

Stakeholder分析手法の比較報告：CVCA vs. OnionModel

Stakeholders Analysis: CVCA vs. OnionModel

嶋津恵子^{†‡}

Keiko Shimazu^{†‡}

[†] 慶應義塾大学 デジタルメディア・コンテンツ統合研究機構

[‡] 慶應義塾大学 大学院 システムデザイン・マネジメント研究科

[†] Faculty of Commerce, Tokyo International Univ.

[‡] Faculty of Science and Technology, Keio Univ.

要旨

最近の報告によると、情報システム構築プロジェクトの失敗の原因は、Vee モデルで表現されるシステムエンジニアリングの左辺の最上位に位置する要求獲得にあると考えられている。そこで、近年発表が相次いでいるステークホルダ分析手法を試用し、効用を検討した。直感的に扱えるもの(CVCA)と、網羅性高くかつシステムティックにステークホルダを抑えられるもの(OnionModel)を採用した。今回の試みでは、CVCA の導入は容易であったが、重要な(サービス仕様を決定するのに注目すべき)影響を与えるステークホルダを見落とした。一方、OnionModel の採用に際し、利用方法の習得にリードタイムを要した。今後情報システムやソフトウェアの構築プロジェクトにおける十分な実証検証と、改良が必要だと思われる。

1. はじめに

21 世紀に入ると日本では、情報システムの構築や更新に関する重大な問題が相次いだ。2000 年問題への産業界あげての対応、2001 年に相次いだ携帯電話機のソフトウェア不具合による回収騒動、2002 年の第一勧銀、富士銀行、日本興行銀行が合併して出来た「みずほ銀行」のオンラインの大規模障害、最近では 2005 年ジェイコム株の誤発注事件があり、さらに 2007 年にはシステム障害により全日空の国内線のチェックイン端末が使用できなくなった。これらの騒動を通じ、日本の各産業界では、ソフトウェア品質の重要さが認識されるようになった。それにも関わらず、情報システム構築やソフトウェア開発のプロジェクトにおける具体的な解決策は提示されず、現在も依然として人海戦術で、情報システムをリリースする直前に可能な限りバグ(不具合箇所)をみつけ修復するという方法でしか、品質維持対策への対応が採られていない。しかもそれらの多くが、計画段階で見積もれなかった、“手戻り”(Incidental Development)作業だと言われている[1, 2]。[3]によると、ソフトウェアの開発に際し平均 280 件の不具合が、テストと実装段階でほとんど発見されている。一方、それら不具合がどの工程で発生したかを見ると、実装段階が 47.3%、要求分析から設計までの上流工程でも 40.5%発生している。つまりテスト工程では、実装時の不具合だけでなく、上流工程で発生した不具合も解決することが求められている。仮に上流工程で作成された仕様が正確であれば、テスト工程では仕様どおりに修正する作業に集中できる。ところが仕様そのものに誤りがあった場合、これを精査しその上で情報システム全体を見直すことになる。最悪の場合は、ほぼすべての実装結果が無駄になることもある。できるだけ既実装部分を活かそうとするとアドホック的なパッチファイルの量産になる。問題が複数みつかり、これらの作業の複雑さはさらに増す。これは、言わば上流工程で発生した問題をすべて下流工程で帳尻を合わそうとする“無理”であり、このような方法での要求仕様の正規化の精度は、期待できるはずがない。この“無理”は、保守・運用のフェーズにまで増員・コスト増加を引き起こしている[4]。要求分析フェーズに情報システム構築プロジェクト上の問題があるという見方は、(社)日本情報システム・ユーザ会が毎年行っている企業 IT 動向調査の結果にも表れている。2002 年以降の毎年の調査で、要求仕様の作成の不十分さを挙げる声が多い。この要求仕様の不十分さは、同報告に挙げられている別の問題点“システム規模が大きくなるに従い予算、工期および品質に対する満足度が低下する”にも直結している。さらに 1995 年の調査発表で IT プロジェクトに関わる者たちに大きな衝撃を与えた ChaosReport の 2000 年版には、実装した機能の活用状況が報告されている。何とか完成させた情報システムやサービスであっても、平均 45%の機能がまった

く利用されておらず常に利用する機能は 7%にしか過ぎない。これらのことから、情報システムやサービス構築プロジェクトにおいて、成功の鍵は要求分析の正確さに大きく依存すると言える。

一方、(情報システムだけでなく広く)システムエンジニアリングは、異なる分野の専門的アプローチを組み合わせ、全体最適化を実現することで完成度の高いシステムを構築することを目指す[5]。[5]では、システムエンジニアリング上での全体最適化を実現するためのフレームワークとして Vee モデルによるマネジメントを推奨している(図1)。Vee モデル上の左辺上の(任意の)作業工程と、水平線上に位置する右辺上の作業工程間で最適化を計ることで、システム全体のそれを実現するものである。このフレームワークは、1990年代から導入されているが、特に最近では上流工程の最適化が求められ、顧客分析・特定(Stakeholder Analysis)の手法提案が注目されている[6]。今回、直感的導入が可能な CVCA と大規模システムエンジニアリングで実績のある OnionModel に注目し、それぞれの特徴を調べた。

本書は次の構成を採る。2章に検証に用いたプロジェクトとその成果目標である情報システムについて概説する。3章と4章にそれぞれ CVCA と Onion Model を利用したステークホルダ分析の例を示す。5章に、考察とまとめを述べる。

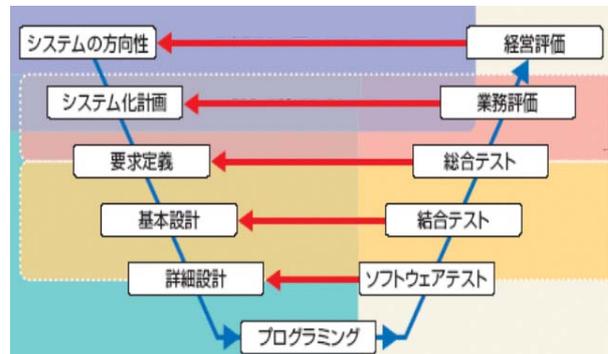


図1 システムエンジニアリングにおける Vee モデル
IPA のホームページから転載

2. 検証対象情報システム概要

今回ステークホルダ分析の調査用に、慶應義塾のネットワークを横断して統合的に学術情報を検索するシステムを構築するプロジェクトを採用した。最も権威がある世界の大学ランキングといわれる“2007 TIMES-QS World University Ranking”(source: QS Quacquarelli Symonds)が、2007年11月8日に発表された。このランキングは通称 THES-QS と呼ばれているが、イギリス TIMES (The Times Higher Education Supplement)社が QS(Quacquarelli Symonds)社と共同で、2004年から毎年実施している。2007年の発表では、ソースデータとしてオランダ Elsevier 社の「Scopus」を採用したことが話題になった。THES-QS は04年以降ソースデータとして、米国 Thomson Scientific 社のデータを利用してきた。初めて採用された Elsevier 社の「Scopus」は、ScienceDirect と称され、研究成果へのアクセス用のシステム改善と、それらの利用者動向の調査結果を盛り込んでいることが特徴の一つだった。我々は、THES-QS の2007年度のランキングの上位10校に注目し、大学のネットワーク(Intra-University)上に公開されている学術コンテンツに対する、利用者への公開サービスが、どういう方法で実現されているか確認した。多くの大学で、学術的に価値のあるコンテンツを専門分野ごとにアーカイブを整備し、大学のインターネット・ホームページからアクセスできるようになっている。また、検索サービスにより、キーワードを指定もしくは専門領域を選択することで対象のアーカイブを特定する。反面、アーカイブ同士を横断した学術コンテンツ検索や、学外のアーカイブシステムを利用・連動・統合した検索サービスの例は存在しない。一方、こういったサービスは、エンタープライズサーチシステムにおいてはすでに標準機能として考えられている。

慶應義塾も、学術的に価値あるコンテンツを集積したアーカイブサイトや eLearning サイトの整備が充実されている一方で、サイト間を横断して学術コンテンツを参照するサービスは存在しない。そこで我々は、これを実現するシステムの構築を目的とするプロジェクトを立ち上げ、実施した[7]。

3. Stakeholder Analysis: CVCA

CVCA (Customer Value Chain Analysis)は、プロジェクトの目標である開発対象のシステム製品や、実現するサービスに対する真の顧客の特定するためのステークホルダ間の関係の表現方法である[8]。一方 ER(Entity- Relationship)モデルは、情報システム構築において、バイブルとして用いられてきた[9]。この

モデルはデータベースにデータとして搭載する問題領域に対し、概念設計や業務分析するために用いられる。ERモデルが実体とそれらの関係だけで対象領域を表現しているのに対し、CVCAは、実体の中の特に”人”に注目して、ステークホルダを楕円やシンボルを用いて表現する[10]。さらに人物以外の実体は、流通するモノ (Object) としてとらえ、ステークホルダ間の関係を示す矢印線上に個別のアイコンで表記する。関係を持つステークホルダ間で製品、代金、クレーム情報が引き渡されるのが直感的に把握できる。CVCAは、ERの単純明快さを継承しつつ価値の連鎖 (Value Chain) を視覚化することに成功している。

前章で示したプロジェクトにおける開発対象システムのステークホルダ群を、CVCAを用いて一覧したのが右の図2である。大元のスポンサーは、学外に存在するが、資金をプロジェクトに配分しているのが、所属研究組織内の事務室であることと、業務指示とプロジェクトの成果を評価するのが同じく所属研究組織内の運営委員会(Executive)であることから、この両者を significant stakeholders とした。要求仕様の特定に際し、この両者に対する聞き取り調査を中心におこなった結果、(Spec1)各コンテンツに対するメタデータ自動付与機能を含めた semantic Web への対応と、(Spec2)情報技術に不慣れな研究者用のデジタルコンテンツアップロードサービスの充実に焦点が置かれた。

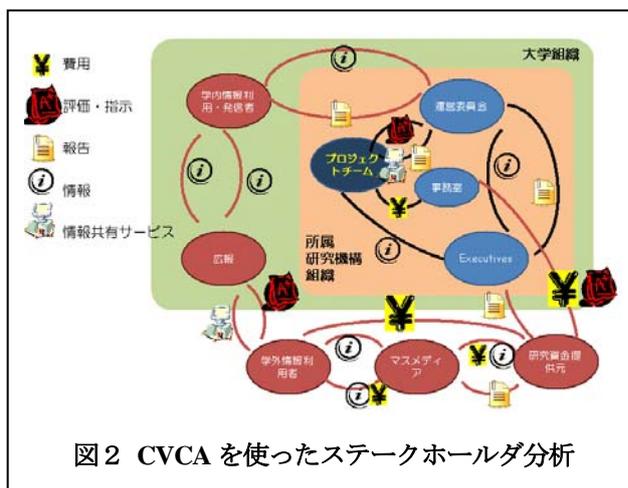


図2 CVCAを使ったステークホルダ分析

4. Stakeholder Analysis: Onion Model

[11]は、システム開発における上流工程で、真のユーザを特定することの重要性を主張している。一般にシステム(何らかの立場で)に関わる人物を“ユーザ”と表現することが、誤った要求を仕様化することに繋がるとし、システムの企画、設計、開発、テスト、運用(maintenance および operation)のすべての工程にかかわる「人」を stakeholder とみなす。そしてそれらを[11]が提案するモデル図上に展開することで、stakeholder 間の関係性(成果物に対する立場と他の stakeholder への影響力)を把握し、成果物(システム)の仕様に反映する最優先意見(viewpoint)を特定することが容易になっている。CVCAを作成する際、システムに関係する人物や組織を並べ挙げるのに対し、OnionModelの作成では、雛型が用意され、それに該当する人物や組織を特定し埋めていく。配置用のマップは、同心円で描かれ、中心に開発するソフトウェア/ハードウェアプロダクトをおく。中心円から外側に向かい、構築するシステム全体、直接接続されている他のシステム、環境として影響を与えるもの(社会システム)を表現する。また各ステークホルダ雛型は、Slot と呼ばれ、ステークホルダとしてのシステムに対する役割(Role)別に用意されている。それぞれのステークホルダ間の影響力(pressure)を矢印で示すことで、仕様に反映させる重要要求を発見する。

2章で示したプロジェクトにおける開発対象システムのステークホルダ群を、OnionModelを用いて一覧したのが右に示した図3である。分析(図の解釈)から、①利用者は、最先端の情報アクセス

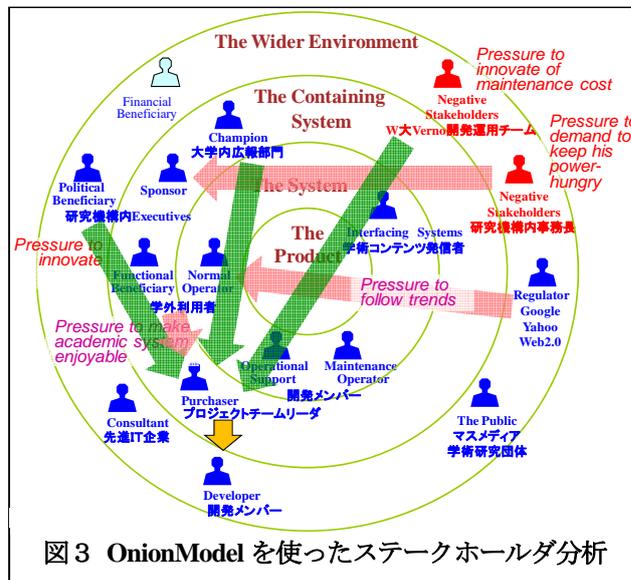


図3 OnionModelを使ったステークホルダ分析

環境の影響を強く受けており、そこでの利用形態や制約を標準（当たり前品質）であると認識している、②sponsor 兼 political beneficial(研究組織内 executive)は、(なんらかの)先進性を求め、また③competitor である Negative stakeholder が大きな課題として取り上げた maintenance cost の削減の実現が求められていると結論付けた。特に①に関し、(①-1)直感的に操作方法を理解できるユーザーインターフェース であること、(①-2)24 時間 365 時間稼働し、アクセス元(場所や組織)を限定しないこと、その一方で(①-3)知的好奇心を刺激する面白さを提供する(単純なキーワード検索だけでなく)新検索サービスを付加することだと結論付けた。

5. 考察とまとめ

今回の検証では、CVCA を用いたときには見落とされた Regulator が、OnionModel を用いたときには、明記された。さらにステークホルダ間の影響力を書き加えると、この Regulator が他のステークホルダに大きく影響を与えていると確認された。これにより CVCA による分析を行って設計した情報システムと、OnionModel による分析を行って設計した情報システムとで、実装仕様が異なった。特に CVCA では、重要視されたコンテンツのアップロード機能と、Semantic Web への対応は、Onion Model で分析すると必ずしも必要なものとしては認識されなかった。さらに所属研究組織内事務長の存在に至っては、CVCA では significant stakeholder と認識されたが、Onion Model 作成時に用意されている Role 分類[12] に当てはめた結果 Onion Model では Negative stakeholder であると確認された(図 4)。

一方、モデルの作成に関し(両者とも専用ツールは使用せず)、要した手間にも差が発生した。CVCA は、作成ルールが単純であるため、特に学習の時間を必要とせず(プロジェクトメンバーとの打ち合わせを含めても)、2~3 時間で完成する。一方、Onion Model は、オブジェクト指向を基盤にしていることもあり、クラス概念を理解しておくことや、雛型として用意されているステークホルダの役割を学習しておくことが求められる。筆者の実験では、事前の学習に 1 日以上を要した。

両モデリングともシステム化対象領域に対するセマンティカルな解釈を前提としており、ER モデリングが実用性を高めたように、正規化の方針が必要であろう。

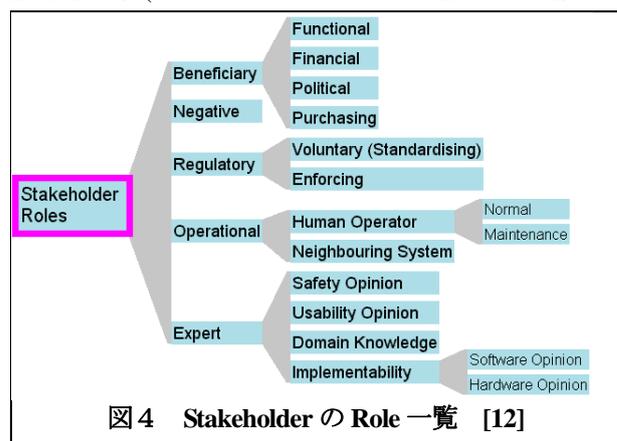


図 4 Stakeholder の Role 一覧 [12]

参考文献

- [1] 渡部, 「手戻りゼロ」で開発コスト半減に挑戦, 日経コンピュータ, 09 月 28 日号, pp.118-134, 1998
- [2] (社)日本情報システム・ユーザ協会, 企業 IT 動向調査報告書, 2006 年 4 月 5 日, 2006
- [3] 経済産業省, 2005 年版組み込みソフトウェア産業実態調査報告書, 2005
- [4] 中村, 矢口, 2003 年情報化実態調査, 日経コンピュータ, 2003 年 11 月 17 日号, pp. 50-63, 2003
- [5] Hanskins, C., Forsberg, K. and Kruger M., Systems Engineering Handbook, ver 3.1, August 2007, INCOSE-TP-2003-002-03.1, 2007
- [6] Kessler, C. and Sweitzer J., Outside-in Software Development: A Practical Approach to Building Successful Stakeholder-based Products, IBM press, 2007
- [7] 嶋津, “知の流通と再編を実現する攻めのコンテンツマネジメント”, IDG Japan 主催 Contents Management Forum 2006, 基調講演, 2006
- [8] Ishii, K. Course Materials, Design for Manufacturability (ME317) .Stanford University. USA, 2003.
- [9] Chen, P., The Entity-Relationship Model--Toward a Unified View of Data, ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976, pp. 9-13, 1976
- [10] 京屋, 野口, 中野, 東芝レビュー, Vol.60 No.1, pp.1-36, 2005
- [11] Alexander, F. Ian., A taxonomy of Stakeholder, Human Roles in System Development, International Journal of Technology and Homan Interaction, Vol1, 1, pp.23-59, 2005
- [12] Alexander, F. Ian., 10 small steps to Better Requirements, IEEE software, Mar/Apr 2006