2013 年度

アシスト ソリューション研究会

分科会 活動報告書

システム開発コストの継続的な削減方法を考える 分科会
（中日本）

目次

1　はじめに

景気の長期低迷や激化する競争のなかで、私たちシステム部門においてもコスト削減が強く求められている。
「標準化」によるコスト削減は古くから取り組まれているが、変化の激しいこの時代においてどのような課題があるのか、今一度見直してみた。

2 分科会の概要

2.1 メンバー紹介

本分科会のメンバーは表 2.1-1 の通りである。
一般企業 8 名、アシスト 2 名の計 10 名のメンバーで一年間の研究活動を行った。

表 2.1-1：分科会メンバー

氏名	会社名	役割
石川　敬也	日本アイ・ビー・エム・サービス株式会社	リーダー
高木　健児	ＮＤＳソリューション株式会社	サブリーダー
加藤　純一	河村電器産業株式会社	サブリーダー
森本　英治	トランコムＩＴＳ株式会社	
堀川　沙貴	株式会社中電シーティーアイ	
渡辺　祐真	日東工業株式会社	
坂口　享	百五コンピュータソフト株式会社	
河村　泰宏	豊田合成株式会社	
高木　哲	株式会社アシスト	
西尾　由美	株式会社アシスト	

（敬称略、順不同）

2.2 活動実績

本分科会は表 2.2-1 のとおり、研究活動を行った。

表 2.2-1：活動実績

回	日程	場所	内容
1	1/17	アシスト	発足会
2	2/5	アシスト	
3	3/15	アシスト	
4	4/17	アシスト	幹事会
5	4/19	アシスト	
6	5/10	百五コンピュータソフト	
7	5/23	ノリタケカンパニーリミテド	
8	6/21	トランコムＩＴＳ	
9	7/10	アシスト	
10	7/19	アシスト	幹事会
11	8/20	ＮＤＳソリューション	
12	9/19	ノリタケカンパニーリミテド	
13	10/10	アシスト	幹事会
14	10/23	中電シーティーアイ	
15	11/5	ノリタケカンパニーリミテド	
16	11/12	ノリタケカンパニーリミテド	
17	11/14	アシスト	リハーサル
18	11/20	トランコムＩＴＳ	
19	12/9	アシスト	
20	12/12	メルパルクＮＡＧＯＹＡ	発表会

2.3 テーマ選定の理由

システム開発におけるコスト削減として、「ムダをなくし生産性を向上させる事」や「手戻りをなくし品質を向上させる事」が挙げられる。そして、「生産性向上によるコスト削減」と「品質向上によるコスト削減」の両方を満たすアプローチには「標準化」がある。
この論文中では、各用語を以下のように定義する。

「生産性向上」　……時間・難易度・労力の観点から、少ないインプットで大きなアウトプットを生むこと。
「品 質 向 上」 ……品質のバラつきをなくし、一定の品質を確保すること。
「標　準　化」 ……誰が実施した場合でも一定レベルの結果を得るための基準やルールをあらかじめ定めておく手法。各担当者が一から考慮する必要がないため、品質が均一になることに加え早く開発ができる。
「開 発 標 準」 ……標準化によって定められた基準やルール。

システム開発においては、ドキュメント作成基準、コーディング基準、ネーミングルール、各種マニュアルなど実に多くの開発標準が存在する。しかし、標準化によって決められたルールがシステム開発の現場で本当にコスト削減の効果を生み出しているのだろうか。
標準化を利用して開発を行う現場からは「開発標準に従うことで、無駄なテストを実施してコスト高になっている」、「開発標準とされている開発フレームワークを使っているが、無駄なコーディングが多いのではないか」、「ルール通りにドキュメントを揃えることが目的になっているのではないか」と言ったような声が聞こえてくる。これは、開発標準を作成した意図と、利用する現場の実感にズレが発生しているということではないだろうか。

標準化は古くからコスト削減の定石として取り組まれてきた方法で、それ自体を否定するつもりはない。しかし、変化の激しいこの時代において、単に開発標準を作るだけでは解決しない課題があるのではないかと考え、標準化でコスト削減をするために「開発標準を作ること以外に何が必要か」を研究することにした。

2.4 「コスト削減」についてのメンバーの考え

分科会メンバーそれぞれが考えるコスト削減のための施策はどのようなものか意見をまとめた。

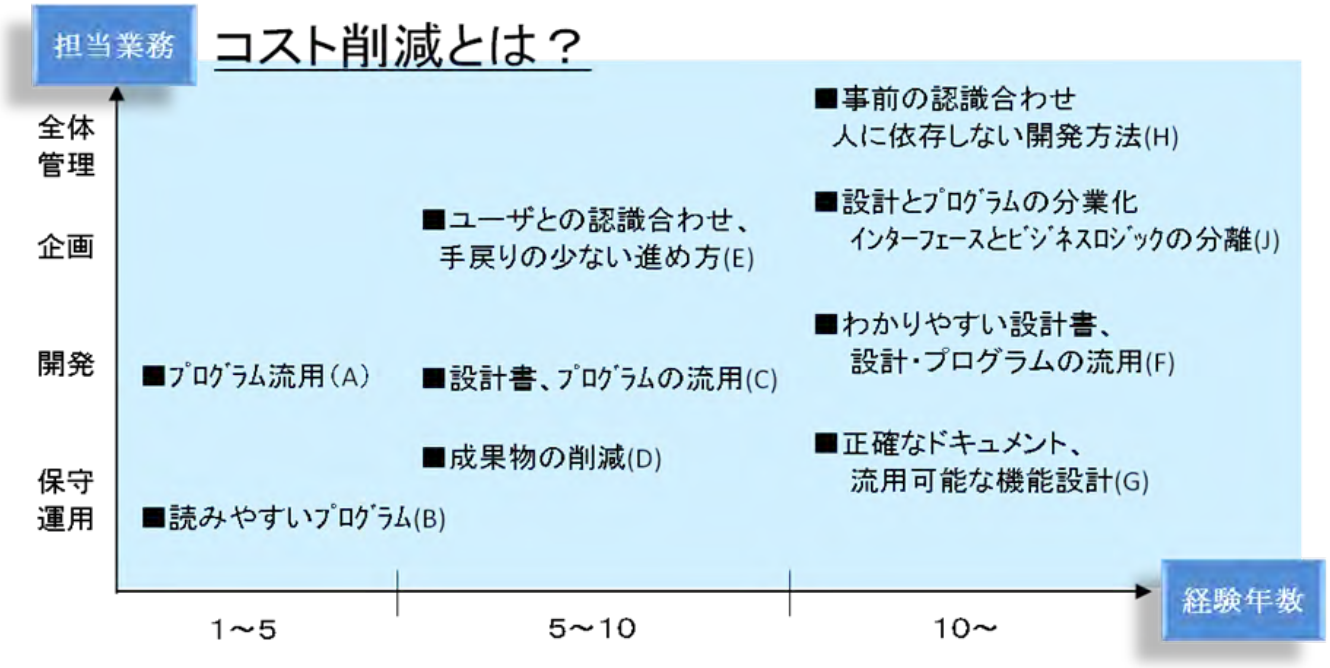

図 2.4-1：メンバーの考え

担当業務や経験年数、立場によって、コスト削減に対する視点が違うため多様な意見が集まった。

図 2.4-1 は、縦軸に業務内容、横軸に経験年数を定義しており、縦横交わるところが各自の意見である。例えば、E さんは経験年数 5ー10 年であり、企画業務を担当している。コスト削減に対する考え方は、「ユーザとの認識を合わせ、手戻りの少ない進め方をする」ことである。

全体的な傾向としては、経験年数によって対象となる範囲が広がっていることがわかる。

このように「コスト削減」の捉え方は担当業務や経験年数により様々であり、多面的にコスト削減の研究ができるのではないかと期待感が高まった。

3　開発工程と課題

3.1　開発工程と課題の洗い出し

議論の視線を合わせるために、システム開発の工程ごとに問題を整理し、標準化すべき課題を洗い出すことにした。システム開発工程は、さまざまな団体や研究者によって既に整理されているが、我々は以下の５つの工程を採用した。また、標準化を行なうことによる期待する効果を、「均一」、「速さ」、「省力」と定義した。（図 3.1-1 参照）

図 3.1-1：開発工程と期待する効果

次に、各工程を標準化の課題設定ができる粒度にまで掘り下げ、問題・課題・対応策・期待する効果を整理した。

「問題」 ……現在コストがかかりすぎていると考えていること。解決策があると思っているが、実現していないこと。
「課題」 ……問題を解決するために何をすべきか
「対応策」 ……課題に対する具体的な活動案
「期待する効果」 ……対応策を実施することによって、得られる効果（問題の軽減、除去）

次ページ図 3.1-2 が洗い出した各工程の課題をまとめた一覧の一部である。

開発工程	大分類	問題	課題	対応策	期待する効果
システム企画	SLA定義	求められるサービスのレベルが不明確だと必要なリソースを洗い出すことができない。	求められるサービスのレベルを定義して必要なリソースを洗い出す。	ドキュメントに記載しSLAの定義を明確化する	求められるサービスを明確化し、必要なリソースを洗い出すことで、ユーザーと同意した運用が可能となる。
		開発側・利用者側で、サービスのレベルが合意されないまま、システム構築が行われて、サービス提供に関するユーザーの不満に対応出来ないことがある。 システムの可用性と、停止した場合の業務への影響を分析されず曖昧な為、後々問題となる。 サービスレベルの基準がないと非機能要件が決められない為、ハードウェア構成を決められない。 ユーザーレスポンスの具体的な数値が決まっていない為、後々問題となる。	システム概要	ビジネスに対する影響度を考慮する	開発、運用に必要なコストやスケジュール算出根拠が明確になる ・サービス提供に関するユーザーの不満に対応出来る。 ・必要なコストとサービスのレベルを合意できる。
				対象とするシステムの業務範囲を選定する	
				利用するユーザの人数や対象を選定する	
			可用性	サービスを停止しなければいけない時間を決める	
				年間の許容されるシステムダウンの時間を決める	
				障害を検知した時間から、障害が回復してユーザがサービスを受けられるまでの時間	
				システムが故障し、回復してからまた故障するまでの時間	
				システムが故障し、回復するまでの許容できる上限の時間	
				障害が発生してから、ユーザに障害が発生したことを通知するまでの時間	
			パフォーマンス	処理要求を送ってから出力が始まるまでの時間	
				システムに処理要求を送ってから、結果の出力が終了するまでの時間	
				単位時間当たりの処理数	
			サポート	ヘルプデスクの問い合わせ先	
				平均呼び出し回数	
				問題解決時間	
			保全性	特定のユーザのディスク容量	
				ディスク使用率が一定値を超えないこと	
				バックアップを取得する頻度	
				バックアップしたメディアを何世代保管するか	
				システム上のデータベース、ログの保持期間	
				ウィルスチェックの有無や頻度、アクセス制御の実施、監査証跡の取得とチェック、侵入検知、セキュリティパッチの適用と頻度など	
要件定義	開発方針	前提条件・制約条件をクリアできない開発環境を選択すると、要件の実現が出来な	前提条件・制約条件をクリアするための開発	予算を遵守するために、イニシャルコスト（開発、ライセンス、インフラ）、ランニングコスト（保守）を意識して言語・データベースを選択する	全ての条件をクリアする環境を選択することで、要件漏れや手戻り
				開発言語を選択する場合、調達できる開発要員の条件を考慮する。	
				コストを抑える為にクラウドサービス利用(Saas、Paas、Iaas)の可能性を探る。	
				開発生産性、性能、拡張性などを考慮してクエリ、ストアド、PL/SQL等、特定のRDBMS製品に依存する機能を使用するか方針を明確にする	

図 3.1-2：課題一覧（一部）

3.2 課題の分析

具体的に分析した結果の一例を、画面設計を題材に紹介する。

3.2.1 問題

画面毎に仕様がさまざまで、画面レイアウトやプログラムコードの流用ができない。

3.2.2 課題

画面のユーザインターフェースをデータベースに対する処理機能単位（登録、更新、削除、照会）にパターン化することで開発者単位もしくはプロジェクト単位にユーザインターフェースを一から考える必要がなくなり、画面設計工数が削減できる。

3.2.3 対応策

画面ユーザインターフェースのパターンの洗い出しを行う（単票形式、親子形式、一覧形式など)。また、データベースの設計モデルパターンの洗い出しを行う（トランザクションが単一テーブル、複数テーブルなど)。
それらをあわせて標準プログラムパターンとする。画面レイアウト（１画面構成、ヘッダ・ディテールなどのフレーム構成など)､データベースアクセスロジック(イベントアクション時でのデータベースアクセスが単一テーブル、複数テーブルなど）をサンプルコードとして、複数の標準パターンを準備する。
作業者は要件に合わせて、いくつかの標準サンプルの中から必要なサンプルコードを利用し、要件固有部分のみ設計する。

3.2.4 期待する効果

設計者、プログラム開発者は標準サンプルで準備された部分を独自に開発する必要がなくなり、開発の速さと省力化が見込まれる。
標準サンプルを採用することで、ばらつきがなくなり品質の均一化が見込まれる。

3.3　課題分析結果

コスト削減を開発工程の課題として分析したことは結果的に標準化策定の追体験をすることになった。我々が分析した課題や対応策は各企業において既に整備されている標準化そのものであった。これは標準化がコスト削減に効果を上げると我々が考えていたことを証明したに過ぎず、既存の標準化への取り組みの重要性を改めて認識させられることになった。

では、冒頭で述べた標準化へのやらされ感や不信感とは何が原因なのかという疑問が残った。

そこで、時間軸で標準化の課題を捉えることにした。前述の課題分析では開発標準がないところに標準化を導入することがゴールであった。しかし、多くの企業では既に何らかの開発標準が整備されていて、標準化を導入すること自体が課題ではない。開発標準制定から時間が経過するにつれ、開発標準のみを守ることに注力し、本来の目的であるコスト削減のために開発標準自身の改善を行なっていないのではないだろうか。

そのため結果的に開発標準を使う人間が実態に合わない、誰もが感じる問題に対して誰も手をつけようとしない事態に陥っているのではないだろうか。（図 3.3-1 参照）

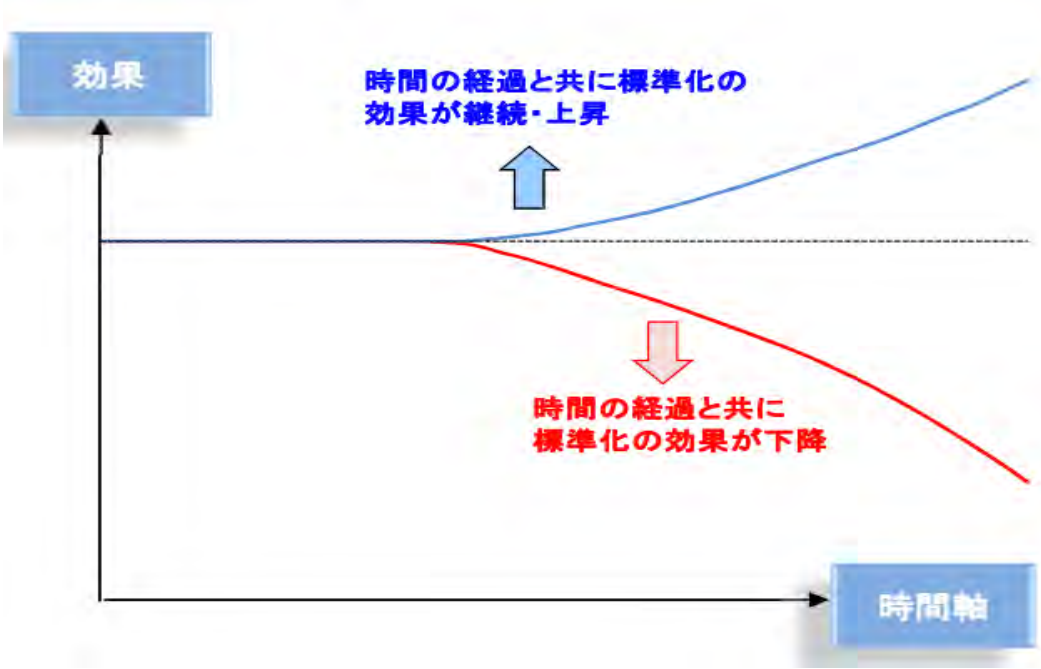

図 3.3-1：効果の時間経過による変化

3.4　標準化の継続的な改善

開発標準が見直されていない場合は、以下の２点の問題が起きていると考えられる。
１点目は、改善意識が失われていることである。開発標準に従っていれば良いという心理が生まれ、より良く改善していく意欲が失われてしまっている。本来の目的はコスト削減であったにもかかわらず、その手段である標準化を目的としてしまっていることである。
２点目は、変化に対して追随する意識が低くなっていることである。プロジェクトメンバーが入れ替わることによって標準化への理解が薄まることや、新しい技術が普及し開発標準が対応できなくなることで、いつの間にか開発標準が陳腐化してしまっている。そのように陳腐化した開発標準に従うことはかえって現場の開発生産性を低下させてしまう恐れがある。

開発標準を制定することだけではなく､継続的に標準化がコスト削減の効果を上げるように改善をし続けることが、標準化に対する新たな課題である。継続的に改善するためには標準化をＰ（計画）Ｄ（実施）Ｃ（評価）Ａ（改善）のサイクルにのせ、常にコスト削減効果を最大にできるようにする必要がある。

標準化のＰＤＣＡとして実施するべきことを次ページにまとめた。（図 3.4-1）

3.4.1 Plan（計画）

- 開発標準を管理する体制を作る。
- 標準化を計画する。
- 開発標準を制定する。
- 開発標準に対する評価指標を設定する。

3.4.2 Do（実施）

- 設計者・開発者に対して標準化の教育を行う。

3.4.3 Check（評価）

- 評価指標の結果を収集する。
- 設計者・開発者より問題点の意見収集を行う。
- 期待する効果に達しているか評価する。

3.4.4 Act（改善）

- 評価結果に対し、改善案を策定する。
- 新技術に対する情報を収集する。
- 今の開発標準に足りない追加項目をまとめ、改善案を作成する。

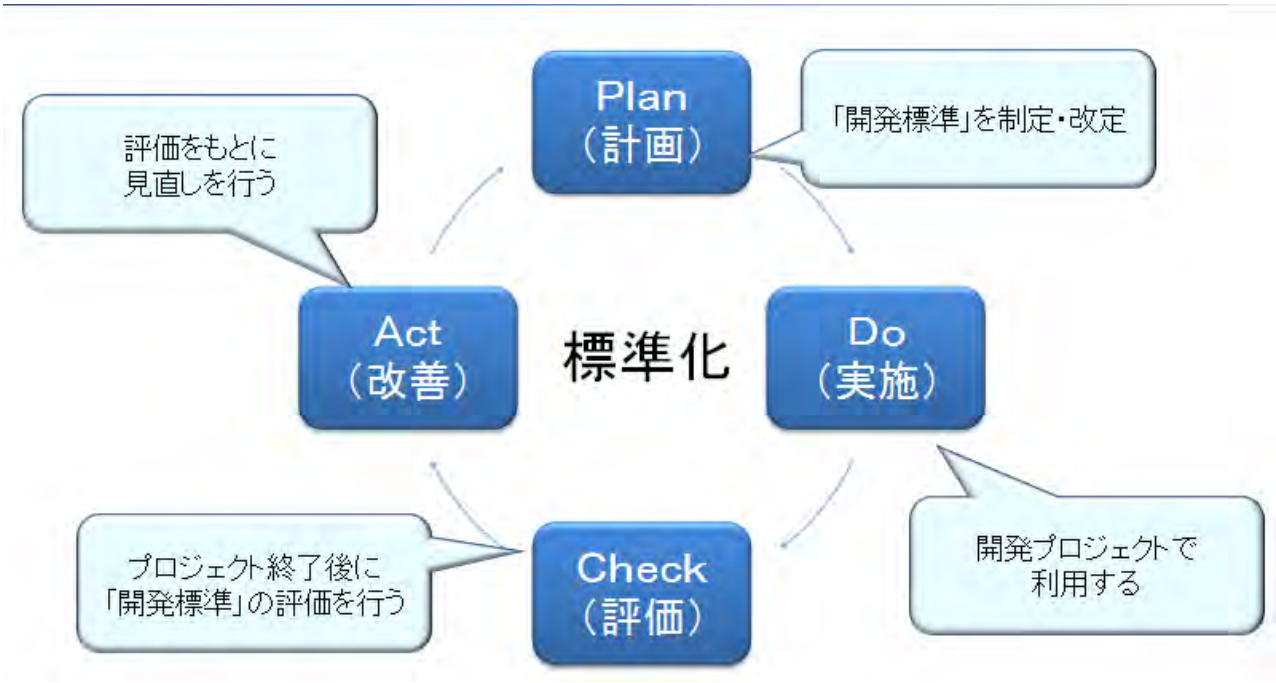

図 3.4-1：PDCA サイクル

4　標準化の適用範囲

4.1　「標準化」に関わる3つの立場

前章では一つの時点ではなく、時間軸で捉えることによって見えた課題を認識できた。次に範囲について着目した。分科会メンバーで議論をする上で、それぞれの立場や経験で捉えているため標準化を想定している範囲にばらつきがあることに気づいた。

そこで、標準化をどの範囲に適用するか以下の３つの軸で考えた。また、３つの立場を設定して、標準化の適用について考具体例で考えてみた。（表 4.1-1）

３つの軸

・　どの立場で考えるか

　　実務年数の違いやプロジェクトでの役割の違いを立場として捉える。

・　どの範囲で考えるか

　　システム開発やプロジェクト管理で標準化の適用範囲を範囲の軸として捉える。

・　どんな点で考えるのか

　　標準化によってどんなコストが削減されるかを着目点として捉える。

３つの役割的立場

・　プログラム開発者

　　自分の担当する作業においてのみ効率化を優先し、狭い範囲で標準化を考える立場。

・　プロジェクト管理者

　　プログラム開発工程だけではなく、上流工程や他人の立場での効率化も考慮して検討し、比較的広い範囲で標準化を考える立場。

・　組織管理者

　　プロジェクト単位だけでなく、管理組織全体を考慮し、効率的に標準化を運用できる。また、より広い範囲で標準化を考える立場。

表 4.1-1：3 つの軸と立場

視座	視野	視点	コスト削減の発想例
プログラム開発者	コーディング量の削減	プログラムの生産性	使用関数の標準化
プロジェクト管理者	システム全体の品質、開発生産性	プロジェクトにおけるシステム開発生産性	プロジェクトルール
組織管理者	組織全体の品質・開発生産性・保守性	生産性・保守性を高める	役割分担

次の章では、３つの立場の違いを「システム開発工程における開発フレームワークの適用」という具体例で説明する。フレームワーク適用の工程を選択した理由は、３つの立場すべてに関係があることと、分科会メンバー全員が経験しているため議論がしやすいと考えたからである。

4.2 プログラム開発者

4.2.1 プログラム開発者とは

プログラム開発者とは､プロジェクトメンバーの中でも業務の範囲が狭く､経験年数の少ない開発者のことである。この立場の者は、コーディング量の削減をすることでコスト削減を実現しようとする。プログラムの開発生産性を高めることを目指し、使用する関数の標準化など、コーディングに直結することを考える。

4.2.2 開発フレームワークに期待する効果

プログラム開発者は開発フレームワークを導入することで、自身のコーディング量が減ることを期待する。たとえば以下の２点のようなことである。

- 定型的な部分をフレームワークによって実現し、業務処理のみをコーディングするだけでシステムが構築できるようになること。
- フレームワークを拡張し、同一のコーディングをなくすこと。

4.2.3 プログラム開発者のフレームワーク適用例

プロジェクトにおいて、市場にある汎用フレームワークを適用することで開発の生産性を向上させる。プログラム開発者の立場でのフレームワーク適用は、１プロジェクト内の担当しているプログラム内に止まり、プロジェクトを超えることはない。

4.2.4 プログラム開発者の問題点

プログラム開発者の視点には、２つの問題がある。

１つ目は、自分以外への影響力がないことである。自身の担当する作業の改善にのみ注力し、上流工程との連携や他システムへの再利用などは思慮されない。

２つ目は、当人の能力に依存する部分が多いことである。個人スキルから脱却できないため、繰り返し成功を収めることが難しい。また、他の人が作業を引き継いだ場合には、却って生産性を低下させることも考えられる。

4.3 プロジェクト管理者

4.3.1 プロジェクト管理者とは

プロジェクト管理者とは、プロジェクトリーダやプロジェクトマネージャのことである。この立場の者は、プロジェクト全体の品質を均一にすることや、生産性を向上させることを目指す。要件定義などの上流工程、運用やレビュー方法など、プログラム開発者と比べると格段に広い範囲を意識する。

4.3.2 開発フレームワークに期待する効果

プロジェクト管理者は、全体が均一であることを期待する。そのため、一般的な機能をあらかじめフレームワークで用意することで、コーディング担当者による違いを軽減しようとする。

4.3.3 プロジェクト管理者のフレームワーク適用例

プロジェクト全体で利用することを目的とするため、当該プロジェクトに特化したフレームワークを構築する。これには汎用フレームワークの機能を補完し、標準的な共通関数や、帳票出力機能、共通画面などを用意する。

4.3.4 プロジェクト管理者の問題点

第一の問題点として、複数の標準化ができてしまうという点が挙げられる。プロジェクトごとにフレームワークを作成すると、プロジェクトごとの標準を理解しなくてはならない。これは、プロジェクトを横断して開発者を配置する際に妨げとなる。

また、プロジェクト管理者達がそれぞれに開発生産性を高める良いアイデアや成果物を持っていてもプロジェクトに埋没してしまい、組織全体の資産として切り出せない状態となることも挙げられる。

4.4 組織管理者

4.4.1 組織管理者とは

組織管理者とは情報システムの部長や、SI 企業で複数のプロジェクト管理を横断して管理する統括マネージャのことである。組織全体を統括的に管理する立場である。
開発工程だけでなく、保守工程の生産性も含め、システムのライフサイクル全般を通してのコスト削減を図る。また、単一プロジェクトのみではなく組織が責任を担うすべてのプロジェクトを意識する。

4.4.2 開発フレームワークに期待する効果

組織管理者は組織全体の開発コスト削減を考えるため、複数のプロジェクトにフレームワークを適用することで標準化を実現することを期待する。フレームワークによりよいシステム開発にするためのノウハウを集約することで、管理者が違うプロジェクトのコスト削減を図る。

4.4.3 組織管理者のフレームワーク適用例

自社フレームワークを作成し、複数プロジェクトに適用する。この自社フレームワークには組織全体のノウハウを集約する。フレームワークにはシステム開発工程の改善が組み込まれていることや、簡単に利用できることが求められる。

4.4.4 組織管理者の問題点

組織管理者の視点で理想的な開発フレームワークを適用するためには、高いスキルと費用が必要となる。例えば、フレームワークを利用するプログラム開発者はフレームワークの内容を知らなくても使えるようにするため、サンプルコーディングや利用者視点に立ったドキュメント整備など、教育を実施する必要がある。また、実際に開発プロジェクトで利用し、挙がった改善要求を取り込むために、拡張できることを前提にフレームワークを作り運用する必要がある。プロジェクトに特化した機能か、再利用可能な機能なのかを見極めてフレームワークを拡張するかしないかの判断をする必要もある。

4.5 「標準化」の適用範囲を見極める

以上の３つの立場で考えた結果、広い範囲に適用する開発標準は、より大きなコスト削減効果が期待できる。ただし、短期的にコストがかかり、汎用的に作るには難易度が高い。

むやみに範囲を広げ、過度な効果を期待するのではなく、適用範囲を明確にして、コストと効果の適切なバランスがとれた標準化を構築することが重要である。そのためには、この３つの立場のうちどれか一つに特化するのではなく、３つそれぞれの立場で考え、また、考える立場を変えることにより新たな問題が見えてくる。一人一人が自分の立場だけでなく、お互いの立場で物事を考えることが大事であることがわかった。

5　まとめ

5.1.1「標準化」で継続的なコスト削減をするために

今までの研究を通して標準化がコスト削減に寄与していない理由として、以下の 2 観点を挙げた。

・時間の経過による開発標準の形骸化、陳腐化

・開発標準の適用範囲のアンマッチ

上記 2 点の解決策として、一旦作成した開発標準が陳腐化していないか確認し、最適な適用範囲をチェックするための定期的な見直しが必要と考える。

5.1.2 PDCAサイクルを回す

標準化がコスト削減に寄与する状態を維持・向上させるには、開発標準を作るだけで終わらせず、開発標準自身を継続的に改善していく必要がある。

例えば、プロジェクト終了時、新技術への対応時、大規模改修時などポイントとなるタイミングで PDCA を回し常に効果を確認する事が重要である。

5.1.3「標準化」の適用範囲と効果を見極める

標準化を適用する範囲によってコスト削減の効果は異なる。

しかし、私たちが使用する標準化には適用範囲が定義されないまま策定されているケースが多いと思われる。

まずは、標準化の策定・改定を行うにあたり適用範囲を明確に定義する必要がある。その上で適用する範囲の違いよる効果を正しく見極める事が重要である。

ただし、5.1.2 及び 5.1.3 を実施する上では、以下を考慮する必要がある。

5.1.4 効果と投資・コストのバランス

標準化の適用範囲により、コスト削減の効果は異なるが、同時に標準化の導入・維持には投資やコストも必要となる。

適用範囲が広くなるほど、標準化の策定時に考慮するポイントが多くなったり、教育する人が増える為、投資も膨らむ。作られた標準化を維持するためのコストについても同様である。

その為、無作為に標準化を行うのではなく「ベンダー企業であれば開発コストに寄与する標準化を考える」、「ユーザ企業であれば保守コストに寄与する標準化を考える」というように自社の業務の特性に合った標準化を考える必要がある。その上で効果とコストのバランスを見極め有益な標準化を選ぶ事が重要である。

6　さいごに

6.1　分科会を通しての気づき

この研究を通じてメンバーから次のような意見が出た。
「それまでルールとしか捉えていなかったが、標準化を見直したいと思うようになった」
これまでは標準化に縛られるだけでただ従ってきたが、本当に効果があるのか見直してみたいという意見である。

「自身の目線をひとつ上に上げることで、それまで考えられなかった問題点が見えてくる」
これまでプロジェクト管理者の視点で考えていたメンバーは、組織管理者の視点で考えることによって、組織全体としてのコスト削減にはどんな方法があるのかということに考えを巡らすようになったという意見である。

6.2　基本に立ち返ることが大切である

標準化を見直す事や、範囲を考える事は、一定以上の見識がある者にとっては、基本的な事かもしれない。しかし、多くのシステム開発現場においては、その基本的なことがおろそかになっているのではないだろうか。永遠の課題と言える「コスト削減」、もう一度基本に立ち返ることの大切さを学んだ。
『プロジェクトには終りがあるが、システム開発のコスト削減には終りがない』