

■ITベンダから見たメリット・デメリット

分類		メリット	デメリット	課題解決の方向性
開発生産性 /運用分野	開発のスピード・生産性	<p>■<u>文法が直感的に分かりやすい</u></p> <p>文法が英語に近く人間のイメージを表現しやすいため、開発生産性が高い。Webアプリケーション開発での素早いリリースや頻繁な変更にも対応しやすい</p> <p>■<u>習得しやすい</u></p> <p>Javaに比べて早く習得できる (PHPやJavaのWebアプリケーションフレームワークは多種多様であるが故に、開発エンジニアが分散化しやすく共通認識を得にくい。RubyはRailsにほぼ集約されているので、開発エンジニアの共通認識を得やすい)</p> <p>システム開発経験豊富なCOBOLエンジニアの再生も可能→利便性に優れた高品質なシステム開発の実現につながる</p> <p>■<u>少ないコードで簡単に素早くアプリケーション開発できる</u></p> <p>あるWEBアプリケーションプログラムについて、同等開発規模レベルでRubyとJavaの比較では、Rubyコード量がJavaの1/3~1/5の記述量であるなど、開発時間も短縮でき生産性が高い</p>	<p>■<u>ライブラリのメンテナンスやバージョン対応など整備が不十分</u></p> <p>ライブラリを使いこなすことが難しい (メンテナンスされていないライブラリも多く、使用している環境で動作しないケースもある) 稼働環境のバージョンアップに伴い、既存システムが稼働できなくなるリスクもある</p> <p>■<u>コーディングなどの情報量が少ない</u></p> <p>Java、Perl、PHPなどと比べてコーディングなどに関する情報量が少ない ソフトウェア資産や開発手引き書が少ない</p>	<p>■<u>Ruby開発基盤(プラットフォーム)構築とライブラリーの拡充</u></p> <p>・Rubyアソシエーション(松江市)を中心にビジネス分野でのRuby活用対応が進められている</p> <p>・生産性に優れ、多機能なライブラリの充実したRuby開発基盤(プラットフォーム)の構築が期待されている (中国経済産業局の補助事業により、ITベンダ連携によるRuby開発基盤を構築中)</p> <p>・合わせて、それを使いこなすための人材育成と作業標準化を進める必要がある</p>
	開発の自由度	<p>■<u>コード記述はプログラマの自由度が高い</u></p> <p>(簡単にコーディングできるコードである)</p> <p>■<u>開発しながら修正や改良がしやすい</u></p> <p>コンパイルを必要としないインタプリタ型言語であるので、プログラム記述して即結果を確認できることから修正や改良等作業が効率的</p> <p>■<u>開発作業が効率的な仕組み</u></p> <p>クラスを継承することによって、クラス定義のみで同じことを繰り返さなくてデータベース操作が可能 「設定よりも規約」の考え方で定義などの作業は1回だけで済む</p>	<p>■<u>共同作業や作業継続性(メンテナンス性)に難点がある</u></p> <p>大規模システム開発などの共同作業や作業継続性(メンテナンス性)に難点がある 作業の標準化と技術者の育成が必要 自由度が高く、簡単にコーディングできるコードだけに、一歩間違えると汚いソースになりかねない</p> <p>■<u>プログラム実行速度が遅い</u></p> <p>インタプリタ型言語であるがゆえ、実行速度は遅くなる (特にアクセスが大規模化した時に実行速度が遅くなる)</p>	<p>■<u>プログラム開発の標準化</u></p> <p>コーディング規約を厳しく設けることでプログラミング作業の標準化を図る手法が求められる</p> <p>■<u>処理速度問題の対応策</u></p> <p>ハードウェアの性能向上により実行速度はある程度解消できる また、RailsとデータベースのMySQLのチューニングによって大量のデータアクセスによる性能低下を解消できる</p>
	クラウドコンピューティング アジャイル開発とRuby	<p>■<u>テスト駆動開発が可能で迅速かつ適応的に開発する(アジャイル開発)に適している</u></p> <p>(プログラム本体よりも先にテストケースを書く手法で、データベースを用いたアプリケーションにおいてもテスト用のデータベースを簡単に生成できる) (開発対象を多数の小さな機能に分割し、反復(イテレーション)開発で機能を作り込む)</p> <p>■<u>Webアプリケーション開発、クラウドコンピューティングでのアプリ開発に向いている</u></p> <p>アジャイル手法で開発し、プロトタイプを早期にリリース可能。顧客のニーズを開発途中で取り込みやすい。(仕様が決まっていない場合の開発や仕様のスピードアップを図っていくのにアジャイル開発が適している)</p> <p>特に向いているのはSaaS型のモデル。パッケージ、プロダクト、サービスというのがアジャイル開発をする可能性のあるところ</p>	<p>■<u>基幹業務系での開発実績が少ない</u></p> <p>ビジネス分野や基幹業務に不可欠な帳票処理などのバッチ処理について、Ruby一貫開発を実現するためには共通基盤化が求められる Rubyと多言語を組み合わせた開発も考えられるが、両方の技術者の準備など効率面で難点がある 基幹業務系での開発は、お客様との請負契約の関係上アジャイル開発は工数等が事前に明確にできないため、「ウォーターフォール手法」にならざるを得ない</p> <p>■<u>アジャイル開発でのユーザーの専門人材が不足</u></p> <p>本来のアジャイル開発は、ユーザーが開発プロジェクトの中に入り、ジャッジメントをしていく必要があるが、ユーザーに知識を必要とするため専門人材不足から困難なケースが多い</p>	<p>■<u>「ウォーターフォール開発」における柔軟な開発手法の適用</u></p> <p>アジャイル開発のよい点をウォーターフォール開発に取り込んだ開発など柔軟な開発手法によって基幹業務系での開発に対応することが一つの解決策として考えられる</p> <p>例えば、当初計画(設計・製造・テスト)に基づき、1回の開発でシステム構築を完了。設計工程からPG要員も参加させ、プロジェクト全体を通じて同一メンバーで開発。設計書・結合/総合テスト手順書・成績書を作成するが、PG設計書は作成しない。チームワークを保ったモチベーションの高い協力体制が構築でき、高品質につながるメリットも出てくる。工程間のドキュメントを最小限に抑え、生産性を向上させることができる</p> <p>■<u>IT専門家(ITコンサルタントやITコーディネータ等)の活用</u></p> <p>ユーザーに開発に参画するための知識や人材が求められるが、ユーザーの代わりにITコンサルタントなどの専門家の活用も解決の一つの手法</p>
	信頼性	<p>■<u>堅牢なシステムが構築可能</u></p> <p>Railsが提供するセキュアな機能の効果と、開発規約を用いてセキュアな実装を徹底することで堅牢なシステムが構築可能</p> <p>■<u>安定稼働、演算精度については問題ない</u></p> <p>基幹業務開発で問題となる演算精度は、固定小数点演算ライブラリがサポートされているので問題ない 安定稼働は、Ruby+Rails+MySQL+Apache+LinuxのWebアプリケーション開発で多くの実績もあり安定性が確保されている</p>	<p>■<u>基幹業務系での開発実績が少ない</u></p> <p>基幹業務系での開発実績が少ないので、安心感が弱い</p>	<p>■<u>Rubyのサポート体制の確立</u></p> <p>Ruby開発に関するサポート体制を置く必要がある サポート体制の確立はユーザーの安心感と普及拡大につながる セキュリティ、開発手引き書の整備等も含めたサポート体制が期待されている</p> <p>■<u>Rubyの標準化</u></p> <p>Rubyの国際標準化(ISO登録)に向けて作業中であり、Rubyの標準仕様書の作成も行われているところ ISO登録されれば、信頼性の高い言語として普及しやすい</p>

■ITユーザーから見たメリット・デメリット

分類	メリット	デメリット	課題解決の方向性
<p>コスト削減／保守性</p>	<p>■調達コスト削減効果が高い</p> <p>従来のエンタープライズ系システムに比べて、OSS活用によるためコスト削減効果が高い ONEインストール毎にかかるコストを削減できる</p> <p>■開発期間の短縮</p> <p>Rubyが向いているアジャイル開発(ユーザーニーズを機敏に取り込む開発手法)を実践することで、必要な機能を必要となる時期に活用することが可能 真に必要なでかつ使える(役に立つ)システムを早期に手に入れることができる</p> <p>■最適なシステムへの要求盛り込みがしやすい</p> <p>開発途上でシステム画面等のイメージが確認でき、操作性等のニーズを要求しやすい</p>	<p>■システム開発要求仕様知識がより求められる</p> <p>アジャイル開発でユーザーも開発に参画する場合、ユーザーも要求仕様を開発者に伝える能力が必要となってくる</p> <p>■新たな言語の管理などが発生する</p> <p>保守しなければならない言語が増えることにより、維持費(人材の確保)が増加する</p> <p>■開発期間短縮効果はアジャイル開発手法に縛られる</p> <p>アジャイル開発であれば、特徴を引き出しながら最適なシステム開発が短時間で可能であるが、一括請負の作業委託では、アジャイル開発の実施が難しい</p>	<p>■開発参画の努力</p> <p>アジャイル開発によるメリットを受けるためには、開発途上で画面回りや操作性などを細かく確認しながら要求をするなど、システム開発に参画することが必要となる</p> <p>■IT専門家(ITコンサルタントやITコーディネータ等)の活用</p> <p>ノウハウを持たないユーザーについては、IT専門家の活用で、OSS採用やシステム最適化などが可能となる また、アジャイル開発の中でITベンダとユーザー側のIT専門家との連携で最適なシステム構築が実現できる</p>
	<p>■保守費用も低減できる</p> <p>言語プログラムの可読性(プログラムの読みやすさ)が高いため、Ruby技術を有する企業ならどのITベンダでもメンテナンスなどの対応が可能であり、保守費用低減につながる。開発から保守・運営まで地域のITベンダによる小回りがきいた対応が可能となり、コスト削減につながる</p>	<p>■Rubyの稼働環境のバージョンによつて稼働しない場合もある</p> <p>Rubyの稼働環境のバージョンアップにともない、既存システムが稼働できなくなるケースもあり、リスクが存在する 開発するシステムは、担当するエンジニアのスキルに依存し、保守フェーズに入った以降にコストが増加する可能性がある</p>	<p><提言></p> <p>■ユーザーニーズを実現できるRuby開発人材の育成と技術者集積などが必要である</p> <p>■保守サポート体制が必要で、最低5年(サーバーが5年でリリース)維持できることが必要である</p>
<p>信頼性</p>	<p>■堅牢なシステムが構築可能</p> <p>Railsが提供するセキュアな機能と、開発規約を用いてセキュアな実装を徹底することで堅牢なシステムが構築可能</p> <p>■安定稼働、演算精度など信頼性は確保されている</p> <p>Web系のアプリケーション開発での安定稼働は、Ruby+Rails+MySQL・Apache・LinuxのWebアプリケーション開発で多くの実績もあり安定性が確保されている</p> <p>一方、基幹業務開発で問題となる演算精度については、固定小数点演算ライブラリがサポートされているので問題ない</p>	<p>■基幹システムでの適用が少ない</p> <p>情報系のシステムに使われているが、事業の根幹となる基幹システムへのRuby適用事例が少ないため不安がある</p> <p>インタプリタ言語であるため、実効性能に不安がある</p>	<p>■保守等サポート体制の整備</p> <p>保守を含めたサポート体制を充実する必要がある</p>