

WebTech の開発

— “提案書の見える化” —

1. 背景

IT 業界では、“デスマーチ”なる言葉が日常的に使われ、納期遅延、機能削減、深夜残業などが日常化している。その原因の一つが上流工程にあることは、従来から指摘され続けているが、いまだ抜本的な解決策のない状態が続いている。この問題の解決のために“提案書の見える化”が必要だと考えている。

2. 目的

システム開発における要件定義書は、IT の専門知識がなければ検証が難しい。しかし、ここに大きな問題がある。なぜなら、完成したシステムを使うのは IT の専門知識を持たない現場の人たちだからだ。完成間近になってから現場の人たちに問題を指摘されては、納期遅延などは免れない。開発の初期の段階から現場の人たちと一緒に進めることが必要なのである。現在でも、簡単な画面デザインを用意して、打ち合わせに活用しているケースは多い。しかし、モデルから独立して作成された画面デザインでは、論理的な整合性や実現可能性を確認できず、その後の開発がうまくいかない場合が多い。モデルの精度を向上させながら、モデルと矛盾しないプロトタイプでコミュニケーションを取らなければ、事態の解決は難しい。WebTech はこれを実現するものである。モデル精度の向上と、円滑なコミュニケーションの両方を実現することで、その後の順調な開発を目指すものである。

3. 開発の内容

3-1. WebTech の全容

- 開発方法論 - Typical first -
- アプリケーション - Ryoma -
- フレームワーク - Mutsu -

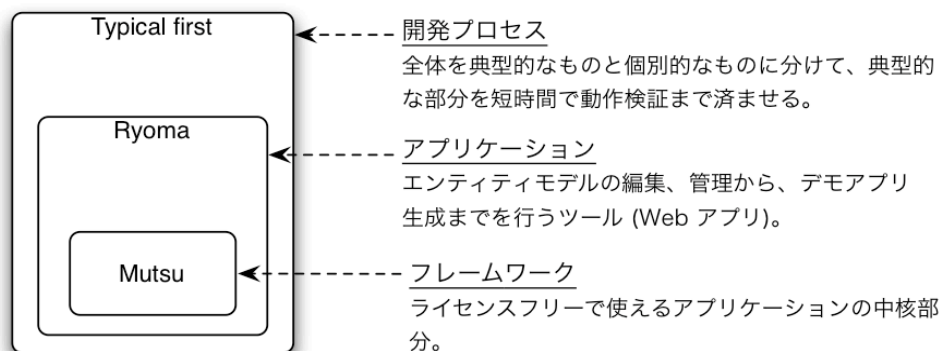


図 1 : WebTech 全容

3-2. 開発方法論 - Typical first -

機能をサブシステムに分割して開発することは一般に行われている。それとは別に、新たに典型的な部分と非典型部分とに分ける - Typical fist - を提唱する。これは、自動化可能な典型部分を短時間で先に片付け、難易度の高い非典型部分に十分な期間を割り当てるものである。

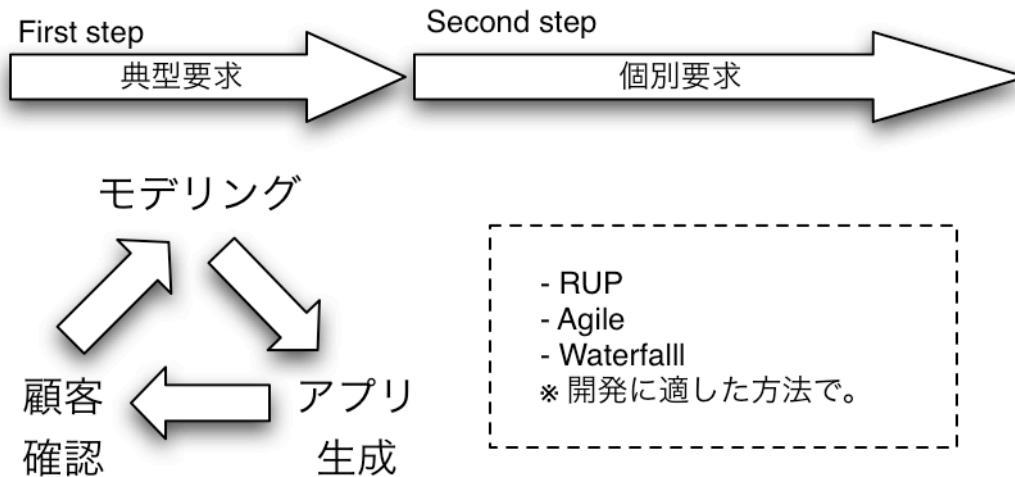


図2 : Typical first

簡単なことと難しいことを区別すること無く、すべて同じ開発プロセスを経て開発することは、非効率的だと言える。Typical first は、難易度に応じて適切な開発方法論を適用する“ベストエフォート開発”を提案するものでもある。

3-3. アプリケーション - Ryoma -

Typical first を実現するツール Ryoma は、エンティティモデルを入力し、モデルと矛盾しないプロトタイプを生成する。プロトタイプ生成の前に、テーブル設計や画面設計を作成するが、Ryoma が自動生成するため、従来の作業は大幅に軽減される。

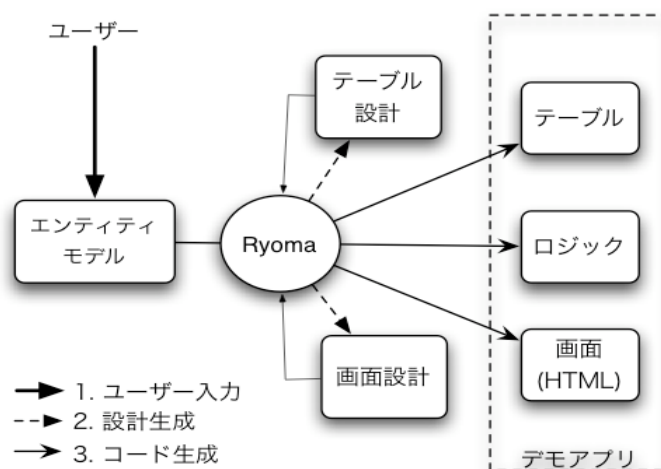


図3 : Ryoma I/O

Ryoma の画面は、およそ下図の通り。左が編集画面、右が ERD である。



図 4 : Ryoma 編集画面

編集画面でモデルを入力の後、Ryoma にコード生成を指示すると、下図のようなプロトタイプが生成され、参照や入力を実際に試すことができるようになる。



図 5 : プロトタイプ画面

図は、プロトタイプを生成した後で、スタイルシートを加工したものであるが、

Ryoma が生成するプロトタイプは、図のようなデータの入力フォーム、複数レコードの一覧表示、レコードの詳細表示、ツールバーなどの部品を組み合わせたものである。

3-4. フレームワーク - Mutsu -

Ryoma が生成するプログラムコードはフレームワーク 'Mutsu' に対応するものである。Mutsu は、エンティティのキャッシュや、エンティティ関連の自動トレースを組み込んだ、独自のフレームワークである。MIT ライセンスで配布するため、Mutsu に依存するアプリケーションでも、Mutsu のライセンスに無関係に独自のライセンスを設定することができる。

4. 従来の技術(または機能)との相違

UML を初めとするモデル表記を読むためには専門知識が欠かせない。しかし、モデルに基づいてプロトタイプを生成する Ryoma では、実際にプロトタイプを使ってみることで検証できるので、特別な専門知識を持たない人たちとコミュニケーションを図りながら、モデルの精度を高めてゆくことができる。モデル、画面、永続化が連動しているツールは、Ryoma をおいて他に見当たらない。

5. 期待される効果

従来であれば、開発途中で価値に疑問があるとされても、無理にシステム開発に繋げるしかなかった。そこまでの開発で膨大なコストがかかっていたために、後戻りが難しいからだ。Ryoma なら安価にプロトタイプが作ることができる。そのため、プロトタイプを試した後で、「捨てる」ことができるようになる。さらに、複数のプロトタイプを試した上で、望ましいものを選ぶといった開発手法をとることもできるようになるため、プロトタイプ作成がビジネスとして活性化すると考えている。

また、Ryoma は、「設計に基づいてどこまで自動生成できるか?」の実証的側面も持っている。Ryoma は、Ryoma 自身によって作成されているが、自動生成部分に追加したコードは全体の 20% に過ぎない。一般的な要件であれば、プログラムコードの 80% は機械的に生成することができると言える。今後のシステムの開発のありかたを変えてゆくものと確信している。

6. 普及(または活用)の見通し

システム開発の要件定義に関わるコンサルタントや開発者に使っていただくことを想定したのだが、システム開発を依頼する利用側が、試行錯誤の上で厳密な RFP を作成するのに活用されることを期待している。広く活用されるために、オープンソース化とコンサルティングサービスを提供することで、普及に努めたい。

7. 開発者名(所属)

徳富 優一 (フリーランス)

(参考)開発者URL <http://octopussoft.dip.jp/>