

「見積情報伝達シート」による調査ノウハウの伝達

Communication of Empirical Knowledge by "Estimated Information
Communication Sheet"

主査	清水 吉男 (株式会社システムクリエイツ)
副主査	飯泉 紀子 (株式会社日立ハイテクノロジーズ)
アドバイザー	足立 久美 (株式会社デンソー)
研究員 (リーダー)	村嶋 宏太 (株式会社日立ハイテクソリューションズ)
	秋山 桂樹 (株式会社リンクレア)
	千田 哲義 (NECソフト株式会社)
	富島 鉄矢 (AJS株式会社)
	南部 妙水 (アンリツエンジニアリング株式会社)
	森成 勇一 (アンリツエンジニアリング株式会社)

研究概要

小規模かつ短納期の派生開発が多い組織では、製品知識や経験を持った熟練者が開発案件を短期間に精度良く見積もり、経験の浅い若手技術者が中心になって開発を行っているところが多い。こういった組織では、開発案件の増加に伴い、熟練者がより見積もりに特化することで開発に関与する機会が減少し、見積側と開発側のコミュニケーションの希薄化を引き起こしている。特に、熟練者が見積もった際の調査内容が開発者に十分に伝達されないことによって、変更の影響箇所の特ミスに起因する変更漏れを招いている。

そこで我々は、熟練者が見積もった際の調査内容を効果的に伝達させ、変更漏れ等の不具合を防止することを目的として、見積側と開発側をつなぐ「見積情報伝達シート」を考案した。これを実際のプロジェクトに適用した結果、変更漏れに起因する不具合を減少させることに成功した。なお、このシートによって、熟練者の調査ノウハウが若手開発者に伝承されるといった人材育成上の効果も得ることができた。

Abstract

There are many projects which an experienced engineer estimates development matters precisely and quickly and inexperienced young engineers develop in the many derivative developments which have a lot of small scale and short delivery matters. According to an experienced engineer is dedicated to the estimation, it is decreasing communication of an estimator and developers as the increase in development matters. Especially the information which experienced engineer investigated when estimate is not hand over to the developers sufficiently, this cause to the forgetting changes resulting from mistakes on specific mistakes in an influence part.

Then we established a new estimation format named "Estimation Information Communicate Sheet" to connect estimator and developers for the purpose of communicate information which experienced engineer investigated when estimate with developers efficiently and prevents troubles like forgetting changes. As a result of applying this format to an actual project, it succeeded in decreasing the troubles resulting from the forgetting changes. In addition, the know-how which has experienced engineer's got across to the young developers. The effect of developers growing up was able to be acquired.

1. 研究動機

ソフトウェアの開発案件は、小規模なほど厳しいコストと納期が要求される傾向にあるが、特に派生開発ではその傾向が強い。小規模な開発が多い組織では、売上確保に向けて幾つもの案件を見積もるため、高精度の見積もりを短期間に行う必要がある。そのため、製品知識や経験を持った熟練者が見積もりを行い、熟練者の指示のもと、経験の浅い若手技術者を中心に開発を行う体制が取られることが多い。以前は、見積案件が適度な量であったため、熟練者は十分に若手技術者に指示することができていた。しかし近年、熟練者の受け持つ見積案件が急増したため、若手技術者の育成が間に合わず、熟練者の役割が開発と分離しつつある。この分離に伴い、熟練者と開発メンバーのコミュニケーションも疎になってきている（図 1）。そして、これに起因する不具合が増加している。

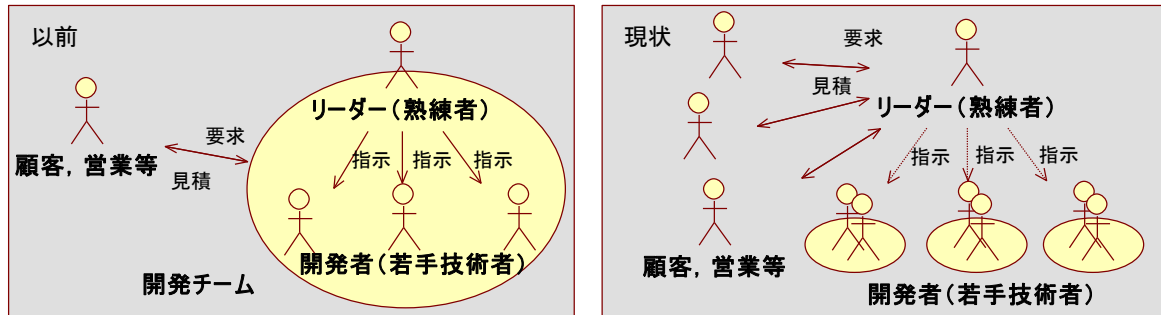


図 1 開発チームの移り変わり

2. 現状分析

熟練者の関与が希薄になったことで発生した不具合の特徴や性質を調べるため、研究メンバーの所属するプロジェクトを対象に聞き取り調査を実施した。その結果、変更仕様に直接関係する箇所の変更漏れではなく、影響箇所の特定漏れによる不具合が目立ってきていることがわかった。影響箇所の漏れは、熟練者でないと検出することが難しい。しかし熟練者が多忙になり、緊急の見積作業や客先対応などでレビューを十分に行えないケースが増えたため、漏れを検出できなくなっているという。

このことを裏付けるために、影響箇所の変更漏れに起因する不具合 16 件のデータを持ち寄り、熟練者がレビューに参加していれば検出できたかを分析した。その結果（図 2）、熟練者がレビューしていれば防げた不具合が 12 件あった。さらに、これら不具合の半数を超える 7 件は、その原因となった影響箇所を、熟練者が見積もり時に発見していたことがわかった。

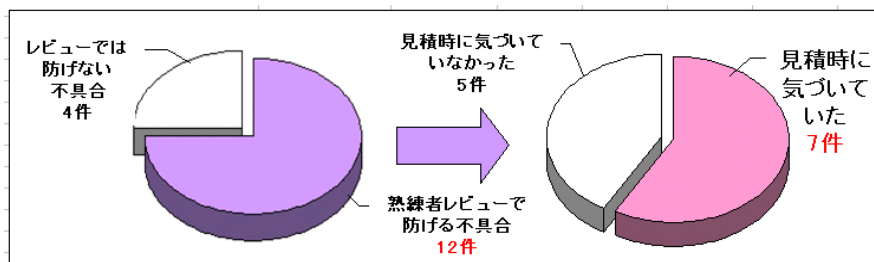


図 2 影響箇所の変更漏れ分析結果

これらの分析結果から、以下の状況によって開発者に伝わらなかったと考えた。

- 熟練者は、若手開発者もわかっているだろうという思い込みと、ここまで書いておけばやってくれるであろうという期待から、情報の伝達を省いてしまうことがある。

- 熟練者は、情報の伝達を試みるときに、何をどこまで記述すれば良いかわからないまま自己の判断で記述内容を決めてしまう。そのため、開発者にとって必要な内容が省略されてしまうことがある。
- 若手の開発者は、伝達されていない事項を表面的にしか理解していないにもかかわらず、これで良いだろうという思い込みで開発を進めてしまう。

上記の原因から、解決すべき課題を下記の2点にフォーカスした。

- (1) 見積もり時のアウトプットを開発のインプットとするためのプロセスがないため、見積もり時の気づきを開発につなげられるようにする。
- (2) 見積もり時に気づいたことを手間をかけずに文書に残せるようにする。

先行研究として関野ら^[1]が考案した、暗黙の仕様や設計理由に関する情報をマイスター情報としてドキュメントに残す方法がある。これは開発中の引継に起因する変更ミスや漏れを防止するもので、XDDPの変更要求仕様書とトレーサビリティマトリクスを活用している。これらの文書は開発者が作成するもので、見積もり時には存在しない。また、詳細まで記述するフォーマットであるため作成に時間がかかり、見積もり時には使用できなかった。

3. 解決策

我々はまず分断されている見積プロセスと開発プロセスをつなげる必要があると考えた。そのためには見積プロセスのアウトプットであり開発プロセスのインプットとなる情報が必要である。そこで見積者が、作業を行いながら開発者に有用な情報をアウトプットできるよう「見積情報伝達シート」を考案した。

これを見積プロセスで作成し、開発プロセスで利用させることで、若手開発者のみでは発見できなかった変更箇所や影響箇所を事前に認識できるようになる。その結果、品質の向上と開発工数の削減（主に変更箇所と影響箇所の調査工数）が可能となる。

3.1 見積もりと開発をつなぐプロセスフロー

定義したプロセスフローを図3に示す。

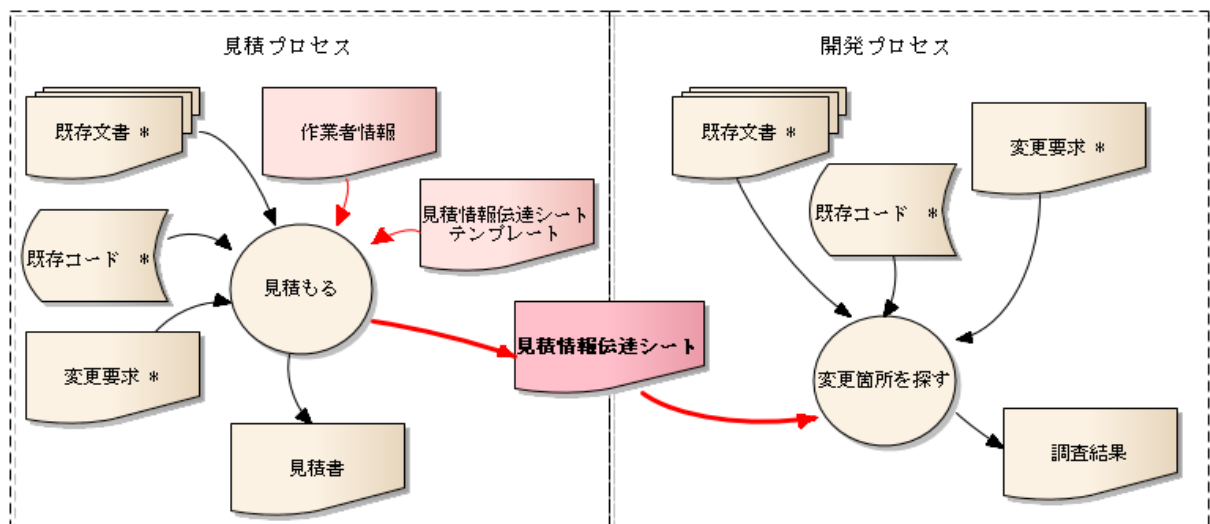


図3 見積もりと開発をつなぐプロセスフロー

見積プロセスのアウトプットとして「見積情報伝達シート」を作成し、開発プロセスのインプットに使用する。

3.2 見積情報伝達シートのポイント

「見積情報伝達シート」を定義するにあたり、意識したポイントは以下の2点である。

- (1) 開発者に変更箇所や影響箇所が伝わること
- (2) 忙しい熟練者でも作成でき、見積者の暗黙知が可視化できること

これらのポイントを満たすフォーマットとして、「見積情報伝達シート」を考案した。「見積情報伝達シート」のフォーマットを図 4 に示す。

(ア)	見積者氏名						
	開発者氏名						
(イ)	変更要求の背景・経緯						
(ウ)	見積情報	規模	要求1	要求2	要求3		
		難易度					
		工数					
	要求に対する 引継ぎ情報（気づき、考慮点）						
	① 見積ベース・前提	仕様・ドキュメント					
		暗黙仕様					
			要求1	要求2	要求3	規模	引継ぎ情報 （気づき、考慮点）
	② 作業・変更箇所	作業					
		変更箇所 （大項目）					
			要求1	要求2	要求3	規模	引継ぎ情報 （気づき、考慮点）
	③ 変更箇所詳細	ファイル等					

図 4 見積情報伝達シート

見積者に開発者を意識したシート作りを促すため（ア）に開発者の氏名を入力欄を設けた。開発者は見積もり時に確定していないこともあるが、開発者が確定していれば見積者はその開発者を意識してその開発者用の情報を記入することができる。

その下にある（イ）要求の背景欄には、必要に応じて要求そのものの解説を記述することができる。

（ウ）調査マトリクスはこのシートのメイン部分であり、見積者は調査を行いながら判明したこと、考えたことなどを記述する。マトリクスは横軸を「変更要求」とし、3つの縦軸①「見積ベース・前提」、②「作業・変更箇所」、③「変更箇所詳細」で分割された入力エリアで構成する。

① [見積ベース・前提]×[変更要求]

見積もりに使用した仕様書等の設計文書、規格・規程等の参照文書、および文書化されていない知識など、仕様を表現する情報を記述する。

② [作業・変更箇所]×[変更要求]

工数見積もりが可能な作業を表現する情報を記述する。①より細かく③より粗い、機能やコンポーネント単位の項目を記述する。

③ [③変更箇所詳細]×[変更要求]

ソースファイルやモジュールなど、②より詳細な変更項目を記述する。

調査結果を記述する入力欄を①、②、③と粒度ごとに分け、プロジェクトごとの異なる状況に対応できるようにした。また、この形式によって、要求に対してどの粒度で何を変更すればよいのか、どのような影響があるのかを一目で把握できる。さらに、見積者の暗黙知を記述できるように、テキストを自由に入力する「引継ぎ情報」欄を設けた。

また見積作業の増加は、見積作成時に調査した内容をシートに書き写すだけであるため、記述する時間で済む。

4. 解決策の検証

聞き取り調査を実施したプロジェクトから3件のプロジェクトを選択し、「見積情報伝達シート」の効果を検証するシミュレーションを行った。

シミュレーションでは、プロジェクト実施時と同じ見積者が「見積情報伝達シート」を使用した見積もりを行い、作成したシートを開発メンバーの1名に渡した。見積者は見積もりに要した工数を計測し、その増分が実業務において許容可能かどうかを判断した。開発メンバーはプロジェクト実施時と比べ、見積者から新たに得られた情報があるかを判断し、予防できた手戻りや不具合の件数、省略できた調査の工数を報告した。また、見積者と開発者の両方に、シートの使い勝手について聞き取り調査を行った。

シミュレーションを実施したプロジェクトを表1に示す。

表1 検証プロジェクト一覧

	プロジェクト1	プロジェクト2	プロジェクト3
開発内容	旅費・経費精算を行うパッケージソフト 顧客向けシステムの機能追加	会計データを共通フォーマットに変換して送信するシステム	特定ソフト専用の入出力データ加工ソフト 連携ソフト追従を目的とした機能追加
プロジェクト規模	約1人月	約0.75人月	約16人月
開発メンバー内の熟練者の割合	0%	33%	0%
開発者の調査規模	約1人日	約3人日	約80人日
見積情報の伝達方法	機能別工数一覧表と口頭の作業指示	機能別工数一覧表	要求と外部規格の対応表
不具合・手戻り原因となった欠落情報	過去に実施した変更の仕様（文書化されておらず調査が困難）	ベースソフトの内部仕様や開発ツールの使い方	外部規格・関連ソフトの仕様に起因する影響箇所
伝達方法の問題点	影響箇所の情報が欠落	開発者のスキルを考慮していない	伝達しにくい書き方

4.1 検証結果（プロジェクト1）

プロジェクト1は、過去に基本となるソフトウェアに軽微なカスタマイズを数回行ったシステムに対する開発プロジェクトである。このプロジェクトでは基本ソフトウェアの設計文書は存在するが、顧客ごとに異なるカスタマイズの仕様は文書化されていなかった。しかも、その内容を把握している熟練者は開発メンバーに作業指示を行うための時間を十分に確保できなかったため、見積者が気づいていたカスタマイズ部分の影響箇所を開発メンバーが発見できず、変更漏れによる不具合が1件発生した。

①見積ベース・前提	仕様・コメント	20**年実施時カスタマイズ設計書	in	
		20**年実施時カスタマイズ設計書	●	
		20**年実施時カスタマイズ設計書	in	
		20**年実施時カスタマイズ設計書	●	
		20**年実施時カスタマイズ設計書	in	
	暗黙の仕様	SQLLOADER設定資料	in/Out	
		既存マスタ取込処理は利用不可	◎	
		リソースファイルの管理方法 リソースエディタによる編集継続	◎	
		ワークフロー制御が既存処理と異なる (20**年実施時カスタマイズ設計書)	◎	
		***連携が既存処理と異なる (20**年実施時カスタマイズ設計書)	◎	

図5 プロジェクト1の「見積情報伝達シート」
①部分の抜粋

シミュレーションを実施したところ、開発者は設計文書に記述されていない情報を「見積情報伝達シート」の①から得ることができるようになり（図 5）、前述の不具合を防止できることがわかった。さらに、開発者が行っていた変更箇所・影響箇所に関する文書を判別するための調査が不要となり、これまでの調査工数と比べ約 10%の工数を削減できることもわかった。

見積もりに要した工数は既存の見積工数から約 10%の工数が増加したが、見積者は以下の理由でこの増分を許容可能とした。

- 既存の見積もりでは気づかなかった影響箇所を調査するため工数が増えたが、本来見積もり時でも必要な調査であったため
- 文書化されていない項目についても記述したことで、開発メンバーによる問合わせ件数が減り、その対応工数が減少することが見込めるため

また、シミュレーションを行った見積者と開発者から、今回作成したシートを基本ソフトウェアが同じ別のプロジェクトに展開し再利用することができるという見解を得た。

4.2 検証結果（プロジェクト 2）

プロジェクト 2 ではベースソフトウェアの知識が乏しく、開発ツールの習熟度も低いメンバーが開発を行った。その上、見積者は別案件を抱えメンバーに詳細な変更箇所を指示できなかったため、開発者は指示のなかった影響箇所を特定することができず、変更漏れによる不具合が 1 件発生した。

シミュレーションを実施したところ、見積者は開発者の氏名からスキルを考慮して変更方法を記述した（図 6）。この記述によって、開発者が影響箇所を特定することが可能となったため、前述の不具合 1 件を防止できることがわかった。さらに、関係のないソースコードの調査に使用した工数を 2 人日から 1 人日に 50%削減できることがわかった。

		要件1	相違	引継ぎ情報 (重点)
③ 変更箇所詳細	ファイナル・モジュール・変更箇所	マスタメンテナンス_A権限登録ソース	小	開発ツールの特性を明記した作業指示 実装できない
		マスタメンテナンス_B権限登録ソース	小	実装できない
		マスタメンテナンス_C権限登録ソース	小	各画面ごとに確認 ツールの特性上、共通で実装できないため、各画面ごとに確認
		マスタメンテナンス_D権限登録ソース	小	ツールの特性上、共通で実装できないため、各画面ごとに確認

図 6 プロジェクト 2 の「見積情報伝達シート」③部分の抜粋

見積もりに要した工数は既存の方法から約 30%増加したが、開発者からの口頭での問い合わせが減り、他の案件の見積もりに専念できることから見積者はこの増分を許容可能とした。

加えて、見積者が開発者の氏名からスキルを判断し、詳細な変更方法をシートに記述することで、開発者が要求からプログラムの変更範囲まで確認することができた。さらに開発者のベースソフトウェアの知識習得にも効果が見られた。

4.3 検証結果（プロジェクト 3）

プロジェクト 3 ではプロジェクト実施時の見積者が不在であったため、開発メンバーの 1 人にシミュレーションを依頼した。開発メンバーの知識不足は、当時の見積者が残した調査メモを参照しながらシミュレーションを実施することで補完した。

変更箇所、影響箇所の調査には、外部規格や関連ソフトウェアに精通している必要があった。見積者が作成した調査メモには、変更箇所を特定するために使用した関連文書やキーワードが残っていたが、要求ごとに列挙されているだけだったため、そこから見積者が発見した変更箇所を開発メンバーが読み取ることができなかった。また、調査メモには影響箇所や脚注を記述できず、見積者は推測を残さなかった。

このため、開発メンバーが影響箇所を見逃し、約 10 人日の手戻りが発生した。

シミュレーションを実施したところ、「見積情報伝達シート」の外部文書を記述する①と変更箇所を記述する③を使って、外部文書の章番号と変更箇所を対応させることができた。この結果、開発メンバーが対応づけ作業に要した工数（調査工数の約10%）を削減できることがわかった。また、類似関数等の推測した影響箇所についても記述があった(図7)。これによって、前述の手戻りが防止できることがわかった。

		追加機能1		追加機能2		追加機能3	
		機能1-4					
③ 変更箇所詳細	ファイル 変更箇所 ・ モジュール	関数B					
		関数C					
		関数D			状態1との遷移は大丈夫か??		
		関数E			状態2との遷移は大丈夫か??		
		関数F			V		
		関数G			V		
		ファイル4	データD更新				
	関数D				関数AとDは?		
	関数E				V		
	関数F				V		
	関数G						
	ファイル5						
	関数A						
	関数B						
	関数C						

図7 プロジェクト3の「見積情報伝達シート」③部分の抜粋

見積もりに要した工数の増分は既存の方法と比べ10%未満であることから、見積者はこの増分を許容可能とした。

さらに開発メンバーから、すべての要求が1つのマトリクスに集約されたことで、調査の注力ポイントや重複箇所を事前に把握可能となり、調査効率が向上するという見解を得た。

4.4 検証結果のまとめ

見積者と開発者から、既存の情報伝達方法との違いを評価するため、「見積情報伝達シート」の効果について聞き取り調査を行った(表2)。シミュレーションに参加した見積者と開発者の双方が、作成したシートによって開発に必要な情報をより多く伝達することができたと回答した。また、このシートのフォーマットを使うことで、見積者は記述方法に、開発者は読み方に悩む必要がなくなり、伝達の効率が上がるという回答もあった。

表2 「見積情報伝達シート」の聞き取り調査の結果

	プロジェクト1		プロジェクト2		プロジェクト3	
	見積者	開発者	見積者	開発者	見積者	開発者
変更箇所・影響箇所の書きやすさ/読みやすさ	○	○	○	○	○	○
上記以外の情報の書きやすさ/読みやすさ	○	○	○	○	-	-
1件あたりの書やすさ/読みやすさ	-	-	○	○	×	-
全体の書きやすさ/読みやすさ	○	○	○	○	○	○
その他の効果	見積者の見積効率・精度の向上		熟練者不在でもOJT効果あり		要求の追加削除への対応が容易	

○: 改善あり, -: 変化なし, ×: 改悪あり

プロジェクト3の調査メモには主に関連文書等の、変更箇所・影響箇所ではない情報が記述されていたため、本シートを使うことによる情報量の増加は見られなかった。また、これらの情報が文章で記述されていたため、見積者は記述スペースが狭く書きにくいという評価を出した。一方で、開発者は列幅の調整で十分に対応可能であり、読みやすさに変化はないと評価した。

4.5 考察

検証結果から「見積情報伝達シート」を見積プロセスの成果物に追加することで、以下の効果があることがわかった。

- 成果物として定義することで、見積者と開発者の両者に情報伝達の必要性を認識させる。見積者は開発者に伝える前提で調査を行うため、開発者に必要な情報が残りやすくなる。
- 定型化されることで、見積者は記述方法を検討する手間が削減され、記述すべき情報に集中できる。開発者は読み方に慣れるため、情報に焦点を充てることができる。
- 開発者名を入力することで、スキルを想定した記述が促されるため、情報が伝達されやすくなる。
- 熟練者の知識がアウトプットされることで開発者からの問い合わせが減少し、熟練者の負担が軽減される。

今回のシミュレーションの結果、見積者、開発者双方に、有効性が認められたことから、組織にプロセスを導入する際の事例として活用できる。

さらに、「見積情報伝達シート」には以下に示す副次的効果も確認できた。

- 見積根拠が残るため、再見積もりの際に再利用できるようになり、見積工数を削減することができる。
- 見積もり妥当性の検証にも活用できるようになる。

5. 本研究のまとめ

5.1 研究成果

これまで、影響箇所の変更漏れによる不具合や設計のやり直しが多いという問題に悩まされていた。不具合の原因を調査し、見積もり作業をプロセスとして見直したところ、見積もりと開発のプロセスが分断されていることに気づいた。そこで、「見積情報伝達シート」を考案し、見積もりと開発のプロセスをつなげることが可能か、シミュレーションによって検証した。この結果、見積もり時に発見された影響箇所の変更漏れは予防できることがわかった。

5.2 今後の進め方

現状分析の結果から、見積もり時に発見できない影響箇所の変更漏れもあることがわかった。これらによる不具合・手戻を防ぐためには様々な施策を積み重ねて改善する必要がある^[2]。

その1つとして、若手開発者のスキルアップを促進し、調査やレビューで影響箇所を熟練者と同じように発見できるようにすることがある。熟練者の知識・経験を若手開発者に伝承する効果が「見積情報伝達シート」にあると考えているが、その検証には至っていない。今後は、本シートをどのように活用すれば、若手開発者のスキルアップを効果的に促進できるかについても検討したい。

6. 参考文献

- [1] 関野浩之, 大坪智治, 大内智之, “後任担当者視点を取り入れた設計背景の形式知化による派生開発の品質向上策,” 日本科学技術連盟第27年度ソフトウェア品質管理研究会成果報告集, pp. 140-150, 2011.
- [2] 独立行政法人情報処理推進機構(IPA) ソフトウェア・エンジニアリング・センター(SEC), “続 定量的品質予測のススメ(SEC BOOKS)”, p25, 2011
- [3] 清水吉男, “「派生開発」を成功させるプロセス改善の技術と極意”, 技術評論社, 2007

付録1 見積情報伝達シート テンプレート

見積者氏名							
開発者氏名							
変更要求の背景・経緯							
		要求1	要求2	要求3			
見積情報	規模						
	難易度						
	工数						
要求に対する 引継ぎ情報（気づき、考慮点）							
① 見積ベース	キョウ ト	仕様・ド					
	ル ル	暗黙仕様					
		要求1	要求2	要求3	規模	引継ぎ情報 （気づき、考慮点）	
② 箇所 変更	作 業						
	（大 項目）	変更箇所					
		要求1	要求2	要求3	規模	引継ぎ情報 （気づき、考慮点）	
③ 詳細 変更箇所	フ ア イ ル 等						

付録2 シミュレーションで作成した見積情報伝達シート

見積者氏名		*****				
開発者氏名		*****				
変更要求の背景・経緯						
△△マスタ取込テーブル定義、取込ファイル定義変更にもなる取込処理の変更						
△△マスタ取込変更						
見積情報		規模				
		難易度				
		工数				
各要求に対する 引継ぎ情報(気付き、考慮点)			既存プログラムとは別の処理 (SQLLoader)を利用しているので注意すること			
① 見積 ベース ・ 前提	キ ュ メ ン ト	20**年実施時カスタマイズ設計書	in			
		20**年実施時カスタマイズ設計書	●			
		20**年実施時カスタマイズ設計書	in			
		20**年実施時カスタマイズ設計書	●			
		20**年実施時カスタマイズ設計書	in			
	暗 黙 の 仕 様	SQLLOADER設定資料	in/Out			
		日本語環境 利用	◎			
		英語環境 利用				
		利用ユーザ数 (100ユーザ未満)				
		利用ユーザ数 (500ユーザ未満)				
		利用ユーザ数 (1000ユーザ未満)				
		利用ユーザ数 (1000ユーザ以上)	◎			
		仕訳データ出力処理あり	◎			
		既存マスタ取込処理は利用不可	◎			
		リソースファイルの管理方法が異なる リソースエディタによる編集継続中	◎			
ワークフロー制御が既存処理と異なる (20**年実施時カスタマイズ設計書)	◎					
チェック連携が既存処理と異なる (20**年実施時カスタマイズ設計書)	◎					
納品時に****サービスは手動設定すること	◎					
企業A 企業Bの企業は設定内容が異なる	◎					
見積ベース前提						
変更箇所・作業			△△マスタ取込変更	規模	引継ぎ情報(気付き、考慮点)	
② 作業 ・ 変更 箇所	作 業	WEB画面 追加/変更	●		新規追加	
		バッチプログラム 追加/変更	◎	2	変更 対象はSQLLoader設定部	
		DB定義 追加/変更	◎	1	変更	
		リソース(文言)追加	◎	0.5	リソースファイルの管理方法が異なる リソースエディタによる編集継続中	
		バッチプログラム 追加/変更				
	大 目 的	Loader****.Config	●		不要	
		マスタ取込サポートクラス	●		不要	
		SQLLOADER設定	◎	2	既存のマスタ取込処理は利用不可 固定長データの取込	
		DB定義				
		Business****.Manager.config	◎	0.25		
		Entity****.Manager.config	◎	0.25		
		Create Table	◎	0.25	再定義	
		自動生成DTOプログラム	◎	0.25		
		バッチプログラム実行設定データ追加	●		不要	
		マスタ取込処理設定データ追加	●		不要	
リソース(文言)追加						
Resource****.Config	◎	0.25				
自動生成リソースファイル	◎	0.25	リソースの自動生成はエディタを介しているため 通常の自動生成は不可。詳細は別途資料確認のこと			
△△マスタ取込変更						
変更箇所詳細			規模		引継ぎ情報(気付き、考慮点)	
③ 変 更 箇 所 詳 細	フ ァ イ ル 変 更 箇 所 ・ モ ジュール					

図 8 プロジェクト1で作成した見積情報伝達シート

見積者氏名		*****		
開発者氏名		*****		
変更要求の背景・経緯				
現状では、1ユーザにつき1つの権限しか保持できないが、業務的には、1ユーザが2つ以上の権限を有するため、権限定時のチェックロジックを変更する必要がある。				
		要件1		
見積情報	規模			
	難易度 工数	低		
各要求に対する 引継ぎ情報(気づき、考慮点)		データを参照する画面にて、操作しているユーザの権限により、データの表示を制御しているため、一覧系の画面への考慮が必要。		
① 見積ベース・前提	仕様・ドキュメント	マスタメンテナンス_A権限登録_画面設計書	in/out	
		マスタメンテナンス_B権限登録_画面設計書	in/out	
		マスタメンテナンス_C権限登録_画面設計書	in/out	
		マスタメンテナンス_D権限登録_画面設計書	in/out	
		E機能<><<一覧_画面設計書	in/out	
	F機能<><<一覧_画面設計書	in/out		
	G機能<><<一覧_画面設計書	in/out		
	要件定義書_権限設計	in/out		
	暗黙の仕様	社員マスタに登録されている社員のみ使用可	in	
		登録権限と承認権限を兼ねても、自身の登録したデータは承認不可	in	
上位権限の方が優先になる		in		
見積ベース・前提		要件1	規模	引継ぎ情報(気づき、考慮点)
変更箇所・作業		要件1	規模	引継ぎ情報(気づき、考慮点)
② 作業・変更箇所	作業	一覧画面表示表のview修正	■	
	(大項目) 変更箇所	マスタメンテナンス_A権限登録	◎	
		マスタメンテナンス_B権限登録	◎	
		マスタメンテナンス_C権限登録	◎	
		マスタメンテナンス_D権限登録	◎	
E機能<><<一覧	■			
F機能<><<一覧	■			
G機能<><<一覧	■			
見積もりベース・前提		要件1	規模	引継ぎ情報(気づき、考慮点)
③ 変更箇所詳細	ファイル 変更箇所 モジュール スケジュール	マスタメンテナンス_A権限登録ソース	小	ツールの特性上、共通で実装できないため、各画面ごとに確認
		マスタメンテナンス_B権限登録ソース	小	ツールの特性上、共通で実装できないため、各画面ごとに確認
		マスタメンテナンス_C権限登録ソース	小	ツールの特性上、共通で実装できないため、各画面ごとに確認
		マスタメンテナンス_D権限登録ソース	小	ツールの特性上、共通で実装できないため、各画面ごとに確認
		E機能<><<一覧	テスト	プログラム修正はないが、表示されるデータが変更されるためテストのみ実施
		F機能<><<一覧	テスト	プログラム修正はないが、表示されるデータが変更されるためテストのみ実施
		G機能<><<一覧	テスト	プログラム修正はないが、表示されるデータが変更されるためテストのみ実施
		画面表示用view1	中	1ユーザに複数の権限を持たせるとデータを権限分表示してしまうので、権限分表示しない様修正
		画面表示用view2	中	1ユーザに複数の権限を持たせるとデータを権限分表示してしまうので、権限分表示しない様修正

図 9 プロジェクト2で作成した見積情報伝達シート

		追加機能1		変		
		機能1-1	機能1-4			
見積情報	規模	1	1			
	難易度	2	1			
	リスク分計数	2	1			
	見積 (week)	2	2			
	見積入月	0.6	0.55	1.4		
各要求に対する 引継ぎ情報(気づき、考慮点)						
①見積 ベース・ 前提	仕様・ ドキュメント	外部仕様書1	7.7.2	7.7.2		
		外部仕様書2				
		外部仕様書3	v	v		
		外部仕様書4	v	v		
		外部仕様書7				
		外部仕様書8				
		外部仕様書9				
		外部仕様書10				
		外部仕様書11	7.2.1, 7.2.2	7.2.7	7	
		暗黙の仕様 ・ルール	参考資料1	v	v	
			参考資料2	v	v	
補足	動作実績等	あり	環境のみ	いまは		
見積ベース・前提		追加機能1				

図 10 プロジェクト3で作成した見積情報伝達シート (①部分)

		のみ	環境のみ	いまは	
		追加機能1		変	
		機能1-1	機能1-4		
補足					
動作実績等					
見積ベース・前提		追加機能1			
変更箇所・作業					
作業	テスト設計/仕様update				
	テスト 総合テスト物品手配 総合テスト				
	取扱説明書更新				
	インストラ作成				
	引継ぎ情報(気づき、考慮点)		設定追加	メモ1	メモ
②作業・ 変更箇所	シーケンス	シーケンス1	v		
		シーケンス2	v		
		シーケンス3	v		
		シーケンス8	v		
		シーケンス9		v	
		シーケンス10		v	
		シーケンス11			
		シーケンス12			
		パラメータ (入力)	パラメータ1	v	
			パラメータ2	v	
			パラメータ3		
			パラメータ20		
パラメータ21					
パラメータ (出力)	パラメータA	メモリ1, メモリ2 (条件A)			
	パラメータB				
	パラメータC		メモリ3		
	パラメータD				
	パラメータE				
見積ベース・前提		追加機能1			

図 11 プロジェクト3で作成した見積情報伝達シート (②部分)

パラメータ1		見積もリベース・前提		追加機能1		変
変更箇所・作業		機能1-1	機能1-4			
③ 変更箇所詳細	ファイル・変更箇所 モジュール	ファイル1				
		関数A		シーケンス9対応追加		
		関数B		パラメータC対応追加		
		ファイル2	パラメータA解析追加			
		関数A	呼び出し追加			
		関数B	呼び出し追加			
		関数C				
		関数D				
		ファイル3				
		関数A				
		関数B				
		関数C				
		関数D				
		関数E				
		関数F				
		関数G				
		ファイル4				データ
		関数A	影響しないこと			
		関数B	影響しないこと			
		関数C	影響しないこと			
		関数D	変更			
		関数E				
		関数F				
		関数G				
		ファイル5	データA更新			データ
		License check		√	√	

図 12 プロジェクト3で作成した見積情報伝達シート (③部分)