Stakeholder分析手法の比較報告：CVCA vs. OnionModel
Trail experiments of 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
† The Research Institute for Digital Media and Content, Keio Univ.
‡ Graduate School of System Design and Management, 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，システムエンジニアリングのプロセスの上流工程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ー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マティックにステークホー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ールダを見落とした．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エアの開発プロジェクトに両手法を適用し，実践的な観点からより深い検証を行いたい．

1. はじめに

ソフトウエア開発における上流工程，特に要求分析の問題が指摘されて久しい．日本の経済産業省が 2005 年に発行した組み込みソフトウエア産業実態調査報告書は注目に値する[1]．1 つの組み込みソフトウエアの開発に際し，平均すると 280 件の不具合があるとの結果であった．特に，これら不具合の発見された工程と，不具合を発生させた工程の調査が興味深い．不具合のほとんどは，テストと実装段階で発見されている．一方，それら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． ChaosReport が 1995 年に報告した IT プロジェクトの失敗率の高さは，産業界に衝撃を与えた．同誌の 2000 年版では，実装した機能の活用状況の調査結果が報告された．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．

一般に，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．問題が複数みつかると，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[2]．この”無理”は，保守・運用のフェーズに対する増員・コスト増加まで引き起こしている[3]．今や，ソフトウエア開発において，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．

こういった状況を受け，最近では要求獲得の精度と質をあげることを目指し，stakeholder 分析手法が提案されている．2008 年の Jolt award [1]の General books 部門の finalist にノミネートされた[4]は，ソフトウエアの企画から開発，運用，評価までの全行程のエンジニアリングをシステマティックに体系化した解説書である．この本では，一つの章を割いて stakeholder 分析を取り扱っている．

今回, 直感的に利用可能な stakeholder 分析方法 CVCA[5]と大規模システムエンジニアリングで実績のある Onion Model[6]に注目し, 同一の情報システム開発の上流工程に利用し, それぞれの特徴を調べた.

[1] http://www.joltawards.com/

本書は次の構成を採る． 2 章に Onion Model を，3 章に CVCA を略説する． 4 章にそれぞれの手法を検証用に用いた情報システム利用した例を示す． 5 章に考察を記し，6 章にまとめを述べる．

2. Stakeholders Analysis: Onion Model

[4]は, stakeholderを“the people who affect the success of your (software) product, and are affected by it"とし，さらに“Sometimes what is in the best interest of one stakeholder may not be in the best interest of another stakeholder...”と説明している．そしてstakeholderを，insider stakeholders，principal stakeholders，end-user stakeholders，partner stakeholdersの4つに分類し，これらを抜けもれなく特定することが，機能や性能仕様の決定に利用するべき重要な情報を浮かび上がらせることに直接つながると主張する．さらにこれを実現する具体的な手法として，[6]の提案するOnion Modelを挙げている．

[6]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．一般によくつかわれる用語“ユーザ”は，何らかの立場でシステムを操作する人をさすが，これにより誤った要求を仕様化することに繋がるとしている．[6]は，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 とみなしている．それらを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)を特定することが可能になっている．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(3 章参照)，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エア/ハードウエ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．また各ステークホールダ雛型は，Slot と呼ばれ，ステークホー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ールダ間の影響力(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．

3. Stakeholders Analysis: CVCA

[5]は，要求獲得工程においては、顧客の生の声（VoC：Voice of customers）をそのまま顧客要求として捉えるのではなく、真の顧客要求を把握するための構造化アプローチが重要であると主張する。具合的には，提供するシステムを（操作する）利用者だけでなく，幅広く関係者を想定するためにシステムを対象とするユースケースを数多く挙げ，そのケースごとに出現する人とそれらの関係を明示化することを提案している[7]. この作業を直感的に行えるようにしたのが CVCA（Customer Value Chain Analysis）であり，システムによる価値連鎖の把握と顧客の特定を可能にしている． CVCA では，製品に関係するすべてのステークホールダを楕円で表現し，ステークホールダ間の“お金”の流れや“クレーム（要求)”の流れを矢印で表現することで，価値の連鎖（バリューチェーン）を可視化する。この結果を分析することにより，どのステークホールダが製品開発を成功させる上で重要な顧客であるかを把握し，VoC の収集対象を絞ることができる。現状の販売経路の問題点や，見落としていた重要なステークホールダを発見できる場合もある。実用事例は，機械製品electrocardiogram (EKG) machineの開発，組み込みソフトウエア製品pacemaker alert systemの開発，NGO提供サービスdonor-funded micro-irrigation pumpの開発と幅広い[8].

4. Stakeholder analysis 手法の検証

今回ステークホー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． 慶應義塾には，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た[9]．そのシステムエンジニアリングおける上流工程の Stakeholder 分析で，

CVCA と OnionModel を利用し，要求定義に直接影響する重要な情報の獲得結果にどのような差が発生するかを調べた．

4.1 Stakeholder Analysis: CVCA

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 1 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．

4.2 Stakeholder Analysis: Onion Model

前節と同じ領域を対象に，OnionModel を用いて stakeholder を一覧したのが図 2 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（当たり前の品質）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，利用者のアクセス元(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．

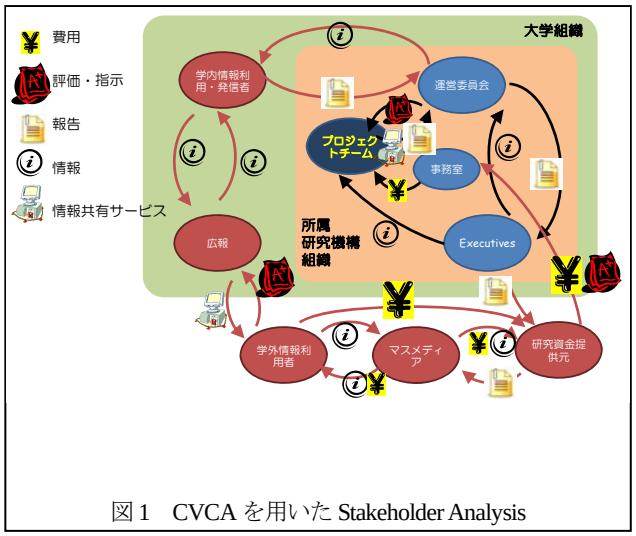

図 1　CVCA を用いた Stakeholder Analysis

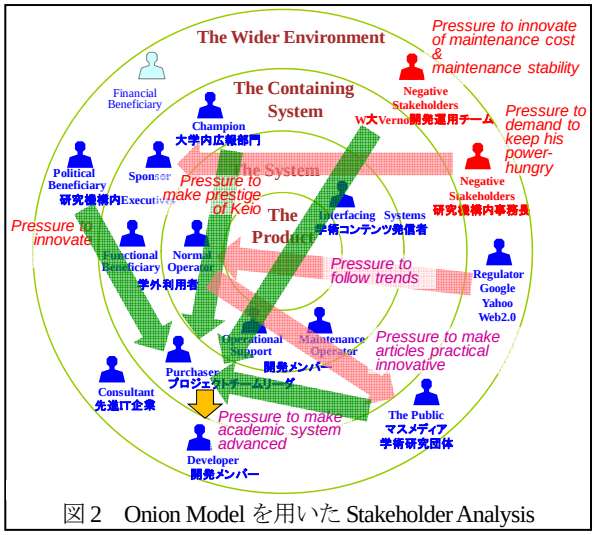

図 2　Onion Model を用いた Stakeholder Analysis

5. 考察とまとめ

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．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[10]．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[11]．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．これにより，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．

一方，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．さらにステークホールダ間の影響力を書き加えると，この Regulator が他のステークホー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．特に OnionModel を使った stakeholder 分析では，Competitor 等に代表される negative stakeholder も網羅的に把握できることが，要求獲得に貢献した．例えば，今回の事例では，他大学で同様のシステムを開発し，昨年 3 月に停止された事実から，この大学を stakeholder にリストした. Onion Model 作成時に用意されている Role 分類(図 3 参照)に viewpoint を記入し，他の stakeholder との関係を議論した結果，これが negative stakeholder であり，かつ significant stakeholder であることが認識された．これにより，OnionModel を採用し作成された要件定義に基づいたシステムは，運用上の安定性を特徴とする仕様になった．

また，これらの stakeholder analysis に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，クラスの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ールダの役割を学習してくことが求められる．筆者の実験では，事前の学習に 1 日以上をかけた．

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標準化が図られていない．ER や DFD(data flow diagram)[13]の両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．

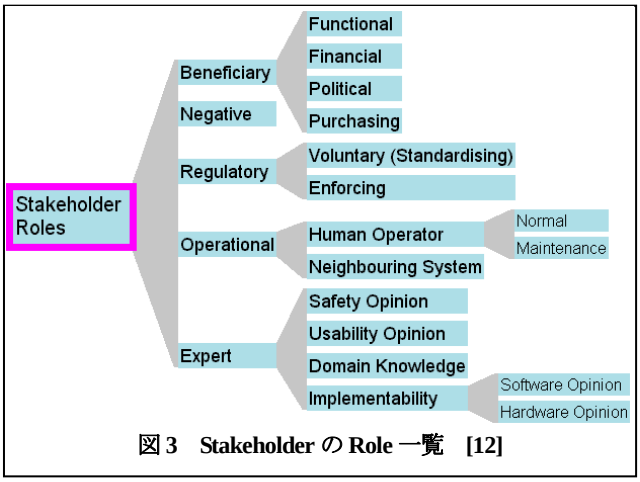

図 3　Stakeholder の Role 一覧　[12]

参考文献

[1] 経済産業省，2005 年版組み込みソフトウエア産業実態調査報告書，2005
[2] 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
[3] (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
[4] Kessler, C. and Sweitzer J., Outside-in Software Development: A Practical Approach to Building Successful Stakeholder-based Products, IBM press, 2007
[5] Ishii, K. Course Materials, Design for Manufacturability（ME317）.Stanford University. USA, 2003
[6] Alexander, F. Ian., 10 small steps to Better Requirements, IEEE software, Mar/Apr 2006
[7] Beiter K., Yang, T. and Ishii, K, Preliminary Design of Amorphous Products, Procs.DVD of the ASME Design Engineering Technical Conference,Paper #9968, 2006,
[8] Donaldson, K. M., Ishii K. and Sheppard, S. D., Customer Value Chain Analysis, Research in Engineering Design, Volume 16, Number 4 , pp. 174-183, 2006
[9] 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
[10]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
[11]京屋，野口，中野，東芝レビュー，　Vol.60 No.1, pp.1-36，　2005
[12] 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
[13]Adler, M., An algebra for data flow diagram process decomposition, Software Engineering, IEEE Transactions onVolume 14，Issue 2，pp.169 – 183, 1988