

# Ruby on Rails によるアプリケーション開発

## —Rails は実業務で使えるのか?—

### アブストラクト

#### 1. 研究の背景と課題

Ruby on Rails (以下 RoR) は、オブジェクト指向スクリプト言語 Ruby をベースにしたオープンソースの Web アプリケーションフレームワークであり、DRY (Don't Repeat Yourself/繰り返しを避けよ)、CoC (Convention over Configuration/設定より規約重視) の設計思想や、高い生産性で注目を集めている。本分科会のメンバーは、自社でのシステム構築の際に、RoR の適用により開発のスピードアップやコスト削減が図れるのではないかと期待を持っていたが、同時に次のような課題もあった。

- (1) **実用アプリケーション構築の経験不足**・・・書籍などから基本的な実装例を学ぶことはできるが、実業務で構築するシステムでは多種多様な機能が要求される。まだ RoR の知識や経験に乏しいため、それらの機能が実現できるかどうかの判断がつかない。
- (2) **適用すべき局面の判断材料不足**・・・プログラミング言語やフレームワークの選定において、RoR が選択肢にあがった場合の採用根拠となり得る情報を持ち合わせていない。
- (3) **開発プロジェクトを推進するノウハウの不足**・・・一定規模以上のシステムを構築するには、相応の人数によるチーム開発が必要となる。その場合の開発体制や手順の指針が定まっていない。

本分科会では、これらの課題を解決するための作業を通じて、実業務でのシステム構築にあたって RoR が現実的な選択肢となり得るのかを見極めることとした。

#### 2. 研究のアプローチ

従来、Java や .NET で Web 基幹系業務システムを開発する際には、開発プロセスを定めて、プロジェクト要件を実現するための考慮点や手順を整理するような工夫をするのが一般的である。分科会では、RoR を用いて業務を構築する際も同様な工夫が必要であると考え、標準プロセス体系 (SDEM) をベースとした開発プロセスに則り、RoR による基幹系業務アプリケーション開発を試行することとした。そして、その体験から得られた知見を集約し、開発基盤として RoR を選択する場合の判断基準や、実際に RoR で開発を行う場合の注意事項が盛り込まれたドキュメントを作成することとした。

#### 3. 研究内容

分科会では、以下の手順により研究を実施した。

- (1) **適用指針整理表の策定**・・・基幹系業務システムを構築する際に検討すべき項目や実装すべき機能を評価ポイントとして抽出し、適用指針整理表を策定した。まずは、メンバーの経験に基づき、現時点で保有している一般的な知識を記入した。
- (2) **開発ガイドラインの策定**・・・SDEM 開発プロセスの工程をベースとして、RoR 適用時に特別に考慮が必要と考えた事項を盛り込んだ開発ガイドラインを策定した。
- (3) **実証システムの仕様策定**・・・基幹系業務システムを想定した実証システムの仕様を策定した。実証システム「RUBYLS」は、書籍販売業の基幹業務を想定しており、書籍検索/注文、書籍情報管理、注文管理、在庫管理の機能を有している。その際、適用指針整理表に挙げた評価ポイントの中から重要度の高い機能を盛り込んだ。
- (4) **実証テーマの設定**・・・開発ガイドラインに盛り込まれている内容から、開発プロセス上で特に重要で実証システム構築により検証が必要と考えた項目を実証テーマとして設定した。
- (5) **実証システム開発と開発結果の評価/フィードバック**・・・10 人のチーム編成 (RoR 開発経験者無し、Ruby 経験者 1 名) により、策定した開発ガイドラインに沿って、実証システムを開発した。また、開発から得られた知見を整理し、適用指針整理表と開発ガイドラインへフィードバックした。

#### 4. 研究成果

分科会での研究活動の結果、以下に示すような成果が得られた。

##### 4.1 知見としての成果

- (1) **実装から得られた定性的結果**・・・実証システムの開発を通じて、開発コスト、開発要員（開発体制、開発に必要なスキル）、品質特性（提供される機能集合、セキュリティ機能、ドキュメント類の整備状況、習得のし易さ、規約の実用性、ソースの見易さ、仕様変更への強さ、バージョンアップの状況）、開発ツールについて評価した。
- (2) **機能としての充足**・・・基幹系業務システムを構築する上で重要度が高いと判断した機能を選定し、実証システムに盛り込み、確認した。
- (3) **実証テーマの評価**・・・実証システムの開発を通じて、実証テーマの具体的な作業内容や留意事項について明らかにした。実証テーマは、「Scaffold ジェネレータの利用」、「Ajax 技術の利用」、「プラグインの利用」、「帳票機能の実装」、「本番環境の構築方法」、「MVC 及びプログラム構造」、「トランザクション処理の実装」、「開発中 DB 変更の管理」、「RDoc によるドキュメント生成」、「コーディングレベルのセキュリティ対策」、「チーム開発」、「デバッグ方法」、「テスト機能」の 13 テーマである。

実証システム開発を通じ、RoR の機能充足を確認した一方、考慮すべき点を明確にすることができた。

##### 4.2 ドキュメントとしての成果

- (1) **適用指針整理表**・・・評価ポイントを以下のように整理し、研究活動で得られた知見を示した。
  - 【A表】適用判断の観点・・・開発コスト、開発要員、品質特性、開発手法、開発環境、データ移行、ライセンス
  - 【B表】機能実装の観点・・・認証系、エラー制御、DB 系、バッチ系、UI 系、セッション管理、外部システム連携、言語仕様、セキュリティ対策、ヘルパー機能
- (2) **開発ガイドライン**・・・開発プロセスの各工程での作業時の指針として有用と思われる知見を、文書化した。アプリケーション分野（SDEM のカテゴリ D）での UI～IT の各工程を対象とし、「各工程の作業指針と RoR 導入のポイント」を示している。また、RoR 導入を容易にする補足情報として、「RoR 概要」、「インストール方法」、「コーディング規約」、「RDoc 利用方法」、「本番環境構築方法」も示している。

#### 5. 評価／提言

分科会での研究活動を通じ、RoR は実業務でのシステム構築にあたって十分な機能を備えており、基幹系業務システム開発の現実的な選択肢になり得るとの判断に至った。コーディング量の少なさから品質や保守性向上の効果につながるものが RoR の魅力である。性能面に関しては、本分科会の実証システム上での問題は発生しなかったが、実適用時には評価が必要と認識している。仮に性能的な問題が発生した場合でも、ハードウェア・ソフトウェア双方の観点から適切なチューニングを実施することにより解決することができると判断している。研究当初に持っていた課題に対しては、次のような結論を得た。

- (1) **基幹系業務システムを RoR で構築する妥当性**・・・機能要件を達成する観点では、RoR は Java と比較しても遜色はない。性能や信頼性は不都合ない程度には確保されている。
- (2) **RoR の適用を検討する際の留意事項**・・・RoR の技術は変化が激しく、バージョン間で非互換が発生する可能性がある。また、実装作業をスムーズに進めるためには、Ruby 言語を含め、RoR への幅広い理解が必要である。このようなことから、RoR やプラグインのバージョンを早期に確定し、RoR 開発経験者が開発要員を支援できるような体制を作ることが不可欠である。
- (3) **開発プロジェクト推進時の留意事項**・・・RoR では規約に則って開発することが生産性・品質に直結する。そのため、規約をプロジェクトへ浸透させ、早い工程から設計上の考慮に入れる必要がある。

現在、RoR は黎明期から普及段階に差し掛かっており、既に先進的な企業での採用実績も多くなってきている。今後は多くの企業でのシステム構築に Java と並んで RoR の適用が検討され、RoR による開発プロジェクトを推進していく局面は増えていくと予想される。本分科会では企業内の業務システム開発における RoR 導入を一足先に体験することができた。

今後、RoR 適用の検討、あるいは、RoR を使用した開発を行う企業の皆様に、本分科会の成果物を活用していただき、プロジェクト成功の一助となれば幸いである。