

顧客との合意後の仕様齟齬検出意義と メトリクスについて

～仕様齟齬の予兆検出および最適な対処方法選択の提案～

一般財団法人日本科学技術連盟
第32年度ソフトウェア品質管理研究会
成果発表会

第1分科会 ソフトウェアプロセス評価・改善 チョコザイルチーム

主査 : 三浦 邦彦 (矢崎総業株式会社)
副主査 : 中森 博晃 (パナソニック ファクトリーソリューションズ株式会社)
: 山田 淳 (東芝ソフトウェア・コンサルティング株式会社)
リーダー : 山本 久代 (サントリーシステムテクノロジー株式会社)
研究員 : 市川 孝裕 (株式会社インテリジェンス ビジネスソリューションズ)
: 海野 和由 (矢崎総業株式会社)
: 小林 道央 (株式会社インテック) ※発表者

アジェンダ

- 背景
 - 仕様齧齧について
 - 仕様齧齧とウォーターフォール型開発
- 提案手法
- 検証
- 今後の展望

仕様齟齬について

仕様齟齬の影響

➤ 成果物の作り直し



➤ コスト増



➤ スケジュール遅延



➤ 顧客信用の低下



OH NO...



重大な影響

仕様齟齬経験の割合

頻度高く聞く言葉だが、本研究実施のアンケートによると4社43人中、36人(約83.7%)が仕様齟齬を経験していると回答。

仕様齟齬とウォーターフォール型開発

開発を複数の工程で区切り、順番に進めていく。仕様確定は上流工程で行い、後の工程では仕様齟齬がないことを前提で進めるため、仕様齟齬の発覚が受け入れ検査時になる。

アジャイルなど仕様齟齬対策へ有効な開発プロセスもあるが、契約上の問題や過去の実績からウォーターフォール型開発を採用することがある。



ウォーターフォール型開発の下流工程においても、仕様齟齬に対応する手法が必要。

仕様齟齬とウォーターフォール型開発 補足

ウォーターフォール型開発の下流工程においても、仕様齟齬に対応する手法が必要。

□「ウォーターフォール型開発はアジャイルなど他の開発プロセスより優れている」という主張ではない

→ ウォーターフォール型開発を採用するプロジェクトは今後も相当数ある。ウォーターフォール型開発向けの手法を研究する必要がある。

□「仕様齟齬に上流工程で予防する必要はない」という主張ではない

→ 上流工程で予防するのが前提であるが、完璧な予防はなく、下流工程でも対応する必要がある

提案手法



予兆を監視

定期的な
モニタリング
で、仕様
齟齬の予
兆を検知
を判断す
る。

**仕様齟齬の予兆を
発見！**

→ **プロジェクトに適した
対処パターンで対策**

**本研究で提案する
プロセス**

**開発の特性・メトリクスを監視
顧客との仕様齟齬をできるだけ早期に発見する！**

プロジェクトが最も重視する観点（QCD等）で発見後の対処パターンで対策



提案手法：監視方法

「開発の特性」と「メトリクス」で予兆を監視

開発の特性 (製品・顧客・開発チームに分類)

対象	開発の特性
製品	製品の新規性 (新規／保守／継続)
顧客	開発内容の理解度
開発	経験・実績

開発の特定に応じて
監視するメトリクスの
閾値が変動



メトリクス

メトリクス
仕様変更回数 (多さ)
プロジェクト全体での仕様変更回数 (多さ)
開発組織内レビューまたは顧客レビューの工数・時間 (少なさ、短さ)



プロジェクトX (新規)

開発特性により重要視するメトリクス変動

提案手法：監視方法の「開発特性」・「メトリクス」詳細

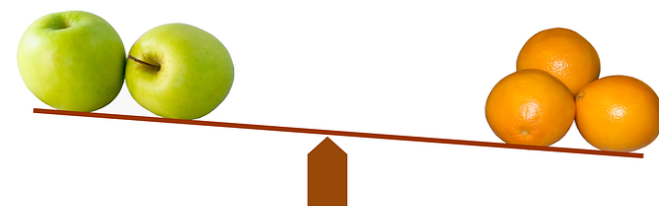
「開発の特性」 14項目定義

「メトリクス」 9項目定義

目的	ID	対象	開発の特性	ID	メトリクス
予兆を判断するため	C1	製品	製品の新規性（新規／保守／継続）	M2	仕様変更回数（多さ）
	C7	顧客	開発内容の理解度	M9	プロジェクト全体での仕様変更回数（多さ）
	C9	顧客	開発途中での担当者の交代の有無	M4	開発組織内レビューまたは顧客レビューの工数・時間（少なさ、短さ）
	C11	開発チーム	経験・実績	M5	Q&Aの数（少なさ）
	C3	製品	顧客内で仕様についての意見が分かっていた機能があ	M1	関連するQ&Aの数（多さ）
	C4	顧客	経験・実績	M6	前の工程の内容だったQ&Aの数（多さ）
	C13	開発チーム	業界知識・製品ドメイン知識	M7	Q&Aの数の変化（増加量大）
	C6	顧客	性格（意見をコロコロ変える等）		仕様の確定が遅れた機能が（有無）
					Q&Aの数に対する仕様変更回数（少なさ）

「具体的な活用例」

		機能A	機能B	機能C	...
開発の特性	製品	特性1	機能毎に、 「開発の特性」と「メトリクス」にて、 仕様齟齬の予兆を総合的に判断したい		
		特性2			
		...			
	顧客	特性1			
		特性2			
		...			
	開発チーム	特性1			
		特性2			
		...			
メトリクス		メトリクス1			
		メトリクス2			
		...			
仕様齟齬の予兆の判定		無し	有り	無し	...



開発特性は定性的に1回で取得、メトリクスは定量的に定期的に取得

提案手法：予兆への対策

予兆
検知

本当？に仕様齟齬かどうかを確認するための
5つの方法を整理

対処パターン

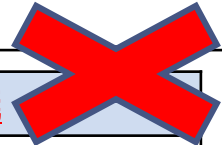
プロトタイプやテストシナリオを作成して顧客に確認する

気になる機能のみをスケジュール前倒しでプロトタイプ（育てて最終的に使うもの）を作成し顧客にデモ実施

気になる機能のみをスケジュール前倒しでモックアップ（最終的に捨てるもの）を作成し顧客にデモ実施

気になる機能のみをスケジュール前倒しで実装（完成版）をおこない顧客にデモ実施

顧客に詳細設計書を再度レビューしてもらう



対処パターンと判断基準の組合せる



対象プロジェクトに適した
対処パターンを提案！



分類	判断基準 9項目定義
D:納期	顧客が確認できる状態になるまでの期間の短さ
C:コスト	仕様齟齬が見つかった場合の無駄の少なさ
変更難度	顧客との調整（協力の引き出し易さ、顧客の時間の融通の）のし易さ
Q:品質	齟齬の発見のしやすさ =「顧客に見せるモノ」の精度の高さ

分類	判断基準	対処パターン1	対処パターン2	対処パターン3	対処パターン4	対処パターン5
D:納期	顧客が確認できる状態になるまでの期間の短さ	—	A	D	C	B

判断基準に対する対処パターンの優先順位（A：最も高い→D：最も低い）⁹

提案手法のアンケートによる検証

本研究で提案する手法

定期的なモニタリング
で予兆を判断

予兆を検知したら、
状況確認

対処パターンを決定

対処パターンを実行

No.	対象	開発の特性	用途
C1	製品	製品の革新性 (新規/保守/継続)	予兆の判断材料として
C2		技術の革新性 (既存技術のみ/新規技術的)	予兆の判断材料として
C3		顧客内代性機、についての意見が分かれている機能があるか?	予兆を感知した後、確認の優先順位を決めるため
C4		顧客/市場	予兆の判断材料として
C5	顧客	顧客の知識、製品ドメイン知識	予兆の判断材料として
C6		開発途中での担当者間の交代	予兆の判断材料として
C7		以下は、予兆を感知した後のもっと詳細に利用	
C8		コア機能はどうか	予兆を感知した後、確認の優先順位を決めるため
C9	製品	仕様が複雑な機能はどうか	予兆を感知した後、確認の優先順位を決めるため
C10		顧客分析、業種について名詞/フレーズはどうか	予兆を感知した後、確認の優先順位を決めるため
C11			
C12	開発チーム		
C13			
C14			

開発の特性

No.	測定単位	メトリクス	予兆検知/判断する手法
M1	製品	前回の工場の稼働率に比べての差 (多)	前回の工場の稼働率/ (仕様変更のペース、または現場の稼働) が大きい
M2		仕様変更回数 (多)	仕様変更回数/ (仕様変更のペース、または現場の稼働) が大きい
M3		顧客の問い合わせ (増加/減少)	顧客の問い合わせ/ (仕様変更のペース、または現場の稼働) が大きい
M4	予兆検知の単位		
M5			
M6			
M7	製品		
M8			
M9			

メトリクス

No.	分類	対処パターンと判断基準、その優先順位
C12	分機	
12	分機	
13	分機	
14	分機	
15	分機	
16	分機	
17	分機	
18	分機	
19	分機	
C12	分機	

**対処パターン
と判断基準、
その優先順位**

アンケートでの検証
(4社43人)

・有効性

・有効性
・実現可能性

・有効性
・実現可能性

注) 特性に応じて監視するメトリクスの決定、監視開始タイミングの決定、齟齬の予兆が検出されたときに何を優先するか、仕様齟齬を疑うメトリクスの閾値の決定、この手法を使う場合のコスト、仕様齟齬の予兆をみる単位の決定については範囲外

アンケートの実施方法

アンケートの回答

開発の特性

メトリクス

対処パターンと
判断基準、
その優先順位



【回答様式】

	有効性	実現性	...
選択肢	1)非常に有効である	1)容易に実現できる	...
	2)ある程度有効	2)実現できる	...
	3)あまり有効でない	3)実現が困難	...
	4)全く有効でない	4)実現不可能	...



+ 自由記述欄

分析方法

1. 選択肢のうち肯定的な1)または2)に回答した人の割合を好感度とし、好感度が70%以上のものを効果的であると判断
2. 好感度が70%以上の回答者の属性（開発経験、マネジメント経験、開発プロセス経験等）や、自由記述欄を考慮し、条件付きで効果的であると判断できるか分析

開発の特性とメトリクスの検証結果

目的	ID	対象	開発の特性	好感度	目的別の好感度順位
予兆を判断するため	C1	製品	技術の新規性（新規技術のみ／新規技術あり）	97.7%	1
	C7	顧客	開発内容の理解度	97.7%	1
	C9	顧客	開発途中での担当者の交代の有無	97.7%	1
	C11	開発チーム	経験・実績	97.7%	1
	C3	製品	顧客内で仕様についての意見が分かっていた機能があるか	95.3%	5
	C4	顧客	経験・実績	95.3%	5
	C13	開発チーム	業界知識・製品ドメイン知識	95.3%	5
	C6	顧客	性格（意見をコロコロ変える等）	90.7%	8
	C8	顧客	開発への関与度、協力度	90.7%	8
	C10	顧客	窓口は一本化されているか	90.7%	8
	C5	顧客	新規顧客／継続顧客	88.4%	11
	C12	開発チーム	性格（先入観が強い、慎重等）	88.4%	11
	C14	開発チーム	開発途中での担当者の交代	88.4%	11
	C2	製品	技術の新規性（既存技術のみ／新規技術あり）	86.0%	14
予兆検知後に、状況確認の優先順位を決めるため	C16	製品	仕様が複雑な機能はどれか	100.0%	1
	C15	製品	コア機能はどれか	97.6%	2
	C17	製品	顧客が特に気になっている機能はどれか	97.6%	2
予兆検知後に、対応パターンを決めるため	C7	顧客	開発内容の理解度	97.7%	1
	C8	顧客	開発への関与度、協力度	90.7%	8



 : 好感度70%以上
 : 好感度70%未満

開発の特性
 提案した17項目中
 17項目で効果があると判断した。

ID	測定単位	メトリクス	有効性に対する好感度	実現性に対する好感度	好感度順位
M2	予兆検知の単位	仕様変更回数（多さ）	97.7%	97.6%	1
M9	プロジェクト全体	プロジェクト全体での仕様変更回数（多さ）	97.7%	95.2%	2
M4	予兆検知の単位	開発組織内レビューまたは顧客レビューの工数・時間（少なさ、短さ）	88.4%	88.4%	3
M5	予兆検知の単位	Q&Aの数（少なさ）	83.7%	85.7%	4
M1	予兆検知の単位	関連するQ&Aの数（多さ）	83.7%	85.7%	4
M6	予兆検知の単位	前の工程の内容だったQ&Aの数（多さ）	81.4%	85.4%	6
M7	予兆検知の単位	Q&Aの数の変化（増加量大）	74.4%	83.3%	7
M8	予兆検知の単位	O&Aの数に対する仕様変更回数（少なさ）	69.0%	71.4%	9

メトリクス
 提案した9項目中
 8項目で効果があると判断した。

対処パターンの検証結果



 : 好感度70%以上
 : 好感度70%未満

ID	対処パターン	有効性の 好感度	実現性の 好感度
P2	テストケースやテストシナリオを作成して顧客に確認する	93.0%	88.4%
P4	気になる機能のみをスケジュール前倒しでプロトタイプ（育てて最終的に使うもの）作成し顧客にデモ実施	97.7%	67.4%
P5	気になる機能のみをスケジュール前倒しでモックアップ（最終的に捨てるもの）を作成し顧客にデモ実施	90.7%	65.1%
P3	気になる機能のみをスケジュール前倒しで実装（完成版）をおこない顧客にデモ実施	86.0%	46.5%
P1	顧客に詳細設計書を再度レビューしてもらう	41.9%	39.5%

実現性が低い理由をコメントから分析したところ技術面、コスト面、顧客との調整面が原因と考えられる。以下の対応をして、実現性を向上できる。

- 技術面：人材育成、開発ガイドの整備
- コスト面：組織的な対応力
- 顧客との調整面：顧客との信頼関係構築、柔軟な契約

判断基準の検証結果



 : 好感度70%以上
 : 好感度70%未満

ID	判断基準	好感度	高感度 順位
J3	顧客が確認できる状態になるまでの期間の短さ	100.0%	1
J7	他の工程、全体スケジュールに与える影響度の低さ	95.3%	2
J8	顧客との調整（協力の引き出し易さ、顧客の時間の融通の）のし易さ	95.3%	2
J9	仕様齟齬が見つかった場合の無駄の少なさ	93.0%	4
J2	齟齬の発見のしやすさ＝「顧客に見せるモノ」の精度の高さ	92.9%	5
J1	仕様齟齬を確認するためのモノを作る作業のコストの安さ	90.7%	6
J4	顧客側にかかる作業負荷の軽さ	90.7%	6
J5	開発側に必要な技術、ノウハウの難度の低さ	88.4%	8
J6	ウォーターフォール型開発からの移行のし易さ	67.4%	9

判断基準

提案した9項目中8項目で
効果があると判断した。

優先順位の検証結果

 : 好感度70%以上
 : 好感度70%未満

表中の英字表記は、判断基準に対する各対処パターンの優位順（A:最も良い→D:最も悪い）を示す。

ID	判断基準	対処パターン					優位順 (A~D) に対する 好感度	高感 度 順位
		P1	P2	P3	P4	P5		
		詳細設計書を再度レビュー	テストケースやテストシナリオを作成して確認	スケジュール前倒しで実装(完成版)	プロトタイプ(育てて最終的に使うもの)作成	モックアップ(最終的に捨てるもの)作成		
J3	顧客が確認できる状態になるまでの期間の短さ	—	A	D	C	B	90.7%	1
J9	仕様齟齬が見つかった場合の無駄の少なさ	—	A	D	C	B	90.7%	1
J8	顧客との調整(協力の引き出し易さ、顧客の時間の融通の)のし易さ	D	A	A	B	A	81.4%	3
J7	他の工程、全体スケジュールに与える影響度の低さ	—	A	C	D	B	81.0%	4
J1	仕様齟齬を確認するためのモノを作る作業のコストの安さ	—					79.1%	5
J2	齟齬の発見のしやすさ =「顧客に見せるモノ」の精度の高さ	D					78.6%	6
J4	顧客側にかかる作業負担の軽さ	—					76.7%	7
J5	開発側に必要な技術・ノウハウ	—	A	A	D	A	72.1%	8
J6	ウォーターフォール型開発からの移行のし易さ	—	A	A	D	A	66.7%	159

優先順位
 提案した9項目中8項目で
 効果があると判断した。

検証結果振り返り

対処パターンの実現性を除いて、おおむね肯定的な結果

実プロジェクトでの試行・検証の準備として、95%以上の好感度がある特性を特に有望だと判断し、それについて、監視開始タイミングと特に関連のあるメトリクスを整理した。

開発特性

メトリクス

推奨 順位	監視開始 タイミング	開発の特性				メトリクス	
		ID	対象	特性名	リスク 要因	ID	メトリクス名
1	要件定義	C1	製品	製品の新規性（新規／保守／継続）	新規	M4	開発組織内レビューまたは顧客レビューの工数・時間（少なさ、短さ）
						M5	Q&Aの数（少なさ）

監視開始工程

今後の展望、課題

今後の展望

下流工程において仕様齟齬に対応する手法の意義について、アンケートでは全ての回答者が肯定的に評価した。この方向性での研究を継続する価値がある。

課題

- 実プロジェクトでの試行・検証
- 対処パターンの実現性の向上
- メトリクスの閾値の決定方法
- 「開発の特性」と「メトリクス」の効果的な組み合わせ

**以上で発表終わります。
ご清聴、ありがとうございました！**