

COBOLエンジニアをRubyエンジニアに！

～ Ruby業務システム開発 ～

2012年5月25日

株式会社 テクノプロジェクト

代表取締役社長 吉岡 宏

TPJ



1984(S59)年設立 富士通40%、山陰中央テレビ40%、山陰合同銀行10%
本社:島根県松江市

売上 2011年度 33.8億(内:機器2.0億)

社員数 190名

主なソリューション

自主ビ

CIMA(クリニック・小病院・療養型病院向け電子カルテ)、医療ネット

Ruby開発(業務系)、地域の製造業・サービス業向けソリューション開発

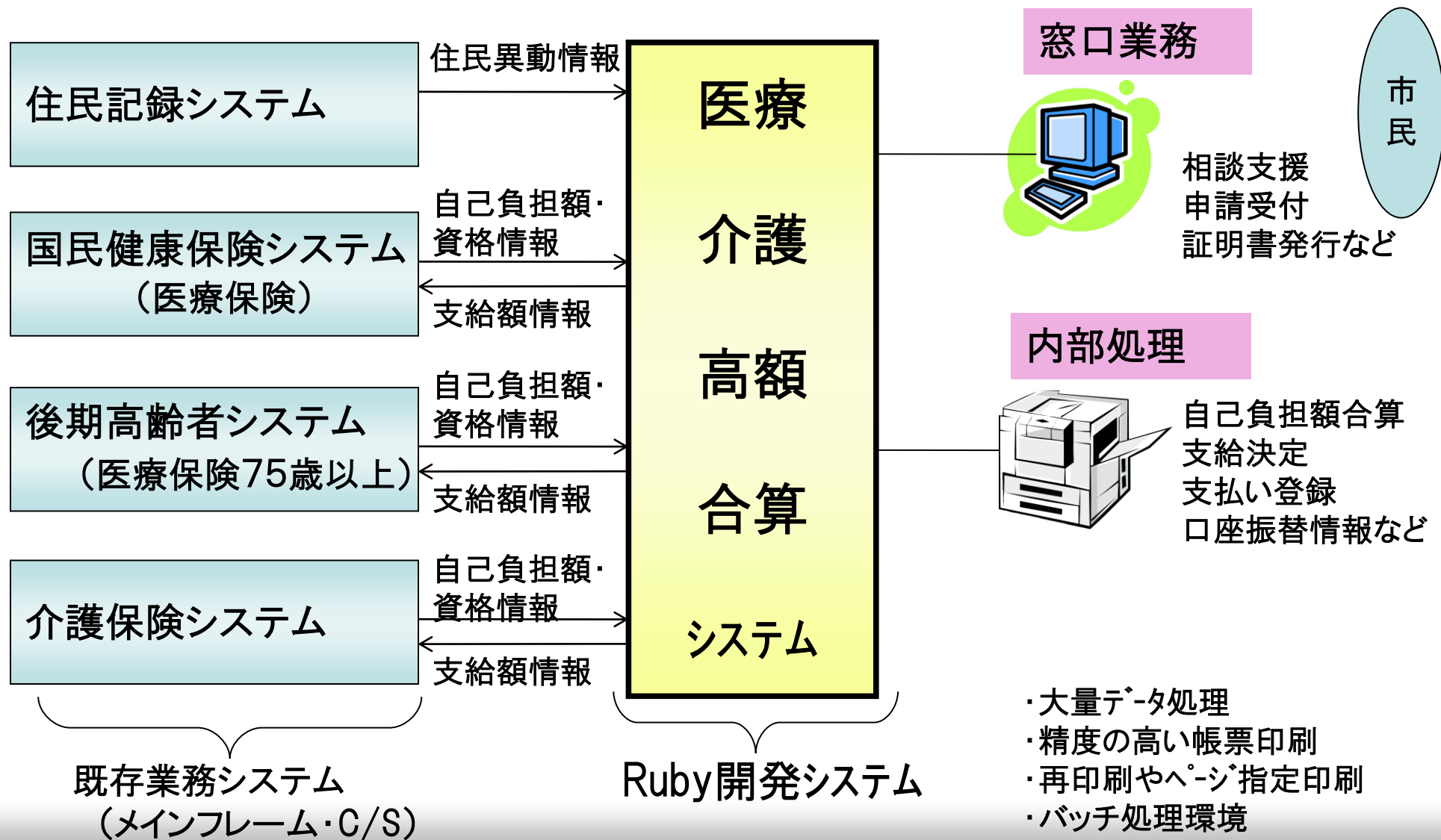
グループ企業

(株)マツケイ FCA会員 1975(S45)年設立 2003年:全株式取得
データセンター、データパンチ、システム開発
社員数 120名

- 2007年 1月:Ruby開発へ参加決定、3月:Ruby勉強会開催(Rails)
7月:IPA自治体向け業務システムの実証実験採択(7千万)・・・高額合算(松江市)
- 2008年 IPAのRuby調査事業、施設予約など
- 2009年 地域企業向けシステム開発、県の部局システム
県外IT企業支援、教育コース
マツケイ:県の先進開発補助:帳票エディターThinReports
- 2010年 地域企業向けシステム開発、県の部局システム
医療機関の管理システム
県の先進開発補助:アジャイル開発(製造業)
- 2011年 松江市)MISALIO周辺システム、県の部局システム(予算編成)
教育コース
- 2012年 全県医療ネットワーク、島根県)予算編成

Ruby開発推進室 社内の開発ノウハウの蓄積と活用 ⇒ 開発標準作成 ⇒ 2012終了

2007年度 IPA公募事業 医療・介護の高額合算システム(松江市殿)



Ruby業務システム開発の実績

2007年 IPA(情報処理推進機構)実証事業

IPAオープンソフトウェアセンターがRuby/Railsを基幹業務に適用する実証事業を実施。

⇒ 松江市殿)医療・介護高額合算システム
(日本初:バッチ処理を含む基幹業務全体)

約50人月 2007. 9～2008. 3

オン:14本、バッチ:24本

業務開発チーム:6名

(株)テクノプロジェクト、(株)マツケイ

共通技術チーム:3名

(株)ネットワーク応用通信研究所



支給申請入力

窓口業務担当ユーザ 2008/01/22 16:18:25 ログアウト

被保険者状況照会 自己負担証明書申請入力

登録 修正 削除 状態: 受付 > 合算済 > 確認済 > 審査済 > 発行済 > 依頼中 > 支払済

制度: 2:後期 年度: 2008 被保険者番号: 3000003800 個人番号: 1000003340 被保険者番号検索 受付番号: 0000000006

被保険者名: 高額太郎(コカ'タロウ) 区分: 合算 世帯集約番号: 3003120000 給付制限状況: 所得区分: 70未満:-- 70以上:現
生年月日: 昭和5年4月25日 年齢: 77歳 性別: 男 住所: 松江市松江町217-1

| 主 | 個人番号 | 氏名(氏名カナ) | 生年月日 | 年齢 | 性別 | 制度 | 保険者 | 被保険者番号 | 資格加入期間情報 | | | 支給申請情報 | |
|---|------------|--------------|-----------|-----|----|----|-------|------------|----------|----|----------|------------|----------|
| | | | | | | | | | 取得 | 喪失 | 得喪事由 | 受付番号 | 申請日 |
| | 1000003340 | 高額太郎(コカ'タロウ) | S05.04.25 | 77歳 | 男 | 後期 | 後期島根県 | 3000003800 | 09/03/26 | | 転入による資格取 | 0000000006 | 09/11/15 |
| | | | | | | 介護 | 介護松江市 | 4000008200 | 09/03/26 | | 転入による資格取 | | |
| | | | | | | 後期 | 鳥取県後期 | 3900000200 | | | | | |
| | | | | | | 介護 | 境港市介護 | 4900000300 | | | | | |
| | 1000003350 | 高額花子(コカ'ハナ) | S07.05.18 | 75歳 | 女 | 後期 | 後期島根県 | 3000003900 | 09/03/26 | | 転入による資格取 | 0000000033 | 09/11/15 |
| | | | | | | 介護 | 介護松江市 | 4000008300 | 09/03/26 | | 転入による資格取 | | |

自己負担額証明書登録: 登録 修正 削除

申請日: 2009/11/15 本人電話番号: 0852-12-3515

申請者: 郵便番号: 690-0041 住所: 松江市松江町217-1
氏名: 高額太郎 電話番号: 0852-12-3515

送付先: 1:本人 送付先: 郵便番号: 住所: 氏名: 電話番号:

支払方法: 1:口座振替

口座情報: 金融機関: 0001 松手銀行 店舗: 001 本店 金融機関検索
口座種別: 1:普通 口座名義人: 高額太郎 口座名義人カナ: コカ'タロウ 口座番号: 0000182

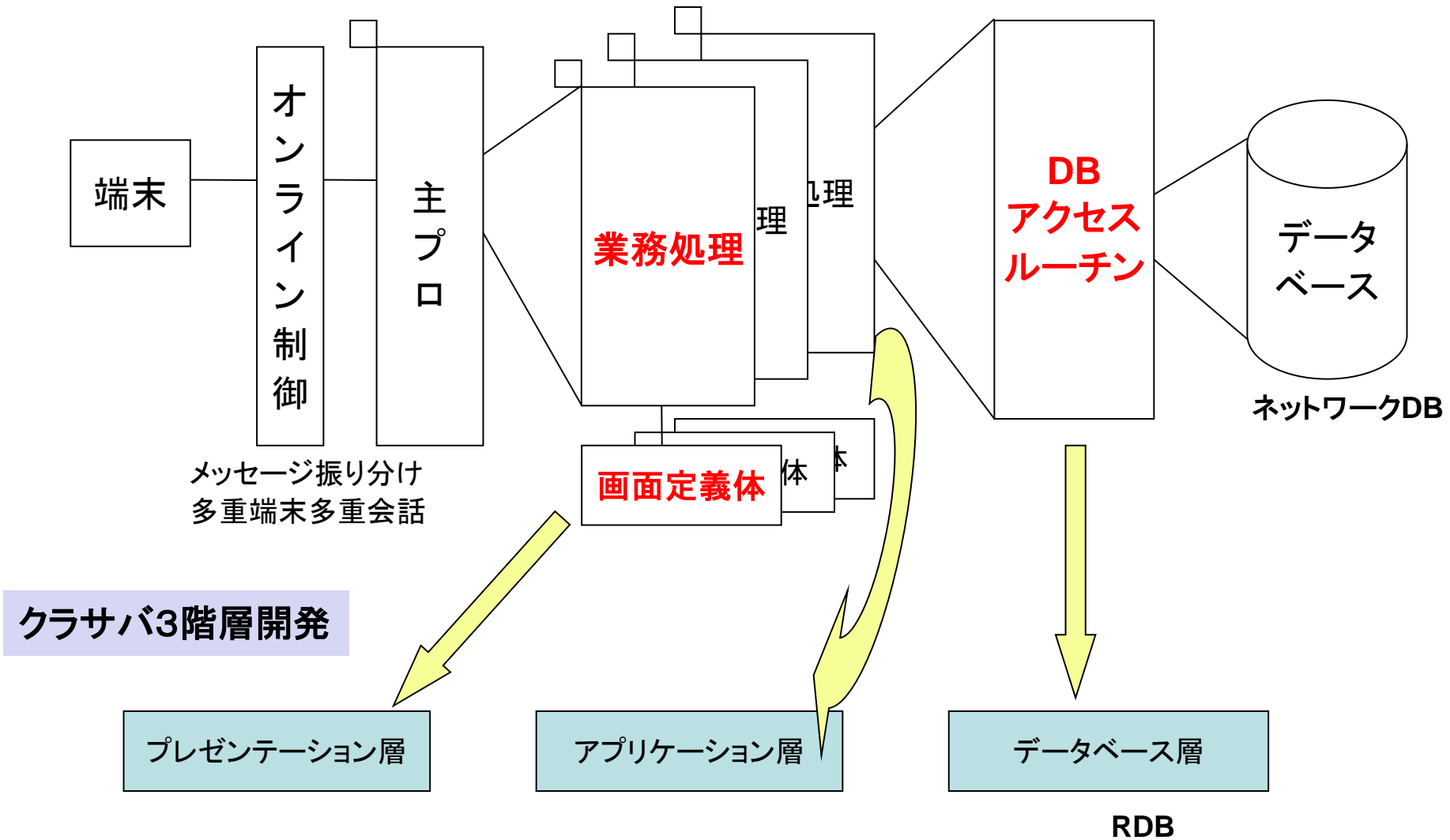
Railsでも、C/Sシステムと同様なキメ細かな画面

窓口で市民が来た時の対応

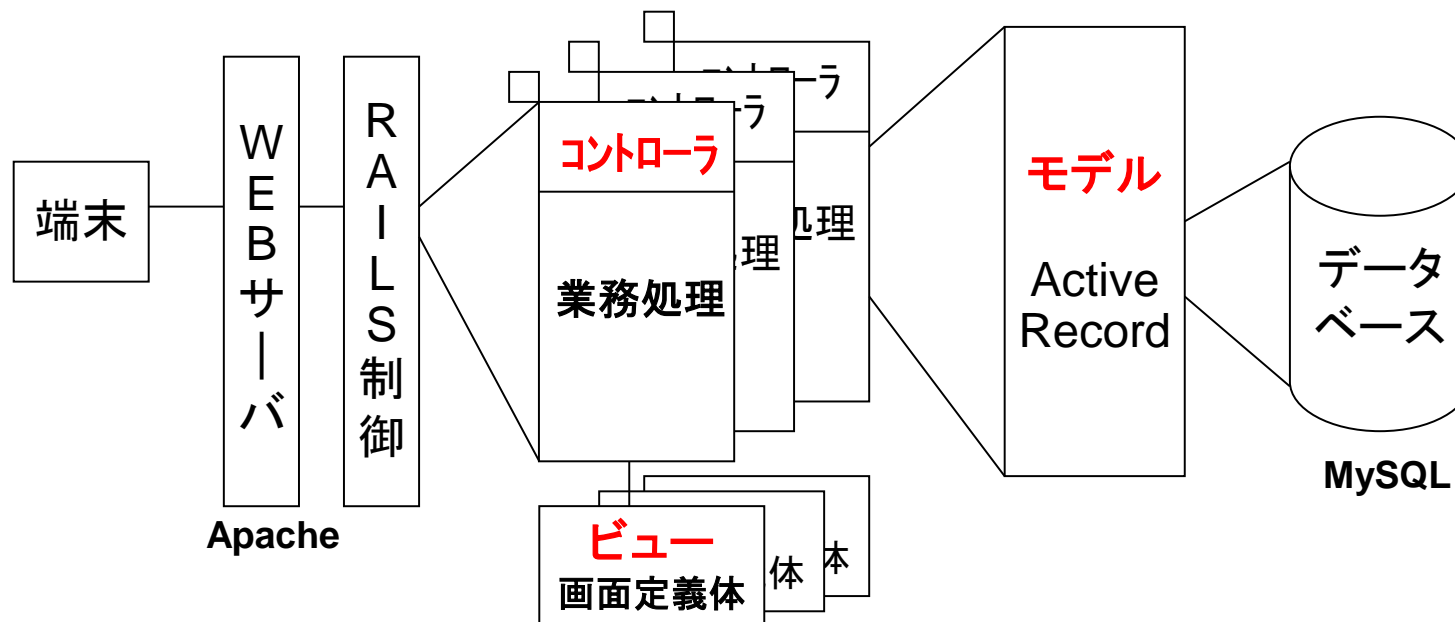
(上段:申請者情報、中段:受給履歴、下段:申請内容)

確認 戻る 実行 印刷 終了

メインフレームのオンライン構造



Ruby on Rails (機能分割: モデル・ビュー・コントローラ)



※Ruby on Rails

- M:モデル**(DBマクロ) 検索・挿入・更新・削除のSQL(一部業務処理を含む)
- V:ビュー**(画面定義) HTML・ヘルパー ... HTMLエディター(Webデザイナー)
- C:コントローラ**(オンライン制御・業務処理) ... 業務ロジック

開発標準: COBOLerの復活(2007~)

ベテランSE(COBOLer)がWebシステムの開発に参加できる

- ①ウォーターフォール開発 (設計⇒製造⇒テスト)
- ②厳格なコーディング規約 (Rubyの自由度をなくす)
- ③日本語のDB項目名称(設計書とソースの見易さ)
- ④SQL記述(判り易さ、但し便利な部品が使えない)
- ⑤バッチ処理も同一言語で開発
- ⑥構造化設計(業務ロジック重視、従来のドキュメント類)
- ⑦少ない部品(少量の部品から始める、覚え易さ)
- ⑧画面定義(HTML)は、専任化:Webデザイナー
- ⑨クラス設計:テーブル毎の業務ロジックを含めたアクセスルーチン

COBOLer向けRuby/Rails教育コース by テクノプロジェクト

COBOLエンジニア向けRuby教育(1)

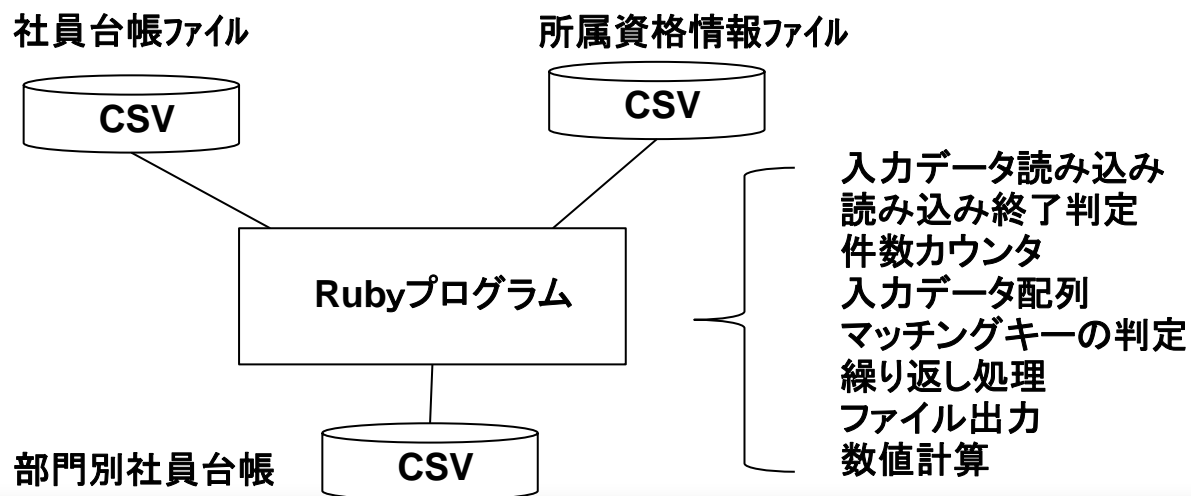
ネック

- ・オブジェクト指向の概念が判らない(クラス、メソッド、継承、・・・)
- ・サブルーチン(外部ライブラリ)が膨大
- ・命令が多い(業務システムでは不要なものが多い:文字列操作)
- ・Rails Webアプリの構造・仕組みが判らない

COBOLエンジニア向けRuby開発入門(Ruby編) 1日コース

- ・COBOLコーディングとの比較
- ・言語の基礎 メソッド・変数・定数・配列・予約語

演習問題



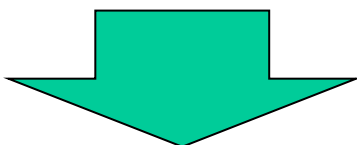
COBOLエンジニア向けRuby教育(2)

COBOLエンジニア向けRuby開発入門(Ruby on Rails編) 1日コース

- ・開発環境(フォルダ・プロジェクト・ソースコード)
- ・DB(データベース作成・DB操作コマンド・DBアクセスルーチンActiveRecord)
- ・Webアプリの構造、Railsの仕組み(コントローラとビュー)

演習問題

- ・自動生成機能で作成したM・V・Cの修正(機能追加)
- ・見やすい多機能な画面の作成 ⇒ スタイルシート
- ・画面の細かな制御(項目の入力抑制・色変更・数値編集・日付編集など)



通常のRuby・Ruby on Railsの研修を受講する

※当初のネックが解消された状態。

⇒ オブジェクト指向の特徴を活かしたWebアプリの開発スキルを習得できる。

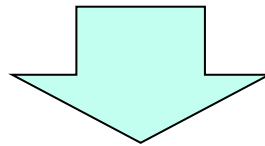
開発標準への不満

面白くない！！

- ・新しい便利なものを使えない
ライブラリ、ツール

アジャイル開発

- ・製造業の製造機器管理システム
- ・スクラムの伝道師による指導
- ・島根県の公募事業(半額補助)



熱い集団 ==> 開発の楽しさ

- ・新しいことへのチャレンジ
- ・英文サイトの調査(米国発信の情報が多い)

でも、全てのエンジニアが出来ることではない。
アジャイルは少人数のチームだから出来る。

スプリントプランニング



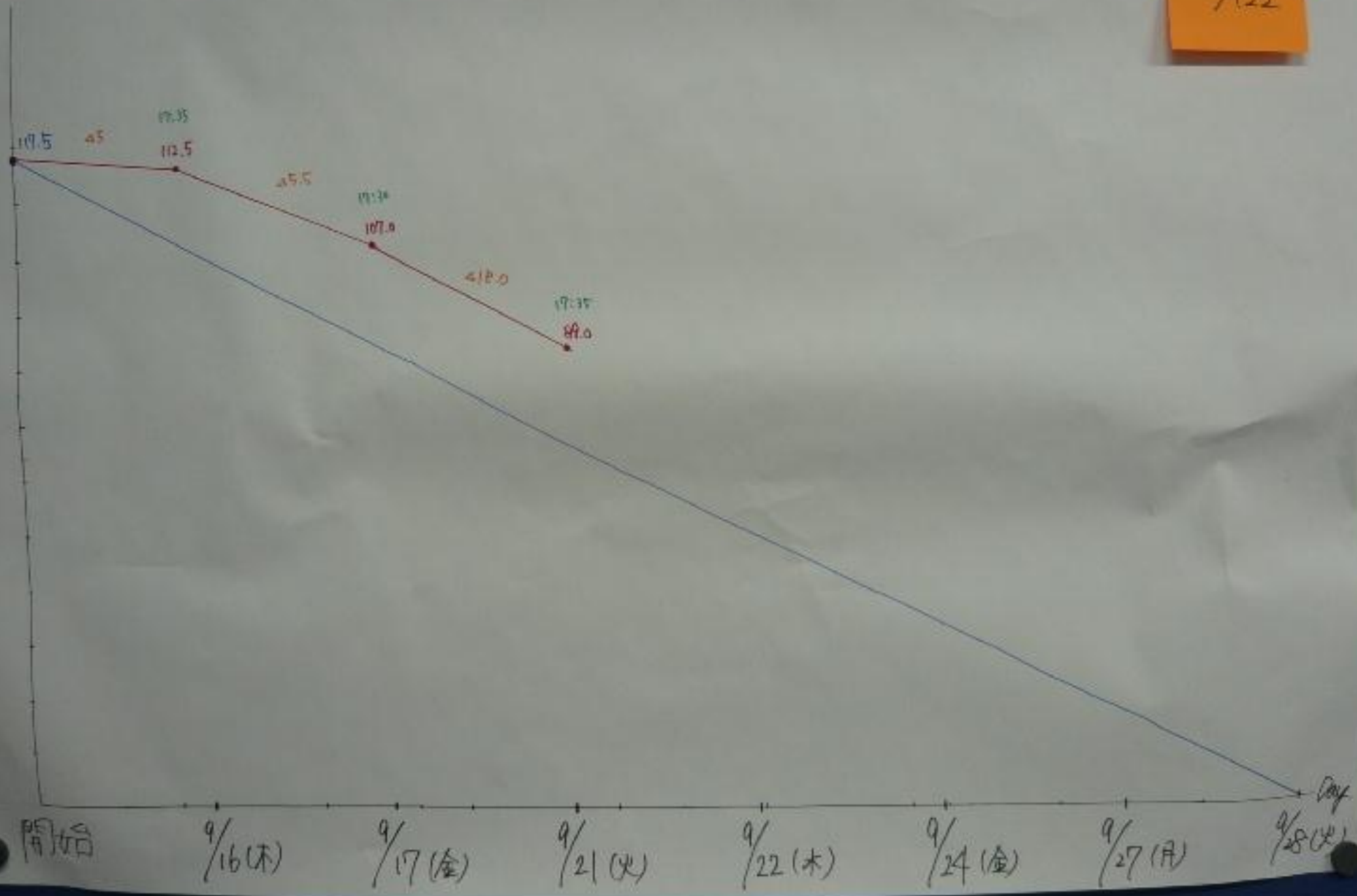
タスクボード

The image shows a Kanban board on a blue background. The board is organized into three main columns: **Todo**, **Doing**, and **Done**. The **Todo** column contains a large grid of yellow sticky notes, with some notes having handwritten text and others being blank. The **Doing** column contains a vertical stack of yellow sticky notes, with some notes having handwritten text and others being blank. The **Done** column contains a vertical stack of yellow sticky notes, with some notes having handwritten text and others being blank. There are also several blue and orange sticky notes scattered around the board, some with handwritten text and others being blank. The board is used for tracking tasks and their progress.

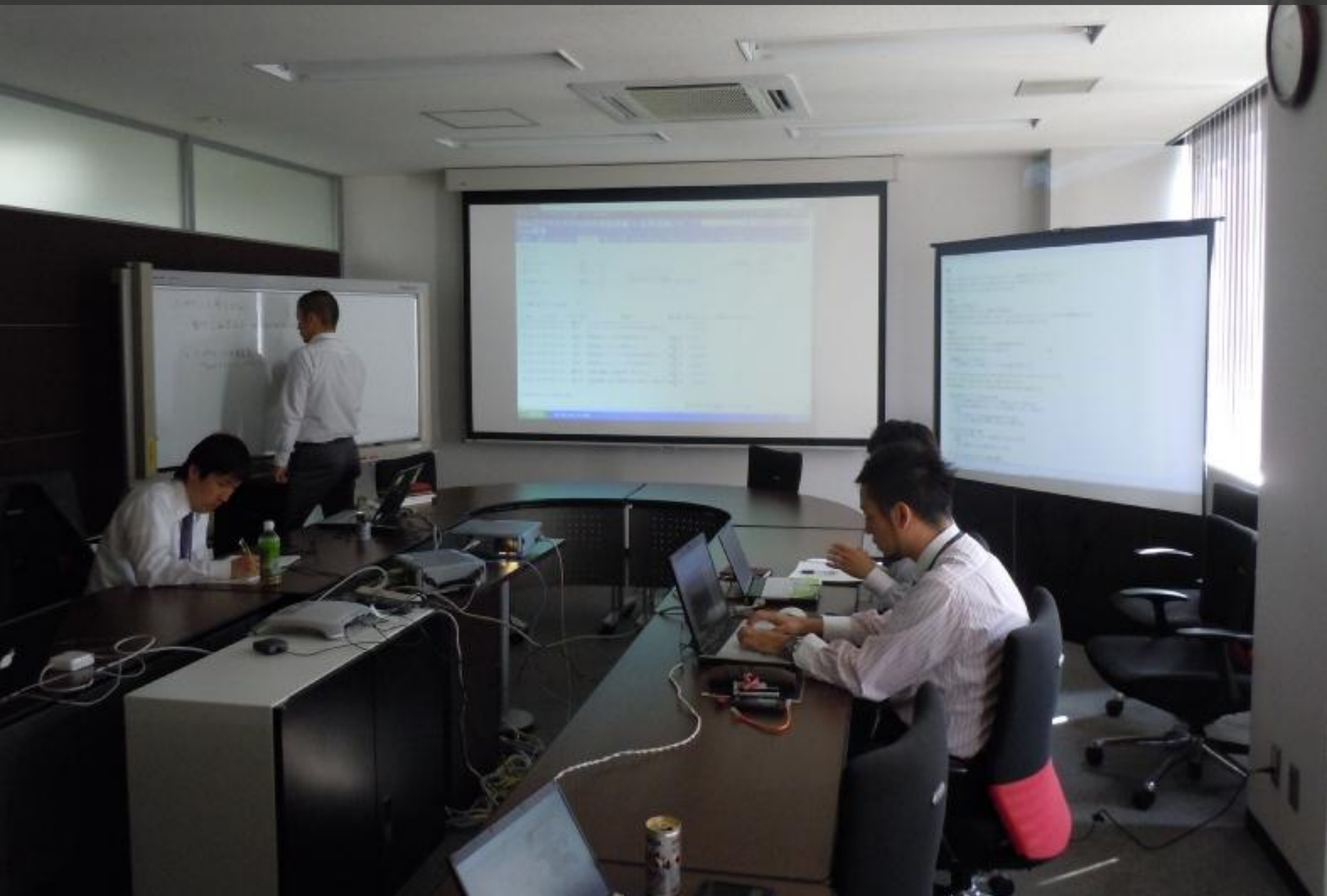
スプリントバーンダウンチャート

117.5
/ 122

残時間



スプリントレビュー



ペアワーク



リリースふりかえり



時系列に、「KPT」「スプリントバーンダウンチャート」「取り入れたプラクティス」などを壁に貼り、良かったこと/悪かったことをふりかえった。

アジャイル+業務システム開発+請負契約

開発手法

- ①開発方式とドキュメント
- ②開発チーム
- ③テスト駆動開発
- ④ソースレビューとペアプログラミング

ツール

- ⑤プロジェクト管理・開発ツール
- ⑥外部ライブラリの活用
- ⑦帳票作成ツール

参考 業務システム (ここでの想定)

- ・ 業務処理が複雑(処理手順が多階層化している、例外がある)
- ・ きめ細かな帳票(チェック用、管理用、お客様用)
- ・ 他システム連携

①開発方式とドキュメント

(1)設計:ウォーターフォール

業務システムは、一般的に制度・法律・慣習が詰まった複雑なもの。
DBアクセスでは、多くの更新系処理が発生する。イレギュラー処理。

⇒ 設計が重要、特にDB設計

最終ドキュメント:メンテナンス手引書として整備(含む納品物)

構成品管理表、基本設計書・詳細設計書、操作・運用マニュアル

※開発途中は、ホワイトボードのデジカメ化でもOK

(2)実装(PG設計～結合テスト):アジャイル=素早く、繰り返し

⇒ 繰り返しの中で、**早め早めの顧客確認(満足度)**と**開発チームの整合性確認**

※顧客リリースは2～3カ月の単位で段階的に行う。

実装は2～3週単位に。進捗と品質の確認を行う。

| | | |
|----|-------------------------------|-----------|
| 参考 | COBOLでは | 全工程(100%) |
| | 設計工程(基本設計、詳細設計) | 30% |
| | 製造工程(PG設計、プログラミング、単体テスト) | 40% |
| | テスト工程(結合テスト、総合テスト、運用テスト、操作研修) | 30% |

②開発チーム

短期開発(1年以内): **最初から全員投入**

⇒ チームワーク・コミュニケーション

PGの役割: SE補助作業・業務習得・先行開発

“若手エンジニアの業務ノウハウ習得の場” ← 全工程に参加

朝会(スタンドアップミーティング): 短時間→一人一人の進捗と予定の報告

× ダイナミックな作業割り振り(朝会で調整。前提:全員が同一な業務ノウハウ)

⇒ 複雑な業務では、不可能である。

③テスト駆動開発

テスト手続きの登録 & 自動テスト(含む、テストデータ登録)

効果: 品質向上(テストの自動化 + ソースの最適化 → リファクタリング)

単体・機能テスト(ロジックの内部仕様) 評価: △ ← 生産性と効果より

活用: ホワイトボックステスト(モデル・ビュー・コントローラ・モジュール単位)

繰り返し[1テストケース登録 → ロジック記述(数ステップ) → リファクタリング]

機能テスト(業務の外部仕様) 評価: ◎

活用: ブラックボックステスト → ユーザサイドのテスト

レグレッションテストの容易性、ライフサイクルでの活用 ⇒ 品質向上

④ソースレビューとペアプログラミング

(1)基本パターン

ソースレビュー コーディング規約遵守、第三者の目に触れることが重要
タイミングは単体テストの終了段階

開発者: 中級レベル以上は単独開発が基本

(2)オプション

ペアプログラミング

開発者: 初級レベル(Rails初心者、新入社員アソシエイト、業務初めて)

→ 早期立ち上げを目的に、中級レベルのエンジニアがトレーナ役を担当し、つまづきを最小化する

初心者が質問しやすい環境を強制的に作る

⑤プロジェクト管理・開発ツール

Ruby:1.9 系+ Rails:3系

Redmine 案件管理・ToDo

Subversion ソース履歴管理(Redmineと連携)

NetBeans 開発ツール(IDE+ソースエディター)

Cucumber 結合テストツール(画面単位、外部:ユーザサイトテスト)
日本語によるテスト仕様記述

Capistrano デプロイ(本番環境への更新モジュール反映)

※データベース

新規開発 → MySQL・PostgreSQL

既存クライアントサーバシステムのWeb化 → Oracle・SQLSeverの継続使用
複合キーのサポート部品あり

⑥外部ライブラリの活用

便利な部品の活用(ネット上+社内実績)

部品の活用度に応じて、生産性・品質向上

※ 部品の管理が重要

活用ノウハウ・逆引き・更新情報・ライセンス(MIT)・実績プロジェクト
利用者・使用上の注意事項・Ruby/Railsのバージョン 等

例: Pagenation 帳票・画面の一覧表および次ページ送りの部品

ライブラリ : GEMパッケージ、Railsプラグイン

⑦帳票ツール

・ Ruby用: **ThinReports** (帳票エディタ、帳票作成)

オープンソースで**株マツケイ**より提供(TPJグループ企業)

HTML5ビューア開発・PDF化機能・外字サポート

帳票作成ライブラリ



オープンソース帳票ツール：シンレポーツ

ThinReports Editor



日高克也氏

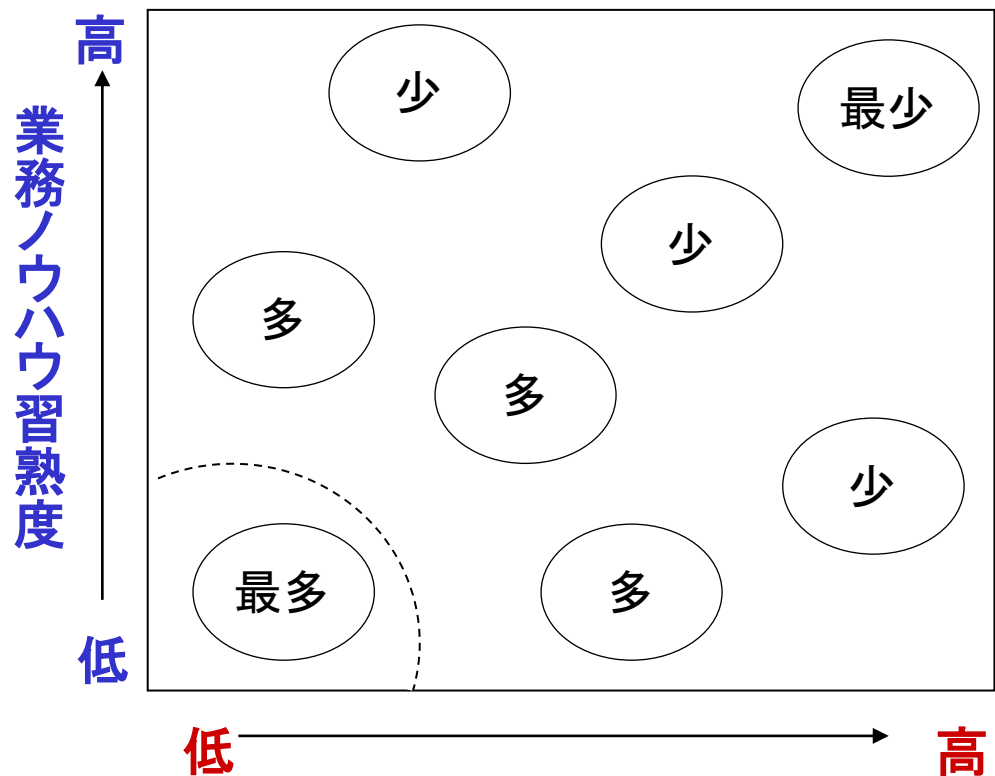


日経BPより

Matsukei Open Source Center
<http://osc.matsukei.net/>

繰返し回数最適化

繰返し頻度(顧客リリース回数)



プロジェクトチームの開発経験度
(Ruby/Railsアプリ、新技術を含む)

第三の軸

顧客要件や利用者ニーズの変化のスピード

変化の速いネットビジネスなどは、変化を吸収するフェーズ(繰返し)を当初計画に盛り込む

最少 : プロト開発
設計工程で実業務相当の画面を作成し、顧客確認(操作性)
→ 1回(設計+実装)

少 : スパイラル開発、サブシステム単位
設計1回 → 実装2~3回
基本設計1回
→ 2~3回(詳細設計+実装)
※1回目は主要画面中心

多 : アジャイル開発
基本設計1回
→ n回(詳細設計+m回(実装))

最多 : “無謀”

※ 実装 : プログラミング・テスト・結合テスト+顧客確認

Ruby業務システム開発

- ・初期コストが少ない・・・全てオープンソースで構築できる(DBも)
- ・テストが容易である・・・Rails + アジャイル ⇒ 短期繰り返し開発



非定型の業務へ

- ・EXCELを使った統計分析の報告業務・・・担当者の交替により継続性がなくなる
定期的な報告時の基礎データをシステム化し、継続性のある報告(質の向上)
- ・エビデンスのための定期的な記録作成
点検・ヒアリング(装置類・システム・プロジェクト・組織・企業)をシステム化し、
時系列分析や情報の見える化が可能
- ・基幹の業務から外れたもの
トラブル報告、問題点管理、派遣社員の作業時間管理、・・・



非定型業務のシステム化による生産性・業務品質の向上

Rubyで新たな価値の創造を

Ruby × Techno Project

