

## 60

### 要求明確化フレームワークによる開発者の支援<sup>1</sup>

#### 1. 概要

本編では、スマートデバイスを活用した業務システムを対象に、顧客の使いやすさに関する要求を、開発者が開発初期の段階で明確化する方式を構築し適用した事例を紹介する。この使いやすさに関する要求には、システムの表層的な画面や操作に限らず、システム利用者がどう使って仕事を進めたいか(例:外出先でレポートを作成したい／商品のデモをしたい)、どんな結果(例:余分な仕事が減る／対話がスムーズになる)を得たいかまでが含まれる[1]。

NEC の社内には、業務システムの SI (System Integration) に向けた開発標準が整備されている[2]。これまで、利用者の使いやすさの要求を明確化するため、人間中心設計 (Human-centred design: HCD) [3]の考え方にに基づき、超上流工程 (企画工程や要件定義工程) [4]において開発者が実施する具体的な作業、手順 (要求明確化のプロセス) を、この SI 向け開発標準の中に組み込んできた[5]。HCD は、顧客・利用者の要求を満たすため、設計と評価を繰り返す開発アプローチである。HCD の考え方に基づくこの要求明確化のプロセスでは、開発者は早期にシステムの利用イメージを具体化し、顧客がシステムに何を望んでいるのか、どのようなシステムであれば顧客が望むことを満たせるのかを顧客と確認することで、顧客の使いやすさの要求を明らかにしていく。それにより、利用時のよい体験や使い勝手を考慮したシステムの設計・開発ができる。

しかし、この要求明確化のプロセスは理想形を定めたものであり、現場に広く普及させるためには、実際の開発現場の活動を十分に考慮した支援方式が必要になる[6]。そこで、今回、特に過去のノウハウの蓄積が少なく顧客の要求を共有することが難しいスマートデバイスを活用した業務システムをターゲットに、実プロジェクトの分析を行い、開発現場の超上流工程の活動を明らかにした。これらを元に、上記プロセスを現場展開が容易な方式として構築した。

本編では、「2.取り組みの目的」で、解決すべき従来方式の課題について述べている。また、「3.要求明確化フレームワーク」では今回提案する支援方式の概要(「3.1」、「3.2」)、支援方式の構築方法(「3.3」)について順に述べた。最後の「4.取り組みの実施と効果」では、実案件での実施について述べている。

#### 2. 取り組みの目的

SI 向け開発標準の利用状況を調査したところ、実装、テストなどの下流工程に比べ、超上

---

<sup>1</sup> 事例提供：日本電気株式会社 大久保 亮介 氏

流工程に組み込んだ要求明確化のプロセスは適用が進んでおらず、現場へ広く普及させるためには課題があることが分かってきた。この要求明確化のプロセスは、HCD の専門家の視点で一つのあるべき姿を定めたものだ。しかし、現場には様々な開発プロジェクトがあり、開発者の状況も様々であるため、普及に向けては現場の活動をさらに反映させる施策が必要である。具体的には、次の2つの課題を解決することである。

#### 課題① 多様なプロジェクトへの適応：

超上流工程の活動は、プロジェクトの性質（システム利用者層、採用する実装技術など）や状況（実施中の工程、開発の期間など）によって様々である。特に、スマートデバイスでは、システムの用途や開発の進め方がこれまで以上に多様化している。そのため、画一的、普遍的なプロセスでは、現場（例えばプロジェクトの責任者）の視点からでは適用が難しいと考えられてしまう。

#### 課題② 多忙な開発者の作業工数削減：

現場の開発者の工数は常にひっ迫している。特に、スマートデバイスの場合、開発者にノウハウの蓄積が少なく効率的な開発が難しい。開発作業そのもの以外に新たな学習や実施項目があり、開発者に負荷がかかる様な施策では利用してもらえない。

課題①を解決するためには、普遍的なプロセスを提供するだけでなく、プロセスを多様なプロジェクトに合わせて変更・詳細化（テーラリング）する仕組みが必要になる[7]。課題②を解決するためには、開発者に一から成果物を作成させるのではなく、軽微な修正作業で済むように、適切な成果物のサンプルを活用する仕組みが必要になる[8][9]。

前述の2つの課題を解決するために、定義済みの要求明確化のプロセスに対して、プロセスをテーラリングする仕組みと、成果物のサンプルを活用する仕組みを組み込む(図 60-1 参照)。プロセスをテーラリングする仕組みとして、プロセスを適用するプロジェクトの特性に応じて、プロセスで定義した手順の一部を変更するテーラリングルール（手順で作成する成果物を切り替える／手順を実施しないなど）を設ける。また、成果物のサンプルを活用する仕組みとして、スマートデバイスを用いて実施する典型的な業務ごとに、プロセスで定義した一通りの成果物のサンプルをあらかじめ用意しておく。開発者は手順ごとに成果物を作成する際に、これらのサンプルを選択、修正する。

今回、開発者自身がプロジェクトに合わせて適用できるように、テーラリングとサンプル活用の2つの仕組みを要求明確化のプロセスに組み込んだものを、「要求明確化フレームワーク」として提案する。

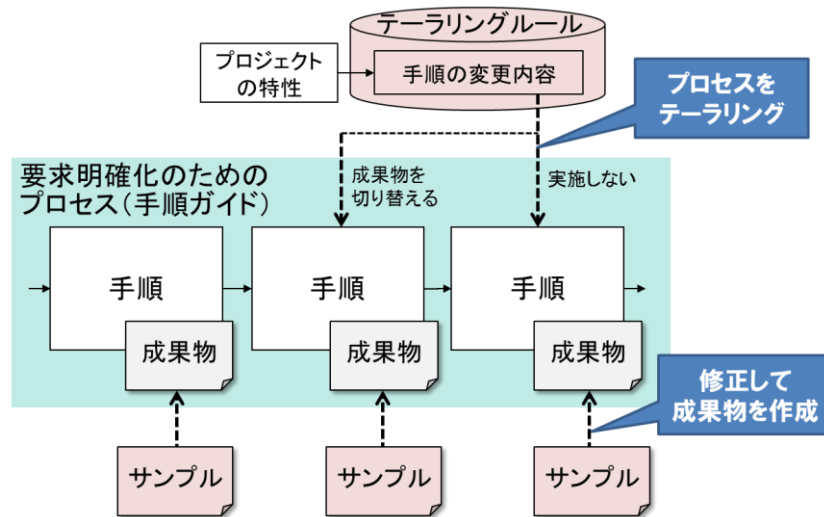


図 60-1 課題を解決するためのプロセスのテーラリングとサンプル活用の仕組み

### 3. 要求明確化フレームワーク

要求明確化フレームワークは、多様なプロジェクトの中で多忙な開発者が要求を明確にするために必要とされるプロセスと実行支援の仕組みを体系的にまとめたものである。実行支援の仕組みとして、「2.取り組みの目的」で述べたとおり、「プロジェクトに応じてプロセスをテーラリングする仕組み」と「サンプルを活用してプロセスの成果物作成を支援する仕組み」を構築した。

「プロジェクトに応じてプロセスをテーラリングする仕組み」を実現するためには、実際のプロジェクトを分析し、目的/期間など進め方に大きく影響する要因（プロジェクトの特性・状況）と、合意内容/合意のタイミングなど活動の特徴を明らかにした。これらを活用して、プロジェクトに応じてプロセスを分岐、取捨選択できるようにした。また、「サンプルを活用してプロセスの成果物作成を支援する仕組み」を実現するためには、事例を分析し、スマートデバイスを活用した業務システムの典型的な利用シーンごとに、プロセスで定義した成果物のサンプルを構築した。開発者がサンプルを選択し、それをベースにすることで、短時間で一定品質の成果物を作成できるようにした。

要求明確化フレームワークでは、開発者自身が要求明確化のプロセスをテーラリングしてプロジェクトに適用し、成果物のサンプルも加工してプロセスを実行することで、「2.取り組みの目的」で述べた課題の解決を狙う。「3.要求明確化フレームワーク」では、まず課題解決の前提となる要求明確化のプロセスについて述べた後で、今回提案する要求明確化フレームワークの概要を述べる。その後、フレームワークの構築の考え方、構築の過程などの構築方法について述べる。

### 3.1. 要求明確化のプロセス

「2 取り組みの目的」で述べたように図 60-1 に示した課題を解決するための仕組みをもとに、開発者が要求明確化を開発現場で実施するためのプロセスを定義した（図 60-2）。

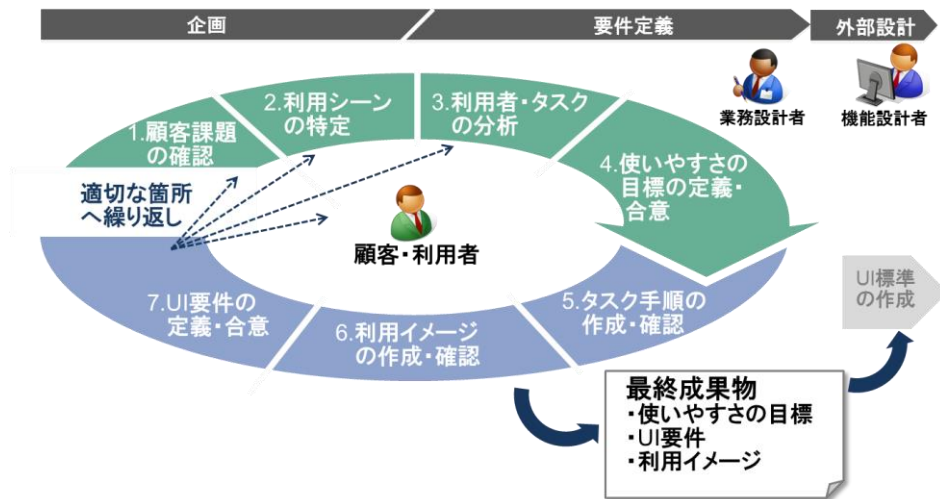


図 60-2 要求明確化のためのプロセス

このプロセスを構成する手順は以下の通りである。

- (1) 顧客課題の確認…顧客がシステムによって改善・向上を目指す課題を特定する。
- (2) 利用シーンの特定…課題に密接に関係するシステムの利用シーンを特定する。
- (3) 利用者・タスクの分析…利用者と実施する主なタスクの特性を明らかにする。
- (4) 使いやすさ目標の定義・合意…課題解決に必要なシステムの使いやすさを顧客と定義・合意する。
- (5) タスク手順の作成・確認…タスクを利用者視点で具体化する。
- (6) 利用イメージの作成・確認…タスクを実施するシステムの利用イメージを顧客・利用者に体験してもらうことで要求を確認・詳細化する。
- (7) UI要件の定義・合意…あわせてこの要求を要件として定義し、以降関わるステークホルダー間で共有する。

これらの手順は、システムへの要求を利用イメージと要件として顧客と具体的に共有できるまで、繰り返し実行する。この繰り返しの判断基準の例として、利用イメージに関する要求が新たに得られなくなることが挙げられる。このプロセスでは「使いやすさの目標」、「利用イメージ」、「UI要件」を最終成果物とし、これらの最終成果物は外部設計工程に引き継ぎ、特にUI標準を作成するための入力となる。

### 3.2. フレームワークの概要

要求明確化フレームワークは、プロジェクトに合わせて手順の変更を推奨するテラリングルールと、成果物作成のベースとなるサンプルを「3.1」で述べたプロセスに組み込んだ仕

組みであり、その適用手順とコンテンツを含むドキュメントを開発者に提供する。以下ではこのテーラリングルール、サンプルについてそれぞれ説明する。また、これらは「3.3」に示す方法で構築し、開発者に提供するドキュメントの作成までを行った。

### 3.2.1. テーラリングルール

「3.1」で述べたプロセスは要求明確化のための一つのあるべき手順を定めたものだが、超上流工程の活動はプロジェクトによって様々であるため、この手順が必ずしもそのままプロジェクトに適用できるとは限らない。そこで、このフレームワークでは、プロジェクトの特性・状況（例えば、開発期間／金額／支援フェーズ）に応じて、手順ごとに、その実施内容を変更して適用するためのテーラリングルールを定めている。このテーラリングルールは、プロジェクトの特性・状況を入力とし、手順を変更すべき箇所と内容を出力する。この手順の変更内容として手順で作成する「成果物の変更」と、「手順のスキップ」の2種類を定義した。

「成果物の変更」は、各手順の中で作成する成果物の内容を変更することである。「(6) 利用イメージの作成」を例にとると、成果物である利用イメージについて、簡易的なラフイメージから完成システムと同等のモックアップまで複数のレベルで定義し、プロジェクトの特性・状況に応じたレベルの利用イメージに変更する。具体的には、プロジェクトの特性・状況の1つ「支援フェーズ」が企画（企画段階での支援）の場合、「成果物として利用イメージはラフイメージを作成する」というようにテーラリングルールを設定している。

「手順のスキップ」は、手順の実行自体を一旦スキップし、それ以降の工程やプロセスの繰り返しの中で改めて実行することである。例えば、プロジェクトの特性・状況の1つ「支援フェーズ」が提案初期（提案初期段階での支援）の場合、この時点では、具体的な設計要件よりも、どのような場面でどう役立つシステムを目指すのかを顧客と意識合わせすることが重要になるため、具体的な設計要件を定義する「(7) UI 要件の定義・合意」を一旦スキップし、手順の(1)から(6)を繰り返す。その後、提案活動が進み、システムの構成や見積もりを考える段階で、スキップしていた「(7) UI 要件の定義・合意」を実施する。

ここで例に挙げた「支援フェーズ」以外にも、テーラリングルール適用のために入力するプロジェクトの特性・状況として「開発期間」、「開発者のスキル」、「システムの利用者」などを定義している。

このテーラリングルールの構築方法は「3.3」で述べる。

### 3.2.2. サンプル

要求明確化フレームワークでは、常に工数がひっ迫している現場の開発者でも、HCDの専門知識を前提とせず短時間で一定品質の成果物を作成できるようにするために、プロセスの中で成果物のサンプルを提供し、それをベースに成果物を作成できるようにしている。要求明確化フレームワークでは、図 60-3 に示す通り、特に支援が必要な手順に対して「利用シーンサンプル」、「タスク手順サンプル」、「画面サンプル」、「配色サンプル」、「画面要素サンプル」の5種類のサンプルを提供する。以下では各サンプルの概要を説明する。

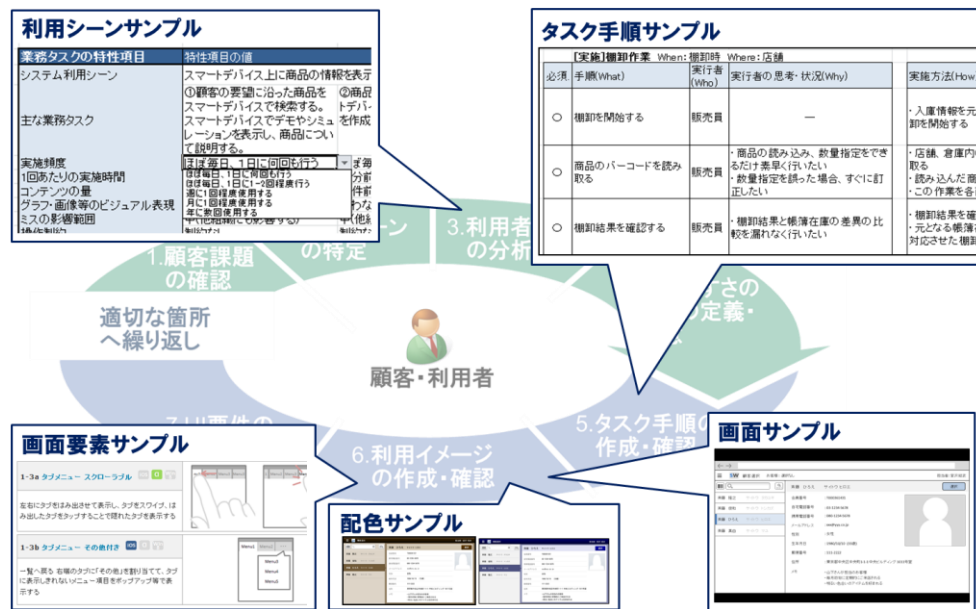


図 60-3 サンプルの全体像

「利用シーンサンプル」は、システムがターゲットとする利用者とタスクの特性・状況（利用者の業務知識や、タスクの実施頻度など）を明らかにすることを目的とする「(2) 利用シーンの特定」と「(3) 利用者・タスクの分析」を支援する。利用シーンのパターンごとに特性の値をサンプルとして定めている。開発者は、対象システムに近い利用シーン、利用者、タスクを選択することで、これらの特性を容易に決定することができる。

「タスク手順サンプル」は、顧客とスマートデバイスを活用した業務の進め方の意識を合わせることを目的とする「(5) タスク手順の作成・確認」を支援する。利用シーンのパターンごとに典型的なシステムのタスク手順をサンプルとして定めている。開発者は、システムが対象とする業務に近いタスク手順サンプルを選択し、加工することで、これらのタスク手順を容易に作成することができる。

「画面サンプル」、「配色サンプル」、「画面要素サンプル」は、顧客とシステムの利用イメージの意識を合わせることを目的とする「(6) 利用イメージの作成・確認」を支援する。「画面サンプル」は、利用シーンのパターンごとに典型的なシステムの画面、操作、遷移をサンプルとして定めている。開発者は対象システムに近い画面サンプルを選択し、加工することで、これらのシステムの利用イメージを容易に作成することができる。また「配色サンプル」

は「画面サンプル」を完成イメージに近づけ、配色を設定する際に利用する。「信頼感」など訴求したいイメージごとに、配色を施した画面の画像と色値のセットを用意した[10]。さらに「画面要素サンプル」は画面のレイアウト、入力フォーム、ジェスチャーなどの画面の構成要素をサンプルにし、「画面サンプル」に対して要素の入れ替えを可能にする。さらに、各要素の使い分けに重点を置いた解説を含んでいるため、「画面サンプル」を顧客に提示する際に、なぜこの要素を組み合わせているのか画面の設計理由を説明することができる。

これらのサンプルの構築方法は「3.3」で述べる。

### 3.3. フレームワークの構築方法

「3.3」では、「3.2」で説明したテラリングルール、サンプルの構築の考え方、構築の過程など、構築方法について述べる。

要求明確化のプロセス（「3.1」）はHCDに基づくあるべき手順の一つであるが、開発現場の実際の活動との間には乖離があるため、これを埋めるようにテラリングルール、サンプルを構築する。具体的には、要求明確化のプロセスと関連する開発現場の活動を抽出し、両者をできるだけ合致させるようにテラリングルールを定義する。また、抽出した活動の中で必要とされている支援を提供できるようにサンプルを定義する。

そのために実プロジェクトに対し、超上流工程の活動をヒアリングし、現状実施している手順とその課題の分析を行った。この現状の手順との合致、課題の解決を図るという方針のもと、フレームワークの構築を行った。

#### 3.3.1. ヒアリングの実施

ヒアリングは、社内のスマートデバイスを活用する業務システム開発の9プロジェクトに対し、超上流工程を担当する開発者へ対面式で行った。ヒアリングを行った9プロジェクトは、利用者（社員、一般消費者など）、業種（金融、流通など）、支援フェーズ（企画、要件定義）と、その特性、状況が異なるものを幅広く選定した。ヒアリングの中では、どの工程を担当したのか、どの程度の期間だったのかなど、プロジェクトの特性・状況を明らかにした上で、超上流工程の手順やそこでの課題について質問した。定義済みの要求明確化のプロセスを基準にして、プロジェクトで実施している手順をヒアリングするために、あらかじめプロセスをもとにした質問内容と回答の選択肢をヒアリングシートとして用意し、その内容を元に実施した。

#### 3.3.2. ヒアリング結果の分析

各プロジェクトの超上流工程の手順を同一の視点で比較、把握するために、ヒアリング結果を図 60-4 に示す例のように視覚化した。この視覚化表現では、現状のプロジェクトで実施している手順のうち、要求明確化のプロセスと関連があるものを抽出し、開発工程軸上に配置し、手順の実施者（顧客を含む）ごとに整理した。さらに、各手順についての開発者のコメントをマッピングした。

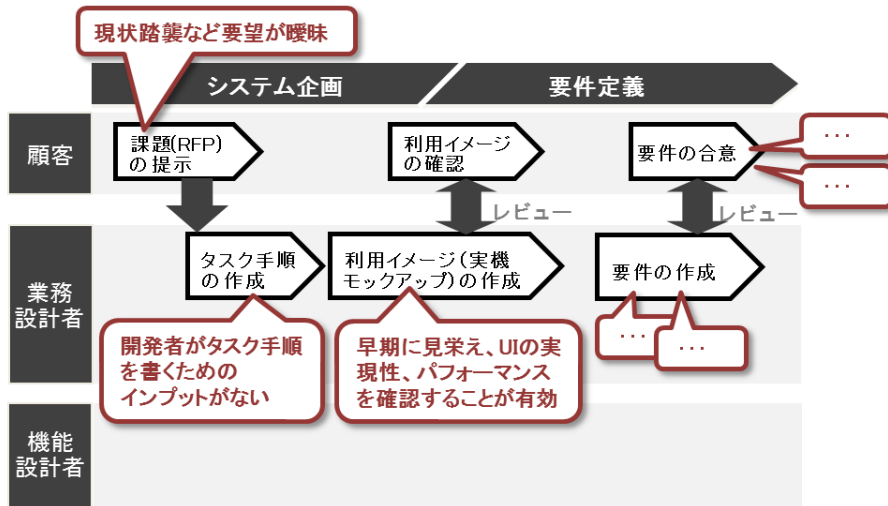


図 60-4 視覚化したヒアリング結果の例

この結果に対して、テラリングルールの定義を目的として、プロジェクト間の手順を比較し、その差異を抽出した（分析1）。また、提供すべきサンプルの定義を目的として、プロジェクトの各手順にマッピングされたコメントを、その手順で必要とされている支援の観点で分類した（分析2）。

分析1では、各プロジェクトの手順を比較し、プロジェクトごとの手順の差異点の抽出を試みた。この手順の差異点の抽出について、図 60-5 に示したように支援フェーズ、期間、スキルなどの特性・状況が異なる3つのプロジェクト A、B、C のヒアリング結果を例に用い説明する。

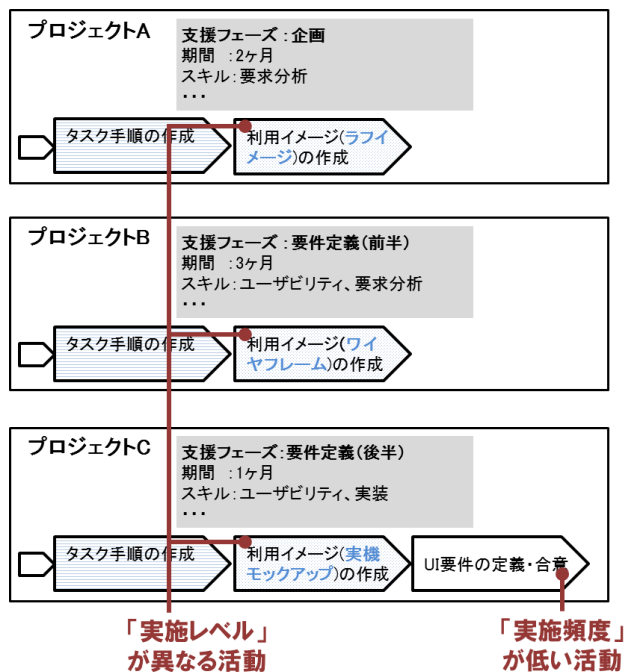


図 60-5 プロジェクトの比較の例



プロジェクト間での目立った手順の違いについて考察した結果、手順は同じだが成果物の内容が異なる場合があることが分かった。例えば、利用イメージとして、プロジェクト A では大まかな構成のみを示した図を、プロジェクト B ではより詳細に描画した図を、プロジェクト C では実機で動作するモックアップを用いた。この差異のタイプを「実施レベル」と定義した。また、どのプロジェクトでも実施する手順がある一方、特定のプロジェクトでしか実施されない手順があるなど、プロジェクトで実施する手順の頻度に差があることが分かった。例えば、プロジェクト C の手順「UI 要件の定義・合意」は、他のプロジェクトでは実施されなかった。この差異のタイプを「実施頻度」と定義した。この2つのタイプの差異に注目し、「実施レベル」が異なる手順をグループ化して抽出、「実施頻度」が他より低い手順を抽出した。

さらに、ここで抽出した手順の差異点は、プロジェクトの特性・状況とどのように関連するのかを分析した。例えば、前述のとおり、実施レベルの異なる手順として「利用イメージの作成」を抽出したが、企画工程にあるプロジェクト A ではラフイメージを、要件定義工程の後半にあるプロジェクト C では実機モックアップを作成する。この例では、「支援フェーズ」が進むにつれて「利用イメージの作成」の実施レベルが詳細になることが説明できる。同様に、実施頻度の低い手順として「UI 要件の定義・合意」を抽出したが、企画工程および要件定義工程の前半にあるプロジェクト A とプロジェクト B ではこの手順を実施しない。この例では、「支援フェーズ」が早い段階では、手順「UI 要件の定義・合意」の実施頻度が低くなることが説明できる。ここで例に挙げた支援フェーズ以外も含め、プロジェクトの特性・状況の中から、プロジェクト間の手順の差異点を説明できるものを抽出した。

以上のように、分析 1 ではプロジェクト間で手順を比較し、実施レベルの異なる手順のグループ、実施頻度の低い手順を抽出した。さらに、これらの手順の差異点と「支援フェーズ」などプロジェクトの特性・状況の関連付けを行った。これらの手順の差異点を要求明確化のプロセスに入れ込み、プロジェクトの特性・状況に応じてプロセスの手順を切り替えられるように、テラリングルールを構築することとした。

次に、分析 2 では、サンプルで支援すべき手順を明らかにするため、プロジェクトの各手順にマッピングされたコメントを必要とされている支援を観点に分類した。結果、支援を必要としているというコメントは表 60-1 に示す通り、「知識が必要」、「手法が必要」の2つに分類することができた。開発者の知識不足を補うためには、成果物の典型的なサンプルを揃えて提供することが有効と考え、「知識が必要」のコメントがマッピングされた手順を抽出し、この手順をサンプルで支援することとした。

表 60-1 コメントの分類

コメント分類	説明
知識が必要	手順を実行するために知識が必要とのコメント
手法が必要	手順を実行するために手法が必要とのコメント
その他	支援の必要性に関係のないコメント

このコメントの分類と支援する手順の抽出について、図 60-6 に示したプロジェクトのヒアリング結果を例に説明する。このプロジェクトの手順の1つである「タスク手順の作成」にマッピングされたコメント「開発者がタスクの手順を書くためのインプットがない」は、「知識が必要」に分類した。ここから、手順「タスク手順の作成」を支援するために、タスク手順のインプットをサンプルとして提供することとした。同様に「知識が必要」に分類したコメントがマッピングされた他の手順に対しても、コメントの内容から必要な知識をサンプルとして提供する。なお、「手法が必要」に分類したコメントがマッピングされた手順については、手順の詳細な実行方法として一般的な手法（フィールド観察、インタビュー等）を組み入れるようにプロセスを更新することとした。

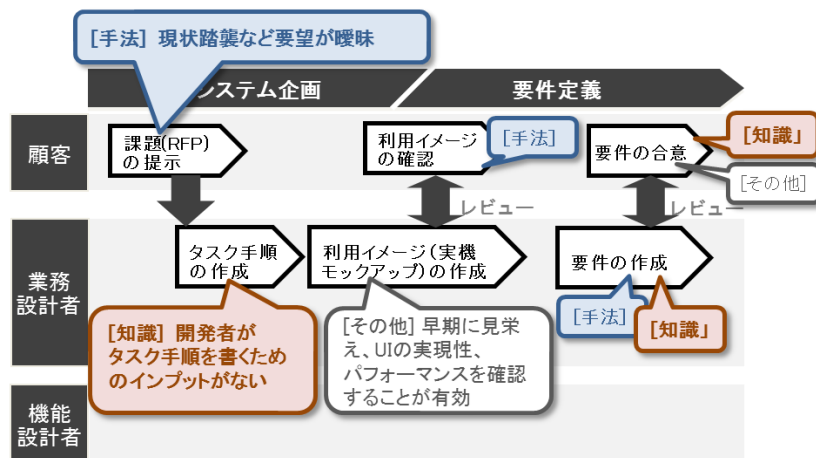


図 60-6 コメントを分類したヒアリング結果の例

以上の2つの分析結果は、幅広い特性・状況が取りそろえよう選定した9つのプロジェクトから得られたものであり、スマートデバイスを活用した業務システム開発に対して広く当てはめられるものと考えられる。

### 3.3.3. フレームワークの構築

「3.3.2」で述べた実プロジェクトの分析結果を活用し、以下のようにテーラリングルールとサンプルの構築を行い、要求明確化のプロセスへ組み込んだ。

#### (1) テーラリングルールの構築

「3.3.2」の分析によって、各プロジェクトの活動には「実施レベル」と「実施頻度」の違いがあることを明らかにした。各プロジェクトで「実施レベル」が異なる活動はグループ化し、プロセスの1つの手順の中に組み込んだ。すなわち、この手順には、ラフイメージ、ワイヤフレームなど、実施レベルの異なる複数の成果物を定義し、プロセスの実行時にプロジェクトの特性・状況に応じて、この手順で作成する「成果物の変更」をするようにルールを設定した。「実施頻度」が低い活動は、プロセスの手順に対応付け、プロジェクトの特性・状況に応じて「実行のスキップ」を選択できるよ

うにルールを設定した。このようにして、「成果物の変更」と「手順のスキップ」の2種類のテーラリングルールの構築を行った。

## (2) サンプルの構築

「3.3.2」の分析の結果、「知識が必要」に分類したコメントをもとに、現場の開発者がプロセスの各手順を実行するために必要な知識を補う成果物のサンプルを提供する。この考えのもと構築したサンプルの詳細を、表 60-2 に示す。1 列目に、「知識が必要」に分類したコメントを得た要求明確化の手順を、2 列目にそのコメントから明らかになった必要な知識を、3 列目に知識を提供するためのサンプルの構築方針を、4 列目に方針に基づき構築するサンプルを、それぞれ記載している。例えば、「(5) タスク手順の作成・確認」に対しては、タスク手順の第一案を記述するための何らかのインプットが必要になり、これに対し、サンプルとして、あらかじめスマートデバイスを活用した業務システムの典型的なタスク手順を複数用意しておくとした。

表 60-2 構築するサンプル一覧

手順	必要な知識	サンプルの構築方針	サンプル
2.利用シーンの特定	・スマートデバイスを活用する業務の内容	スマートデバイスを業務に活用した典型的な利用シーンのサンプルを提供	・利用シーンサンプル
3.利用者・タスクの分析	・現場の開発者が分析、ペルソナ・シナリオ作成する知識	利用シーンごとに典型的な利用者とタスクのサンプルを提供	
5.タスク手順の作成・確認	・手順を書くためのインプット	利用シーンごとタスク実行の手順のサンプルを提供	・タスク手順サンプル
6.利用イメージの作成・確認	・UIの善し悪し(スマデバの開発経験がないため) ・なぜこのUIがいいかの説明 ・イメージの議論方法(初めから詳細を提示すると、議論が発散する。修正に時間がかかる)	・利用シーンごとに画面レイアウト、画面遷移のサンプルを提供 ・ラフなものから詳細なものまで複数の具体度のサンプルを用意 ・設計理由を明示	・画面サンプル -ラフイメージ -ワイヤフレーム -モックアップ 配色サンプル ・画面要素サンプル

また、これらのサンプルはスマートデバイスの業務活用を網羅するように取り揃える必要があるため、スマートデバイスの業務活用の事例を収集、分類し、典型的な利用シーンのパターンを作成、それをもとにサンプルを構築した。

まず、社内プロジェクト、社外の公開情報を対象として、268 件の事例を収集した。各事例からは、業種、利用者、対象業務、スマートデバイスを利用した業務の進め方を抽出した。この事例を、業種の違いではなく、スマートデバイスを誰がいつどこでどうやって使うかといった業務の進め方、手順を観点として分類した。例えば、ビル設備点検と、訪問介護の事例では、業種は異なるが、訪問先でスマートデバイスを用いた作業内容確認、結果確認という手順が共通するため同一に分類した。結果、表 60-3 に示すように、事例から「営業業務」、「店舗業務」、「フィールド業務」、「精算業務」の4つの大分類と業務内容で細分化し9つの小分類を作成した。この9つの分類の事例をペルソナ・シナリオとして記述したものを利用シーンのパターンとした。各パターンについて、それぞれ類似した案件の担当者にヒアリングを行い、すべてのペルソナ・シナリオがありうること、現場の業務感覚と概ね合致していることを確認

し、妥当性を検証した。この検証済みの利用シーンのパターンを元に各サンプルを作成した。このように利用シーンのパターンに基づいて、各種のサンプルを作成するため、一つの利用シーンのパターンに基づくサンプルの間（例えば、商談のタスク手順サンプルと画面サンプル）では、前提条件、設計指針の一貫性は保たれている。

表 60-3 事例の分類

事例の大分類	事例の小分類
営業業務	商談
	商談管理
店舗業務	販売
	案内
	管理
フィールド業務	点検
	運輸
	回診・訪問看護
生産業務	作業実施

## 4. 取り組みの実施と効果

今回構築した要求明確化フレームワークを、社内プロジェクトに対してパイロット適用した。そして、その効果を確認した後 SI 向け開発標準に組み込み、社内に展開した。ここでは、パイロット適用の結果と、社内展開を行うにあたって関係者から得たフィードバックについて、それぞれ述べる。

### 4.1. パイロット適用

「2 取り組みの目的」で述べたとおり、本フレームワークでは課題①「多様なプロジェクトへの適応」と、課題②「多忙な開発者への作業工数削減」を解決することを目指している。今回のパイロット適用では、課題①に対しての有効性の確認を行った。

具体的には、社内で進行中の 2 プロジェクトに対し、直接参入してその特性・状況に合わせたフレームワークのテーラリングと適用を行い、プロジェクトごとに事前に設定した「明らかにすべき要求」が得られることを確かめた。これにより、フレームワークに組み込まれたプロセスのテーラリングの仕組みが適切に機能することを実証した。

表 60-4 に、2 つのパイロットプロジェクトの特性・状況を示す。また、表 60-5 に、ガイドドキュメントに記載のテーラリングルールの内、2 つのプロジェクトの特性・状況に関わる部分を抜き出し示した。表 60-5 に示したテーラリングルールでは、横列にプロジェクトの特性・状況を、縦列に要求明確化のプロセスの手順を割り当て、各セルにテーラリング内容を記載している。

表 60-4 適用プロジェクトの特性・状況

要求明確化の プロセスの手順	プロジェクトの特性・状況	
	プロジェクト1	プロジェクト2
システムの概要	営業担当者がタブレットを使い営業活動を行う	一般消費者が施設利用時にスマートフォンを使い情報を参照
開発の形態	受託開発	製品開発
支援フェーズ	提案	要件定義
支援の目的	顧客自身曖昧なシステムの利用シーンの明確化	広く使ってもらうために必要な機能の選定と使いやすさの実現
支援する工程での開発者	営業x2、業務設計者x1	業務設計者x2、実装担当者x1
開発期間	1年	6ヶ月
UIの開発難易度	低：標準部品の組合せ	中：一部、独自実装の必要性があり
...	...	...

表 60-5 テーラリングルールの例

	支援フェーズ		システムの利用者	開発期間	開発者スキル	...
	企画・提案	要件定義(後半)				
1.顧客課題の確認	—	[成果物の変更] 入手済み情報を整理した 顧客課題を作成	—	...	...	...
2.利用シーンの特定	—	[成果物の変更] 入手済み情報を整理した 利用シーンを作成	—	...	...	...
3.利用者・タスク の分析	—	[成果物の変更] 入手済み情報を整理した 特性項目を作成	[成果物の変更] 利用者へのインタビュー、 観察結果を追加	...	...	...
~~~~~						
6.利用イメージの 作成・確認	[成果物の変更] ラフイメージのみ作成	[成果物の変更]ワイフ レームを詳細に作成	[成果物の変更]モック アップまで作成	...	...	...
7.UI要件の定義・合意	[実行のスキップ]	—	—	...	...	...

表 60-4 に示す通り、プロジェクト 1 は、タブレットを用い営業担当者の営業活動（客先プレゼン、報告書作成など）を支援するシステムの開発であり、プロジェクト 2 はスマートフォンを用い、一般消費者が施設利用のための情報を入手するシステムの開発である。両者の大きな違いとして、開発の形態は、プロジェクト 1 が「受託開発」、プロジェクト 2 が「製品開発」である。また、本フレームワークの支援フェーズは、プロジェクト 1 が「企画工程」、プロジェクト 2 が「要件定義（設計着手後）」である。一般的に工程が上流であるほど顧客の要求はあいまいになる。

続いて、本フレームワークを各プロジェクトに適用した詳細について述べる。プロジェクト 1 は、営業活動にタブレットを活用したいという漠然とした顧客の要望があるのみで、具体的な利用シーン（タブレットでどんな業務を行うのか）が絞れておらず、提案活動が止まっていた。そこで、本フレームワークの適用により明らかにすべき要求を「顧客自身あいまいなシステムの利用シーン」と定め、提案活動の推進を狙った。企画工程にあるプロジェクト 1 は費用や時間などのリソースが限られており、本フレームワークをそのまま適用することが難しかった。そこで、ラフな利用シーンを素早く提示し、顧客の反応を反映しながら要

求を引き出すようにプロセスのテーラリングを実施した。

具体的には、表 60-5 に示したテーラリングルールに従い、支援フェーズが企画工程であることから、「(6) 利用イメージの作成・確認」について、提案内容のイメージを素早く共有するため簡易的なラフイメージのみを作成するように「成果物の変更」を適用した。さらに、ラフイメージを顧客に提示し要求を引き出した後に、具体的な設計要件を決定するため、この段階では「(7) UI 要件の定義・合意」には「実行のスキップ」を適用した。

こうした結果、限られた提案の期間を超過することなく、利用シーンの3つの候補を具体的なタスク手順、利用イメージとして可視化することで、今回明らかにすべき要求である、顧客が最も必要としている利用シーン「商談のきっかけ作り」を特定した。これによって提案活動を進めることができた。

その他、プロジェクト 1 への適用で明らかになった本フレームワークの効果として、「可視化と評価のサイクルを素早く回せること」と「議論の発散、すれ違いを防げること」がある。前者の理由は、タスク手順や利用イメージのサンプルを活用することで、可視化にかかる労力、時間が削減できたことと、顧客との議論の場で別案を具体的に示すことができたためである。後者の理由は、初めから出来上がりを想定したシステム構成や画面を用意するのではなく、抽象度の高いタスク手順やラフイメージから議論していくことで、機能の詳細やデザインの細部ではなく、「顧客自身あいまいなシステムの利用シーン」の議論に集中できたためである。

プロジェクト 2 は、プロトタイプ作成まで実施したものの、実装中の機能が、広く利用してもらうために必要かどうかの確証が得られず、本開発へ進めずにいた。そこで本フレームワークの適用により明らかにすべき要求を「ターゲットの利用者が必要とする機能」と定め、開発活動の推進を狙った。プロジェクト 2 では既に設計に着手し、プロトタイプ開発を行っていたため、本フレームワークをそのまま適用して一から超上流工程をやり直すことが難しかった。そこで、入手済みの情報を整理して、利用シーンを分析し、それをもとに詳細な画面仕様のレベルまで利用イメージを作成するようにプロセスのテーラリングを実施した。

具体的には、支援フェーズが、画面設計に着手している要件定義工程の後半であることから、表 60-5 に示したテーラリングルールに従い、「(1) 顧客課題の確認」から「(3) 利用者・タスクの分析」について、顧客・利用者から新たに情報を入手するのではなく、入手済みの情報を整理し分析を行うように「成果物の変更」を適用した。また、「(6) 利用イメージの作成・確認」について、詳細な画面レイアウトや画面遷移を表現したワイヤフレームを作成するように「成果物の変更」を適用した。

こうした結果、適用前には想定が不十分だった利用者・タスクの特性、使いやすさ目標を定義し、これらを判断基準にして5つの機能の候補から、今回明らかにすべき要求である、利用者に優先して提供すべき3つの機能を決定できた。

その他、プロジェクト 2 への適用で明らかになった本フレームワークの効果として、「実際には使われない機能を開発してしまうリスクを低減できること」がある。その理由は、タ



ターゲットとする利用者の特性やその目標を定義し、これらを判断基準とすることで、開発やビジネスの視点ではなくシステムの利用者視点を第一にした機能の選定ができたためである。

以上、2つのプロジェクトへのパイロット適用を通して、本フレームワークでは、特性・状況が異なるプロジェクトに対してプロセスをテーラリングすることができ、その結果、各プロジェクトで使いやすさの要求の明確化が可能になることを確認できた。

## 4.2. フィードバック

本フレームワークをSI向け開発標準に組み込み、社内展開するまでに、HCD 専門家との議論、開発標準の主管部門とのレビューや内容調整、社内展示会での事業部門（フレームワークの利用部門）との意見交換など、関係者から多くのフィードバックを得た。これらの中には、既に本編で紹介したフレームワークに反映済みのものも多くあるが、ここでは要求明確化フレームワークの今後の展開に関係するフィードバックを抽出して示す。表 60-6 は、発言元の関係者（HCD 専門家、開発標準の主管部門、事業部門）ごとに、良い点／改善すべき点で分類したフィードバックコメントである。

表 60-6 今後の展開に関するフィードバックコメント

	HCD専門家	開発標準の主管部門	事業部門(利用部門)
良い点	<ul style="list-style-type: none"> <li>・サンプルを、ケースにあわせて細分化して用意することで非専門家でも取り組める(これが最適なアプローチと感じる)</li> <li>・サンプルは事例分析がベースなので信頼感があり、現場でそのまま使ってもらえる</li> <li>・ガイドとサンプルは教育用の教材にする発展性がある</li> </ul>	<ul style="list-style-type: none"> <li>・サンプルの多くは選択入力方式なので、ドキュメント記述の内容や粒度に悩む現場の課題に合致している</li> <li>・各手順の成果物間で内容を転記する必要がなく開発者に余計な負担をかけない</li> </ul>	<ul style="list-style-type: none"> <li>・HCDの教育だけでなく、このような標準、サンプルの整備は重要</li> <li>・プロセスを辿ることで、設計案の理由を説明することができる</li> </ul>
改善すべき点	<ul style="list-style-type: none"> <li>・現場に効果を伝え、モチベートするための施策も必要</li> </ul>	<ul style="list-style-type: none"> <li>・目標やUI要件の記述は開発者には高度なため、選択式にして欲しい</li> <li>・設計・開発工程のツールと連携したい</li> <li>・開発工数を考慮する支援が欲しい</li> </ul>	<ul style="list-style-type: none"> <li>・サンプルを選ぶ支援も欲しい</li> <li>・ガイドドキュメントのページ数(65頁)が多く、手に取りづらい</li> <li>・効果を定量的に説明できないと、実際に導入しづらい</li> </ul>

まず、良い点として、サンプルを活用し成果物の作成を支援する本フレームワークのアプローチが、HCD の知識・経験に乏しい開発現場の支援に適している主旨のフィードバックがいずれの関係者からも得られた。特に、HCD 専門家からは、通常、サンプルは適用先の案件と内容が離れており効果は見込めないものが多いが、今回のフレームワークでは、実際の事例に基づき、ケースにあわせて細分化されたサンプルを提供するため、現場でそのまま利用可能と評価された。一方で、今回のサンプルはフレームワーク構築時点の事例に基づくもののため、今後も新たなサンプルの追加や更新など継続的な整備が必要になると考える。

また、事業部門からは、プロジェクトの工程を進める上では、顧客や自部門の上長へ案に至る理由を説明し合意を得ることが必須になるので、本フレームワークのプロセスの前半で定義した顧客課題、利用者・タスクの特性等が、後半で作成する要件や画面の根拠となる点が評価された。このフィードバックを受け、合意形成に向けて、どのように要求に優先順位付けをするか、どのステークホルダーにどのような表現形式で要求を伝えるか、を判断する

ための更なる支援とそのため新たな研究が必要と考える。

次に、表 60-6 に示した改善すべき点として、今回開発したフレームワークでは、UI 要件の記述、開発工数の考慮、適したサンプルの選定など開発者にとって高度となる判断を支援しきれていないことが分かった（開発標準の主管部門と事業部門のコメント）。現在、これらの支援のために方式の拡張を進め、一部は既にフレームワークに組み込みを行っている。

また、超上流工程で決定した内容をその後の工程へ正しく引き継ぐことは開発全体を見たときに非常に重要であることが分かり、その施策の一つとして、開発標準の主管部門が社内に展開する設計・開発工程のツール、プロセスとの連携も進めている。

フレームワークをいかに現場に普及させるかの観点では、現場活動に合致した仕組みを提供する取り組みに加えて、その効果を導入判断者に分かりやすく伝えることが必要であることを再認識した（HCD 専門家、事業部門のコメント）。例えば工数削減など、効果として何を伝えることが現場のモチベーションにつながるのか、効果をどう定量化するのかは、今後の大きな研究課題である。

## 5. 今後の取り組みと考察

本編では、スマートデバイスを活用した業務システム開発を対象に顧客の要求を明確化する方式について、多様なプロジェクトへの適応と多忙な開発者の作業工数削減を実現するためのフレームワークを提案、構築した。

多様なプロジェクトへ方式を適応するために、特性・状況の異なるスマートデバイス向けシステム開発の実案件を分析し、プロジェクト間の手順の違いと目的、期間など手順に大きく影響するプロジェクトの特性を明らかにした。これらを活用し案件に応じテーラリング可能なフレームワークを実現した。また、多忙な開発者が方式を効率的に実施できるようにするために、スマートデバイスの導入事例の分析を行い典型的な利用シーンを抽出、ここからスマートデバイスの開発案件を網羅する成果物のサンプルを構築した。

今回構築したフレームワークを社内の進行中のプロジェクトに適用した結果、以下の効果を確認した。

- ・特性や状況の異なる複数のプロジェクトに対し、フレームワークが備えるテーラリングの仕組みによって要求明確化が可能になること（課題①に対する有効性）。
- ・フレームワークには、「可視化と評価のサイクルを素早く回せる効果」、「議論の発散、すれ違いを防げる効果」、「実際には使われない機能を開発してしまうリスクを低減できる効果」があること。

今後の課題として、サンプルを活用することで開発者自身がフレームワークを効率的に適用できるか課題②の検証を行う。また、今回構築したフレームワークをより開発現場に普及させていくために、要件の記述、開発工数の考慮、サンプルの選定など開発者にとって高度な判断が必要になる作業の負荷軽減や、本フレームワークがターゲットとしている超上流工程から設計工程、開発工程へと分断なく成果物や合意事項を引き継ぐための他の開発プロセ



ス、ツールとの連携を進めていく。さらに、数多くのプロジェクトに適用し、テーラリングルールやサンプルをより現場のプロジェクトに即したものになるように改良、増強を進めていく。

## 参考文献

- [1] 山之内亜由知：使いやすい業務システム開発のために—ユーザーと業務の状況に適した最新 UI の選定手法—、野村総合研究所 IT ソリューションフロンティア 2011 年 12 月号、pp.10-13、2011
- [2] 小林茂憲、小泉健：スマートデバイス対応アプリケーション開発を効率化する 業務システム構築基盤「System Director Enterprise」、NEC 技報 Vol.65,No.3,2012
- [3] ISO 9241-210:2010 Ergonomics of human-system interaction – Part210:  
Human-centred design for interactive systems
- [4] IPA ソフトウェア・エンジニアリング・センター：経営者が参画する要求品質の確保～超上流から攻める IT 化の勘どころ～ 第 2 版、オーム社、2006
- [5] Tanikawa, Y., Okubo, R., Fukuzumi, S. : Proposal of human-centered design process support environment for system design and development; AHFE International Conference 2012, pp.7825-7834,2012
- [6] 大久保亮介、谷川由紀子、福住伸一：開発者を支援する要求明確化フレームワークの構築と実践、ヒューマンインタフェース学会論文誌 Vol.17,No.4,2015
- [7] Pedreira, O., Piattini, M., Luaces, M., Brisaboa, N. : A systematic review of software process tailoring; ACM SIGSOFT Software Engineering Notes,32.3,pp.1-6,2007
- [8] 足立哲朗、田口謙太郎、吉田貴宏、木内英紀：SaaS ビジネスにおける運営管理方式に関する考察、経営情報学会全国研究発表大会要旨集 2011,pp.99-103,2011
- [9] 橋本隆成：図解 はじめての「開発のための CMMI」とプロセス改善 第 2 版、日刊工業新聞社、第 2 章,2013
- [10] 矢野有美、大久保亮介、谷川由紀子、福住伸一：画面配色パターンを活用した顧客要求明確化方式の提案、ヒューマンインタフェースシンポジウム 2013、3331L,2013

掲載されている会社名・製品名などは、各社の登録商標または商標です。

独立行政法人情報処理推進機構 技術本部 ソフトウェア高信頼化センター (IPA/SEC)