

フレームワークを利用した Web アプリケーション開発の研究 ーこれであなたも今日からフレームワーカーだ！ー

アブストラクト

1. 研究にあたって

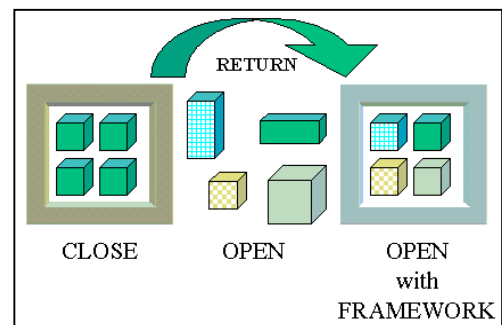
今日の技術革新のスピードと経営環境の変化のスピードは、驚くほど速い。そしてそのスピードは当然システム開発の現場にも多くの影響を及ぼしている。特に IT の支柱である Web アプリケーションの開発に対し、『短期間・低コスト・高品質』という厳しい要求が強く突きつけられている。

そして、この要求を満たすための一つの方策として、『フレームワーク』というキーワードが存在する。本分科会においては、この『フレームワーク』というものが果たして本当に短期開発や品質の向上に役立つものなのかという単純な疑問から出発し、その本質に迫るべく研究を行った。

2. 今なぜフレームワークなのか ～枠組みの帰還～

『フレームワーク』を少し広い視点から捕えたとき、「フレームワークとはシステムを構築する上でのひとつの枠組みである」と言うことができる。要するに枠を当てはめることによって、システムをパターン化してしまうものという理解である。

しかし、その概念自体は新しいものではない。かつて、汎用機でのオンラインシステム開発においては、設計にしる実装にしるある意味、枠にはめられていた。それは『フレームワーク』という言葉でこそ呼ばれてはいなかったが、まさに枠組みに他ならなかった。ところが、汎用機というクローズな世界からサーバを中心としたオープンな世界への流れの中でその枠組みは消失してしまった。ユーザの選択肢は確かに広がったものの、あまりに自由度が高くなりすぎて收拾がつかなくなってしまったのである。それゆえ、消失した枠組みに変わる新たな枠組みが求められた。それが、『フレームワーク』である。



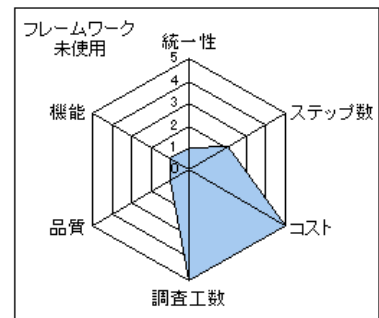
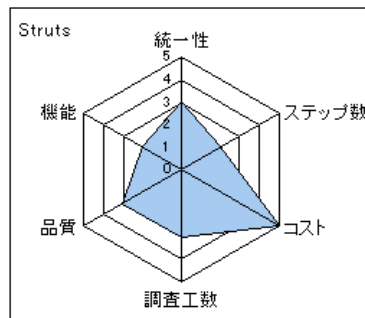
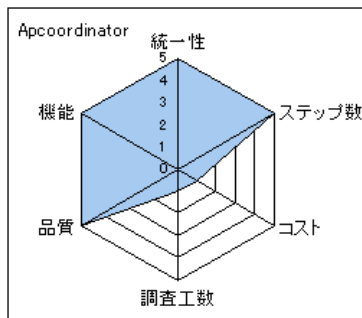
3. フレームワークの有効性 ～フレームワークの真実～

「フレームワークは有効なのか？」本分科会ではその検証を、フレームワークユーザである S 社へのヒアリングと実際の Web アプリケーション開発によって行った。

S 社では、『短期間・低コスト・高品質』なシステム開発を行なうためには、アプリケーションデザインの統一が必要であると考えており、そのためにフレームワークを導入していた。つまり、まず『標準化』という考え方があり、それを実現する手段として、フレームワークを利用していたのである。

では、実際の Web アプリケーションの開発において、フレームワークは本当に、『短期間・低コスト・高品質』を実現するのか？、また『標準化』との関わりはどうか？、本分科会では、その検証のために、簡単なアプリケーションの開発を行い、フレームワーク未使用、Struts 使用、富士通製 Apcoordinator 使用という、3つのケースについて比較検討したのである。

その結果、フレームワークの効果が最も顕著に現れたのは、統一性の実現においてであった。フレームワーク未使用の場合、開発者により作り方はバラバラであった。しかし Struts 使用の場合、機能に含まれていない部分での相違は見られたものの全体的には概ね似た作りになった。そして Apcoordinator 使用の場合は、開発者によらずほとんどが同一の作りになっていたのである。また、開発経験があまりないメンバが作成したにもかかわらず、個別の品質の差がほとんどなかったのだ。フレームワークを標準化の手段としている S 社の正しさが裏付けられたと言えよう。



・短期開発 (生産性)

ステップ数は、コードの自動生成機能がある Apcoordinator が一番少なくすんだが、機能の少ない Struts では、フレームワーク未使用とほとんど変わらなかった。逆に開発工数中の調査時間割合については、Apcoordinator が一番多くなっている。フレームワークの機能を学習するための時間がどうしても必要になるためである。つまり、機能が豊富であればステップ数は削減できるが、その分学習時間は増える。従って、フレームワークの学習時間を早期に短縮することが、生産性向上のポイントとなる。

・低コスト

経験の少ない開発者でも、均一なコードが作成可能であり、あえてスキルの高い (=コストの高い) 人員を用意する必要がなくなるので、フレームワークに多少のコストをかけても、全体的なコストは抑えられる。

・高品質

統一されることで、開発者ごとのプログラム品質のばらつきを抑えられることが期待される。また、Apcoordinator のようにコードの自動生成機能があれば、「手書き」でのコーディング量を減らせ、その分品質を上げることができる。

・保守性

統一性があるということは、誰が作成したものでも、同じようなつくりになっているということであり、保守の容易さにつながる。

4. 提言 ～フレームワーカーへの道～

フレームワークは、導入すればすぐ使えて役に立つという魔法の道具ではない。本分科会の研究の中で明らかになったのは、『フレームワーク』は『標準化』と切っても切り離せないものということである。すなわち、フレームワークを導入するということは、Web アプリケーション開発における一つの標準を適用するということに他ならない。汎用機唯一の時代からオープンへという流れの中で、冒頭の厳しい要求を満たすための方策として考えられているのは、原点回帰とも言える『標準化』なのである。

ただし、標準化は、その標準が浸透するまではその効果が表れてこない。いくつもの開発をその標準で行なうことで、品質は向上し、工数の削減につながってゆくというのは、いまさら説明する必要もない事実である。

フレームワーク導入の効果をあせってはいけないのだ。一番やってはならないのは、フレームワークをツールと同一視し、複数のシステムで異なるフレームワークを別々に導入してしまうことである。これは、複数の標準化が適用されている状態と同じなのだ。標準化は、ひとつに定められてこそその効果を発揮する。フレームワークの導入もしかりだ。我々は、フレームワークを使いこなし、「短期間・低コスト・高品質」という命題を実践する人々をフレームワーカーと呼びたい。そして最後に以下を提言する。

★メーカーへ	すぐに使える、使いやすいフレームワークの提供を！
★開発者へ	フレームワークを凌駕する技術の蓄積を目指せ！
★経営者へ	フレームワークの効果を焦ってはいけない！