

## 情報システム開発現場での要員育成に伴う実践と品質確保\*

2Q-04

鶴澤 亨 宮崎 肇之 清水 理恵子 只野 完二 小室 彦三 †

日立製作所 システムソリューション事業部 生産技術統括本部 §

### 1 はじめに

近年における Web システム開発案件の増大に対し、弊社では J2EE(Java™2 Platform, Enterprise Edition)をベースとしたアプリケーション開発スタイルを設定し、社内展開を進めている。しかし開発現場ではオブジェクト指向設計や Web システムのアーキテクチャに対するスキルを持った要員確保の不安定さから、OJT(On Job Training)によって実践と育成を並行して即戦力を得ようとする場合が多く、結果的に品質確保が難しくなる現象も生じている。

本稿ではオブジェクト指向設計の特性を生かし、OJT による実践を通しての設計スキル習得と品質確保の両立を狙った要員育成向けの生産技術とその試行事例について紹介する。

### 2 業務プログラム開発における量産向け生産技術

オブジェクト指向設計による業務プログラムの量産向け生産技術としては、次に示す2つの方法、もしくはその組み合わせが良く用いられている。

#### (1) コンポーネント

粒度の小さなデータやロジック、プロトコルなどをカプセル化する。これらは幾つかのメッセージやパラメータを与えることで容易に操作できる。業務プログラムはこれらを操作する上位プログラムを生産する。

#### (2) フレームワーク

粒度が大きく汎用なアーキテクチャをカプセル化する。業務プログラムはこのクラスを継承し、その特化機能の差分プログラムを生産する。

大規模な業務プログラム開発の現場では予め(1)と(2)の層を共通技術チームが出来るだけ厚く提

供することにより、業務プログラムの開発量を圧縮を図るのが一般的である。このことは本来、設計の自由度を制限するデメリットともなる反面、設計の画一化を図ることで、業務プログラムの能力を業務設計に集中させ、高い生産性で高品質な成果物を得るメリットも大きい。

### 3 量産向け生産技術を要員育成に適用する弊害

しかし経験の乏しい新任者を量産向け生産技術を用いたOJTで育成した場合、次の様な弊害が起こる。

#### (1) オブジェクト指向設計の不十分

ロジックの共通化について十分なオブジェクト指向設計を施せない。特に未経験者の場合、他の習熟要素による生産性向上の立ち上がりが大きく、オブジェクト指向設計の不十分による生産性の伸び悩みは見えにくい、後に成果物の保守性が著しく悪いなどの問題が多発する。

#### (2) アーキテクチャの理解不足に関わる設計不良

Web システムの場合、例えばセッション継続/切断に関するアーキテクチャを十分理解できずに保持すべき情報を失う、もしくはむやみに保持してしまうことでリソースを過剰消費するなどの設計不良を作り込み易い。

量産向け生産技術は、前提としてプリミティブなプログラム開発をある程度経験した作業者がそのメリットを理解した上で適切に適用することで効果を発揮するが、十分な要員育成期間を取ることができない場合、理解不足からこのような不適切な適用をされる危険性がある。

### 4 要員育成向け生産技術

上記の問題を踏まえて、設計スキルの向上に適し、かつOJTに適用し実践と育成を並行した場合も設計不良を生じにくい生産技術について提案する。

#### (1) コンポーネント

コンポーネントは幾つかのパラメータやメッ

\* Practice and Quality-Maintaining with Worker Training in Workshop of Information System Development.

† Toru Uzawa, Tadashi Miyazaki, Rieko Shimizu,

Kanji Tadano, Hikoza Komuro

§ Hitachi, Ltd. Solution Systems

Engineering Support Management Division

ページを与えることによって操作するが、そのデフォルト設定は持つべきではない。つまり常に明示的な操作を行った場合のみ、それに従った振る舞いをする。

このことにより要求仕様の実現に必要な最低限のパラメータと操作手順を理解する必要が生じる。また明示的なパラメータやメッセージが不足していた場合、静的なコンパイルエラーなどで不良の早期発見が容易になる。

## (2) フレームワーク

Web システムのアーキテクチャを理解するために、プリミティブなイベントモデルを崩さない設計とすべきである。例えば実際の HTTP プロトコルのやりとりを実感できる差分コーディングが可能な設計とする。

クラス `javax.servlet.http.HttpServlet` 自体がこの要件をかなり満たしている。

## 5 試行事例

この要員育成向け生産技術を用いて、未熟練者 5 名を Web システム開発に投入した際の試行事例について紹介する。

このシステムでは CORBA オブジェクトとして実装された既存の業務プログラムを利用する画面層プログラムを開発した。画面層は Servlet + JSP + JavaBean による MVC モデルを適用している。

要員育成向け生産技術として、以下を提供した。

### (1) コンポーネント

CORBA オブジェクトのアクセスに必要な手順を隠蔽したラッパークラスなど。

### (2) フレームワーク

`javax.servlet.http.HttpServlet` を継承し最小限のシステム共通機能を加えたもの。

### (3) サンプルプログラム

上記を用いた最小限のサンプルプログラム。

この 5 名は予め Java™ 言語、Servlet、JSP について計 9 日間の基礎教育を受けているが、その理解度はまちまちである。ここでは基礎教育での理解度を 3 段階評価し、良い方から [A]=1 名、[B]=2 名、[C]=2 名と分類する。

## 6 試行結果

今回、最も基礎レベルの高い [A] がサンプルプログラムを元に最初に業務プログラムを完成させた。遅れて同じ手順を踏まえて [B] が業務プログラムを完成させた。また [C] は [B] から流用した業務プログラムを担当業務用にカスタマイズして完成させた。

品質について、[A] と [B] はほぼ同様に良いものとなったが、[C] では明らかにカスタマイズ漏れの不良が何件か発生した。

## 7 試行評価

平均的スキルを持った [B] が [A] と同等の品質を実現できた点は、今回目指した要員育成と品質保持の両立という目的に適用結果として評価できる。

また [B] → [C] の業務プログラムの流用にあたり、一般には流用元が完成されたプログラムであり流用率が高い場合、高品質が維持できると考えられるが、今回カスタマイズ漏れによる品質低下が起きたということは、理解不足のうちにプログラムを完成させるという、量産向け生産技術にあった弊害がここでも起きたと考えるべきだろう。

[A] は施行期間中にフレームワークを用いないプリミティブな Web プログラムを開発するレベルに達したが、これは個人の能力に依存した可能性が高く、要員育成向け生産技術でイベントモデルを崩さなかった方針がより高いスキル習得への橋渡しになったとは断言できない。

## 8 今後の課題

要員育成向け生産技術については更に多くの適用事例を収集し、定量的評価を行う必要がある。

また要員育成向け生産技術が大規模な業務プログラム開発にとって有益なことを確認するには、プリミティブな Web プログラム開発をある程度経験したに相当するスキル向上が得られ、量産向け生産技術を支障なく活用できることについて評価を行う必要がある。

## 参考文献

[1] 宮崎肇之 他: アプリケーション開発スタイルの標準化, 情報処理学会第 63 回大会, 2001.