

# 情報システム開発における設計工程への標準化導入事例

大 宮 望

## 1. はじめに

情報システムは現在様々な企業に導入され、なくてはならない存在となってきた。しかし、情報システムの開発における、品質や生産性向上については課題があり解決が望まれている。情報システム開発における品質及び生産性向上へ寄与する要因の1つとして、開発方式が挙げられる。情報システムの開発方式は、菅野（1992）によると一般的に情報システム開発の生産物であるソフトウェアは、他の製品と同様に工場生産方式と設定することが一般的な傾向である、と述べている。また、野口（1990）も同様にソフトウェアの開発とハードウェアの生産について比較し、その違いについて述べている。このように情報システム開発は、ハードウェアの生産と同様な開発方式を採用しているが、いまだハードウェアと同等な品質、生産性向上が得られたとはいえない。そこで、はじめにソフトウェアの開発方式とハードウェアの生産方式の違いを整理し、情報システム開発における開発方式の問題点を明らかにする。

まず、ソフトウェアとハードウェアの生産物の違いについて明らかにする。情報システムの生産物は菅野（1992）によるとプログラムとドキュメントの2種類であると述べている。このプログラムとドキュメントには表1に示す様々な種類が存在している。同様に野口（1990）も、ソフトウェアはプログラムや設計書などのドキュメントに表されると述べている。一方ハードウェアにおいて

も、製品と図面などのドキュメントの2種類であり、ソフトウェアとハードウェアの違いはないと述べている。しかし、生産方式では違いがある。野口（1990）はハードウェアでは言語構造が問題にはならない。しかし、ソフトウェアでは、人工言語そして自然言語が問題となる。つまり、ハードウェアは人間が介在し、直接生産するが、ソフトウェアは人間が自己の知的独創力を発揮し開発したものをソフト生産物として機械と結合して開発する点が異なる。さらに、ハードウェアは物的生産物として表される。しかし、ソフトウェアは無形の生成物であるプログラムや設計書などのドキュメントに表され、それゆえ標準化と部品化に至る過程も違い、ハードウェアでは、物的生産物のプロセスが重要であるのに対して、ソフトウェアは着想とシステム設計の段階が重要である、と述べている。このようにソフトウェアとハードウェアでは生産方式が大きく異なることが判る。次に、野口（1990）が述べている、標準化と部品化の過程の違いについて明らかにする。まず、標準化は段階的に行われていくのが一般的であり、ハードウェアとソフトウェア標準化の段階の違いは野口（1990）によると以下の通りである。

### A. ハードウェアの標準化の段階

- A-1 作業の標準化
- A-2 工程の標準化
- A-3 製造方法の標準化
- A-4 生産方式の標準化

表1. プログラムとドキュメント

生産物の分類	分類階層1	分類階層2
ドキュメント	仕様書	基本仕様
		機能仕様
		設計仕様
		製造仕様
		検査仕様
	設計資料	運用取扱説明
保全取扱説明		
その他ドキュメント		
プログラム	ユーザプログラム	
	オペレーティングシステム	制御プログラム
		言語処理プログラム
		ユーティリティプログラム
	応用プログラム	
	治工具	
	試験プログラム	
検査プログラム		

B. ソフトウェアの標準化の段階

- B-1 部分的標準化
- B-2 ある程度の標準化
- B-3 工程の標準化
- B-4 ソフト製造方法の標準化, 生産方式の標準化

このように標準化の段階においても、ソフトウェアとハードウェアの標準化の段階が異なることが判る。しかし一般的な、標準化におけるねらい、対象業務及び利益について緒方（1990）は下記のように述べている。標準化によるねらいでは表2に示す通り、生産性の向上安定、トラブル一掃などと述べており、ソフトウェアと同様と考える。次に、表3に示す標準化の対象業務では、繰り返し頻度の高いもの、不良等の発生頻度が高いものなどと述べており、これもソフトウェアと

同様であると考えられる。最後に表4に示す標準化による利益であるが、これは利益を得る対象部門が生産、設計に分けられている。情報システム開発では、生産はプログラミング工程、設計は、設計工程と読み替えることによって情報システム開発と対比できる。緒方（1990）は標準化に伴う利益について、図面の標準化により図面が見やすくなり誤解が減るなどと述べており、これは図面を設計書に置き換えるとソフトウェアでも、同様であると考えられる。上述した通り、一般的な標準化はソフトウェアで目指している標準化の目的と合致していると考えられる。しかし、ソフトウェアでこのような一般的な標準化がすべて適用されているとはいえない。これは、上述のようにソフトウェアは生産物が、知的生産物であり、物的生産物ではなくプログラムや設計書といったドキュメントが生産物となる違いがあり、ハードウェアと

表2. 標準化のねらい

項番	標準化の狙い
1	互換性, 少数化, 貿易, 安全
2	生産性の向上安定, 再発防止, コンピュータ化, 技術の蓄積
3	トラブル一掃, 自動化, 教育, 作業の指導書
4	モラルの向上, 商取引, 伝達手段, 品質の安定向上
5	コストダウン, 改善, 量産, 審査用
6	共同会社の利益, 品質管理の基本, 単純化

表3. 標準化の対象業務

項番	標準化の対象業務
1	繰り返し頻度の高いもの 例：作業標準, 伝票処理手続き, 工具貸し出し
	系列化できるもの 例：部品規格, コード化, モジュール
3	需要量の多いもの 例：製品仕様, 購入仕様, 在庫管理規定
	同じような手続き, 方針, 手法, 手順が多くの人々によって繰り返し行われるもの 例：各種伝票処理手続き, 事務標準, EDP化のための手法統一
5	不良, 障害, クレームなどの発生頻度の高いもの 例：クレーム処理規定, 作業基準, 検査規格, 設計基準

同等な標準化が行いにくく、独特な標準化が必要とも考えられる。情報システム開発では、標準化、部品化はハードウェアと比較し遅れ気味であるが、昨今の情報システム開発の手法や標準化は独自に発展している。データ中心設計手法やRAD (Rapid Application Development) 開発手法の導入、プログラムの部品化により情報システム開発のコスト削減、納期短縮、品質向上に効果을上げる企

表4. 標準化に伴いもたらす利益

項番	対象部門	標準化に伴いもたらす利益
1	生産	部品, 材料の標準化により生産の集約ができ, 加工能率向上, 段取費の低減, トラブルやロスの減少, 手持ち在庫による経費, 面積の減少ができる。
2	生産	部品構造の標準化により共通段取りが使える, したがって工具費の減少, 加工能率の向上, トラブルの減少が出来る。
3	生産	図面ミスなどによるトラブルが減る。
4	生産	図面の標準化により, 図面が見やすくなり, 誤解がへる。
5	生産	はめあい等の工作程度や方法の標準化によってあわせ加工や調整などの手間が減少する。
6	生産	工場自動化, 量産化に対する基礎となる。
7	生産	工場側から設計に対するフィードバックが的確に出来る。
8	生産	公差, 複雑な加工等, 設計の工場に対する必要以上の要求を防止することが出来る。
9	設計	製図方法の標準化により, 手数が少なくなり誤りが減少する。
10	設計	部品の標準化により, 設計製図の手間が省かれる。
11	設計	設計方法や計算方法の標準化によって, 設計製図の手間が省かれ, 誤りが防止される。
12	設計	はめあい等の工作程度や方法の標準化により, 設計製図の手間が省かれ誤りが防止される。
13	設計	設計手法, 図面様式, 部品などの標準化によって, 設計の自動化, 機械化が可能になる。
14	設計	図面様式などの標準化によって設計管理が容易になる。
15	設計	設計基準, 選択規準の規定によって, 設計に先立って必要な, 基本的な情報を容易に得られることが出来, かつ自然に全社的な統一が出来る。
16	設計	固有技術 (ノウハウ) の体系的な蓄積利用が可能になる。

業も現れてきた(澤口ら2001, 中村ら2000)。さらに、21世紀となりソフトウェアの部品化が容易な、オブジェクト指向言語のJavaの登場により、コンポーネント(部品)を組み合わせる、組立型の開発形態であるJavaフレームワークの構築により成果を上げている(城後2004, 中村1999)。しかし、コンポーネント型開発やJavaフレームワークにより成果が報告されているのは、主にオブジェクト指向言語(青山2003)を前提にしたプログラミング工程が中心であり、設計工程を対象にした事例は少ない。設計工程とは、生産物であるドキュメント(これ以降、設計工程のドキュメント全般を設計書と呼ぶ)を作成することが目的である。この目的を達成するために行う作業は、コニカミノルタ情報システム(以降HJSと略)では、ツールにワープロツールを利用し、作業は、ワープロツールを利用して図及び文章を作成することが設計作業である。具体的な例として画面の設計書を作成する作業では、図作成は画面のレイアウトを作成し、文章は、画面の表示項目のデータ表示方法や、画面へのデータ入力時にデータをデータベースに格納する方法などを文章にして設計書に記載する作業である。この設計書の作業で作成される、図及び文章は、設計工程の前工程であるシステム方式設計工程で作成される設計書をさらに詳細化する設計書である。このため、設計工程では、システム方式設計工程の設計書を見ながら作成している。このシステム方式設計工程で作成される設計書には、設計工程でも全く同一の図及び文章が含まれている。例えば、画面の設計であれば画面の項目名が代表的であり、画面の項目名は、同一文章がシステム方式設計工程の設計書に含まれ、これを設計工程では、設計工程の設計書に転記している。HJSではシステム方式設計工程から設計工程でも利用する文章について、工程をまたぎ継承される情報であるととらえ、工程間継承情報(大宮ら2006, N. Oomiya et al. 2006)と呼んでいる。しかし、標準化の進んでいない設計書作成作業では、設計書のフォー

マットの作成や工程間継承情報の継承方法が、上述の通り単純な転記となってしまう、設計書間の整合性維持や仕様変更時の修正に工数が掛かり、工程品質の低下が保守後も問題となっていた。そこで、この解決策として表3に示した、標準化の対象業務の項番1, 4及び表4に示した項番9, 13の標準化の利益に着目し、下記する2つの標準化を試験的に適用することを決定した。1つ目は設計書の標準化。2つ目は、部品による設計の自動化、機械化である。設計書作成作業方式にこれを適用し、上述した工程間継承情報を設計工程に部品(これ以降、設計工程の部品であるため設計部品と呼ぶ)として利用する。上述した2つの標準化を設計書作成作業方式に適用し、上述した問題の解決策とした。この解決策の具体的な実現方法として、上述した2つの標準化を組み込んだ新たな、開発支援システムを開発し、これを利用した新たな設計書作成作業方式を採用することによって、生産性と設計品質の向上を実現した。

本論文では、情報システム開発における設計工程に標準化を導入した新しい設計書作成作業方式による品質、生産性向上の事例を解説する。

## 2. 従来の情報システムとその問題点

情報システムの開発工程の定義は各企業によって様々であり不明確である。このため、HJSにおける情報システム開発工程定義を図1に示す。

HJSにおける情報システム開発方式はウォーターフォールモデル(日本ユニシス2000, 玉井2004, 経営情報研究会2000)をベースとした一般的な開発方式を採用している。この方式は各工程に前後関係がある。例えば図1に示すシステム方式設計工程はシステム方式工程の前に要件定義工程を実施する。これを、該当工程の前に実施される工程であり、前工程と呼んでいる。また、システム方式設計工程の次に実施する設計工程はシステム方式設計工程の後に実施される工程であり、後工程と呼んでいる。このようにウォーター



図1. 開発工程

フォールモデルでは、前工程から後工程へ順次工程を進め情報システムを完成させていく方式である。この方式により情報システムを開発する設計工程では、次の順序で設計書を作成している。

- 1) 標準フォーマットをベースに、設計書の記載内容検討を行う。
- 2) 前工程で作成された設計書を元にコンピュータ上で要件仕様を実現するための情報システムの設計を行う。
- 3) 2) で行った設計の結果を、図及び文章として、ワープロツールに転記する。
- 4) ワープロツールに転記した設計の結果を、ワープロファイル形式で記憶メディアに保存する

## 2.1 設計工程の問題点

HJSにおける設計書作成作業の特徴として、まず設計工程の設計書フォーマット記載内容の検討を行う。ここで確定したフォーマット及び記載内容を担当者の利用頻度の高いワープロツールにて上述の通り、図及び文章をワープロファイルとして設計書としてファイルサーバへ格納する作業である。この従来作業の設計書作成作業では次に記すような問題点が発生している。

- 1) プロジェクトごとに設計書のフォーマットを再検討するため、毎回検討に時間が掛かる。
- 2) 設計書のフォーマットがプロジェクトごとに異なり、必須情報が欠落し設計書の品質が低下する。
- 3) 設計書は各設計書単位に別々に作成され、他設計書でも再利用できる工程間継承情報を再利用できず、すべて手作業で設計書を作成している。
- 4) 設計書は格納されるファイルサーバのディレクトリが様々であり、設計書を閲覧、印字する

際に、設計書をファイルサーバから探す時間が掛かる。

- 5) 設計書の閲覧権限ルールは、手作業依存によって守られるものであり、強制力がなく、機密漏洩の可能性がある。

特にワープロツールで行う設計工程の設計書作成作業では、前工程であるシステム方式設計工程で作成される設計書に、後工程でも利用する工程間継承情報を数多く使用する。しかし、従来の設計書作成方式である手作業を主体とする労働集約型の設計書作成方式では、手動継承となってしまう、設計書の整合性維持や仕様変更時の修正に工数が掛かり、情報システム開発における設計工程の品質低下をまねき保守・運用後も課題となっている。具体例として、システム方式設計工程で作成される設計書であるテーブルレイアウトの工程間継承情報は、この工程間継承情報を利用して作成される設計工程の設計書である画面項目仕様書を調査（22本のJavaプログラムで1本平均800ステップ）した結果、約130箇所同一工程間継承情報が利用されていた。この結果から元設計書となるテーブルレイアウトに変更があった場合、画面項目仕様書では約130箇所の修正が必要であることが判明している。

## 3. 新しい開発方式

### 3.1 アプローチ

2章で分析した設計工程における設計書の作成、修正に工数が掛かる課題を解決するアプローチについて検討した。その結果、課題は大きく2点に分類されることが判った。1点目は標準設計ルールの課題、2点目は設計書作成ツールの課題である。ルールの課題については、現状の開発方

法において、ルールが強制できないことが大きな課題である。上述の課題で述べたように、ルールは存在するが手作業依存により維持されるルールであり、厳密にルールが遵守されない場合がある。この課題について以前からHJSでは、問題視されており維持、監視する仕組みを構築し様々な対応を行ったが、抜本的な解決に至っていないのが現状である。設計書作成ツールの課題については、ワープロツールで、作成される設計書は別々に作成され、工程間継承情報が個々の設計書のみに有効になってしまう点が課題である。この結果、工程間継承情報の維持が手作業依存となり、工程間継承情報の維持作業の改善が必要であった。これを解決するアプローチとして上述した、表3、4に示す標準化の対象業務及び標準化に伴いもたらす利益から解決策の検討を行った結果、上述した課題は表3に示す標準化の対象業務の、項番1、4の繰り返し頻度及び同じような手続き等を多くの人々が繰り返すに解決策が合致すると考えた。さらに、表4に示す、標準化に伴いもたらす利益についても、項番4、13にて用いられる用語である図面を情報システム開発の設計書と置き換えることによって、同様の利益が得られるのではないかと判断した。

標準化の適用方法として、1つ目は設計書作成作業方式をワープロから後述する開発支援システムに変更する方法。2つ目は、部品による設計の自動化、機械化である。部品化については、上述した設計部品を利用する。具体的には、設計書作成ツールをワープロから脱却し、ワープロツールに代わる設計書作成ツールに変更する設計書作成方式を新たに開発した。すなわち、ワープロツールに代わる設計書作成ツールにより標準化が自動的に適用される。また、設計書内で記述する文章を設計部品として独立させ、複数の設計書で共通に利用できるツールを社内イントラネット上に新設し、クライアント/サーバ型とリレーショナルデータベースシステムにより開発標準（設計ルール）を盛り込んだ“開発支援システム”として開

発し、これを利用し設計工程の設計書を作成する作業方式である。

### 3.2 開発支援システムを用いた開発方式における追加機能

開発支援システムを利用し情報システムの設計書作成を行うに当たり様々な開発方式を検討したが社内に定着している従来通りのウォーターフォールモデルを採用し、従来採用していた設計書作成ツールのワープロの代わりとして開発支援システムを利用する方式とした。開発支援システムの新機能を以下に列挙した。各標準ルールが自動的に適用される機能として5つの機能を追加している。

#### 1) 標準フォーマット転写機能

本機能は、開発支援システムに入力された情報をリレーショナルデータベースから取り出し設計情報を標準フォーマットに転写する機能である。

#### 2) 必須情報欠落防止機能

本機能では各設計書に必須情報をあらかじめ定めることにより、開発支援システムに情報入力（プロジェクト名称や作成者などの情報）時には、その必須情報を入力しないと入力完了できない機能のことである。

#### 3) 該当設計書抽出機能

本機能では開発支援システムに入力された情報を設計書の作成又は、修正時に、設計書の格納先となる、ファイルサーバのディレクトリを意識することなく、設計書を抽出できる機能である。

#### 4) 閲覧制限機能

本機能では開発支援システムに利用者をあらかじめIDにて登録することにより、そのIDによって閲覧権限を特定する機能のことである。また、この閲覧制限機能には開発支援システムの利用範囲を特定する権限機能も備えている。

#### 5) 設計書リレーション機能

本機能は、開発支援システムへの入力結果をリ

レーショナルデータベースに格納することによって、その格納された入力結果に工程間継承情報を付加し設計部品とする機能のことである。この機能によって、工程間継承情報を設計部品とし複数の設計書で共通に利用できる。

### 3.3 開発支援システムにおける範囲特定

図1に示した情報システム開発の各工程の設計書から、開発支援システムにて設計書作成作業をサポート対象とする設計書の特定を試験的に行った。要件定義工程における設計書は記載内容もプロジェクトによって様々な図及び文章が記載されており、再利用可能な設計部品化は困難であることが導かれた。しかしシステム方式設計工程以降の工程では、プロジェクトが異なっても同様な図及び文章が多く見受けられ、設計部品化も容易であると判断した。この結果全体で約70種類あった設計書から、設計部品化が容易と判断した18種類の設計書（全体の約26%）を開発支援システムにてサポートする範囲とした。また、選定した設計書は、実際に18種類以上の設計書が存在したが、記載内容が多少異なるが、内容を分析すると集約できる設計書が多く、これを集約して表5に示す18種類の設計書とした。なお、設計書内には図及び文章が混在記載されている設計書も存在したが、試験的な取り組みとして文章のみに限定した。

### 3.4 18設計書のデータ分析

次に設計部品の再利用を容易にするために、設計部品ごとの再利用可能な範囲の特定を行った。設計部品には再利用可能な範囲があり、この範囲を特定することによって再利用時に必要な設計部品のみを利用可能とした。この範囲を特定することによって開発支援システムにおいて、設計書を作成する際に必要な設計部品のみ選択可能となり設計書作成工数を軽減できる。設計部品の再利用できる範囲とは、プロジェクトや工程を指す。設計部品によって、プロジェクト共通で再利用でき

る設計部品や、各工程に特化して再利用可能な設計部品が存在している。例えば、システム方式設計工程や設計工程の設計書のヘッダに記載される“プロジェクト名称”は全工程で再利用される。しかし、他プロジェクトでは再利用されない。また、工程の手順に沿った順番で開発を行うウォーターフォールモデルで情報システム開発を行っているので、システム方式設計工程で生成された設計部品は後工程である設計工程で再利用される。逆に後工程である、設計工程の設計部品がシステム方式設計工程で再利用するような工程の手順が逆転することはない。このように該当プロジェクトや工程に必要な設計部品は本来決まっており、該当プロジェクト以外の設計部品は不要である。このため、該当プロジェクトや工程で再利用できる設計部品を特定することによって、必要な設計部品の利用を容易にする。具体的には、後述する開発支援システムで設計書を作成する方式は、設計部品をドロップダウンや、リストボックスを用いて、選択し設計書を作成する方式を採用しており、このドロップダウンやリストボックスに表示する設計部品を必要最低限にすることによって、設計部品の選択が容易になるからである。設計部品の再利用可能な範囲を特定するに当たり、18設計書をさらに分析した。分析方法は、設計部品が設計書に記載される工程の順序や、同一工程内における設計書の作成順序（以降、タイミングと呼ぶ）の早さによって、順位付けを行う方法で再利用可能な範囲の特定を行った。この順位付けを行う理由として、設計部品が記載されるタイミングが早い設計部品は、他の設計部品がなくても生成できる設計部品であり、後工程の設計書で再利用される可能性が高く、逆にタイミングが1番遅い設計部品は、自工程における再利用に限られる。言い換えれば、タイミングの早い設計部品は、広範囲の設計書、工程に利用され、タイミングの遅い設計部品は、狭い範囲の設計書、工程で利用されることが特定できるからである。順位付けを行うに当たって、設計部品を個々に行うのではなく、設

表5. 対象設計書一覧

項番	設計書名称	記載概要	記載イメージ
1	プロジェクト一覧	担当開発範囲内のプロジェクト一覧	プロジェクト開始日, 終了日, プロジェクトオーナー等
2	プログラム一覧	該当プロジェクト内プログラム一覧	プログラムID, 名称, 担当者等
3	ペンディング一覧	該当プロジェクト内ペンディング事項一覧	未決事項, 起案者, 起案日等
4	Q&A一覧	該当プロジェクト内Q&A内容一覧	質問, 回答, 質問者, 回答日等
5	不具合表	該当プロジェクト内不具合内容一覧	現象, 調査, 対応等
6	仕様変更表	該当プロジェクト内仕様変更内容一覧	変更内容, 変更方法, 変更日等
7	テーブル一覧	該当データベース内テーブル一覧	テーブル論理名称, 物理名称等
8	テーブルレイアウト	該当データベース内テーブルレイアウト	論理項目一覧, 項目物理一覧, 主キー情報等
9	データ定義	該当データベース内データ定義	データ論理名称, 物理名称, 物理レンジ等
10	データベース変更ログ	該当データベース内データベース変更ログ	データベース変更内容, 変更依頼者等
11	機能概要仕様書	該当プロジェクト内特定プログラムの機能概要	該当プログラム主要機能等
12	UI部品仕様書	該当プロジェクト内特定プログラムの画面, 帳票で利用する共通部品一覧	該当プログラム画面表示項目種別(ドロップダウン, テキスト等)
13	標準項目仕様書	該当プロジェクト内特定プログラムの画面, 帳票で利用する項目の基本編集仕様	該当プログラム画面表示項目種別(ドロップダウン, テキスト等), ソート順等
14	コンボボックス仕様書	該当プロジェクト内特定プログラムの画面で利用されるドロップダウンリストの表示仕様	該当プログラム画面表示ドロップダウン表示データソート順, 表示データ取得方法等
15	項目編集仕様書	該当プロジェクト内特定プログラムにて項目を編集する仕様	該当プログラム画面, 帳票イベント別表示データ取得方法等
16	データチェック仕様書	該当プロジェクト内特定プログラムにて項目へ実施するチェック仕様	該当プログラム画面イベント別項目チェック方法等
17	DB更新仕様書	該当プロジェクト内特定プログラムにてテーブルへ行う操作仕様	該当プログラムイベント別データ項目セット方法等
18	仕様書変更履歴一覧	該当プロジェクト内特定プログラム各仕様書にて発生した変更履歴一覧	該当プログラム仕様書変更履歴, 変更日等

計部品を類別し(以降, 部品種別と呼ぶ), この部品種別ごとに順位付けを行うこととした。分析結果では, 部品種別は5種類存在することが判った。分析結果を表6に示す。次にこの部品種別への順位付けを行った。順位付けの結果として, 表7の結果が得られた。この表7の順位が高い順に設計部品が再利用可能な範囲(以降, 部品適用範囲と呼ぶ)を図2の通りに定義した。図2では四角で囲っている範囲が部品適用範囲である。また, 四角内の楕円はどの部品種別であるかを表してい

る。具体的には1番大きな四角である開発基準情報は様々なプロジェクト情報から再利用されるが, 逆に1番小さな四角内にある設計書情報は他のプロジェクト情報から再利用されることはない。ただし, 表7の分析結果では, データベース設計情報は, プロジェクト情報以降に発生するのが一般的であるが, データベース設計情報が, 全社的なデータベース設計情報のような場合, 新たなプロジェクトであっても, 再利用する必要がある。表7の通りの部品適用範囲では定義しなかつ

表6. 部品種別

部品種別				
開発基準情報	プロジェクト情報	データベース設計情報	設計書情報	プログラムID情報
成果物に共通して記載されている作成者や工程名称などは異なるプロジェクトにも共通で記載されていることが判明した。これら、プロジェクトに依存しない共通情報を“開発基準情報”と位置づけた。	成果物には、プロジェクト固有の情報があることが判った。これは開発工程には依存しないがプロジェクト全般に利用される情報である。例として、Q&Aや不具合情報などは開発工程全般で利用される。これらプロジェクト共通情報を“プロジェクト情報”と位置づけた。	分析段階では、設計書情報の類であり、設計書の分類を想定していた。しかし、分析を進めるとデータベース関連設計書情報はデータベース設計以降工程成果物に頻繁に記載されていることがわかった。この分析結果から設計書情報とは別な部品種別とした。	プログラミング工程の上流である、要件定義工程などの工程では、成果物記載内容もまちまちでありデータ化するのには困難であることがわかった。しかし、逆にプログラミングに近い工程の設計工程では記載内容が類似しており、データ化することも容易であることが判明した。この分析結果から独立した部品種別とした。	プログラムIDはプログラミング工程以降の成果物では、ほとんどの成果物に記載されていることが判明した。また、この情報はプログラミング工程以降では、各種成果物を特定するメインのキーになっていることが多いことがわかった。この分析結果から独立した部品種別とした。

表7. 部品種別順位付け結果

部品種別	代表例	データ発生タイミング	順位
開発基準情報	個人名称	開発工程定義とは無関係であり、HJS入社または協会会社からの派遣時で他情報と依存関係無し	1
プロジェクト情報	プロジェクト名称	開発工程定義の企画段階（プロジェクト発足時）	2
データベース設計情報	アトリビュート定義	開発工程定義のシステム方式設計段階（プロジェクト内仕様確定時）※但し別プロジェクトに流用されることもある	3
設計書情報	項目転送仕様	開発工程定義の設計段階（システム方式設計仕様確定時）	4
プログラムID情報	プログラムID	開発工程定義の設計段階（システム方式設計仕様確定時）	4

た。このような場合プロジェクト情報より先にデータベース設計情報は存在するので設計部品種別の発生順位は逆転することとなる。この関係から表7の発生順位とは異なったデータベース設計情報を2位、プロジェクト情報は3位とした設計部品適用範囲の定義とした。

入力ではなくドロップダウンやリストボックスから選択する方式とすることにより、情報継承を可能とした。開発支援システムの画面イメージを図3に示す。

### 3.5 開発支援画面

前述の部品種別及び部品適用範囲によって類別した設計部品を用いた、開発支援用システムを開発した。開発支援システムでは、設計部品は新規

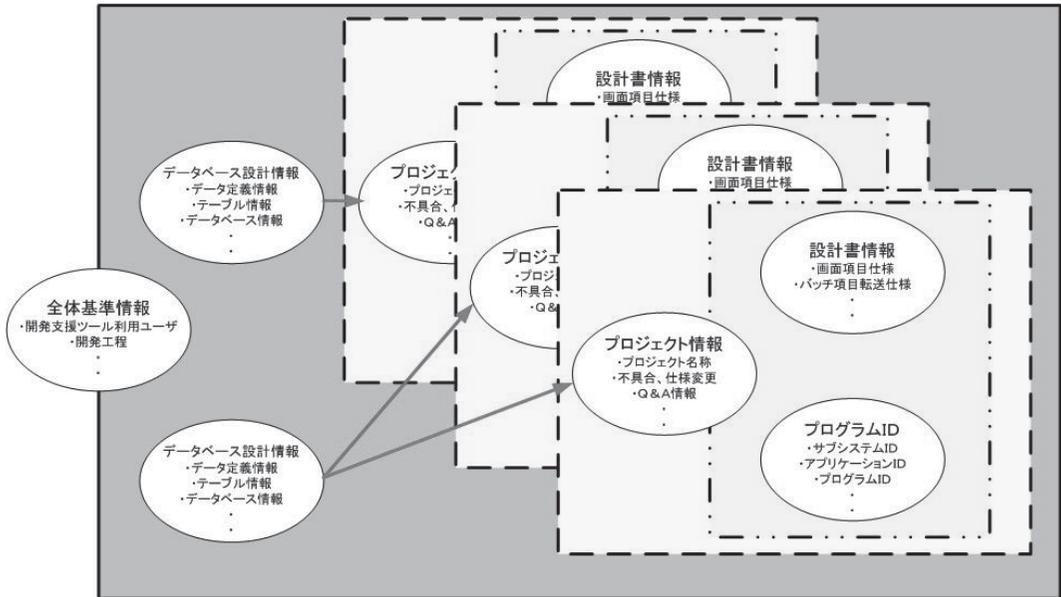


図2. 部品適用範囲



図3. 開発支援画面

#### 4. 評価

開発支援システムを開発し試用した結果、効果を得ることが出来た。しかし、評価を行うに当たり、HJSにおける評価データの収集が十分に確立されていないため評価指針を下記する方針で行う。

1) 評価は、開発支援システムの使用、未使用によって、収集されているデータの評価項目に差があったかの評価を行う。

なお、効果確認は試験的に1プロジェクトに適用し計測を行ったため、1プロジェクトに対しての効果確認となっている。

表8. 成果物作成における工数削減

成果物作成における工数削減, 品質向上	
開発支援システム—未使用	開発支援システム—使用
工数=約18日	工数=0日
各担当が表5の18設計書フォーマットをそれぞれ1から作成していた。1設計書フォーマット作成には平均1日程度の工数が掛かっていた。また個人によりフォーマットもばらばらで品質が安定しなかった	開発支援システム利用者は開発支援システムに入力することにより、標準フォーマットが自動的に適用され設計書フォーマット考慮時間が削減された。また、必須情報が欠落しなくなり成果物の品質が向上した

表9. 成果物修正における工数削減

成果物修正における工数削減, 品質向上	
開発支援システム—未使用	開発支援システム—使用
工数=約2日	工数=約0.2日
表5に示す対象設計書一覧の「データ定義」に変更があった場合影響する設計書は18設計書中6設計書であり、かつこのデータ定義を利用するプログラムが最小見積で2プログラムとして約2日必要となる	開発支援システム利用者は「データ定義」を修正することによって格納された情報に継承情報が付加されるので他設計書修正は不要となり工数削減。また、修正内容が自動継承されるため陳腐化も抑制でき品質が向上した

#### 4.1 設計工程における開発支援システムの評価

本項では、上記で報告した設計工程に適用した、開発支援システムを実際の開発現場で試用し、品質、生産性面で有効であるか否かを比較する。比較は、1プロジェクトにて開発支援システムを使用し開発支援システムで提供する18設計書を作成した工数と、開発支援システム未使用時の工数の比較を行った。評価結果として、設計書作成時には、標準ルール自動適用によって表8の通り成果物フォーマットの考慮時間が短縮された。これは開発支援システムを利用し成果物を作成することによって、標準ルールが自動的に適用される標準フォーマット転写機能の効果である。その他の機能によっても効果が得られている。必須情報欠落防止機能によって必須情報の欠落も防止され品質が向上した。該当成果物抽出機能では成果物を検索する工数が短縮されている。最後に閲覧制限機能では機密管理が行われ機密漏洩防止に貢献できた。次に成果物修正時には、設計書内で記述する文章を設計部品として独立し、複数の

設計書で共通利用することによって表9の通り成果物修正工数削減の効果が得られた。これは、開発支援システムを利用し該当設計部品を修正することによって、格納された設計部品に継承情報が付加され関連成果物に修正結果が自動的に反映される、設計書リレーション機能の効果である。

## 5. おわりに

情報システム開発を行うに当たり標準化の一貫として、従来採用していた設計書作成ツールのワープロの代わりとして、開発支援システムを利用する作業方式を開発し、評価結果の通り有効であることを立証できた。今回の取り組みの中で、今後の課題として次の3点が挙げられる。

- 1) 今回範囲外とした開発工程の情報継承方法
- 2) ネットワーク環境接続不能時対応
- 3) 図標準の取り込み
  - 1) については今後範囲外工程の工程間継承情報の分析をさらに進める必要がある、工程間継

承情報分析結果を待って対応することが必要である。2) については、イントラネットに接続できないような外部委託開発で、工程間継承情報をどのように上述の仕組みに取り込むか開発支援システムを含めて検討が必要である。3) については、一部の図を文章に変換し格納する試みを行ったがアウトプット時の表現力不足があり、思った効果が得られなかった。この結果を受けて、表現力のある図をアウトプットするか、図と同等レベルの文章表現を模索するか検討が必要である。

### 参考文献

- 青山幹雄, 中谷多哉子 (2003) 「ソフトウェアは開発の必須技術オブジェクト指向に強くなる」, 株式会社技術評論社
- 大宮望, 大場允晶, 山本久志 (2006) 「工程間継承情報を利用した開発支援システムの構築—情報システム開発の場合—」日本設備管理学会誌
- 大宮望, 大場允晶, 山本久志 (2006) 「情報システム開発における工程間継承情報を利用した開発支援システムの構築」(社)日本設備管理学会, 平成18年度春季大会予稿集
- 大宮望, 大場允晶, 山本久志 (2006) 「情報システム開発における設計工程に自動化を導入した開発支援システムの構築」(社)日本経営工学会, 平成18年度春季大会予稿集
- 緒方健二 (1990) 「おはなし社内標準化」, 日本規格協会
- 経営情報研究会 (2000) 「図解でわかるソフトウェア開発のすべて」, 株式会社日本実業出版社
- 澤口優, 加藤真樹 (2001) 「Webアプリケーション開発におけるJavaフレームワークの適用—はじめてのJavaへのアプローチ」, 第39回IBMユーザー・シンポジウム論文
- 城後匠 (2004) 「企業におけるコンポーネント型開発の実践と課題」, 第42回IBMユーザー・シンポジウム論文
- 菅野文友 (1992) 「経営情報のためのソフトウェア製品生産工学入門」, 株式会社日科技連出版社
- 玉井哲夫 (2004) 「ソフトウェア工学の基礎」, 株式会社岩波書店
- 中村伸裕, 谷本取 (2000) 「オブジェクト指向技術によるJavaフレームワーク構築」第38回IBMユーザー・シンポジウム論文
- 中村伸裕, 谷本取 (2003) 「コンポーネント再利用によるシステム開発の工業化」, 第41回IBMユーザー・シンポジウム論文
- 日本ユニシス株式会社情報技術研究会 (1999) 「システム開発の体系」, 東京電気大学出版局
- 野口祐 (1990) 「ソフトウェアの経営学」, 有限会社森山書店
- N. Oomiya, M. Ohba, H. Yamamoto (2006) 「Construction of Development Support System Based on Succession Information between Processes — The Case of Information System Development —」, 11th International Symposium on Logistics, Beijing, China 2006