

# Webシステムテスト・ルネッサンス

高品質と短納期の両立へ向けて  
 横河情報システムズ株式会社 林田智也

# アジェンダ

1. 研究背景
2. レビューについて
3. テストについて
4. まとめ・提言



本研究は、ユーザ企業・情報子会社など、様々な業種・立場の人間が集まって  
**富士通(株)のユーザ会であるリーディングエッジシステム研究会(LS研)にて**  
 研究した成果です。  
 まさに、Webシステムのテストにおける、企業情報システム開発の現場による現場  
 のための問題解決策としての集大成です。

# 研究背景

## Webシステム開発の現状

〇月×日までにXXX業務実施のためのシステム  
 化が必要だ。

短納期での依頼

急いで、業務要件を整理・検討を行う。

不具合発生の要因

納期優先で、仕様確認と平行して、開発着手

問題を発見できない

仕様変更も多く、開発が遅れる。

手戻りが多い……  
 不具合を増加

納期遵守のためテストを省略

問題を発見できない

**トラブル発生！！**  
**あの時確認していれば……**

# 研究背景

## 整理すると……



テストが  
 軽視・省略されている



ビジネスの  
 意思決定スピードが向上  
 ⇒「高品質」「低コスト」に加えて、  
 「短納期」であることは当然！

カットオーバー遅延  
 システム稼働後のトラブル発生  
 費用増加

テストを実施しながらも、  
 短納期への希求に応えるにはどうしたら良いか？  
 ⇒テストを見直す(ルネッサンスする)

# 研究の2本柱

## レビュー

設計のテストとして有効なレビュー(手戻りの抑制)

- 作りこまれた不具合の発見
- 業務要件に盛り込まれない要件の注入  
 (例えばパフォーマンス・セキュリティ)
- 効率的なレビュー実施の模索

## テスト

テスト工程の標準化

- 効率の良いテストの模索(必要なタイミングに必要なテスト)
- 「インフラ」「アプリケーション」のカテゴリ分け
- テスト内容の具体的な方法についてノウハウ化

テスト方法の確立



# 研究の2本柱

## レビュー

設計のテストとして有効なレビュー(手戻りの抑制)

- 作りこまれた不具合の発見
- 業務要件に盛り込まれない要件の注入  
 (例えばパフォーマンス・セキュリティ)
- 効率的なレビュー実施の模索

## テスト

テスト工程の標準化

- 効率の良いテストの模索(必要なタイミングに必要なテスト)
- 「インフラ」「アプリケーション」のカテゴリ分け
- テスト内容の具体的な方法についてノウハウ化

テスト方法の確立



# レビューとは？

## レビューの定義

ソフトウェアの品質を高めるために客観的な視点を交えて、開発成果物の記載内容を確認し、改善の項目を探す作業



客観的な視点  
記載内容・改善の  
項目を探す

関係者を交えて  
内容の精査、検討  
確認を行う

## 対象ドキュメント

要求仕様書、外部仕様書、内部仕様書、  
プログラム仕様書、テスト仕様書、ソースコード等



# レビューの課題整理

## 手戻りの発生

要件漏れ  
仕様の欠陥



レビューが有効に  
機能していない！

## レビューの5つの課題

仕様の漏れを抑えきれていない

効果的に行われていない

納期との関係から省略されている

形式化している

仕上げ工程になっている

レビューを有効化 ⇒ 『手戻り』の抑制につながる

# レビューの課題へのアプローチ

## 課題へのアプローチ

- 1.レビューの手法を整理し、重要性を再認識する
- 2.レビューの指針となるノウハウをまとめ、内容の標準化を図る(属人化を抑制する)

- ①業務要件以外の要件漏れを抑制する
- ②Webシステムで重要となる要件を整理する



## 成果物

- 1.レビューの重要性と指針をまとめた「レビューガイドライン」
  - ・標準的なレビュー手法の説明
  - ・場面に応じた効果的なレビュー選択の指針
- 2.開発上流工程レビューの内容を標準化するための「要件チェックシート」「設計チェックシート」
  - ・重要確認が一目瞭然！



# 課題解決のための成果物

## 【レビューガイドライン】

- ・レビュー概要
- ・要件定義レビュー
- ・設計レビュー
- ・トレーサビリティマトリクス

## 代表的なレビュー手法

- ・インスペクション
- ・ピアデスクチェック
- ・チームレビュー
- ・パスアラウンド
- ・ウォークスルー
- ・アドホックレビュー
- ・ペアプログラミング
- etc



レビューの重要性と指針

## 要件チェックシート

要件定義書のレビュー時に使用、個々の製品や技術への依存度・属人的要素をできるだけ排除できるよう作成

## 設計チェックシート

システム開発工程ごとのレビュー時の確認ポイント、タイミング、対象ドキュメントを記載しておりレビューの効率化、品質の向上を目的に作成



具体的解決策

# 要件チェックシート・設計チェックシート

## 要件チェックシート

大分類	中分類	小分類	確認項目	要件	Webで 重要な項目	確認実施者
性能要件	レスポンス	アプリサーバ	画面切り替え時の応答レスポンスタイムを確認する。		○	
		サーバ	サーバへの実装、組合せ、ファイルダウンロード時の許容レスポンスタイムを確認する。			
		処理時間	バッチの処理時間(日次、週次、月次、年次)を確認する。			
		処理開始のタイミング	処理開始のタイミングを確認する。(レスポンスを測定するときの条件)			
	システム負荷	最大同時クライアント接続時におけるレスポンスタイムを確認する。				
	タイムアウト	タイムアウトの設定時間を確認する。				
アクセス	多重アクセス	複数端末からの同一人物アクセスの可否を確認する。				
	同一端末からの複数人物アクセスの可否を確認する。					
	最大ユーザー数	最大利用ユーザー、母体の想定最大利用ユーザー数を確認する。				
	アクセス制限	アクセスコントロール(人数制限)の実施可否ならびに人数を確認する。				

## 設計チェックシート

大分類	中分類	小分類	確認項目	確認結果	IS/人確認	Web 重要な項目	対象 ドキュメント	UI	業務工程	確認工程	備考
セキュリティ	セキュリティ(外部)	画面保持	個人情報の特定および位置に依存しないよう、操作が完了したURLからのCookieを受け入れを設定しないようになっているかを確認する。			○				○	
			Cookieの無効化を考慮し、Cookieの有効期限・有効ドメインを適切に設定しているかを確認する。			○				○	
			Cookieにセッション情報以外の重要な情報を持たせていないかを確認する。(重要な情報は通常サーバ側で管理する)			○				○	
			Cookieを暗号化したい場合、SSL/TLSを使用した通信を行い、Cookieの暗号化を適切に確認する。							○	
			送信内容の暗号化を保持した場合は送信を暗号化しているかを確認する。(暗号化が完了した場合は表示に注意しているかを確認する。(表示しない、あるいはその暗号化で表示する))							○	
			HTMLコメントを残す場合、関係に関するものを残していないかを確認する。			○				○	

# レビューガイドラインの評価

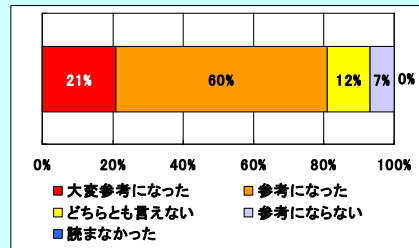
## アンケート

### 実施概要

- 対象企業数:分科会メンバー内の企業6社
- 対象者数:計42名

Q.今後のシステム開発にガイドラインは参考になりましたか？

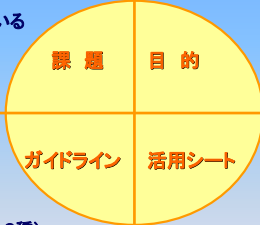
全体で81%の方から「大変参考になった」「参考になった」という高評価！



## レビューの総括

- 仕様の漏れを抑え切れていない
- 効果的におこなわれていない
- 納期との関係から省略されている
- 形式化している
- 成果物の仕上げ工程になっている

- レビューの手法を整理し、重要性を再認識する
- レビューの指針となるノウハウをまとめ、標準化を図る(属人化の抑制)



- レビューの現状と効果、原則(16種)
- レビュー手法(10種)と選定
- レビューの流れと完了条件

- レビューに関するトラブルシューティング
- 要件定義で避けるべきあいまいな用語リスト
- 要件/設計チェックシート
- SLA確認シート
- トレーサビリティマトリクス

-13-

© 2005 リーディングエッジシステム研究会

## 研究の2本柱

### レビュー

- 設計のテストとして有効なレビュー(手戻りの抑制)
  - 作りこまれた不具合の発見
  - 業務要件に盛り込まれない要件の注入(例えばパフォーマンス・セキュリティ)
  - 効率的なレビュー実施の模索

### テスト

- テスト工程の標準化
  - 効率の良いテストの模索(必要なタイミングに必要なテスト)
  - 「インフラ」「アプリケーション」のカテゴリ分け
  - テスト内容の具体的な方法についてノウハウ化

テスト方法の確立

-14-

© 2005 リーディングエッジシステム研究会

## テストの標準化の必要性

### 問題

- やるべきテストがわからない<テストの漏れ>
- 具体的なテスト方法が不明なことが多い。<方法が不明確>
- テスト計画書作成から実施に時間がかかる。<工数の増大>

### 原因

- テストのフローが不明確。
- テストの項目・内容のノウハウの蓄積が少ない。

### 解決策

- テストのフローの定義。
- テスト項目・内容・具体的な方法をあわせてナレッジ化

問題を解決できるものとして

### 成果物

テスト実施ハンドブック

テスト項目抽出ツール

-15-

© 2005 リーディングエッジシステム研究会

## テスト実施ハンドブックの概要

### 課題

- テストをどのタイミングでやれば良いのか分からない
- テストの属人化(経験による差が大きい)
- システム構成要素が多くなり、問題発生時の切り分けが難しい

### 対策

- Webシステムのテスト手順を明確化
- Webシステムのテストをナレッジ化
- インフラテストにも焦点を当てる

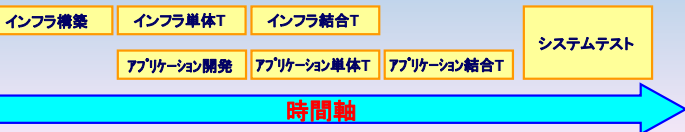
明確な手順によるテストが可能になる!

-16-

© 2005 リーディングエッジシステム研究会

## テスト実施ハンドブック概要

章	項目	主な記載内容
1章	はじめに	テストハンドブックの目的 テストハンドブックの対象者
2章	インフラテストについて	インフラテストの目的と効果 インフラテストの概要 インフラテストの実施方法、完了条件
3章	アプリケーションテストについて	アプリケーションテストの目的と効果 アプリケーションテストの概要 アプリケーションテストの実施方法、完了条件
4章	テスト実施方法	テスト実施の概要 テスト項目抽出ツールの使用方法



-17-

© 2005 リーディングエッジシステム研究会

## テスト項目抽出ツール概要

### 課題

- どんなテストをしたら良いのか皆目検討もつかない場合がある
- テスト実施の際、具体的に何をすれば良いのか分からない
- 簡単に持ち運べて、必要なテストを抽出したい

### 対策

- 具体的な項目、作業例や注意事項を説明
- テスト作業やテスト結果の確認方法を明記
- Excelのみで実現(出先にも持ち込める)

具体的なテスト項目・方法が出力できる!

-18-

© 2005 リーディングエッジシステム研究会

## テスト項目抽出ツール

インフラ単体			共有制・注意事項	作業・確認事項	留意・リスク
大項目	中項目	テスト項目			
インフラ単体	システム構成	OS, DB,  middleware のインストールが正常に完了したか?	どのシステムでも必ず実施する。 OS, DB,  middleware のインストール確認もする。 Solaris→Linux など信頼ツール WebLogic→ネットワークツール(カスタムインストール)が必要なら Oracle, VMware, Network Services,  OS などインストール	インストール自体はインストール済みのイメージを確認する。	OS, DB,  middleware に関する書き換え作業は確認する
インフラ単体	パフォーマンス	アプリケーションサーバの性能は正常か?	アプリケーションサーバの性能は正常か? OS, DB,  middleware までの最大プロセス 数やメモリ使用量を確認し、 アプリケーションサーバの稼働状況 (CPU, ディスク)の稼働状況を確認する。 (マルチスレッドの実行数も確認し、必要に応じて確認)	アプリケーションサーバに対して、 実際の動作環境に近い状態で、 負荷テストを実施し、必要に応じて パフォーマンス改善作業も確認する。	パフォーマンス改善作業も確認する。
インフラ単体	障害通知	SMTPサーバ/アプリケーションサーバに障害があった場合、エラー通知が正しい行われるか?	SMTPサーバ/アプリケーションサーバに障害があった場合、エラー通知が正しい行われるか? システムログでエラー発生を確認しているか確認する。 404エラーなどで結果を確認する。 ファイアウォール設定でエラー発生を監視し、結果を確認する。	SMTPサーバ/アプリケーションサーバに障害があった場合、エラー通知が正しい行われるか確認する。	SMTPサーバ/アプリケーションサーバに障害があった場合、エラー通知が正しい行われるか確認する。
インフラ単体	アクセス権限	テスト用のサーバへのアクセスが正しいか?	テスト用のサーバへのアクセスが正しいか? テスト用のサーバへのアクセスが正しいか確認する。	テスト用のサーバへのアクセスが正しいか確認する。	テスト用のサーバへのアクセスが正しいか確認する。

- ・具体的な参考例やテスト時注意事項の記載
- ・作業方法・確認方法の記載
- ・実施した場合の効果や実施しないリスクなどを明記

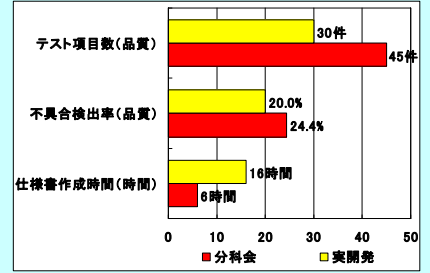
## テスト項目抽出ツールの実践評価

### 実践評価

モデルシステムを使ってテストを実施

- 利用ユーザ数:100人
- 開発規模:2人月程度
- ステップ:3,000ステップ

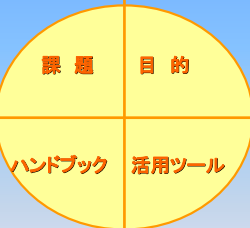
品質を向上させ、  
仕様書作成時間を  
60%削減!!



## テストの総括

- ・やるべきテストがわからない  
<テストの漏れ>
- ・具体的なテスト方法が不明なことが多い  
<方法が不明確>
- ・テスト計画書作成から実施に  
時間がかかる<工数の増大>

- ・テストのフローの定義
- ・テスト項目・内容・具体的な方法を  
あわせてナレッジ化



- ・Webシステムのテスト手順を明確化
- ・Webシステムのテストをナレッジ化
- ・「インフラ」「アプリケーション」の  
カテゴリ分け
- ・具体的な作業例や注意事項を記載
- ・テスト作業やテスト結果の確認方法を明記
- ・Excelのみで利用可能

## まとめ

### 問題意識

Webシステムで「短納期」「高品質」の両立

### 研究の目的

- 1.レビューの重要性の再認識・標準化
- 2.テストの項目・手順・方法の標準化、ナレッジ化

### 研究成果物

- レビューガイドライン/活用シート(仕様の不備が防げる)
- テスト実施ハンドブック(テスト工程の整理ができる)
- テスト項目抽出ツール(質の良いテスト仕様書が作れる)



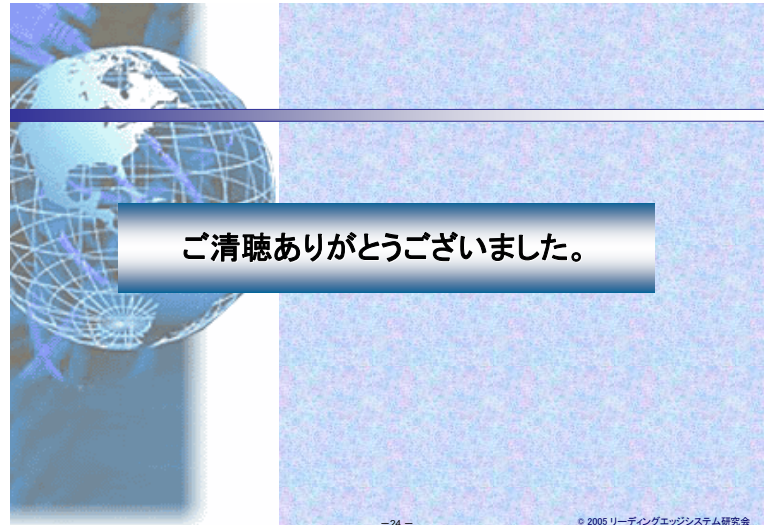
## 提言

### メッセージ

- レビュー内容を拡充・標準化し、隙のないレビューで「手戻り」を防止しよう
- テストをナレッジ化・標準化してミスをなくし、効率化しよう



レビューとテストを見直し(ルネッサンスして)  
「短納期と高品質の両立」を実現しよう



ご清聴ありがとうございました。