

要求の外部完全性についての考察

山本修一郎

名古屋国際工科専門職大学

愛知県名古屋市中村区名駅 4-27-1

A Consideration on External Requirements Completeness

Shuichiro Yamamoto

IPUT in Nagoya

4-27-1, Meieki, Nakamura-ku, Nagoya Aichi Japan

概要

システム内部で欠陥が発生しないように、機能例外に対応するための機能要求の完全性が内部完全性である。デジタル技術で業務プロセスを再構築する「デジタルイゼーション」では、業務プロセスと整合する要求仕様が必要である。このような要求仕様は、システムの機能要求の内部的な完全性ではなく、システムの運用プロセスからみた要求の完全性が必要である。

本稿では、このような業務プロセスも含めた要求の外部完全性を提案する。また要求の外部完全性を分析できる「欠陥未然防止図」を説明する。欠陥未然防止図では、受付条件、資源条件、判断条件と例外条件に基づいて、アドホックな業務プロセスの発生を抑止し業務の欠陥を未然防止できる。

Abstract

Internal completeness of functional requirements is defined by functional exceptions so that defects do not occur within the system. Digitalization, the reconstruction of business processes using digital technology, requires requirement specifications that are consistent with business processes. Such requirement specifications require completeness of requirements from the perspective of the system's operation process, rather than internal completeness of the system's functional requirements.

This paper proposes external completeness of such requirements. It also explains a "defect prevention diagram" that can analyze the external completeness of requirements, including business processes. A defect prevention diagram can prevent the occurrence of ad-hoc business processes and prevent business defects based on acceptance conditions, resource conditions, decision conditions, and exception conditions.

1. はじめに

システム内部で欠陥が発生しないように、機能例

外に対応するための機能要求の完全性が内部完全性である。これに対して、システムを利用する外部が

らみた要求の完全性が外部完全性である。

以下では、まず 2 章で要求仕様の完全性についての関連研究を説明する。次いで、3 章で要求の外部完全性を提案する。さらに、4 章でビジネスにおける、要求と完全性について議論する。5 章でまとめる。

2. 関連研究

2.1 要求仕様の検査基準

要求仕様の検査基準として Boehm[1]が完全性、一貫性、実現可能性、試験可能性を定義した。
[完全性]すべての要求について要求を構成する属性、機能、制約が記述されていること
[一貫性] 要求記述間に矛盾や関係の抜けがないこと
[実現可能性] 計画された期間と予算内で要件を実現できること
[試験可能性] 外部インタフェース数、交換データ数、可能な状態数が限定できること

2.2 要求仕様の完全性

IEEE 830[2]では、要求仕様が完全であることを、
a)設計制約、属性、外部インタフェース b)すべての状況における有効・無効な入力に対する応答 c)参照・用語の定義としている。Leveson[3]は、要求仕様の完全性を要求のあいまい性に基づいて定義している。すなわち、要求仕様に含まれている情報が、ソフトウェアの望ましい挙動と望ましくない挙動を設計者が判断する上で不十分であるとき、要求仕様はあいまいである。あいまいでない要求仕様は完全である。したがって、システムが動作する状況下で、ソフトウェアの安全な挙動を規定するためには、要求仕様が完全である必要がある。

筆者ら[4]は、アクタ状況完全、イベント応答完全、アクタ間相互作用完全を提案した。アクタ状況完全は、すべてのアクタとその状況が抽出できていることである。イベント応答完全は、すべてのイベントと応答が抽出できていることである。アクタ間相互作用完全は、すべてのアクタ間の相互作用が抽出できていることである。

2.3 User story

User story では、ユーザー(Who)が何(What)をなぜ必要とするか(Why)だけを簡潔に記述する。User Scenario では、User Story を時系列(When)で整理する。これに対して、要求仕様では、ユーザー(Who)が、どの状況で何を契機として(When)、どんな入力に対して、どう処理して、何を出力すると(What)、どんな応答があり、相手のユーザに何が起きるのか(Why)を定義する。したがって、User Story と User Scenario は要求仕様に含まれている。

2.4 ステークホルダ要求

ISO/IEC/IEEE 29148:2011 [5]には、ステークホルダ要求工学、システム要求工学、ソフトウェア要求工学がある。

ステークホルダ要求工学では、ステークホルダの意図、運用概念、システム運用概念に基づいて、ステークホルダの公式要求 StRS を作成する。システム要求工学では、ステークホルダの公式要求に基づいて、システム要求 SyRS を作成し、ソフトウェア要求工学では、システム要求からソフトウェア要求 SRS を作成する。

Carson[6,7]による 3 段階の要求完全性定義における要求インタフェース条件の明確化で、前提条件と応答条件の完全性を確認するためには、StRS に基づく SyRS が必要である。

2.5 自工程完結

生産工程の自工程完結 Ji-Koutei-Kanketsu (JKK) [8]は、特定のプロセスだけでなく、生産プロセス全体を最適化する手法である。JKK を導入するには、業務の流れを定義する業務手順だけでなく、業務要件を定義する要件整理シートを定義する必要がある。要件整理シートは、業務プロセスごとに必要な項目・情報欄、業務入力欄、業務出力欄から構成される。必要事項・情報欄では、製品の品質の条件として入力、ツール、方法、能力・権限、理由を記入する。入力欄には、いつ、どこで、何を受け取るかなどの基準を記述する。出力欄には、どこに、いつまでに、何を生産するかを記述する。基準欄には、

「プロセスの出力が良品である」と判断するための基準を記述する。

JKK の要件整理シートでは、ビジネスプロセス要素ごとに完全性条件を明確にできるので JKK はビジネスプロセスにも適用できる。

筆者が提案した欠陥未然防止図[9,10] は、6 個の頂点を持つ六角形ノードを要素とする図式である。頂点には 6 側面として、入力、出力、受取条件、資源条件、例外条件、判断条件が対応している。受取、入力、資源、および判断の側面は、外部要素からのアウトサイドインの流れを表す。出力および例外の側面は、外部要素へのインサイドアウトの流れを表す。例外と出力を別矢印で分離した点に、欠陥未然防止図の特徴がある。

3. 要求の完全性

3.1 要求の変更要因

要求の内部完全性と外部完全性に基づいて、6 個の要求変更要因[11]を分析した結果、要求の誤りを除いて他の 5 個の要求変更理由（発注者の知識の変化、技術・計画・経費、優先順位変更、システム環境の変化、組織変更）が外部不完全性によることが明らかになった（表 1）。

表 1 要求変更要因の分析

変更要因	説明	完全性
要求の誤り	要求分析、確認段階や設計工程以降の下流工程で検出された要求の誤り・矛盾を訂正する	内部
発注者の知識の変化	要求を詳細化することにより、発注者がシステムに対して何を必要としているかがより明確になるために要求の内容を変更する	外部 StRS
技術・計画・経費	要求を実現するための経費・期間・所要技術の成熟性などの理由で、実現すべき要求の内容が変更される	外部 StRS
優先順位変更	システム開発の途中で、当初想定したビジネス環境が変化することにより、実現すべき要求の内容が変更される	外部 StRS
組織変更	システムの利用組織の変更に伴う業務プロセスの変更に伴い、要求の内容が変更される	外部 StRS
システム環境の変化	システムの運用環境が変更されると、システム要求に依存する要求の内容が変更される	外部 SyRS

ここで、発注者の知識の変化、技術・計画・経費、優先順位変更、組織変更は、StRS の変更である。また、システム環境の変化は、SyRS の変更である。

3.2 ビジネス IT 整合性

グリコでは、システム再構築した結果、物流セン

ターでの出荷データ不整合等が発生し、想定受注に対して処理が、間に合わず出荷の停止を判断した[12,13]。この不具合の解消に 2 か月の時間がかかると想定されている。全社レベルでデータを「見える化」するシステムなので不具合原因を特定して、システムの一部を改修しても、システムの信頼性を保証するためには、システム全体の試験作業が必要だと考えられる。また、食品の生産を制御するシステムなので、生産された食品の安全性を確認することにも必要になる。

IT システムが往々にして失敗しがちなのは、技術的な能力と効率性に過度の重点が置かれ、IT システムを使わなければならない人々に関する問題や IT システムが関連するより幅広い組織的な問題にあまり注意を払っていなかったからである[14]。

システム思考は、人間や人間の新年、価値観と利害といったものに多くの関心を注ぐとともに、技術と組織の構造と戦略との関連をも視野に入れている。そのためシステム思考は、情報システム開発に責任がある人々に対して大きな貢献ができる。

Rittel と Webber[15]は、社会技術的なシステムは、社会環境や技術環境の変化に対応する必要があるから、システムが扱う問題も解も定義している間に変化してしまう厄介な問題であると指摘している。同様に、Lehman と Belady[16]もビジネス環境に埋めこまれる情報システムは常に進化し続ける問題を扱うので、問題を定義することが困難であるだけでなく、最適解としての情報システムを開発することが難しいと指摘している。

ステークホルダ要求仕様 StRS[5]の内容では、「はじめに」で、ビジネスの目的 ビジネスの範囲、ビジネスの概要、ステークホルダを明かにする。次いで、「ビジネスマネジメント要求」で、ビジネス環境 ゴールと目標、ビジネスモデルを記述する。また、「ビジネス運用要求」で、ビジネスプロセス、ビジネス運用ポリシーと規則 ビジネス運用制約、ビジネス運用モード、ビジネス 運用品質、ビジネス構造を記述する。

4. ビジネス、要求と完全性

ビジネスアーキテクチャとビジネス、要求と完全性の関係を図1に示す。



図1 アーキテクチャ、要求、完全性の関係

アーキテクチャには、ビジネスアーキテクチャと情報システムアーキテクチャがある。ビジネスアーキテクチャに対して、ビジネスモデルとビジネスプロセスが対応する。情報システムアーキテクチャに対して、ステークホルダ要求、システム要求、ソフトウェア要求がある。ビジネスアーキテクチャと情報システムアーキテクチャの相互作用に対してビジネスIT境界の整合性を明かにする必要がある。本稿では、要求についての完全性を定義した。

ビジネスモデルとビジネスプロセスに対する完全性についても明かにする必要がある。

4.1 ビジネスモデルの完全性

顧客価値はビジネスモデルによって創造される[17]。これまでに、多くのビジネスモデルが提案されている。ビジネスモデルの要素に抜けがないことがビジネスモデルの完全性であると定義すると、ビジネスモデルごとに完全性を定義できる。

たとえば、アジャイル開発では、顧客価値を生む製品やサービスを迅速に開発する必要がある。しかし、単に早く動くものを作ることがアジャイル開発ではない。顧客価値を生まないサービスを迅速に開発していたのでは、かえって価値を生み出すまでに時間を浪費してしまうことになる。価値を生まない機能を開発する意味がないことから、Minimum Viable Product (MVP)が提案されている。MVPをモデル化するMVP canvasの要素は、顧客、ユーザーシナ

リオ、提供製品 MVP、機能、実行計画、ゴール、仮説検証基準である[18]。このために、MVP Canvasでは、ペルソナを明記して顧客による製品の利用プロセス（ジャーニー、利用シナリオ）を抽出して製品と機能を定義する。また、成果と仮説の検証基準を記述します。この理由はMVPを満たす製品であることを評価するために必要になるからである。また、仮説を定義するだけでは意味がない。仮説の妥当性を判断する基準を、製品を実現する前に定義しておくことが、顧客が望む「正しい製品」を開発するために重要になる。

Business Model Canvas (BMC)[19]の要素は、ビジネスを協働で支えるパートナー、主要ビジネス活動、ビジネスを実現するために組織がもつ主要資産、顧客に提供する価値、顧客関係、ビジネス対象の顧客層、ビジネスを提供するための販売経路、ビジネスを運営する経費構造、ビジネスの収益構造である。これらの要素に抜けがないことがBMCの完全性である。

筆者らによるアクタ状況完全、イベント応答完全、アクタ間相互作用完全は、ビジネスアクタ間の関係についての完全性であることから、ビジネスモデル完全性の一つである。また、ビジネスモデルが生む価値については、個別的に仮説検証基準を設定して確認する必要がある。

4.2 ビジネスプロセスの完全性

ビジネスプロセスの実行段階では、設計段階では見えていない多くの例外が発生する[20]。この理由は業務工程を適切に実行するための条件を明確になっていないために、不完全なビジネスプロセスになっているからである。ビジネスプロセスの不完全性に対応するために、実行段階で個別例外に対応する必要がある。

ビジネスプロセスの完全性については、佐々木による自工程完結が知られている[8]。自工程完結では、業務フロー図と要件整理シートを作成する。要件整理シートでは、工程ごとに、入力と道具方法などからなる良品条件、出力、これで良いと判断できる判断基準を記述する。しかし、自工程完結の要件整理

シートでは、例外を扱っていない。

筆者ら[9,10]は、自工程完結の条件要素を受取条件、資源条件、判断条件として再構成するとともに、例外を工程ごとに定義する欠陥未然防止図を提案している（図2）。欠陥未然防止図では、受取条件、資源条件、判断条件を満たすとき、入力から出力を生成する。ここで、受取条件、資源条件、判断条件のいずれかを満たさないとき、例外が発生する。したがって、受取条件、資源条件、判断条件が成立しないとき、例外条件を識別できる。

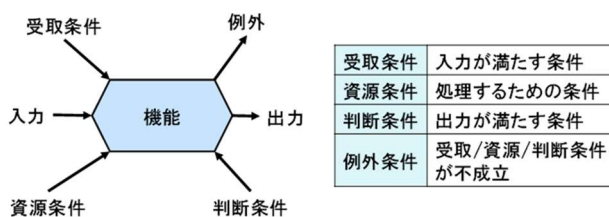


図2 欠陥未然防止図

欠陥未然防止図では、望ましい挙動である出力を機能が生成するための条件を、受取条件、資源条件、判断条件で明確にしている。また、これらの条件が成立しない場合、例外条件によって、望ましくない挙動である例外を生成する。この意味で、欠陥未然防止図は、Levesonの完全性条件を満たしている。

4.3 異なる完全性間の関係

ビジネス組織がITを使用してビジネス目標を達成するプロセスがビジネスIT整合性である。従来から、ITシステムがビジネスと対応できないために、多くのITシステム障害が発生している。

前述したように、Jackson[14]は、ITシステムが失敗しがちなのは、ITシステムを使わなければならない人々に関する問題やITシステムが関連するより幅広い組織的な問題に注意を払っていなかったからだと指摘している。

ステークホルダ要求仕様StRSには、ビジネスマネジメント要求でビジネスモデルを記述するとともに、ビジネス運用要求で、ビジネスプロセスを記述する。また、ユーザ要求では、ユーザ入力に対するシステム出力を定義して、ユーザとシステムの相互作用条件を記述する。さらに、提案システム概念として、

運用概念と運用シナリオを記述する。StRSでは、ビジネスモデルとビジネスプロセスを含めてユーザ要求とともに記述している。このように、StRSでは、実現するビジネスモデルとビジネスプロセスを明記しているから、ビジネスIT整合性を保証する要求記述である。また、StRSではシステム要求SyRSを含めて記述しているから、StRSを達成するSyRSであれば、StRS整合性を持つSyRSであることが保証される。

5. 考察

5.1 新規性

本稿では、要求の外部完全性を提案した。また、要求の変更要因の多くが外部完全性の欠陥によることを明らかにした。

これまで、要求の完全性を確認する手法は提案されているが、要求の外部完全性条件は明確ではなかった。

5.2 適用性

表1では、金融系業務要求の変更要因に本稿で提案する外部完全性を適用することにより、要求変更のほとんどが内部完全性の欠陥ではなく、外部完全性の欠陥であることを判明した。したがって、要求の外部完全性を確認することで要求変更に対応できる可能性がある。

5.3 限界

本稿では要求の外部完全性を提案した。しかし、要求の完全性を確認する具体的な手順を示していない。今後、要求の外部完全性を確認する手法を明かにするとともに、多くの事例に対して適用することにより定量的に有効性を評価する必要がある。

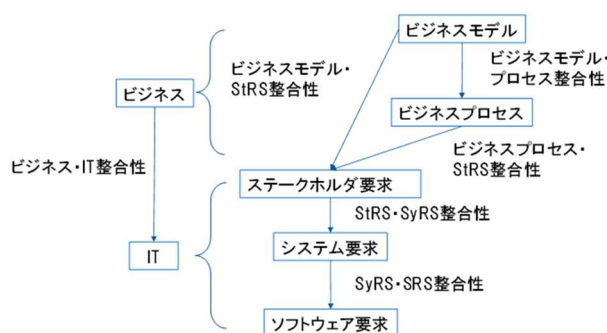
筆者は保証ケースを用いて非機能要求の完全性を確認する方法[21]、ならびに例外に基づく非機能要求の記述法[22]を提案している。今後、要求仕様の外部完全性についての提案手法とこれらの手法との関係を明らかにしていく予定である。

6. おわりに

本稿では、要求の完全性を内部完全性と外部完全性に分類した。内部完全性では、ソフトウェア要求仕様内部の完全性である。外部完全性の対象は、ソフトウェア要求仕様の外部としてのステークホルダ要求、システム要求である。ステークホルダ要求の内部でビジネスモデルやビジネスプロセスを扱う。ビジネス要求とシステム要求についてはビジネス IT 整合性が必要である。したがって、要求完全性の対象として、ビジネスモデル完全性、ビジネスプロセス完全性、ビジネス IT 整合性、外部完全性、内部完全性がある。

欠陥未然防止図では、ビジネスプロセスの例外への対応を補完できるので、例外としての欠陥に対応できるビジネスプロセスを定義できる。

今後、4.3 で指摘したように、これらの完全性について、図 3 に示すような包括的な関係を明らかに



する必要がある。

図3 ビジネス IT 要求整合性

参考文献

- [1] Boehm, B. (1984). Verifying and Validating Software Requirements. IEEE Software, Jan.1984, p. 75-88
- [2] IEEE Recommended Practice for Software Requirements Specifications, IEEE std. 830 -1998
- [3] Nancy Leveson, Safeware- System Safety and Computers, Addison-Wesley, 1995, 松原友夫監訳, セーフウェア, 翔泳社, 2009
- [4] Noboru Hattori, Shuichiro Yamamoto, Tsuneo Ajisaka,

and Tsuyoshi Kitani, Proposal for Requirement Validation Criteria and Method based on Actor Interaction, IEICE TRANSACTIONS on Information and Systems, Vol.E93-D No.4 pp.679-692, 2010.12

[5] ISO/IEC/IEEE 29148:2011, Systems and software engineering -Life cycle processes - Requirements engineering, 5.2.3, 2011

[6] Ronald S. Carson, REQUIREMENTS COMPLETENESS: A DETERMINISTIC APPROACH, Eighth Annual International Symposium of the International Council on Systems Engineering. 1998.

[7] Ronald S. Carson, Erik Aslaksen, Abd-El-Kader Sahraoui, Requirements Completeness, INCOSE International Symposium, 2004

[8] 佐々木眞一, 自工程完結-品質は工程で造り込む, JSQC 選書, 日本品質管理学会, 2014

[9] 山本修一郎, 自工程完結図式の提案, 電子情報通信学会, KBSE 研究会, 5.18, 2024

[10] Shuichiro Yamamoto, Business Process Completeness, eKNOW 2024, 5.27

[11] 服部昇, 山本修一郎, 知識イノベーションを支える IT 要求品質分析, NTT 技術ジャーナル, pp.20-22, 2008.5

[12] めざまし media, 消えたプッチンプリン, <https://mezamashi.media/article/15268806>

[13] 江崎グリコ株式会社, 当社グループにおけるシステム障害について, 2024.4.22

[14] Mike Jackson, Systems Thinking and Information Systems Development, Journal of the Japan Society for Management Information Vol.6 No.3, Dec.1997, pp.5-16

[15] Horst, Rittel, Melvin, Webber, Dilemmas in a General Theory of Planning, Policy Sciences, 4:2, p.155-169, 1973

[16] M.M. Lehman and L.A. Belady, Program Evolution - Process of Software Change, Academic Press, 1985

[17] Henry Chesbrough 著, 大前 恵一 朗 訳, Open Innovation ハーバード流イノベーション戦略のすべて, 産業能率大学出版部, 2002

[18] Paulo Caroli, Lean Inception- How to Align People and Build the Right Product, Editora Caroli, 2018

[19] Alexander Osterwalder & Yves Pigneur, Business Model Generation, Wiley, 2010

[20] McKinsey and Celonis bring the power of process mining to business transformations, <https://www.mckinsey.com/about-us/new-at-mckinsey-blog/>, March 15, 2024

[21] 山本修一郎, 限定自然言語仕様による品質保証手法, KBSE2023-36, Vol.123, No.270, pp.25-30, 2023

[22] 山本修一郎, 例外に基づく非機能要求の記述法, KBSE 研究会, KBSE2023-71 pp. 37-42, 2024