

# プロトタイプを利用した画面設計と 開発工程への設計データの シームレスな連携

住友電気情報システム株式会社  
フレームワーク技術グループ  
池田 和壽

# 目次

---

- 当社のシステム開発環境
- 現状の問題点
- 課題への対応
- 成果と今後の課題

# 住友電工情報システム株式会社概要

- 設 立 1998年10月1日
- 資本金 4.8億円
  - 住友電気工業株式会社 60%
  - 住友電装株式会社 40%
- 従業員 360名
- 事業内容
  - パッケージソフトウェア(楽々シリーズ)の開発・販売
  - 情報処理システムの開発受託
  - コンピュータ運用業務の受託
  - 情報機器の販売
- URL <http://www.sei-info.co.jp/>



本社 (SORA 新大館 21)

## 現在のシステム開発環境

- 社内システムは全てWeb対応、開発言語にJavaを採用
- プロセス標準化やツール整備で生産性向上

年度	方式	OS	言語	DB
~80	ホスト集中処理	IBM S370 NEC ACOS	COBOL	IMS ADBS
81~90	汎用機分散設置	IBM 4300 NEC ACOS		DB2,DL/I ADBS
91~94	分散処理 (telnet)	UNIX	Informix-4GL	Informix
95~96	C/S		Developer2000	Oracle
97~98		Windows NT	Cold Fusion	
99~	<b>Webシステム</b>	<b>Linux</b>	<b>Java</b>	<b>DB2, Oracle PostgreSQL</b>

# 当社の開発プロセス

- ウォーターフォール型の開発形態

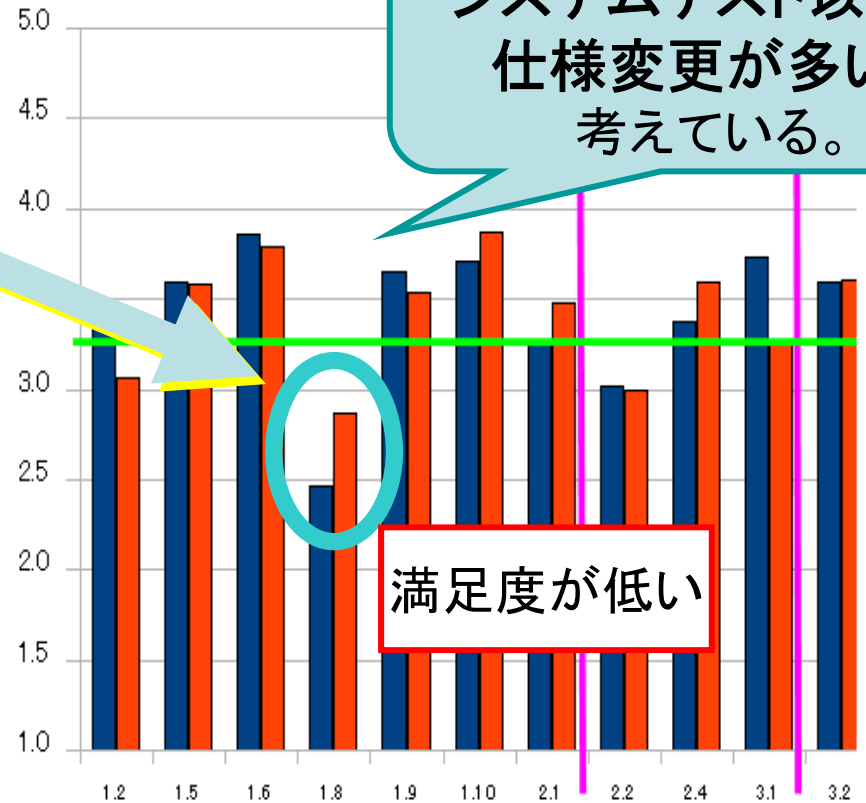


# 現状の問題点

## ■ システムテスト以降の手戻りが多い

2009年2月に実施したユーザー満足度調査結果

No(開設問内容)	利用者	開発者
1.2	3.40	3.06
1.5 操作性はいかがですか	3.59	3.58
1.6 システムの応答速度はいかがですか	3.85	3.78
1.8 システムテスト中および本番稼働後に仕様変更や改善要望は多い(少ない)と考えますか	2.87	2.87
1.9 総合的な評価はいかがですか	3.65	3.54
1.10	3.71	3.87
2.1	3.25	3.48
2.2	3.02	3.00
2.4	3.38	3.59
3.1	3.73	3.26
3.2	3.59	3.61
3.3	3.53	2.93
4.1	3.96	3.92
5.1	4.08	3.44



ユーザーも開発者もシステムテスト以降の仕様変更が多いと考えている。

満足度が低い

参考:セッション4C『サービスサイエンスを活用した外部設計プロセスの定義』

# 手戻りが発生する背景



## ユーザーからの意見:

- ◆ 動作しない画面だけを見ても画面遷移をイメージできない
- ◆ 実際のシステムを使ってみないと操作性が分からない

実システムに近い動作イメージを伝えることが必要

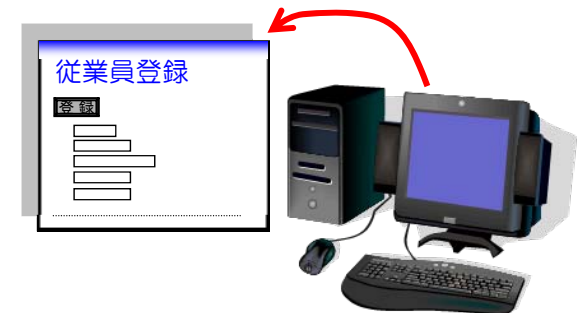
外部設計にプロトタイプの手法を導入

# プロトタイピング設計導入における課題

1. 本番システムと同じ画面を出力できる必要がある
  - ✓ 本番化後をイメージできるレイアウト及び画面遷移
2. プロトタイプ作成による工数増加をできるだけ抑えたい
  - ✓ 効率的にプロトタイプを作成するための手法が必要
3. 現行の構成管理に組み入れる必要がある
  - ✓ 外部仕様書からプログラムまでの一貫した構成管理の仕組みがある

## コーディング工程のツールを外部設計に適用

静的なHTMLを作成する方法では  
実システムと同じ動作にできない





# 課題1. コーディング工程のツール紹介

- 粒度の大きな部品を組み合わせで開発できるフレームワーク
- 画面項目をパラメータファイルで指定するだけで動作する

【例 受注プログラム】

```
jyutyu_no  
jyutyu_mei  
tantosya  
.....
```

【例 出荷指示プログラム】

```
syukasiji_no  
kokyaku_mei  
tantosya  
.....
```

プログラム開発工程の  
生産性3倍(COBOL比)

受注

受注

受注NO

受注品名

担当者

住友 太郎  
住友 二郎  
住友 三郎

出荷指示

出荷指示

出荷指示NO

顧客名

担当者

住友 太郎  
住友 二郎  
住友 三郎

項目定義

項目: *tantosya*  
名称: 担当者  
桁数: 50  
入力: SELECT

# 課題1. コーディング工程のツール適用

- データベースなしでプログラムが動作
  - CSVで作成した本番データを読み込む
  - サンプルデータがなくてもダミーデータを自動表示
- 項目名が頻繁に変更される
  - 日本語の項目名に対応
  - 名寄せによる項目名の置換機能

プロトタイプをユーザーが実際に触って、イメージを確認できるようにした

受注プログラム プロトタイプ

受注	
<input type="button" value="受注"/>	
受注NO	<input type="text"/>
受注品名	<input type="text"/>
担当者	<input type="text"/> ▼
	住友 太郎
	住友 二郎
	住友 三郎

CSV  
ファイル

データベース

# 課題1. 作成したプロトタイプと仕様書の例



受注登録

製品リスト

No.	製品コード	製品名	製品分類	受注数量
1				
2				
3				
4				
5				
合計				



TOP | システム一覧 | 楽々FW技術者人材評価システム 試し本部 テスト実施 | 外部仕様書 | 画面A-1. 分野メンテナンス (登録)

画面A-1. 分野メンテナンス (登録)

ID	Rev	発行者	発行部署	作成者	作成部署	作成日	発行日	更新日
a.1.1.5	Rev.003				フレームワーク技術グループ	2009-08-25	2009-08-25	2009-08-25

a.1.1.5 画面A-1. 分野メンテナンス (登録)

-1. 画面・帳票イメージ (表示切替 | 別Window)

受注登録

検索 全件一覧 新規登録

概要 更新 申請 一覧に戻る

受注

注文番号	A004	氏名	営業 七郎
顧客名	KK情報システム	部門名	システム技術グループ
受付日	2009-01-26	納品日	2009-01-26
受注金額(千円)	89	直送形態	直送
連絡事項	連絡事項 × × × × ×		
注文書添付	見積書添付		

4件

No.	製品コード	製品名	製品分類	受注数量
1	RakFW	楽々 Framework II	閉発ツール	1
2	RakWF	楽々 ワークフロー II	文書管理	4
3	OSPortal	QuickInkution Portal	検索エンジン	1
4	RakFW	楽々 Framework II	閉発ツール	4
合計				10

-2. 入出力データ

入出力	区分	対象データ
OUT	DB	分野

-3. 入出力データ項目

-4. ロジック

(a) 入力チェック

No.	項目名称	MSG No.	チェック内容
1	分野	A104M	既存DBと同じ文字列の「分野」を入力している場合エラーとする。
2	分野	A104M	画面上の他の行の入力欄と同じ文字列の「分野」を入力している場合エラーとする。

(b) データ加工

特になし

(c) DB更新処理 (先行イベントの処理等)

特になし

(d) ソート

特になし

(e) その他

No.	処理内容
1	新規入力行はデフォルト5件とし、10行 -> 20行 -> 50行 -> 100行 -> 200行 -> 500行 と追加できるようにする。

-5. 画面カスタマイズ

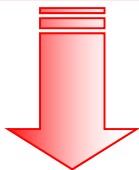
(1) 画面表示

No.	処理内容
1	ボタンは下部にのみ出力する。

## 課題2. 画面設計の工程分析

- 動作するプロトタイプを作成するため工数増加のおそれ

現状の設計作業を分析した結果  
もっとも時間がかかっている  
画面レイアウト作成の工数に着目



コーディング工程ツールの特長である  
画面設計機能を活用して作成工数削減

→画面設計工数を削減することで準備時間の増加を吸収

		時間(分)	
サブプロセス		平均	MAX
1.	仕様説明受け	15	
2.	仕様理解	15	
3.	パターン選択	-	-
3.1.	パターン選定	5	
3.2.	実装可否調査(JavaDoc等)	10	30
4.	画面作成	-	-
4.1.	ISDOC第4レベル作成	10	
4.2.	画面レイアウト作成	120	160
4.3.	画面遷移図作成	15	
4.4.	SE確認後修正	30	
		220	280

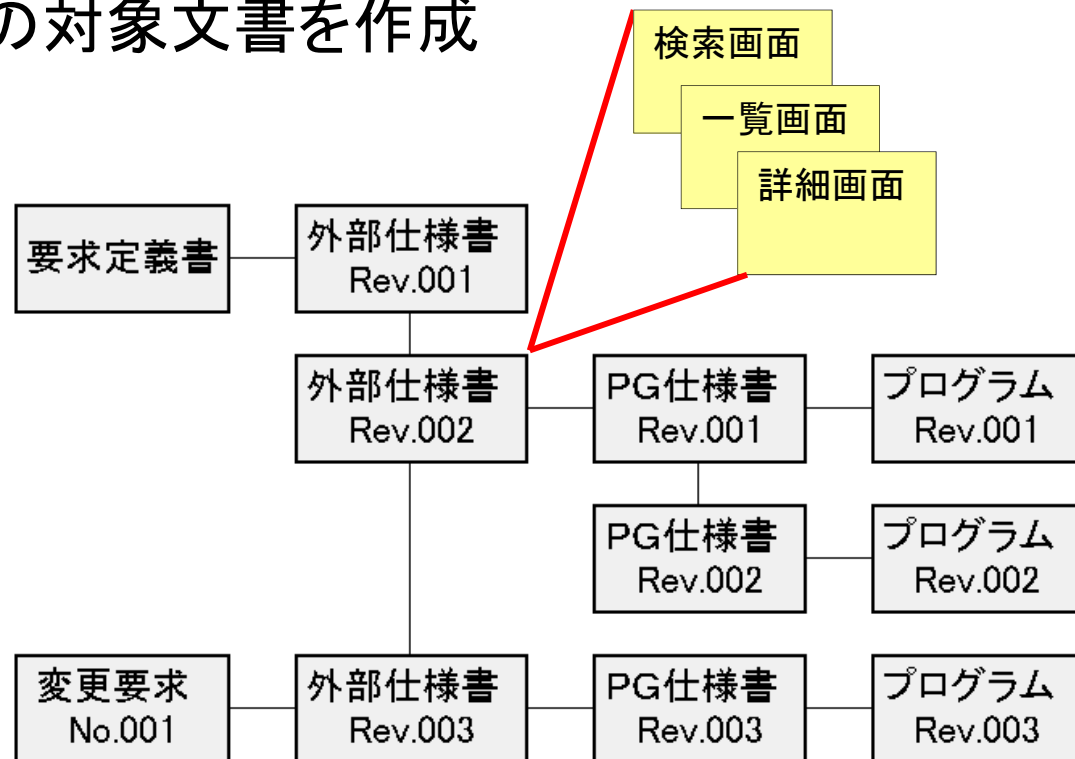
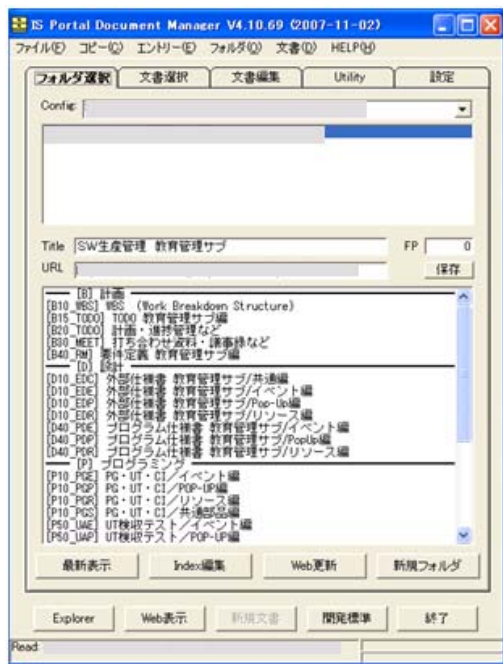
## 課題2. 効率的な画面設計ツール

- 初心者でもマウスを使って簡単に画面設計できる
- ツールを使いながら部品を使い方を覚えられる
- データベース設計途中でも設計情報を取り込んで初期画面生成

The screenshot shows the Rakrak Framework II development environment. The top menu bar includes 'Program', 'Spec', 'XPD', 'Copy', 'Edit', 'Off', 'DD/MD', 'ON', 'OFF', 'Debug', 'ON', 'OFF', 'Lang', '0', '1', '2', 'Develop', 'OFF'. Below the menu, there are tabs for '検索' (Search), '全件一覧' (All list), '新規登録' (New registration), and 'CSV登録' (CSV registration). A red box highlights the '新規登録' tab, with a red arrow pointing to a '受注' (Order) button. A blue box contains the text: '開発モードで実行しカスタマイズしたい部分をクリックすると設定画面がポップアップします。' (Execute in development mode and click the part you want to customize, and the setting screen will pop up). The main window is titled 'データ項目配置' (Data item arrangement) and has tabs for 'Fields一覧設定' (Fields list setting), 'テーブルオプション' (Table options), 'テーブル' (Table), 'SQL', and 'ロジック' (Logic). The '表示部品設定' (Display component setting) window is open, showing fields for 'プログラム名' (Program name) 'salesorder\_EFa', 'プログラム名称' (Program name) '受注(EFa)', 'ページ番号' (Page number) '0', and 'ページ説明' (Page description) '照会の検索画面 [0]'. There are '適用' (Apply) and '画面を指定して適用' (Apply to specified screen) buttons. Below, a list of fields is shown with their values and reference buttons: '受注ID' (0), '注文番号' (XXXXXXXXXX), '従業員番号' (XXXXXXXXXX), '企業ID' (0), '部門ID' (0), '従業員番号' (XXXXXXXXXX), '受注日' (2008-11-26), '納品日' (2008-11-26), '受注金額(千円)' (0), '直送形態' (checkboxes for 直送 and 在庫), and '連絡事項'. A red dashed box highlights the '従業員番号' field, with a red arrow pointing to a yellow starburst containing the text: 'ドラッグ & ドロップによるレイアウト編集' (Layout editing by drag & drop).

## 課題3. 現行の構成管理

- 仕様書とプログラムの関係を維持して版管理
- バージョン不整合検出等のサポートのためツールを自社開発
- 画面単位で版管理の対象文書を作成



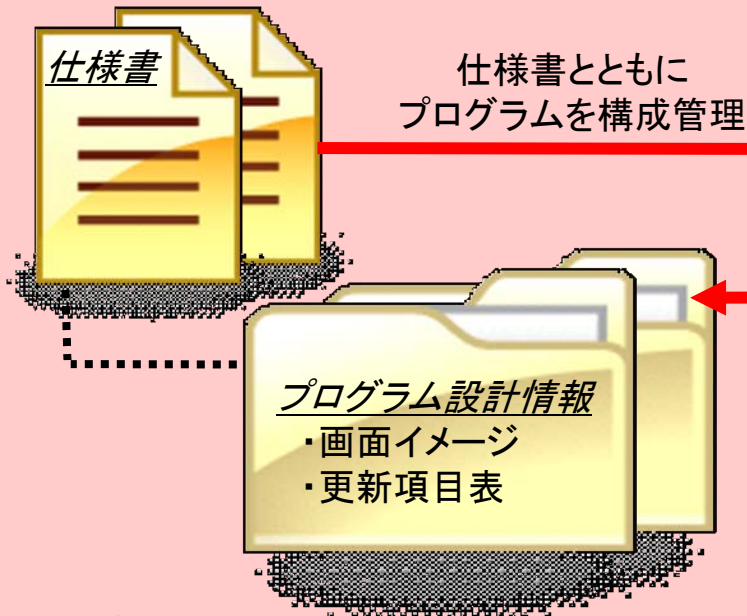
参考: セッション1C 『効率的な測定と構成管理の実践』

## 課題3. プロトタイプの構成管理

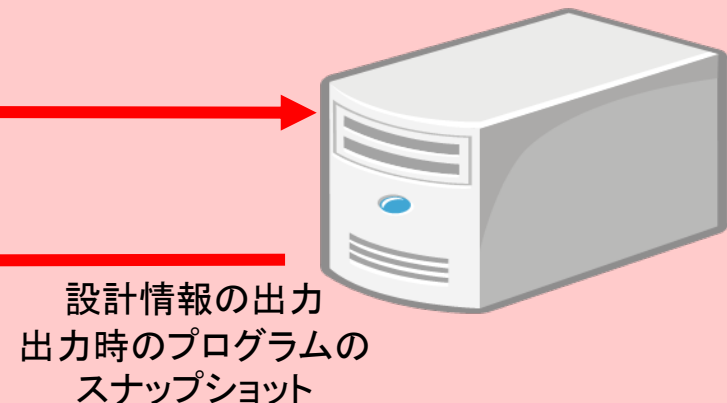
- プログラム本体を仕様書と一緒に保管
- 画面レイアウトを静的に作成して構成管理のフォルダに保存
- 各画面を外部仕様書の適切な文書内に一括保管

コーディング環境を単純に流用すると仕様書とずれが生じる

仕様書 (ファイルサーバ)



開発サーバ



全ての作業をボタン1つで自動化

# パイロットシステムへの適用

- ・画面設計工数平均15%の削減
- ・手直しに時間がかかる場合あり

従来手法		時間(分)	
サブプロセス		平均	MAX
1.	仕様説明受け	15	
2.	仕様理解	15	
3.	パターン選択	-	-
3.1.	パターン選定	5	
3.2.	実装可否調査(JavaDoc等)	10	
4.	画面作成	-	-
4.1	isdoc第4レベル作成	10	30
4.2.	画面レイアウト作成	120	160
4.3.	画面遷移図作成	15	
4.4.	SE確認後修正	30	
		-	-
		220	280

手戻りの実績測定は  
間に合わず...

今回手法		時間(分)	
サブプロセス		平均	MAX
1.	仕様説明受け	15	
2.	仕様理解	15	
3.	パターン選択	-	-
3.1.	パターン選択(画面作成WIZ)	3	
3.2.	実装可否調査(JavaDoc等)	8	
3.3.	パターン修正	10	30
4.	画面作成	-	-
4.1.	項目作成	10	30
4.2.	画面遷移調整	10	
4.3.	画面レイアウト作成	60	120
4.4.	オプション設定	10	
4.5.	プラグインで書くべきボタン等追加	0	40
4.6.	ダミーデータ作成	0	30
4.7.	SE確認後修正	30	
5.	ISDOCへ出力	-	-
5.1.	ISDOC第4レベル作成	10	
5.2.	紐付け機能実行	5	
		186	356



# プロトタイピング設計適用に対する意見

## 1 ユーザーレビューでの理解度が向上

- ❑ 従来の紙芝居よりはるかに理解しやすい
- ❑ 実システムを使っているイメージでレビューできた
- ❑ プロトタイプを見たことによる指摘が出た



## 2 開発者にもメリット

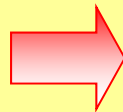
- ❑ 実装可能性の検証ができる
- ❑ コーディング工程のツールを使用するので、実現困難な設計ができない

# プログラム設計と外部設計の連携

- プロトタイプのメタデータを活用
  - プロトタイプに物理設計情報(SQL、物理項目名)を追加
  - コーディング工程への自動変換ツールを開発中

## 現在のプログラム仕様書

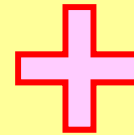
メニュー制御
使用部品一覧
入出力項目、検索方法
更新項目表
エラーチェック
追加ロジック
データ取得方法(SQL等)
部品へのオプション設定



## 仕様書の変更案

メニュー定義
使用部品一覧
入出項目定義
更新項目表
部品へのオプション設定

## 外部仕様書



更新項目表
エラーチェック
追加ロジック
データ取得方法(SQL等)
部品へのオプション設定

## プログラム仕様書

コーディング工程のツールを設計に  
適用したことの副次的な効果で  
プログラム設計工数を削減できる

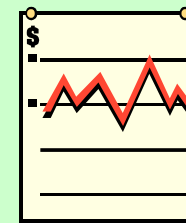
# 成果と今後の課題

## 成果

- ユーザーにプロトタイプを見てもらうことで理解が深まった
- プロトタイプを作成しても設計工数減少

## 今後の課題

- 手戻りと全工程での工数変化の分析
- 成果物の移行機能の開発
- プログラム仕様書の改善と標準化
- プロトタイピング手法の全社展開
- 外販に向けてのパッケージング



ご清聴ありがとうございました