Ruby on Railsの実力を検証
～ エンタープライズ適用に向けて ～

日立ソフトウェアエンジニアリング株式会社
Rubyセンタ
正村 勉

2009年12月10日

HitachiSoft
創る、支える、拓く

会社概要

社　名	日立ソフトウェアエンジニアリング株式会社（略称：日立ソフト）
設　立	1970年 9月21日
資本金	341億円（2009.3月末）
売上高	<連結>1,658億円（2009.3月期） <単独>1,524億円（2009.3月期）
従業員数	<連結>7,151人（2009.3月末） <単独>5,283人（2009.3月末）
事業内容	システム開発、サービス、プロダクト＆パッケージ、 情報処理機器の提供
品質保証	ISO 9001 認定、　CMMI　レベル5達成
	Rubyアソシエーション認定システムインテグレータ

CMMI及びCapability Maturity Model Integrationは、米国カーネギーメロン大学ソフトウェア工学研究所が開発した
ソフトウェア企業の生産性や品質向上を目的として、開発のプロセスを見直し、改善するための世界的な標準の指標です。

目次

1. エンタープライズ適用への課題
2. RoR(Ruby on Rails)適用の検証項目
3. 実力の検証
4. 今後の取り組み

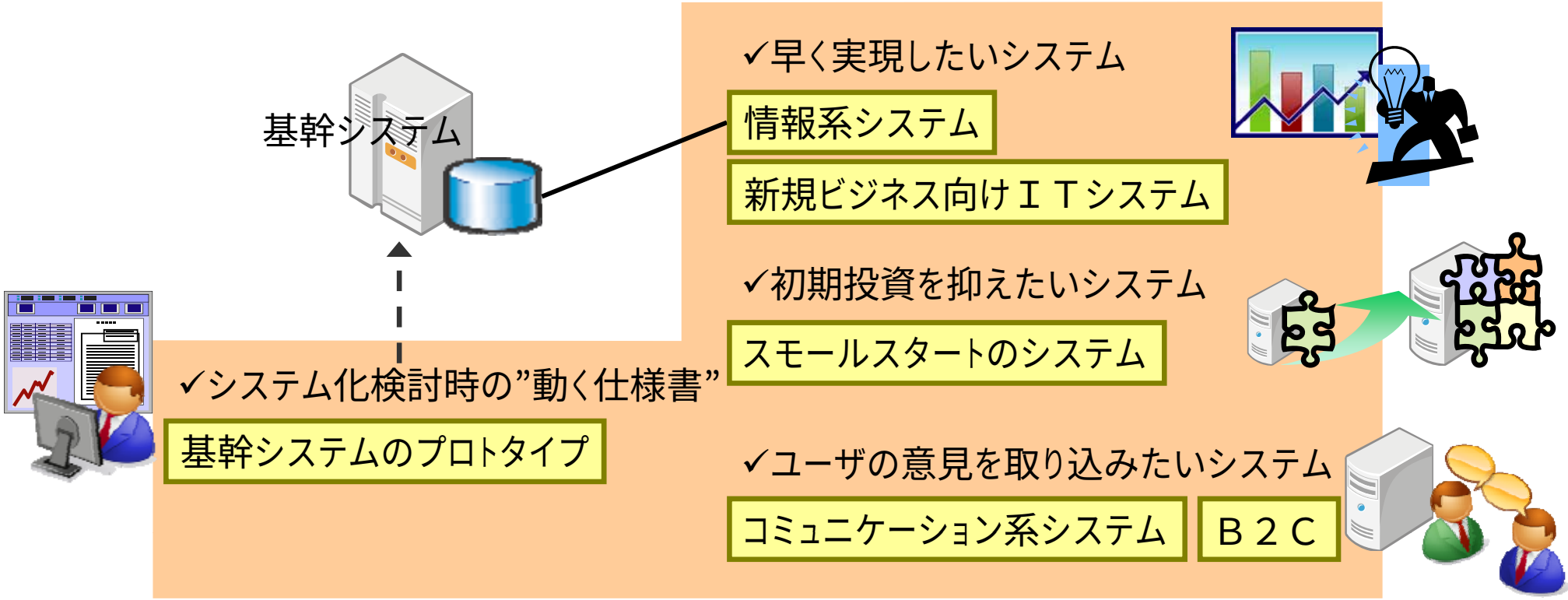
１．エンタープライズ適用への課題

1.1 RoR(Ruby on Rails)の適用分野

◆ RoRの特徴

- ✓ 高い生産性
- ✓ Webアプリケーション開発に特化した開発環境
- ✓ プログラムの変更が容易

◆ RoRが向いているシステム



1.2 エンタープライズで利用するためには

- RoRの特長を活かしたアジャイル開発ができること
- 開発、保守を行うことができる人材が豊富に存在すること
- 容易に開発環境、実行環境が構築できること
- 開発時、運用時に適切なサポートが受けられること
- 導入コストが低いこと

2. RoR(Ruby on Rails)適用の検証項目

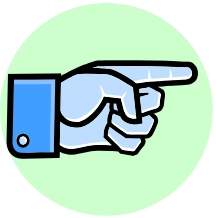
2.1 RoR適用にあたっての検証項目

- ◆ 生産性
  - ✓ 他環境と比較して、良いのか悪いのか？
  - ✓ 注意すべき観点は何か？
- ◆ 信頼性
  - ✓ 性能:
    - インタプリタであるが実用に耐えられるか
  - ✓ 品質:
    - 不良の作りこみはどうか？
    - 摘出のし易さや修正コストはどうか？
- ◆ 拡張性
    - システムの変更はし易いか？
    - 変更に掛かるコストはどの程度か？

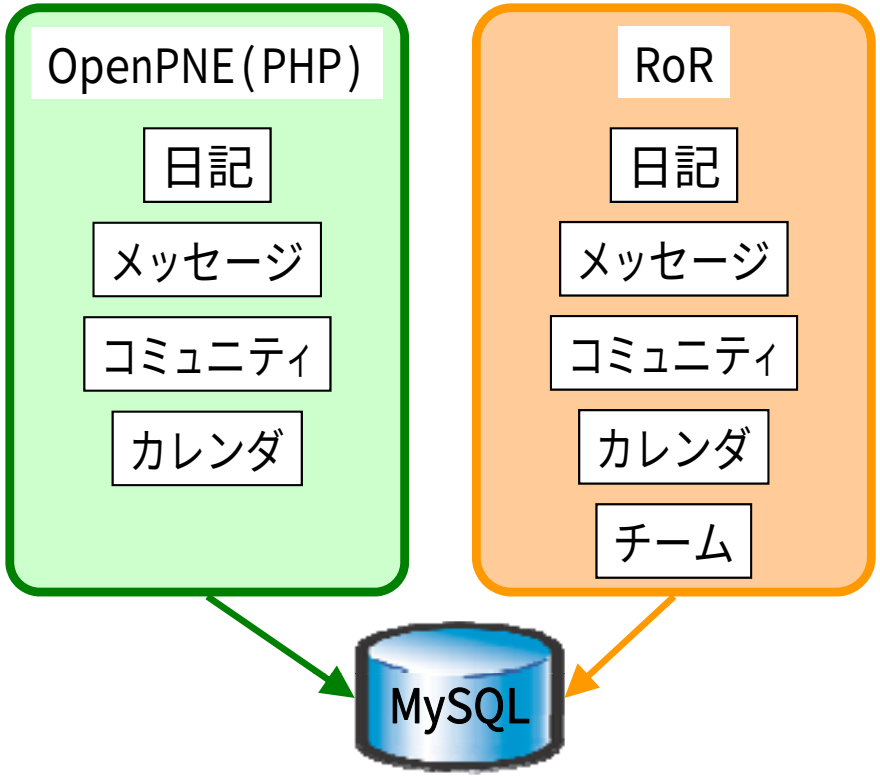
システム開発にRoRを適用し、評価を実施

2.2 検証対象システム

- 大まかな機能仕様が確定している
- 既存システムとの比較が可能であること
- 開発後、ユーザの観点からシステムの実力を評価できること
  - ✓ 毎日利用するシステムに適用する
  - ✓ 実際に業務として利用する



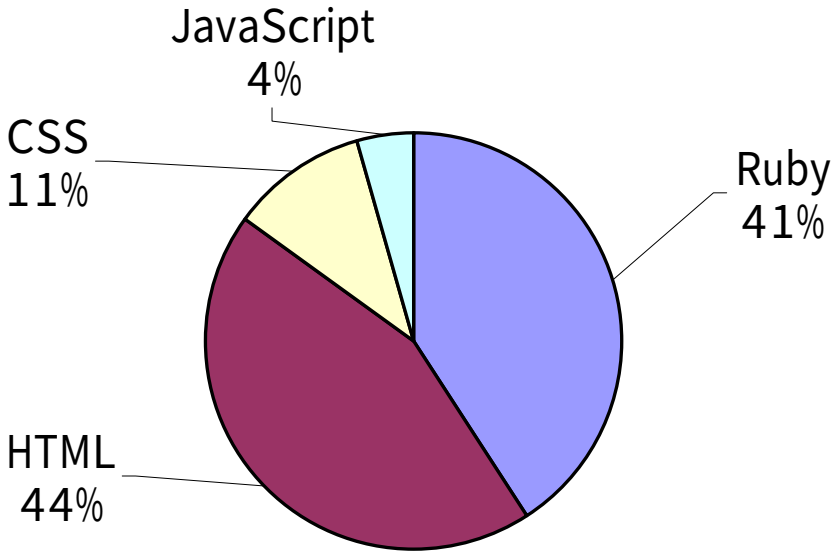
PHPで作られたOpenPNEを
RoRで書き換え



2.2 画面サンプル

2.3 開発規模

画面数	99［画面］
ステップ数	8,119［step］



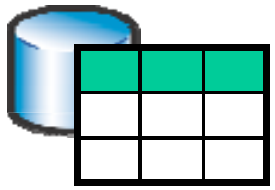
ビュー
- HTML 3,573
- JavaScript 873
- CSS 354

コントローラ
- Ruby 1,919

モデル
- Ruby 917

ヘルパ
- Ruby 309

42テーブル

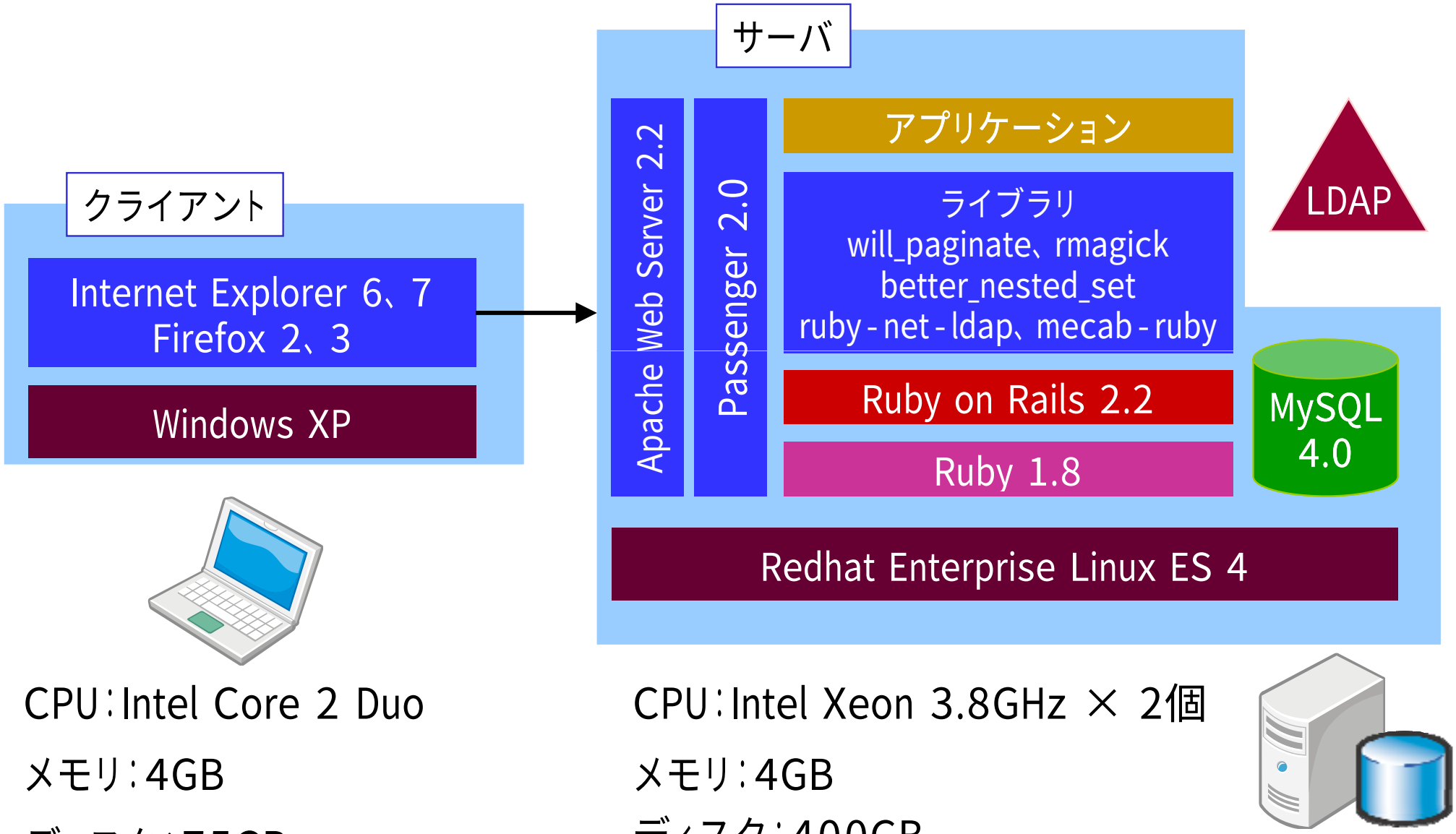


バッチ
- Ruby 174

2.4 利用状況（２００９年）

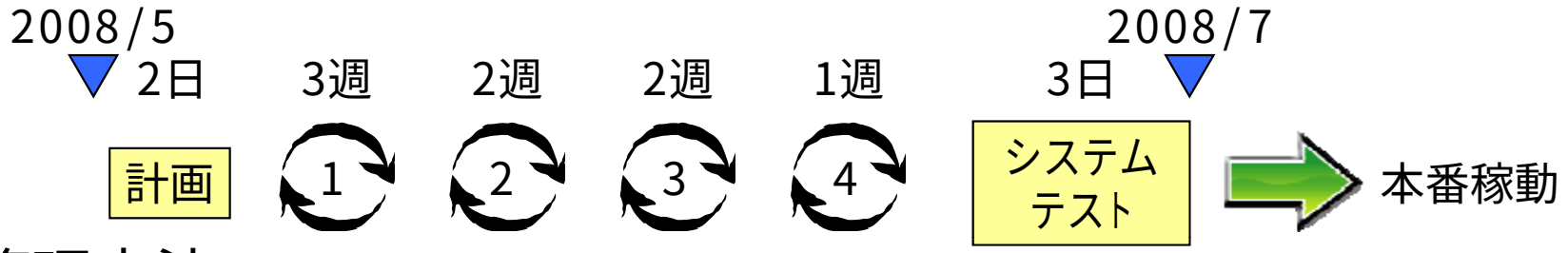
登録ユーザ数	5,000 [ユーザ]
ログインユーザ数	550 [ユーザ/日]
ページ閲覧数	24,000 [PV/日]
投稿数	250 [件/日]

2.5 実行環境



2.6 プロジェクト管理方法

◆ プロジェクト概要
- ✓ アジャイルを意識した管理
- ✓ イテレーションを4回実施



◆ 管理方法
- ✓ 機能　：具体的に実現する機能群
- ✓ タスク：機能を実現するための個々の作業

機能	8
タスク	182

◆ 管理ツール
- ✓ オープンソースのtracを活用
  - マイルストーンで機能を管理
  - チケットでタスクを管理

3. 実力の検証

本来なら、１００プロジェクト以上の値から
評価するべきですが、今回は参考値として
本プロジェクトの値をご紹介します。

3.1 生産性の検証

- アジャイル開発
  - ✓ 仕様書は作成していない
  - ✓ ウォーターフォール型の開発と単純比較することは危険
- 体制

開発者	A	B
Ruby経験	×	○
その他の言語	Lotus Notes、他	Java、他
Webアプリケーション	○	○
経験年数	11年	4年

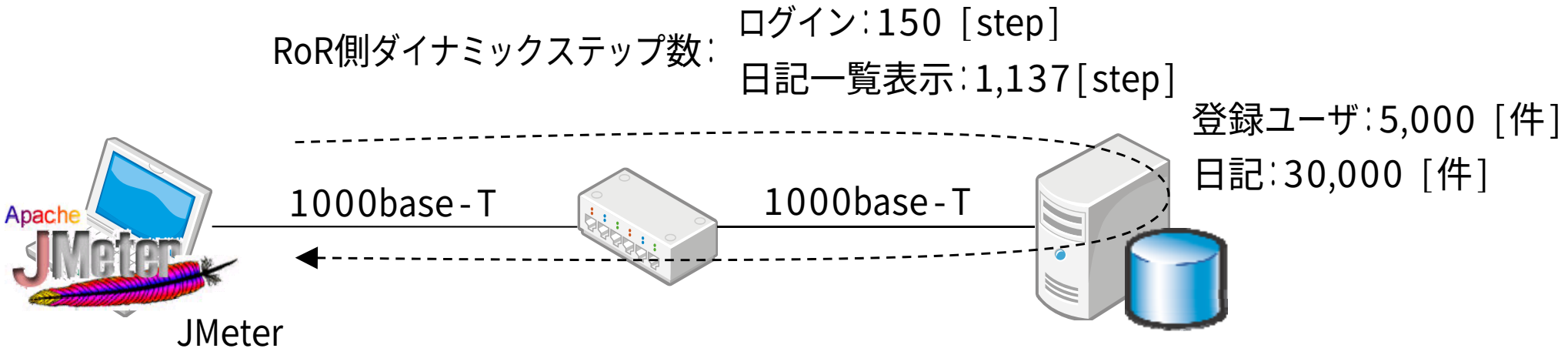
- 開発工数：638［h］（計画 ～ システムテスト）
- 開発規模：8，119［step］、99［画面］
- 生産性

$$\frac{8{,}119\ [\text{step}]}{638\ [\text{h}]} = 12.7\ [\text{step/h}] \qquad \frac{8{,}119\ [\text{step}]}{99\ [\text{画面}]} = 82\ [\text{step/画面}]$$

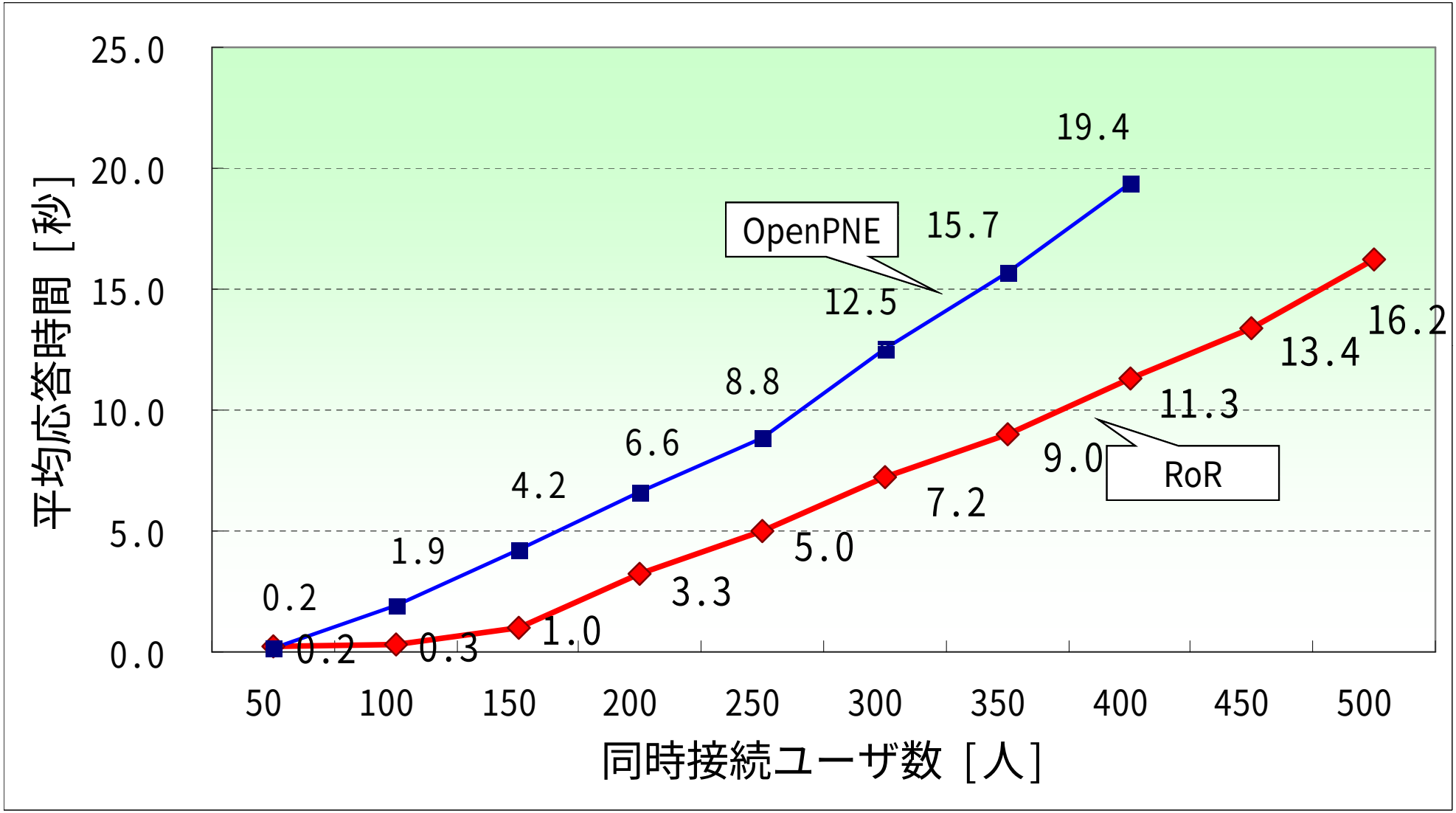
$$\frac{638\ [\text{h}]}{99\ [\text{画面}]} = 6.4\ [\text{h/画面}]$$

3.2 性能測定の環境と測定方法

- ◆ 測定環境
  - ✓ CPU　：Intel Xeon 3GHz × 1個
  - ✓ メモリ　：5GB
  - ✓ ディスク：73GB(SCSI) × 1個
- ◆ 負荷
  - ✓ 同時接続ユーザを50から500まで測定
- ◆ シナリオ
  - ✓ ログインして日記一覧を表示（30,000中20件表示）
  - ✓ 同じシナリオでRoRとOpenPNEを測定



3.3 測定結果

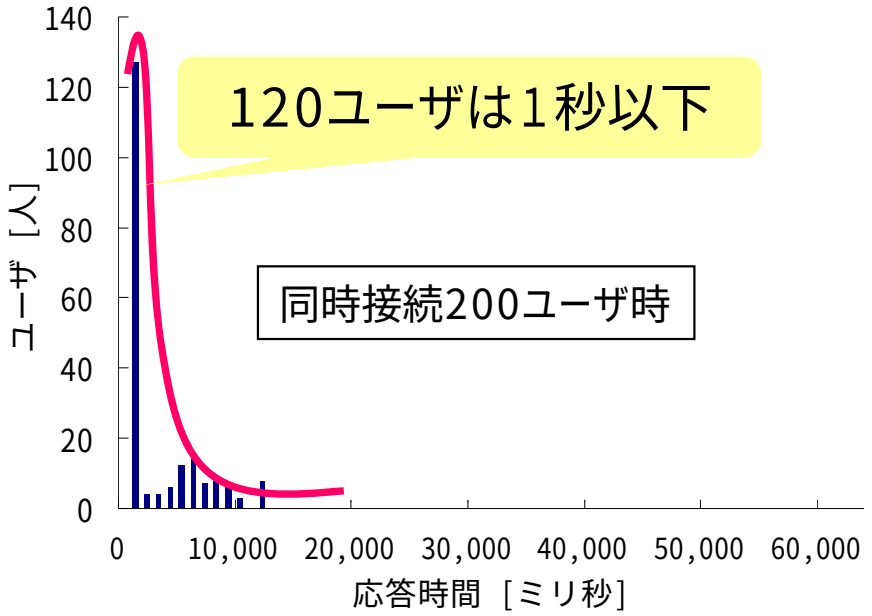


OpenPNEにおいてユーザ450、500は無応答エラーが発生し、測定不可

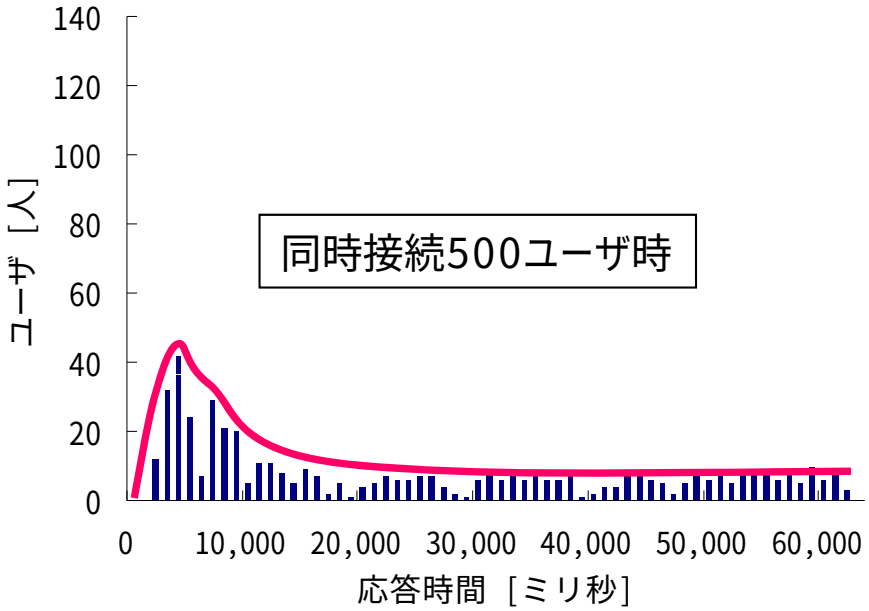
3.3 性能の評価

実用に耐えられるかの検証

◆ 200ユーザ接続時
- ✓ 平均応答時間は、3.3秒であるが、応答時間1秒以内が120ユーザとなる
- ✓ 最大値は12秒
- ✓ 日々の利用には問題のない性能



◆ 500ユーザ接続時
- ✓ 平均応答時間は、16.2秒であるが、応答時間3秒以内が40ユーザとなる
- ✓ 1秒以内はゼロ
- ✓ 20秒以上は250ユーザとなる
- ✓ 最大値は60秒
- ✓ ハードウェアの増強が必要と考える



3.4 不良の発生状況

◆ テスト方法
- ✓ 開発メンバーによるモンキーテスト
- ✓ テスト環境を使用し、数十名によるユーザテスト

◆ 平均修正工数：2.6［h/件］

本番リリース

イテレーション	#1	#2	#3	#4	#5	#6	計
バグ［件］	31	8	1	1	16	0	57
修正工数[h]	87.0	21.6	3.0	1.0	38.4	0.0	151
件数あたりの修正工数［h/件］	2.8	2.7	3.0	1.0	2.4	-	2.6

◆ イテレーションの進捗で修正に掛かる工数に変化はない

3.5 拡張性の検証

- リリース時点
  - ✓ 機能改善数：113［件］
  - ✓ イテレーションあたりのタスク数：28.3［件/回］
  - ✓ 平均工数：4.3［h/件］
- 本番稼動の後
  - ✓ 機能改善数：95［件］
  - ✓ イテレーションあたりのタスク数：47.5［件/回］
  - ✓ 平均工数：4.2［h/件］

本番リリース

イテレーション	#1	#2	#3	#4	#5	#6	計
機能改善［件］	69	29	4	11	78	17	208
工数［h］	283	143	19	36	321	75	877
件数あたりの修正工数［h/件］	4.1	4.9	4.8	3.3	4.1	4.4	4.2

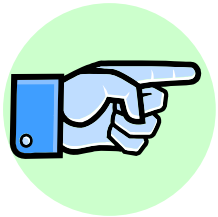
3.6 人材の育成

RoRを使ったことがない人材が多い

RoR未経験者の戦力化が必要

◆ 技術者の育成
- ✓ 知識の教育
- ✓ 作り方の教育

◆ 作業の標準化
- ✓ 統一した実装をさせる
- ✓ 同じ問題を繰り返さない



開発ガイドラインの作成

3.6 技術者の育成

◆「Rubyプログラミング講座」：２日間
- ✓ Rubyプログラムを作成するための基本的な知識を習得

◆「RoRプログラミング講座」：２日間
- ✓ 実習を通じてRoRを使ったアプリケーション開発を習得
- ✓ 10画面程度、2～3[kstep]を開発

◆ 学習に必要な教材を用意

開発チュートリアル	独学で簡単なアプリケーションを作るための教材
リファレンスコード	お手本となるソースコード
サンプルアプリケーション	オープンソースとして利用可能なサンプルプログラム（7つ）

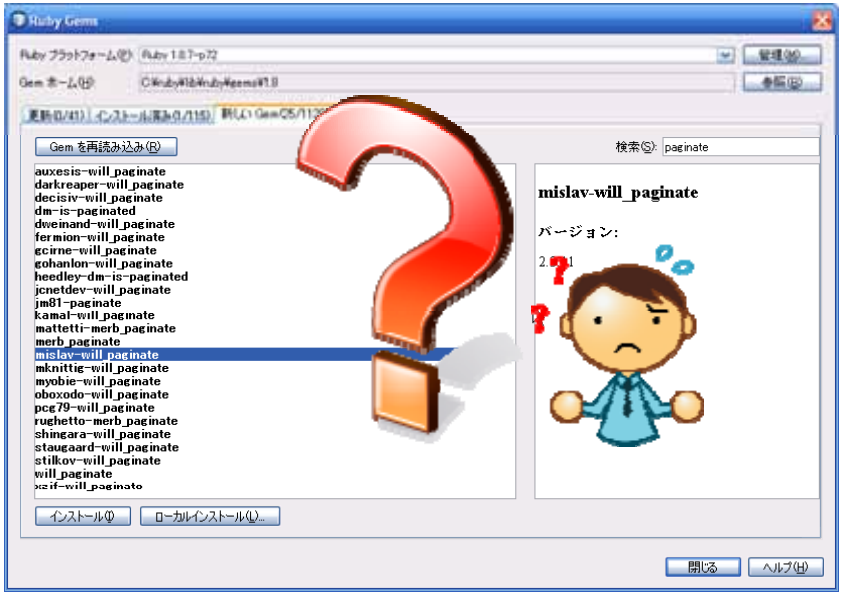
3.6 開発ガイドライン

◆ 開発に必要な各種ガイドラインを作成

開発手法ガイド	クラス設計、命名規則、コーディング規約など 開発時のルールを記載したガイド(43ページ)
開発環境構築ガイド	開発の環境を構築するためのガイドとツール、 ライブラリの利用方法を記載 (17ページ)
実行環境構築ガイド	実行環境の構築ガイド(17ページ)

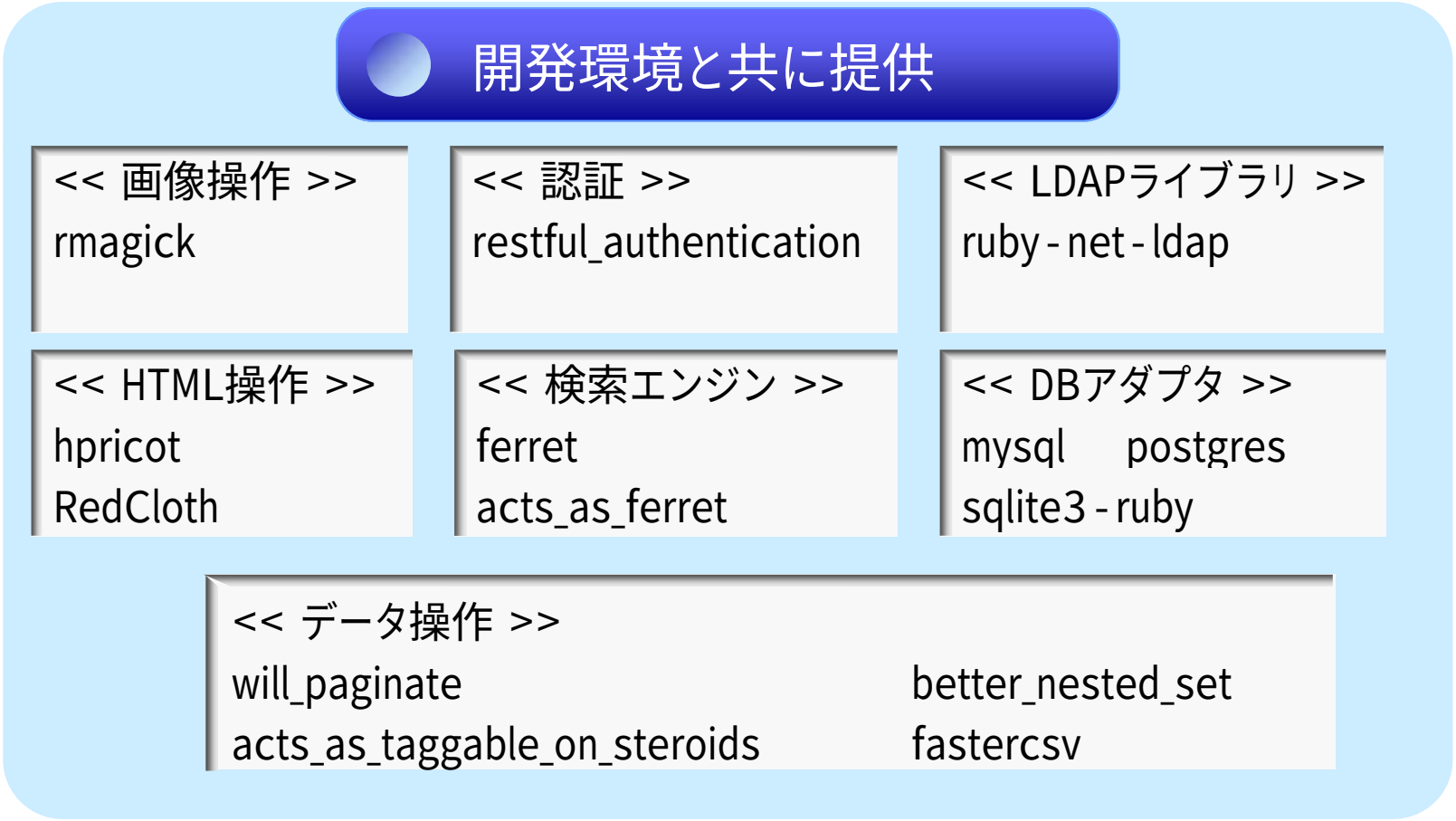
3.7 ライブラリを使いこなす

- ライブラリを使いこなすことが困難
  - ✓ 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がある
  - ✓ 1つのライブラリから派生したものが多くあり、それぞれ機能が異なる
  - ✓ 例：データのページング・ライブラリ
    - 数：25種類
- ライブラリを見つけ出すには手間がかかる
  - ✓ 見つけ出すことができず、作りこんでしまった
    - 例：検索機能



3.7 推奨ライブラリを選別

- 推奨ライブラリを選別
  - ✓ よく利用されるライブラリを検証し、有用なものを選別
  - ✓ 各種アダプタ（DB、LDAP）、検索エンジン、データ操作、認証など
- 使い方がわかるサンプルを用意



4. 今後の取り組み

4.1 Rubyソリューションのメニュー

システム開発

プロトタイプ開発

アジャイル開発コンサルテーション
- プロジェクトコンサルテーション
- 開発コンサルテーション

システム診断･改善

開発環境提供

運用、保守

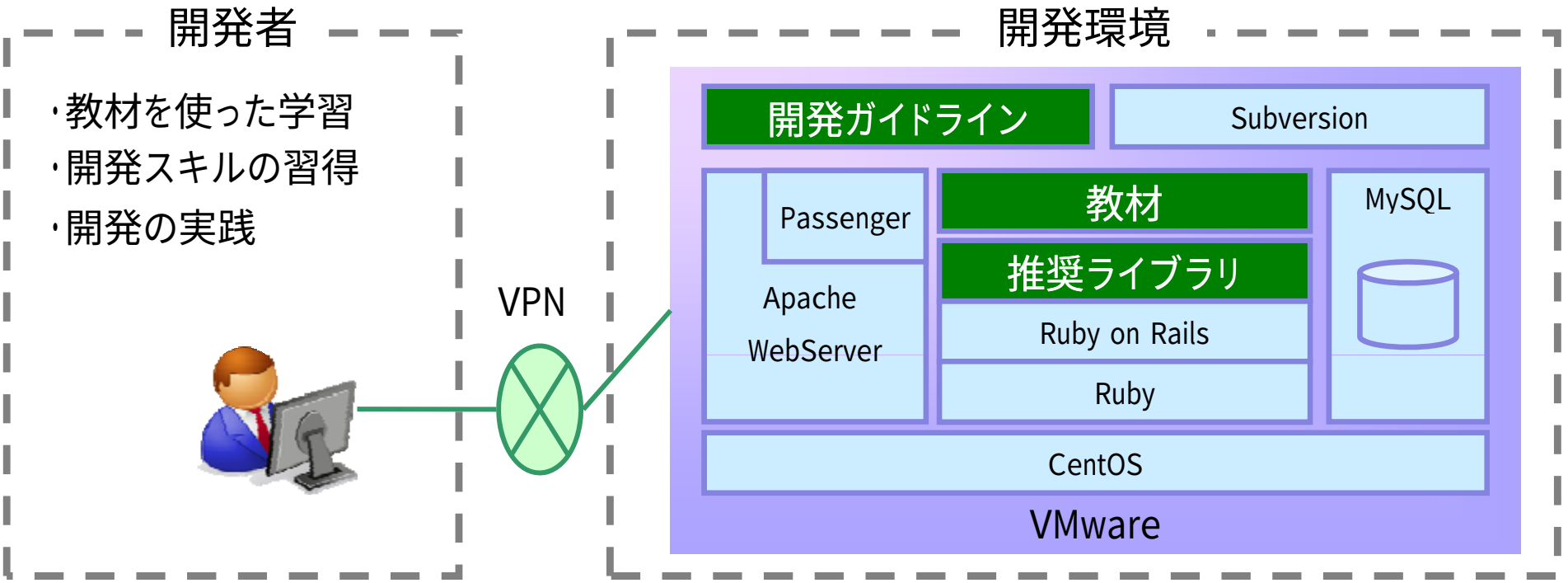
教育
- セミナー
- 学習環境提供サービス

実行環境構築

実行環境提供

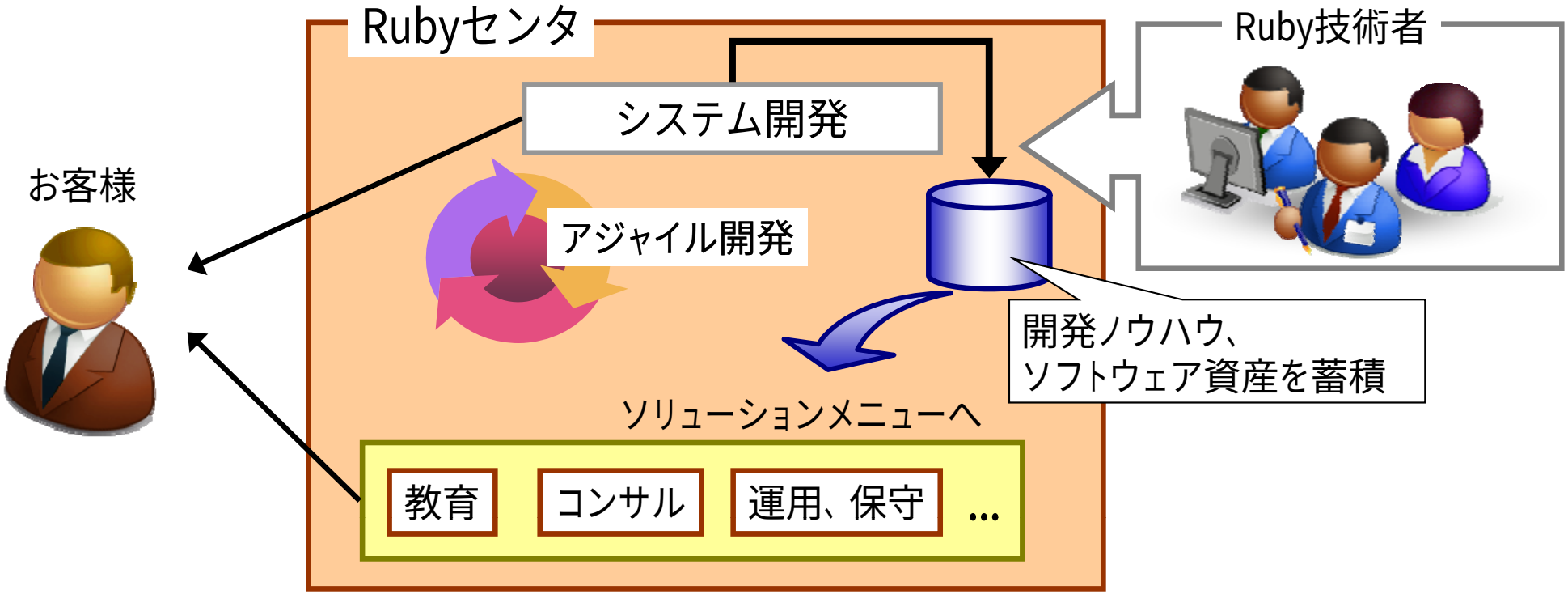
4.2 学習環境提供サービス

- ◆ 学習環境をPaaSとして提供
  - ✓ 導入と同時にRoR環境を利用可能
  - ✓ 月額サービスで必要な期間だけ利用
- ◆ 学習コンテンツ
  - ✓ RoR学習の教材 (チュートリアル、リファレンスコード、サンプルアプリ)
  - ✓ 開発に必要な開発ガイドライン
  - ✓ 推奨ライブラリ



4.3 システム開発

- Ruby案件を扱う専門組織「Rubyセンタ」を設立
- 技術者を集めて、Rubyの実力を発揮するための徹底したアジャイル開発を実施
- 開発ノウハウ、ソフトウェア資産を蓄積
- その結果を教育サービス、コンサルテーション、運用保守サービスに活用



ご清聴ありがとうございました。

- Apacheは、Apache Software Foundationの登録商標または商標です。
- DB2は、米国International Business Machines Corporationの米国およびその他の国における商標です。
- Firefoxは、Mozilla Foundationの米国およびその他の国における登録商標です。
- Linuxは、Linus Torvaldsの米国およびその他の国における登録商標または商標です。
- Intel, Core 2 Duo, Xeonは、 Intel Corporationまたはその子会社の米国およびその他の国における登録商標または商標です。
- MySQLは、MySQL AB社の米国およびその他の国における登録商標です。
- NetBeansは、米国Sun Microsystems, Inc.の米国およびその他の国における登録商標または商標です。
- OpenPNEは、株式会社手嶋屋の登録商標です。
- ORACLEは、米国Oracle Corporation の登録商標です。
- Red Hatは、米国Red Hat, Inc.ならびにその子会社の登録商標です。
- UNIXは、The Open Groupの米国およびにその他の国における登録商標です。
- VMwareは、VMware, Incの米国およびその他の地域における登録商標または商標です。
- Windows、Windows XP、IIS、InternetExplorer、SQL Serverは、Microsoft Corporationの米国およびその他の国における登録商標または商標です。

- その他、記載されている会社名、商品名は、各社の登録商標または商標です。