

## 短納期開発現場への XDDP 導入手法

### The method of introducing XDDP

### to the quick delivery development

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): 森 哲史(三菱プレシジョン株式会社)  
研究員 : 南迫 祐樹(富士ゼロックスアドバンステクノロジー株式会社)

#### 研究概要

短納期開発現場では、工期の短さが原因で変更設計や影響範囲の特定を十分に実施できないことが多い。その結果、不具合が発生し、さらに工期を圧迫している。

この状況を打開する方法として XDDP(eXtreme Derivative Development Process)が提唱されているが、現場では XDDP の教育にかかる時間や XDDP 導入による作業時間の増加が心理的な障壁となり、導入に踏み切れない。

そこで我々は、現場の既存成果物から XDDP の成果物サンプルを作成できる手法「REX-P」を考案し、このサンプルを使った説明によって、教育時間に対する不安の解消を図った。また、現場の作業を XDDP の作業に置き換えた際の時間の減少効果を数値で示すことのできる「ながら作業置換表」を考案し、作業時間の増加に対する不安の解消を図った。

今回考案した手法によって、現場に馴染みのある事例のサンプルを使って XDDP を説明できるようになり、導入によるメリットを数値で具体的に示せるようになった。その結果、短納期開発現場の心理的障壁を解消でき、XDDP を試行導入することに成功した。

#### Abstract

At the quick delivery development, change design and specific of effect range couldn't be often executed sufficiently for short construction period. As a result, construction schedule was compressed by defects occur.

As the way to overcome this situation, XDDP(eXtreme Derivative Development Process) has been proposed, but they couldn't try to introduce it because time that takes for education of XDDP and increase of working time for introduction of XDDP is psychological barrier.

Therefore, we devised "REX-P" method that can create sample of artifacts of XDDP based on existing artifacts at the development and tried to eliminate concerns of education time by descriptions using sample. In addition, we tried to eliminate concerns of increase of working time by devising "Nagara Work Replacement Table" can shown numerically that time is reduced when replacing work at the development to work of XDDP.

By devising this method, we can now explain XDDP by using sample of cases that is familiar to the development and illustrate numerically benefit for introduction. As a result, we can eliminate psychological barrier at the development and introduce XDDP to the quick delivery development.

## 1. 研究動機

### 1.1. 短納期開発の実態

我々の派生開発の現場は、開発期間 0.5～1.0[月]，開発要員 1～2[人]，開発規模 1.5～5.0[kSLOC]の短納期開発である。短納期開発現場では工期の短さが原因で変更設計、及び変更による影響範囲の特定が十分に行われぬまま実装が進められる場合が多い。そのため表 1 に示すような「変更ミス」や「変更漏れ」の不具合が発生し、これが工期を圧迫するという負のスパイラル状態に陥っている。

表 1. 代表的な不具合事例(組込系の派生開発)

不具合の種類	不具合事例	
	研究員 A	研究員 B
変更ミス	誤って配列の領域外にアクセスする処理を埋め込んだため、障害が発生した。	状態の取得方法，及び処理対象のバッファサイズを誤り，障害が発生した。
変更漏れ	明日の天気予報を複数の箇所に表示するシステムがあり，表示内容を文字から絵文字に変更した所，一部のプログラムで変更漏れが発生し，一部の表示が文字のままとなった。	パスワード 8 回の失敗で全データを消去する体重計があり，新たなデータが追加されたときの変更漏れによって，パスワード 8 回失敗しても，そのデータだけ消去されずに残ってしまった。

### 1.2. XDDP 導入の効果と導入に対する障壁

我々はこのような状況を打開するため，XDDP(eXtreme Derivative Development Process)の導入を検討した。XDDP は，派生開発で発生しやすい問題(「部分理解の罨」，「変更漏れ」など)を解決するために開発された派生開発専用のプロセスである。

そこで，我々の現場で発生した代表的な不具合事例の中で XDDP によって対応が可能な不具合の割合を，XDDP 3 点セットの仕組みと 7 つの設計原理の関係<sup>[1]</sup>を用いて調査した。その結果，不具合事例の約 78% が XDDP によって対応できる問題であることが判明した。(図 1)

XDDP は，実装を行う前に 3 点セット(「変更要求仕様書」，「トレーサビリティマトリクス(TM)」，「変更設計書」と呼ばれる成果物の作成とレビューを行うことで，「変更ミス」や「変更漏れ」に気づくための仕組みを持つ。この仕組みによる実装前の十分な品質確保により手戻り作業を減らすことができれば，不具合の削減や生産性の向上が期待できる。そのため我々は，不具合の発生によって工期を圧迫している現場の状況を，XDDP の導入によって改善できると考えた。

しかし，開発の工期が短く，日頃から納期に追われている現場では，新たな手法を導入するための心理的障壁は高く，XDDP を導入することは非常に困難であった。

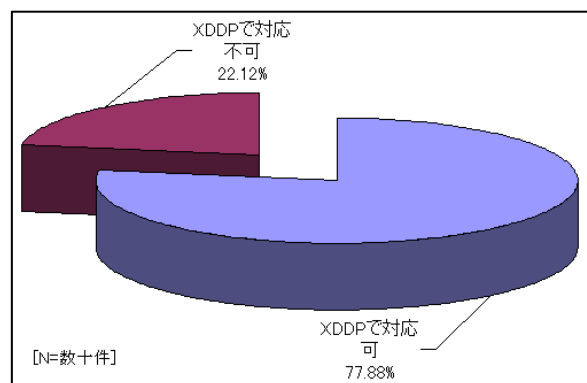


図 1. 不具合事例の調査結果

## 2. 現状分析

### 2.1. 現場での心理的障壁

我々は XDDP の導入について、現場でどのような障壁が発生するかを調べるため、事前に現場への意識調査アンケートを実施した。意識調査アンケートの結果から最も顕著であった意見を抜粋すると、以下の 3 点になる。(意識調査アンケートの内容と結果は、付録 1 と付録 2 を参照)

- (1) 世間で本当に効果が出ているか、疑問に思う。
- (2) 効果は得られるかもしれないが、それ以上に作業時間が増える。
- (3) 十分な教育なしに方法論だけ導入しても効果が上がらない。

上記 3 点の意見から、短納期開発現場では XDDP の導入に対する障壁として心理的障壁が中心となっていることが判明した。そこで我々は上記 3 点の意見が現場のどのような心理状態から発生するかを考察した。

(1) の意見については、XDDP の現場に対する効果が分からないという心理状態から生じた不安(「効果に対する不安」)であると考えた。この心理状態は、何らかの改善を行いたいとは考えているが、適切な改善策が思い浮かばない現場で発生しやすいものと思われる。

(2) の意見については、XDDP の効果を期待しているが「作業時間の増加に対する不安」が強くて導入に踏み切れないという心理状態から発生していると考えた。この心理状態は「作業時間が増えなければ導入するが、作業時間が増えるから導入できない」とも解釈でき、時間の増加に不安を持つ現場で発生しやすいものと思われる。

(3) の意見については、「十分な教育があれば効果は上がるが、十分な教育がなければ効果は上がらない」という考えから発生したものである。しかも、時間の増加に不安を持つ現場では十分な教育を受けるための時間がない(「教育時間に対する不安」)。

以上の考察から、我々の現場では、「効果に対する不安」、「作業時間の増加に対する不安」及び「教育時間に対する不安」の 3 つが心理的障壁となって、XDDP の導入に踏み切れないと結論づけた。

### 2.2. 先行技術

XDDP の導入に対する現場の心理的障壁を扱った先行研究<sup>[2]</sup>では、XDDP の心理的障壁を「XDDP への不信」、「最初の 1 歩への不安」及び「変化への抵抗」に分類しており、実在のプロジェクトに対する XDDP のシミュレーションを実行することで「XDDP の効果に対する疑念」を解消する方法を挙げている。この方法は我々の現場で発生した「効果に対する不安」を解消する方法として有効である。しかし、「作業時間の増加に対する不安」、「教育時間に対する不安」を解消する方法は提示されていない。

また別の先行研究<sup>[1]</sup>では、XDDP を部分的に導入する方法を挙げているが、この方法は SLCP-JCF2007(Software Life Cycle Process-Japan Common Frame 2007)などによって、各工程での規律が確保された組織にのみ容認された方法である。従って、それ以外の組織においては、XDDP の部分的な導入による十分な効果は期待できない<sup>[3]</sup>。

そのため我々は、現場の「作業時間の増加に対する不安」と「教育時間に対する不安」を解消し、XDDP を全面的に導入する方法を考案する必要があった。

### 3. 解決策

短納期開発現場に XDDP を導入するためには、現場作業者の「作業時間の増加に対する不安」と「教育時間に対する不安」を解消する必要がある。そこで我々は、以下の施策を試みた。

- (1) 「教育時間に対する不安」を解消するために現場のデータを使って 3 点セットのサンプルを作成する手法「REX-P」<sup>1</sup>を考案し、このサンプルを使った説明会を実施する。
- (2) 「作業時間増加に対する不安」を解消するため、「ながら作業置換表」を作成し、調査や確認のための作業時間が減少する仕組みを説明する。

#### 3.1. サンプル作成手法「REX-P」を使った現場サンプルの作成

XDDP を現場に導入する際、多くの組織では、導入推進者<sup>2</sup>が一般的な XDDP の事例を現場作業者に紹介し、それを参考にして現場作業者が XDDP に取り組む、という方式を採っている。この場合、現場作業者には、次のような「教育時間に対する不安」が生じる。

- (1) 一般的な事例だけでは、自分達の作業を具体的に想像できない。
- (2) 不明点を導入推進者に質問した時、導入推進者も文献情報しか知らないため、適切な回答を得られない。

「教育時間に対する不安」を解消するためには、導入推進者自身が XDDP を実践し、現場のデータを使った XDDP 成果物のサンプル(以下「サンプル」)を作成するのが効果的である。しかし、導入推進者がサンプルを作成するためには、事前に対象プロジェクトの知識を習得する必要がある、非常に大きな手間が掛かる。また、導入推進者の努力によって対象プロジェクトの知識を習得することができたとしても、プロジェクトによっては、要求仕様書の内容とソフトウェアコードの変更内容との対応付けができないために、サンプル作成が困難になる場合もある。

そこで我々は、既存プロジェクトの成果物であるソフトウェアコード情報を元に、定型化した作業によりサンプル作成を行う手法「REX-P」及び「REX-P」で使用する様式「変更内容一覧表」(付録 3 参照)を考案した。「REX-P」では、既存のソフトウェアコード情報を利用して、サンプルを作成する。この時「変更内容一覧表」には、ソフトウェアコードの差分情報(変更前と変更後)と要求仕様書等の文書情報との対応付けに必要な情報が書き込まれる。これにより、対象プロジェクトの知識を習得できない場合や、要求仕様書の内容とソフトウェアコードの変更内容との対応付けができない場合でも、導入推進者によるサンプル作成が可能になる。

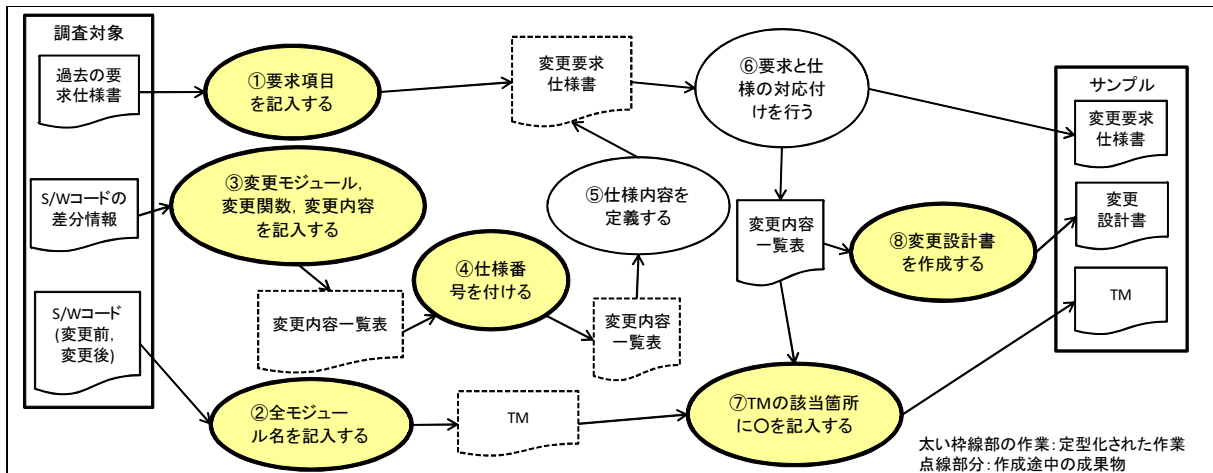
「REX-P」によるサンプル作成手順を図 2 に示す。またサンプル作成作業の例を付録 4 及び図 3 に示す。

「REX-P」を用いることによって導入推進者は、現場データを使ったサンプル作成が可能になる。現場作業者は、現場データを使ったサンプルを参照することで自分達の作業を具体的に想像できる。その結果、現場作業者の「教育時間に対する不安」が解消する。

---

<sup>1</sup> 「REX-P」：リバーエンジニアリング XDDP を略して命名した。

<sup>2</sup> ここで言う導入推進者には、導入推進者から依頼を受けた外部講師等を含んでいる。



No	内容	
①	要求項目を記入する	過去の要求仕様書等より要求項目を抽出し、変更要求仕様書の要求欄に記入する。
②	全モジュール名を記入する	ソフトウェアコード(変更前、変更後)から全モジュール名を抽出し、TMのモジュール名を記入する。
③	変更モジュール、変更関数、変更内容を記入する	diff等で調査したソフトウェアコードの差分情報に基づき、「変更内容一覧表」の変更モジュール、変更関数、変更内容(変更前、変更後)を記入する。 この時、「変更内容一覧表」の変更内容欄に変数名や定数名を直接記入しておく、続く作業で変更内容間の関係を見つけやすい。
④	仕様番号を付ける	③で記入した変更内容間の関係を調べる。この時、つながりのある複数の変更内容(例えば同じ変数、定数を使っている等)が見つかった場合、「変更内容一覧表」の仕様番号欄に共通の仕様番号(例えば“仕様1”、“仕様2”等)を記入する。他とのつながりが見つからなかった変更内容については、それぞれ固有の仕様番号を付ける。このようにして「変更内容一覧表」の仕様番号欄を全て埋める。
⑤	仕様内容を定義する	④で付けた仕様番号を変更要求仕様書の仕様番号欄に番号順に記入する。次にそれぞれの仕様番号について「変更内容一覧表」の内容を元に仕様内容を定義し、変更要求仕様書に記入する。仕様内容は日本語で動詞とする。
⑥	要求と仕様の対応付けを行う	変更要求仕様書の中で要求と仕様との対応付けを行う。①で記入済みの要求と対応付けが可能な仕様については、対応する要求項目にぶら下げる形で仕様を移動する。もし仕様に対応する要求項目が存在しない場合は、変更要求仕様書に新しい要求項目を追加する。変更要求仕様書の仕様番号を要求項目に対応づける形で定義し直す。この時「変更一覧表」の仕様番号も同様に定義し直す。
⑦	TMの該当場所に○を記入する	「変更内容一覧表」のモジュール名と仕様番号に基づき、TMの該当個所に○を記入する。
⑧	変更設計書を作成する	「変更内容一覧表」の変更内容に基づき、変更設計書を作成する。

図 2. 「REX-P」によるサンプル作成手順

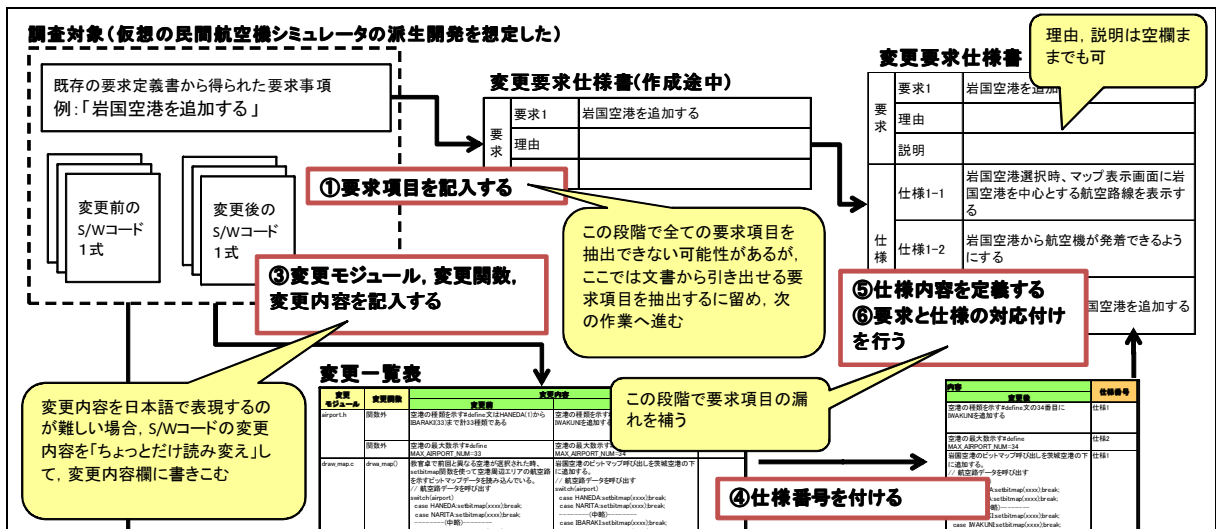


図 3. 「REX-P」によるサンプル作成作業の例(一部)

### 3.2. 「ながら作業置換表」による作業時間減少の説明

次に我々は「作業時間の増加に対する不安」を解消するための施策を検討した。まず我々は現場作業がなぜ「作業時間の増加に対する不安」を感じるのかを分析した。

XDDPを導入すると、3点セットと呼ばれる成果物を作成する必要がある。そのため現場作業にとっては、XDDPの3点セット作成作業は、新たに追加される作業と位置付けられる。

一方、先行研究によれば、XDDPは従来の派生開発における作業の順番を効率的に進められるように変えたものである。3点セットの作成についても、従来の作業に追加して実施するものではなく、従来の作業を置き換えるもの、としている<sup>[3]</sup>。

現場作業の思考とXDDPの仕組みの違いを図4に示す。我々は、現場作業者にXDDPの仕組みを理解させることで、現場作業者の不安を解消できると考えた。

先行研究によれば、従来の派生開発では実装作業<sup>3</sup>の中で「ながら作業」<sup>4</sup>が多く含まれており、これが実装作業の生産性を悪化させている<sup>[3]</sup>。しかし、短納期開発現場では「ながら作業」が現場作業にとって必然の作業となっているため、現場作業者は「ながら作業」による生産性の悪化を認識できない。

そこで我々は、以下の手順で「ながら作業」の実態調査を行い、XDDP導入による作業時間減少の予測を定量的な形で現場に示した。

- (1) 「ながら作業」に関するアンケートを使って現場における「ながら作業」の実態を調査した。「ながら作業」に関するアンケートの内容を付録5に示す。
- (2) XDDPを導入した場合「ながら作業」がどのようにXDDP作業に置き換えられ、作業時間がどのように変化するかを説明するための様式「ながら作業置換表」を作成した。
- (3) 「ながら作業置換表」に(1)のアンケート結果を入力した。
- (4) XDDPを導入した場合、現状の「ながら作業」がどのようにXDDP作業に置き換えられ、作業時間がどのように変化するかを「ながら作業置換表」で説明した。

「ながら作業置換表」による説明の手順を図5に示す。また「ながら作業置換表」を付録6に示す。

「ながら作業置換表」を用いることによって導入推進者は、XDDP導入による作業時間減少の予測を定量的な形で現場に示すことができる。現場作業者は、XDDP導入による作業時間減少を理解できる。その結果、現場作業者の「作業時間の増加に対する不安」が解消する。

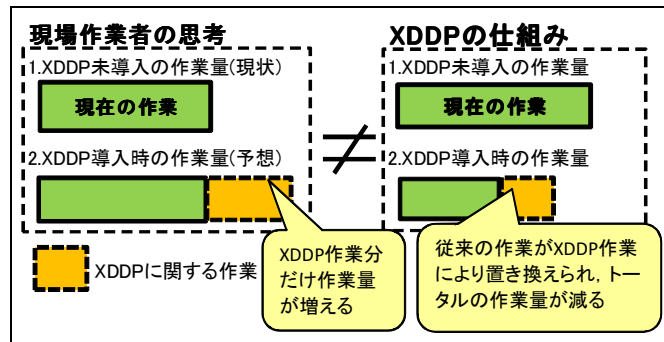
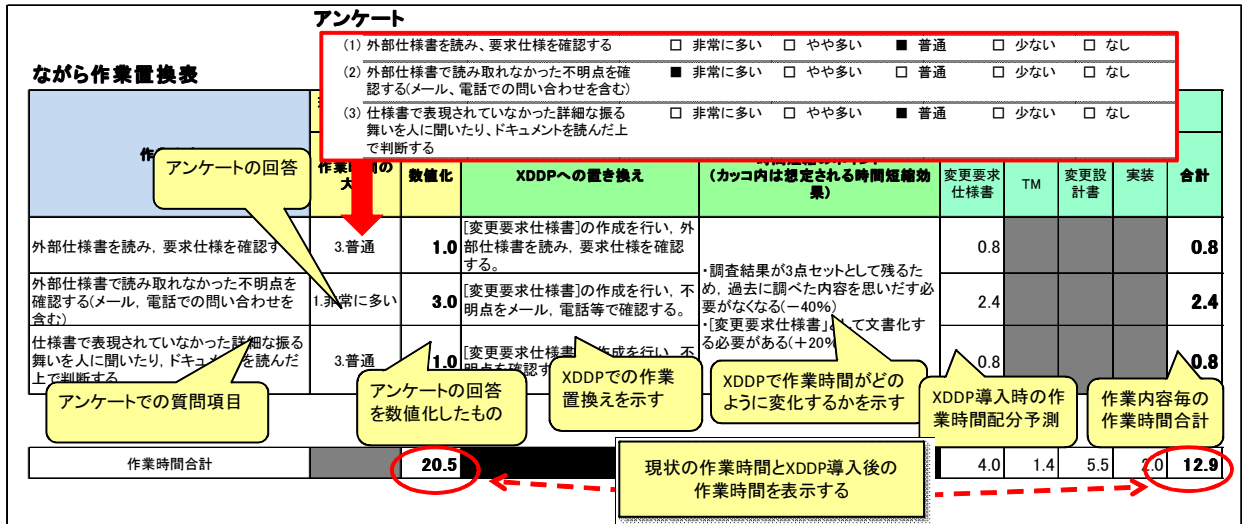


図4. 現場作業者の思考とXDDPの仕組みの違い

<sup>3</sup> ソフトウェアコード作成工程で行われる作業をここでは「実装作業」と呼ぶ

<sup>4</sup> 「ながら作業」：ここでは実装作業の中で行われるソフトウェアコード作成以外の作業、例えば仕様の確認、影響個所の検討等を指す。



#### 4. 解決策の検証

##### 4.1. 「REX-P」によるサンプルの作成

我々は2つのプロジェクトを選択し、「REX-P」を使ってサンプルを作成した。サンプル作成に利用したプロジェクトの概要及びサンプルの概要を表2に示す。この作業により我々は「REX-P」を使うことによってサンプル作成が可能であることを確認した。

表2. サンプル作成に利用したプロジェクト概要とサンプルの概要

	プロジェクト A	プロジェクト B
プロジェクトの概要	交通管制インフラ制御ソフトの機能追加	機器 C の制御対象の追加
開発規模、開発期間	0.5 人月・0.5 ヶ月	0.6 人月・0.6 ヶ月
S/W コード規模(変更量)	0.4 kSLOC	0.4 kSLOC
S/W コード規模(流用元)	9.3 kSLOC	82.5 kSLOC
サンプル	変更要求仕様書/TM	A4 5 枚
	変更設計書	A4 43 枚
サンプル作成に要した時間	16 時間	11 時間

次にサンプルを使って現場説明会を実施した。説明会実施後にアンケートを行い「教育時間に対する不安」の変化を調査した。

アンケート結果を図6に示す。研究員 A の現場については、「不安あり」の割合が説明会実施前の 70% から説明会実施後は 41% に減少した。一方「不安なし」の割合が 0% から 18% に増加した。研究員 B の現場については、「不安あり」の割合が 100% から 29% に減少し、「不安なし」の割合が 0% から 71% に増加した。

アンケートの結果から、研究員 A、研究員 B のいずれの現場でも「教育時間に対する不安」が減っていることを確認した。

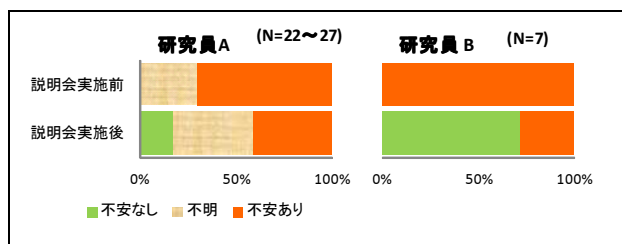


図 6. 「教育時間に対する不安」の変化



## 4.2. 「ながら作業置換表」を使った説明会の実施

「ながら作業置換表」を使って現場説明会を実施した。説明会実施後にアンケートを行い「作業時間の増加に対する不安」の変化を調査した。

アンケートの結果を図 7 に示す。

研究員 A の現場については、「不安あり」の割合が説明会実施前の 41% から説明会実施後は 23% に減少した。一方「不安なし」の割合が 0% から 18% まで増加した。研究員 B の現場については、「不安あり」の割合が 57% から 43% に減少し、「不安なし」の割合が 0% から 29% に増加した。

アンケートの結果、研究員 A、研究員 B のいずれの現場でも「作業時間の増加に対する不安」が減っていることを確認した。

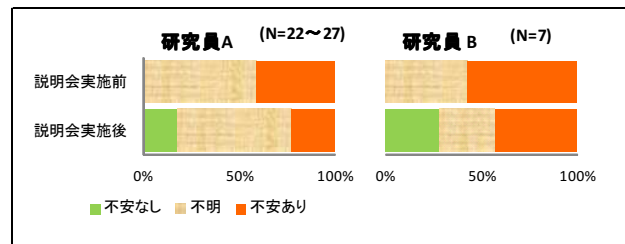


図 7. 「作業時間の増加に対する不安」の変化

## 5. 結論

### 5.1. 取組みと成果

短納期開発現場へ XDDP を導入する上で、「教育時間に対する不安」と「作業時間の増加に対する不安」が心理的障壁になっていた。その心理的障壁を解消するために「REX・P」と「ながら作業置換表」を考案し、これらを使って現場に対する説明会を実施した。説明会後のアンケートでは、現場作業者の「教育時間に対する不安」と「作業時間の増加に対する不安」が減っていることを確認した。その後我々は、2つの短納期開発プロジェクトに XDDP を試行導入することに成功した。

導入成功の要因は、今回考案した手法によって、現場に馴染みのある事例のサンプルを導入推進者側で作成できたことと、導入によるメリットを数値によって具体的に示したことにある。

### 5.2. 今後の課題

研究員それぞれの現場における XDDP 導入が試行レベルに留まっており、広く適用するには至っていない。

今後は本研究の成果を生かして XDDP の導入を進めていく。また XDDP 導入によって得られる現場の実績データを「ながら作業置換表」にフィードバックすることで、「ながら作業置換表」の精度向上を図っていく。

本論文が XDDP やその他様々な改善施策推進の一助になることを切に願う。

## 6. 参考文献

- [1] 良知敦他, “派生開発で成功するための施策—部分的に XDDP の仕組みを取り入れた設計書の提案”, ソフトウェア品質管理研究会第 26 年度(2010 年度)分科会成果報告, 日本科学技術連盟, p1-8, 2011
- [2] 長友優治他, “派生開発に XDDP を導入する際の障壁とその解消に向けたアプローチ”, ソフトウェア品質管理研究会第 25 年度(2009 年度)分科会成果報告, 日本科学技術連盟, p1-8, 2010
- [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, p373-387, 2007



# 付録1. 意識調査アンケートの内容

派生開発における開発プロセス改善についてのアンケート			
	所属	氏名	
質問1 貴方が1年間で担当しているプロジェクト数はいくつですか。概算で構いませんので教えてください。	<input type="checkbox"/> 0~9工事	<input type="checkbox"/> 10~19工事	<input type="checkbox"/> 20~49工事
	<input type="checkbox"/> 50~99工事	<input type="checkbox"/> 100~199工事	<input type="checkbox"/> 200工事以上
質問2 自分達の組織(課又はグループ)で該当すると思われるものを選んで下さい。			
a) 仕事量に比べて人員数が少ない	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
b) 仕事の工期が短い(適正な工期が与えられていない)	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
c) 職場内のコミュニケーションが足りない	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
d) 担当者の技量が不足している。	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
e) 仕事を円滑に進めて行く上で障害となる無駄作業が多い。 (具体的に「無駄」だと感じている作業があれば、お書き下さい)	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
f) 仕事の進め方が最適化されていない。	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
g) 管理者の管理能力が不足している。	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
h) その他(自由に書いて下さい)			
質問3 自分達の組織の問題について、改善が必要だと思いますか	<input type="checkbox"/> 思う	<input type="checkbox"/> 思わない	<input type="checkbox"/> どちらとも言えない
(理由)			
質問4 自分達の組織の問題について、自分達自身(自分と同じチームのメンバーだけ)の力で改善できると思いますか	<input type="checkbox"/> 思う	<input type="checkbox"/> 思わない	<input type="checkbox"/> どちらとも言えない
(理由)			
質問5 我々は派生開発における問題解決の手法としてXDDPという方法に着目しています。XDDPについてはご存じですか	<input type="checkbox"/> 知っている	<input type="checkbox"/> 名前は知っている	<input type="checkbox"/> 知らない
質問6 XDDPを当社に導入することについて貴方のお考えをお聞かせ下さい	<input type="checkbox"/> 賛成	<input type="checkbox"/> 反対	<input type="checkbox"/> 条件付きで賛成
(理由)			
質問7 XDDP導入するに当たり不安な要素は何ですか。			
a) 今までのやり方を変えてまで導入する必要を感じない	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
b) 世間で本当に効果が出ているのか、疑問に思う。	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
c) 作業時間が増えるだけで、効果が上がらない。	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
d) 効果は得られるかもしれないが、それ以上に作業時間が増える。	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
e) 我々の仕事には適合していない可能性が高い。	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
f) 十分な教育なしに方法論だけ導入しても効果が上がらない	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
g) 導入準備に時間を取られるのは嫌だ	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
h) 社内の組織標準に合致していない	<input type="checkbox"/> はい	<input type="checkbox"/> いいえ	<input type="checkbox"/> どちらとも言えない
i) その他(自由に書いて下さい)			

付録2. 意識調査アンケートの結果

質問内容	それぞれの組織における回答数					
	研究員A			研究員B		
	はい	いいえ	どちらとも言えない	はい	いいえ	どちらとも言えない
自分達の組織(課又はグループ)で該当すると思われるものを選んで下さい.						
a)仕事量に比べて人員数が少ない	26	1	2	6	0	1
b)仕事の工期が短い(適正な工期が与えられていない)	17	3	9	6	0	1
c)職場内のコミュニケーションが足りない	9	11	9	1	1	5
d)担当者の技量が不足している.	8	10	11	3	0	4
e)仕事を円滑に進めて行く上で障害となる無駄作業が多い.	8	5	16	3	0	4
f)仕事の進め方が最適化されていない.	7	4	18	6	0	1
g)管理者の管理能力が不足している.	3	13	12	2	0	5
自分達の組織の問題について、改善が必要だと思いますか	17	1	1	6	0	1
自分達の組織の問題について、自分達自身(自分と同じチームのメンバーだけ)の力で改善できると思いますか	0	17	11	0	2	5
XDDPIについて知っていますか	1	22	6	4	2	1
XDDPを導入することについて賛成ですか	2	4	16	1	0	5
XDDP導入するに当たり不安な要素は何ですか.						
a)今までのやり方を変えてまで導入する必要を感じない	4	9	14	0	5	2
b)世間で本当に効果が出ているのか、疑問に思う.	12	2	13	4	2	1
c)作業時間が増えるだけで、効果が上がらない.	5	2	20	0	3	4
d)効果は得られるかもしれないが、それ以上に作業時間が増える.	11	0	16	4	0	3
e)我々の仕事には適合していない可能性が高い.	2	6	19	1	3	3
f)十分な教育なしに方法論だけ導入しても効果が上がらない	19	0	8	7	0	0
g)導入準備に時間を取られるのは嫌だ	7	2	18	1	3	3
h)社内の組織標準に合致していない	1	3	23	0	2	5

付録3. 変更内容一覧表

変更 モジュール	変更関数	変更内容		仕様番号
		変更前	変更後	

変更関数名を記入する。関数外の場合は"関数外"と記入する

変更内容をBefore/Afterの形式で記入する。  
その時、ソースコードの変更内容を出来る限り忠実に(ただしコピーペーストすることなしに)記入する

変更モジュール名を記入する

変更内容1件につき、仕様番号を付与する。その時、関連のある複数の変更内容については、同じ仕様番号を複数個所に記入する

# 付録4.「REX-P」によるサンプル作成作業の例

調査対象(仮想の民間航空機シミュレータの派生開発を想定した)

既存の要求定義書から得られた要求事項  
例:「岩国空港を追加する」

## 変更要求仕様書(作成途中)

要求1	岩国空港を追加する
理由	

## 変更要求仕様書

要求1	岩国空港を追加する
理由	
説明	
仕様1-1	岩国空港選択時、マップ表示画面に岩国空港を中心とする航空路線を表示する
仕様1-2	岩国空港から航空機が発着できるようにする

理由、説明は空欄  
ままでも可

## 変更設計書

Module	draw_map.c	作成者	xxxxxx
仕様	岩国空港選択時、マップ表示画面に岩国空港を中心とする航空路線を表示する	修正者	xxxxxx
		確認者	xxxxxx

Module	draw_map.c	作成者	xxxxxx
仕様	岩国空港選択時、マップ表示画面に岩国空港を中心とする航空路線を表示する	修正者	xxxxxx
		確認者	xxxxxx

Module	draw_map.c	作成者	xxxxxx
仕様	岩国空港選択時、マップ表示画面に岩国空港を中心とする航空路線を表示する	修正者	xxxxxx
		確認者	xxxxxx

①要求項目を記入する

③変更モジュール、変更関数、変更内容を記入する

⑤仕様内容を定義する  
⑥要求と仕様の対応付けを行う

## 変更一覧表

変更モジュール	変更関数	変更内容	仕様番号
airport.h	関数外	空港の種類を示す#define文はHANEDA(1)からIBARAKI(33)まで計33種類である	仕様1
airport.h	関数外	空港の最大数示す#define MAX_AIRPORT_NUM=33	仕様2
draw_map.c	drwa_map()	教官卓で前回と異なる空港が選択された時、setbitmap関数を使って空港周辺エリアの航空路線を示すビットマップデータを読み込んでいる。// 航空路線データを読み出す switch(airport) case HANEDA:setbitmap(1); break; case NARITA:setbitmap(2); break; case IBARAKI:setbitmap(33); break; default:break;	仕様1-1 仕様1-2

この段階で全ての要求項目を抽出できない可能性があるが、ここでは文書から引き出せる要求項目を抽出するに留め、次の作業へ進む

この段階で要求項目の漏れを補う

④仕様番号を付ける

⑧変更設計書を作成する

変更内容を日本語で表現するのが難しい場合、S/Wコードの変更内容を「ちょっとだけ読み変え」して、変更内容欄に書きこむ

②全モジュール名を記入する

モジュールの分割単位は、ファイル単位、コンポーネント単位、ディレクトリ単位等がある。どの単位で分割するかは、概ね「20~30個の範囲で収まる数」を目安とする。

## TM(作成途中)

airport.h	weather.c	control_databases
weather.h	airport.dat	int.airport.c
int.airport.h	int.aircraft.c	draw_map.c
int.weather.c	flight.c	engine.c
instrument.c	trouble.c	engine.c
engine.c	weather.c	control_databases

モジュールの分割単位は、TMと一致させておくこと

ソースコードの変更内容と"変更内容"欄の記述が1対1対応するぐらいの細かさで書く

⑦TMの該当個所に○を記入する

変更一覧表の変更内容に基づいて機械的に○を記入していく

変更一覧表の変更内容を機械的にコピー&ペーストする

同じ情報(変数等)扱っている複数の変更について同じ仕様番号を割り当てる

S/Wコードの変更内容を漏れなく書きこむのがコツ

要求/仕様	airport.h	weather.h	int.airport.h	int.aircraft.c	draw_map.c	flight.c	engine.c	trouble.c	weather.c	control_databases
要求1										
理由										
説明										
仕様1-1	○									
仕様1-2					○					
仕様1-3										○

注:①~⑧の数値は、本文の図2と一致させた。

## 付録5. 「ながら作業」に関するアンケート

### ソフトウェア開発における実装作業の実態調査

所属	氏名
----	----

S/W開発において上流重視が叫ばれて久しいですが、現実には工期の制約、要員不足等の問題があり、実装工程(コーディング)を先行させながら、設計作業を並行して実施しているのが実態ではないかと思われます。そこで今回、実装工程における作業の実態を調査するため、簡単なアンケートを実施します。下記の質問の中から回答欄で最も相応しいと思われるものを1つ選び、マークして下さい。感覚的なもので構いませんので、ご協力をお願いします。

質問1 貴方の全勤務時間に占める実装フェーズ作業の割合はどのぐらいですか。  70%以上  50～69%  30～49%  10～29%  10%未満

質問2 以下の内容は、実装工程時に行われているであろう作業内容をリストアップしたものです。リストアップした作業内容について、ご自身が感じている作業時間の大小をお答えください。ただし作業時間は実装工程に入った後の作業時間のみを考慮します。実装工程前の作業時間(例えば実装工程前に行う設計作業)は考慮しないで下さい。

- |   |                                |                               |                             |                              |                             |
|---|--------------------------------|-------------------------------|-----------------------------|------------------------------|-----------------------------|
| (1) 外部仕様書を読み、要求仕様を確認する  | <input type="checkbox"/> 非常に多い | <input type="checkbox"/> やや多い | <input type="checkbox"/> 普通 | <input type="checkbox"/> 少ない | <input type="checkbox"/> なし |
| (2) 外部仕様書で読み取れなかった不明点を確認する(メール、電話での問い合わせを含む)                      | <input type="checkbox"/> 非常に多い | <input type="checkbox"/> やや多い | <input type="checkbox"/> 普通 | <input type="checkbox"/> 少ない | <input type="checkbox"/> なし |
| (3) 仕様書で表現されていなかった詳細な振る舞いを人に聞いたり、ドキュメントを読んだ上で判断する                 | <input type="checkbox"/> 非常に多い | <input type="checkbox"/> やや多い | <input type="checkbox"/> 普通 | <input type="checkbox"/> 少ない | <input type="checkbox"/> なし |
| (4) 内部仕様書を読み、内部構造を確認する  | <input type="checkbox"/> 非常に多い | <input type="checkbox"/> やや多い | <input type="checkbox"/> 普通 | <input type="checkbox"/> 少ない | <input type="checkbox"/> なし |
| (5) ソースコード上での今回の要求に対する変更箇所を検討し、決定する                               | <input type="checkbox"/> 非常に多い | <input type="checkbox"/> やや多い | <input type="checkbox"/> 普通 | <input type="checkbox"/> 少ない | <input type="checkbox"/> なし |
| (6) 今回の変更に影響する影響範囲を調査する(Grep等での検索行為含む)。場合によっては他製品、他モジュールへの影響調査も含む | <input type="checkbox"/> 非常に多い | <input type="checkbox"/> やや多い | <input type="checkbox"/> 普通 | <input type="checkbox"/> 少ない | <input type="checkbox"/> なし |
| (7) 変更対象となるソースコードを調査する。   | <input type="checkbox"/> 非常に多い | <input type="checkbox"/> やや多い | <input type="checkbox"/> 普通 | <input type="checkbox"/> 少ない | <input type="checkbox"/> なし |
| (8) 前任者や他の担当者から聞いた情報を頭の中で整理する。                                    | <input type="checkbox"/> 非常に多い | <input type="checkbox"/> やや多い | <input type="checkbox"/> 普通 | <input type="checkbox"/> 少ない | <input type="checkbox"/> なし |
| (9) 実装方法を検討する。場合によっては言語仕様等に関する文献調査やネット上での情報収集を行う                  | <input type="checkbox"/> 非常に多い | <input type="checkbox"/> やや多い | <input type="checkbox"/> 普通 | <input type="checkbox"/> 少ない | <input type="checkbox"/> なし |
| (10) 他の担当者が追加した機能から受ける影響を検討する                                     | <input type="checkbox"/> 非常に多い | <input type="checkbox"/> やや多い | <input type="checkbox"/> 普通 | <input type="checkbox"/> 少ない | <input type="checkbox"/> なし |
| (11) 設計書で表現されていなかった詳細な内部仕様を、人に聞いたり、ドキュメントを読んだ上で判断する               | <input type="checkbox"/> 非常に多い | <input type="checkbox"/> やや多い | <input type="checkbox"/> 普通 | <input type="checkbox"/> 少ない | <input type="checkbox"/> なし |
| (12) すでに変更した箇所との調整のためのソースコードを確認する                                 | <input type="checkbox"/> 非常に多い | <input type="checkbox"/> やや多い | <input type="checkbox"/> 普通 | <input type="checkbox"/> 少ない | <input type="checkbox"/> なし |
| (13) 実装する(コーディング)コンパイル、ビルド等の行為を含む                                 | <input type="checkbox"/> 非常に多い | <input type="checkbox"/> やや多い | <input type="checkbox"/> 普通 | <input type="checkbox"/> 少ない | <input type="checkbox"/> なし |

(13) その他

(具体的に「作業時間が長い」と感じている作業があれば、記入して下さい)

ご協力、ありがとうございました。

付録6. ながら作業置換表

ながら作業置換表（[現状の実装工程における作業量]と[XDDP導入後の作業の置換]による作業時間の比較）

作業内容	現状の実装工程における作業量		XDDPによる作業置換と、その際の時間短縮効果		XDDP導入後の作業時間予測					
	作業時間の 大小	数値化	XDDPへの置き換え	時間短縮のポイント (カッコ内は想定される時間短縮効果)	3点セットの作成					
					変更要求 仕様書	TM	変更設計 書	実装	合計	
外部仕様書を読み、要求仕様を確認する	3.普通	1.0	[変更要求仕様書]の作成を行い、外部仕様書を読み、要求仕様を確認する。		0.8					0.8
外部仕様書で読み取れなかった不明点を確認する (メール、電話での問い合わせを含む)	1.非常に多い	3.0	[変更要求仕様書]の作成を行い、不明点をメール、電話等で確認する。	・調査結果が3点セットとして残るため、過去に調べた内容を思い出す必要がなくなる(-40%) ・[変更要求仕様書]として文書化する必要がある(+20%)	2.4					2.4
仕様書で表現されていなかった詳細な振る舞いを人に聞いたり、ドキュメントを読んだ上で判断する	3.普通	1.0	[変更要求仕様書]の作成を行い、不明点を確認する。		0.8					0.8
内部仕様書を読み、内部構造を確認する	4.少ない	0.5	[TM]の作成を行い、内部構造を理解し、影響箇所を特定する。			0.1				0.1
ソースコード上での今回の要求に対する変更箇所を検討し、決定する	1.非常に多い	3.0	[TM]の作成を行い、仕様に対するソースコード上の変更箇所を検討し、該当する箇所に○を付ける	・調査結果が3点セットとして残るため、過去に調べた内容を思い出す必要がなくなる(-40%) ・以前に、同じような作業があれば、その[TM]の内容を参考にできる(-50%)		0.6				0.6
今回の変更に影響する影響範囲を調査する(Grep等での検索行為含む)。場合によっては他製品、他モジュールへの影響調査も含む	2.やや多い	2.0	[TM]の作成を行い、今回の変更に関する影響範囲を調査し、変更が必要な箇所に○を付ける。場合によっては他製品、他モジュールへの影響調査も含む。	・[TM]の該当する項目を探して"○"を記入する作業が発生する(+10%)		0.4				0.4
変更対象となるソースコードを調査する。	3.普通	1.0	[変更設計書]の作成を行い、他のソースコードの実装方式を調査する。				1.0			1.0
前任者や他の担当者から聞いた情報を頭の中で整理する。	4.少ない	0.5	[変更要求仕様書][TM][変更設計書]の作成を行い、前任者や他の担当者が示した[TM]や[変更設計書]の項目を参照する				0.5			0.5
実装方法を検討する。場合によっては言語仕様等に関する文献調査やネット上での情報収集を行う	4.少ない	0.5	[変更設計書]の作成を行い、実装方法を検討する。場合によっては言語仕様等に関する文献調査やネット上での情報収集を行う	・調査結果が3点セットとして残るため、過去に調べた内容を思い出す必要がなくなる(-40%) ・[TM]の1項目ごとに「変更設計書」を1枚作成する作業が発生する(+40%)			0.5			0.5
他の担当者が追加した機能から受ける影響を検討する	4.少ない	0.5	[TM]上で他の担当者の担当部分を参照し、○が重複しているモジュールを特定する。特定されたジュールの[変更設計書]を調査する				0.5			0.5
設計書で表現されていなかった詳細な内部仕様を、人に聞いたり、ドキュメントを読んだ上で判断する	5.なし	0.0	[変更設計書]の作成を行い、不明点を確認する。							
すでに変更した箇所との調整のためのソースコードを確認する	3.普通	1.0	XDDPではソースコードの逐次変更は認めていないので、すでに変更されている状況は発生しない。	すでに変更されている状況は発生しないので、調整する作業も必要なくなる。(×0)						
実装する(コーディング)コンパイル、ビルド等の行為含む	2.やや多い	2.0	なし	なし				2.0		2.0
作業時間合計		16.0			4.0	1.1	2.5	2.0		9.6