

# 設計情報の可視化によるシステム改修における 作業全貌の把握作業支援システムの提案

Proposal of the Support System to Estimate the Modification Area of a Software System by Visualizing Design Information.

福田貴三郎<sup>1</sup> 村瀬健太郎<sup>1</sup> 松本滋<sup>2</sup> 笈田佳彰<sup>3</sup> 岡田伊策<sup>3</sup>

Takasaburo FUKUDA<sup>1</sup>, Kentaro MURASE<sup>1</sup>, Shigeru MATSUMOTO<sup>2</sup>, Yoshiaki OIDA<sup>3</sup>,  
and Isaac OKADA<sup>3</sup>

<sup>1</sup> 株式会社富士通研究所

<sup>1</sup> Fujitsu Laboratories Limited

<sup>2</sup> 株式会社富士通システムズ・イースト

<sup>2</sup> Fujitsu Systems East Limited

<sup>3</sup> 富士通株式会社

<sup>3</sup> FUJITSU LIMITED.

**アブストラクト:** パッケージソフトや既存システムの改修において、改修対象となる機能の規模の見積りにおける具体的な WBS (Work Breakdown Structure) 等の作業全貌の把握作業には、システムの全体構成や個々の機能の詳細に関する知識が不可欠であり、豊富な経験と高度なスキルを有するベテラン SE のノウハウが必要である。本研究では、パッケージソフトや既存システムの設計書に記述されている情報を基に、作業全貌の把握作業時にベテラン SE が頭に思い描いている、改修の影響範囲や改修における各機能の重要度などの設計情報を、簡単に把握できる形で可視化し、作業全貌の把握作業を支援するシステムを提案する。

## 1. はじめに

経済産業省の調査<sup>[1][2]</sup>によると、日本の企業では汎用的なパッケージソフトを導入するにあたり、約 90 パーセントのケースにおいて何かしらの改修・カスタマイズを行っている。また、システム構築を行った企業のうち、約半数以上が新規の開発では無く、従来のシステムの再構築を行っていると報告されている。このように、現在の日本の企業におけるシステム開発では、パッケージソフトや既存システムの改修を行うケースが大きな割合を占めている。

パッケージソフトや既存システムの改修において、改修対象となる機能<sup>1)</sup>の規模の見積りにおける具体的な WBS などの作業全貌の把握作業には、システムの全体構成や個々の機能の詳細に関する知識が必要不可欠である。しかし、ベテラン SE がそれらの

知識をどのように頭に思い浮かべ、作業全貌の把握作業を行っているかは、ノウハウとして暗黙知化されており、現実には属人性により生産物の品質にはばらつきが生じている問題がある。また、そのようなノウハウを持ち合わせたベテラン SE であっても、限られた時間の中での概算見積りが求められる場合には、改修対象となる機能の見落としなどにより、誤った見積り結果を算出してしまうケースがある。

そこで本研究では、パッケージソフトや既存システムの改修時において、ベテラン SE が頭に思い描いている設計情報を分析し、その設計情報を可視化する事により、作業全貌の把握作業の効率化を行う事が出来ると考えた。本稿では、ベテラン SE へのヒアリングを基に設定した二件の可視化要件、実際に開発した可視化 UI、および可視化要件のうち「改修対象となる各機能の重要度」について行った評価結果について述べる。

<sup>1)</sup> 画面・帳票・バッチ等を組み合わせて、データ処理の単位としてまとめたもの。

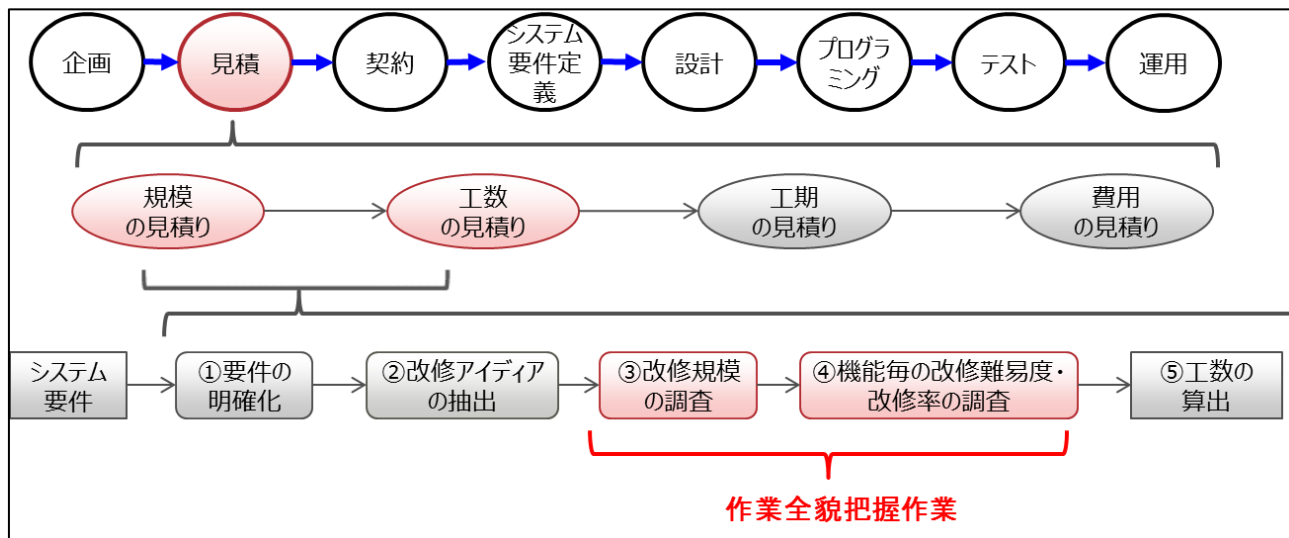


図1 システム改修における見積り業務フロー

## 2. 見積り業務フローと可視化要件

### 2.1 システム改修における見積り業務フロー

パッケージソフトや既存システムの改修における、見積り業務のフローを図1に示す。図中一段目のフローは、「SDEM<sup>2)</sup>」に準拠した開発プロセスであり、二段目のフローは、開発プロセスにおける見積り業務の工程を細分化したものである。そして三段目のフローは、見積り業務の工程における規模・工数の見積りを、ベテランSEへのヒアリングを基に、独自により細かく分解したものである。図中三段目のフローに即して、見積り業務の工程の流れを簡潔に説明する。

#### ① 要件の明確化

顧客から提示されたシステムの改修要件を掘り下げる事により、具体的な要件を作成する作業。

#### ② 改修アイデアの抽出

要件を実現するために、必要となる改修アイデアを検討する作業。複数のアイデアが検討される事もある。

#### ③ 改修規模の調査

改修アイデアを実現した場合に、どれくらい改修範囲が及ぶのかを分析する作業。

#### ④ 機能毎の改修難易度・改修率の調査

改修アイデアを実現した場合における、個々の機能毎の改修難易度・改修率を算出する。アイデアが複数ある場合には、採用するアイデアを決定

する。

#### ⑤ 工数の算出

機能毎の改修難易度・改修率を基に、トータルの工数を算出する作業。

この見積り業務の工程の流れで、本研究では作業全貌の把握作業に該当し、ベテランのノウハウ依存が特に強い領域である。「③アイデア毎に改修規模の調査」および「④機能毎の難易度・改修率の算出」に焦点を当てる。①、②に関しては、一般的に要件定義と呼ばれる分野であり、本研究ではスコープ外とする。また、⑤に関して、一般的に工数を算出する手法として、FP（ファンクションポイント）法<sup>3)</sup>が広く知られている。FP法では、機能数および複雑度から、開発工数を算出するが、パッケージソフトや既存システムの改修では、図1の③、④に該当する、改修対象となる機能の数や、各機能の改修難易度・改修率の調査結果が入力データとして必要である。これらの工程は、従来ではベテランSEのノウハウに頼っていた領域であり、本研究では、ベテランSEへヒアリングを行い、これら工程においてベテランSEが頭に思い描いている設計情報を分析し、必要となる可視化要件の抽出を行った。

### 2.2 可視化要件の抽出

「③アイデア毎に改修規模の調査」では、まず機能要件を基に、まず直接的に改修が必要となる「項目<sup>3)</sup>」、「テーブル（マスタ）」「サブシステム<sup>4)</sup>」、「機能」を考える。そして、各要素を改修した場合に、その影響がどれくらいの範囲で及ぶのかを考え、間接的に改修が及ぶ範囲を分析する必要がある。例えば、ある項目名の桁数を改修する要件の場合、直接的に改修を行うのは該項目であるが、間接的には、

<sup>2)</sup> <http://fenics.fujitsu.com/itil/introduction/standard.html>

<sup>3)</sup> テーブルのカラムと同義。

<sup>4)</sup> 機能の集合を単位としてまとめたもの。

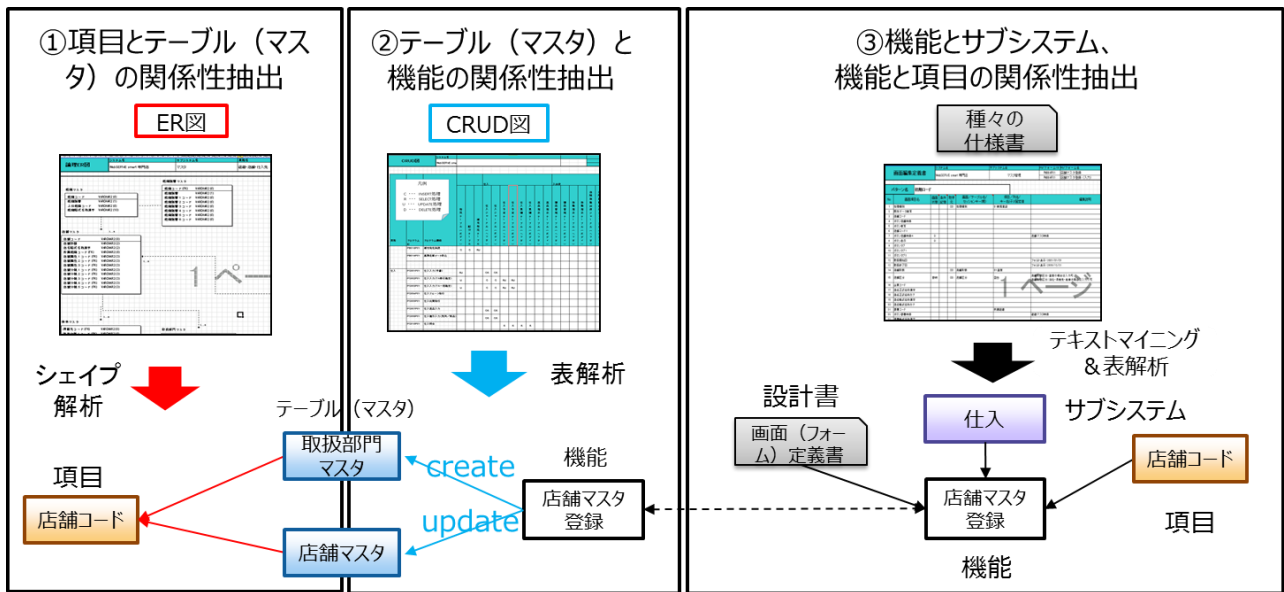


図2 サブシステム・機能・テーブル・項目間の関係性の取得

該項目を持つテーブル、およびそのテーブルに入出力を行っているサブシステムや機能にまで影響が及ぶ。対象となるパッケージソフトや既存システムに精通しているベテランは、間接的な影響が及ぶ範囲を、設計書を読みながら上記の項目・テーブル・サブシステム・機能の関係性を頭に思い浮かべる事で、改修規模を正確に推定する。しかし、設計書を精査する時間が無い場合や、経験の浅い非ベテランSEが見積りを行う場合には、当然見落としが発生するリスクが存在する。そのため、この工程では、改修の影響のある項目・テーブル・サブシステム・機能の関係性を可視化してユーザに提示する事で、改修規模調査の効率・精度を向上する事が出来ると考えられる。

「④機能毎の難易度・改修率の算出」では、影響が及ぶ各機能の設計書を精査し、改修難易度および改修率を調査する必要がある。しかし、影響範囲が広い場合、限られた時間内ですべての機能を細かく精査するのは困難である場合が多々ある。また同様に、アイデアが複数ある場合に、全てのアイデアに対して、改修難易度や改修率の調査を行おうとすると、多大な工数が発生する。実際にベテランSEにヒアリングを行ったところ、ベテランSEの場合は、精査すべきと考えられる重要な機能と、そうで無い機能との見極めを行うことにより、効率的に作業を行っている事が明らかになった。一方、非ベテランSEの場合は、影響が及ぶ各機能を一から順番に精査を行っているため、限られた時間内では、本来精査すべき機能が精査出来ていないケースがある事が分かった。そのため、この工程では、重要な機

能とそうで無い機能とを判別するための「重要度」を可視化する事で、機能毎の難易度・改修率の算出の効率を向上出来るのでは無いかと仮説を立てた。また、機能の重要度を可視化する事によって、アイデア間の比較効率が良くなる効果も期待できる。

以上の事を踏まえ、本研究で提案する可視化UIでは、以下の2つの可視化要件を設定した。

- (1) 改修の影響範囲となる、サブシステム・機能・テーブル・項目間の関係性が読み取れる
- (2) 改修対象となる各機能の重要度が読み取れる

### 3. 可視化に必要なパラメータの取得

#### 3.1 改修の影響範囲の取得

可視化要件(1)の改修の影響範囲を可視化するためには、事前に設計書情報から、サブシステム・機能・テーブル・項目間の関係性を取得する必要がある。事前に関係性を取得しておく事で、改修対象となる項目名などを入力とした際に、繋がりのある各要素を影響範囲として可視化する事が可能となる。設計書情報からサブシステム・機能・テーブル・項目間の関係性を取得する方法をまとめたものを図2に示す。

本提案システムでは、設計書から「①項目-テーブル」、「②テーブル-機能」、「③機能-サブシステム、機能-項目」の関係性を取得し、メタ情報として管理している。設計書ならびメタ情報の管理には、中村<sup>[4]</sup>らが開発した文書管理プラットフォーム「KA SHIWADE」を活用している。

「①項目-テーブル」および「②テーブル-機能」

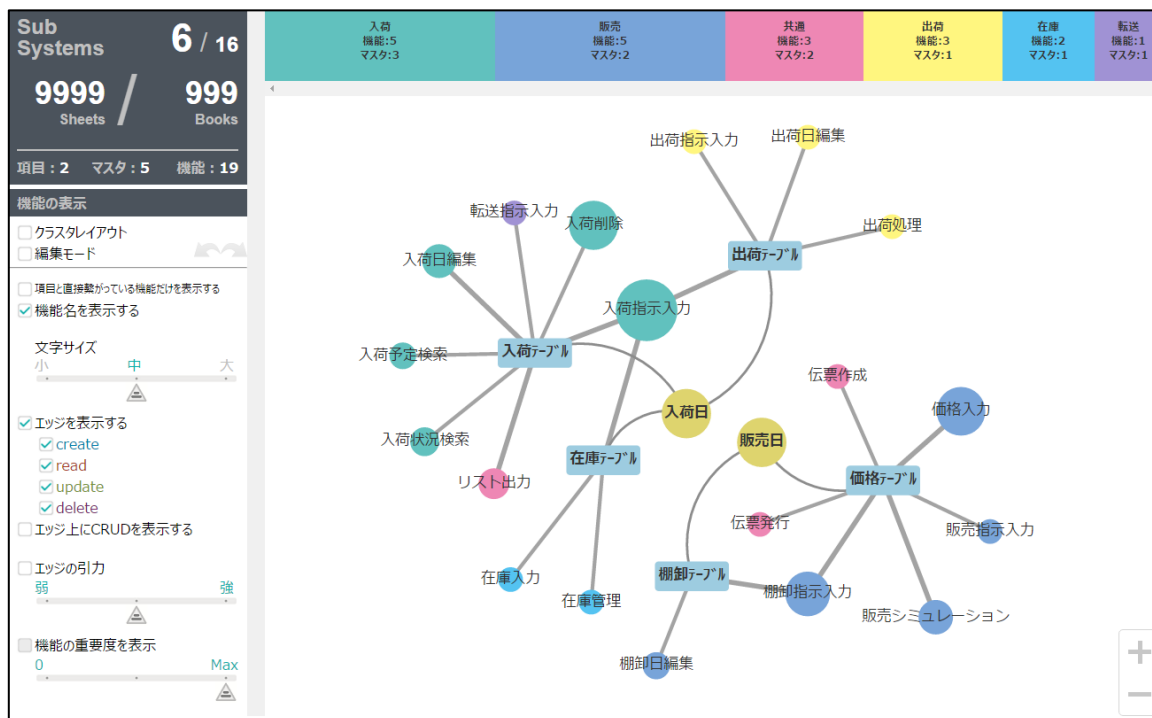


図3 開発した可視化 UI

の関係性は、それぞれ ER 図、CRUD 図をシェイブ解析および表解析を行うことによつて取得している。なお、これらの取得方法に関しては、文献<sup>[5]</sup>を参考にしている。

「③機能-サブシステム、機能-項目」に関しては、各機能に紐づいている設計書から取得を行っている。サブシステム・項目共に、事前に辞書を構築しておき、表構造でサブシステム名および項目名らしさの高いキーワードに対して辞書を用いてフィルタリングをかける事によつて、高精度に取得している。

### 3.2 改修対象となる各機能の重要度の計算

可視化要件(2)の機能の重要度を算出するにあたり、本研究ではベテラン SE の考える機能の重要度を体系化する事を目指した。以下に、ベテラン SE へのヒアリングを通して得られた、機能の重要度に影響を及ぼすと考えられるパラメータを列挙する。

- ・各機能の種別
  - 直接改修の発生する機能か否か
  - 直接改修は発生しないが、結合テストが必要か否か
- ・機能に属する設計書の数（重要な機能ほど設計書が多い）
- ・設計書内に出現する改修の必要な項目名の出現度

数（何度も該項目名が出現する機能ほど重要）

- ・機能が参照するテーブルの数（重要な機能ほど参照するテーブルの数が多い）
- ・テーブルとの CRUD 関係（Create や Update を行う機能ほど重要）

各機能の種別に関して、直接改修が発生するか否かは改修を行う項目を直接参照しているかどうかで判別を行っている。本研究では、事前にベテラン SE に学習データ用のパッケージソフトの各機能に重要度のスコア付けを行った正解データセットを作成してもらい、上記パラメータを変数として重回帰分析を行うことにより、機能の重要度を算出する計算式を作成した。機能  $i$  に属する設計書  $j$  内において改修する項目名の出現スコアを  $f(j)$ 、機能  $i$  に紐づく設計書数を  $w$ 、機能  $i$  の種別を  $v$ 、機能  $i$  が接続するテーブルの次数を  $t$ 、機能  $i$  の CRUD スコアを  $c$  とした場合の機能  $i$  の重要度  $g(i)$  を下記式(1)により算出する。

$$g(i) = \alpha \sum_{j=0}^w f(j) + \beta v + \gamma t + \delta c \quad (1)$$

※ CRUD スコア  $c$  は、Create:4, Update:3, Read:2, Delete:1 として算出

※  $\alpha, \beta, \gamma, \delta$  は重回帰分析により算出

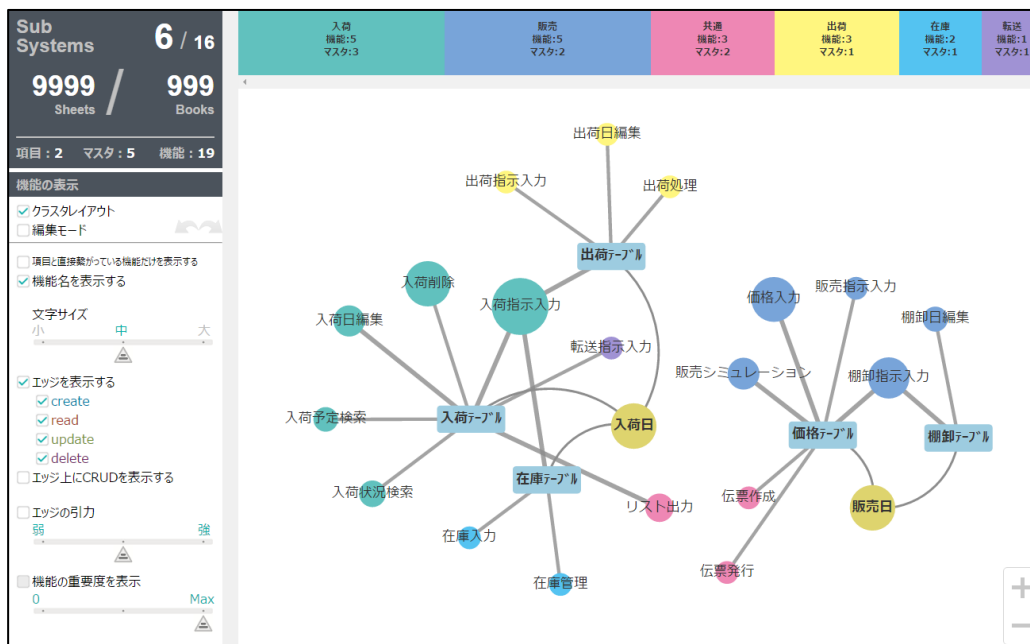


図4 クラスタレイアウト

## 4. 可視化 UI の開発

2章で述べた可視化要件を実現するために開発した可視化 UI を図3に示す。

図3は、「入荷日」と「販売日」の二つの項目を改修した場合において、改修影響のある範囲を可視化したものである。項目を金色の丸いノード、テーブルは青い四角、機能は金色以外の丸いノードで表しており、「項目ーテーブル」間のエッジは曲線、「テーブルー機能」間のエッジは直線で表現している。各ノード間の配置は、グラフ描画アルゴリズムの一つである力学モデル<sup>6)</sup>によって位置情報を計算しており、各ノードは重ならず自動で配置される。各サブシステムと機能の繋がりについては色で表現しており、ネットワーク図上のサブシステムバーの色と、各機能のノード色が対応している。また、サブシステム毎に機能の集合をより直感的に俯瞰出来るように、「クラスタレイアウト」というレイアウトアルゴリズムを開発した。このレイアウトを実行すると、サブシステムが共通している機能間に張っている見えないエッジの引力を強める事により、図4で示すように、サブシステム毎にノードが集まり、サブシステム毎に改修の影響範囲を俯瞰する事が出来る。

このように、サブシステム・テーブル・機能間の関係性を表現する事により、2章で述べた可視化要件(1)「改修の影響範囲となる、サブシステム・機能・テーブル・項目間の関係性」を実現している。また、

可視化要件(2)「改修対象となる各機能の重要度」への対応として、機能の重みをノードサイズとして表現している。例えば図5の例では、機能「入荷指示入力」などが重要な機能であると読み取る事が出来る。

## 5. 評価

本章では、二つの可視化要件のうち、「改修対象となる各機能の重要度」に関して、機能の重要度の精度ならび、システムが重要度を提示することにより、ユーザの機能の重要度の理解度が促進されたかどうかについて行った評価結果について述べる。システムが機能の重要度を提示し、ユーザの機能の重要度の理解が促進された場合、改修難易度・改修率の調査効率およびアイデア間の比較効率の向上が見込まれる。以下、評価1として、まずシステムが算出する機能の重要度が、ベテラン SE の考える機能の重要度とマッチしているかどうかの分析を行った。そして評価2として、実際にシステムが算出する機能の重みを本可視化 UI で被験者に提示する事により、機能の重要度の理解が促進されるかどうか検証するためにケーススタディを行った。

### 5.1 評価1:機能の重要度の精度

システムが算出する機能の重要度の精度を評価するにあたり、ベテラン SE に過去の見積り経験から見積りタスクを計5件作成してもらい、各タスクにおいて改修が発生した機能毎に、重要度のスコアを5段階で付与したデータセットを計108件用意した。

このデータセットと、システムが算出した機能の重要度との相関係数ならび、重要度のスコアを基に重要な機能と非重要な機能の二値分類による線形判別を行い、比較を行った際の適合率、再現率を表1に記す。

表1. 機能の重要度の精度

相関係数	0.65
適合率	100%
再現率	60%

相関係数は0.65であり、システムが算出した機能の重要度と正解スコアとの間に中相関がある事を実証した。また判別分析の結果に関して、適合率は100%であるが、再現率は60%という結果となった。この結果は、システムが重要な機能だと検出した機能に関して誤りは無いが、システムが検出出来ない重要な機能が存在する事を示唆している。実際にFalse Negativeであった機能に関してヒアリングを行ったところ、以下の2つの尺度をシステムは検出出来ない事が判明した。

- ・業務フロー上のデータの流れに基づく重要度
- ・サブシステムの特性にに基づく重要度

これらの課題に関しては、人手で事前にデータをインプットする運用方法、または設計書から自動で抽出する方式の検討を今後行っていく。

## 5.2 評価2:ケーススタディ

可視化UIを用いることにより、機能の重要度の理解が促進されるかどうかを評価するにあたり、可視化UIを用いた場合とそうでない場合とで、改修を行う機能の重要度を概算してもらったケーススタディを行った。タスクは先述の評価1で作成したタスクを使用し、2名の被験者A、Bにそれぞれ可視化UIを使用した場合で2題、使用しない場合で2題の計4題出題を行った。なお、被験者Aはタスクで使用しているパッケージソフトを扱った業務に約1年携わっている若手SE、被験者Bは5年以上携わっている中堅SEである。ケーススタディの結果を以下表2に記す。

表2. ケーススタディの結果

	従来	可視化UI使用時
被験者A	0.46	0.70
被験者B	0.53	0.82

被験者A、被験者B共に正解スコアとの相関係数が開発した可視化UIを提示した場合に0.2~0.3向上

しており、開発した可視化UIを活用する事で機能の重要度の理解が促進される事を実証した。また、実験後に行ったアンケートにおいて、可視化UIを見たことによって、従来気付かなかった重要な機能に気付く事が出来たと回答があった。この事から、本可視化UIは、機能の重要度の理解を促進する効果があるだけでなく、重要な機能の見落としを防ぐ効果がある事も新たに確認された。

## 6. まとめ

本研究では、作業全貌把握作業において、ベテランSEが頭に思い描いている設計情報を可視化するシステムの開発を行った。また、システムが提示する、改修対象となる各機能の重要度の精度ならび、可視化効果について評価を行い、可視化UIを活用する事で機能の重要度の理解が促進される事を実証したが、重要度の算出方法についてはまだ課題がある事を確認した。今後の研究の進め方として、可視化要件(1)について評価指標を設定し、有効性を検証すると共に、評価1で明らかになった現状システムが検出出来ない二つの重要度の尺度に対する改善策の検討を進めて行く。

## 参考文献

- [1] 経済産業省:「中間とりまとめ(案) 我が国産業の強さを活かすIT投資の在り方」, <http://www.meti.go.jp/report/downloadfiles/g70702c01j.pdf>, 2016年2月20日参照
- [2] 経済産業省:「平成25年情報処理実態調査報告書の概要」, [http://www.meti.go.jp/press/2014/05/20140528004/20150528004\\_1.pdf](http://www.meti.go.jp/press/2014/05/20140528004/20150528004_1.pdf), 2016年2月20日参照
- [3] ISO/IEC, ISO/IEC 14143-1: Information technology-Software measurement-Functional size measurement-Part 1: Definition of concepts, ISO/IEC, (1998)
- [4] 中村覚, 大和裕幸, 稗方和夫, 岡田伊策, 齋藤稔, 笈田佳彰: ドメイン知識の記述支援と活用のためのデジタル資産管理システム基盤「KASHIWADE」の開発: 情報システム開発における設計書・設計知識への適用, Vol.29, pp.1-4, (2015)
- [5] 稗方和夫, 大和裕幸, 深田直人, 中村覚, 岡田伊策, 齋藤稔, 笈田佳彰, 渡辺郁雄, 松本滋: システムの仕様変更調査における設計情報を用いた影響分析システムの開発, 設計工学・システム部門講演会講演論文集, Vol.2014, No.24, pp.1305-1-1305-10, (2014)
- [6] Stefan Hachul, Michael Jünger: An experimental comparison of fast algorithms for drawing general large graphs, Graph Drawing 2005, pp.235-250, (2005)