
Java 開発における生産性向上策

(株) メイテツコム

■ 執筆者 Profile ■



1995年 入社
入社以来、放送局 WEB ベースのシステム開発に従事

2011年 現在
社会情報ソリューション部第1担当所属
自治体向け WEB システム開発に従事



加藤 雅文

1987年 入社
入社以来、メインフレーム中心のオンラインシステム開発に従事

2011年 現在
社会情報ソリューション部第2担当所属
交通系ICカード開発に従事

■ 論文要旨 ■

国産 Java フレームワークである「Seasar2」とテストツール「Selenium」を利用して品質を保ちながら短期間でプロジェクトを遂行する手法を、筆者が平成 22 年度に開発した 2 つの WEB システムの開発経験に基づき記述する。

■ 論文目次 ■

1. はじめに	《 3》
1. 1 当社の概要	
1. 2 本論文の趣旨	
1. 3 開発したシステムの概要	
2. Seasar2 フレームワークの活用	《 3》
2. 1 Seasar2 フレームワークとは	
2. 2 Seasar2フレームワークのソフトウェア構成	
2. 3 Seasar2を用いた開発手順	
2. 4 Seasar2フレームワーク利用時の問題点と対応	
3. Selenium の活用	《 9》
3. 1 Selenium とは	
3. 2 Selenium のテスト手順	
4. 今回のシステム開発の結果	《 11》
4. 1 Seasar2 による開発工数の削減効果	
4. 2 Selenium を利用して得られたメリット	
4. 3 開発の総括	
5. おわりに	《 12》

■ 図表一覧 ■

図 1 システムの共通した処理パターン	《 3》
図 2 プログラム開発の経過日数と進捗の関係	《 4》
図 3 Seasar2フレームワークを活用したWEBシステムのソフトウェア構成	《 5》
図 4 Seasar2で開発するプログラムの一例	《 6》
図 5 Seasar2フレームワークの開発手順	《 7》
図 6 Seleniumの操作画面	《 9》
図 7 Seleniumを使用したテスト手順	《 10》
表 1 2つのプロジェクトの画面実装の生産性	《 11》

1. はじめに

1. 1 当社の概要

当社は名古屋鉄道グループの情報システム子会社として 1976 年に設立した。当社は、交通・物流・流通・文教・公共・電子商取引・旅行業と幅広い業種を取り扱い、IT 戦略の企画から、システム開発、保守・運用まで IT の全領域をワンストップで提供する。

1. 2 本論文の趣旨

本論文は、国産 Java フレームワークである「Seasar2」とテストツール「Selenium」を利用して品質を保ちながら短期間でプロジェクトを遂行する手法を、筆者が平成 22 年度に開発した 2 つの WEB システムの開発経験に基づき記述する。

1. 3 開発したシステムの概要

筆者は平成 22 年に 2 つのシステム開発を行った。システム要件を確認したところ、システムの画面数はプロジェクト開始時点で 2 つのシステムいずれもが 50 画面と予測し、さらには XML の生成と SOAP 通信といった内部処理も必要であった。

この 2 つのシステムの開発期間は非常に短く、1 つ目のシステムは開発から稼動まで 4 ヶ月、2 つ目のシステムは稼動まで 3 ヶ月という非常に短期で開発しなければならなかった。

このような厳しい条件ではあるが、2 つのシステムの根幹は、業務内容の違いはあるが図 1 に示すように、①WEB ブラウザによる入力、②データベースへの保存、③XML を他システムへの SOAP 通信の、3 つの機能をもつシステムであり、この 3 つの機能の開発手順を標準化もしくは共通化できれば、品質を維持しながら納期内の開発が可能だと考えた。

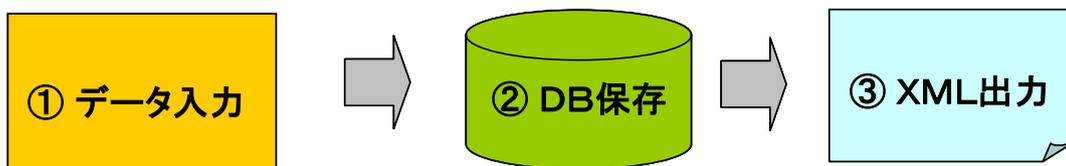


図 1 システムの共通した処理パターン

具体的な開発にあたっては、XML 出力機能は、過去に弊社で開発した XML 生成モジュールおよび SOAP 通信モジュールを活用して対応できると考えた。データ入力機能とデータベース保存機能は、新規に作成が必要であった。新規開発にあたっては、フレームワークとテストツールを活用することとした。

2. Seasar2 フレームワークの活用

2. 1 Seasar2 フレームワークとは

Seasar2 は、非営利団体「Seasar ファウンデーション」が提供する Java によるアプリケーションフレームワークである。フレームワークを利用すると、開発者はフレームワークが規定する開発手法に従って開発するため、開発者によるブレが少なく品質の高いプログラムを開発できる。

さらに、Seasar2 は日本で作成されたフレームワークでありインターネットや書籍等、日本語の情報が豊富にあり、開発者の負担を軽減できる。某メガバンクの大規模なシステムに対しても適用されかつ評価が高い。

筆者が所属する部署では、従来は Struts を使って開発していた。Struts で開発する場合、開発当初は過去の経験を活かして開発が進むが、その後のテストに工数がかかり、本番稼働まで開発が続くと予測した。

本稼働に間に合わせるため他のシステム開発プロジェクトの開発手法を探したところ、Seasar2 を短期間で習得し開発した実績が見つかった。この実績を基に、今回のシステムの工数を見積もったところ、納期前の試験期間を十分確保できると考えた(図2)。

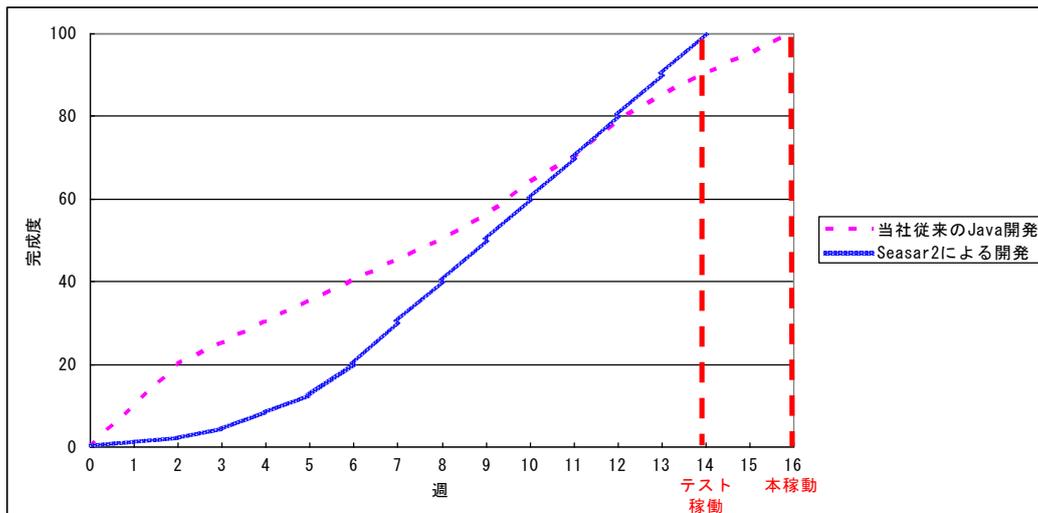


図 2 プログラム開発の経過日数と進捗の関係

2. 3 Seasar2 フレームワークのソフトウェア構成

図 3 に、今回作成した WEB システムのソフトウェア構成を示す。

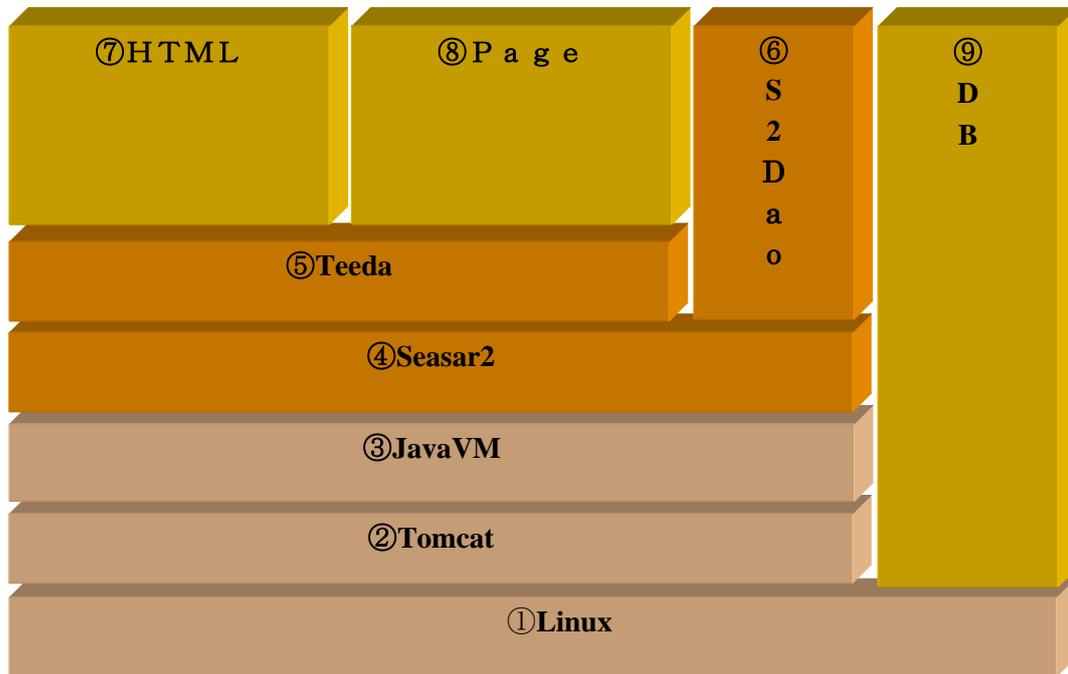


図 3 Seasar2 フレームワークを活用した WEB システムのソフトウェア構成

今回利用した Java フレームワークは Seasar2, Teeda, S2Dao で構成されている。いずれも「Seasar ファウンデーション¹」が提供するフレームワークであり、機能を簡易に連携できる。Seasar2, Teeda, S2Dao については後述する。

開発した Java プログラムはサーブレットコンテナ Tomcat で管理する。WEB サービスの利用者は、Tomcat 上に配置した Java アプリケーションを WEB サーバ (Apache) を介して呼び出す。Java プログラムは JavaVM で実行される。

今回のデータベースは PostgreSQL を用いたが JDBC 接続できるデータベースならば変更が可能である。

これらを実行する OS は今回 Linux で構築したが、Microsoft Windows Server でも同様の構成となる。

続いて Java フレームワークを構成する Seasar2, Teeda, S2Dao について説明する。

Seasar2 は、プログラムを連携するための基本的なフレームワークである。Seasar2 は、データベースに対するトランザクションの制御、コネクションプーリングといった機能を提供する。

Teeda は、プレゼンテーション層を開発するフレームワークである。HTML のタグ要素に付けた id と、Java プログラムの内部変数との橋渡しをする。この他、入力値の型変換や妥当性検証、リンクやボタンの処理、ページ間をまたぐ入力値の保持といった機能がある。

¹ 参考 URL <http://www.seasar.org/>

S2Dao は、データベースの項目と Java プログラムの内部変数とを関連付ける。従来のデータベースでは、項目定義ファイルを XML ファイルに記述しなければならなかったが、S2Dao では Java プログラムソースに項目と型を定義するのみでデータベースの項目と連携できる。また、SQL 文を外部ファイルに分離できる特徴があり、プログラムの可読性向上や開発時の分業が可能である。

これらフレームワークの特徴は、いずれも AOP (Aspect Oriented Programming : アスペクト指向プログラム) および DI (Dependency Injection : 依存性注入) へ対応していることである。

AOP とは、例えばトレースログの出力処理、トランザクション処理、セッション処理などシステム全体で共通利用する処理とその定義を、本来のプログラムを変更することなく追加するプログラム手法である。プログラムの可読性の向上はもちろん、コーディングミスを防ぐことができる。

DI とは、プログラムで利用するクラス間の依存関係を取り除き、プログラムの実行時に依存関係を持たせる仕組みである。具体的には、プログラムのクラスを置き換えるときには XML ファイルに利用するクラスを記入するだけでよく、旧来行われていたクラスを利用するプログラムの修正や実行モジュールの構築などの作業を省略でき、開発効率を向上させることができる。

図 4 に、ログイン画面を Seasar2 で開発する場合に作成するファイルの一例を示す。

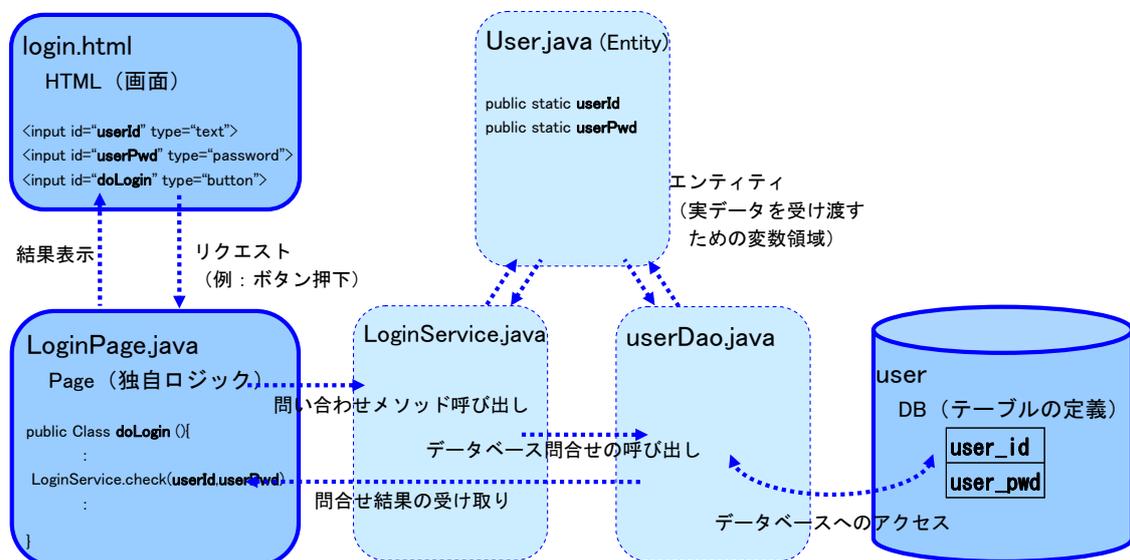


図 4 Seasar2 で開発するプログラムの一例

プログラマは、図 4 の実線で示した HTML (画面)、Page (独自ロジック)、DB (テーブルの定義) を作成する。Seasar2 は、その他の点線で示した通常必要な機能や処理プログラムを自動的に生成する。自動的に作成されたプログラムでは処理が不足する場合は、作成されたクラス内に新たな処理を加えることになるが、新たな処理は自動生成されたプログラムを真似して作ることができる。

これら自動的に作成されるプログラムは、フレームワークで品質が確保されており、単純なプログラムミスはもちろん開発工数・テスト工数が削減できる。

2. 4 Seasar2 を用いた開発手順

Seasar2, Teeda, S2Dao フレームワークを用いた開発手順を図 5 に示す。

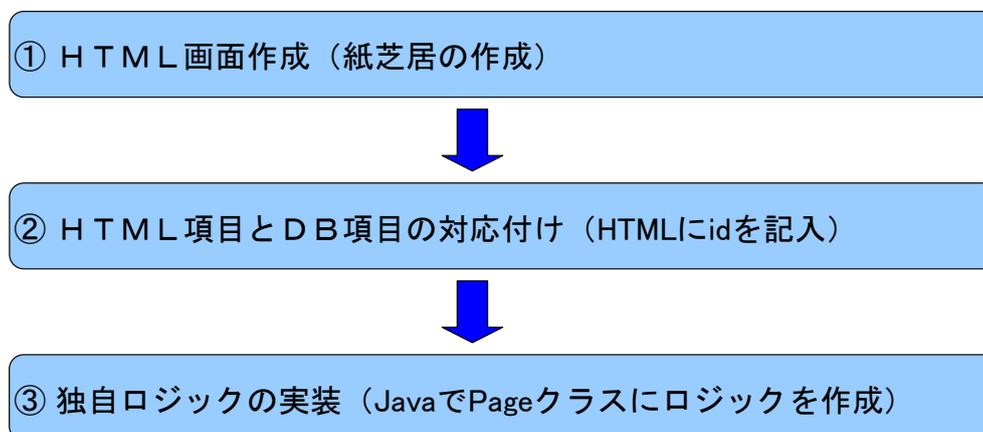


図 5 Seasar2 フレームワークの開発手順

① HTML 画面作成

Seasar2 における開発では、最初に HTML でモックアップ（紙芝居）を作成し、ユーザーレビューをしながら仕様を固める。ユーザーもシステムのイメージがつかみやすく、ユーザーと開発者の仕様のずれによる後戻りのリスクが軽減できる。

さらに、作成した HTML を利用して開発工数を削減できることはもちろん、作った HTML を動かしながら外部設計書・内部設計書を書き起こすことができるため、ウォーターフォール型の開発と比較して設計書の作成・レビュー・修正といった工数が削減できるメリットがある。

② HTML 項目と DB 項目の対応付け

HTML モックアップの作成後、入出力項目を抽出しデータベースのテーブル設計をする。このテーブル設計は、Seasar2 とは関係なく通常の開発と同等の作業である。

開発者は完成したテーブル設計に基づき、Java プログラムでデータを格納する入れ物に相当する Entity を定義する。その後、Dao (Data Access Object : Java プログラムがデータベーステーブルにアクセスするためのクラス) を定義する。データベースを利用するための基本的なメソッドは Seasar2 の機能によりほぼ自動で生成される。Entity や Dao の定義クラスはスペルミスを防ぐため、すでに作成したテーブル設計を元に自動生成するとよい。

次に、HTML のタグに Seasar2 の規約に従って id 属性を記述する。例えば、HTML 項目と DB 項目の対応付けでは、HTML の入出力項目の id 属性の値にデータベースのカラム名をキャメル記法で記述する。例えば、データベースのテーブルカラム名が emp_name ならば empName と記述する。

③ 独自ロジックの実装

画面から情報を受けて、サーバ上で処理する独自ロジックは Page クラスに記述する。Seasar2 は基本的な Page クラスを自動で生成し、HTML や Entity 等とのインターフェースとなるプロパティも記述される。

Seasar2 では、Page クラスやプログラムを記述すべきクラスの階層も規定されており、結果としてプログラムの可読性が向上し、また、自動処理される部分についてはコーディング工数も大幅に削減できる。

独自ロジックを実装する際には、入力フォームを処理する場合は do+画面名、別の画面に遷移する場合は go+次画面名といった簡単な Seasar2 の規約に従って HTML と Page クラスを記述する。

このように、Seasar2 のフレームワークを活用することにより、例えばデータベース・変数・画面などにデータを代入する処理を作成するときには発生しがちな、多くのコーディングミスやスペルミスを防ぐことができ、一定以上の品質を確保したプログラムを作成することができる。また、コーディングが減ることにより、単体テスト工数も削減できるため、開発工数も削減できる。

2. 5 Seasar2 フレームワーク利用時の問題点と対応

初めて Seasar2 を利用した筆者が、調査に時間を要した問題点を 3 つ挙げる。

最も困った問題は、本番環境での動作確認テストの際、WEB アプリの動作中にメモリリークが原因でシステムが動作不能に陥ったことである。

Seasar2 の初期設定では、アプリケーションの開発とデバッグを効率的に行うため、プログラムに関係するソースを実行する都度、自動的にメモリにプログラムを再読み込みする設定となっている。開発期間中はプログラムを停止することが多く、プログラムの実行に伴うメモリ消費に気がつかなかった。

この設定のまま本番環境へ移行した結果、プログラムの実行に伴ってメモリ消費量が増加し、Java に割り当てたメモリを全て使い切ってしまう現象が頻発した。

この問題は、設定ファイルを変更しメモリへのプログラムの読み込みを起動時のみとすることで解決した。

2 つ目の問題は、エラーメッセージの制御である。初期設定のエラーメッセージは、どの処理で問題があったかを画面に表示するが、このままインターネットに公開するとセキュリティホールとなるため WEB 画面からは隠匿する必要がある。今回の開発中に、Session タイムアウト時の処理が、他のエラー処理と同じ処理では対応できないことがわかった。この問題は、必要なセッション処理をする新たなクラス (SessionFilter クラス) を作成し、設定ファイルへこのクラスを追加することで解決した。

3 つ目の問題は、セッションの引き継ぎである。当初はプレゼンテーション層を処理する Page クラスの親クラス (AbstractPage クラス) を作成して、セッションをクラスの属性として管理できると考えていたが、うまく機能しなかった。この問題に対しては、セッ

セッション情報を扱うクラスを新たに定義してセッションを引き継ぐことで解決した。

このように、いくつか問題が発生したが、日本で作られたフレームワーク Seasar2 の情報はインターネットや書籍等に日本語で豊富に存在し、致命的な問題となることなく解決することができた。

3. Selenium の活用

3.1 Selenium とは

「Selenium²」は WEB アプリケーションのテストを自動化する GUI ツールである（図 6）、

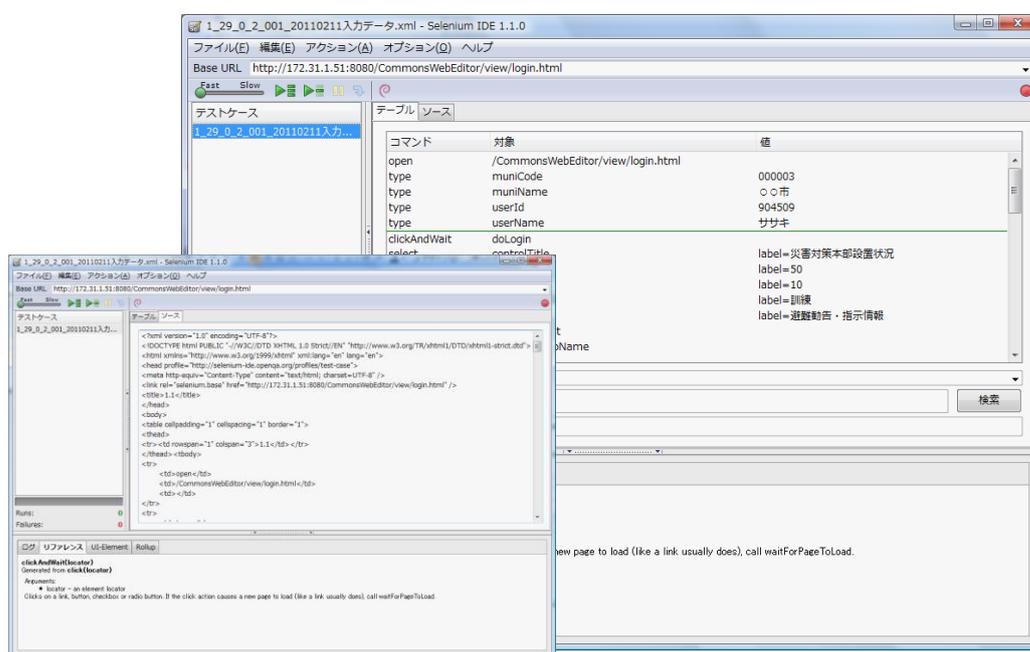


図 6 Selenium の操作画面

Selenium は、ユーザーの画面操作をスクリプト化して記録する機能と、記録した内容を再生する機能を持っている。

Selenium を用いると、記録されたスクリプトと、テストパターンを網羅したテストデータを用意することで、WEB 画面のテストを自動化できる。例えば、不具合修正後の再テストや、画面操作が同じで入力パターンが異なるテスト、Internet Explorer や Firefox 等の複数種類のブラウザで動作確認テストを行う場合など、テスト工数の削減に役立つ。

² 参考 URL <http://seleniumhq.org/>

3. 2 Selenium のテスト手順

Selenium のテスト手順を図 7 に示す。



図 7 Selenium を使用したテスト手順

① 画面操作の記録

Selenium の記録開始ボタンをクリックしてから、WEB 画面でデータ入力などの操作をすると、自動的に HTML 形式のスクリプトとして操作が記録される。

② スクリプト編集

Selenium が記録したスクリプトをテキストエディタで編集する。例えば、入力値の変更や繰り返し入力のスクリプトを編集する。あるいは、網羅的なテストパターンを作成する場合は、すでに作成したスクリプトを複製して値だけを書き換えることで作成できる。類似画面をテストする場合は、すでに作成したスクリプトの一部あるいは全てを複製し必要な部分を書き換えることで、操作の記録スクリプトを作成する時間も短縮できる。

さらに効率的にテストパターンを作る場合は、表計算ソフトなどで CSV ファイル等の外部ファイルを作成し、Selenium へ取り込むことになる。

ユーザーインタフェースに変更が無い、同じ操作を繰り返すテストや内部ロジックの変更後の再テストでは、スクリプトを編集することなくそのまま再利用できる。

③ スクリプト実行と確認

記録もしくは編集したスクリプトを実行する。

実行すると、WEB 画面が自動遷移し、ブラウザで結果を目視確認できる。実行速度を調整する機能もあり、画面の切替えが速過ぎて目視確認できなくなるようなことはない。

4. 今回のシステム開発の結果

4. 1 Seasar2 による開発工数の削減効果

プロジェクトを遂行すると、開始当初は Seasar2 の習熟とプロジェクト内でのルール決めに時間を要したが、概ね 2 週間で Seasar2 の習熟とプロジェクト内ルールが修練されていった。表 1 に、先に開発したシステム（システム 1）と、後から開発したシステム（シ

システム2)の画面実装の生産性を比較してみた。

表 1 2つのプロジェクトの画面実装の生産性

	画面数 (サブ画面含む)	開発工数 (人日)	1画面あたりの 開発工数(人日)
システム1	45	312	6.9
システム2	48	150	3.1

画面機能が違うため単純に比較できない部分もあるが、先行したプロジェクトと比べて Seasar2 の扱いに慣れた後発のプロジェクトでは、画面当たりの生産性が約 2.2 倍向上している。

このことは、Seasar2 が習熟しやすいフレームワークであること、また、習熟後は生産性が飛躍的に向上することを表していると考えている。

4. 2 Selenium を利用して得られたメリット

Selenium を使って、本論文で述べた2つのシステム開発では、3つのメリットがあった。

まず、Selenium が記録したスクリプトを活用することにより、WEB アプリケーションをテストするための画面操作時間を削減できたことである。特に入力項目の一部だけを書き換えて網羅的にテストを実施する場合は、事前にテストデータやスクリプトが準備でき、テスト時間を大幅に削減できた。

続いて、エビデンスとなる画面キャプチャファイルを自動で保存でき、エビデンス管理に要する時間を省くことができたことである。

最後に、クロスプラットフォームの試験が可能であったことである。今回は Internet Explorer の複数のバージョンと FireFox 等のマイクロソフト製以外の WEB ブラウザーの動作検証も必要であったが、Selenium はこれらブラウザのテストに対応しており、作成したスクリプトを使って自動的な試験ができた。この点でも、従来の手作業によるテストと比較して、工数が削減できたことである。

4. 3 開発の総括

本論文で述べた2つのシステム開発は、フレームワークやツールを利用してコーディング量を少なくし、質の高いプログラム開発ができた。

フレームワークを利用して品質が確保されたプログラムは、短期間開発の場合にありがちな単純ミスによる不具合の発生もなく、テスト工程も予定通り終了することができた。

さらに、テストそのものもテストツールを利用して自動化することにより、不具合発生時や新機能の追加後の再テストも効率よく行うことが出来た。

5. おわりに

本論文のシステム開発は予定通り納期を守ることができたが、システムの開発後に Seasar2 や Selenium の活用事例を調査したところ、開発に役立つ機能や事例が見つかった。

Seasar2 フレームワークを使った開発では、Ajax 対応のフレームワーク Teeda Ajax の利用を今後図りたい。

今回のアプリケーションでは、複数の項目を選択をするために同じ画面を何度も再描画している画面があり、描画時間がかかっている。この画面で選択項目がさらに多くなると、利用者は選択操作をする前に画面の再描画後に画面をスクロールして選択項目を表示しなければならず操作性が悪い。このような画面では、Ajax を活用して描画速度と画面操作性を向上するが、今回のシステム開発は初めて Seasar2 を利用することもあり、Ajax を利用する設計ができなかった。

今後は WEB アプリケーションの利用者に対するレスポンスの向上と操作性の向上のため、Teeda Ajax を活用したいと考えている。

一方、テストツール Selenium は、テスト結果が意図した通りか確認するアサーションチェック機能の利用がある。このほか、テストケースをエクセルで作り連動する工夫ができ、テストシナリオの編集が便利にできることがわかった。

これらの機能は、品質向上のためのテスト工数をさらに削減すると考えられ、是非とも利用したいと考えている。

今後のシステム開発に有用だと思い、今後の課題として記す。