論文

「越境」の観点から見たシステム開発のユーザニーズをめぐる現場の実践

渡部　宣弥　　竹中　慧　　中村　雅子

システム開発・web デザインにおけるユーザのニーズを把握し，よりよいシステムデザインを行うため，「越境」の観点から現場のシステムエンジニア（以下 SE）や SE を含む会議体が実践している現場の工夫を探索的に調査・分析した．SE を中心とするシステム開発関係者へのインタビューを行い，その結果，要求生成のプロセスにおける SE の工夫や web デザインにおける柔軟なニーズ対応などが明らかになった．ここからシステム開発において，参加者自身が意識的に越境という観点を持つことの必要性が示唆された．

キーワード：情報システムデザイン，越境，システムエンジニア，ユーザ，デザイナ，要件定義

1　問題意識

現在，様々な社会的組織が，活動の効率化などを目的として情報システムの導入を行っている．しかし，情報システムを導入することにより，情報システムを扱う人々やその活動に支障をきたすブレイクダウン［ウィノグラートら］がしばしばみられる．ブレイクダウンは，単に情報システムの技術的な問題だけにとどまらず，そのシステムを実際に使う人々の活動への配慮や理解の不足により引き起こされることも多い．結果として情報システムが期待通りに機能しない，それどころか，情報システム導入前の活動に支障をきたしてしまう場合もある．

典型的な例として［ウィノグラードら］は，工場への自動プランニングシステムの導入事例を紹介している．実際の工場の作業には機械のトラブルやオペレーター人数の変動など，多くの作業状況を構成する要素がある．しかし，このシステムは，実践の中に埋め込まれている様々な要素を考慮せず，自動的に作業プランを生成するものであったためブレイクダウンが生じてしまったとされる．

これまでにもシステムのブレイクダウンに対しては様々な対策が試みられているが，実際には最近になってもしばしばブレイクダウンが引き起こされており，解決方法にはさらなる探究の余地があると考えられる．

我々はこのブレイクダウン問題を扱うにあたり，情報システム開発の中でも，とくに問題が生じやすいとされる上流工程，中でも要件定義に着目した．要件定義とは，ユーザ側の要望をシステムに取り入れるための要件化を行う工程である．

研究をすすめるにあたって，この問題を扱う中で「越境」の概念が新しい切り口になるのではないかと考えた．越境とは，異なる立場や専門性をもつ者同士がコミュニケーションを行う現場において，両者の間やその内外に存在する境界を越え，「良い方向へ［原田・日根・南部・須藤］」変化する活動を意味する．組織内外に存在する境界は，所与のものではなく，人々の認識によって存在したり，しなかったりする場合がある可変的なものだと考えられている．

この越境という観点を用いて，システム開発現場における現状と課題，それらを解決するためのシステムエンジニア（以下 SE）たちの振る舞いの分析を試みる．

2　先行研究

2．1　要求工学の問題点と状況論的アプローチ

これまで多くのシステム開発プロジェクトでは，ウォーターフォール・モデルが利用されてきた．ウォーターフォール・モデルは，滝の水が上から下に流れるようにプロセスが進行し，直前の作業フェーズの成果物を次のフェーズに利用するものである．開発工程の例としては「要求分析」「外部設計」「内部設計」「プログラミング」「テスト」「運用・保守」という流れが一般的である［大場ら］．

しかし，ウォーターフォール・モデルは，要件定義フェーズで対応の必要のあるユーザ・ニーズ（要求）すべてを洗い出さなければならない．要件定義フェーズで成

WATANABE Yoshiya
東京都市大学メディア情報学部社会メディア学科 2017 年度卒業生
TAKENAKA Satoru
東京都市大学メディア情報学部社会メディア学科 2017 年度卒業生
NAKAMURA Masako
東京都市大学メディア情報学部社会メディア学科教授

果物に含まれるべき要求が，下流工程で要件漏れとして発見された場合，手戻りによる再作業が発生し，多額のコストと時間のロスが発生してしまう．このような問題は実際には頻発しており，1980 年代前半からウォーターフォール・モデルの課題として指摘されていた［Gladden］［McCracken, Daniel & Jackson］.

近年は，ウォーターフォール・モデルに代わって，アジャイル型のモデルが注目されるようになってきている．アジャイル型は，ウォーターフォール・モデルの要求分析からプログラミング，テストまでの作業工程をプロトタイプの形で短期間に反復的に繰り返すことで，荒削りの状態から精緻化された状態へと段階的に完成に近づける手法である．これにより，ウォーターフォール・モデルで問題となっていた顧客要求の確認漏れや変更を，計画的に吸収することができる．

［独立行政法人情報通信研究機構］の調査によると，アメリカ等では，情報システムを実際に使用する部署とそれを開発するシステム開発の部署が同一組織に属している状況で開発が行われることが多いという．その場合，開発に企業間契約を介さないため，ソフトウェアの変更に対応しやすい．これはアジャイル型開発の「開発前の要求の固定を前提としない」という特徴に合致し，アジャイル型開発が普及した要因とされている．

一方日本では，IT 技術者の多くが IT 企業で働いており，他企業のシステム開発を請け負う形で開発が行われる．そのためいったん仕様が確定して契約がかわされた後に変更が生じても対応が難しいことが多い．そのためアジャイル型開発が普及せず，依然としてウォーターフォール・モデルを用いるプロジェクトが多くなっていると考えられる．しかし実際には仕様が確定して契約がかわされた後に新たな要求が発生することが多く，時にはそれが開発の失敗につながる（注 1）.

ユーザの要求の「抽出」は大きな課題であり，1990 年代から「要求工学」という視点が生まれてきた．要求工学ではさまざまな手法でユーザの要求を把握する工夫を生み出している．例えばシナリオ設計の手法などもその 1 例である［Carroll］.

しかし要求工学は暗黙の前提として，ユーザの要求が固定的で所与のものであると捉えている．この前提は，ウォーターフォール・モデルにおいて，一度に全ての要求を引き出す（引き出せる）というプロセスモデルにも合致している.

これに対して［中村・上野］は，状況論的アプローチに立脚して，システムの導入は，組織あるいは活動そのものを再編するものであると捉え，このような“いかにユーザの要求を引き出すか”という視点では情報システムデザインとその導入を理解できないことを事例を挙げて示した．［松嶋］でも，現場の実践が規範的な開発プロセスとは大きく異なっていることを指摘している.

これらの議論では，組織や活動などの社会的な要素と情報システムなどの非人間的な要素を，別々に考えるのではなく，一体のものとしてデザインする必要性が論じられている．実際の情報システムデザインには，SE やプログラマと，システムを用いる人々・コミュニティとの同盟関係のあり方が埋め込まれている．デザイナー（SE など）とユーザを明確に区分けせず，いわば境界を超えて，1 つのネットワークとしてデザイン主体と捉えることが必要である.

こうした観点から，アクターネットワーク論［Latour］をベースとしたデザイン主体の生成［田丸・上野］やネットワーク指向のデザイン・アプローチ［上野・永田・ソーヤー］が必要であることが指摘されている.

ユーザと IT 技術者が参加し，循環的に要件定義を行うアジャイル型開発は，システムの導入を組織や活動そのものを再編することに繋がると考えるデザイン発想とも親和的である.

本研究においても，情報システムのデザイン（開発）の現場でユーザのニーズそのものが変容・生成していくという状況論的アプローチの観点から要件定義の実践について分析し，こうしたアプローチがブレイクダウンを防ぐシステムデザインのあり方を考える手がかりになるかどうかを検討する.

2．2　越境的な対話や学習

情報システム開発は，大きく分けるとユーザとデザイナという 2 つのグループの間の境界を越えて，システムをデザインしていくという，越境的な活動として捉えることができる．システムデザインにおける越境の事例として，［原田・日根・南部・須藤］では，病院に導入された情報システムについて実施された障害カンファレンスと呼ばれる会議を分析した事例が述べられている．この障害カンファレンスは，大規模病院に導入された電子カルテの稼働後に報告されたシステム障害について，対応していくための病院側と開発・運用を担っている IT 企業側のメンバーが参加する会議である.

この研究において，障害カンファレンスには一部の医師，病院内 SE，システム開発企業側の SE が会議に参加していた．この会議の中で，本来ならば別の境界である，IT 企業の開発側 SE と病院側の出席者の間の組織的境界と，実ユーザである医師と病院側の代表者（一部の医師・病院内 SE）との間にある境界とが，開発側 SE からは「1 つの境界」として認識されることにより，議論のすれ違いが起きたとされる．一方，会議に参加している医師や病院内 SE が代理性を持っているのと同じように，開発側 SE もまた会議の場にいない IT 企業側の代理人である側面もあることが示されている．会議に参

加しているSEは，障害カンファレンスで得た情報をもとに，実際にシステムを開発しカスタマイズしている開発部隊（プログラマ）に対応を要請する役割だが，両者の間にもしばしば境界が存在する．

この事例はシステム導入後に出てきた問題に対して行われた会議を対象にしたものだが，情報システムの新たな開発・導入においても，このような越境をめぐる諸問題が起こりうるのではないかと考えた．

3　目的

本研究では，システム開発の当事者，主にSEへの探索的なインタビューを通じて，要件定義を中心とするコミュニケーションの課題，および現場のSE，またはSEを含んだデザイン主体が実践の場で行っている課題への対処，工夫を状況論的アプローチの観点から分析することを目的とする．

4　方法

IT企業において，業務系，ネットワーク系，web系などのシステム開発に従事しているSE等を対象にインタビュー調査を行った．主に機縁法を用いて調査協力を依頼し，個人インタビュー，グループインタビューを併用して非構造化インタビューによる探索的調査を実施した．

主なインタビュー内容は，要件定義についての一般的な参考書と現場での要件定義の手法の違い，実際に要件定義において苦労した事例やその場合の対処方法，ユーザからの要望をシステムに反映するための工夫や，ユーザとのコミュニケーションについての企業における担当者研修の有無，内容などである．合わせて6件，12名にインタビューを行った．調査の概要は表1の通りである．

なお，今回の取材の範囲では，依頼元が他企業（他組織），親会社―子会社関係，あるいは同一企業内のいずれの場合でも，基本的にユーザ自身がシステムを開発するのではなく，ユーザとシステムを開発するSE側は異なる組織・部署であった．

5　調査結果

限られた取材範囲からではあるが，ある程度共通して得られた知見を以下にまとめた．

5．1　要求生成のプロセス

業務システムでは，ユーザ側がシステムの要求事項を提案依頼書などの形で示すが，多くの場合開発側はその要求が十分でないと認識している．そのためデザイナ側が能動的なアプローチをすることで新たな要求を生み出していくための現場の工夫が見られた．ここではその工夫についてヒアリングから得たいくつかの特徴をまとめてみる．

(1) 人的リソースの活用

デザイナ側にも様々な専門性を持つエンジニアがいる．例えばユーザ側と直接，対話を行うフロント・システムエンジニア（フロントSE）や，フロントSEが聞き取った要求をもとに実際にプログラミングしてシステムを作り上げていくバックエンドSE，プログラマ等がいる．ユーザ側のシステムに対する要求が曖昧だった

表1　調査の概要

取材番号	取材日	担当業務内容等	取材時間
1	2017年7月13日	Aさん：経営システム部長 （情報システム部門）男性	1時間29分
2	2017年7月21日	Bさん：SE（設計、開発）男性 Cさん：SE（上流工程：フロントSE）男性 Dさん：SE（プログラマ）男性 Eさん：関連企業　営業　男性	1時間38分
3	2017年7月27日	Fさん：SE(ネットワーク系）男性	1時間25分
4	2017年10月14日	Gさん：SE(web系）女性	1時間35分
5	2017年10月30日	Hさん：SE（ネットワーク）男性	1時間35分
6	2017年10月31日	Iさん：SE（主任）男性 Jさん：SE（上流工程）男性 Kさん：SE（上流工程）男性 Lさん：SE（上流工程）男性	3時間5分

場合に，システムを実際に作る担当者（バックエンド SE 等）に要件定義の場に同行してもらい，直接会議に参加してもらうことで，ユーザ側，デザイナ側双方に具体的なシステムイメージを理解しやすくするという試みが行われていた例があった．

その他にも会議に誰を参加させるか，という点で人的リソースを動員する方法が複数の対象者から語られた．システムのハードウェアなど，別なパートを担当する他社に協力要請をし，他社の SE に非公式に参加してもらうという事例も見られた．

[開発において本来は要件定義により工数をかけなくてはならないが，結局作りながら考えるアジャイル的な開発になる傾向があることに言及して]
お客さん自体がふわっとしてて，何をやりたいのか紐解いていくところで実際にダイレクトに作る人からお話させたりですね・・・
（取材 2　フロント SE の C さんのインタビューより）
※［　］内は著者らによる説明補足，以下同じ

(2) アジャイル的手法・プロトタイプの作成

主に web 系において，アイデアをひとまずプロトタイプという形でユーザ側に提示することにより，要求した内容が実際にどういう形になったのかを視覚的に確認できるようにしていた．これによりユーザ側が新たな（またはより詳細な）要求生成がしやすくなる．当初の要求を固定的なものとして考えずに可変的なものとして扱う現場の工夫と見ることができる．

具体的なデザイン案を PowerPoint ベースでちょっと作ってみて, どうですかって言って紙にプリントして．で, 具体的になんか私たちがどうですかって出したものを赤入れしてもらうっていう方法．どっちかっていうとデザイン案はもう当社は何とか用意するけどっていう感じで，赤入れ形式は確かに早いな．
（取材 4　web 系 SE の G さんのインタビューより）

契約上はウォーターフォール型でも，実質的にアジャイル型の手法を取っている場合もある．ユーザ側とデザイナ側の企業が単発の契約でなく，長期的な取引関係にある場合（親会社ー子会社関係など）にはこのような方法が取りやすい．

[営業支援のシステムを開発する場合に，営業スタイル自身が変化していく可能性があるために]
何年もかかってから作りましたって［言っても，その間に］営業スタイルまた変わっちゃうじゃないですか．リーンスタートアップって最近よくやるんですけど，要件を曖昧なまま，わかるところだけ作っていく，ということをしています．そうするとそこだけ効果が早く出て来るんですよね．
（取材 1　経営システム部長の A さんのインタビューより）

(3) SE 自身の経験の活用

SE としての経験を基に要求生成の補助を行っている事例が多く見られた．ユーザの要求に漏れや問題がある場合，SE としての類似の事例の経験で気づいた点を伝えることで，新たな要求の生成や修正に繋げている．一度定義した要件を再定義することになるが，後で手戻りやブレイクダウンが起こらないことを重視して，適切な要求を生み出すことにつなげている．

[教育系企業の授業・実習ネットワークシステムの構築において，作業中に問題点に気づき，提案したら要件が増えたという話題]
後からこっちも気づくこともあるんですよね，やってて．こうなっちゃったときどうするんだろうっていうのに後から気づいたりすると，本当は要件として固まっちゃって定義されて始めてるけど，気づいちゃって言うと，あ，いいですね．それ入れてくださいって言われちゃうので，言わなければいい［のだが，それでは後日問題が生じるのであえて伝える］
（事例 5　ネットワーク系 SE の H さんのインタビューより）

5．2　代理性の認識と越境の工夫

組織の境界を超えて協力関係を築き，共有する目標（よりよいシステムの開発）の達成につなげていくことが越境である．越境を生み出すための障害の一つとして，相手がその場にいない組織・グループ等の代表（代弁者・代理）として関与しているにも関わらずそれを見落としてしまうという問題がある．これを「代理性問題」と呼ぶこととする．

このことを明確に意識しないコミュニケーションは，会議の場では越境が生まれたように見えても，持ち帰った後で問題が生じる可能性が高い．

取材で明らかになったのは，このような問題はデザイナであるフロント SE によく理解されている場合が多く，そのため現場の実践としてさまざまな工夫が行われているということである．それらの工夫が，プロジェクトを「良い方向へ」変化させることへ繋がっている．

デザイナ側であるプログラマとユーザ側担当者は使う用語も予備知識も異なるので，コミュニケーションが難しい．そのためにフロント SE と呼ばれる上流工程に携わる SE が間に入り，コミュニケーションを行う．つ

まり，このフロントSE自身がプログラマの代理人であるという側面も持っている．

開発チームとお客さんっていうので，話してるとやっぱ，言葉通じないんですよ．まず．そこにこういう，我々でいうフロントみたいな人を真ん中に立てて，で，上手いことやり取りする．もう，なんか違う言葉喋ってるみたいな感じですよね．いきなり行ったら．
（取材６　フロントSEのJさんのインタビューより）

要求生成に時間がかかることとの関連で明示的にユーザ側にも代理性があることへの認識が語られている．

［開発するシステムの利用者は全国にいるため］
各拠点の人に聞いて，どこがどういうふうに使いにくいんだとか，どこがどう変えたいんだっていうのを色々吸い上げた上で，やるんで，その辺でもう要件をお客さん自身がまとめるっていうのも少し時間が掛かってたりしてます．実際に使ってる人ではないっていうのがちょっと，まあ，ミソというか．
（取材６　フロントSEのJさんのインタビューより）

また下記の事例は，ユーザの行う業務全般について詳しい専門家の参加がシステム開発に不可欠であることから，フロントSEがユーザ顧客側の中でも業務に詳しい人に要件定義の会議に参加してもらうよう働きかけるという語りである．「一番忙しい人」とは，つまり業務の中心にいて，もっとも業務に知悉している人を意味する．

いや一番忙しい人に担当してもらわないと，しかもその人ちょっと業務から抜くぐらいのことやらないとダメです．
（事例２　フロントSEのCさんのインタビューより）

会議参加者の実際のユーザとの代理性以外に，組織の意思決定において参加者の所属部門の中での決定権が上司にあり，担当者の上司を説得しなければならないという場合もある．このような事例では，説得材料となる資料をフロントSEが代わりに作成して提供することもある．

それでシステムはこう変わります，こう改善されます．システムとしてはここはどれくらいの効果が認められますっていう資料をですね作ってあげるんです．作ってあげて，彼らはそれをもとに上司を説得する．
（取材２　フロントSEのCさんのインタビューより）

SEがインフォーマルに相手企業の実ユーザとコンタクトを取り，コミュニケーションを図った例もあった．ただし，このような回路はあくまで非公式であり，そこで要望を聞いたからと言ってそれが契約上の仕様に反映することはない．しかしあえてそのような機会を作ることは，文字通りユーザの「顔が見える」状況を作り出し，より良いシステムへのデザイナのモチベーションを高めるなどの効果があると述べられている．

J：これは我々のモチベーションの話かもしれないですけど．やっぱり実際にこうやって使ってくれる人がいるのだから，っていうので，やる気は出てきましたね．誰だかよくわかんないっていうよりは，やっぱり見えたほうが，作ってるほうとしてもやっぱり楽しいというか，そういう気持ちはありますね．
筆者の一人：そういうの大きいですよね．すごく．
J：「あの人はあんなこと言うかもしれないからちゃんと作っとこ」［と考える］
筆者の一人：顔を思い浮かべながら．
K：そうです（笑）
（取材６　フロントSEのKさんのインタビューより）

5．3　制度的な要因による課題

官公庁系のシステム開発の場合，開発前に入札がある．官公庁側から資料提供仕様書（RFI）が公示され，提案依頼書（RFP）が作成される．その後各社が提案書を作成し入札を行う．提案依頼書の作成後，国会で実行予算として計上されるまで２年ほど掛かる（取材６より）．

官公庁では職員の定期異動があり，システム開発部門でも担当者が２年程度で交代する．つまり入札や発注の段階とその後の開発のフェーズで担当者が変わってしまい，それまでの議論が引き継がれていないという構造的な問題が指摘された．

加えてこれも構造的な問題として，システム開発に関わる受注の不正を防ぐため，要件定義と実際の開発は同じ企業が受注できないというルールがある．

これらの問題によって，官公庁系のシステム開発には他の事例と比較しても深刻な組織的・コミュニケーション的な境界問題が生じている．

J：（前略）官公庁系って職員さんが毎年人事異動とかで変わるんで，あまりその辺が共有されてないんで，また毎年毎年おんなじことの繰り返しに，ちょっとなりがちなところはあるかなっていうのは．それはちょっと官庁系の特徴なのかな．
（中略）
K：それが今Jが言ったように，お客さんが大体２年ごとに部署が異動しちゃうんです．

（取材 6　フロント SE の J, K さんのインタビューより）

これらの問題に対処するためにデザイナ側の企業は開発部分の落札後に，事実上，再び要件定義をやり直すという手法を用いている（基本設計フェーズに含ませる）．一方で，このような対応のために要件定義を行うために使える期間が極端に短くなる問題が生まれる．

K：まずは，要件定義を出し切るというか．で，その中で予算にハマる内容で，もう一回もう組み直すんですよ．
筆者の一人：落札した後にある意味では，もう一回要件定義し直すという感じで．それは言い過ぎかもしれないですが．
K:いやほとんどそうですよ．なのに期間が, 異様に短い．（笑）
（取材 6　フロント SE の K さんのインタビューより）

5．4　開発中に生まれるニーズへの対応の違い

システム開発においては，後から出てきたニーズに関して柔軟に対応しにくく，別作業に仕分けるなどの対処法も見受けられた．下記の事例では予算と期間を踏まえて対応しきれない要求について，保守・運用作業に回すことが語られている．

[開発時に対応できない部分を運用に回す方法]
やっぱり予算とか期間の関係で今回ここまでしかできないので，こういうケースは「ちょっとすみません．運用でカバーしてください」っていうのはあらかじめ言うこともあったりはします．で，まあそのあとで実際に機能開発とかして，まあ本対策をしていくっていう流れ．
（取材 5　ネットワーク系 SE の H さんのインタビューより）

ユーザ側だけでなく，SE 側でも既に定義した要件の不足部分に気づくこともある．このように SE 側からも，後からシステムに対する要求を生み出し修正を図っている．

作ってる間に疑問が湧いたり，「あれ？ちょっと違うよね」っていうのがあるんですよ，やっぱり，いくら詰めても．そういうときは週に 2 回くらい 2 時間程度集まって，お互い進捗の進み具合とかを話し合ったり，あと業務要件側も世の中変わってきてるんで，当時はこう言ったけどちょっと変えてほしいっていうのはここで吸収しています．　（取材 1　A さんのインタビューより）

一方，要件漏れなどで後から出てくるニーズに対してトラブルにならないように SE 側で対策をしたり，要求をコントロールしようという試みが見られる．ヒアリングでは次のようなデータが得られた．

[要件漏れがあった際の対応について]
K：それって，どっちの責任なのって．IT 企業側がちゃんと聞き出さなかったからじゃないのっていうのは，よく喧嘩になるわけです．お客さんが要件を伝えるのが正しいんですけど，伝えてもらえなかったものなんで，しょうがないっていうのが，なかなかそこはやっぱり難しいですね．そういうときも結局，要件として伝えられてないからできてないんですって一方的に言うと，それはそれでまた［問題に］なるんで．
筆者の一人：次の仕事に差し支えるということですね．
K：やっぱりそこは，「だったらこういう案があるんじゃないですか」って指し示した上で，乗り切ってくってのが［求められる］．だから，要件って聞き出すんですけど，どうしても漏れたとき．なので J もさっき言ってたんですけど，やることだけ書いてると漏れてるのが何なのかっていうのがちょっと難しくて．
筆者の一人：でも漏れてるってのはその時，気づいてないので，多分「できないリスト」にも書けないですよね．
K：なんかグレーかなって思うところを全部もうやらないって書いて．でも結構割りと「これ言われるんじゃないかな」って意外とあると思うんで，それ書いてるだけで，そこから，「またこれできないんですか」みたいな話になってくるんで．そこんところスコープを定めていくというか．
（取材 6　フロント SE の K さんのインタビューより）

要件漏れがあった際に代替案を出して対応するということや，当初より「やらない項目のリスト」として対応できない部分を明文化している．結果として業務系システム開発では，後から出てきたニーズに対し即座に対応していくのではなく，違う工程への移行や代替案などで対応し，場合によっては追加の要求が必要以上に出ないようにコントロールすることで対応しているケースも多く見受けられた．

web 系開発，とくにサイトデザインにおいては，追加の要望が出た場合でも，過去の事例やテンプレートを転用して対応しやすく，できないということは起こりにくいとのことだった．一方，業務システム系においては作業にかかる時間や納期の関係上，違う工程に回す場合などもあり，後から出てきたニーズへの対応の柔軟性や即応性の違いが顕著に見られた．

[web 系業務において，あらたにサイトコピー禁止のコ

マンドを組み込んでほしいという要望が出された事例で］まず過去のお客さん先の事例で似たような件が無いか調べて，ないなと思ったら調べて（実際には前例が）あったので，それをもちろんHTMLに入れていけるなって．で，コピー禁止にしました．これでiPhone長押しで画像コピーできないようにしました．でも「スクショ［スクリーンショットを取られることを防止するのは］無理です．お願いします」みたいな感じで．割とね，できないってことはあまりない．

（取材４　web系SEのGさんのインタビューより）

6　考察

6．1　越境の実践について

今回の調査について越境がどのようにみられたかを考察する．

先行研究で紹介したのは，病院内の医師が利用する，レセプトシステムの事例である．実際には，その導入後に行われた病院の医師が利用するレセプトシステムについての障害カンファレンスでは，会議の場の出席者以外にも多数の関係者がいる．図１でみると，会議の場のユーザ側（病院側），デザイナ側（IT企業）の２つのグループ間の境界Aだけでなく，その場では見えない関係者と会議の場の参加者の間にもある境界（境界B）を認識し，越えていく活動が真の越境といえるだろう．

図２は，境界に関わる課題が特に多いと考えられた官公庁系のシステム開発の図を同様に描いたものである．，ガイドライン上，本来は開発段階では要件定義という工程は存在しないが，上述のように要件定義と担当企業が異なり，さらに官公庁側の異動があるため，基本設計フェーズの枠のなかで要件定義（の再構成）にあたる作業を行っている．

その際，公式には図２のようにデザイナ側企業のフ

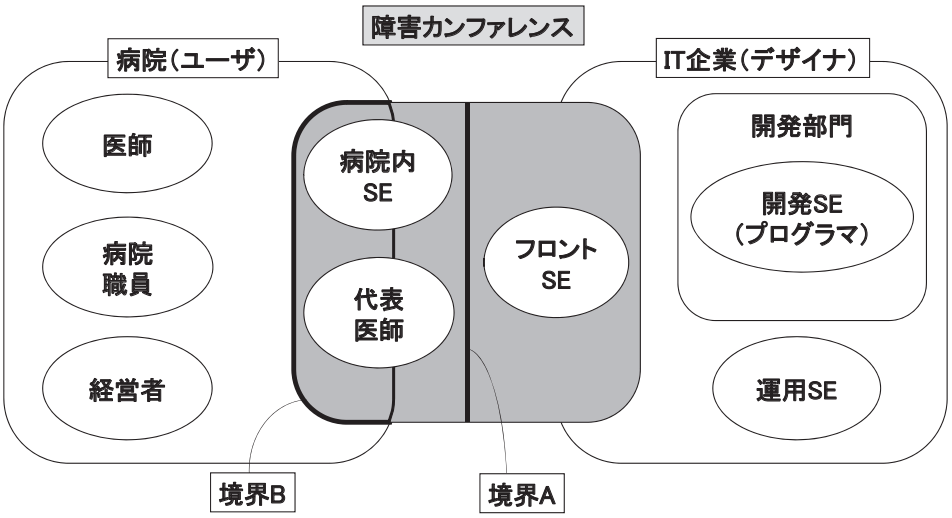

図１　障害カンファレンスの関係者の組織・コミュニケーション構造
（［原田ら］から筆者らが作成）

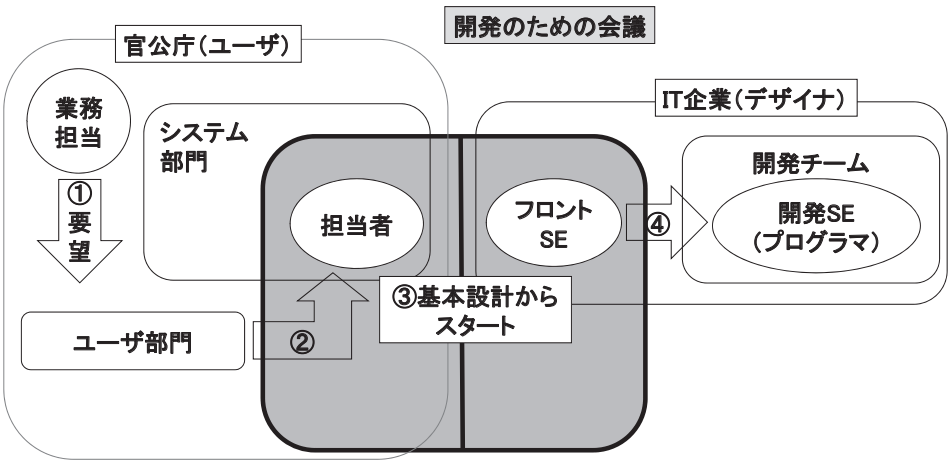

図２　官公庁システムの事例における関係者の組織・コミュニケーション構造（規範的構図）
（筆者らが作成）

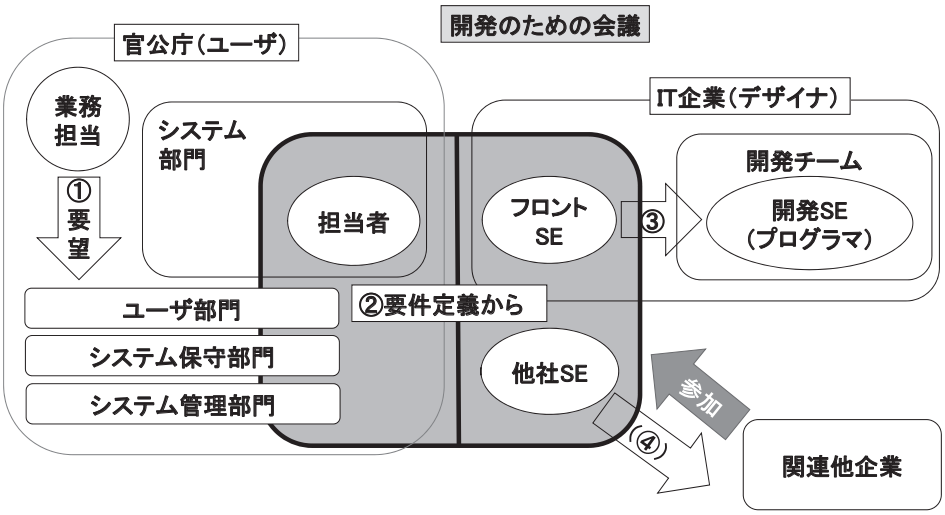

図 3　同事例における実践（越境）の状況
（筆者らが作成）

ロント SE とユーザ側である官公庁のシステム部門の担当者が会議の場に集まる．

規範的なプロセスとしては，①官公庁側のユーザ部門がその部門の構成員から要望を受け取りまとめる．②ユーザ部門がシステム部門に対して要望を送る　③システム部門がシステム仕様の一部として要件定義の場に要望を持ち込む．このため会議の場では基本設計のフェーズから開始する．④完成した仕様に沿ってバックエンド SE が実際の開発を行う．

しかし今回の取材によれば，実際には越境的な活動として，デザイナ側が図 3 のようにさまざまな働きかけをユーザ側である官公庁に対して行っていた．

会議の場には実際には，より実ユーザに近いユーザ部門や，関連するユーザ側のシステム保守，システム管理などの部門の担当者，さらに受注した開発に関連するハードウェア系の開発を担当する他企業の関係者にも参加要請をすることがある．

このような現場の工夫によって，よりスムーズな開発が試みられているのである．これらはデザイナ側からの非公式な要請になるが，実際にこのような形で会議が行われているのは，官公庁側の理解があってこそであり，双方が協力して越境による「より良い」システム開発を目指しているといえるだろう．

一方，これらの工夫のために「会議室不足」という課題が派生していることも指摘された．このような工夫の結果として，30 名以上の大規模な会議になることも少なくなく，官公庁内の限られた会議室の取り合いになることもある．そのため各部署からの参加人数に制限を設けている状況だという．

6. 2　現場の実践

調査の中でシステム開発が抱える多くの社会的な制約が明らかになった．特に官公庁系のシステムでは，国会で決められた予算やガイドラインで制定された開発フロー，ユーザ側の担当者の異動，納期の厳格な設定など，本来，状況的に生成されていくはずの要求や，それに伴う計画の変更が許容されにくい構造が存在することが明らかになった．

一方，SE は実際のシステム開発において，目の前にいる担当者だけでなく，背後に様々なステークホルダーが介在し，お互いの組織内でも一枚岩ではなく，多くの重層的な境界が存在することを認識して行動していた．それらを踏まえた上で，越境を試みるための実践的な工夫，いわば現場の知恵をもってシステム開発を行っている．

その中でも web デザイン系の開発では比較的ユーザとデザイナの敷居が低く，また当初から変更可能性を織り込んだアジャイル型に近いデザインが行われていた．これはシステム自体の構造の複雑さや関与するステークホルダーの規模の違いによるところも大きいだろう．

社会的，制度的制約やシステム特性などの要因でシステム開発のあり方が大きく異なるため，一概に決まった手法を提案することは難しいが，ユーザ側も含めた関係者の多くが，システム開発においてニーズの状況的な可変性や，代理性や越境が重要な鍵となる認識を共有することが，大きなトラブルを減らすために有効ではないかと考える．

最後に SE 教育について言えば，今回の取材を通じて，SE に対して，多くの IT 企業がプログラミング研修を行っている一方で，ユーザ・コミュニケーションについての組織的な新人研修や教育を行っている例はなかった．

一般論での教育に意味が見出されにくいためかもしれないが，このような側面でのSEの研修も非常に重要だと考えられる．

これに準ずる教育の機会となっていると思われるのが，聞き取りの中で言及されていた「プロジェクト完了報告会（取材6)」である．特定の事例について案件が完了した後に技術的，社会的課題の洗い出しという意味で関わったIT企業内の関係者が集まってリフレクションをする場である．現状ではこのような報告会は実際に関わった関係者のみ参加するとのことだが，このような機会をより広く，組織的に活かしていくことができれば，案件の関係者以外の若手SEにとっても実践的な学習の機会にできるのではないだろうか．

謝辞

本研究にご協力いただいた取材先の皆様に心より御礼申し上げます．

注

（注1） 2017年8月に控訴審判決が出された旭川医科大学と東日本電信電話株式会社（NTT東日本）の係争はその一例である．電子カルテを中核とする病院情報管理システムが，開発過程で要求事項が膨れ上がったために失敗に終わった責任を巡り，旭川医科大学とNTT東日本が争った．なお控訴審判決は一審判決を覆して，旭川医科大学に約14億1500万円の支払いを命じる内容だった．

参考文献

［1］ Carroll, John M.（2000）Making use：scenario-based design of human-computer interactions, MIT Press（郷健太郎訳）『シナリオに基づく設計―ソフトウェア開発プロジェクト成功の秘訣』共立出版，2003

［2］ 独立行政法人情報処理推進機構 技術本部 ソフトウェア・エンジニアリング・センター（2012）「非ウォーターフォール型開発の普及要因と適用領域の拡大に関する調査報告書（非ウォーターフォール型開発の海外における普及要因編)」

［3］ Gladden, G. R.（1982）Stop the life cycle, I want to get off "Software engineering notes：an informal newsletter of the Special Interest Group on Software Engineering Vol.7, No.2, p.35-39

［4］ 原田悦子・日根恭子・南部美砂子・須藤智（2015）「業務電子化が引き起こす疑似越境とその修復－電子カルテ障害カンファレンスの縦断分析」香川秀太・青山征彦（編）『越境する対話と学び：異質な人・組織・コミュニティをつなぐ』，p.109-135 新曜社

［5］ Latour, B.（1987）Science in action：How to fol-low scientists and engineers through society. Harvard University Press.（川崎勝，高田紀代志訳）『科学が作られているとき―人類学的考察』産業図書，1999

［6］ 松嶋 登（2015）『現場の情報化 -IT利用実践の組織論的研 -』有斐閣

［7］ McCracken, Daniel D. & Jackson, Michael A.（1982）Life cycle concept considered harmful. "Software engineering notes：an informal newsletter of the Special Interest Group on Software Engineering.，Vol.7, No.2, p.29-32

［8］ 村田嘉利，大場みち子，伊藤恵，佐藤永欣（2013）『情報システムの開発法：基礎と実践』共立出版

［9］ 中村雅子・上野直樹（2008）「ネットワーク指向のデザイン・アプローチの提案：情報システムの運用開発事例の分析から」Cognitive Studies，15（4），p.627-643

［10］田丸恵理子，上野直樹（2002）「社会 - 道具的ネットワークの構築としてのデザイン」日本デザイン学会誌，9（3），14-21

［11］上野直樹，永田周一，ソーヤー理恵子（2006）「学習環境デザインのためのネットワーク志向アプローチ」上野直樹，土橋臣吾編『科学技術実践のフィールドワーク：ハイブリッドのデザイン』せりか書房，56-74

［12］テリー・ウィノグラード，フェルナンドフローレス（1989）『コンピュータと認知を理解する－人工知能の限界と新しい設計理念』産業図書