

ユーザ視点から機能要求・非機能要求を導出するアプローチに関する検討

尾形 慎哉^{*1} 亀和田 慧太^{*1} 相沢 直人^{*1}

Study on the approach to derive the functional requirements non-functional requirements from a user perspective

Shin-ya Ogata^{*1}, Keita Kamewada^{*1} and Naoto Aizawa^{*1}

Abstract - ISO9126 defined software quality as "quality of use". The industry has become an issue or how to improve quality. To solve these problems, which may point to a string with the user's true requirements of non-functional requirements of the system functional requirements. Therefore, in this study, we consider an approach to derive the functional requirements non-functional requirements from a user perspective, and verified its effectiveness.

Keywords: functional requirements, non-functional requirements, user perspective, user interface

1. はじめに

システムの品質を高めることは、システム開発における大きな課題であり、長年、多くの取り組みが行われてきた[1]。そして、最近の日経コンピュータの調査[2]によると、品質を高めるための改善すべきシステム開発の工程で重要視されていることは、「ユーザ要求の取りまとめ」と「システム要件の定義」であることが明らかになっている。

ユーザ要求の取りまとめという課題に対応するためには、ユーザの真の要求を汲み取り、ユーザ要求事項として明確化する活動が重要である。そのような活動について、近年では人間中心設計の開発アプローチが認知されはじめ、システム開発においても、その方法が導入されつつある[3]。

システム要件の定義については、ビジネス要求とシステムの技術的な制約条件を考慮して、的確な機能要求および非機能要求を特定することがポイントとなる。

機能要求とはシステムが具備しなければならない機能に関する要求のことで、非機能要求とは機能要求以外の性能に関わる要求として、ISO9126においては6つの品質特性と各品質特性を詳細化した副特性から定義されている[4]。

機能要求は、ビジネスゴールの達成に必要な要求に基づいて、必要な機能を割り出していくことで論理的に捉えることがしやすく、技術的な要求や制約条件からシステムでできることが制限されるので、そこから導出される機能要求は特定しやすい。

一方で、非機能要求の定義は、これまで現場担当者の判断に委ねられていたのが実情であった。しかし、日本情報システム・ユーザー協会による非機能要求仕様のガ

イドライン[5]や、非機能要求グレード[6]の発行により、非機能要求についても、導出のアプローチが明確になりつつある状況にある。

しかし、昨今、リリースしたシステムが、結果としてISO9126で定義されている「利用時の品質」に関わる問題によって欠陥としてリコール対応をしなければならないという状況が発生している。特に、システムに具備されている機能を利用したときの違和感が、ユーザにとっては不具合と感じてしまうことが非機能要求の問題として浮かび上がっている。

このような状況については、開発の観点からすれば、機能要求は問題なくとも、システム利用の観点からの非機能要求が満たされていないといえる。その意味では、前述の非機能要求仕様のガイドラインや、非機能要求グレードにおいて、ユーザ視点から定義すべき要求内容を特定する手続きについては明確には記述されていない。

つまり、このような現状を改善するためには、要件定義段階でユーザ視点から導出された要求事項を機能要求および非機能要求へ反映できるようなシステム開発の仕組みが必要である。

それに対して、平沢ら[7]は User Experience を組み込みシステムに実装するための開発プロセス（IPPRUD：Integrated Process with Planning, Requirement, User-interface Design）として、それまでプロセスとして定義されていないかった企画構想およびユーザ要求定義を、システム要件定義の前段階に統合した開発のフレームワークを提案している。このフレームワークに準拠した活動を行うことで、ユーザ視点から機能要求だけでなく、非機能要求も特定することができると考えられる。しかしながら、具体的な活動内容が特定されていないため、開発現場レベルではどのようにこのフレームワークを活用してよいか明確にはなっていない状態となっている。

そこで、本稿では、開発現場レベルで活用できる粒度

*1: 株式会社 U'eyes Design

*1: U'eyes Design Inc.

で、活動内容を詳細化し、ユーザ視点から機能要求だけでなく、非機能要求も導出するアプローチを明確化することを目的として検討を行う。

2. 非機能要求特定の難しさ

2.1 機能要求と非機能要求の定義

ここで、機能要求と非機能要求について説明する。

Robertson[8]は、機能要件（要求）と非機能要件（要求）を次のように説明している。

機能要件

- ・システムの機能性の仕様であり、
- ・システムが取らなければならない行動であり、
- ・システムの基本的な目的から引き出されるものであり、
- ・質ではない。

非機能要件

- ・ユースケースまたは機能要件によって表された作業の性格である。

具体的には下記の要件を非機能要件としている。

- ・ロック・アンド・フィール要件
- ・使用性要件
- ・パフォーマンス要件
- ・運用・操作要件
- ・保守性・可搬性要件
- ・セキュリティ要件
- ・文化的・政治的要件
- ・法的要件

一方で、ISO9126 では下記をソフトウェアの品質特性として定義しており、これらを一般に非機能要求として捉えられている[4]。

- ・機能性
- ・信頼性
- ・使用性
- ・効率性
- ・保守性
- ・移植性

また、情報システムの信頼性向上に関するガイドライン(案)では、利用者の要求を満足するためにソフトウェアが実現しなければならない機能についての要件を機能要件としている。これに対して、機能要件以外のすべての要素（性能、容量、情報セキュリティ、拡張性等）を非機能要件と定義している。

このように、機能要求がシステムに具備しなければならない機能として定義できるのに対して、非機能要求は論者によって定義している特性が異なる。そのため、一意に非機能要求の特性を定義することは現状では難しい。

2.2 機能要求と非機能要求の具体例

ここで、非機能要求の特定の難しさを明示するために、

具体例を用いて確認する。

ここでは、商業施設のバックヤードに設置されているエレベータのユーザインタフェースを例とする。

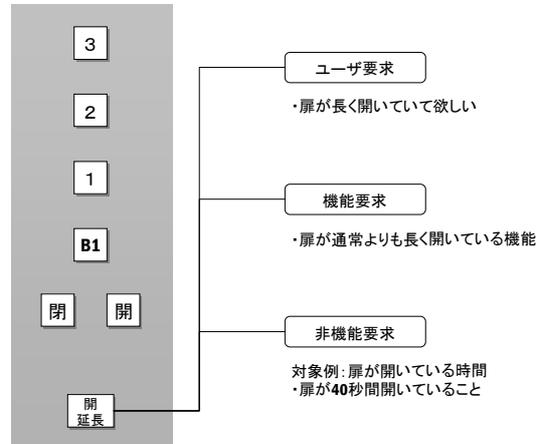


図 1. 機能要求と非機能要求の例

Fig.1. Examples of functional requirements and non-functional requirements

店舗のバックヤードでは、業者が荷物の搬入出を一人で行うことが多く、エレベータの扉がすぐに閉まらないようになって欲しいという要求があったとする（ユーザ要求）。

このユーザ要求を実現するためには、エレベータの扉が通常よりも長く開いている機能、つまり開延長機能が備わっていること（機能要求）が求められ、それが実現されるとユーザ要求が満たされることになる。

しかし、その扉の開いている時間はどの程度が適切か（非機能要求）ということになると具体的な時間まで特定することは難しくなる。それは、開いている時間が短かすぎれば、荷物の搬入出をしている最中に扉が閉まってしまう。一方で、時間が長すぎれば他の階からの呼び出しに対して対応が遅れてしまい、荷物の搬入出業務全体が滞りがちになってしまう恐れがある。

この例から、利用時の品質という観点で問題を考えると、まず、そもそも当該エレベータに、開延長機能が備わっていない場合（つまり、機能要求が満たされていない場合）、頻繁に荷物の搬入出するユーザにとっては、そのエレベータを使えないと感じてしまう。これは、当該ユーザにとっては、有効性に関わる問題となる。

また、開延長機能が備わっていたとしても、扉の延長時間が荷物を搬入出するには短すぎる、または長すぎる場合（つまり、非機能要求がユーザの要求に合致していない場合）も、当該エレベータのユーザにとっては使いにくいと感じてしまう。これは、有効性や効率性の問題と捉えられる。

2.3 非機能要求特定の難しさの要因

前述のエレベータの事例から、利用時の品質の問題が発生する要因について整理すると、機能要求の不備による問題とユーザニーズに対する非機能要求のミスマッチという点から考えられる。

機能要求の不備

この問題は、ユーザ要求に対する機能があるかないかの問題であり、システム要件定義の時に、ユーザ要求事項に対する抜け・漏れがないかでチェックしやすい。

ユーザニーズに対する非機能要求のミスマッチ

どの程度の性能値が許容範囲かは、実際にユーザが利用してみないと判断しにくい。現状は、開発現場担当の経験則や過去事例をそのまま適用してしまい、想定外のことが発生することで大きな問題が発生してしまう。つまり、開発現場担当が、想定外のことを想定しなければならないことに、利用時の品質の問題につながる非機能要求の検討の難しさがある。本稿では、この課題を解決するためのアプローチを検討する。

3. ユーザ視点から機能要求・非機能要求を導出するフレームワークについて

3.1 開発プロセス

平沢ら[7]は、組込みシステム開発において、従来までESPRで定義されていなかった企画構想とユーザ要求定義のプロセスをシステム開発プロセスの前に統合提案した開発方法論の構想を提案した(図2)。この方法論では、企画のコンセプトからユーザ要求事項を特定し、システムの機能要求および非機能要求を導出するフレームワークができています。しかし、詳細な活動定義までは確定していない。詳細な活動を定義することは、検討内容のインプットとアウトプットを明確にすることができ、それ

らがつながることで、ユーザ視点から機能要求および非機能要求を導出するアプローチとしてのトレーサビリティも確保することができる。そこで、本プロセスの特にユーザニーズの獲得に関わるユーザ要求定義プロセスについての活動定義を行う。活動定義にあたっては、前述の非機能要求特定の難しさへの対応として、各プラクティスにおいて、次の活動を強化することを検討した。

・利用状況の明確化

適切な機能要求を導出するために、真のユーザ要求事項を特定する

・要求事項の分析

真のユーザ要求事項から機能要求を特定した上で、その非機能要求としての想定外のことを想定して致命的な問題の発生を防止するためにも、極端なユースケースを想定する。

3.2 利用状況の明確化

利用状況の明確化のプラクティスでは、製品企画書の内容をインプットとして、ユーザの現状の利用状況を把握することが求められる。

そのためには、まずステークホルダを確認し、サービスに関わる人やものの関係性を把握する。現システムが稼働している場合は、実際の現場の利用状況を調査して、実態を把握する。最近では、エスノグラフィや長期的なモニタリング調査を行う傾向がある。もし現システムがない場合は、あるべき利用状況を明文化することになるが、どちらの場合も、利用状況のシナリオを記述することが推奨されている。シナリオとは、アクタ（ユーザ）、

プロセスカテゴリ	企画	ユーザー要求定義	システム開発		ソフトウェア開発	
プロセス	企画構想	ユーザー要求定義	システム要求分析	システムアーキテクチャ設計	ソフトウェア要求分析	ソフトウェアアーキテクチャ設計
プラクティス	企画の構想	利用状況の明確化	システム要求事項の特定	システムアーキテクチャの設計	ソフトウェア要求事項の特定	ソフトウェアアーキテクチャ設計
	企画案の確証	要求事項の分析	システム要求事項の分析	システム要求事項の割当て	ソフトウェア要求事項の分析	インタフェース設計
		ユーザー要求定義	運用環境の分析	システムインタフェースの定義	運用環境の分析	ソフトウェアアーキテクチャの評価・改善
		ユーザー要求定義の確証	システム要求事項の重み付け	システムアーキテクチャの評価・改善	ソフトウェア要求事項の分類と重み付け	
			システム要求事項の評価・改善		ソフトウェア要求事項の評価・改善	
			システム妥当性確認テストのための評価計画作成		ソフトウェア妥当性確認テストのための評価計画作成	

図 2. HCD 開発プロセスモデル

Fig.2. HCD process models

目標、背景情報、イベントの系列などが記述されたもので[9]、その状況が明確に記述することができる。

ここで注意することは、ユーザのタスク（作業）の把握である。表層的なタスクの把握に留まらず、ユーザがあるタスクを行う際に、なぜそのタスクを行っているのか？を洞察することが、ユーザのメンタルモデルを把握することにつながる。

その際のポイントは、タスクとユーザの思考は同時に考えなければならないことである。基本的に、ユーザは何らかの考え、思考に沿ってタスクを行っている。その思考とタスクを同時に記述したユーザ視点フロー[10]は、時間軸に沿って、ユーザのタスクと思考を合わせて記述することで、利用状況を把握する際に、ユーザのメンタルモデルを明文化する基礎情報として役立たせることができる。そのユーザ視点フローの基本的な記述のルールは図3の通りである。



図3.タスクフローの記述フォーマット

Fig.3. Task flow format

思考には、ユーザの期待や不安を記述する。例えば、あるタスクを行うときに「こうだったらいいな」「これがないと不安だな」といった記述である。

それが、ユーザ要求事項の元となり、ユーザ視点から機能要求および非機能要求を特定するための出発点となる。

3.3 要求事項の分析

利用状況が明確化され、ユーザの思考が明らかになったら、ユーザの要求事項を特定する。

ユーザの要求事項は、思考を十分に考慮した上で、その期待に応えるような事項や、不安を払拭するような事項を要求として記述する。

例えば、デジタルカメラの電源を入れる、というタスクを考えると、シャッターチャンスを逃したくないから素早く電源を入れたいというユーザ要求が考えられる。その際、期待としては「どのような場面でもストレスなくシャッターチャンスに対応できる電源 ON のスピードであって欲しい」という要求が見いだされる。

それと同時に、利用上、重要なタスクがあれば、そのタスクに重要度をつけておく。そうすることで、タスクに優先度がつき、優先度の高い要求事項も特定することができる。優先度がない場合、要求が出るごとに、シス

テムに対する機能要求が増えることになるが、利便性を高めるために機能を絞り込むことが難しくなる。

そして、ユーザ要求事項が特定できたら、そのユーザ要求事項を満足できるような機能要求を検討する。前述のデジタルカメラの例だと、「シャッターチャンスに対応できる高速の電源 ON 機能」などが考えられるが、これが機能要求となる。これで、ユーザ視点から機能要求が特定されたことになるが、まだ非機能要求が特定されていない。

非機能要求は、機能要求に対して特定されるが、難しいのは何をどう決めればよいか明確になっていないことである。そこで、下記のような非機能要求を特定するステップを考案した。

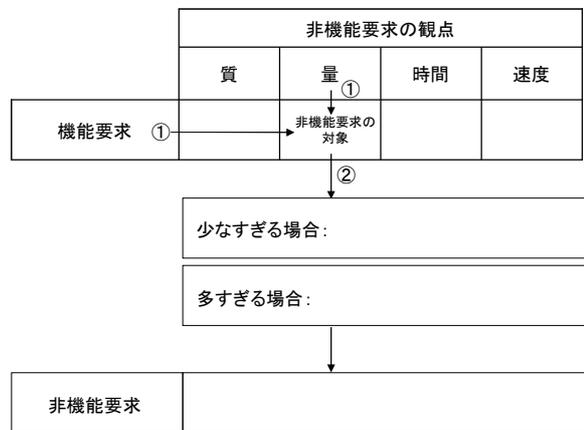


図4. 非機能要求導出のプロセス

Fig.4. Non-functional requirements of the derivation process

ステップ 1: 特定した機能要求に対して、「質」「量」「時間」「速度」の観点で考えられる非機能要求の対象を特定する

当該機能要求における非機能要求を検討する際の観点を与えることで、定義すべき非機能要求の検討漏れを防ぎやすくていいものと考えられる。そこで、ポイントとなるのが、よい観点を与えることである。できる限り広い視野から観点を検討できるようにしたいため、まず粒度の高い観点を設定した。それを、「質」「量」「時間」「速度」とした。

前述の例だと、高速の電源 ON 機能において時間の観点から、「電源 ON までの時間」という非機能要求の対象を見いだすことができる。

ステップ 2: 対象となる非機能要求を見いだしたら、該当する観点から極端な場合を想定した上で、妥当な性能値の範囲を特定する

致命的な非機能要求に対する問題を発見するためにも極端なユースケースを検討する。そこで、「質」「量」「時

間」「速度」の各観点で、両極端な場合を想定する。

前述の例だと、電源 ON までの時間が「短すぎた場合」と「長すぎた場合」を想定する。短すぎて問題となることは想定しがたいが、長すぎた場合はシャッターチャンスに対応できない、というユーザの要求に合致できないユースケースが考えられる。図 5 は、これらの考え方をツールとしてまとめたものである。



図 5. 非機能要求導出ツール

Fig.5. Non-functional requirements derived Tools

3.4 ユーザ要求定義・確認

ユーザ要求定義においては、これまで見いだしてきた機能要求および非機能要求をもとに、プロトタイプを作

成し、評価、修正を行い、利用時の品質に影響する問題を改善する。その際は、ペーパープロトotyping[11]の手法が有効である。

一方で、これまでのどのユーザ要求から導出された機能要求・非機能要求のトレーサビリティ確保という観点から、要求を下図のように一元管理できるようなシートにまとめると管理がしやすくなるものと考えている。

4. 有効性検証

4.1 検証目的

本稿で提案したアプローチにおいて、システム開発担当者による機能要求の特定と、特定が難しい非機能要求を定義するアプローチの有効性を検証するため、次の実験を行った。

4.2 方法

- ・概要：システム開発担当者によるユーザ要求、機能要求および非機能要求の定義
- ・実験対象：メールソフトの送信タスクにおけるユーザ要求、機能要求、非機能要求の特定
- ・実施日時：2011年7月16日
- ・方法：システム開発経験3年程度の2名がペアで、2回に分けてユーザ要求、機能要求、非機能要求を特定
 - 1回目：特にやり方を指示せず実施
 - 2回目：本稿で提案している観点（ツール）を提供し、実施

利用状況シナリオ	タスクフロー	ユーザ要求事項	機能要求	非機能要求
<p>■設定</p> <p>タスク名：メールを送信する</p> <p>ユーザ：PCメール利用者</p> <p>ゴール：確実にメールを相手に送信する</p> <p>■利用状況シナリオ</p>	<p>①メールを立ち上げる</p> <p>・受信メールはあるかな？</p> <p>②受信メールを確認する</p> <p>・迷惑メールはあるかな？</p> <p>③目的の相手にメールを送る</p> <p>・アドレスや件名が間違っていないかな？</p> <p>・ファイルの添付忘れはないかな？</p> <p>・送る相手が多いと宛先入力や確認が面倒だな。</p> <p>・きちんと送信できたのかな？</p>	<p>急ぎの返信が必要な受信メールはすぐに確認したい</p>	<p>送信しなければならぬメールを替えてくれる機能</p>	<p>替えるメールは一回につき20通まで</p>

図 6. 要求記述フォーマット

Fig.6. Request description format



図 7.実験風景

Fig.7. Experimental situation

4.3 結果

本稿で提案したツールを使ったアプローチによる機能要求、非機能要求の数を比較すると、ツールありの方がより多くの機能要求、非機能要求の特定に至った。

表 1. 実験結果

Table 1. Experimental result

	ツールなし	ツールあり
ユーザ要求数	6	19
機能要求数	14	17
非機能要求数	4	11

参加者の主なコメント

- ・(ツールにより) 観点が与えられたので、新たな気づきは得られた
- ・現場で情報を共有するにはよいツールかも知れない
- ・元々考えていたことを記述するだけ、という面もあった

4.4 考察

本稿で提案したアプローチによって、ユーザの要求獲得数は、ツール(ユーザ視点フロー)を利用することで飛躍的に伸びていることが分かった。

さらに、非機能要求においても大きく伸びている。これは、ツールにより非機能要求の観点が与えられたことで考えるべき非機能要求を検討するきっかけが有効に働いたことが影響していると思われる。

また、ツールなしの場合、ユーザ要求数が少ないにも関わらず、機能要求が数多く特定された。これは、機能中心に検討をしていることが現れているものといえる。

つまり、本稿で提案しているアプローチでの使用を考えていたツールとしては、システム開発の現場で利用されることで、下記の点で効果がある傾向が確認された。

- ・ユーザ要求から機能要求を特定する
 - ・機能要求から考えられる非機能要求を幅広く検討する
- 一方で、課題として、機能要求から粒度の高い観点を与えたが、そこからどのような非機能要求を対象として定義すればよいかを具体的に検討する段階においての飛

躍が大きく、難しさを感じるものが課題として明らかとなった。

5. おわりに

本稿では、ユーザ要求から機能要求および非機能要求を導出するアプローチについて検討を行った。そのアプローチに対して使用できるツールを用いた検証を行い、その結果、開発現場でツールを利用することの有効性と課題が確認された。

今後、ユーザ要求から機能要求および非機能要求を導出する連動性を高めることで、より効果的に開発現場で使えるツールを検討していく予定である。

謝辞

本稿作成にあたり貴重なご助言をいただいた小樽商科大学平沢尚毅教授、実験に協力いただいた青山氏、前原氏に深謝いたします。

6. 参考文献

- [1] Myers, Glendord J.; The Art of Software Testing, New York: John Wiley & Sons. (1979)
- [2] 日経コンピュータ, 2011.1.20, p.120 (2011)
- [3] 渡辺, 高田, 小豆島, 河崎: エクスペリエンス指向アプローチによるシステム開発上流工程の取り組み, 日立評論, Vol.92, pp.29-32 (2010)
- [4] ISO/IEC 9126-1:2001 Software engineering
- [5] 経済産業省情報処理振興課, NTT データ経営研究所, 日本情報システム・ユーザー協会: 非機能要求仕様定義ガイドライン, 日本情報システム・ユーザー協会, ISBN978-4-903477-13-8 (2008)
- [6] 独立行政法人情報処理推進機構ソフトウェア・エンジニアリング・センター; 非機能要求グレード, ソフトウェア・エンジニアリング・センター (2010)
- [7] 平沢, 尾形, 鱗原; User Experience を組込みシステムに実装するための開発プロセスに関する提案, 商学討究, Vol.60, No.4, pp73-88 (2010)
- [8] Robertson, Suzanne, James Robertson.; 要件プロセス完全習得法, 三元社, (2002)
- [9] 大西, 郷; 要求工学, 共立出版 (2002)
- [10] 尾形, 鱗原, 青島; UI (ユーザインタフェース) 設計とシステム設計を結ぶモデリング技術の可能性について, ヒューマンインタフェースシンポジウム 2004 論文集抜粋, pp.1073-1076 (2004)
- [11] Carolyn, S., 黒須; ペーパープロトタイプング 最適なユーザインタフェースを効率よくデザインする, オーム社 (2004)