

ソフトウェア技術者の思考の特徴：その役割認識と価値観

Cognitive features of software engineers: Their role recognition and values in information system development

谷川 由紀子^{1,4}, 鈴木 栄幸², 加藤 浩³, 福住 伸一¹, 原田 悦子⁴
Yukiko Tanikawa, Hideyuki Suzuki, Hiroshi Kato, Shin'ichi Fukuzumi, Etsuko Harada

¹ NEC 情報・ナレッジ研究所, ² 茨城大学, ³ 放送大学, ⁴ 筑波大学
NEC Corporation, Ibaraki University, The Open University of Japan, University of Tsukuba
y-tanikawa@cw.jp.nec.com

Abstract

We investigate cognitive features of software engineers who design and develop information systems. Interviews for software engineers and usability experts about their working experiences, working environment, and recognition to work were executed, and the results were analyzed focusing on "their role recognition" and "values" in information system development. It was revealed that there is a difference between software engineers and usability experts in role recognition tendencies, which are based on system centered viewpoint and user centered viewpoint. It was also found that difference between them in sense of values that are similar to promotion focus and prevention focus in regulatory focus theory of Higgins (1997, 1998). These are suggested to be related to conflicts in collaboration between software engineers and usability experts.

Keywords Information systems, software engineers, thinking, role recognition, and values

1. 背景と問題

従来、情報システム開発、特に企業、官公庁、自治体等で使われる業務システムの開発はソフトウェア技術者が中心となって行ってきた。近年、システムの複雑化、大規模化に加えて、使いやすさ向上も強く求められるようになり、ソフトウェア技術者の負担がこれまで以上に重くなっている。この対応策として、使いやすさ向上に関わる専門的な知見^[1]を持つユーザビリティ専門家が開発プロジェクトに参画し、ソフトウェア技術者と協業する場面が増えてきている。

しかしながら、協業がうまくいかず、しばしば問題が起こっている。その原因を、E.Metzer^[2], T.Memmel^[3]は、ソフトウェア技術者が適用する情報システムの開発プロセスと、ユーザビリティ専門家が用いる使いやすさ向上のプロセスとの違

いにあると述べているのに対して、A.Seffah^[4-5], X.Ferre^[6]は、各々のプロセスを実践する人々のコミュニティの違いにあると捉えている。すなわち、ソフトウェア技術者とユーザビリティ専門家は、言葉や道具に加えて、設計活動におけるものの見方、役割認識、価値観、規範といった文化の異なるコミュニティに属しており、それが協業の大きな障害になっていると指摘している。

筆者らは、使いやすさ向上のためのソフトウェア技術者支援の研究に取り組む中で、既存のシステム開発プロセスに使いやすさ向上のプロセスを統合する試みを行ってきた^[7]。しかし、プロセスを統合しても、協業がうまくいかないケースが依然として発生している。問題ケースの分析^[8]から、その原因には、A.Seffah^[4-5]らの指摘する、ソフトウェア技術者とユーザビリティ専門家との、もの（対象・課題）の見方・捉え方、またその背景にある役割認識、価値観といった文化の違いが関係するのではないか、と考えた。

使いやすさ向上における協業の問題を解決するには、両者の認知特性、また文化の特性を明らかにする^[9-10]必要がある。しかし、A.Seffah^[4-5]らの研究においても、各々の特性に関する具体的な言及はわずかしかない。

そこで、本稿では、ソフトウェア技術者の情報システム開発における役割認識や価値観に焦点をあてて、ものの見方や捉え方の特徴^[11]との関連も含めて、ユーザビリティ専門家との比較をもとに特性を分析し、考察する。

2. 方法

2.1. インタビュー

ソフトウェア技術者 13 名（実務経験 10 年以下の若手 8 名とベテラン 5 名）およびユーザビリティ専門家 10 名を対象として、普段の仕事への取り組み方や認識、仕事に関わる環境（所属する組織や実行体制）、仕事の経験や受けてきた指導・教育等について、半構造化インタビューを行った。さらに、その半年後に、特に仕事の責任範囲や優先度の認識、目標意識をはじめとする「仕事に対する認識」を掘り下げると共に、システム開発に関わる実務経験をより詳細に確認することを目的とする半構造化インタビュー（フォローアップインタビュー）を行った。インタビューの時間は、初回インタビューとフォローアップインタビューを併せて一人平均 3.5 時間だった。また、インタビューの様子は全てビデオデータとして記録した。

フォローアップインタビューにおいては、システム把握の「3 層モデル^[11]」と、初回インタビューの内容をもとに各参加者の実務経験を年表形式にまとめた「パーソナルワークヒストリー」を活用した。「パーソナルワークヒストリー」は、参加者の技術者 / 専門家としての育成過程を可視化すると共に、参加者間での比較分析を容易にすることを目的として、実務経験を、共通項目に基づいて記述するものとした。設定した項目は、参加者個人の属性情報として、担当顧客の種類（業種・業務領域）、担当した開発フェーズ（要求分析、概要設計等）、プロジェクトにおける役割（メンバー、リーダー等）、また開発システムの属性情報として、開発の種類（新規開発、既存システム改修等）、開発規模、対象とする業務種類、の 6 項目である。参画したプロジェクト毎に、これらの項目の該当属性と経験年数とを対応付けて記述した。

2.2. 分析の手続き

23 名のインタビューデータおよびフォローアップインタビューデータからトランスクリプトを作成した。このトランスクリプトにインタビュー時のメモを加えて、情報システム開発における「役割認識」および「価値観」に着目した予備的な分

析を行った。具体的には、インタビュー参加者が語ったエピソードの中から「役割認識」や「価値観」が関係していると思われるものを抽出し、その構造を分析した。この分析結果をもとに、トランスクリプトの詳細分析の観点（コード）として、次の 5 点を抽出した。すなわち、ソフトウェア技術者およびユーザビリティ専門家の各々の、開発プロジェクトにおける「ミッション」「制約」、その中で感じている「葛藤」と「葛藤解決のアプローチ」、さらに、「普段の仕事の進め方」である。

インタビュー参加者のうち、顧客システムの受託開発（システムインテグレーション：SI）に主に携わるソフトウェア技術者 3 名（実務経験 10 年以下の若手 2 名とベテラン 1 名）およびユーザビリティ専門家 4 名（いずれも実務経験が 10 年を超えるベテラン、うち 2 名は、ユーザビリティ専門家としてだけでなく、ユーザビリティの知見を持つソフトウェア技術者としても活動している）を対象として、上記 5 つの観点での詳細な発話プロトコル分析を行った。まず、1) 各参加者のトランスクリプトの中から、ソフトウェア技術者にとっての 5 つの観点（例：ソフトウェア技術者の制約）に関係する発話セグメントと、ユーザビリティ専門家にとっての同様の観点（例：ユーザビリティ専門家の制約）での発話セグメントを抽出した。その後、2) 抽出した全ての発話セグメントを観点別にまとめて、3) そのうちのソフトウェア技術者に関するものに絞って、セグメント毎に発話内容の要約を作成した。4) この要約をもとに、主旨の同じものをまとめてグループを作り、各グループについて、内容のエッセンスを表す短文（例：予算を超えた作業はできない、レビューは減点をなくすためのもの）を作成した。5) この短文をもとに、各観点に対応する分類名（例：予算の制約、仕事を進める上での考え方）を決めた。この時、短文の内容は異なるが主旨は同一と考えられるものについては、まとめて大分類とした。さらに、6) 観点間での関連を分析するために、各分類に対して識別番号を割り当てた（例：制約 2、思考特性 4-1）。

次に、開発プロジェクトにおける「ミッション」「制約」、その中で感じている「葛藤」と「葛藤解決のアプローチ」について、相互の関連を分析した。また、「普段の仕事の進め方」については、同一観点内での分類間の関係と、主に開発プロジェクトにおける「制約」との関連を分析した。

3. 結果と考察

3.1. システム開発におけるステークホルダー

開発プロジェクトにおける「ミッション」「制約」、その中で感じている「葛藤」と「葛藤解決のアプローチ」を観点とした分析をもとに、顧客システムの受託開発におけるステークホルダー^[12]を役割別に集約すると、図1に示す4者になる。すなわち、開発した情報システムを利用して業務を実施する「現場の利用者」、顧客におけるシステム開発の責任者である「発注者」、顧客からの委託を受けてシステム開発を担う「ソフトウェア技術者（SE）」と使いやすさ向上の視点から開発に加わる「ユーザビリティ専門家（UE）」である。

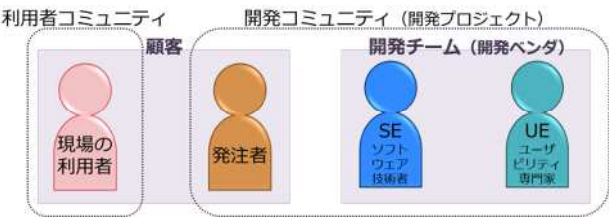


図1 情報システム開発におけるステークホルダー

このうち、「現場の利用者」と「発注者」は、同一の顧客企業に属する業務部門と情報システム部門であるケースが多い。また、情報システムの開発は、顧客側の「発注者」とシステム開発ベンダの「ソフトウェア技術者」を基本として、状況によって「ユーザビリティ専門家」が加わった開発プロジェクトとして実施される。

3.2. ステークホルダーの役割と関係性

開発プロジェクトにおける「ミッション」を観点とした分析を通じて、3.1 に示したステークホルダーのシステム開発場面における役割と、開発プロジェクトにおけるステークホルダー間の関係を整理した。この際、3.1 で述べたように、受託型のシステム開発プロジェクトにおいては、顧客

企業の「発注者」と開発ベンダの「ソフトウェア技術者」が基本の構成要員となることから、この2者に「現場の利用者」を加えた3者の関係にまずは着目した。次に、開発プロジェクト内の「発注者」と「ソフトウェア技術者」の役割分担を、標準的な開発工程とそこでの実施内容をもとに整理して加えた。その結果、システム開発プロジェクトにおける各ステークホルダーの役割と関係性には、主に次の5つのパターンがあることが明らかになった。

● パターン1： 製造請負型

「発注者」がビジネス上の要求と「現場の利用者」からの業務実施に関わる要求をもとに、システムに関わる要求を大まかな機能要求レベルまでまとめて「ソフトウェア技術者」に提供し、「ソフトウェア技術者」がシステムとして実現する。この際、「発注者」は「現場の利用者」から上がってくる要求をコントロールすると共に、優先順位付けする役割を担う。また、原則として、「ソフトウェア技術者」が「現場の利用者」と接触することはない。

● パターン2： 設計請負型

図2に示すように、「発注者」がビジネス上の要求と「現場の利用者」からの業務実施に関わる要求をもとに、システムに関わる要求をまとめて「ソフトウェア技術者」に提供する。

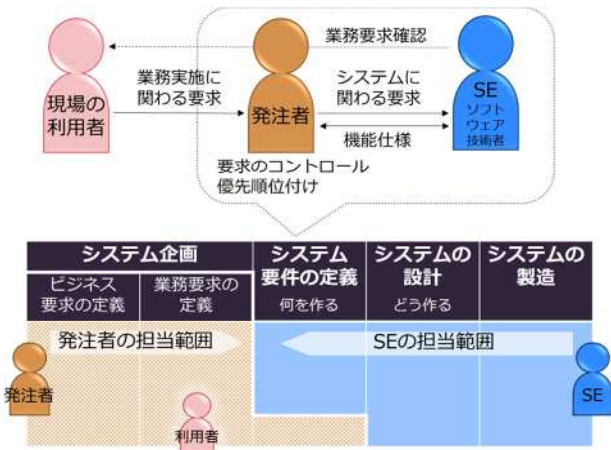


図2 設計請負型（パターン2）の場合

これを受けて、「ソフトウェア技術者」が要求をシステムとしてどのように実現するかを機能仕様にとめる。これに「発注者」の合意を得た上で、

「ソフトウェア技術者」がシステムとして実現する。この際、「発注者」は「現場の利用者」から上がってくる要求をコントロールすると共に、優先順位付けする役割を担う。また、「ソフトウェア技術者」が「現場の利用者」の要求を確認する場合には、原則として「発注者」を介して行う

● パターン 3：現場要求理解型

設計請負型（パターン 2）と同様に、「発注者」がビジネス上の要求と「現場の利用者」からの業務実施に関わる要求をもとにシステムに関わる要求をまとめて「ソフトウェア技術者」に提供する。これを受けて、「ソフトウェア技術者」が要求をシステムとしてどのように実現するかを機能仕様にまとめるが、その際「発注者」は、「ソフトウェア技術者」に「現場の利用者」からの要求を「発注者」と同様に把握していることを求める。「発注者」は「現場の利用者」から上がってくる要求のコントロールと優先順位付けを主導し、「ソフトウェア技術者」は合意した機能仕様をシステムとして実現する。

● パターン 4：現場仲介型

基本的な役割分担は現場要求理解型（パターン 3）と同じだが、業務システムの範囲・規模の拡大によって、「発注者」が個々のシステムにおける業務の細かいところまでは把握しきれなくなったために、図 3 に示すように、「ソフトウェア技術者」が「発注者」に代わって、「現場の利用者」の業務要求とシステムの仕様を繋ぐ役割も担う。

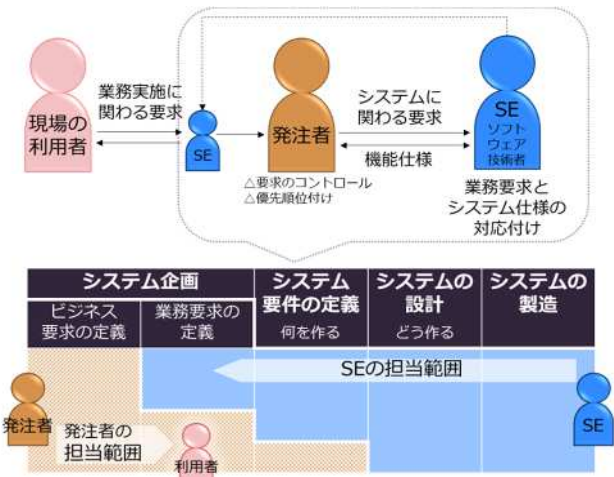


図 3 現場仲介型（パターン 4）の場合

また、「現場の利用者」から上がってくる要求のコントロールや優先順位付けについても、ここでの「発注者」の対応は十分ではないことが多い。一方で「ソフトウェア技術者」が「現場の利用者」と接触する場合には、設計請負型（パターン 2）と同様に原則として「発注者」を介して行う。

● パターン 5：企画参画型

「発注者」はビジネス上の要求をまとめる。一方、「現場の利用者」からの業務実施に関わる要求の収集とコントロール、優先順位付け、またそれらをもとにしたシステムに関わる要求の定義については、図 4 に示すように、「ソフトウェア技術者」と「発注者」が協力して、あるいは「ソフトウェア技術者」が「発注者」を先導して行う。これを受けて、「ソフトウェア技術者」が要求をシステムとしてどのように実現するかを機能仕様にまとめる。機能仕様の作成以降は、パターン 1（製造請負型）、パターン 2（設計請負型）と同様に進める。

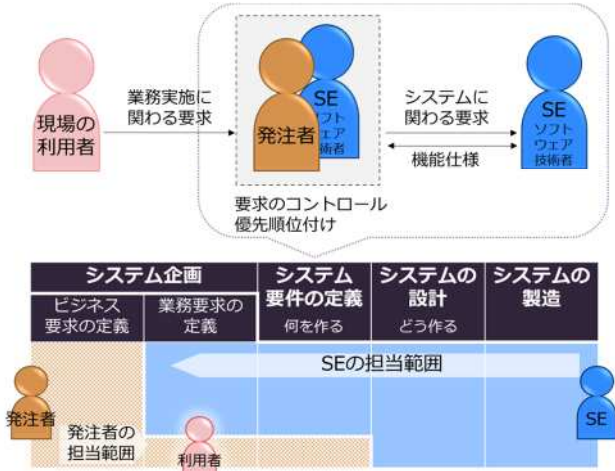


図 4 企画参画型（パターン 5）の場合

従来の受託型のシステム開発においては、上記のうち、設計請負型（パターン 2）や現場要求理解型（パターン 3）のケースが多かったのに対して、昨今は、現場仲介型（パターン 4）や企画参画型（パターン 5）のケースが増えてきていることを、分析を通じて確認した。また、そういったケースに際してソフトウェア技術者が戸惑っている状況も見えてきた。表 1 に、このような「ソフトウェア技術者の戸惑い」を示す対話例を示す。この例のインタビュー参加者は、これまで主に設

計請負型（パターン 2）で継続してプロジェクトに携わってきたが、新たに立ち上がったプロジェクトが企画参画型(パターン 5)になりそうだと、その戸惑いを語っている。発話例中、T はインタビュー参加者、K はインタビュー者を示す。

表 1 期待される役割の変化に関する対話例

K1:	そういう風にその本当の末端のユーザーと直にこう、間がなく、えー対話をするっていうような新しいこういう体制っていうのはウェルカムですか、それともむしろ勘弁してよっていう感じですか？
T1:	実際はまあ勘弁してよって思うんですけど、その人たちが実際に使うので、その人たちが実際に使うシステムには近くなるのかなあって、一応理にはかなってるんですけど、こう誰かがコントロールしないと、まあ逆に崩壊のほうへ向かっていっちゃうはずなので、難易度が高いのかなって気がします。

3.3. ソフトウェア技術者が感じている制約
開発プロジェクトにおける「制約」を観点とした分析の結果、ソフトウェア技術者が開発プロジェクトの中で感じている制約には、大まかに 10 種類あることが明らかになった。特に抽出した発話セグメント数(頻度)が多かった「制約」を表 2 に示す。なお、頻度は、異なる話題に対する発話セグメント単位でカウントした。

表 2 プロジェクトの中で SE が感じている制約

カテゴリ No.	制約種類 (頻度)	制約の内容	制約主体
4	体制 (8)	現場ニーズを知ることができない	SE
5-1	顧客との関係性 (7)	発注者に現場情報提示を依頼できない	SE
5-2		発注者には意見できない	SE
8-1	発注者の意識 (13)	現場のニーズを把握していない	発注者
8-2		現場より発注者の意向を優先する	発注者
8-3			

表 2 の制約の内容に着目すると、インタビュー参加者が頻繁に表出した「制約」が、プロジェクトの体制からくるもの、発注者自身の役割認識に関わるもの、また発注者との関係性に関わるもの、といったように、全て「発注者」との関わりの中で生じていることがわかる。また、

こういった「制約」は、多くのケースで複合して発生している。例えば、次の表 3 の対話例では、3.2 に示した設計請負型（パターン 2）の体制と役割分担で開始したはずのプロジェクトが、発注者側がその役割に慣れていないために製造請負型（パターン 1）に近くなる中で、担当のソフトウェア技術者(インタビュー参加者)が、発注者による設計（要件定義）の進め方に不安を感じていることを語っている。この発話例において、表 2 の制約のほとんどが語られている。

表 3 制約に関する発話例

T2:	登録系の機能があまり整理できなくて、今設計しているところは利用部門の人の話を直接聞くことがあまりできなくて、間に中継してる方がいるんですけど、その人がもう勝手にもうこう作って設計してしまって、うち側があまり口出しができない状態になってる
-----	--

3.4. ソフトウェア技術者の葛藤
開発プロジェクトにおける「葛藤」を観点とした分析の結果、ソフトウェア技術者が開発プロジェクトの中で感じている葛藤として、大まかに 12 種類、葛藤の原因の違いを加味して分類すると 22 種類のカテゴリが抽出された。特に抽出した発話セグメント数が多かった「葛藤」を、大カテゴリ（大まかな分類）の単位で表 3 に示す。

表 4 プロジェクトの中で SE が感じる葛藤の代表例

カテゴリ NO.	葛藤の種類 (頻度)	葛藤の原因 保有者	関連する制約
1	発注者がSEに期待する役割とSEの役割認識との不一致(11)	SE 開発PJ	3.スキル
3	発注者の役割認識と顧客との関係性の制約(15)	開発PJ	5-2.顧客との関係性 / 7.顧客内の関係性 / 8-1.発注者の意識
4	発注者のスキル不足と顧客との関係性の制約(9)	開発PJ	5-2.顧客との関係性
5	発注者の意識の制約(12)	発注者	8-1 / 8-2 / 8-3.発注者の意識
9	PJ体制の制約(7)	開発PJ	4.体制

表 4 のうち、「発注者が SE に期待する役割と SE の役割認識との不一致 (カテゴリ 1)」は、3.2 に示した各ステークホルダーの役割と関係性の変化がきっかけとなって生じるものである。すなわち、従来馴染んできた製造請負型 (パターン 1)、設計請負型 (パターン 2)、現場要求理解型 (パターン 3) から、発注者側の事情をトリガとして現場仲介型 (パターン 4)、あるいは企画参画型 (パターン 5) に変わっていることにソフトウェア技術者が気付かなかったり、また気付いても、現場の「業務」や「人」を見る視点が十分でない (ソフトウェア技術者のスキルの制約) ために、うまく役割を果たせずに感じる「葛藤」である。表 5 に、発注者が期待する役割の変更 (従来の設計請負型 (パターン 1) から企画参画型 (パターン 5)) に気付かずに行動したソフトウェア技術者の「葛藤」の様子を、支援に入ったユーザビリティ専門家が語っている例を示す。

表 5 役割認識の不一致による葛藤の対話例

T3: で、それで良くしてほしいんだけど、お客様自身も良く仕方がわからなくて、
K2: うん、
T3: ベンダにお願いしてるんですけど、ベンダ側も、なんか、言われたら付けたし、言われたら付けたしで、
K2: ああ、そうなっちゃいますよね。
T3: 全然良くないっていうんで、お客様は言ったから付けたしたのに、また変えろとか言われてっていうんで、収集つかなかったんですね。

一方、最も多い葛藤は「顧客との関係性の制約」、すなわち「発注者に意見できないこと」が関係するものである。「発注者の役割認識と顧客との関係性の制約 (カテゴリ 3)」では、発注者が開発プロジェクト内で役割を果たしていないと意見できない、また、開発プロジェクト内での役割に対して発注者の意識が低く、ソフトウェア技術者に余計な負荷がかかっているが意見できない、といった「葛藤」をソフトウェア技術者は感じている。また、「発注者のスキル不足と顧客との関係性の制約 (カテゴリ 4)」では、発注者にシステム

企画や開発プロジェクトを進めるスキルの不足を感じるが意見できない、との「葛藤」をソフトウェア技術者は感じている。表 6 に、表 3 の発話例に続けてインタビュー参加者 (ソフトウェア技術者) が語った「葛藤」の例を示す。この例において、ソフトウェア技術者は、制約の中で「発注者に意見できない」まま作業を進めてしまっている自分自身に対する「葛藤」を語っている

表 6 SE の制約に起因する葛藤の発話例

T4: うち側があまり口出しができない状態になってるのがよくない、よくないというか結果として作りながら本当に使えるのかなあって状態で完成させてしまってる

さらに、発注者が期待する役割の変更によって「現場の利用者の要求を把握すること」が求められるようになったにも関わらず、プロジェクトの体制としては現場から遠い位置づけで、相変わらず発注者を介さなければ現場と接触できない、また、発注者の現場の要求把握に対する意識が低く、ヒアリングなど現場の意向を直接確認する場の設定を依頼してもなかなか繋いでもらえない、といったところに、ソフトウェア技術者が「葛藤」を感じていることも明らかになった。

3.5. 葛藤解決のアプローチ

3.4 に示したような葛藤を解決するために、「ソフトウェア技術」は、表 7 に示す 6 種類の解決アプローチをとっていることが、「葛藤解決アプローチ」を観点とする分析から明らかになった。

表 7 葛藤解決のために SE がとるアプローチ

カテゴリ No.	アプローチ種類	葛藤解決者
1-1	現場を意識する	SE
1-2	現場との接点をつくる	SE
1-3	現場要求の反映を心がける	SE
2	発注者に意見 / 合意獲得に努める	SE
3	専門家に依頼する	専門家(UE)
4	PJの約束事を決める	開発PJ

開発プロジェクトにおいて、「現場の利用者」の要求を把握することが、ソフトウェア技術者にも役割として求められるようになる中で、特に若手

のソフトウェア技術者は、自分ができるだけ現場を意識する、また「発注者」に働きかけて現場と繋いでもらう、といったように、自身の努力で葛藤解決を試みていることが確認された。一方で、3.4 で示した「発注者が SE に期待する役割と SE の役割認識との不一致」は、ソフトウェア技術者自身のスキルの制約もあって容易に解決できるものではない。そのため、期待される新たな役割を、「ユーザビリティ専門家 (UE)」にそのまま担ってもらおうとするケースが増えてきている。具体的には、現場要求理解型 (パターン 3) のケースについて、図 5 に示すように、ユーザビリティ専門家にベンダ側の開発チームに加わってもらうことで、現場要求把握の役割を担ってもらう場合がある。この時、表 8 の発話例に示すように、発注者とソフトウェア技術者との間を繋ぐ役割を、ユーザビリティ専門家が同時に担うこともある。

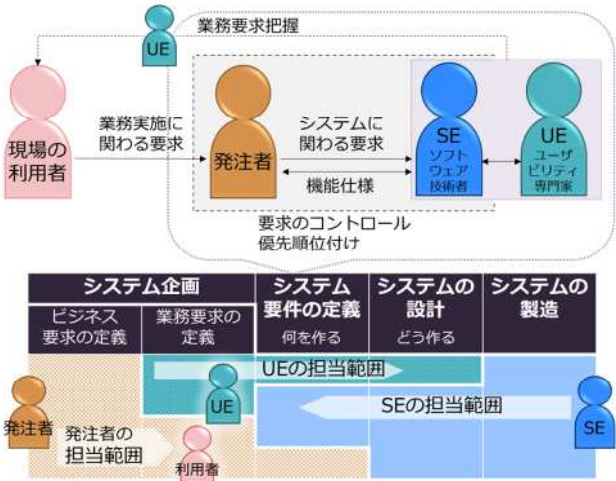


図 5 現場要求理解型における UE の役割

表 8 UE の役割に関する対話例

T4:	もうお客さんと話が収束しないので、
K3:	あー
T4:	まあ、お客さんとの間に入ってくれてというのが一番多いですね。実は直してくれよりも。
K3:	あー
T4:	あの一、会話がまずできないっていうので、入ってくださいっていうのが、

また、企画参画型 (パターン 5) のケースにおいて、図 10 に示すように、発注者と協力・先導

して現場からの要求のコントロールや優先順位付けを行う役割を、ユーザビリティ専門家に全面的に委託する場合がある。

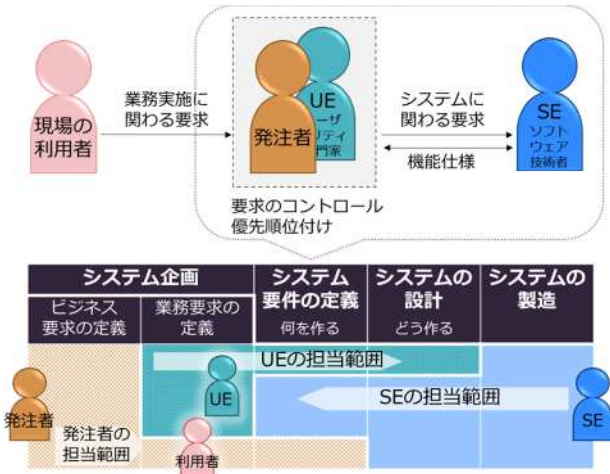


図 6 企画開発型における UE の役割

このように、新たな人的リソースの確保を伴う解決アプローチは、主にベテランのソフトウェア技術者がプロジェクトの状況によって試みている。しかしながら、表 8 の発話例にもあるように、プロジェクトに問題が発生してから動きは始めるケースも多いため、トラブルの解決 (問題の火消し役) がユーザビリティ専門家の開発プロジェクトにおける主要な役割になっていることが、分析を通じて明らかになった。

3.6. ソフトウェア技術者の考え方や行動の傾向

3.2 ~ 3.5 に示したソフトウェア技術者のシステム開発プロジェクトにおける感じ方や行動の背景を探ることを目的として、ソフトウェア技術者が普段仕事を進める中で、どのような考え方や行動をとる傾向があるのか、またどのような基準で行動しているのかを、プロジェクト活動に限らない「普段の仕事の進め方」を観点とした分析を通じて抽出した。表 9 に、抽出したソフトウェア技術者の考え方や行動の傾向を一覧にして示す。

ソフトウェア技術者は、「予算達成 (1-1)」「費用・期間厳守 (1-2)」を主要な行動の基準としていることから、常に「作業工数がひっ迫 (5-1)」している。そのために、余計な作業を発生させる可能性のあること (例えば、成果をより良いもの

表9 仕事における考え方や行動

カテゴリ No.	特性の種類	思考の特性	関連カテゴリ / 制約
1-1	行重加の基準	予算達成	5-1
1-2		費用・期間厳守	1-2、制約10
1-3		発注者に従う	制約5-2、2-1
2-1	行動の傾向	目の前の仕事に集中	1-1
2-2		質向上のための議論はしない	2-2、4-2
3-1	役割認識	「システムとして正常に動くことを保証する」 のが自分の役割	7
3-2		現場のことを考えるのは自分の役割ではない	制約4
3-3		担当範囲をしっかりと作るのが自分の役割	5-1
4-1	仕事を進める上での	レビューは減点をなくするためのもの	
4-2	考え方	責任をとれないことには関わらない	2-2、1-2、
4-3		知見の共有を重視しない	9-2
5-1	仕事をする環境の制	作業工数に余裕がない	1-1
5-2	約	現場ニーズを知ることができない	制約4、4-2
6	システム利用に關する 評価基準	自分が使えればOK	
7	システムの見方	機能中心	3-1
8	明文化したルール	開発効率・品質を落とさないためのルール	
9-1	目指す人材像	プロジェクトをコントロールできる	
9-2		専門分野(業種/技術)を持つ	

にするための議論や他のメンバによる評価(2-2)、設計時の不明点確認等)は避けるよう行動する。これは、余計な作業は余計な費用を発生させ、今ある利益を減らすものだからである。一方、作業の効率や品質レベルを落とさないことを目的として、明文化したルール(例:コード規約)や標準プロセス(例:設計書のレビュー(4-1))に従う。このような考え方や行動の基準は、Higgins^[13-14]の制御焦点理論における防止焦点(目標を義務の遂行と認識し、最低限のことを確実にこなすべく行動を制御する。損失の有無に敏感)と、類似するものと考えられる。

また、ソフトウェア技術者は、「発注者に従う(1-3)」を、もう一つの主要な行動基準とする。さらに、「システムとして正常に動くことを保証する(3-1)」を、開発プロジェクトにおける基本的な役割と認識している。しかし、先の3.3に示したような開発プロジェクトにおける「発注者」の期待する役割の変化に伴って、「発注者に従う(1-3)」, また「システムとして正常に動くことを保証する(3-1)」だけでは仕事が進まなくなって

きていることも認識している。そこで、ソフトウェア技術者として重視する基本的な考え方や行動の基準、すなわちそれらに現れる「価値観」を変えずに、この開発プロジェクトにおける変化を乗り切るアプローチとして、「ユーザビリティ専門家」との協業を捉えていると考察される。

3.7. 協業における価値観のぶつかり合い

ソフトウェア技術者は、自分自身が重視する基本的な考え方や行動の基準に現れる「価値観」を変えずに、発注者が期待する新たな役割に対応すべく、ユーザビリティ専門家に開発チームに加わってもらう方法をとる。しかし、ユーザビリティ専門家は、例えば、成果をより良いものにするための議論や他のメンバによる評価(SEの考え方や行動の特徴2-2の対極にあるもの)設計時の不明点に関する発注者への確認等を積極的に行う。すなわち、Higgins^[13-14]の制御焦点理論における促進焦点(目標を理想の実現と認識し、できるだけ良い成果を上げるべく行動を制御する。利得の有無に敏感)に類似する考え方や行動をとる。また「現場のことを考

えるのは自分の重要な役割（SE の考え方や行動の特徴 3-2 の対極にあるもの）」と考えて、行動する。

そのため、様々な場面で両者の間に行き違いが発生し、新たな「葛藤」を生じている。表 10 に、行き違いに関する発話例を示す。この例では、インタビュー参加者であるユーザビリティ専門家が、ソフトウェア技術者の 1 人として開発プロジェクトに加わった際に経験した「考え方の違い」について、インタビュー者として同席した他の専門家と語り合っている。

表 10 SE と UE の行き違いに関する対話例

- T5: 気持ちの持ち方だと思うんですけど、怒られるからなんかやだってなっちゃうんですよ。そうじゃなくて書けないからどうするか考えようって言うんですけど、あまり考えてくれないというか、その違いっていうのが、同じようなものがあるけど、ニュアンスだけ違うじゃないですか。その気持ちの持ちようが違うっていうか。
- K4: 多分本人たちはそういう風に言われても、じゃあ怒られないようにどうしようって戻るんですね。同じこと言われてるのに、その気の持ちよりの違いはありますね。確かに、なんで怒られたのって突き詰めたら自分たちが悪かった、お客さんに迷惑かけてるからって。

4. おわりに

情報システムの設計開発を担うソフトウェア技術者とユーザビリティ専門家に対して、仕事の経験や仕事に関わる環境、また仕事に対する認識についてインタビューを行った。インタビューの内容を、ソフトウェア技術者の開発プロジェクトにおけるミッションと制約、その中で感じている葛藤と解決アプローチ、さらに、普段の仕事の進め方に着目して分析した。その結果、ソフトウェア技術者は、従来の受託型のシステム開発とは異なる役割分担と体制を顧客から期待されるようになったことを受けて、自分たちの「役割認識」や「価値観」を変えずに、新たに加わった役割をユーザビリティ専門家に委託することで対応を試みていることが分かっ

た。しかし、協業の中で発生する行き違いの事例から、Higgins^[13-14] の制御焦点理論における予防焦点と促進焦点に類似する「価値観」の違いや、「システム視点」と「利用者(現場)視点」に基づいた「役割認識」の違い等が、両者の齟齬の原因となっていることが示唆された。

今回の詳細分析の対象としなかったユーザビリティ専門家について同様に分析することで、ソフトウェア技術者の役割認識や価値観、およびそれらを生む背景要因としての育成過程や仕事の環境等について、今後さらに詳細化を進める。また、それらをもとにして、ソフトウェア技術者が、プロジェクトの要請に応える新たな「価値観」や「役割認識」を受け入れて、使いやすさ向上に取り組むことができるようにするための育成プログラムや支援方法の策定に取り組んでいく。

参考文献

[1] D Norman. (1988): The psychology of everyday things. Basic Books (AZ), 1988.

[2] E Metzker and M Offergeld. (2001): "An Interdisciplinary Approach for Successfully Integrating Human-Centered Design Methods into Development Processes Practiced by Industrial Software Development Organizations." 8th IFIP International Conference, EHCI 2001: 19 ~ 33

[3] T Memmel, F Gundelsweiler, H Reiterer. (2007): "Agile human-centered software engineering." BCS-HCI '07 Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 1: 167-175

[4] A Seffah, and E Metzker. (2004): "The obstacles and myths of usability and software engineering." Communications of the ACM 47.12 (2004): 71-76.

- [5] A Seffah, J Gulliksen, M C. Desmarais, eds. (2005): Human-Centered Software Engineering-Integrating Usability in the Software Development Lifecycle. Vol.8. Springer, 2005.
- [6] X Ferre. (2003): "Integration of Usability Techniques into the Software Development Process." Bridging the Gaps Between Software Engineering and Human-Computer Interaction (2003): 28.
- [7] Y Tanikawa, R Okubo, S Fukuzumi. (2012): "Proposal of human-centered design process support environment for system design and development", Proceedings of the 4th Applied Human Factors and Ergonomics (AHFE) International Conference, pp.7825-7834.
- [8] Y Tanikawa, , et al. (2014): "Problems in usability improvement activity by software engineers - Consideration through verification experiments for human-centered design process support environment", HCI International 2014 Proceedings Vol.12 LNCS8521, pp641-651
- [9] 海保博之 (1992) : “ インタフェースの認知科学的諸問題 ”, 認知科学の発展 第 5 巻 特集「インタフェース」, 日本認知科学会, 講談社, 1992 : pp.1 ~ 4
- [10] 佐伯胖 (1992) : “ ヒューマン・インタフェースは異文化交流の場である ”, 認知科学の発展 第 5 巻 特集「インタフェース」, 日本認知科学会, 講談社, 1992 : pp.5 ~ 27
- [11] 谷川由紀子他 (2014) : “ 情報システム開発におけるソフトウェア技術者の思考の特徴に関する考察 ”, 2014 年度日本認知科学会第 31 回大会発表論文集, pp.149-158
- [12] 岩本 隆志(2014):システム開発におけるステークホルダーマネジメントについての一考察,生産管理:日本生産管理学会論文誌 20(2), 100-105, 2014-04,日本生産管理学会
- [13] E Higgins. (1997) : "Beyond pleasure and pain. ", American Psychologist, 52, 1280-1300.
- [14] E Higgins. (1998) : "Promotion and prevention: Regulatory focus as a motivational principle", Advances in Experimental Social Psychology, 30,1-46.