

# シンプル・便利・基本的! フリー・ソフトで高品質プログラミング

ご購入はこちら 中村 晋一郎

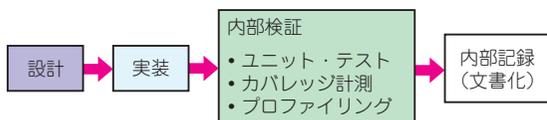


図1 ちゃんと品質を担保しながらプログラムを作成する基本フロー

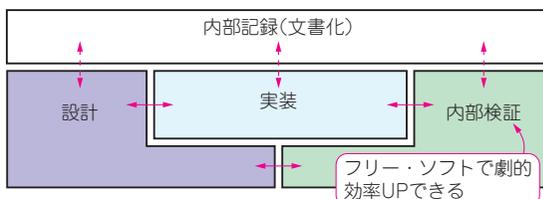


図2 設計・実装・検証を繰り返すことで品質を高めていく

## 品質のよいプログラムを作るための必殺の「型」

どんな仕事にも必ず仕事を全うするための骨組み(型)が存在します。ここでは、ソフトウェアの品質を担保しながら開発するための型となる「開発の流れ」を説明します。図1に設計の流れを、図2に項目の関係を示します。

### ● ステップ1：設計

どんなソフトウェアであっても設計は必要です。システムに求められる要求に対して、一体何を実現するのか?をはじめに整理し、そこからどのように実現するのか具体的な解決方法に落とし込んでいきます。

全体的な戦略から細部詳細設計まで設計の粒度はさまざまです。この設計段階でポイントとなるのが、対象システムの物理的および論理的な構成要素を整理しておくことです。設計段階で物理的、論理的な構成要素を整理できていなければ、その結果として実装、検証、記録(文書化)と後々の段階でさまざまな弊害が出ます。構成要素を整理しておくことで、全体設計で必要な構成を確認できる上、開発の全工程で矛盾なくシステムを実現できます。ここで物理的、論理的な構成要素を整理し、設計・実装しておけば、将来の改良や拡張などにも対応しやすくなります。

### ● ステップ2：実装

設計が完了したら実装に移ります。どの部分を実装しているのか常に自分で把握するために設計書を手元で見ながら実装します。もし設計どおりに実装を

行っていく、実装段階で気づいた新たな課題や問題があれば、設計へのフィードバックも視野に入れます。

理想的な世界では、設計に何ら問題もなく、スムーズに実装まで完了しますが、実際にそのように進まないこともあります。その場合、臆することなく設計の見直しを行わなければなりません。後回しにして問題を放置すると、ユーザの手に渡ってから問題に発展する可能性もあります。よって、実装であっても常に対象システムに対する考察を加えながら行わなければなりません。単なるタイピングの作業ではないという認識が非常に重要です。

### ▶ 品質UPのコツ

また、この段階で気づいたことや注意すべきポイントは、個条書きなどにして記録にとどめておくでしょう。プロジェクトでRedmineやMantis BTやTracなど課題管理システムを使っている場合には、個別にチケットを発行しておく、後から検証項目洗い出しやユニット・テストの設計実装時の実作業とも結びつけることができお得です。

### ● ステップ3：内部検証

設計、実装に並んで非常に重要なのが、内部検証(プログラム自体の検証)です。システム内部で行うべき検証という意味で、ここではわざわざ「内部」を付けています。内部検証には、ユニット・テスト、カバレッジ計測、プロファイリングなどが含まれます。これらは、それぞれ互いに関連しあって機能します。これについては後ほど詳しく述べます。