

## 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—

#### Measure to Successful Derivative Development —Proposal of Specification That Partially Adopted XDDP—

主査	足立 久美	株式会社デンソー
副主査	飯泉 紀子	株式会社日立ハイテクノロジーズ
副主査	清水 吉男	株式会社システムクリエイツ
研究員（リーダー）	良知 敦	NEC アクセステクニカ株式会社
	伊藤 祐馬	伊藤忠テクノソリューションズ 株式会社
	大立 薫	ベックマン・コールター株式会社
	鈴木 孝輔	株式会社日立ハイテクソリューションズ
	瀬戸 一郎	株式会社アドバンテスト

### 研究概要

我々の派生開発の現場では、検査工程で検出された不具合の大半は「要件定義／方式設計」で作成する設計書に起因する問題である。最近ではこのような問題を解決するために、XDDP (eXtreme Derivative Development Process)を導入した成功事例が多く報告されている。XDDP では3点セットと呼ばれる「変更要求仕様書」「トレーサビリティマトリクス」「変更設計書」がすべて揃うことで最大限の効果を発揮する。しかし、これら3点セットを開発現場に一括導入するには、開発プロセスと作成する成果物を同時に変更しなければならないため、実践は容易ではない。

そこで我々は、3点セットの1つである「変更要求仕様書」が「要件定義／方式設計」で作成する従来の設計書と密接に関連していることに着目し「変更要求仕様書」による変更の仕組みのみを既存のプロセスに取り入れる方法を研究した。実際に開発プロセスや成果物を変更する場合には、現場の制約条件を考慮する必要がある。そこで我々の3つの開発現場によるシミュレーションにより、「変更要求仕様書」を導入するための運用方法を検討し、「変更要求仕様書」による不具合防止の効果を検証した。その結果、XDDPの部分的な導入の条件を明らかにすることができた。

### Abstract

Most defects found in the test phase are caused in the design specifications, which are documented in the requirements/designing phase of the development process. Recently, to reduce such defects, XDDP (eXtreme Derivative Development Process) has been introduced and it brought many best practices. XDDP consists of three parts: Change Request, Traceability Matrix and Change Specification, and it works most effective when all these three come together as a set. However, in order to adopt them into the project all at once, it requires changing the development process and its derivatives at the same time, and it is not that easy to put in practice.

As we studied on, we noticed that Change Request is closely interlinked with existing specifications documented in the requirement/designing phase, then, we focused on adopting only the change methodology with Change Request. When changing the development process or its deliverables, it is crucial to consider the restrictions of the project. With that context, we developed three simulation patterns based on our projects and sought for ways to adopt Change Request. We verified the effect of preventing defects using Change Request and defined the conditions for adopting XDDP partially.

## 1. 研究動機

我々の開発現場では、検査工程で検出された不具合の大半が「要件定義/方式設計」で作成する設計書に起因するものであり、その対策に多くの手戻りコストが発生している。設計書に起因する不具合が発生する理由のひとつに我々の開発現場は派生開発が多いことが考えられる。一般的に派生開発では「短納期」「低コスト」を要求されているため、設計書の作成やメンテナンスが追いつかないまま次の開発を進めざるを得ない環境であり、このような開発を繰り返すたびに設計書とソースコードとの乖離が広がり、既存のソフトウェアの理解が難しくなる。さらに他人が作った設計書やベースのソースコードを流用するため、部分理解の制約によって「変更要求の解釈間違い」「変更箇所の間違い」「変更方法が不十分」といった問題を誘発している。最近ではこの負のスパイラルを断ち切るために、XDDP[1]を導入した成功事例が多く報告されている。XDDP では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(以下 TM と称する)、変更設計書)がすべて揃うことで最大限の効果を発揮する。しかし XDDP の導入にはいくつかの障壁があると過去の研究論文[2]で報告されていることや、開発プロセスと作成する成果物を同時に変更しなければならず実践は容易ではない。

そこで我々は3点セットの1つである「変更要求仕様書」の仕組みを使い慣れた SLCP-JCF2007 の共通フレーム 2007(以下 SLCP と称する)[3]の成果物に取り入れる方法を研究した。本研究により「要件定義/方式設計」で作成する成果物に「変更要求仕様書」の仕組みを取り入れることで、その仕組みの効果によって早期に設計書の欠陥検出が可能となり、結果として検査工程での設計書に起因する不具合を低減できると考えている。

## 2. 研究目標

本研究の目標は「要件定義/方式設計」で作成する設計書に起因する不具合を低減することである。研究は次の手順で進めた。

- (1) XDDP で用いられる3点セット(変更要求仕様書、TM、変更設計書)の仕組みやその効果について、文献や先行研究を調査し把握する。また我々の開発現場における成果物と何が異なるのかを把握する。
- (2) 我々の開発現場における検査工程で発生している不具合を調査し、その原因を分析する。そしてその不具合が XDDP を導入していれば防止できたか否かについて検証する。
- (3) 3点セットの1つである「変更要求仕様書」の仕組みに着目し、その仕組みを「要件定義/方式設計」で作成する設計書に取り入れることで、設計書に起因する不具合を低減できるか検証を行う。
- (4) 設計書に起因する不具合の未然防止に有効な「変更要求仕様書」の仕組みを抽出する。

なお我々の開発現場では SLCP をベースに開発しているため、SLCP で定義されている「ソフトウェア要件定義」と「ソフトウェア方式設計」の工程において作成される成果物を研究の対象とした。

## 3. XDDP で用いられる3点セットの仕組み

XDDP は設計プロセスを変更プロセスと追加プロセスで区別しており、XDDP で用いられる3点セットは変更プロセスにおける成果物である。本章では XDDP で用いられる3点セットの仕組みや効果について、文献や先行研究を調査し把握する。また我々の開発現場における成果物と何が異なるのかを把握する。

### 3.1. 「3点セット」の仕組みと7つの設計原理による効果

3点セットの仕組みと効果について表1にまとめた。表1からわかるように3点セットは様々な仕組みを持っている。例えば「変更要求仕様書」では「変更要求と変更仕様を階層関係で捉える」(仕組み④)ことで、変更要求全体から階層を下るに従って変更仕様の詳細レベルの理解を可能とし、「仕様を変更の前後(Before/After)で表現する」(仕組み⑤)ことで、仕様の変化を可視化し影響箇所に気付かせることを可能とする。

3点セットの仕組みを「不具合を作り込まないために考慮すべきプログラム構造上の7つのポイント(7つの設計原理[4])」と対比した結果、「階層原理」や「対称原理」等の7つの全ての原理を満たしていることが分かった。このことは要件定義書や基本設計書の読み手の「理解度」を深め、変更箇所の気づきを誘発することで不具合の作り込みを防止する効果があることを示している。また先行研究[5]では、3点セットを導入すると「変更要求の明確化」や「変更の影響範囲の確定」や「不具合発生率の減少」などに効果があると報告されている。

表1 XDDP の3点セットの仕組みと7つの設計原理

XDDP の変更における成果物	説明	仕組み	7つの設計原理
変更要求仕様書 (What, Why)	<ul style="list-style-type: none"> <li>「要求仕様書」としてすべての変更を扱う</li> <li>顧客の要求とそれを実現する仕様について記述する</li> <li>設計終了後にベースの仕様書へマージを行う</li> </ul>	①文書の構成(書式)を定義する ②要求と仕様を明確に分けて記述する ③「理由・背景」を付ける ④「変更要求」と「変更仕様」を階層関係で捉える ⑤仕様を変更の前後(Before/After)で表現する ⑥グループ化でテーマを整える ⑦保守性も要求と捉え記述する	同系原理 単純原理 安全原理 階層原理 対称原理 階層原理 線形原理
TM (Where)	<ul style="list-style-type: none"> <li>変更要求のあった仕様に対応する変更箇所を関数単位で特定する</li> </ul>	①列にモジュールを全て書く ②モジュールと変更仕様の関係を交点で表現する 交点は関数名とする	安全原理 明証原理
変更設計書 (How)	<ul style="list-style-type: none"> <li>どのように関数を変更するのかについて具体的に記述する</li> <li>設計終了後にベースの設計書へマージを行う</li> </ul>	①文書の構成を定義する 関数/データ/マクロの変更を書く 確認(単体テスト)項目を書く ②設計内容の変更を前後(Before/After)で文章表現する	同系原理 対象原理

### 3.2. 「3点セット」の一括導入の必要性について

XDDPはこの3点セットを一括導入することが基本であり、部分導入を推奨していない。その理由は図1に示すように変更要求仕様書からTMに展開することによって、変更する仕様と影響する箇所(関数)の関係が確認でき、また変更要求仕様書と変更設計書をTMにより結びつけることで、変更の意図と変更方法が互いのBefore/Afterで確認できるからである。このことは先行研究[6]の「XDDPの3点セットを用いたレビューにより、レビューの質や効率が向上する」との報告で裏付けられている。また図1の3点セット間の矢印の向きを見てわかるとおり、3点セットの中で変更要求仕様書が一番根元にあり、変更要求仕様書が正しく記述できなければその後のTMや変更設計書が正しく記述できないことを表している。このことから、変更要求仕様書が最も重要な成果物であることがわかる。

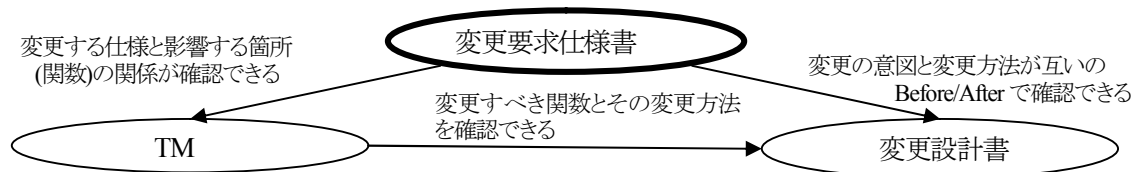


図1 3点セットの関係

### 3.3. 「3点セット」とSLCPの成果物との比較

我々が普段作成しているSLCPの成果物との違いを把握するために、SLCPとXDDPの成果物を比較した結果を表2に示す。この表より、XDDPの変更における成果物はSLCPの複数の成果物を関連付けしており、これが「要求」と「仕様」間の関連が見えないことで発生する不具合を防止する役目を果たす仕組みと考えることができる。

表2 SLCPにおける成果物とXDDPにおける成果物の比較

開発プロセス	XDDPの追加における成果物／ 開発現場の成果物と記述する内容	XDDPの変更における成果物		
		変更要求仕様書 (What, Why)	TM (Where)	変更設計書 (How)
ソフトウェア要件定義	要件定義書	○	○	×
ソフトウェア方式設計	基本設計書			×
ソフトウェア詳細設計	詳細設計書	×		○

※○印はカバー範囲を表す

## 4. 設計書に起因する不具合を低減させる XDDP の仕組みの抽出

XDDP の3点セットの中で「変更要求仕様書」が最も重要な成果物であることがわかったので「変更要求仕様書」の仕組みに着目し「要件定義書/基本設計書」に適用することを以下の手順で試みた。

- (1) 我々の開発現場の検査工程で検出された不具合の原因を調査し、「変更要求仕様書」の仕組みで防止できる原理を抽出する。
- (2) 3点セットの「変更要求仕様書」の仕組みに着目し、その仕組みを我々の開発現場に適した方法で「要件定義書/基本設計書」に取り入れる事例を考案し、各現場適用における制約事項や運用方法を明確にする。
- (3) 考案した方法で我々の開発現場の検査工程で検出された不具合が防止できたか否かを検証する。

### 4.1. 検査工程で発生した不具合分析

我々の開発現場の検査工程で検出された不具合を付録 A に示す。さらに原因別に分析・集計し、原因解析を行った結果を表3と付録 D に示す。これらの結果から、検査工程で検出された不具合は原因工程別で集計すると「要件定義」「方式設計」合計で20件あることがわかり、不具合全体の約70%を占めることがわかった。

表3 「要件定義」「方式設計」で作成する設計書に起因する不具合

No.	上流工程で作成する設計書に起因する不具合原因	XDDP で防止できる原理	件数
1	新規開発踏襲のための変更部分(変更・仕様)が分かりづらく、変更を見落とした	仕様の変更を前後(Before/After)で表現する(対称原理)	13
2	追加・変更する要求や仕様間の矛盾を見落としてしまった	「変更要求」と「変更仕様」を階層関係で捉える(階層原理)	8
3	提示された仕様をそのまま受け取り改造・修正を行い、要求を理解していなかったため、関連する機能への影響を見落とした	要求と仕様を明確に分けて記述する(単純原理)	6
4	ベースとなる製品から流用したが、運用条件を配慮せず変更を行ってしまった	「理由・背景」を付けている(安全原理)	2

表3の集計結果から3点セットの「変更要求仕様書」が不具合を未然に防ぐ効果が高いと判断し、この「変更要求仕様書」の仕組みを使い慣れた SLCP の成果物に取り入れることができれば、設計書に起因する不具合を低減できると考えた。

### 4.2. XDDP の変更要求仕様書の仕組みを取り入れた設計書の試み

本研究では、表4に示す9つのプロジェクトに対し3点セットの「変更要求仕様書」を取り入れる方法を模索した。参考として被験者の業務分野・職種・および経験年数を付録 B に示す。

表4 研究対象としたプロジェクト概要

XDDP	プロジェクト数	プロジェクト全体の規模	新規	変更	流用
未導入	9	100～200KL	10～20KL	10～20KL	100～200KL

※KL…キロステップ数

(1) 運用方法 ※導入ガイドは付録 C に示す。

従来のプロセスでは、図2に示すように追加および変更分の内容を既存の仕様書に追記し作成を行なっている。我々が考案する設計書は図3に示すような成果物を作成し運用する。その手順を以下に示す。

**手順1.** 「ソフトウェア要件定義」「ソフトウェア方式設計」の工程で、追加および変更分の内容を「変更要求仕様書」として作成する。

**手順2.** 品質が安定したシステムテスト終盤に「変更要求仕様書」の内容を「要件定義書/基本設計書」に反映し、「詳細設計書(差分)」の内容を「詳細設計書」に反映する。

**留意点:** 追加および変更分の内容のみを取り扱うとシステム全体が把握できず「部分理解」の状況が発生してしまうため、システム全体を表す図表などの資料を変更要求仕様書に盛り込むこと。

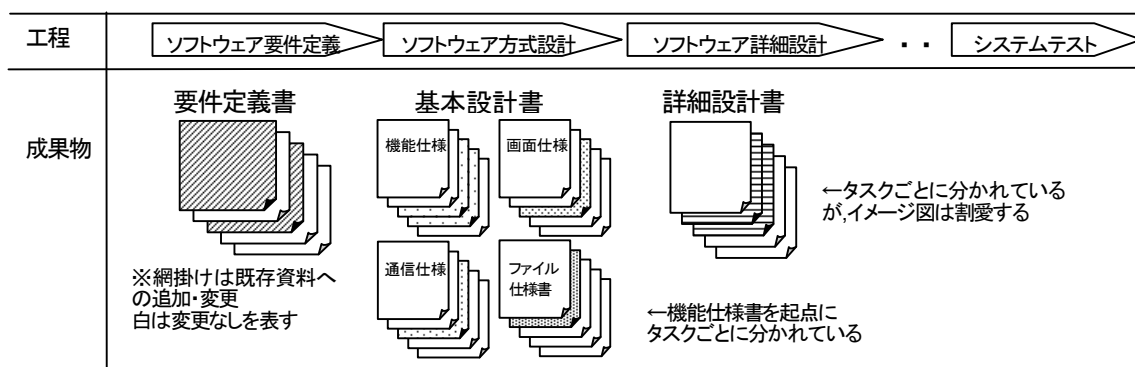


図2 SLCP をベースにした開発プロセスにおける成果物の運用

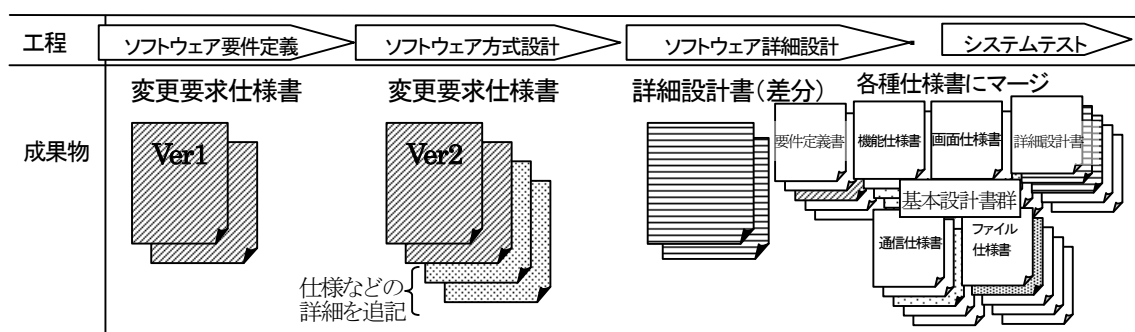


図3 変更要求仕様書を取り入れた成果物の運用

なお変更要求仕様書には、

- ・既存の仕様への影響を検討しやすくするための変更要求と設計書との対応表(表5)
- ・1つの変更要求がどの基本設計書に影響するか、また、モジュールごとに分かれている基本設計書がどの変更要求に関連しているかわかりやすくするための変更要求と基本設計書との TM (表6)

を記載する。この 2 つの表の記述はどちらもソフトウェア要件定義工程から始める。

表5 変更要求と設計書との対応表の記述例

要件定義書			基本設計書		詳細設計書
変更要求(大項目)	変更要求(小項目)	項目 No	変更仕様(概要)	設計書(設計)	変更モジュール
別装置と接続できるように新しい I/F を追加する。  変更理由: ○○○のため。 前提条件: △△△であること。	接続をユーザが任意に止めることができること。	A-1	ボタンを追加する。	画面仕様書	○○○モジュール
		A-2	ボタンにより、他装置へメッセージを送信すること。	通信仕様書	△△△モジュール
	AとBの間の通言を記録できること。	A-3	通言ログを保存する。	ファイル仕様書	□□□モジュール
	エラー発生時にユーザに告知すること。	A-4	警報メッセージを追加する。	警報仕様書	×××モジュール

表6 変更要求と基本設計書の TM の記述例

変更要求/基本設計書	画面仕様書	通信仕様書	ファイル仕様書	警報仕様書
変更要求 A		変更対象	変更対象	
変更要求 B		変更対象		
変更要求 C	変更対象			変更対象

## (2) 変更要求仕様の記述例

変更要求仕様の記述は XDDP の「変更要求仕様書」から要求・理由・仕様を記述するルールのみを引用し、表7のように記述する。記述開始タイミングは(1)～(3)はソフトウェア要件定義工程、(4)はソフトウェア方式設計工程から行う。



表7 変更要求仕様の記述例

(1) 要求	<before>従来の3種類の保温設定に、煎茶モードを追加し、<after>4種類の保温設定を可能とする。 従来の3種類:高温モード=98℃,節約モード=90℃,ミルクモード=60℃
(2) 要求の理由・背景	煎茶をおいしく飲むには、80℃のお湯が良いとされている。 他社機でもこのモードのサポートが行われ、弊社のポットだけサポートできていない。
(3) 要求の補足説明	特になし
(4) 仕様	a)煎茶モードが選択されたときは、80℃で保温する。 b)保温ボタンを1回押すごとに、保温モードを<before>高温→節約→ミルクから、<after>高温→節約→煎茶→ミルクに変更する。

### 4.3. 検証結果

表8に示す各現場の制約条件と運用方法において、4.2章の「変更要求仕様書」の仕組みを「要件定義書/基本設計書」に取り入れることで表3に示した問題を未然防止できたか否かを、3つの現場におけるシミュレーションにより検証した。そこで得られた評価結果を現場の事例別に表9に示す。

表8 各現場の制約条件と運用方法

現場	制約条件	運用方法
現場1	「要件定義/方式設計」で作成する成果物を統合した変更要求仕様書の作成が可能なプロジェクト	「要件定義/方式設計」に3点セットの「変更要求仕様書」の仕組みを取り入れる方法を用いる
現場2	・「要件定義書」はユーザと取り交わすドキュメントのため変更できないプロジェクト ・「方式設計」で作成する成果物のみ変更できるプロジェクト	「基本設計書」のみに3点セットの「変更要求仕様書」の仕組みを取り入れる方法を用いる
現場3	組織内の標準プロセスや顧客との慣例などにより、XDDPの成果物をそのまま導入することができないプロジェクト	「要件定義書/基本設計書」の両方に3点セットの「変更要求仕様書」の仕組みを取り入れる方法を用いる

表9 各事例における評価結果

事例	利点	欠点	期待する導入効果	評価結果
事例1 (現場1)	変更要求による基本設計書や詳細設計書の変更箇所が明確になるため、影響範囲を特定することができる	「要件定義書」と独立しているため、要求の追加時は「要件定義書」と図3の「変更要求仕様書」を共に変更しなければならない	レビュー時に影響範囲漏れに気づきやすくなることや、「変更理由、背景」を把握できることで、「変更要求仕様」と「テスト仕様」の同時設計の効果[7]を向上させることができる	高 (16件)
事例2 (現場2)	従来のツールを使うので、既存の文書へのマージの負担を少なくできる	「変更要求仕様書」は、基本的に差分だけの記述となるため、この文書だけで変更によるシステム全体への影響を把握することが難しい	「要件定義書/基本設計書」を統合することで、要求と仕様を1つの成果物で見ることができ、可読性やレビュー効率も向上する	中 (11件)
事例3 (現場3)	表現方法を変更するだけなので、従来の成果物をそのまま流用することができる	「要件定義書」と「基本設計書」が独立した成果物のため双方を合わせて、要求と仕様を理解せざるを得ない	要求と仕様をトレースできるので、要求を理解して設計、実装を行うことはできる。しかし、「要件定義書」と「基本設計書」セットで要求と仕様を理解する必要があるため、メンテナンスは容易ではない	低 (3件)

※括弧内の件数は、検査工程で発生した不具合防止件数を示す。

上記評価結果から事例1の導入効果が最も高いことがわかった。この事例1の適用前後における不具合件数の比較を表10に示す(詳細は付録D参照)。この結果は3点セットの「変更要求仕様書」の仕組みを設計書に取り入れることで、「要件定義/方式設計」で作成する設計書に起因する不具合が20件から4件へと大幅に改善できることを示している。

表10 適用前後の「要件定義書/基本設計書」に起因する不具合件数の比較

方法	SLCP の成果物		合計
	要件定義書	基本設計書	
従来(SLCP)の成果物(XDDPを導入していない現場)	7 件	13 件	20 件
XDDP の仕組みを取り入れた SLCP の成果物 (事例1)	1件	3 件	4 件

導入効果について研究員にヒアリングした結果,変更の前後(Before/After)で表現することや「変更要求」と「変更仕様」を階層関係で捉えることが効果的であるという回答が多かった. 具体的に検査工程で検出された不具合5件をピックアップし,事例1を用いた結果得られた効果を表11に示す. この内容から表1で述べた変更要求仕様書の仕組みはほぼ満たしており,導入効果を得ることは可能であると判断した.

表11 3点セットの「変更要求仕様書」の仕組みを取り入れた設計書の効果

No.	具体例	効果のあった7つの設計原理[4]	得られた効果
1	従来は変更後の要求だけで検討していたため,要求の範囲について顧客と認識違いを起こしていたが Before/After により,従来実現していた要件のどこを変えて欲しいか特定されるので,変更の規模や難易の状況も推定しやすくなり,精度の高い見積もりが可能となる	・仕様の変更を前後(Before/After)で表現する (対称原理)	変更範囲の見積り精度が高くなる
2	印刷のタイミングをAからBに変更するという仕様だけでは関連する機能への影響がわからないため,変更箇所に漏れが発生したが,要求と関連付けることで関連する機能への影響に気づくことが可能となる	・「変更要求」と「変更仕様」を階層関係で捉える. (階層原理) ・仕様の変更を前後(Before/After)で表現する (対称原理)	関連する機能への影響を見落とさなくなる
3	要求と仕様のトレースが容易になったため,テスト設計時に要求から関連する機能のテストケースを洗い出すことが可能となる また Before/After により,変更部分のテストと変更しなかった部分の回帰テストを明確に分けてテスト設計することが可能となる	・要求と仕様を明確に分けて記述する (単純原理) ・「理由・背景」を付けている (安全原理) ・仕様の変更を前後(Before/After)で表現する (対称原理)	テスト範囲が特定でき効果的な機能検証が実施できる
4	今までの方式設計のレビューは「その設計で機能が正しく動作するか」ということしかレビューされていなかったが,背景や理由を意識することで,顧客が要求する機能がそれで満足できるかをレビューすることが可能となる	・要求と仕様を明確に分けて記述する (単純原理) ・「理由・背景」を付けている (安全原理)	顧客要求を正しく方式設計時に盛り込むことが出来る
5	「文字の大きさを8ポイントから14ポイントに変えて欲しい」という要求の背景が,意を引くためとすると,注意を引くために文字の大きさだけでよいのか,文字の色を変えた方がいいのではないのかという実現手段の妥当性を議論することが可能となる	・「理由・背景」を付けている (安全原理)	要求の本質を捉えることができ,実現手段の妥当性を議論できる

#### 4.4. 考察

「要件定義書/基本設計書」に起因する不具合を未然防止に有効な「変更要求仕様書」の仕組みは,以下3つである.

(1) 変更の前後(Before/After)で表現すること

変更の前後(Before/After)の変化が見えることで変更部分が理解しやすくなり,変更範囲の判断ミスが起きにくくなったことが考えられる. 変更を見落とす不具合には有効であることがわかった.

(2) 「変更要求」と「変更仕様」を階層関係で捉えること

表5のように変更に関わる要求と仕様とモジュールが見えることで,お互いのつながりがわかり易くなる. そのため工程間の作業が漏れにくくなり,表6のように変更要求と基本設計書の関係が見えることで考慮すべき影響範囲がわかるため,検討の漏れが発生しにくくなることが考えられる. 追加・変更する「要求と仕様の矛盾」や「仕様間の矛盾」の不具合には有効であることがわかった.

(3) 「要求」と「仕様」を明確に分けて記述すること

表5のように要求と仕様を各項目に分けていることや,要求の理由がわかることでその意図が設計書を読む

人に的確に伝わるため、実現内容に間違いがなくなることが考えられる。要求と仕様が区別できにくいことで関連する機能を見落とす不具合には有効であることがわかった。

上記の仕組み(1)～(3)が全て適用できた事例1と事例2は導入効果が高いことがわかった。しかし、(1)しか適用できなかった事例3は導入効果が低く、個々の要素を取り入れるだけでは不十分であることもわかった。また、TMの仕組みがないことにより未然防止ができない不具合もあった。

以上より、表4のようなプロジェクトのうち、現場1や現場2のようなケースにおいては、本研究で考案したアイデアによりXDDPのシームレスな導入が可能と判断する。

## 5. まとめ

### 5.1. まとめ

XDDPで用いられる3点セットの仕組みを調査した結果、3点セットで最も重要な成果物が「変更要求仕様書」であり、それぞれの持つ役割を補完し合うことで効果を最大限に発揮することもわかった。この調査結果を活用して考案した「変更要求仕様書の仕組みを使い慣れたSLCPの成果物に取り入れたアイデア」により、「要件定義/方式設計」で作成する設計書に起因する不具合を低減することができ、手戻りコストも低減できることがわかった。「仕様の変更を前後(Before/After)で表現すること」「変更要求と変更仕様を階層関係で捉えること」「要求と仕様を明確に分けて記述すること」といった基本的な作法をきちんと理解した上でXDDPの成果物を使うことが派生開発で成功する施策である。

本アイデアは既存の設計書の改善という低いハードルで実現でき、既存プロセスの変更を伴わないといった利点がある。既存の関連ドキュメントにきちんとマージするプロセスを遵守さえすれば、XDDPの導入を検討している開発現場へのXDDPのシームレスな導入がさらに容易になると考えられる。

### 5.2. 今後の課題

「変更要求仕様書」の仕組みを取り入れた設計書によって、設計書に起因する不具合を低減できたが、下記のような課題も残った。

- ・TMの仕組みがないことで防止できなかった不具合もあった。
- ・TMがなくても成果が得られたということは、影響の範囲があまり散らばらない状況であったとも考えられる。
- ・影響範囲の定量的な把握を検証することができなかった。

今後はTMの仕組みの取り入れ方法や影響範囲の定量的な把握を検討し、稼働プロジェクトに適用して実績を積み上げより効果的な適用ノウハウを研究していきたい。

## 参考文献

- [1] 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
- [2] 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
- [3] 共通フレーム2007(第2版),独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター編,オーム社
- [4] ソフトウェア品質知識体系ガイド ―SQuBOK Guide―,SQuBOK策定部会,オーム社
- [5] ソフトウェアの改造で悩んでいませんか? ～派生開発の品質と効率の向上を目指して～,伯田 誠,派生開発カンファレンス2010
- [6] 成果物にこだわるとレビューが変わる,加藤 由之,ソフトウェア品質シンポジウム
- [7] 設計者自身による設計品質作り込み,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
- [8] 組込みソフトウェア管理者・技術者育成研究会「話題沸騰ポット要求仕様書」



## 付録 A 不具合事例分析で使用した不具合事例

表 A-1 XDDP により防止できたと判断した不具合事例

No.	不具合内容	原因工程	解決する XDDP の成果物
1	未対応の情報を入力すると,Runtime Error になる.	要件定義	変更要求仕様書
2	停止中のシステムに対してデータ取得を行なうと,Runtime Error が表示される.	要件定義	変更要求仕様書
3	あるソフトウェアで出力したファイルをそのソフトウェアの派生ソフトウェアに入力した場合に仕様の不一致によりエラーが発生する.	要件定義	変更要求仕様書
4	ファイルに#が記述された場合に,コメント扱いにならず,エラーが発生する. そのソフトウェアの派生ソフトウェアでは,#はコメント扱いのため,開発対象のソフトウェアでも同様にコメントとして使用することが求められる.	要件定義	変更要求仕様書
5	通話中のセッション A を変更できない	要件定義	変更要求仕様書
6	通話中のセッション B を変更できない	要件定義	変更要求仕様書
7	XML データが,ヘッダのみの場合,ヘッダ情報がセーブされない. (ヘッダ+コンテンツの場合はセーブ OK)	方式設計	変更要求仕様書
8	指定したデータが出力されない. (出力データ不正)	方式設計	変更要求仕様書
9	複数範囲のデータを指定した場合,データ表示されるまで多大な時間がかかった.	方式設計	変更要求仕様書
10	START および STOP コマンドで同一データを指定すると,指定したデータと,その次のデータが出力される.	方式設計	変更要求仕様書
11	秒の表記が,大文字の S 表記と小文字の s 表記混在で出力される.	方式設計	変更要求仕様書
12	あるタイミングで印刷した場合,指定した順番に印刷されない. (印刷順不正)	方式設計	変更要求仕様書および TM
13	1 ページに収まるデータにも関わらず,2ページ出力された. (改行位置不正)	方式設計	変更要求仕様書および TM
14	処理不一致によりデータ未送信発生.	方式設計	変更要求仕様書および TM
15	画面にて,金額を表示する部分が,「0」で表示されている	方式設計	変更要求仕様書および TM
16	画面から追加登録されたデータが,重複して連動された.	方式設計	変更要求仕様書および TM
17	過去日付のデータで承認依頼取消を行うと承認依頼できない	方式設計	TM
18	メインメニュー画面中のボタン名を変更したが,画面中の見出しには,古いボタン名が表示されている.	詳細設計	変更設計書
19	画面上の項目が未入力であるエラーメッセージの文言が, 画面によってことなる.	詳細設計	変更設計書
20	マスタメンテナンスで,「表示しない」設定にした項目が画面のリストに表示される.	詳細設計	変更設計書
21	計算方法が間違っている	詳細設計	変更設計書

表 A-2 XDDP で防止できないと判断した不具合事例

No.	不具合内容	原因工程
1	エラー・メッセージの内容からエラーになった原因を読み取れない.	要件定義
2	一定時間帯にログ監視が稼動しない	方式設計
3	日跨ぎのジョブがスケジューラから稼動出来ない	方式設計
4	条件1の場合,〇〇中かつ〇〇中に,メニューを開いたら,プログラムエラーが発生	詳細設計
5	WEB 画面から実施するバッチ処理が,他のユーザ様とバッティングした際に,画面側のエラーハンドリングとして「システムエラー」として画面を閉じる仕様になっていた.	詳細設計
6	黒データと赤データについて画面表示がおかしい.	詳細設計
7	通話履歴情報が表示されない	詳細設計
8	マスタメンテナンスの一覧項目がずれている	実装
9	設計書は,項目を 64 バイトづつ分割とあるが,64,65 バイトが全角文字となった場合の処理が不明. CSV データでは,64,65 バイトが全角文字の場合,文字が正しく分割されていない.	実装

表 A-3 XDDP の変更要求仕様書の仕組みを SLCP の成果物に取り入れたことにより防止できた不具合事例

表 A-1 No.	表 A-1 の不具合事例を引き起こした原因	XDDP の変更要求仕様書の仕組みで防止できる原理			
		仕様の変更を前後 (Before/After)で表現する (対称原理)	「変更要求」と「変更仕様」を階層関係で捉える (階層原理)	要求と仕様を明確に分けて記述する(単純原理)	「理由・背景」を付けている (安全原理)
1	本件は仕様書に書くレベルのことではないと思い込んでいた。 テストおよび実装で現象出しと対策ができる範囲と思い込んでいた。	○	○		
2	起動中と実行中は区別できていると思い込んでいた。	○	○		
3	本件は仕様書に書くレベルのことではないと思い込んでいた。	○	○		
4	関連するソフトの動作について考えていなかった。	○	○		
5	変更されていないと思い込んでいた。	○			
6	変更されていないと思い込んでいた。	○			
7	データのパターンが,ある1パターンだけと思い込んでいた。		○		
8	変更すべき点は修正できたが,他機能に影響を受けているとは思わなかった。	○			
9	扱うデータ数が増えたことは認識していたが,処理時間過多となることに「気づけ」なかった。				○
10	本件は仕様書に書くレベルのことではないと思い込んでいた。		○	○	
11	単位について考えていなかった。	○	○		
12	複雑なモジュールであったが,変更内容は容易であった。また,影響する範囲も特定できていた。 そのため,ここを変更すればよいという風に考えてしまった。	○		○	
13	複雑なモジュールであったが,変更内容は容易であった。また,影響する範囲も特定できていた。 そのため,ここを変更すればよいという風に考えてしまった。	○		○	
14	互いにコンフリクトがあると思わなかった。そんな条件があることを想定できなかった。	○	○	○	○
15	有識者が影響無いと思い込んで発言した内容をそのまま設計に採用してしまった。	○		○	
16	有識者が影響無いと思い込んで発言した内容をそのまま設計に採用してしまった。	○		○	
17	影響範囲を詳細に調査しなくても,実装時に対応出来ると思い込んだ。				
18	ボタン名が別の箇所の文言で使用されているとは思っていなかった。		この5件は,XDDP の TM や変更設計書で解決できると判断したものである		
19	ボタン名が別の箇所の文言で使用されているとは思っていなかった。				
20	設計書に記載しなくても,他と同様の実装をすればよいと思い込んだ。				
21	四則演算について常識の範囲であり,機能設計に表す必要がないと認識していた。				

## 付録 B 被験者の業務分野／職種と経験年数

表 B-1 業務分野／職種の分類

業務分野	職種	人数
エンタープライズ系	SE	3
組込み系	SE	1
	SEPG	1

表 B-2 経験年数の分類

経験年数	人数
5年以上～10年未満	3
10年以上～15年未満	1
15年以上	1

## 付録 C 変更要求仕様書導入ガイド[8]

### 目次

1. はじめに.....	1 3
2. 成果物の運用 .....	1 3
3. 変更要求仕様書の構成とルール.....	1 4
3.1. 変更要求仕様書の構成 .....	14
3.2. 変更要求仕様書の記述ルール .....	15
4. 変更要求仕様書の記述例 .....	1 5

### 1. はじめに

本書は、本研究で提案した「従来の成果物に変更要求仕様書を取り入れる仕組み」について示したものである。

### 2. 成果物の運用

変更要求仕様書を取り入れた場合の成果物の運用方法の違いを図 C-1 および図 C-2 に示す。

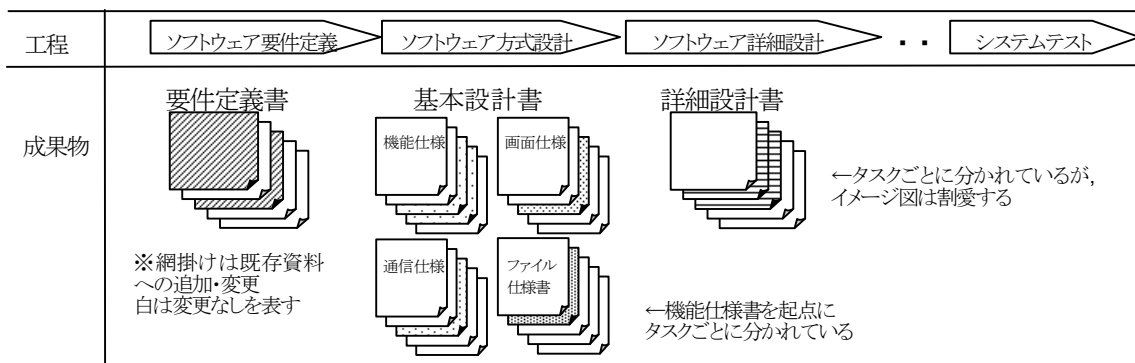


図 C-1 既存の開発プロセスにおける成果物の運用

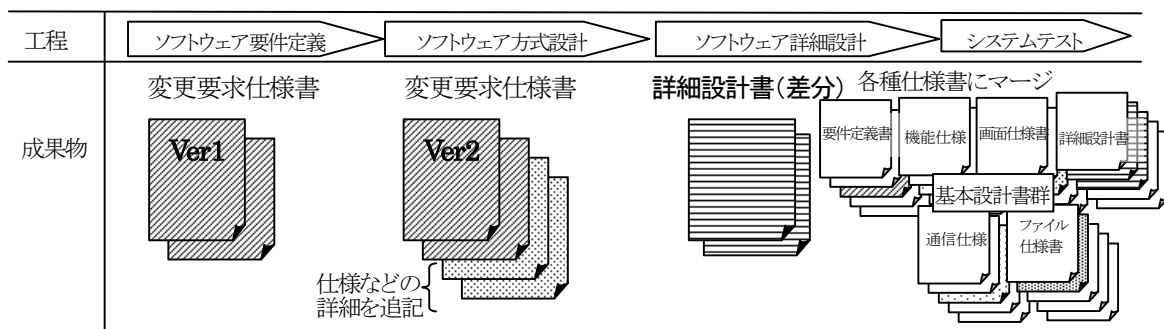


図 C-2 変更要求仕様書を取り入れた成果物の運用

図 C-1 に示すように、従来のプロセスでは、追加および変更分の内容を、既存の仕様書に記述して成果物の作成を行なっている。



図 C-2 に示すプロセスでは、「ソフトウェア要件定義」および「ソフトウェア方式設計」の工程では、追加および変更分の内容を「変更要求仕様書」として作成する。品質が安定したシステムテスト終盤に、「変更要求仕様書」の内容を「要件定義書」および「基本設計書群」に反映し、「詳細設計書(差分)」の内容を「詳細設計書」に反映する。

変更要求仕様書には、既存の仕様への影響を検討しやすくするために、元の書類との関連付けを表 C-1 のように記載する。記述開始タイミングは、ソフトウェア要件定義工程から行う。

表 C-1 変更要求仕様書の記述例

要件定義書			基本設計書		詳細設計書
変更要求(大項目)	変更要求(小項目)	項目 No	変更仕様(概要)	設計書(設計)	変更モジュール
別装置と接続できるよう新しい I/F を追加する。  変更理由: 〇〇〇のため。 前提条件: △△△であること。	接続をユーザが任意に止めることができること。	A-1	ボタンを追加する。	画面仕様書	〇〇〇モジュール
		A-2	ボタンにより、他装置へメッセージを送信すること。	通信仕様書	△△△モジュール
	AとBの間の通信を記録できること。	A-3	通信ログを保存する。	ファイル仕様書	□□□モジュール
	エラー発生時にユーザに告知すること。	A-4	警報メッセージを追加する。	警報仕様書	XXX ファイル

また、1つの変更要求がどの基本設計書に影響するかわかりやすくする(行方向)ためと、タスクごとに分かれている基本設計書がどの変更要求に関連しているかわかりやすくする(列方向)ために表 C-2を作成する。記述開始タイミングは、ソフトウェア要件定義工程から行う。

表 C-2 変更要求仕様書と基本設計書の TM の記述例

変更要求/基本設計書	画面仕様書	通信仕様書	ファイル仕様書	警報仕様書
変更要求 A		変更対象	変更対象	
変更要求 B		変更対象		
変更要求 C	変更対象			変更対象

なお、追加および変更分の内容のみを取り扱うとシステムの全体が把握できず、「部分理解」の状況が発生してしまうためシステムの全体を表す図表などの資料を盛り込むこととする。

### 3. 変更要求仕様書の構成とルール

#### 3.1. 変更要求仕様書の構成

変更要求仕様書は「要件定義書」および「基本設計書」に記述すべき内容で変更がある部分に対してのみ記述する。

要件定義書	基本設計書(機能仕様書の例)	変更要求仕様書
1. 概要 2. ハードウェア説明 ・ハードウェアブロック図 3. ソフトウェア説明 ・機能一覧 ・ユースケース ・制約条件 ・モジュール構成(大) 4. その他 ・性能・品質など非機能要求詳細	1. 概要 2. ハードウェア説明 ・ハードウェアブロック図 ・ハードウェア制御方式 ・タイミングチャート 3. ソフトウェア説明 ・状態遷移図、フローチャート ・画面遷移図 ・モジュール構成(中) ・メッセージ制御シーケンス ・ユースケースとモジュール構成の対応 ・モジュール間のインタフェース説明 4. システムで扱うデータ 5. 例外一覧	基本的には要件定義書や方式設計書に変更がある部分を書く。

このとき変更部分の記述の際は次の項目を必須とする。

- (1) 要求
- (2) 要求の理由・背景
- (3) 要求の補足説明
- (4) 仕様

記述タイミングは(1)～(3)はソフトウェア要件定義工程,(4)はソフトウェア方式設計工程から行う。

## 3.2. 変更要求仕様書の記述ルール

以下に,変更要求仕様書の記述ルールとして各項目の記述で抑えるべきことについて示す。

- ・ 要求と仕様は,変更の前後(before/after)で表現すること
- ・ 要求は動詞で表現する。
- ・ 要求の範囲を分割・階層化して要求の範囲を整理する。
- ・ 個々の要求の中で仕様を捉える。
- ・ 要求と理由は,一般に要求の記述の方が長くなる。逆の場合は理由の中に要求が混じっていないか確認する。
- ・ 説明は仕様や動きを補足するものであって,ソースコードにはならない。
- ・ 要求の細分化は時系列に行う。
- ・ 仕様は必要に応じグループ化(時系列・構成・状態)する。
- ・ 理由は最初にムリに書かなくてもよい。後で書けばよい。
- ・ 要求の補足説明は書けない場合は無理に書かない。ただしなときは「なし」と書き,意思表示をする。(意識的に書いてないのか,書き忘れかを表す)
- ・ 仕様にも必要なら【説明】や【理由】をつける
- ・ 理由は要求の裏返しにならないようにする。最後の動詞の理由を書かない。
- ・ エラーや例外処理は仕様に書く
- ・ 仕様であることの条件は,設計の実現の可能性が見える+検証方法がイメージできる
- ・ ペースト作文は避ける。
- ・ 曖昧な表現を避ける。
- ・ 否定表現があった場合は,その反対,肯定動作の仕様が漏れやすい。
- ・ 非機能要件も仕様として記述しないと実現しない と思え。

## 4. 変更要求仕様書の記述例

以下に変更要求仕様書の記述例を示す。

### 記述例

### 1. 概要

今回設計するポットは,AAA-1 型に以下の機能を追加した家電製品である。

- ・煎茶モードの追加
- ・光センサの追加

以下に変更要求とその仕様とモジュールの関係表と変更要求と基本設計書の関係表を示す。

変更要求とその仕様とモジュールの関係表

要件定義書			基本設計書		詳細設計書
変更要求(大項目)	変更要求(小項目)	項目No	変更仕様(概要)	設計書(設計)	変更モジュール
煎茶モードの追加 変更理由: おいしくお茶がすば やく飲めるようにする ため。	80度を実現すること。	A-1	温度制御の追加	温度制御仕様書	温度制御モジュール
	現在のモードがわかる こと	A-2	操作パネルの表示追加	操作パネル表示 仕様書	操作パネル表示モジ ュール
	モード選択ができること	A-3	ボタン制御の変更	キー制御仕様書	キー制御モジュール
省電力モードの追加 変更理由: 他社機との差別化	省電力→通常モード 遷移における温度制御 ができること	B-1	温度制御の条件変更	温度制御仕様書	温度制御モジュール
			省電力モードの判断	光センサ制御仕 様書	ECO モジュール

変更要求と基本設計書の関係表

変更要求/基本設計書	温度制御仕様書	操作パネル表示仕様書	キー制御仕様書	光センサ制御仕様書
A-1	変更対象			
A-2		変更対象		
A-3			変更対象	
B-1	変更対象			新規追加

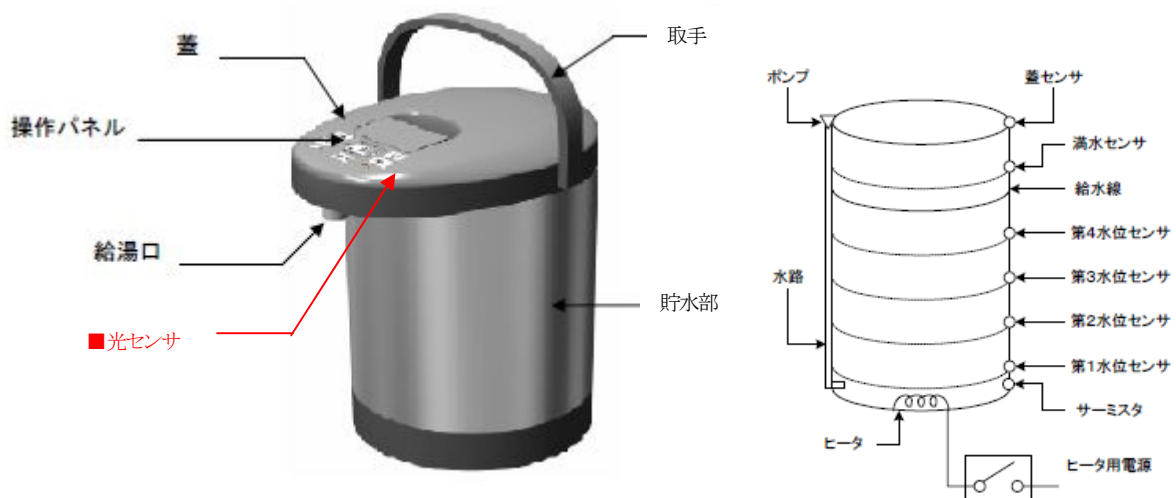
以降の章では、上記追加機能の詳細を説明する。

なお、ベースの要求仕様書については、組込みソフトウェア管理者・技術者育成研究会「話題沸騰ポット要求仕様書(第6版)」を参照のこと(※写真や図はこれを一部引用しています)。

## 2. ハードウェア説明

### 2.1 外形

対象とするポットは以下に示すような概観になっている。追加したものを■で示す。



ここで追加された光センサの意味は次のとおりである。

#### ・光センサ

外の明るさを検出する。1ルクス以上になると ON となる。

#### 【A. 光センサの説明】

##### (1) 要求

光センサの ON/OFF を検知し、ON になったら操作パネルで設定された保温温度にする。

##### (2) 要求の理由・背景

夜間や不在時の消費電力を抑えるため。

##### (3) 要求の補足説明

1ルクスというのは、月明かり程度の明るさで通常この明るさでのポットの使用はないと考えている。

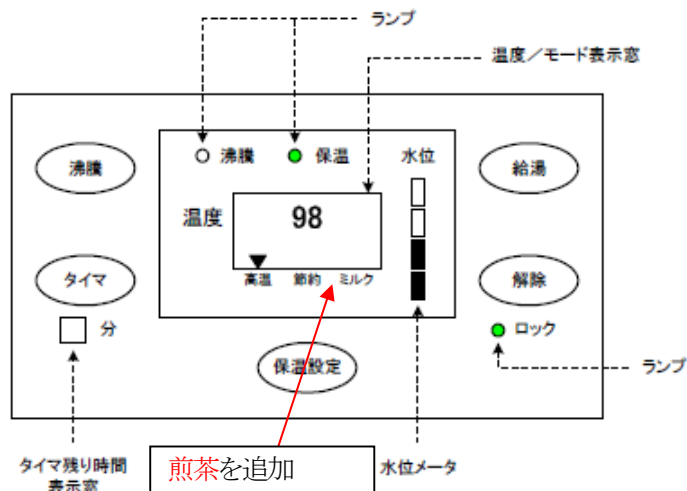
##### (4) 仕様

何ルクスで光センサを ON/OFF するかは、光センサの設定で変更可能とする。

※光センサの使用については、●●メーカーの光センサ設定仕様書を参照のこと。

### 2.2 操作パネル部

ポットの上部にある操作パネルは以下のような構成になっている。



ここで追加された煎茶モードは、保温設定ボタンを押すことで選択が可能となる。

### 【B. 保温設定ボタンの詳細説明】

#### (1) 要求

<before>従来の3種類の保温設定に煎茶モードを追加し,<after>4種類の保温設定を可能とする。

従来の3種類:高温モード=98℃,節約モード=90℃,ミルクモード=60℃

#### (2) 要求の理由・背景

煎茶をおいしく飲むには80℃のお湯が良いとされている。

他社機でもこのモードのサポートが行われ、弊社のポットだけサポートできていない。

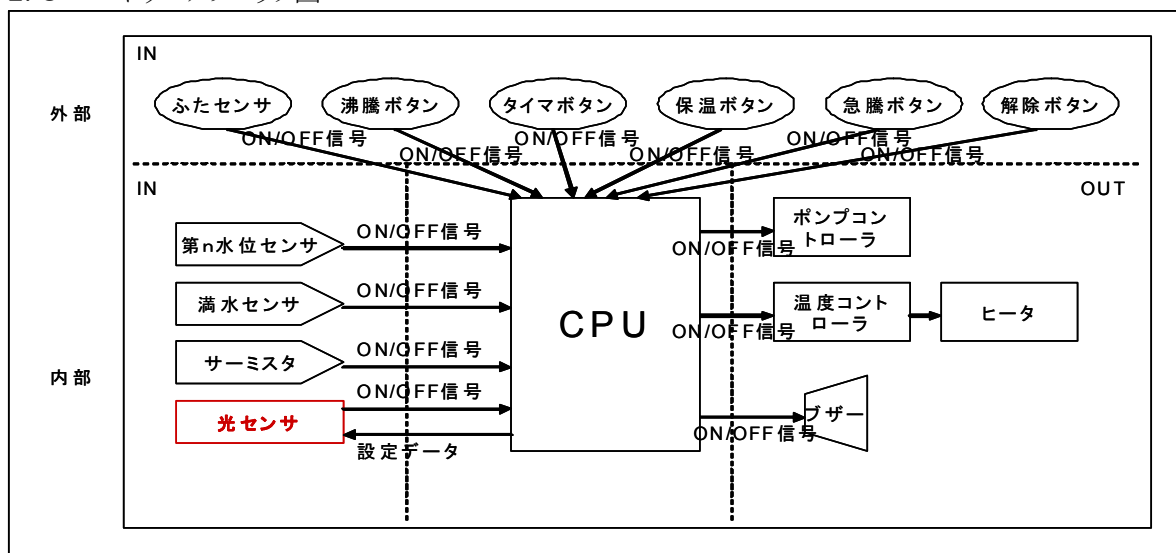
#### (3) 要求の補足説明 特になし

#### (4) 仕様

a) 煎茶モードが選択されたときは80℃で保温する。

b) 保温ボタンを1回押すごとに、保温モードを<before>高温→節約→ミルクから,<after>高温→節約→煎茶→ミルクに変更する。

## 2.3 ハードウェアブロック図



### 3. ソフトウェア説明

#### 3. 1 煎茶モードの追加

##### (1) 要求

従来の3つのモード(高温 節約 ミルク)に煎茶モード(80℃)を追加する.

##### (2) 要求の理由・背景

お茶をおいしくいただけるようにするため.

##### (3) 要求の補足説明

なし

##### (4) 仕様

煎茶モードでは水温を80℃に保つように,PID 制御方式でヒータを制御する.



## 付録 D 表10と付録 A の表との関係

付録 A のうち「要件定義書」と「基本設計書」に起因する不具合を抜粋し、それぞれの方法でも解決できないだろう(○印)と判断した元データを以下に示す。

表 D-1 表10と付録 A の表との関係

表	No	表10					
		各方法における「要件定義書/基本設計書」に起因する不具合件数の比較					
		従来の成果物 (XDDP を導入していない現場)		XDDP の仕組みを取り入れた 成果物 (事例1)		XDDP の3点セットを 導入した場合	
		要件定義書	基本設計書	要件定義書	基本設計書	要件定義書	基本設計書
表 A-1	1	○					
	2	○					
	3	○					
	4	○					
	5	○					
	6	○					
	7		○				
	8		○				
	9		○				
	10		○				
	11		○				
	12		○				
	13		○				
	14		○				
	15		○				
	16		○				
	17		○		○		
表 A-2	1	○		○		○	
	2		○		○		○
	3		○		○		○
表10の 件数		7	13	1	3	1	2
		20		4		3	