要求で問題が生じている部分を抽出する場合も項目一覧を使用して、定義が必要な非機能要求項目を認識しながら、問題を正確に整理することが可能です。例えば、これまで特に問題視されていなかった非機能要求項目も一覧化することによって、複数のステークホルダーのチェックを受けることになり、重要な問題が浮上することも考えられます。

最後に、現行システムの問題点を解決しつつ、次期システムの非機能要求を決定するわけですが、ここでは、現行システムを踏襲した非機能要求と、変更を行った非機能要求について矛盾が生じないように留意する必要があります。

新規開発の際と同様に、非機能要求同士や、非機能要求とコストとの間でトレードオフが発生するケースもありますから、ステークホルダーとしっかりと議論し、要求の精緻化を図っていきましょう。



2.3 保守対象システムの現状を点検する

甲州運輸の基幹系システムは10年以上前から毎年少しずつ拡張開発を行いながら運用を続けていたが、今回の人事異動で勝沼課長がこのシステムの責任者となり、前任者の甲斐課長から引継ぎを受けた。

引継ぎ結果を改めて確認すると、このシステムは場当たりの改修が行われ、機器の更新や増強などの計画も不十分ではないかと感じた。このシステムをこのまま使い続けるためには現状をきちんと把握した上で、保守計画を作成する必要がある。今のスケジュールでは、今年度は10件ほどの要望に対応する改修を行い、年度末にリリースすることになっていた。

「でも、現状の把握ってどうすれば分かりやすくできるんだろうか？アプリケーションの機能は具体的な要望がエンドユーザーから挙がっているから、これを一覧にして優先度を管理していけば良さそうだ。それ以外についても要望は挙がっているが数件しかない。こっちの把握は難しそうだな。そう言えば、『非機能要求グレード』というツールがあったな。あれを使ってみよう。」

勝沼課長は非機能要求グレードに照らしてシステムの現状分析を行うことにした。

頼りになるのは、構築当初の要件定義書、過去5年分の稼働状況レポートくらいだったが、過去の担当者やチームメンバーなどへのヒアリングを行いながら非機能要求グレードの項目を一つずつ確認していった。具体的には設計値と現在の状態の確認。その結果がシステムの稼働状況に与える影響、今後の見通しなどである。

例えば「可用性」では、運用スケジュールは決められていたが、障害時の復旧目標があいまいで、しかも2年前に大きなハード障害によって丸一日システムが停止したことがあった。この場合、障害時の目標復旧時間については、設計値は「なし」で現状は「1営業日以内」ということになる。しかも、復旧に時間を要したのは、当時の報告書を見るとバックアップ機器に交換する判断に時間がかかっていたことがわかっている。判断基準を決めておくことで追加投資をしなくても目標復旧時間のレベルを上げることができそうだ。

勝沼課長は問題点と対策に目標復旧時間について記録し、残りの項目の確認を続けた。

一通りの問題点についての対応案をまとめた後で、優先順位を付け、今後のシステム更改計画としてまとめた。また、今回作成した非機能要求グレードに対応した要求項目については、システム保守の立ち上げ時に、開発、保守、運用を含めたプロジェクトメンバーで共有するようにした。もちろん、ベンダーに委託する際にもRFP（提案依頼書）として用いることにした。

今回、システムの現状を非機能要求グレードにまとめたので、今後はこれを本システムの実態把握と計画的な拡張開発に活かすことにした。これ以降、システム更改の要件定義書には、非機能要求が明示されることになり、システム更改作業も順調に実施されるようになった。

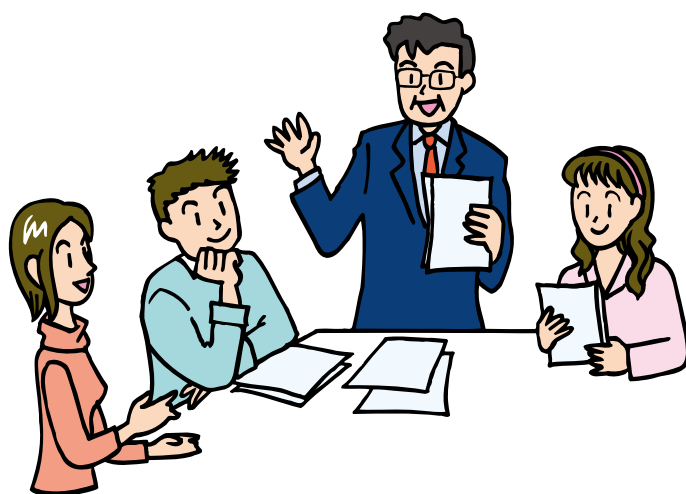


解説

システム開発を行う際に、機能要求と非機能要求が重要であることはよく知られていますが、既存のシステムで全て明確になっているかというと、そんなことはありません。過去の非機能要求が設計書としてまとまっていることは希で、さらに現状を把握するための監視データなどが不十分な場合もあり得ます。そういった状態でもシステムは稼働しますが、いざ問題が起きた時に困ります。例えば、急にシステムのレスポンスが遅くなった、設計値はどうなっているのか？今後のシステム拡張計画は？といった時に慌てることになります。

引継ぎなどで新しいシステムの担当になった時には、過去の設計資料の確認だけでなく、実際に動いているシステムをチェックして、改めて非機能要求がどのように実現されているかを整理しておくといいでしょう。この時、「非機能要求グレード」を用いることで様々な項目を網羅することができるため、確認漏れを防止するのに役立ちます。自社独自の項目を追加しておくことで、さらにきめ細かくシステムを把握することも可能です。

また、同じシステムを長期的に担当する場合も、毎年の健康診断のように確認しておくことでシステムのエンハンス計画作成や予算化などに活用することができます。システムはずっと同じように運用していても、様々な要因で変化します。利用者の増加、データの蓄積、部品の劣化、接続システムの状況など気にしなくてはならないことはたくさんあります。運用データを定期的に監視するのと同じように、システムの非機能要求も定期的に見直して、必要があればシステムの改善提案を行いましょう。



2.4 保守開発による性能低下を防ぐ

スーパー中堅の千石屋は、独自の販売管理システムで差別化を図り、大手と渡り合ってきた地域密着型の小売業である。千石屋の IT 部門に所属する川崎主任は、その販売管理システムのアプリケーション保守を入社時から担当して 5 年目になる。販売管理システムで最も利用される在庫照会機能は在庫マスターを中心にいくつかのデータベースを参照している。今回全く別の人事システムが更改される影響で担当者コードの桁拡張が必要となった。それほど難易度が高い案件ではないため、開発もテストも問題なく終了し予定通り本稼働に向けた切替えを実施した。

ところが、稼働直後にオンラインのレスポンスが極端に悪化する事態となる。早速更改前のプログラムへの切り戻し対応を行ったが、各店舗は一時大混乱となってしまう、IT 部門に非難が集中することになってしまったのだ。川崎主任の上司である戸塚課長は責任者として根本的な原因究明を求められることになった。戸塚課長は早速、川崎主任に状況を確認することにした。

「CPU の使用率が一気に上がってしまったらしい。川崎君、照会件数を増やすとか、大掛かりな対応だったのか？」

「いえ、担当者コードの桁拡張の対応だけです。他は全く変更していません。」

「テストは問題なかったのか？」

「もちろんです。性能には全く問題ありませんでした。複数の打鍵確認もやりましたし。」

「何件くらいだ？」

「10 件くらいでしょうか？テスト環境の端末には制限がありますからね。」

「え、本番並みのトランザクション量を投入していないのか？」

「朝の時間帯は照会機能が一気に使用されますが、本番環境と同じレベルの試験は無理ですからね。」

「何だって？いつからそんな状況になっているんだ。販売管理システムはウチの中核となる重要なシステムだぞ。本番と同等の試験環境が用意されているものだと思っていた。」

「稼働当初はそうでした。二重化はしていませんが、サーバーもストレージも本番と同じになっていたはずですが。ただ、稼働直後から店舗数が飛躍的に伸びていますから、本番のストレージはどんどん増強しています。でも、テスト環境の増強までは手が回らないんですよ。そのようなわけでテスト環境のマスターデータは半分くらいしか入っていません。」

後日、今回の保守開発の内容を詳細に分析したところ、桁拡張にあたって照会機能で参照しているデータベースの検索については全て修正が行われており、しかも SQL 文にミスがあり、テスト環境の 2 倍以上のマスターデータに対し本番並みのトランザクション量を投入しない限り、発見は不可能であったことが判明した。

解説

アプリケーションの保守開発によって性能が損なわれるといったことは、決して珍しいことではありません。

非機能要求の中でも、性能・拡張性については、アプリケーション開発を担当する方であっても比較的常に気にしているカテゴリではありますが、システム基盤の非機能要求と併せて点検・見直しを必ず実施できているケースは多くはないのではないのでしょうか。

項目追加や桁拡張など、一見容易な保守開発を行う際には、時間もコストも限られていることもあって、機能確認を優先してしまいがちです。このケースでは、性能要求が変わっているわけではありません。保守開発の都度、ユーザーとオンラインレスポンスのレベル確認を行っていれば別ですが、ユーザーは機能拡張によって性能が落ちて構わないとは考えていません。しかも今回のケースでは人事システムの更改の影響によるものですから、販売管理システムに影響が出ることもそのものを想定していなかったはずで

一方で、拡張性の要求は稼働当初からどんどんレベルが高くなっています。途中で本番環境のストレージを増強したようですが、統計的に増分予測ができていたのか、メモリやCPU性能まで見直していたのか、もしくは見直したがコストがかかるのでストレージだけにしたのか、疑問が残ります。

利用者が限定されていて、時間の経過によっても処理量やデータ量の増加があまりない社内システムのような場合は、本番並みのテスト環境は必ずしも必要ではありません。しかし、多数の利用者がいて、かつ増加傾向にあり、一定の性能を維持しなければならないようなシステムの場合はそうはいきません。このようなシステムの場合は可用性の要求も高い場合が多いと想定されますから、本番並みのテスト環境が必要なはずで

このケースの根本的な問題は、「非機能要求の維持」と「非機能要求の変化への対応」のいずれもできていなかったということです。保守開発によるトラブルは内在していたリスクが表に出ただけであって、拡張性の点検や、テスト環境の見直しが定期的に行われていれば、仮に保守開発でミスがあったとしても、未然に発見できていたかもしれません。

機能要求の変化に伴う保守開発を行う際は、現行の非機能要求に影響しないか、必ず確認しましょう。分析や設計局面では非機能要求グレードを利用して網羅的に確認をし、可用性要求や性能要求については、総合テストのテストケースにも組み入れ、確実に対応することが重要です。



2.5 法律や制度の改正に伴いシステムの改修を行う

消費税率引上げ時期の政府公表があった翌日、中堅商社である深川商事の情報システム部では、早速システム対応状況の確認が行われていた。部長の清澄が矢継ぎ早に担当者に質問を浴びせていく。

「今回は単純な税率変更ではない。軽減税率の導入によって、取引毎に税率を管理する必要があるが、大丈夫だろうね、森下君。」

長年、販売、仕入、会計アプリケーションを担当してきた森下が胸を張る。

「もちろんです、清澄部長。8%に上がる時点で軽減税率の議論はありましたから、消費税関連のデータベースやプログラムは全面的に見直しています。個々のトランザクション毎に税率を持つことにしていますし、プログラムに直接書かれていた税率は全て外出ししてあります。」

インフラ出身の清澄は、自身満々の森下に少し意地悪をしたくなった。

「さすがだね、森下君。極めてオーソドックスな対応だ。ただ、これまでの消費税が一律であった場合と比べると多少なりとも処理は重たくなると思うが、その点は大丈夫かね。」

ところが清澄の思いに反し、森下は想定通りと言わんばかりに一層得意げな顔つきで答えた。

「チリも積もればということはありませんからね。オンライン業務については1件あたりの処理時間に変動はないか計測した上で、さらに同時入力負荷をかけたテストも実施しました。レスポンスの問題は全くありません。バッチ処理は1.2倍の処理時間になってしまいましたが、翌日のオンライン開始には全く影響ないことを確認しています。個々の明細単位で税率を持つことにより、データベースや帳票のレイアウト変更が必要となりますが、実はこれも8%に上がった際に対応が済んでいます。運用への影響も軽微ですし、運用チームの了解もとっています。可用性、性能・拡張性、運用・保守性、移行、考えられる範囲の確認は実施してきたと思っています。あ、今回はマイナンバーのときは違って、セキュリティ要件の変更はありませんでした。」

まくしたてる森下に対し、清澄は一瞬絶句したのち、遠慮がちに質問を重ねた。

「そ、そうか・・・さすがだな、森下君。チェックリスト代わりに非機能要求グレードでも使ってみたらどうだ、というつもりだったが、余計な心配だったようだ。複数税率対応がそのような状況なら、適格請求書の方も大丈夫そうだね。」

もっと執拗な質問攻めを覚悟していた森下は少し拍子抜けした様子で答えた。

「仕入に関する他社からの請求書や納品書の件ですね。明細ごとに税率が記載されたものが必要となるそうですね。ただ当社の場合は、これまでも仕入伝票と束ねてきちんと保管しています。適格請求書として有効かどうか、経理担当者が点検する作業が加わるものの、大きく業務が変わるわけではありません。ただ、今回を機にこれらの請求書や納品書を電子帳票システムの管理対象とすることになりそうです。」

淀みない森下の説明に清澄は半ば安心してしまったかのように大きく頷きながら聞いていた。

「そうか、なるほどね。ということは、電子帳票システムについても拡張性や運用への影響がないか確認してもらっているということだね？」

森下は安堵した。おそらく清澄の質問はこれで終わりだろう。想定以上に早く終わりそうだ。

「それについては、2年目の白河君に担当してもらっています。白河君はまだ不慣れなところもありますので、今回の軽減税率の対応に関連して影響があると考えられる性能と運用につい

て、非機能要求グレードを利用して重点的に確認してもらっています。」

清澄はすっかり安心した様子で言った。

「そうか。非機能要求グレードを使えば、まだ若い白河君でも漏れなく確認が進められるだろうね。」

解説

新たな法律の施行、政令や条例など、社会の諸制度が変更されると、企業の情報システムにも様々な影響が生じます。監督官庁から出される指針や、業界団体での合意事項なども含めると、諸制度の変更に伴うシステム対応は、相当な割合を占めるかもしれません。

時期も決められており、企業固有の要件や戦略とは全く関係ないにも関わらず、必ず対応しなければならない案件なのですが、人材にしても、予算にしても、投下できる資源には限界がありますから、必須の対応だからこその見極めが重要となります。

「ここを外してしまっただけでは意味がない、きちんと対応しないと法律に違反することになってしまう」といった要件には集中的に資源を投下する必要がありますが、そうでないものには思い切って割り切ることも必要となるでしょう。例えば「オンラインの性能は多少犠牲にしよう」とか「この帳票の発送時期はユーザーに説明して多少後ろ倒しにしよう」とかいったことです。これは制度変更に伴う対応に限ったことではないかもしれませんが、機能要件と同様に、非機能要件についても、「今回は何を重要視しなければならないか」という視点は大切です。

例えば、マイナンバー制度が導入された際には、非機能要件としてのセキュリティ要件が突出して重要視されました。他の要件がどうでもいいという訳ではありませんが、技術的安全管理措置として、分離管理や暗号化など、要件が法律で定められた典型的な制度変更と言えます。

さて、深川商事の消費税対応を今一度検証してみましょう。消費税という税制そのものは完全に定着していますし、税率の変更も今回が初めてではありません。深川商事に限らず、複数税率については既に対応済みという企業は多いことでしょう。未対応であったとしても、森下君が説明しているように、機能要件も非機能要件も何をすべきかを明確化しやすいと言えるのではないのでしょうか？

今回の事例では、消費税対応に伴う拡張性や運用への影響を確認するために、白河君には非機能要求グレードを利用してもらいました。

このような法律や制度の改正に伴い業務の見直しが発生する場合には、非機能要求についても影響がないか確認が必要です。そのために、非機能要求グレードのような鏡となる指標を使って客観的に確認するようにしましょう。

ポイントを整理してみると、以下の2点に集約されます。

- ・ 法律や制度の改正に伴い業務の見直しが発生する場合、非機能要求についても変更がないか確認を行う。
- ・ 非機能要件について変更がある場合、今回は何が重要な非機能要求なのかを見極める。

2.6 事業変革に伴う要求変化に対応する

相羽電子工業は、小型部品や電子パーツのメーカーである。以前は大手の精密機器メーカーとの取引が中心であったが、3年前に立ち上げたホビー市場向けの組み立てロボットキットがヒットし、一般の消費者からもよく知られる存在となった。ロボット事業はいまや売り上げの3割を占める事業に成長している。

情報システム統括部では製造ラインのシステム管理から受発注・在庫管理まで一手に引き受け運用を行っているが、事業拡大への対応に向けてシステム再構築の検討が進んでいた。

「現行踏襲でいいんだ。特に何か目新しいことを頼んでいるわけじゃない。とりあえず刷新したインフラに載せ替えるだけの話だろう？」

奥中統括部長の問いかけに対して、川口課長は厳しい表情で答えた。

「見た目はそうかも知れませんが、我々には刷新した3年・5年先まで、安定した稼働に対する責任があります。要件の見直しから行う必要があると考えます。」

「構築時の要件定義書があるだろう。そんなに大変なのか？理由を説明してくれるかい？」

「まずは実状からお話しします。6年前の稼働開始から3年たったころには、一般消費者向けロボットの事業が急成長し、別に立ち上げた直販サイトからの注文もこちらのシステムに流れてくるようになりました。」

「確かにそうだったな。」

「さらに3年前に、ロボットの保守パーツやオプションパーツも1個から販売する方針となり注文トランザクションは大口の商取引だけだったときに比べると100倍近くなりました。」

「それは本当か？さすがに100倍はないだろう。」

「大口の商取引は繁忙期でも1日せいぜい5とか10件あれば多い方です。それに比べて一般消費者向けは新しいオプションが発売になると、1日300とか500件の突発的なアクセスが発生することがあるのです。」

「ロボット事業といっても本業の2～3割と思っていたが、システムにとってみると大変な事になっていたのか。知らなかったよ。」

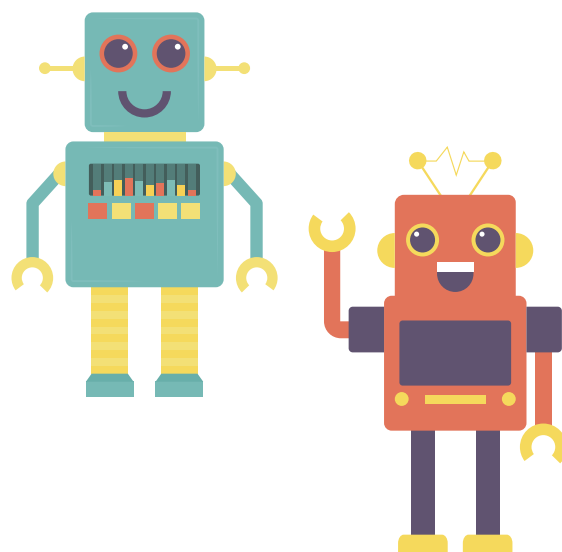
「ですから、3年・5年先を見て要件定義をしっかりとしないと大変な事になります。」

ここで川口課長は部下の太田を呼んだ。

「太田君、この前の非機能要求グレードでアセスメントした結果をもってきてくれるかい？」

太田は資料を奥中統括部長に手渡しして説明を始めた。

「現行システムと、3年前の拡張時のアセスメントの結果です。こうやって比べてみると、3年前はオンラインリクエスト数を3倍と定義していますね。しかし実状はこの3年で10倍



になっています。」

「ロボット事業部にもヒアリングして、次の3年の成長予測もきちんと聞いてこないといけないということだね。」

「はい、ほかの事業でもそういうことが起こらないかどうか、聞いておく必要があります。加えて、一般消費者向けは発売日などにピークがくることがあるのが特徴で、そのあたりも要件として見直す必要があります。」

「なるほどなあ。要件そのものがこの3年で大きく変化していたのか。」

「だからシステム更改は侮れないんです。」

太田は経験を積む良い機会となると考えて積極的に発言した。

「奥中統括部長、川口課長にいつもしわ寄せが行かないように、要件定義からしっかり入らせてください。」

「もちろんだ。頼んだよ。今度は僕も要件定義にしっかり加わることにしよう。」

解説

多くのシステムでは、機器の老朽化・保守期限切れなど、3年～5年で機器の刷新が行われています。こうしたシステムの再構築を、単純に機器入れ替えをするだけと安易に考えていないでしょうか。OSやミドルウェアのバージョンの変更もありますし、それに伴う非互換への対処のためアプリケーションの修正が発生する可能性もあるでしょう。さらには、技術の進歩とともに同じアーキテクチャーの機器が手に入らなくなったり、トレンドの変化により構成の変更を余儀なくされることもあります。

しかしそれ以上に気をつけなければならないのは、システムに対する要求の変化です。事業を取り巻く環境変化のスピードが速まっている昨今では、現行のシステムと再構築するシステムでは時間の経過とともに要求も大きく変化していることがあります。

今回の事例では、3年前のアセスメント時には、3年後にはリクエスト数が3倍になると見込んでいたものに対して、現実には10倍となっていました。さらに今後も伸びが見込まれるとなると、当時の要件定義を踏襲してしまうと、間違いなく大きなトラブルを招くことになるでしょう。

肝心なのは、現行踏襲を前提としているような場合においても、現在の実態から見て、非機能を含めた要求が本当に変化していないかを客観的に把握しておくということです。

上記の例では幸い川口課長が要求の変化に気づいており、奥中統括部長にしっかりと課題とリスクを伝えていきます。また太田さんがアセスメントを繰り返し行っていたことから時系列での要件の差異が明確化されており、次期システムに対する要件を慎重に定義し直すきっかけとなっています。



2.7 売り文句による一面的な判断を防ぐ（クラウド）

「今度のシステムは、クラウドを使うことにする。」

と部長が言っている。システム更改の責任者に新たに着任するなり、幅広く情報収集をしているようだ。

「本当にわかって言っているのかな・・・」

担当になった利根川は大急ぎでクラウドを採用した場合の課題を検討し始めた。性能は？可用性は？セキュリティは？データのバックアップは？など気にしなければいけないことがたくさんある。

ふと利根川は、これらの課題は今まで行ってきたシステム開発とそれほど変わらないということに気付いた。

「非機能要求グレードで確認するとどうなるかやってみよう。」

性能はベストエフォート型が多いみたいで、具体的な数値保証がされているサービスは少ないようだが、少し使ってみたらこれくらいなら許容範囲かもしれない、ユーザーにきちんと説明すればわかってもらえるだろう。可用性は広域災害を考え海外データセンターを是非利用してみたいけれど……。サーバーが故障した時の復旧サービス内容のお値段次第だ。全てのシステムの可用性を高くすると結構な金額になるからメリハリを付けて複数のクラウドを併用しないといけないな。やはり要求の厳しい業務には使えないかも。セキュリティはよく読むと機能はあるけど、結局は自分で管理しなきゃいけないということみたいだ。社内で管理しているIDとの統合はどうしよう……。あと、データはやっぱり自分でバックアップを取得する必要があるそうだな……。

まとめると、クラウドでは機能はもちろん目的に合ったものを選択する必要があるけれど、通常のシステムと同様に非機能についてもきちんと確認しておくことが必要ということがわかった。結構厄介だけどそれぞれきちんと注意して使えばよさそうだな。部長には、非機能要求グレードのそれぞれの項目を説明して、クラウドを適材適所で使ってもらおうようにしよう。

「……。ということで部長、クラウドを業務で使う場合はこういった注意点がありますのでそれをきちんと理解して使うことが必要です。」

「えっ、それじゃ困るな。クラウドを使うのはコストを下げるためなんだから。」

部長はクラウド＝コスト削減ということだけで言い出したようだ。

「ウチでクラウドを使うと何の利点があるんだっけ？」

利根川は用意しておいた非機能要求グレードを取り出して説明を始めた。

「ウチでクラウドを導入するなら段階を踏んで進めた方がよさそうです。今度のシステム更改での採用は時期尚早かもしれません。そもそも、現在のシステムで何が実現できているのか、そして何が足りていないのかもまとめましたので、こちらの説明から始めたいと思います。」

新しい技術やサービスが出てきた時、その技術やサービスの長所がクローズアップされるものです。クラウドの場合、災害対策など可用性がクローズアップされることが多々あります。他方、誰にでも分かりやすい評価軸は「コスト削減効果があるか」でしょう。

クラウドは確かに災害対策として有用ですが、実際にシステム基盤として使うためには、自前でシステム基盤を所有する時と同様、可用性に限らずトータルな確認が必要です。クラウドへ移行する際に、アプリケーションが動作するか、アプリケーションの開発をクラウド基盤で行えるかの確認も欠かせません。

利根川君が気になった地震などの広域災害対策は、BCPの観点です。自前でシステム基盤を構築する場合、どうしてもサーバーの設置は国内になりがちですが、海外のデータセンターを選択できるクラウドは大きな魅力です。広域災害への備えとして有力な候補となるでしょう。

また、柔軟にシステムリソースの増減ができる点もクラウドの魅力です。リソースの増減は、非機能要求グレードでは「性能・拡張性」の項目が該当しますが、リソースの増減はクラウドならお任せなので項目の検討は不要とお考えかもしれません。しかし利用するリソースを増やせば料金の支払いが増えます。将来の業務量やデータ量の増大によりクラウドの料金がどのように増えるかというコストの視点は必須です。業務データの増大に伴う料金の“拡張性”を把握するためにも、非機能要求グレードに掲載された拡張性の項目を明確化しておくことが有益です。

利根川君の上司の部長はコスト削減を目論んでいたものの、期待通りでなかったようです。新技術や新サービスというものは、それが持つ長所に強いニーズがある顧客が導入するものです。自社がクラウドを選択すべきか否かを、非機能要求グレードを用いて分析されることを今一度お勧めします。

このように、クラウドを検討する場合、まず、自社の要求を明確化する必要があります。要求を明確化することで、その要求をどのような技術で実現すべきか検討しやすくなります。この活用シーンでは、コスト面の問題によりクラウドの採用を見送るかもしれません。

しかし、現在は可用性がクローズアップされ必ずしもコスト削減につながるものではないクラウドが、いつかはこなれてサービス価格が低下し、可用性とコスト削減を両立する選択肢になる可能性もあります。また、この活用シーンに出てくる企業において、グループ会社間のシステム連携、スマートデバイスの利用、認証基盤の構築、などからクラウドを採用する強い動機が生まれるかもしれません。

これまで使ったことのない新しいサービスを自社に合ったタイミングで導入するためには、自社が重視する評価軸を決めることが必要です。

2.8 手段にとらわれずニーズを明確化する（クラウド）

あるネット企業ではシステム基盤の抜本的な見直しに着手しようとしている。

同社が顧客に提供するネットコンテンツは、その人気の度合いにより、必要なシステムリソースは常に、かつ急激に変動するという特徴がある。また、サービス提供期間もまちまちである。さらに厄介なことに、そのネットコンテンツの人気度合いをコンテンツのリリース前に予測することが非常に難しい。例えば、同社が人と時間をかけて開発し、多くのシステムリソースを費して準備したコンテンツがすぐに廃止になるケースもあれば、スモールスタートしたコンテンツが大化けして大人気となる場合もある。多種多様なコンテンツを持つ同社では、リソースのやり繰りが多大な負担となっている。

ある日、新しく企画したネットコンテンツのチームリーダーの白山が、システム部長の千石に話を切り出した。

「来月サービスを開始するコンテンツの件で報告があります。サーバーは先週廃止したコンテンツがあるため処理能力に不足はありませんが、ディスクは毎月リリースしたコンテンツのデータ量が想定外に増加したため空スペースが足りません。現在手配中ですが、準備が整うのは来週中になる見込みです。」

社内のいくつものコンテンツを統括する千石はため息交じりに言った。

「調達リードタイムは2週間程度だったなあ。他のコンテンツの廃止が急遽決まればこの調達は無駄になるかもしれない。これまで何度もそんな失敗を繰り返してきたからな。やはり、クラウド化の検討を早急に進めなければな。」

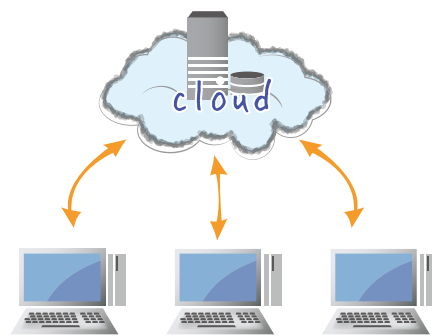
同社では、ネットコンテンツ毎のシステムリソースの増強・縮小を機動的に行うことを主たる目的として、クラウドの導入を検討してきた。今回の震災で、事業継続という新たな観点が増え、さらに導入を急いでいる。千石は気をもみながら言った。

「クラウド事業者へ提示する資料の準備はスケジュール通り進んでいるかね。クラウドを利用するからといって、我が社の現在のシステムレベルを落とすことはできない。非機能要求グレードをベースに取りまとめた要求レベルは社内でもオーソライズ済みなのだから早く事業者を検討を始めさせてくれ。」

チームを統括する白山は言った。

「はい、来週には第1版が上がる予定です。情報提供依頼（RFI）に応じた5社のIaaS型クラウド事業者と、PaaS型クラウド事業者に渡したいと考えています。それぞれの事業者は技術の持ち味が異なりますので、提案も異なるタイプの内容になると思います。」

「わかった。いずれにせよこちらの要求がぶれないよう注意してくれ。あと、紙だけで判断せず、ウチの者に評価版のサービスを実際に使わせて動作確認もしておくように。」



解説

クラウドの利用は、ここ数年徐々に進んできましたが、企業のシステム投資額の削減要求や大規模災害時のBCP対策の見直しなどを契機として、昨今、急速に広がりを見せています。その流れは今後も継続するものと予想されており、自社のシステム基盤をオンプレミス（自社で用意した設備を自社運用すること）からIaaS、PaaS、さらにSaaSといった各種クラウド事業者が提供するサービスへの変更もますます多くなりそうです。

パブリッククラウドの場合、事業者が保証するサービスレベルが所与のものとなることから、自社の要求に適合した事業者をいかにうまく選択するかがクラウド環境を使いこなす鍵となります。このケースで紹介しているネット企業は、コンテンツのアクセス状況や話題性を踏まえてサーバー台数を減らしたり増やしたりすることを重視しています。

非機能要求グレードは、クラウドのサービスレベルと自社システムの非機能要求の適合性を比較検討するツールとして利用することが可能です。利用する手順はオンプレミスのシステム更改の場合とさして違いはありません。まず、既存システムの非機能要求について項目一覧にある項目を一通り確認します。オンプレミスのシステム更改の場合、こうして確認した非機能要求を開発ベンダーと共有する流れとなりますが、クラウドへの置き換えの場合には、クラウド事業者のサービス仕様書等を入手し検討する流れとなります。

しかしここでご留意いただきたいのは、クラウドについては、それを提供する事業者のサービスレベルが所与のものとなるため、オンプレミスでシステム基盤を更改する場合に比べ自由度が低くなることです。また、詳細なシステム構成が公開されているとは限りません。セキュリティが担保されているか、稼働率はどうか、レスポンスなどのシステム特性は自社の非機能要求に合致しているか、といったことが明確でない場合もあります。

そのような場合、自社システムの要求として譲れない部分と譲れる部分を整理し直したり、これぞと思うクラウドを試験的に利用してみるなど、受け身ではない視点と行動が必要です。

このケースで紹介しているネット企業は、5社のクラウド事業者の提案をこれから受けるようです。提案内容の比較検討で何よりも大切なのは、システム部長が発言したとおり、自社の要求がぶれないようにすることです。このネット企業の場合、事業継続性とシステムリソースを柔軟に増減できることにポイントがあるといえるでしょう。事業継続性の観点から、複数のクラウドを組み合わせる使うのが適切かもしれません。

なお、忘れてはいけないのはセキュリティです。この企業の場合、現時点では大きなセキュリティ問題が起きていないかもしれませんが、しかし、顧客の個人情報の流出やネット攻撃によるサービス停止のリスクも考慮しておく必要があります。現在直面している大きな課題に目がいきがちですが、非機能要求グレードにある項目を一通り確認することで、セキュリティを含め検討が必要な範囲を網羅することができます。

現時点で気がかりな事項に偏らず幅広い視点で検討し、自社のニーズの全体像を明確化しましょう。

TIPS システム拡張開発における非機能要求の考え方

システム拡張開発（保守開発）における非機能要求の検討は、最初に既存システムの非機能要求を確認し、次に開発部分の非機能要求を検討し、最後に両方を総合的に勘案して開発部分の非機能要求を決定するという流れになります。

1) 既存システムの非機能要求を確認する

非機能要求が明確に規定されている場合

既存システムの規定された非機能要求を確認する。

非機能要求が暗黙知の状態の実現されている場合

項目一覧を使用して、既存システムの非機能要求を確認する必要がある。

2) 開発部分の非機能要求を検討する

項目一覧を使用して、既存の非機能要求を認識しながら、拡張開発部分の非機能要求を明確にする。既存システムの非機能要求について既に要求レベルが存在しているので、拡張開発部分の非機能要求が既存システムの要求レベルに矛盾しないように検討していく。

3) 開発部分の非機能要求を決定する

既存の非機能要求と開発する非機能要求の両方を総合的に勘案し、既存システムの共用できる部分を利用して、開発部分に必要な非機能要求を決定する。

図 システム拡張開発時における非機能要求の検討の流れ

例えば、ネットワークやネットワーク機器を共用する場合には、セキュリティや可用性の一部は、既に実現されているレベルを利用できることがあります。

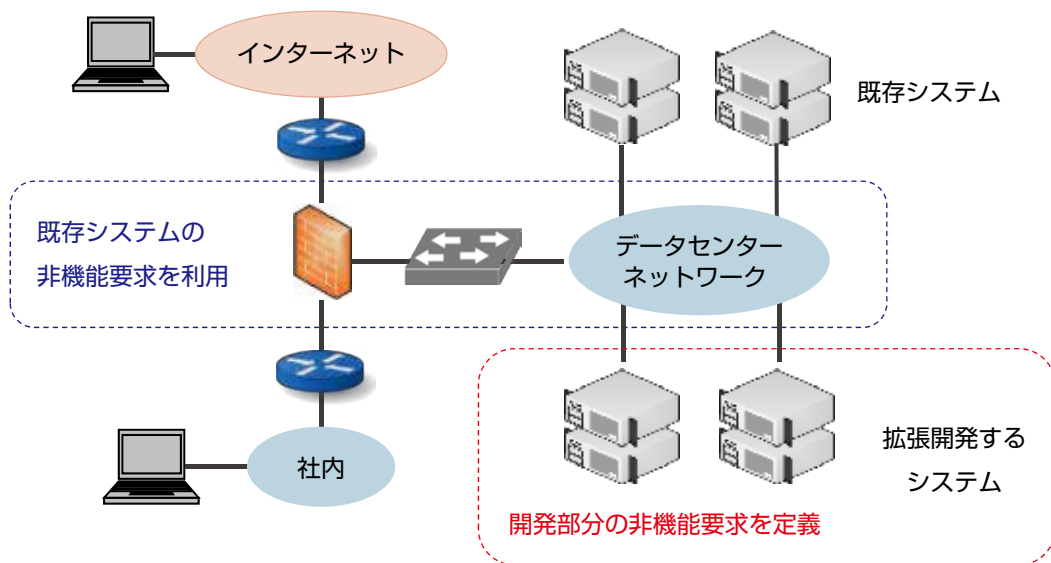


図 既存システムの非機能要求を利用した例

3章

運用

「非機能要求グレードは非機能要求を決めるためのツールだから、要求定義のような上流工程が対象であり、運用の局面では使えないのでは？」と思われる方も多いと思います。また、「非機能要求グレードの良さは理解できるのだが、いかにせん数が多い。活用シートの238項目と解説を見ただけでやる気が失せてしまう。」と判断されている方も少なくないと思います。

業務システムは稼働を開始してからは、ハードウェアの障害、不正なデータの投入といった異常事態にも対応し、安定して稼働し続けることが本来あるべき姿です。しかし、システム企画、システム要求定義の段階でシステムの安定稼働に対する要求（合意すべき項目、実現レベル）を明らかにして実現すべき範囲を合意し、設計の段階では合意された要求を実現する仕組みを十分に検討しておかなければなりません。後になってからシステムへの組み込みを検討したのでは、技術的にも時間的にも難しくなります。結果としてシステムに障害が発生した場合業務が止まってしまう、あるいは遅延してしまう、ハードウェアの部品交換やプログラムの更新が頻発する、更新に伴う再稼働までに時間を必要とするといったことになりかねません。

また、システム開発費用を削減するつもりで運用に関わる設備や機能を必要以上に削ったために、逆にシステム稼働後の維持費用がかかるようになったのでは、システムのライフサイクル全体を通したシステムの投資対効果は上がりません。

運用に関する要求は、システム構築を担当するSIベンダーやシステムのオーナーである利用部門だけではなく社内のシステム運用部門、さらにはハードウェアやミドルソフトウェアを提供するベンダーにも参加してもらい、合意すべき項目とその実現レベルを明らかにしてシステム要求、ソフトウェア要求に反映させていかなければなりません。

また、システムの構築や更改を直近に予定していないような場合であっても、障害発生時における対策の評価、災害発生時における事業継続計画（BCP）の評価といった応用もできるようになります。あるいは年度予算配分の優先付け検討の場面で、複数システムの現状レベルを非機能要求グレードを使って横並びに評価するといったことにも利用できます。

非機能要求グレードを使ってシステムを運用の観点から見てみると、システム全体の運用の平均水準、あるいは業務の違いによるシステム運用に対する重要項目の捉え方をはっきりさせることができます。そうすることで、品質をどこまで担保すべきか、コストをどのようにかけるべきかを客観的に評価できるようになります。

本章では、システム運用時間の調整、障害時の運用に関する要求の捉え方、システム導入コストを重視しすぎたためにシステム稼働後に発生した運用コストの増大問題、稼働中システムの事業継続計画の実現度の点検、年度予算の配分などの利用シーンについて、非機能要求グレードを使ってどのように行えばよいかを解説します。

3.1 システム運用時間とバックアップのトレードオフを考える

情報システム部でシステム運用チームに所属している大塚君。今回、新しい伝票入力システムの運用について要求をまとめる担当となった。これまで先輩の指示にしたがって作業を担当していた大塚君には初めての経験である。先輩に相談し、まずは利用部門である経理部門、営業部門の各窓口ヒアリングを行った。

《経理部門にて》

「新しく開発する伝票入力システムの運用についてお話を伺いに来ました。」

「勤務時間内（午前9時～午後5時）に当日の伝票が入力できればいいですよ。逆の言い方をすれば、毎日午前9時には必ず日中オンラインで伝票が入力できることを保証してください。今までは最悪2時間くらい開始が遅れることがありましたから。システムの人が頑張っているのはわかるけど、その分僕たちの作業が止まって全社に迷惑がかかることは止めて欲しいな。今後は絶対遅れないようにして欲しいです。」

また、入力した会計、入金・支払伝票データは会計監査のために毎日、日別に保存が必要です。業務量的には、当部では月初めにある前月締め処理前日に伝票数が倍くらいになります。その時でも日中オンラインの開始時間は変更しないでください。僕たちとしては逆に開始時間を早めて欲しいと思っているくらいですから。」

「わかりました。ご希望は善処するように努めます。」

《営業部門にて》

「新しく開発する伝票入力システムの運用についてお話を伺いに来ました。」

「営業は業務の都合上、帰社が遅くなる傾向があるので、夜の10時までは当日分の受注伝票の入力を可能にして欲しい。できれば土日也使えると嬉しいです。営業部門は営業ノルマの関係上、月末が近づくほど伝票数が増えていくことをご存知ですよね。」

各部門から要望をヒアリングしてきた大塚君。各部門からの要望を反映したオンラインの利用時間帯を確保した上で、現在の夜間日次処理の処理時間、データバックアップ時間を考慮してシステムの稼働時間を計算してみた。

「日中オンライン処理は経理部門の要求に合わせて午前9時から開始、終了は営業部門から要望のあった午後10時までとしよう。そこから日次の夜間バッチ処理を開始。これまでの記録では日次の夜間バッチ処理時間は月初めの締日に最大で6時間かかるのか。するとバックアップ開始は遅くとも翌朝4時から開始か。データベースのバックアップ処理は平均3時間くらいかかるのか。すると、遅くとも午前7時には終了できるな。あれ？月初のバックアップは最大5時間もかかるの？これじゃ月初めは日中オンラインの開始が午前9時ぎりぎりになってしまう。大丈夫かな？」

頭を抱えている大塚君の頭上に先輩からきつい一言。

「大塚君、システムはバックアップの準備と日次の切替えや予備時間に合計1時間の余裕を当然考慮しているだろうね。」



解説

システムの運用時間は可用性の要求としても、また運用・保守性の要求としても挙げられる要求項目です。

可用性の要求は、「システム全体が止まらないで稼働し続けなければならない要求」を実現する程度を明らかにし、その要求に応じたシステムアーキテクチャーを検討し実現することを目的としています。

これに対して運用・保守性の要求は「システム化対象となる業務が必要とする運用時間に関する要求」、すなわち、1日におけるオンライン処理の利用時間帯、その後のバッチ処理、データバックアップ処理などの処理時間がどのくらい必要かを明らかにし、運用時間の実現レベルを検討・調整することを目的としています。

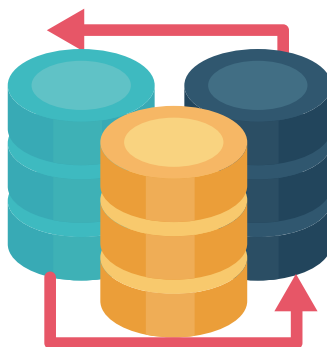
業務上日中のオンライン処理が終了してからでないと日次のバッチ処理は処理できない、または、オンライン処理、バッチ処理の終了後でないとデータのバックアップ処理は実行できないといった制約がある場合は、その制約を考慮して処理時間を検討する必要があります。

可用性で定義した運用時間の実現の程度と運用・保守性で検討した実現の程度に差異が出た場合には、双方の実現レベルが同じになるように調整が必要となります。この場合には運用時間の実現の程度により、システム化予算の見直しや利用部門へのサービスレベルを含めて検討することになります。

もし、全ての処理を合計した時間が24時間（1日）を超えてしまったというような場合には、どこかの処理時間を短縮する必要があるが生じます。実際の設計に入らないと具体的な処理時間の削減方法あるいは回避策、より詳細な処理時間見積もりの検討ができないことも多くあります。この場合には、上流の工程ではとりあえずの仮決めで実現の程度を決め、後続の工程でその実現の程度を見直すというやり方を取るのが賢明です。

また、このケースにも出てきたように、業務の特性により日によって処理すべきデータ量が大幅に変化します。平均の運用時間だけではなく、処理すべきデータ量の多い日にはどのくらいの時間が必要となるのか、また月間のデータ変動により、期間としてはどのくらいの期間を想定し、その期間中のデータ量の増大度はどの程度かを明らかにする必要があります。業務が異なるとデータ量の多い日も変わってきますので、システム全体でデータ量の多い日を明らかにする必要があります。このように整理したデータ量の多い日を「特定日」として、同様なやり方で特定日の運用時間を検討します。

非機能要求グレードでは、運用時間に関して通常時の運用時間と特定日の運用時間に分けて非機能要求を整理するよう考慮されています。



3.2 障害時の運用について業務毎の優先度を考える

情報システム部でシステム運用チームに所属している大塚君。今回、新しい伝票入力システムのシステム保守について検討しているうちに、システムに障害が発生した場合のシステム運用についても要求をまとめることが必要になってきた。上司からは障害が発生した場合に、社内における業務の運用をどうするかというインシデント管理の観点から要求を整理して利用部門に示し了解を取ることが求められている。障害時の運用方法について先輩の音羽さんの知恵を借りることにした。

「音羽さん、障害時の運用ってどう考えればいいんですか？ハードウェアだったら予備機に切り替えて運用する手順を整備するだけでいいと思うんですが。」

「予備機に切り替えるっていつでも予備機がどのような状態になっているかによってシステムが利用可能となるまでの時間は変わってくるよ。また障害発生後に利用部門に改めて使ってもらうなら、その障害発生直前までのデータの保存状態も重要なポイントだね。今のシステムは業務データを障害発生直前の状態に戻してすぐに使えるようになっているかい？そもそもシステムに求められる稼働率の高さ、切替えの速さは可用性に関する要求としてまとめられる。システム運用ではこれを前提としてシステム障害時の運用を考えることが必要だ。必要なら可用性の要求と整合を取る必要もあるよ。システム利用者からは、システムが止まっても業務が継続できる何らかの手段を提供して欲しいとよく言われているからね。」

「音羽さん、どうすればいいんですか？」

「障害が発生した場合、どれだけ急いで復旧させないといけない業務なのか、データの復旧は自動的に行わなければならない業務か、人手をかけて復旧してもよいのか、などを業務毎に要求を整理してはっきりさせることが重要だ。また利用者に、障害がすぐに解決できそうな問題か否かなどを伝えることも重要だね。」

「業務の種類によって、システム障害時の対応の仕方は変わるってことですか。場合によっては、利用部門の方に仕事の手順を変更してもらうなんていうのもありますか？」

「そうだね。業務の代替のやり方を決めておくのは重要だよ。当然、そのやり方もマニュアルとして整備し教育しておかないと実際には使えないよ。」

「日中に発生しうる障害の検知を誰が行うか、障害を検知した場合、誰が、いつから原因究明を行うのか、もしベンダーに対応をお願いするとしたら、どのくらいの時間で来てもらえるかなんて約束事を決めておくことが重要なんですね？」

「そうそう、障害の検知や関係部門への連絡は人手で行うのか、システムで自動的に行うようにするのかといった決定も重要だよ。自動化する場合にはシステムへの作り込みにも影響するからね。」

「そうか。それ以外にも気がつかない点がありそうですね。良い参考書ないかなあ。」



解説

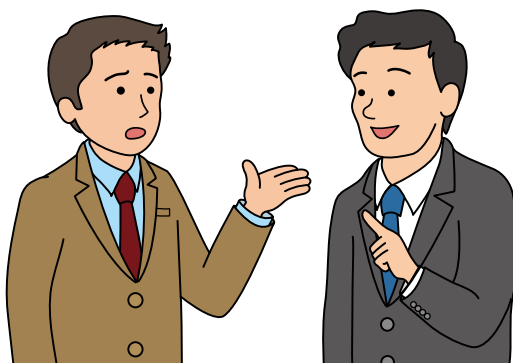
ハードウェアの故障、不正なデータの投入、ソフトウェアの不具合によりシステムが正常に稼働できない状態が生じます。こういった障害が発生した場合でも利用部門にできるだけ影響を与えないように対策をあらかじめ検討しておくことがシステム運用では求められます。システムを止めない、あるいは障害発生しても迅速に復旧させるという要求は、可用性の目標復旧水準(業務停止時)の目標復旧地点(RPO)、目標復旧時間(RTO)として目標水準が検討されますが、システムの利用者の立場から考えると、システムが障害になったからといって自分たちの業務がいつまでも再開できないのでは「使えないシステム」「動かないシステム」との評価につながります。

障害が発生したら大変だといっても、どのくらいの頻度でいつ発生するかわからない障害のために必要以上のコストはかけられません。また利用者の立場から言えば、障害が起こった場合でも、業務の代替の手順が用意されていれば大きな影響を受けずに業務遂行ができます。障害が発生した後に、システム運用部門から単に代替策を検討してくださいと要請されてもどうしてよいかわからずに拒絶されるだけでしょう。業務の重要度によってデータを障害発生前に戻すやり方も異なります。自動化するのか、人手による作業で復旧するのか、はたまた別の業務で出力されたデータを再利用するのかということです。自動化するならばシステム構築時に実装する必要がある、システム開発コストに影響します。人手作業で行うならば、システム構築には影響しませんが、システム運用コストに影響します。発生頻度と発生した場合の影響度/損害の大きさを考慮して、システム構築時にコストをかけるのか、それともシステム構築時点では費用はかけずにシステム運用段階で費用をかけるのか、目標水準を明らかにする必要があります。どちらのコストを優先に考えるかはシステム開発の方針次第です。

同様にいち早く障害を検知するためには、通常からどの程度の頻度でシステムのどの範囲をどのようなやり方で監視するのかということも目標水準として検討すべき事項です。

障害検知した時に、原因の究明を誰の責任で、どの程度の緊急度で行うのかといった点の明確化も重要です。システム運用担当者が責任を持つのか、それともハードウェア保守やソフトウェア保守を担当している会社の担当者が責任を持つのかということです。もしハードウェアやソフトウェアの保守の担当者が究明するとした場合、障害の発生現場にどのくらいで到着できるようにしなければならないかも、可用性(目標復旧時間)を参考に具体化しなければなりません。

システムをかなりの長期間使用する場合、ハードウェアやソフトウェアが使用期間中に正規のサポートの対象でなくなる可能性があります。正規のサポート期間を過ぎてもハードウェアの交換部品や更新プログラムが必ず入手できることの確証を取っておくことも重要です。



3.3 システム開発時に運用コストを検討する

不忍コーポレーションは、長引く景気後退から3期連続の減収減益という決算状況であった。このため経営層からは、収益改善策としてシステム部門に対し、(1) 新規開発案件の絞り込み、(2) システム運用コストの削減が喫緊の取り組むべき課題として指示された。特にシステム再構築を行ってコスト削減したはずのオンライントレードシステムでシステム維持運用コストが高止まりしていることが問題視されていた。

実は、新システム開発にあたり、導入コスト削減を狙って単体あたりの信頼性は低いけれども安価な普及タイプの製品を採用し、システム全体としては故障に強い冗長構成を採用して要求される信頼性を達成したつもりだった。しかし、財務担当の豊島執行役員からシステム部の向丘部長に対して次の検討指示が出された。

「オンライントレードシステムのシステム維持運用コストを何とか下げて欲しい。ハードウェア導入費、保守費、ソフトウェアライセンス費など導入コストは君たちの努力で大幅に下げられた。しかし、システム維持運用コストが逆に増加しており、導入時のコスト削減がトータルを見ると打ち消されてしまっている。何らかの改善を図らないと当初投資回収計画の達成は無理だ。君の手腕に期待しているぞ。」

向丘部長も思い当たる点があった。新システムは開発スケジュールの逼迫から運用方法の検討・設計・構築に十分な時間をとれず、システム構成の変更やデータバックアップなどの自動化の仕組みを構築できなかったのである。当初は人手中心の運用管理費用、ベンダーへの体制維持コストは従来と同等と考えていたのだが、実際には人手に頼った運用のために想定以上の維持コストで高止まりしたままであった。さらに悪いことに冗長構成による接続点数の多さから機器障害の発生頻度がかなり上がり、ソフトウェアのバージョンアップやパッチの適用頻度も大幅に増え、対応工数も大幅に増加していたのである。

早速、部下の根津マネージャーに改善の実現可能性を尋ねてみた。

「運用コストの劇的な削減のためには、バックアップ機器の台数増強、構成管理ツールの導入などの自動化処理の作り込みが必要です。しかし正直に申し上げますと、バックアップ機器の追加導入は可能ですが、新たに管理ツールの開発が必要で相当の期間とコストがかかります。残念ながら、当初計画のオンライントレードシステムの残存期間中には追加投資分を回収するのは不可能と試算しています。」

向丘部長は諦めずに根津マネージャーに期間内で対応可能な解決策がないかを求めた。すると根津マネージャーは、

「劇的な改善を図ることは難しいかもしれませんが、とりあえず、オペレーターのシフトの見直しや人員構成の見直しで可能な限りコスト削減に努めます。そう言えば、『非機能要求グレード』という資料には『運用コストへの影響』という欄がありました。その欄に○がついているメトリクスは、運用コストまで配慮して考えておくべきだと聞いたことがあります。早速チェックして改善点を確認します。」

結局、現行システムの稼働中にシステム維持運用コストを多少改善することはできたが、投資回収計画どおりのコスト削減までは至らなかった。このようなことにならないために、「非機能要求グレード」を使ってシステム開発時点でどのような点に注意しておけばよかったのであろうか。

多くのシステム開発の現場では、システムの運用・保守性という分野が、その他の分野と比較して劣後したものとして扱われてはいないでしょうか？

システム開発の完了期限が迫ってくると、システム開発の主たる目的とも言える機能要求、それに密接に関連する性能・拡張性、およびシステムのオーナーである業務部門（ユーザー部門）が高い関心を持つ可用性、また今日では事業会社には強い社会的要請のあるセキュリティなどが優先して検討が進められ、要求事項も早期に固まる傾向がないでしょうか。

さらに要件定義段階で投資コストを考えると、様々な制約の中ではついつい目先の初期投資（導入コスト）にばかり気をとられ、「とにかく安く」という要求を「導入コスト」の面だけで考え、「早く」「安く」「うまく」作ることに意識がいきがちです。

しかし、システムのライフサイクルを開発期間と運用期間に分けて考えると、圧倒的に後者の方が長い期間です。「早く」「安く」作ることに頭が行き、完成後の運用のしやすさや運用コストを考慮せずにいたのでは、実際に運用してみたら前述のシーンのように「運用コスト」がかさみ、こんなはずではなかったという結果にもなりかねません。

一方で、運用・保守性は最低限の管理機能が実装されていれば、大体のことは人手によりカバーできると思われがちです。このため検討の開始時期も遅く、検討のためのリソースも潤沢とは言えない状況です。検討された運用要求の内容、レベルも業務要求等と比較すると往々にして不十分です。多少開発費用や期間がかかっても、システム開発期間中にシステム稼働後の運用のしやすさを考慮して自動運用の仕組みを入れる、障害発生時の具体的な対応策、代替策を検討し必要な仕組みを組み込んでおくことさえできれば、運用に関わる工数や費用も格段に下げることができるのです。

運用・保守性は一つのシステムのライフサイクルの大部分を占める運用フェーズに常に関わりを持つ非機能要求です。運用要求の検討も前述した他の項目と同様の高いレベルで実施していなければ、前述のシーンでご覧いただいたような想定外の運用コストの高騰といった問題が生じる可能性は低くありません。さらにその問題が顕在化した時点では有効な解決策を実施することが困難な場合もあるでしょう。

「非機能要求グレード」では項目一覧表の中に、「運用コストへの影響」という欄を用意しています。この欄に○がついているメトリクスは、「導入コスト」と「運用コスト」に「トレードオフ」が発生しやすいことを示しており、検討時には十分な留意が必要であることを示しています。この欄を活用して先送りになりがちな運用・保守性に関する要件定義を、他の機能・非機能要求と同様に、早期に社内のステークホルダーやベンダーと合意形成を図ることが、このケースの最後に示した問題提起の回答になるのではないのでしょうか。

また、経験上、自ずと検討が進めやすい可用性、性能・拡張性といった非機能要求ではなく、運用・保守性など検討が滞りがちな非機能要求で非機能要求グレードを気軽に利用するという使い方も一つのアイデアです。

システムの構築にはコストがかかります。より高いレベルの要求をすれば、一般に実現が難しくなり、よりコストもかかります。要求の優先順位や必要性の確認をし、導入コストだけではなく長期にわたって発生する運用コストも考慮しながら最適解を模索していくことを忘れないでください。

3.4 システム毎に BCP を評価する

震災以降、日本中の会社が事業継続計画（BCP）の見直しを行っているが、中堅商社の竹早物産も例外ではない。竹早物産は震災直後、通信インフラが復旧するまでの間に自社のシステムを復旧できず、商社として重要な取引先とのデータ交換システムを稼働させることができずに他社に遅れを取ってしまった。ただ、竹早物産は元々真面目な社風であり、その後システム部門の担当者たちが担当システム毎に自発的に計画の見直しを行っていた。このことが逆に情報システム部門の長である本郷部長の悩みになっているようだ。悩んだ本郷部長は、品質管理の担当責任者である駒場課長に相談することにした。

「重要なシステムである部材調達システムの事業継続計画や災害復旧（DR）手順はさすがに良くなったな。他のシステムも頑張っているが、全体として果たして充分なのだろうか？ システム毎に過不足がないか点検したいし、システム間で不整合がないかも確認したい。客観的に評価できないものだろうか？ 駒場君、何か良い知恵はないかい？」

相談された駒場課長は自信を持って答えた。

「それなら、非機能要求グレードなんてどうでしょう？」

「うーん。前に君から一覧表を見せてもらったが、ちょっと細かすぎるし、多すぎるよ。それに非機能要求グレードは開発の上流工程で使うものではないのかい？」

「確かに、上流工程を想定して作られたものですが、私は運用局面でもどこでも、システムライフサイクル全般に適用できると思っています。確かに全てを一気にやるのは無茶な話かもしれませんが、今回は目的がはっきりしていますよね。主要なシステムをピックアップして、しかも可用性に関する項目だけ点検してみるというのはいかがでしょうか？ 目標復旧時間を横並びで見てみるだけでわかることもきっとあると思います。」

そう言われると本郷部長も実際に使って評価してみたくなった。

「そうか・・・ ちょっと可用性について評価してみてくれるかな？」

駒場課長は早速部下に協力してもらい、非機能要求グレードのうち、可用性の項目だけに絞った評価表を作成することにした。そして情報システム部門の開発課と運用課の担当者向けに説明会を開き、品質管理課まで評価報告を出すよう指示した。現場の抵抗はあったものの項目が限られ、かつほとんどの項目が最近見直したばかりだったので、ほぼ締切通りに各チームから報告された。それを品質管理課で一覧表にまとめ、本郷部長に報告した。

一覧表に目を通した本郷部長は思いのほか満足した顔を見せた。

「非機能要求グレードも意外に使えるなあ。部材調達システムは最も重要なシステムだし、災害対策は進んでいると思っていたが、似たような企業間取引のシステムである、配送監視システムの方が進んでいる項目もいくつかあるようだな。部材調達システムのレベルでも充分なのではないかと思うが、一度関係者で議論してみる必要があるな。」

「そうですね。それは是非実施された方がよろしいかと思います。後は、これが確かに実装や体制で担保されているのかという点も気になりますね。また、今すぐは無理ですが、運用・保守性も点検してみる必要があるかもしれません。」

解説

事業継続計画（BCP）は、ご存知のように自然災害や事故などに対して事業活動が続けるため、または、停止した事業を再開できるようにするために策定する計画のことです。

BCP は教科書的には、まず事業に対する影響分析を行って自社の業務プロセスが抱えるリスクと影響（損害）を洗い出し、その上で優先的に復旧すべき業務とそれに必要な設備やシステムを明らかにして目標復旧時間の設定や復旧手順を計画していくこと、定期的に見直されることが推奨されています。

これを非機能要求の視点から見たらどうでしょうか？

BCP で求められている要素は、そのほとんどが非機能要求グレードでいう可用性の項目に含まれる要素となっています。このことを踏まえて、ケースでは駒場課長が「BCP の点検のために、可用性だけ」という絞込みを行っています。もし、災害対策だけなら、可用性全てではなく、災害対策だけに評価を絞っても良いかもしれません。

このようにテーマを絞り込んで評価を行う場合には、全項目を一気に全て実施するのではなく、重要と思われる評価項目群の単位で実施します。そしてその効果を確認してから段階的に他の項目にも手を広げてみるというアプローチはいかがでしょうか。

絞込みの観点では、他に、

- (1) 重要項目の 92 項目だけでやってみる。
- (2) 全てのシステムではなく、いくつかのシステムでパイロットとして実施する。

といったことも考えられます。



3.5 機器増設要求の妥当性を評価する

製造業の弥生製造は3月末が決算だが、毎年秋口から翌年度予算の編成を始め、12月頃までには予算総枠を固めるのが年間を通しての業務運営になっている。IT部門の予算編成担当である西課長もこの季節になると毎年忙しさが増してくる。以前は開発案件の答申の予算査定に算定根拠がわからず苦勞したが、今は利用部門が投資計画書を作成し要求定義を明確にするルールに変わり、予算査定の苦勞からはかなり解放された。

最近、西課長を悩ませているのがシステム基盤の維持強化のための予算化である。

「今年もあちこちから山のように予算申請が来ているなあ。ストレージの追加とか、CPUの増設とか、ソフトウェアの更新とか……。本当に必要なものなのだろうか？ 現場の要求には応えたいのはやまやまだが、単純合計でも今年の倍の予算になっている。このご時勢、昨年度並みのシステム予算枠の確保だけでも厳しいのに、事業の維持継続のためだけの予算の増額なんてあり得ないよ。かと言って、本当に必要なものまで削るわけにはいかないしな……。増強しなかったから障害が起きたとか、導入しなかったために情報漏洩したとか、なんてことになったら本末転倒だしな。何とか妥当性を判断できるような手立てはないものだろうか？ 同じようなシステムで比較することも必要だな。単純にこっちのシステムで予算化したら、あっちのシステムもということになるに決まっているし。」

困っている西課長を覗き込むように部下の北係長が助け舟を出した。

「西課長、お困りのようですね。では非機能要求グレードを申請資料の一部にしてみたらどうでしょう？」

「北君、それは僕も去年一度検討したよ。非機能要求グレードは確かに精緻によくできているし、大規模開発の予算化には必要だと思う。しかし、我々の仕事は保守予算、運用予算だからねえ。申請資料に、非機能要求グレードを添付してくれなどと言った日には、『予算申請抑制のために、手続を面倒にしているのか』なんて、言われかねないよ。」

「確かに、238項目全ては必要ないし、大変な手間になると思います。でも、重要項目92項目だけならやってもらえるかもしれません。」

「なんだ？ 重要項目って？」

「品質やコストに直結する、特に重要な評価項目です。開発のときは、まずこの重要項目だけでも早い段階で明らかにしておき、徐々に他の評価項目も定義しておくんですよ。」

「なるほど……。それなら、現場もある程度協力してくれるかもしれないな。それに、ストレージとかメモリの増強であれば、性能・拡張性の項目だけでもとりあえずはいいはずだしな。では早速申請方法の原案を作ってみてくれないか。」

「わかりました。早急に用意します。」



解説

西課長のように予算策定を担当されている立場の方は、「公平であること」「必要であること」を、「客観的」かつ「合理的」に判断しなければなりません。

「これこれ、こういう要求なのだから、これは必要なのだ。」

「A システムと B システムは同じレベルの可用性を求められている。現状では B システムの方が劣っており、A システム並みにするために、この予算は必要なのだ。」

と要求部門から言われても、経営層から見ると、劣っているはずの B システムのレベルが期待レベルで、A システムは過剰なレベルになっているかもしれません。このように要求を客観的、合理的に判断し、ステークホルダーに説明し合意を得なければなりません。

非機能要求グレードには、メトリクス毎に 2～6 段階のレベルが設定されています。一つのシステムを定義するだけでなく、複数のシステムを横並びで評価する際にも応用できます。ただし、このケースのように運用コストを査定するような場合に、238 項目全てのメトリクス評価は必ずしも必要ないでしょう。そこで北係長は、品質やコストへの影響が多い重要項目に絞り込むことを提案しています。重要項目の詳しい説明は利用ガイド（利用編）を参照してください。

実際に適用するにあたっては、重要項目だけでは不足する場合もあるかもしれません。例えば、ストレージやメモリの増設を予算化する場合には、現在の不足分だけではなく、将来の増分や余裕率も見る必要があります。さらに、増設によってラックの空きユニットやメモリスロットが不足するかもしれません。

これらのメトリクスは、もちろん非機能要求グレードでもカバーしていますが、重要項目にはなっていません。

必要に応じ、重要項目に追加して補強することが必要になります。



3.6 データセンター移設の影響を洗い出す

上に一枚羽織らないと長くはいられないマシンルームから、幾分蒸し暑くも感じるオフィスに戻ってきた藤原君を目ざとく見つけて声をかけたのは柏木室長である。

「藤原君、マシンルームも手狭になってきただろう？」

「全くですよ、柏木室長。ラックの集約も進めていますが、耐荷重も配慮しなければなりません、排熱の問題もあります。200V 給電も電源がそろそろ枯渇しそうです。」

「まさにその話なんだが・・・」

柏木室長がなにか考えながら切り出すと、藤原君が期待を込めた表情でこう答えた。

「ひょっとして何度も先延ばしにされてきたセンター拡充の件、いよいよですか？」

「いや、そのことなんだが、社内システムはそのまま残置することになった。だが、さすがに顧客向けサービスのシステムは他のデータセンターに預けることになったんだ。しかも来年度中だそうだ。」

「えっ、そんな急には無理ですよ。そもそも預けるってどこにですか？」

「半年ぐらい前、藤原君に非機能要求グレードを使っていくつかシステムのアセスメントをしてもらっただろう？」

「はい、よく覚えています。柏木室長は最初は2システムが対象とおっしゃっていましたが、結果報告に伺ったときに、『そんなに素早くできるなら・・・』と対象システムが追加になって、全部で5システムにアセスメントを実施することになりましたから。」

「いや、あのときはシステム改修の作業も重なってたのに悪かったなあ。」

「まあ、タイミングが悪かっただけです。普段からシステムの状態は把握していますし、他の作業が重ならなければそんなに負担に感じるほどの作業ではありませんでしたよ。」

「それなら良かった。それで、そのときの非機能要求グレードのアセスメント結果も使いながら、システムを他のデータセンターに移設した場合に、運用にどんな影響があるか洗い出してもらいたいんだ。」

「わかりました。基本的にはシステムのデータセンター移設によるアウトソーシングということになりますね。私も運用面が一番大きく変わるのではと思うので、運用を重点的に他の非機能要求も見てみます。」

「それは助かるよ。運用以外も洗い出しをしてもらえるのは安心だね。」

「ただし、移設先のデータセンターの情報をもらえないと困りますよ。」

「それは資料一式をメールで送信しておくよ。運用以外に気になる点があるのかい？」

「データセンターの環境について気になります。当社のマシンルームよりレベルが下がることはないかもしれませんが、非公開になっている部分がないか、人の出入りや移設における媒体の持ち込み・持ち出しなども気になるところです。」

「非公開になっている部分や利用できる媒体か。それについては気になる項目を要求事項としてまとめてもらえば、データセンター側に情報提供を依頼してみるよ。」

「わかりました。それでは非機能要求グレードのアセスメント結果をもとに、まずは運用面と環境面を優先して、移行期間中の作業も含めて洗い出しを進めます。」

解説

システムの運用が大きく変わるタイミングの多くは、システムのおかれる環境が大きく変化する時でしょう。設置場所が変わったり、ハウジングやホスティングなどのアウトソーシングサービスを活用したり、クラウドなどのサービスを利用するようになったタイミングが該当します。

こういったケースではサーバー機器などのハードウェア故障時の連絡フローが変化したり、入退室に関する時間や人的制限が変化したり、ロケーションにより駆けつけ時間が変化したりします。

このように、システムの運用環境やシステム基盤自体が大きく変更となるタイミングでは、非機能要求の見直しや再定義が必要になります。非機能要求グレードを活用することで、見落としや検討漏れがないように現行と変更後の非機能要求の差異を把握することが可能です。その結果、もし変更後には要求レベルを満たせていないことが判明したならば、事前に対策を検討することができます。

藤原君は、自社のマシンルームで稼働しているシステムに対する最新の要求を非機能要求グレードを使ったアセスメントという手法であらかじめ把握していました。それが今回のデータセンター移設の検討に大いに役に立ちそうです。

このケースでは移行後に特に運用や環境がどう変化するかを、まずはデータセンター側から提供されている資料をもとに整理することにしました。非公開となっている部分については、移設後に満たしたい要求レベルを明確化し、データセンター事業者への要求事項として整理することにしています。要求事項が明確になっていれば、データセンター事業者側も要求を満たせるのか確認したり、満たせない場合には代替策を提案したりすることが容易になるでしょう。

また例えば、運用スケジュールが大きく変更となるような場合を考えてみましょう。非機能要求グレードを使って変更前後の要求を確認することにより、運用・保守性の運用時間だけでなく重複項目となっている可用性の運用スケジュールについても影響が出ることに気づくかもしれません。

それから、非機能要求グレードには移行スケジュールや移行対象といった移行に関する非機能要求も示されています。これらの要求を整理しておくことで、移行をスムーズに計画・実施できます。

このように、非機能要求グレードをうまく使うことによって色々な視点から要求を整理することができ、見落としや検討漏れを防ぐことが可能となります。



3.7 情報セキュリティに関する新たな脅威に備える

雨の降る秋の夕方のこと、情報システム部の神谷部長は困ったような表情で、部下の富田さんに声を掛けた。

「富田さん、ちょっと相談があるんだ。システム運用についてもそろそろ次のステップに進める必要があると思っていてね。」

富田さんは不思議そうな顔をして答えた。

「はい、どういったご相談でしょうか。ここ最近では運用トラブルは起きていませんし、運用マニュアルや手順書の整備、訓練についてもかなり対応を進めてきたところです。確かに性能面については不安があると認識していますが、運用面でそれ以上と言いますと・・・」

「まあ、そのあたりは富田さんの働きに感謝しているよ。だからこそ今回も富田さんをお願いしたいと思っていてね。実は運用の体制に関することなんだが・・・」

それでも具体的に相談事項が思い浮かばない富田さんが怪訝そうな表情を浮かべると、神谷部長はこう続けた。

「実はね、セキュリティの話なんだ。最近マルウェアによる感染や情報漏えいなどのニュースが後を絶たないじゃないか。経営層も、うちの対策とかいざという時の体制とか大丈夫なのかと気にし始めていてね。これまでもセキュリティインシデントを起こさない前提で色々と対策はしてきているんだが、ここのところ標的型メールとか攻撃パターンも多様化しているだろう。対策や訓練はしていても、万が一にも何かあった場合のことはあまり考えてなかったと思っていてね。」

富田さんは少し考え込んで、なにか思いついたように返事をした。

「てっきり性能面が優先課題と思っていたので、次期システムもそこから検討を開始しようと思っていましたが、セキュリティに関する対応体制の方が優先課題のようですね。わかりました。」

「そうだね。『セキュリティインシデントは発生してしまうものである』ということを前提として、早期に発見し、システムの被害を最小限にとどめる対処をしたり、復旧方法を検討したりする体制も新たに必要だと思うんだ。」

「言われてみると確かにそうですね。ただ常にそういった体制を用意するとなると、ベンダーも含めた体制維持のためのコストも相当額になりますね。都度体制を敷くのでは対応が遅くなってしまいますし……。わかりました。セキュリティ体制を強化した場合にコスト面も含め他の要件と調整できないかなど、非機能要求グレードを使って整理してみます。」

「それじゃあ社内のセキュリティ管理部門にも声をかけておこう。体制面や対応方法で相談する必要があるだろう。それに、システムごとにセキュリティ体制を用意するのではなく、全社横断でセキュリティ対応チームを作るといったことも検討が必要になるかもしれないね。」

「はい、よろしくお願いします。」

富田さんはさっそく非機能要求グレードを使って、次期システムにおけるセキュリティ体制の検討を開始した。

解説

非機能要求グレードでは、まずはモデルシステムの選定、重要項目のベース値の検討、残りのメトリクスの検討を進めると段階的に素早く非機能要求を決定することができますとしています。ただし、直面する課題や優先して決めなければいけない要件が明確にある場合には、そこから手を付けるというのが良いでしょう。

昨今では、標的型攻撃など組織を標的としたサイバー攻撃が巧妙化してきており、それを完全に防御することは困難となっています。このため、セキュリティインシデントが発生することを前提とした早期復旧を目指す対応を、あらかじめ検討しておくことが必要となってきています。非機能要求グレードを活用すれば、このような内容も要求事項として整理することが可能です。

このケースでは、富田さんは次期システムの要件を、非機能要求グレードを活用して、セキュリティインシデント対応／復旧の検討から着手することになっています。現行システムから大きく変更となる項目を優先することにより、それによる影響などを考慮しながら効率的に検討を進めることができます。

新たにセキュリティインシデント対応／復旧を検討する中で関連部門とも相談しながら、万が一の場合に迅速に招集ができる体制を作るのか、常設の体制を作るのか、さらには社外にアウトソースするのかなど、コストへの影響も考慮して検討を進めていくことになります。



3.8 ビジネス環境の変化に合わせて要求を見直す（クラウド）

事務機器メーカーのグリーン社は商社や特約代理店を通して販売網を拡大してきたが、昨年度からインターネットを介した直販事業に乗り出している。

従来は社内と取引先企業に限定したシステムのみ保有していたが、直販の新システムではインターネットに接続する必要があるため、思い切ってクラウドを利用することとした。無事本稼働してから1年が経過し、特に大きな障害もなく、直販の売り上げも予想通り伸び、順調な滑り出しである。

ある日、直販の担当部署との打合せの中で直販の「一周年記念キャンペーン」の話が持ち上がった。IT部門の係長である佐倉は打合せが終わると、上司の布川部長にキャンペーン対応の話を早速切り出した。

「部長、例の一周年記念キャンペーンの件なのですが。」

「ああ、その件ね。聞いているよ。どんな内容になりそう？」

と、顔を上げた布川に、佐倉は嬉しそうに答えた。

「クイズ形式のページを立ち上げて応募してもらい、抽選でクーポンを発行します。」

「なんだ。賞品とかはないのか。一周年という割には今一つだな。」

布川のがっかりした様子を見て、佐倉は慌てて付け加えた。

「いえ、一等賞だけは目玉として何か考えているらしいですよ。」

布川はキャンペーンを大いに盛り上げるアイデアがまだ出ていないことに苛立ちを覚えながらも、前々から気になっていた点を切り出した。

「ところで、今回のキャンペーンは専用のサイトを立ち上げるのかい？」

「いえ、クラウド上の既存のサーバー上にキャンペーンのページを差し込むだけです。特に今までも大きな問題はありませんでしたし。」

布川はやはりと思いながら言った。

「え？大丈夫なの？クラウド事業者のSLAは確認したの？」

佐倉はあまり気にする風でもなくこれに答えた。

「いえ、特に。拡張性を気にしなくていいのがクラウドの利点ですからね。今回くらいなら大丈夫だと思っています。これまでだって、月末に若干アクセスが集中しても、レスポンスが遅くなったりアクセスできなかったりしたことはありませんでした。」

布川はやれやれと思いながら言った。

「それはどうかなあ。今回はキャンペーンなんだろう？メディアにも広告を打つんじゃないかな？イベントが盛り上がったらどうなる？確認しといた方がいいぞ。それに、キャンペーンでは個人情報も集めるんだよね？セキュリティは考慮しているの？」

するといつも賑やかな佐倉もしばらく黙ったあと自信なげに言った。

「……。確かにそうですね。ちょっと心配になってきました。クラウド事業者とのSLAはあるので、確認しておきます。ただ、比較検討材料がないな……。」

布川は気になっていたことを伝え共有できたことにひとまず安堵した。

「それなら、非機能要求グレードを使ってみたい。性能やセキュリティだけではなく、可用性や移行に関するメトリクスもあるし、具体的なレベルで示されているから比較に使えると思うよ。」

「わかりました。すぐに対応します。」

佐倉は、布川の余裕のある表情を見て少しホッと、またいつもの笑顔を取り戻した。

解説

このケースのグリーン社は、初めて直販のサービスを開始して1年が経過し、事務機器の売り上げ拡大と顧客獲得のためのキャンペーンにチャレンジしようとしています。このキャンペーンでクラウドの拡張性を活用することになれば、クラウドのメリットを早速享受できます。

このように、システムを構築し時間が経過すると、構築当初とはビジネスの状況が変化し、新たなビジネス上の取組みを始めることも多いでしょう。ここで注意しておきたいのは、こういった取組みを行うことで、ピーク時のアクセスやセキュリティに関する要求も変化する可能性がある点です。また、クラウドはビジネスの変化に柔軟に適応することを謳うサービスですが、キャンペーン時の突発的な負荷にすぐさま対応できるとは限りません。

このケースにあるグリーン社の場合、クラウド基盤を新規に導入した当初は、導入時のビジネス環境にふさわしいものであったでしょう。しかし、ビジネスとシステム基盤の変化は相互に影響し合うものです。クラウドの採用後に新たなビジネス上のチャレンジを行うのであれば、クラウドを採用した当時の要求もまた見直すことが必要です。

このケースでは、非機能要求グレードを活用して、直販のシステムの平常時の要求とキャンペーン時の要求の差異をチェックしたり、クラウド事業者と考慮漏れがないかを確認することが有効です。クラウド事業者は、過去のサーバーにおける負荷の統計情報を提供して相談に応じるかもしれません。場合によっては、クラウド事業者に一定期間特別な監視体制を依頼すべきかもしれません。

非機能要求グレードを用いたチェックとしては、「性能・拡張性」と「セキュリティ」の項目を重点的に確認する必要があるでしょう。キャンペーンによる突発的なアクセス集中は「性能・拡張性」の項目を用いて検討することができます。また、キャンペーン時に登録される個人情報への扱いは「セキュリティ」の項目を用いて検討することができます。検討に漏れがないかを確認するため「システム環境・エコロジー」にある利用者数などの情報を整理し、システムに影響する事項が他にないか確認することも有効でしょう。

なお、今後、クラウドに続く形で新たな技術トレンドが現れてくるでしょうが、「要求が変化したか」を継続的に確認することや、クラウド基盤を見直し続けることが必要であることに変わりありません。この継続的な見直しにおいて鍵となるのは「要求」であり、あまり目立たない潜在的な「要求」の変化も含め幅広くチェックする必要があります。

非機能要求グレードは、クラウドの導入・開発のみならず、導入後も継続的に活用できるツールなのです。

TIPS 開発コストと運用コストのトレードオフ

非機能要求グレードにおいて「運用コストへの影響」欄に○がついているメトリクスは下表のとおりです。これらは特に「開発コストをかけることで運用コストを下げられる可能性のある」メトリクスとしています。開発コストと運用コストがトレードオフの関係にあるメトリクスであるとも言えます。

表 開発コストと運用コストがトレードオフの関係にあるメトリクス

項番	大項目	中項目	小項目	小項目説明	重複項目	重要項目	メトリクス(指標)	運用コストへの影響
A.1.1.3	可用性	継続性	運用スケジュール	システムの稼働時間や停止運用に関する情報。	○	○	計画停止の有無	○
A.1.2.2			業務継続性	可用性を保証するにあたり、要求される業務の範囲とその条件。		○	サービス切替時間	○
B.3.1.1	性能・拡張性	リソース拡張性	CPU 拡張性	CPU の拡張性を確認するための項目。将来の業務量の増加に備え、どれだけ CPU に余裕をもたせておくか、どれだけ CPU を拡張できるようにしておくかを確認するための項目。		○	CPU 利用率	○
B.3.1.2						○	CPU 拡張性	○
B.3.2.1			メモリ拡張性	メモリの拡張性を確認するための項目。将来の業務量の増加に備え、どれだけメモリに余裕をもたせておくか、どれだけメモリを拡張できるようにしておくかを確認するための項目。		○	メモリ利用率	○
B.3.2.2						○	メモリ拡張性	○
B.3.3.1			ディスク拡張性	ディスクの拡張性を確認するための項目。どれだけディスクに余裕をもたせておくか、どれだけディスクを拡張できるようにしておくかを確認するための項目。			ディスク利用率	○
B.3.3.2							ディスク拡張性	○
C.1.2.4	運用・保守性	通常運用	バックアップ	システムが利用するデータのバックアップに関する項目。		○	バックアップ自動化の範囲	○
C.1.3.1			運用監視	システム全体、あるいはそれを構成するハードウェア・ソフトウェア（業務アプリケーションを含む）に対する監視に関する項目。		○	監視情報	○
C.1.4.1			時刻同期	システムを構成する機器の時刻同期に関する項目。			時刻同期設定の範囲	○
C.2.1.1		保守運用	計画停止	点検作業や領域拡張、デフラグ、マスターデータのメンテナンス等、システムの保守作業の実施を目的とした、事前計画済みのサービス停止に関する項目。	○	○	計画停止の有無	○
C.2.1.2							計画停止の事前アナウンス	○
C.2.2.1			運用負荷削減	保守運用に関する作業負荷を削減するための設計に関する項目。		○	保守作業自動化の範囲	○
C.2.2.2							サーバソフトウェア更新作業の自動化	○
C.2.2.3							端末ソフトウェア更新作業の自動化	○
C.3.2.1		障害時運用	障害復旧自動化の範囲	障害復旧に関するオペレーションを自動化する範囲に関する項目。			障害復旧自動化の範囲	○
C.4.3.1		運用環境	マニュアル準備レベル	運用のためのマニュアルの準備のレベル。		○	マニュアル準備レベル	○
C.4.4.1			リモートオペレーション	システムの設置環境とは離れた環境からのネットワークを介した監視や操作の可否を定義する項目。		○	リモート監視地点	○
C.4.4.2						○	リモート操作の範囲	○
C.5.1.1		サポート体制	保守契約（ハードウェア）	保守が必要な対象ハードウェアの範囲。		○	保守契約（ハードウェア）の範囲	○
C.5.2.1			保守契約（ソフトウェア）	保守が必要な対象ソフトウェアの範囲。		○	保守契約（ソフトウェア）の範囲	○
F.4.3.1	システム環境・エコロジー	機材設置環境条件	重量	建物の床荷重を考慮した設置設計が必要となることを確認する項目。			床荷重	○
F.5.1.3			環境負荷を抑える工夫	環境負荷を最小化する工夫の度合いの項目。			機材のライフサイクル期間	○
F.5.2.1		環境マネジメント	エネルギー消費効率	システムの仕事量をそのエネルギー消費量で除した単位エネルギー当りの仕事量のこと。			エネルギー消費の目標値	○
F.5.3.1			CO ₂ 排出量	システムのライフサイクルを通じて排出されるCO ₂ の量。			CO ₂ 排出量の目標値	○
F.5.4.1			低騒音	機器から発生する騒音の低さの項目。			騒音値	○

4 章

標準化・アセスメント・展開

稼働中システムのアセスメントや社内システムに対する非機能要求の標準化の場面などでも、非機能要求グレードが役に立ちます。

例えば、システム更改にあたって次期システムの要件定義を行う必要が生じた時、システム監査の際に第三者にシステムのありようを正確に伝えなければならなくなった時、そのようなときどのように伝えるのが良いのでしょうか？ また、システム稼働時の非機能要求を把握していたとしても、現在もそれが有効であるか確認までできているのでしょうか？非機能要求グレードはシステム化の要件から漏れやすかったり、ステークホルダー間の認識齟齬が生じやすかったり、決めにくかったりする非機能要求について、238 のメトリクスに最大 6 段階のレベルで例示されています。さらに、そのうちの 92 項目は重要項目として定義されています。これを利用して既存システムの非機能要求を網羅的にアセスメントすることができるのです。具体的には要件定義段階で決めた要求がきちんと実装されているかどうかをレベルを用いて点検します。この手法は経済産業省の「非機能要求グレード評価委員会 実証評価報告書」¹⁾ にも記載されています。

非機能要求グレードを用いてシステムの新規構築、保守開発・更改、既存システムのアセスメントなどを実施すると、暗黙知の状態であったシステムに求められる非機能要求レベルが明瞭になってきます。もちろん求められる非機能要求は、取引先や顧客への対外影響の有無、ビジネスへの影響の多寡により、各システムによって様々です。また、セキュリティやシステム環境・エコロジーに関する事項を中心に、各種法律や規制、企業のビジネスポリシーなどにより必ず具備しなければならない非機能要求もあるでしょう。システム構築をする際に利用する環境（プライベート／パブリッククラウドやホスティングサービスなど）の制約条件もあるかもしれません。これらの集合体として得られた非機能要求を「社内グレード」として標準化し、くくり出すことによって効率的に要件定義を進めることも可能です。具体的には、項目一覧（活用シート）に社内システムで共通の非機能要求を可能な限り明確にする作業を行います。これにより個々のシステム開発時にステークホルダーとの非機能要求にかかわる検討項目の絞り込みができ、上流工程の短縮、効率的な開発につながります。

非機能要求グレードは、工夫次第で活用の可能性は無限大です。様々な場面で活用が進み、実践的な社内グレードとして標準化すれば、より簡便かつ十分な網羅性で非機能要求を検討することが可能となるでしょう。

本章では、非機能要求グレードを活用したアセスメント手法や社内標準として用いるメリットなどについて具体的なケースを用いて解説していきます。

1) www.meti.go.jp/policy/it_policy/softseibi/grade.pdf

4.1 社内標準を作る

駒込ネットワークス社は、ニュース配信に始まり、現在では多種多様なネットコンテンツを会員に提供し、スマートフォンやタブレットといったモバイル端末の普及とともに急成長を遂げた企業である。現在は、先に開発した大人気ゲームのオンラインコンテンツが、旧作を含めて5シリーズ稼働し、オールドファンからの支持もあり大人気を博している。同社のビジネスプランが的中した結果であるが、非機能要求の取りまとめに非機能要求グレードを用いたことで、サービスのスローダウンやデータベース・ファイル容量の逼迫といったサービスレベルでの劣化の問題が発生していないことも人気を支えている一因と考えられている。

また、同社が運営するオークションサイトは稼働から5年以上を経過し、ハードの老朽化対応に加え、競合サイトの新機能に対抗するために早急なシステム刷新が必要と判断されていた。その再構築プロジェクトにおいても、大人気ゲームと比較すると求められるサービスレベルには違いがあるものの、同様に非機能要求グレードを用いた非機能要求の取りまとめを行った結果、オークションサイトの再構築も期限内に完了し、取り回しがよく、運用負荷の少ないシステムとして社内でも好評となっている。

そうした中、システム部長、コンテンツ事業部長、財務部長、経営企画部長、5支社長といったシステムに関連する社内キーパーソンが社長から直々に招集された。

「今日集まってもらったのは、今後の事業展開について私の思いを伝えることと、それを実現するための方法論について議論したかったからだ。みんなの努力のおかげで、今のところ我が社の業績は極めて順調に推移している。だが、みんなも承知しているように、この業界は常に競争状態にある。ライバルに先駆けて魅力のあるコンテンツを提供し続けて行かなければ顧客はすぐに離れて行ってしまうだろう。」

社長は続けた。

「そこでだが、次の開発の切り口を『ご当地コンテンツ』としたいと考えている。コンテンツの内容は、各支社の事業部門でどんどん検討を進めていて、第一陣の内容はほぼ固まっている。気付いていると思うが、第一陣だけではなく、今後この切り口で検討したものがコンテンツ候補として続々と挙がってくる。これを順次リリースしていくわけだが、とにかくスピードを最優先としたい。一つのシステム開発に従来のような時間をかけることはできないと考えてもらいたい。」

「社長、そういうことでしたらご提案したいことがあります。」

システム部長の田中が語り始めた。

「一連のオンラインゲームの開発や、オークションシステムのリプレースの中で、我が社のコンテンツシステムに求められる非機能要求像というものが出来上がってきたと思います。私が考える標準形は二つで、キラーコンテンツ向けの非機能要求、もう一つは、それ以外の通常コンテンツ向けとなります。今後、開発を進めてゆく『ご当地コンテンツ』がそのどちらに属するコンテンツなのかを判断し、全てに二つの標準形を当てはめて開発を進めるというのはどうでしょう。要件定義からシステム設計までのリードタイムを短縮することができるかもしれません。」

「いいですね、我々もコンテンツの中身に注力できそうです。」

コンテンツ事業部長の斎藤やその他の支社長らからも異論はなく、社長も少し迷いはあったようだが決を下した。

「多少の無駄が出るかもしれないが、やってみよう。進めてくれ。」

「では、非機能要求の社内グレード作成チームを立ち上げ、早急に実プロジェクトに適用可能な非機能要求を取りまとめ、ご報告いたします。」

こうして策定された社内グレードは、順次プロジェクトに適用され、当初の狙いどおり矢継ぎ早にコンテンツをユーザーに提供する大きな原動力となった。

解説

非機能要求グレードが提供するツール類が活躍するのは、基本的に情報システムの新規構築時になるでしょう。機能要求に関しては、その要求が正しく実装されない場合、システム開発のそもそもの目的を達成できないことになるので、慎重かつ十分な検討、および要求定義が行われます。一方で、非機能要求は時に致命的な結果をもたらす場合もありますが、往々にしてユーザー部門や運用部門の妥協を引き出せば回避可能であったり、事後的に対応することでリカバリ可能であったりすることも多く、システム開発の上流工程で割付けられるリソースの優先度は比較的低くなっているのが実情ではないでしょうか。非機能要求グレードは、そうした実態をある意味で肯定し、広範囲にわたる非機能要求を効率よく、かつ網羅的に検討するツールとして開発の現場からのニーズに応えたものです。

ただ近年、企業の業務はほぼシステム化され尽くした感があり、全くの新規構築プロジェクトはほとんどない、というのもまた真実でしょう。では、そうした環境下で非機能要求グレードは活用できないかという点、そうではありません。非機能に関する要求は社会の変化やより高度な IT 技術の普及を伴う時間の経過とともに変化してゆくものです。システムのリプレイスやシステム基盤の再構築だからといって、非機能要求が所与のものとして定まっているものではありません。非機能要求の検討という場面では、全く同じというわけではありませんが、新規構築時と同様の手順を踏むべきものです。

非機能要求グレードを用いた開発サイクルを繰り返すことで、社内システムの暗黙知が文章化され、形式化されることにより、その企業の具体的な非機能要求レベルや非機能要求の合意プロセスが純化し、システム開発の場面で共通的に利用可能なナレッジとして蓄積されてくるでしょう。

駒込ネットワークス社の田中システム部長が「キラーコンテンツ」といった、企業独自の切り口を使ってシステムのレベルを分けたように、そのナレッジを活用することで、情報システムの開発における非機能要求を、より効率よく、かつより網羅的に（言い換えれば、必要かつ十分に）検討することができるものと考えています。ただ、ここで留意が必要なのは、ナレッジから導かれた社内グレードも時間の経過とともに見直す必要があるということです。もちろん、個々のシステム開発で個別に見直しを実施するよりも遥かに作業負荷は少ないことは言うまでもありません。

4.2 プロジェクトのレビューに使う

スガモ商事は、先の大規模開発の際、非機能要求が不明確な状態で開発を進めた結果、ショッピングサイトでの致命的なスローダウンを発生させてしまった。アプリ開発の大幅な遅延の解消に全精力を注ぎ込み、非機能要求の検討がおざなりになったのだ。

これに端を発し、同社のPMOが所管する全ての開発プロジェクトのレビュー観点に、非機能要求の検討状況を加えることを決めた。必要な資料や運用の検討を任されたPMOの志村は、各プロジェクトがどのようなものを用いて非機能要求を定義しているのかを紐解くことにした。

「大方似たような資料を使っているに違いない。うまくマージすれば、そんなに時間をかけずに原案が作れるだろう。」

ところが、志村の予想は裏切られた。各システムの検討項目や記述レベルに結構な差があり、一筋縄ではいきそうにない。一から作るしかない諦めかけた瞬間、志村はふと思いついた。

「そう言えば、非機能要求に関するツールがあると聞いたことがあるな…」

志村は資料を検索してダウンロードすると、非機能要求グレード活用シートを眺め始めた。

「なるほど。システム基盤が前提のようだが、項目の網羅性は十分だ。ボリュームはあるけど重要項目の提示があって、項目毎のレベルの差も分かりやすいな。」

アプリのことを考えると、既存資料を適当にマージし、アプリ関連の項目を追加して原案とすることも可能だが、時間はそれなりにかかるそうだ。一方で、活用シートの高い網羅性も捨てがたい。あれこれ思案した結果、志村は上司であるマネージャーにこう提案した。

「非機能要求の検討漏れを極少化するなら、網羅性の高い資料を使うのが得策です。アプリ用の項目検討は次のステップにして、まずはシステム基盤部分から進めませんか？」

「よし、それで行こう。ここから先はインフラチームと詰めてくれ。」

スガモ商事はデータセンターの統合を機に、社内環境をプライベート・クラウドで構築した。開発したシステムは一部の例外を除いて、この環境で運用するのが基本方針だ。

「社内環境のサービスレベルは明確なはず。確定している項目とレベルを明示すれば、非機能要求の検討も確認もスピーディに行うことができるだろう。」

志村は、活用シートをコピーして社内環境用シートを作り、プライベート・クラウドを利用する場合は社内環境用シートを、その他の場合は別の活用シートを使うことにした。

インフラチームとの作業が始まると、いきなり担当の春日がこう指摘した。

「インフラ側では決められないものもたくさんあるなあ。例えばデータの暗号化。三つのレベルのどれを選択するかはユーザー要求次第でしょ。目標復旧レベルなんかもそうだね。」

言われてみれば当たり前のことだ。志村は、アプリ開発側から見たシステム基盤の項目レベルは、基本的に一意に決まるはずと思い込んでいたのだ。そこで志村は、「社内環境のためアプリ側での考慮不要」「アプリ側の意向でオプションあり」「アプリ側で考慮すべき事項あり」という三つのカテゴリを設け、活用シートの項目毎に明記した。レベルが確定している項目は該当セルに色を付け、シートのヘッダーに「利用ガイドを必ず参照すること」「樹系図等を適宜参照すること」なども追記した。

非機能要求グレードには色々な活用の仕方があります。その一つに、「既存のシステム基盤がどのような状況にあるのかを客観的に評価すること」を挙げることができます。スガモ商事が作成した「社内環境用シート」は、まさにこの目的のために利用できるものです。スガモ商事のように、分散していたデータセンターを統合し、集約的なシステム環境を構築しているような場合は特に、システム基盤が企業のインフラ方針に合致していることの確認や、世の中の動向に合わせたシステム基盤（システム環境）の見直しなどが実施しやすいと言えます。

もちろん、運用しているシステムの非機能要求が一律同じレベル（グレード）とは限りません。スガモ商事を例に取れば、レベルの差はアプリ開発側の意向や考慮事項によって生じているものです。言い換えれば、インフラ方針で決まっている部分は、運用対象システムの大小に関わらず、（仮にオーバースペックであるとしても）一定のレベルは保証されるということになります。スガモ商事のデータセンターは一カ所ですが、センターを複数保有している企業であれば、その拠点毎に評価しておく必要があるでしょう。

また、検討すべき非機能要求項目のうち、既に決まっている（アプリ側の要求には応じられない）もの、選択の余地があるもの、アプリ側で考慮すべきものが明確になっていることで、責任分解点が明確になると同時に、その分スムーズに非機能要求の検討を行うことができるようになります。新システムを開発する際、当然ながら、決まっていることを改めて検討する必要はありません。アプリ側は、自らが責任を持って決めるべきものだけに注力すればよいわけです。

スガモ商事のPMOのようなアセスメントを絡めた使い方も有用ですね。プロジェクトのどの段階でレビューを実施するかは企業毎に設定すればよいのですが、少なくとも「この項目は〇〇フェーズまでに決まっていなければ、カットオーバーに間に合わなくなる」というものがあると思います。これに引っかかるような状況に陥っていないかどうかを、タイミングよくチェックできると効果的です。さらにレビュー観点に組み込むことで、「忙しいから非機能要求の検討は後回しでいいや」ということに対する抑止力が働くのも大きなメリットです。

ともすると優先順位を落としてしまいがちな非機能要求の検討ですが、ツールの活用と効率的な運用によって現場の負荷を抑えるとともに、適切な時期に十分な検討が行えるような仕組みを確立することで、効果をあげた一例といえるでしょう。



4.3 システム統合時に現状を比較する

ブルームーン製薬工業は、今年医薬品などを中心として事業を推進してきたブルーメディスン社と、バストイレタリーの自社商品を中心とした販売網を強みとするムーンケミカル社がお互いの強みを活かし事業統合してできた新会社である。来年度からは様々な社内システムの統合を予定しており、統合の効果を極大化するため、それをできるだけ早く、スムーズに行う必要がある。そのための第一段階として、統合効果の高い勤怠や給与計算などを司る社内システムの統合がまず議論されることとなった。

先月、その社内システムの統合化に関する最初の会議が開かれた。その席上、口火を切ったのは旧ムーンケミカルの情報システム部の大山部長だった。

「ムーンは社員数が元々 8,000 人、ブルーさんは元々 3,500 人と聞いていますから倍以上です。影響度合いを考えれば当然ムーンのシステムに片寄せするのが妥当でしょう。」

旧ブルーメディスンの情報システム部門の小川部長も黙ってはいられない。

「ブルーのシステムは昨年更改したばかりで、適用技術も新しく拡張性にも優れています。追加で 8,000 人を収容するのも容易でしょう。」

この後もお互いのシステムの優れている点を挙げ、どちらも一歩も引かない状態が続き先週の会議も平行線をたどったため、新たにブルームーン製薬工業の人事労務部の部長に就任した中田部長もさすがに心配になった。

「このままではいつまで経っても二つのシステムが融合できない・・・。」

中田は以前の上司で情報システム部門の統括部長を務めたこともある大倉本部長に相談に行った。

「実はシステム比較ばかりで、システムの具体的な設計どころか要求の整理さえいっこうに進んでいないのです。」と中田が切り出すと、大倉は静かに語りはじめた。

「それは本当にシステムの比較をしていると言えるのかね・・・一度標準化という視点で客観的に評価してみたらどうだろうか。『非機能要求グレード』を使ってごらん下さい。先が見えてくるかも知れませんよ。」

中田は自分の席に戻ると早速、非機能要求グレードをダウンロードしてみることにした。

「よし、明日の会議にはこれを持ち込んでみよう、今日の明日で申し訳ないが、大山さんと小川さんには旧会社のシステムの状態について、レベルに丸を付けて示してもらうことにしよう。」

中田はそれぞれにお願いに行くと、意外なことに、二人とも一向に進んでいない現状に危機感を覚えたのか作業を快諾してくれた。

明けて翌日、会議で最初に発言をしたのは中田であった。

「人事労務部の中田です。実は一刻も早く新会社の社内システムを稼働させなければならないと思い、昨日色々無理なお願いをしました。大山部長と小川部長にはそれぞれ『非機能要求グレード』を使って現行システムの分析をしていただきました。それを見ながら今日は会議を進めさせてください。」

2 時間も会議を進めると、より具体的な話が出始めました。

「ブルーさんのシステムは随分セキュリティが高く設定されていますが。」

「実は研究所で新薬の開発や臨床データを扱う関係上、高い認証レベルが必要なのです。もち

ろん入退室の ID カードにもこのシステムが連携しています。」

「なるほど、旧ムーンのシステムではその点は簡単に実現ができないかも知れませんね。」

「1 万人を超える認証をこなすとなると性能面からも相当なシステム増強が必要になりそうですよ。」

「どちらのシステムが優れているかの議論をしている場合ではなかったですね。融合したシステムが、どんな要求を満たさなければならないかもっと真剣に考えないといけなかったのですね。」

解説

ブルームーン製薬工業のケースでは事業統合時の指標として非機能要求グレードを活用しています。

事業の都合上、二つのシステムを統合しなければならなくなりましたが、二つの全く違う発想で構築されたシステムの融合や統合は難しく、その反面、片寄せというのも統合後の事業の多様化を考えるとあまりそぐいません。

一方でシステム統合の検討会議では、どちらに片寄せするかという切り口でしか議論が進まず、二つの旧組織の対立状態となり険悪な状態となります。いまだ欧米並みとは言えませんが、企業買収や事業統合といったことが、以前に比べると身近なものとなった昨今ではよく見られる光景ではないでしょうか。

こうした場合、原点に返り、統合後に求められる非機能要求を非機能要求グレードで定義してみることで、どちらのシステムに片寄せするかではなく、統合後のシステムに何が求められているかを浮き彫りにする、このような使い方も非機能要求グレードの活用の仕方の一つです。

少々特異な例に見えるかもしれませんが、これまで紹介してきた社内の複数システム間で積み上げたナレッジをベースとした社内グレードの考え方と大きく異なるものではありません。複数の企業体を跨ぐシステム間の AsIs と ToBe を、文章化・定量化することにより非機能要求の検討を「合理的」かつ「網羅的」に実施する指針となるものと考えています。事業統合のように通常よりも複雑な利害関係を持つステークホルダーが存在する場合にこそ、非機能要求グレードを利用することによる効果が発揮されやすい状況といえるのではないのでしょうか。

また、事業統合はシステム構築における従来のしがらみを取り払う良いタイミングでもあります。

例えば、それほど信頼性を要求されるシステムではないものの、当初社内の偉い人の肝いりで作られたシステムであり、なかなか信頼性の引き下げや運用コストの低減を言い出せずにいる、あるいは過去の経緯からビジネス特性上の重要度とはそぐわないコスト配分がされている、といった状況を適正な IT ガバナンスに基づき合理的な状態に導くために、非機能要求グレードがその一助になることを期待しています。

4.4 新技術の導入を検討する

大手情報ベンダーで、自社のシステム共通基盤の維持保守を担当している富坂課長は、上司である播磨部長と共通基盤の今後のロードマップについてミーティングを行っています。

「富坂さん。例のシステム共通基盤だがね、ハードの入れ替え時期はまだまだ先ではあるけれど、今後の方向性について頭の整理をしておきたい。思うに、当初想定していたよりもモニタリングなどの運用性がいまひとつ芳しくないし、一部の業務システムには専用アプライアンスを利用する結果になってしまったら、共通基盤を構築したメリットがあまり出せていないと思っているのだけれど、どう思うかな。」

「そうですね、現行の共通基盤はサーバー、ストレージ、ネットワークを個別に組み合わせた仮想環境なので、特定の業務システムがスローダウンしても、原因特定に時間が掛かる上に、有効そうな対処をしても解決できず、時間と労力の無駄が生じたこともありましたからね。おまけにフタを開けてみたら仮想化に対応していないソフトウェアの採用が必須だった、なんていうシステムもありましたし……。運用管理機能を充実させたいという所はもちろん同感ですが、共通基盤を利用するにあたってのポリシーというか、決め事を社内に浸透させることも必要ですね」

「他社では、HCI（Hyper-Converged Infrastructure）の活用が進んでいるようだが、我が社でも検討を始めたいと考えていてね。まあ、経営層から安くて簡単になるらしいじゃない？と水を向けられたものもあるのだけれどね（笑）」

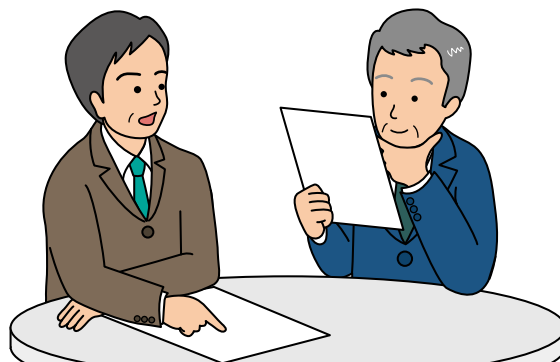
「そうですね、時間的な余裕もありますので、手始めに非機能要求グレードを使って、現行の共通基盤の状況を整理するのに併せて、共通基盤上のシステムの拡張性やピーク値などの要求水準も再整理してみましょう。HCI製品も色々ありますので、我が社の求める要求を満たす製品があるのか情報収集から始めてみるのはいかがでしょうか。」

「それは良いかもしれないね、早速進めてもらえるかな。あと、富坂さんが言っているポリシーの件も考えないといけないなあ。」

「HCIがどの程度のものか、まだ定かではないので、あくまでもジャストアイデアですが、HCIの利用を前提とした利用条件や制約事項を定めた「システム共通基盤利用規約」を取り纏めてはどうでしょうか。非機能要求グレードの項目だけではなく、運用管理の負荷を増大させるような、ソフトウェアの採用は不可とするなどの、独自の項目を設定することも視野に入れてはどうかと。」

「なるほどね、非機能要求グレードにも色々使い道があるというわけか。ただし、あまり厳しい条件にしてしまって、共通基盤を使わないシステムが出てきても本末転倒だから、気をつけないといけないね。」

「はい、共通基盤上のサービスの要求水準を再整理して、各システムが利用できる前提で調査を進めるつもりです。HCIも使い方によっては、コスト増となるケースもあるようなので、そこも要注意かと思っています。」



新しい技術や製品が次々と現れるのが、ITの世界の常ではありますが、そうしたものの導入の検討に際しても、「非機能要求グレード」が役に立ちます。

種々のクラウドサービスを利用するにせよ、垂直統合システムを採用するにせよ、システムのモダナイゼーションにおいて非機能として求められる要件は、非機能要求グレードのメトリクスに網羅的に表現されています。このメトリクス毎のレベルを参考にしながら、まずは現状を確認し、そのサービスや製品が現状もしくは、求めるべき非機能要求を満たしているのか、また、満たしていない場合にそれを容認できるのかという点につき、しっかりと把握する必要があります。

富坂さんは「運用管理機能の充実」を行いたいと考えています。HCIは、サーバー、ストレージ、ネットワークといったシステムの要素をソフトウェアにより仮想的に構成した製品です。そのため、統一的な運用管理機能を搭載しているケースが多く、ハードウェア層に加えその上に構築された仮想マシンを含めた統合的なリソース管理や、スローダウンといったトラブルの原因究明が容易となる可能性が高いと言われています。充実させたい運用管理要件に対して、検討候補となるHCIがどのような具体的な機能を搭載しているのかについて、詳細な検討を進めれば良いでしょう。

一方このシーンでは、HCIの利用を前提とした各サービスの利用条件や制約事項を定めた利用規約の策定にも言及しています。播磨部長が指摘しているように、共通基盤を利用しない（利用できない）システムが出てこないように、各業務システムの非機能要件についても再確認・整理することが重要です。例えば、各業務システムで保有するデータ量は大規模なDB容量の拡張が想定されるのか、また、業務量の増加によりCPU能力の頻繁な増強が想定されるのかといったものです。HCIは部品数も少なく安価であるイメージですが、実態は一般的なx86サーバー等で構成されています。そのため、仮想化基盤の要となるハイパーバイザーやDB管理ソフトウェアは、システム全体のCPUコア数やソケット数に応じてライセンス費用がかかる料金体系になっているケースもあり、むやみにHCIを拡張するとライセンスコストの増加によってシステムライフサイクルコストが跳ね上がることもあり得ます。このような場面でも非機能要求グレードを活用してみましょう。非機能要求グレードには品質やコストに大きな影響を与える重要項目として、シーンにも登場した拡張性やピーク値（業務処理量）を含め92項目のメトリクスがあります。モダナイゼーションの方向性を定めるような大きな舵きりの場面では、これらを重点的に検討することで失敗のリスクを大幅に低減するとともに、共通基盤を利用するシステムと利用しない（利用できない）システムの境界線が、副次的に見えてくることもあります。

また、富坂さんの語る「システム共通基盤利用規約」に関しても、採用した基盤技術のサービスレベルや製品の仕様をもとに非機能要求グレードの各メトリクスを具体化・定量化することにより明確な指針として活用可能となり、各業務システムの上流工程の省力化・効率化に資するのではないのでしょうか。

4.5 更改前に現行システムを評価する

千石フードサービスの情報統括本部長席で瀬川本部長は、「サパールデリ」の先週の注文件数のデータを確認しながら大きく頷くと勢いよく立ち上がった。

「よし、役員会で承認も得たことだし、やるなら今しかないんだ・・・」

今日は情報統括本部の定例会議がある。千石フードサービスの食事のデリバリーサービス「サパールデリ」が急成長をしており、役員会でもこの市場でエリアを拡大し積極的にビジネスを展開していくことに決まったばかりだった。しかし、半年前からは注文を司るシステムの注文件数は当初予想より上回り始めたため、性能もぎりぎりの状況になっていた。会議室に入ると今日は何か大きな発表があるのでは、と聞いていた10数名の情報統括本部の部下たちは、既に席に着いていた。瀬川本部長は静かに話し始めた。

「当社の大きなトライアルとして昨年から手掛けてきた『サパールデリ』は、幾多の困難はあったが、みんなの協力でシステムも安定稼働しており、注文数も順調に伸び続けている。このサービスもいよいよ軌道に乗ったといっていいただろう。」

すると部下の井本次長が素早く手を挙げコメントした。

「本部長、ですが・・・既にシステムは悲鳴を上げている状態です。」

「井本君、まさにそれだよ。当初の状況とは違ってきている。システムの見直しをする時期にきているんだ。それもできるだけ早く対応したいんだ。」

「それはそうですが、システムの更改はいつ頃とお考えですか？」

「井本君、システムが悲鳴を上げているんだろう？すぐにでも・・・だ。」

それを聞いて井本次長は少し考えながらこう言った。

「しかし・・・本部長もご存じのとおり本格稼働のトラブルもありましたから、当初のシステムからはかなり見直した経緯があります。ですから当初の定義した要求も幾分変わっていますし、要件定義からやり直すとなるとそれなりに時間もかかります。」

これには瀬川本部長も腕を組んで黙って天井を見つめるしかない。重苦しい空気一杯になったまま散会となった。

会議の後、瀬川本部長は井本次長に声をかけた。

「本格稼働の時にはトラブルがあったね。あの時は君があっという間にシステムの見直しをして見せたじゃないか。あの時はどうやったのだったかね？」

井本次長は思い出しながら答えた。

「確かあの時は・・・非機能要求グレードを使ったんです。・・・その後、集計システムの再構築でも非機能要求グレードを活用して要件定義したと記憶しています。」

それを聞いて瀬川本部長は井本次長に指示を出した。

「今回も要件定義は非機能要求グレードを使うんだ。だが今回は一からやり直すんじゃない。現行システムの要件定義で使ったシートがここにある。これのレベルを見直す形で進める。これならトラブル対応で現行システムが当初と変わってしまったところと、次期システムで変更したいところだけに注目すればいいはずだ。」

探し出した活用シートを見ながら井本次長も頷いた。

翌日の会議で、井本次長はプロジェクトで映し出した非機能要求定義シートを示しながら説

明を始めた。

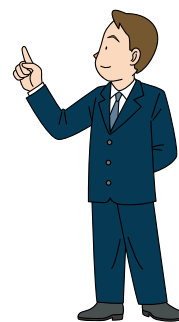
「昨日、現状の非機能要求定義シートを元に見直しをかけてみました。一番大きな変化は利用するユーザー数です。ひやかし程度の潜在ユーザーもいますが、一度でも注文しているユーザーを数えたところ当初の想定の際に倍近くに達していました。それから、以前から気になっているセキュリティもレベルアップすべきでしょう。営業部門から何度かリクエストのあったサービスエリアの拡大ですが、懸案通り来年実施が承認されるという前提で考えると、拡張性についてもいくつか見直しをしておくべき項目が見つかりました。」

見直した項目やレベルについて夢中になって説明を終え、井本次長が振り返るとそこには飯塚社長が笑顔で立っていた。

「いや、ちょっと気になっていてね。なるほど、いわゆる現状のシステムをきちんと把握するところから始めるアセスメントの手法だね。井本君も瀬川君もやるじゃないか。」

解説

多くの業種業態でシステム化が進められてきた結果、ここ最近ではシステム開発の多くが新規開発ではなく既存システムの再構築が占めるようになってきました。ところが要件定義は、「現行通り」などとしてないがしろにされるケースがありますし、既存システムの最初の構築時から時間が経ち、当初の要求と実際のシステムが乖離していることも多いようです。



非機能要求グレードは、短期に精度の高い非機能要件定義を行いたいというニーズを満たすツールですが、システムの再構築でも力を発揮します。現行システムがある場合には、まず現状どのような非機能が実装されているのかを調べてみる（アセスメント）ことから始めます。非機能要求グレードの項目一覧を使ってそれぞれのメトリクスの実現されているレベルを調査していくわけです。一通りのメトリクスを調査し終わった段階で、次期システムの非機能要求を決めていきます。既に現行システムの「アセスメント」は完了しているわけですから、現行システムから次期システムで要求レベルを変更したいメトリクスだけ見直しをしていけば良いわけです。

このように一度アセスメントを行っておくと、システム更改のタイミング毎に一から非機能に関する要件定義をやり直すのではなく、どのメトリクスをどう変更していきたいかと言うことに注力すれば、短期間に精度の高い明確な要件定義が実現できるようになります。

さらに、こうした手法による複数のシステムでのアセスメントを進めていくと、それぞれのシステムの特徴や非機能要求の実現レベルがはっきりしてきます。例えば同じデータセンターに置かれていたり、複数システムで同一の基盤を適用していれば、その部分の要求を括り出すことで、要件定義はさらに効率化できます。括り出した共通の要求を持つシステムをまとめて管理していくことでガバナンスの強化にもつながります。

非機能要求グレードは、様々なシチュエーションで力を発揮するツールなのです。

4.6 多数のシステムに対して利用状況を点検する

石田は、化学製品からアパレルまで手掛ける千石ホールディングスの情報システム部の一員だ。千石ホールディングスは基幹系や情報系、EC サイトまで数多くのシステムを抱えており、それらのシステムを統括するのが情報システム部の仕事である。

ある日、石田は部長の鈴木から声をかけられた。

「石田君、先月、会員対象のポイントボーナスキャンペーンのサイトが大量のアクセスにより停止してしまった件だが。」

「はい。どうかしましたか？」

「比較的早く復旧したことで、特に幹部役員からお咎めはなかったのだが、再発防止は図らないといけないと思っていてな。今回の根本原因は何だと思う？」

「キャンペーン期間は一時的にトラフィックが増える想定はあったのですが、そもそも、現在の会員数があんなに増えていることを基盤チームは把握していませんでした。それが分かっていたら、過負荷制御のために同時セッション数を制限するなど、事前に適切な対処がとれたのではと思っています。」

「そうだよね。そこで再発防止のために、情報システム部が抱える全システムに対して、開発当初の要件から変わっている点がないか調査を行いたいんだ。それも定期的に行いたい。来月にまずは初回の調査が行えるよう、調査要領を整理して欲しい。」

「分かりました。ユーザー数に限らず、非機能要求の全般が調査対象ですね。最近のシステムに関しては、開発時に非機能要求グレードを用いて整理をしています。開発時からの変化をキャッチアップして、現在の状況を整理してみます。それ以外のシステムについては、非機能要求グレードを使って可視化するところからですね。」

「そうだな。それがいいと思う。ただ調査対象のシステムは軽く 100 はあるぞ。非機能要求グレードの全項目について、そんなに多くのシステムの現状を整理するのは時間がかかるかもしれないな。」

「そうですね。非機能要求グレードの項目の中で濃淡をつけて、重要なところを確認するなど工夫が必要な気がします。全体を俯瞰できる樹系図も使って検討してみます。」

「頼んだよ。」

後日、石田は、非機能要求グレードの項目の中から、開発時から変化する可能性があると考えた項目を抽出し、点検シートを作成した。



表 石田が作成した点検シート

システム名		点検日			
No	大分類	点検項目	開発時の要件	現状	差分有無
1	可用性	A.1.1.1 運用時間（通常）			
		A.1.1.3 計画停止の有無			
		A.1.2.2 サービス切替時間			
		A.1.3.1 RPO（目標復旧地点）			
		A.1.3.2 RTO（目標復旧時間）			
		A.1.3.3 RLO（目標復旧レベル）			
		A.1.4.1 システム再開目標			
2	性能・拡張性	A.1.5.1 稼働率			
		B.1.1.1 ユーザ数			
		B.1.1.2 同時アクセス数			
		B.1.1.3 データ量			
		B.1.1.4 オンラインリクエスト件数			
		B.1.1.5 バッチ処理件数			
		B.2.1.1 通常時レスポンス順守率（オンライン）			
		B.2.1.2 ピーク時レスポンス順守率（オンライン）			
		B.2.2.1 通常時レスポンス順守度合い（バッチ）			
		B.2.2.2 ピーク時レスポンス順守度合い（バッチ）			
		B.2.3.2 ピーク時処理余裕率（オンライン）			
3	運用・保守性	B.2.4.2 ピーク時処理余裕率（バッチ）			
		B.4.3.1 トランザクション保護			
4	移行性	—			—
5	セキュリティ	E.1.1.1 順守すべき社内規程、ルール、法令、ガイドライン等の有無			
		E.2.1.1 リスク分析範囲			
		E.4.1.1 セキュリティリスク見直し頻度			
6	システム環境・エコロジー	F.1.2.1 運用時の制約条件			
		F.2.2.1 クライアント数			

解説

このケースのように、システムを利用するユーザーが増えていることを、業務部門は知っているがシステム部門は把握できていないということはないでしょうか。

この他にも、非常に重要な業務の一端を担う機能が追加されたことにより、ほとんど停止が許されないシステムになってしまったということもあるかもしれません。こういった非機能要求の変化を定期的に点検し、点検結果をステークホルダー間で共有することが、システムの安定的な運用を維持するためには重要です。

このような運用中のシステムの点検にも非機能要求グレードが活用できます。このケースでは、点検対象のシステムが多く、また定期的に点検を行うことから、運用中に特に変化しやすいと考えた項目を抽出し点検シートを作成しました。

どの非機能要求についても、システムの運用をしていく中で、開発時に定めたレベルから変化する可能性があります。そのため、非機能要求グレードのすべての項目について点検をすることが望ましいです。しかし、点検対象システムが多い場合や点検を定期的に行う場合などは、ある程度項目を絞ることで、効率的な点検が期待できるでしょう。その項目の絞り方には決まりはありません。このケースでは運用中に変化しやすいと考えた項目を抽出しましたが、他にも特に重要な大項目を対象にしたり、重要項目の92項目をベースに項目をピックアップしたりしても良いかもしれません。



4.7 事業継続計画はビジネス視点で検討する

「大規模広域災害を想定して、現行業務のサービスレベルを見直せ」それが、今回医薬品供給をになう白山メディカル社の情報システム部の特命チームに与えられたミッションだった。

キックオフにあたって、特命チームのリーダーである大石事業本部長が、方針を説明している。

「情報システム部主体で検討するので、システムを切り口にしたサービスレベル検討になることはやむを得ない。ただ、見たいのはあくまで業務…ビジネスだ。その考え方は忘れないで欲しい。」

ベテラン SE の設楽主任が首をかしげた。

「我々は SE ですからねえ、つつい仕組みから考えてしまう。例えばどのように考えたらいいでしょう。」

質問を予期していたらしい大石事業本部長は、笑みを浮かべた。

「例えば、在庫引き当てシステムのサービスレベルなんて決めないことだ。それでは大雑把すぎる。まず、在庫引き当てシステムで、どんな業務を行っているかを考える。」

「えーと、そうすると、受け入れ検品・払出し・発注管理・配送計画・・・。」

数え上げる設楽主任に向かって、大石事業本部長は頷いて見せた。

「そうだ。その業務毎に、広域災害が発生した場合にどうあるべきかを検討しよう。」

柴田課長が、賛成だというように頷いた。

「そうですね。例えば、広域災害が発生した場合、請求書発行と棚卸しができなくなっても当座のダメージはそれほどありません。でも、災害地域で医薬品の提供ができなくなったら、医薬品を提供するサービスとして致命的です。当社は社会的責任が果たせなくなる。」

設楽主任は、また首をかしげた。

「考え方はわかります。医薬品の提供は絶対に止められない。となると、回線の二重化、データベースの二重化は必須になる。電源もです。しかし、そうすると結局、在庫管理や配送管理のシステム全体を二重化しなくてはいけないという結論になりませんか？」

「システムが起点ではないと言った意味はそこだ。」

大石事業本部長が説明した。

「絶対に止められないのは医薬品の提供業務なんだよ。在庫管理システムなんか、別に止まったっていい。」

設楽主任と柴田課長は、あっけにとられてぽかんと口を開けた。

「止まったっていい・・・。」

「というより、広域災害を想定するなら、止まる可能性があると考えなければならない。社会インフラが破壊され、通信も交通も寸断される。回線や電源を二重化してあったところで、どちらも使えないかも知れん。だから、例えば・・・。」

「機械が止まっても、支店備付けの台帳で医薬品の在庫を確認できるとか？」

柴田課長の言葉に、本部長は大きく頷いた。

「そうそう、そういう発想を求めているんだ！」

自然災害などを契機に、BCPについての議論が活発になってきました。

事業継続のために、という観点でシステムに対する非機能要求を改めて見直す機会も多いと思います。そのような場合に注意すべき点が二つあります。

一つは、検討の前提として、想定する脅威を特定しておくことです。機器の故障や回線障害などの事故に備えるのか、事務所や建物レベルの局所災害に備えるのか、都道府県レベルの地域災害に備えるのか、あるいは広域災害までを想定するのか。それを明確にしておかないと、対策を立てることはできません。例えば、RAID ディスクによるデータの冗長化だけでは、マシルーム火災の局所的な脅威に対しては意味を成しませんし、データセンターやバックアップデータを分散しておいても、広域災害で同時に被害を受ければ役に立たなくなります。

もう一つは、システムの継続性ではなく、事業・ビジネスの視点から継続性を検討することです。システムやサブシステムなどの仕組みの単位と、ビジネスの単位は必ずしも一致しません。重要性の異なるビジネスが、同一のシステム基盤によってサポートされていることもあるでしょうし、一つのアプリケーションシステムの中に、重要度や緊急度の異なる処理が混在しているかもしれません。また、ビジネスを継続するための手段は「オンラインサービスの継続」だけではありません。システムに目を奪われるのではなく、一旦は「業務」の切り口から継続性を検討します。

対処する脅威を明確にし、ビジネスの視点から検討すると、システムに対する非機能要求をより明確にすることができます。例えば、機器故障に対しては多重化で即時リカバリできるようにすべきですが、建物被害に対してはオンラインでサービスの継続が難しいので、（オンラインサービスが停止しても事業継続ができるように）重要な業務について緊急時の手順を定めることで対応する。第二センターは必要ない、といった現実的な判断を下すことができます。

このように、非機能要求グレードでは、大規模災害時には平常時とは別に目標復旧水準を定められるようにしています。一言で災害といっても様々ですから、どのような災害の時にどのように対処するのか、システムにはどの要求をどこまで譲歩し、何が人手で行えるかなど、非機能要求グレードを使いながらより具体的に検討を進めるべきです。



4.8 サービス提供者として SLA を検討する（クラウド）

情報処理推進システムズ社は、SI 事業の将来性に危惧を抱いていた。独立系インテグレーターとして中堅にあたる同社では、不安定でリスクな SI 事業から、広く継続的・安定的に利用料を集めるビジネスモデルへのシフトを模索している。当面の目標は、自社製のパッケージソフトウェアによるクラウド型のサービス提供だ。システム構築で稼ぐのではなく、ユーザー数に応じた利用料金を今後の事業の柱にしていくという戦略だった。そんなある日、新設されたクラウドサービス部では、担当係長の富田と課長の江口の間で議論が交わされていた。

「ねえ課長、わが社がサービス提供主体になるのは初めてのことですよね。」

「確かにそうだけど、何か気になることがあるのかい？」

首を傾げた江口に、富田は答えた。

「アプリケーション機能は、パッケージ導入の経験があるので自信を持っているのですが、サービス提供となるとどうも勝手が違います。例えば、よく耳にする SLA（Service Level Agreement）ってやつは、どのように検討したらよいのでしょうか？」

江口は、自信なさそうに答える。

「そうだな、とりあえず他社サービスの利用約款を調べたらどうだ。SLA なんて、どこも似たようなものだろう。」

「約款も手掛かりにはなりますけど、ある意味ガード文言の塊ですよ。要は提供する基盤の課題やサービスメニュー全般について、検討項目を洗い出したいのですよ。」

江口は頷いた。

「うん、まさにそれを考えるのが君の仕事だ。とは言え、初めてのことから苦労しているのもわかる。少し整理してみよう。君はアプリケーションの機能の検討は大丈夫といったよね。要は、機能面は経験があるけど、非機能の検討が進めにくいってことじゃないか？ SLA って、非機能面の品質保証みたいなものだからね。」

「そうなのですよ。私にはアプリ開発の経験しかなくて、非機能を網羅的に検討したことがありません。これまでの導入案件では、非機能要求はお客様の情報システム部が中心となって検討されていました。こちらは検証支援とチューニングだけだったのです。」

江口はサービス提供型事業に対する取組みの難しさ改めて感じた。

「確かにそうだな。このビジネスモデルでは、サービスを提供する側がサービスメニューや品質保証を考える必要がある。市場での競争力を考えると、例えばセキュリティや可用性について、業界の規制やガイドラインに準拠しなければならないだろうしね。」

たまたま課長席の横を通りかかった部長の竹田が声をかけた。

「江口課長、富田君、ちょっと話が耳に入った。初めてのことで悩みが深いようだね。それなら非機能要求グレードが役に立つのではないかな。確か可用性の項に SLA についての記載があったし、法や規制などのコンプライアンスについてはセキュリティに関する事項として記載されていたと思う。とっかかりとして充分なんじゃないかな？」

「その非機能要求グレードっていうのは、クラウド上のサービスにも適用可能なのですか？」

江口の質問に、竹田部長は笑みを浮かべた。

「もちろんだよ。もともと非機能要求グレードは、実装方式に制約されずに非機能要求項目を

網羅したものだから、そのまま利用できる。モデルシステムという考え方で、社会的影響を踏まえて具体的なレベルも例示されているから、良いリファレンスになるだろう。」

「わかりました。検討してみます。」

竹田部長は、冨田の安心した表情を見て、新規事業を預かる不安が少しだけ薄れたように思った。

解説

このケースの情報処理推進システムズ社は、自社のパッケージソフトウェアを使ったサービス提供型ビジネスを考えています。こうしたビジネスモデルのシフトの場合、マルチテナントへの対応といった技術的な課題とは別に、SI と、サービス提供することの本質的な違いを考えることが重要になります。

パッケージソフトウェアのSI型導入では、ユーザーの要求を明確化する要件定義から構築後の運用・保守まで様々な局面がありますが、アプリケーション機能についてはパッケージをベースとした Fit&Gap 分析として進めるのが一般的です。一方非機能要求については、稼働環境や基盤を握っている情報システム部門などのユーザー主体で行われることも多いようです。どちらの要件についても、現行アプリケーションや現在の基盤といった一応のベースがあるわけですが、したがって、業界の規制やガイドラインに沿ったセキュリティレベルとその運用等の非機能要求、コストと要件間のトレードオフ調整などを一から考えて頭を悩ます機会が少ないのです。ユーザーにヒアリングすることで、一通りの非機能要求を定義できると言ってもいいでしょう。

一方、世の中ではサービス形態の多様化が進み、クラウドサービスの普及も加速しています。それに伴い、従来はオンプレミスで構築されていた様々な業務がサービスとして提供される形に変わりつつあります。今後、サービスを提供する組織が、アプリケーション機能だけでなく、クラウド上で運用全般について一から検討しなければならないケースは、ますます増加していくことでしょう。

そうした場合のひとつの課題が、上記の非機能要求の部分です。これをサービス提供側が SLA として定義し完結させなければならないことになります。『ユーザーにきく』のではなく、『自ら創り出す』ことが求められるわけです。

非機能要求グレードは、もともとはユーザーとベンダーなどステークホルダー間の合意形成ツールとして設計されたものですが、その名のとおり非機能要求を網羅的・体系的に洗い出すためのスキームをもっています。

サービス提供者の SLA を検討するツールとしても活用いただければと思います。



4.9 要員育成に活用する

千石データサービス社では、会計システムを自社サーバーで運用してきたが、導入から5年が経ちハードウェアの保守終了が迫っている。情報システム部の定例会議で岩崎部長が切り出した。

「そろそろ会計システムの更改について検討を始めたい。新システムは吉田君にリーダーを任せたいがやってもらえるかね。」

入社5年目の吉田はこれまで現行システムの保守を担当してきたが、システム企画段階から担当するのは初めての経験となる。

「はい、もちろんやらせてください。基本的に現行システムの機能を前提に検討を進めればいいですね。」

「そうだな。でもここ一年、経理部から期末処理のたびにレスポンスが遅くなるとクレームがきているな。原因はなんだったのかな？」

「それは、去年の千石システム社の統合の影響ですよ。想定以上に取引データ量が増えたためです。」

岩崎部長は考えながら答えた。

「そうか。では次の計画では経営戦略部の中期計画も聞いておく必要があるな。吉田君、関連部門とのキックオフを設定してくれないか。そこで、次期システムに求める要求の確認を始めてほしい。」

吉田は困った顔になった。

「機能についてはたたき台を作れると思います。ただ、データ量やユーザー数の今後の見込みや性能などの確認もあると思うのですが、現行の要件定義書ベースだと不足する項目がありそうです。何をどこまで確認すべきでしょうか。」

岩崎部長はアドバイスした。

「非機能要求グレードは知っているかい？ 今回の検討で利用できるのではないかな？」

「でも、非機能要求グレードって、通信システムとか社会的影響が大きい開発で使うものではないのでしょうか？ 一回ダウンロードしたのですが、項目数がとにかく多くて。」

「目的に合わせて必要な項目を利用すればいいんだよ。特に非機能要求を検討する上で品質やコストに大きな影響を与える項目が重要項目として設定されているから、現行システムで問題になっている性能・拡張性の重要項目から確認してみたらどうかな。」

「重要項目というのがあったなんて、気がつきませんでした。確認してみます。」

1か月後の定例会議で吉田の報告を聞いた岩崎部長は満足げに頷いた。半年後のリリースに向けて、計画は順調のようだ。



「非機能要求グレードについて、会計システムチームで勉強会をやったんです。先輩の前田さんが、可用性要件の検討漏れで、ディスク障害時のリストア時間が問題になった経験なども話してくれて、各項目の重要性が実感できました。」

部内のベテラン技術者のノウハウの継承についても最近心配になっていた岩崎部長は、吉田の話は良いきっかけになると考えた。

「それはよさそうだ。その勉強会を部内でも開催してくれないか。他のチームにも非機能要求に関する経験談を発表してもらい、部として非機能要求グレードの活用を推進していきたい。吉田君、進めてくれるか。」

定例会議後、吉田のまわりには発表案の提案をしようと部員が集まっていた。岩崎部長は安堵して会議室を出た。

解説

非機能要求グレードをいざ自分のプロジェクトに適用しようとしても、どこから手を付けるべきか悩んでしまうことがあります。経営層にも協力して欲しいのですが、非機能要求分析の重要性を経営視点で紹介できる資料はないでしょうか。また、組織的に活用したいが、初心者が使い方を学習できる資料はないでしょうか。このような要望に対応するため、非機能要求グレードには経営層向けの小冊子や研修教材も用意されています。

まず、非機能要求グレードを理解し活用のヒントとしていただくための資料が2つあります。

1つ目は、経営層向けの小冊子「経営に活かす IT 投資の最適化」です。自動車の購入という身近な例を題材に「非機能要求」のイメージをつかんでいただくとともに、IT システムにおいて「非機能要求」を軽視した場合に起こり得る、ビジネスに直結する危険性を認識いただくことを目的としています。

2つ目は利用ガイド〔活用編〕（本資料）です。新規構築、保守・更改、運用など局面に沿って参考となるケースを並べていますので、自分の状況に近いケースを見つけ、活用のヒントにしてください。それぞれの背景は違いますが、ビジネス目標の達成に向けた要求についてステークホルダー間で共通の評価軸として、非機能要求グレードを利用しています。いくつかのケースに目を通していただければ、目標にあわせて非機能要求グレードの項目数を絞り込むなど、カスタマイズして使えることがわかるでしょう。今回の千石データサービス社では、非機能要求分析の経験のない若手社員に紹介したことをきっかけに、非機能要求グレード表の部門内への展開につながりました。

さらに、実際のプロジェクトへの適用の前に、実践的に学習したいのであれば、非機能要求グレード研修教材がおすすめです。演習課題として「既存システムの非機能要求の見直し」や「ステークホルダー間での合意のロールプレイ」など実践的な設問が設定され、非機能要求グレード表を使って非機能要求を決める一連のプロセスを学習できます。本教材には、企業内で研修を実施できるよう講義用スライド一式も含まれておりますので、技術研修の題材としてご利用いただくことも可能です。

様々な観点からの検討が必要となるシステム基盤の非機能要求分析において、非機能要求グレードが便利なツールであると感じていただき、活用につなげていただくことを期待しています。

TIPS 非機能要求のカテゴリ別の考え方

非機能要求を、共通基盤／社内グレード／個別調整項目といった3階層のカテゴリで分けて定義し直し社内標準として利用することができます。基幹ネットワークやデータセンターなど共通的に利用できる設備が既に存在する場合や、モデルシステムに類似した独自のシステム分類定義（重要度ランク等と称されることが多い）が存在する場合に下図のように定義します。

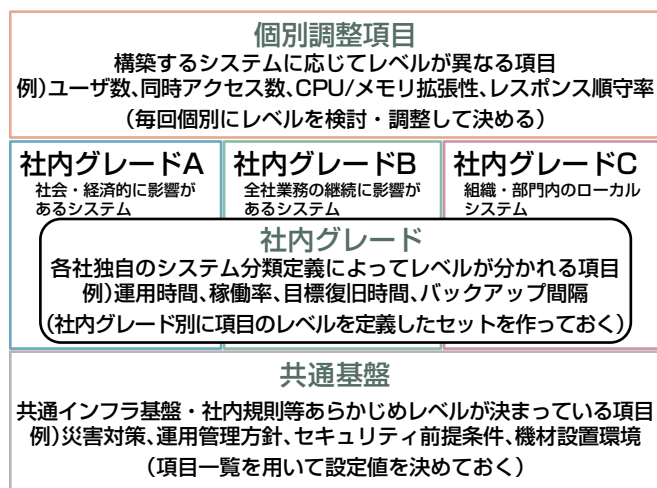


図 項目を3つのカテゴリに分けた場合の階層イメージ
(非機能要求グレード 利用ガイド [利用編] 4章より転載)

さらに、下記はある企業（A社）に「信頼性ランク」と呼ばれる定義がある場合を仮定し、社内グレードを作成した例です。

表 信頼性観点による社内グレードを設定した例
(非機能要求グレード 利用ガイド [利用編] 4章より転載)

大項目	中項目	小項目	メトリクス (指標)	A社 信頼性ランク		
				ランクA (信頼性高)	ランクB (信頼性中)	ランクC (信頼性低)
				選択レベル	選択レベル	選択レベル
可用性	継続性	運用スケジュール	運用時間	5 24時間無停止	4 若干の停止あり (9時～翌朝8時55分)	2 夜間のみ停止 (9時～21時)
		業務継続性	対象業務範囲	4 外部向けオンライン系業務	3 外部向けバッチ系業務	2 内部向け全業務
		目標復旧水準 (業務停止時)	RPO (目標復旧地点)	3 障害発生時点 (日次バックアップ＋アーカイブからの復旧)	2 1営業日前の時点 (日次バックアップからの復旧)	1 5営業日前の時点 (週次バックアップからの復旧)
			RTD (目標復旧時間)	4 2時間以内	2 12時間以内	1 1営業日以内
	耐障害性	稼働率	稼働率	5 99.999%	4 99.99%	2 99.00%
		サーバ	冗長化 (機器)	2 すべてのサーバで冗長化	1 特定のサーバで冗長化	0 非冗長構成
運用・保守性	通常運用	ストレージ	冗長化 (ディスク)	2 多重冗長	1 単一冗長	0 非冗長構成
		バックアップ	データ復旧範囲	2 システム内の全データを復旧	2 システム内の全データを復旧	1 一部の必要なデータのみ復旧
			バックアップ取得間隔	5 同期バックアップ	4 日次で取得	1 システム構成の変更時など、任意のタイミング
		障害時運用	システム異常検知時の対応	5 保守員が常駐	3 保守員到着が、異常検知からユーザの翌営業開始時まで	1 保守員到着が、異常検知から数日中
	運用環境	開発用環境の設置	開発用環境の設置有無	2 運用環境と同一の開発環境を設置する	1 運用環境の一部に限定した開発環境を設置する	0 システムの開発環境を設置しない
		サポート体制	ライフサイクル期間	3 10年以上	2 7年	0 3年
セキュリティ	不正追跡・監視	不正監視対象		2 システム全体	1 重要度が高い資産を扱う範囲、あるいは外接部分	1 重要度が高い資産を扱う範囲、あるいは外接部分

非機能要求グレード 2018

- ・ 8年ぶりの改訂、「非機能要求グレード 2018」を公開
～新たなセキュリティ脅威の増大などを踏まえ、ツール群を改訂～
<https://www.ipa.go.jp/sec/reports/20180425.html>
- ・ 事業案内「システム構築の上流工程強化（非機能要求グレード）」
<https://www.ipa.go.jp/sec/softwareengineering/std/ent03-b.html>
- ・ 非機能要求グレード本体（日本語版）
<https://www.ipa.go.jp/sec/softwareengineering/reports/20100416.html>
- ・ 小冊子「経営に活かす IT 投資の最適化」
<https://www.ipa.go.jp/sec/softwareengineering/reports/20110427.html>
- ・ 各種研修教材
<https://www.ipa.go.jp/sec/softwareengineering/reports/20130311.html>

非機能要求グレード（初版）

- ・ 非機能要求グレード本体（英語版）
<https://www.ipa.go.jp/sec/softwareengineering/reports/20101222.html>
 - ・ 非機能要求グレード本体（中国語版）
<https://www.ipa.go.jp/sec/softwareengineering/reports/20130329.html>
 - ・ 活用事例集
<https://www.ipa.go.jp/sec/softwareengineering/reports/20120424.html>
-

編著者（敬称略）

システム構築上流工程強化部会

「非機能要求グレード改訂 WG」

主査：	小浜 耕己	スミセイ情報システム株式会社
委員：	青山 智彦	株式会社エヌ・ティ・ティ・データ
	大島 正敬	株式会社ソリューション・アンド・テクノロジー
	川口 正高	三菱電機インフォメーションシステムズ株式会社
	川村 冠東	日本電気株式会社
	河野 太基	富士通株式会社
	河野 福司	東京海上日動火災保険株式会社
	斉藤 範彦	第一生命情報システム株式会社
	只野 完二	株式会社日立社会情報サービス
	野間 悠太	スミセイ情報システム株式会社
	早川 剛	株式会社日本取引所グループ
オブザーバ：	飯田 一輝	株式会社エヌ・ティ・ティ・データ
	岩井 誠	富士通株式会社
	酒井 康一	富士通株式会社
	塚本 恭	横河ソリューションサービス株式会社
	成瀬 泰生	ICT インテグレーション ラボ
	松田 拓	富士通株式会社
	柳下 輝男	横河ソリューションサービス株式会社
	吉田 広信	株式会社エヌ・ティ・ティ・データ
	和田 美江子	日本電気株式会社
事務局：	山下 博之	独立行政法人情報処理推進機構
	山本 英明	独立行政法人情報処理推進機構

監修（敬称略）

システム構築上流工程強化部会

主査：	山本 修一郎	国立大学法人名古屋大学
委員：	小野 修一	株式会社エヌ・ティ・ティ・データ
	崎本 壮	株式会社日立製作所
	長山 一	一般社団法人日本情報システム・ユーザー協会
	福田 靖	日本電気株式会社
	細川 泰秀	一般社団法人アドバンスト・ビジネス創造協会
	森崎 修司	国立大学法人名古屋大学
	森田 功	富士通株式会社
	横山 隆介	株式会社日本取引所グループ

非機能要求グレード 2018 利用ガイド [活用編] 第2版

2013 年 3 月 11 日 1 版発行

2018 年 4 月 25 日 2 版発行

2019 年 3 月 28 日 2 版 2 刷発行

監修者 独立行政法人情報処理推進機構 (IPA)
社会基盤センター

発行人 片岡 晃

発行所 独立行政法人情報処理推進機構 (IPA)
〒113-6591
東京都文京区本駒込 2-28-8
文京グリーンコートセンターオフィス
<https://www.ipa.go.jp/ikc/>

© 独立行政法人情報処理推進機構

※本書の図やイラストは、第三者の著作物を利用して作成しています。



古紙パルプ配合率80%再生紙を使用

リサイクル適性(A)
この印刷物は、印刷用の紙へ
リサイクルできます。