

NUnitを使った

ハイクホリテイ プログラミングの ススメ

最終回

テストングフレームワークによる.NET開発

浅井 育 *ASAI, Hitoshi*
株式会社テクノロジックアート
テクニカルデプトシステム
デベロップメントグループ
<http://www.tech-arts.co.jp/>

実践！テスト駆動開発

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:
NUnit

Level



Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥HIQUALITYディレクトリに収録しています。

¥SALESMANAGER
各コンポーネントの仕様をテストするためのプロジェクト

夏休みの宿題とソフト開発テストは似たもの同士

この時期、所沢の山中にある某野球場などに行ったりすると、いつも以上にたくさんの子どもたちに出会います（執筆時は夏休み真っ最中）。レフトスタンドの応援席のド真ん中で、ライト側の某青色の球団の応援歌に合わせてメガホンを叩いていたりと、無邪気かわいいものです。そんな、狭山茶で育った野球好きの子どもたちにも、いつかはやらなければならない宿題があります。

とは言え、宿題なんて楽しいものでもありませんから、当然後回しにして、楽しいことを優先してしまいます。そうすると、結局ギリギリになって日記とか工作とか自由研究あたりが間に合わなくなり、なんとかごまかして逃げる、ということになるわけです。

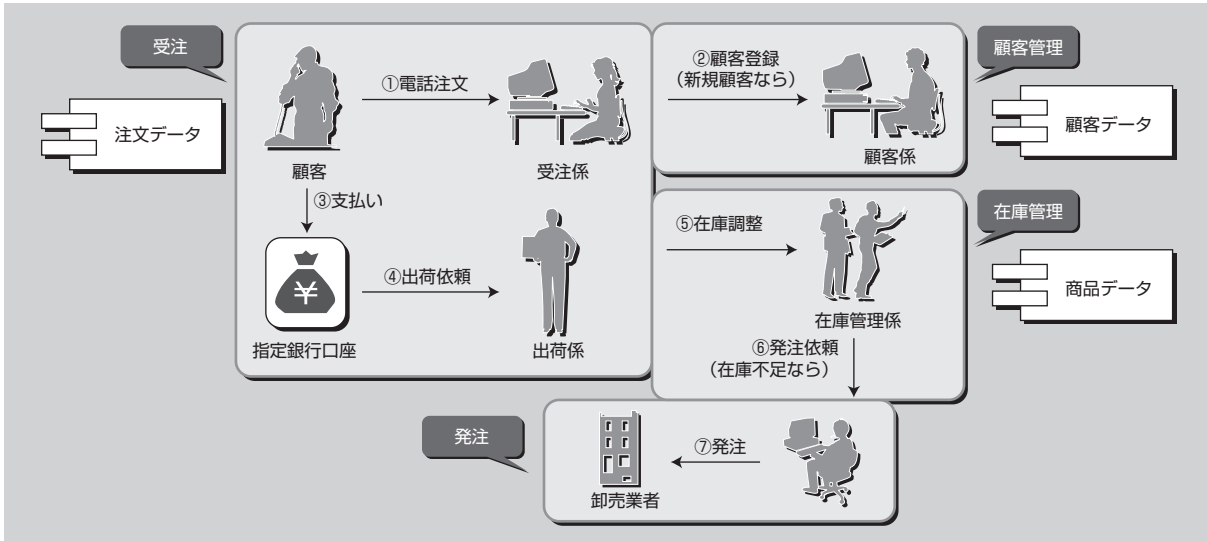
ソフトウェア開発におけるテスト

も同様で、それなりに楽しいプログラミングを優先してしまって、苦痛でさえあるテストを「後回しにしてごまかす」ことが多くなりがちです。

テスト駆動開発の目的は、品質の高いソフトウェアをより高速に作り上げることにあり、そのためにテストを先に済ませてしまう、というアプローチをとります。しかも、先にテストを済ませることができるように、はじめにテストに楽しさを与えることで、目的である品質や開発速度の向上を実現します。

テストの楽しさ、テスト駆動開発の流れ、そしてテストの作り方は前回までで学んできました。そこで、連載最終回となる今回は、経験によって導かれるような知識を身に付けてゆく手始めに、サンプルプログラムを題材にして、テスト駆動開発の一連の流れを経験してゆきます。

図1：コンポーネントに分割された販売管理システム



今回の例題

ここで扱うサンプルは、通信販売を行なう業者の販売管理システムで、以下のような利用を想定しています。

販売管理システムの利用シナリオ

- ・電話で注文に対応
- ・受注担当の係員が端末に受注データを入力
- ・在庫を確認して、必要であれば自動的に卸業者へ発注
- ・入金を確認したら、商品を発送

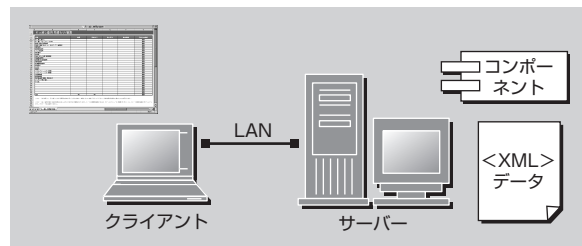
また、このシステムで実現すべき要求仕様は、以下のようになります。

販売管理システムの要求仕様

- ・注文を受け、管理できる
- ・顧客情報を管理できる
- ・在庫を管理できる
- ・商品を発送できる

このように、単純なソフトウェアではなく、業務システムと言われる規模の大きな開発を行なう場合、その開発対象を分割することが非常に重要になってきます。そ

図2：システム構成



のため、ここでは開発対象を目的や責任によって整理したうえで、データを管理するコンポーネントと、実際の処理を行なうコンポーネントの2種類に分割します^[注1]。

このシステムの分析/設計を行ない、分割したコンポーネント構成は図1のようになります。これは、目的に基づいて受注、発注、在庫管理、顧客管理という4つのコンポーネントに分割した後、それらの中で扱われるデータを抽出し、データ管理コンポーネントとして独立させた形です。

また、このシステムは、図2のようなシステム構成で実行されるものとします。

以降は、この販売管理システムのコンポーネントを題材にして、テスト駆動開発を身に付けてゆきたいと思えます。このように分割されたコンポーネントは、単体で

注1) コンポーネントとは、システムを構成する要素を、目的や責任などによって分割した単位。大規模なシステムを効率よく開発するために、コンポーネントに分割することで、ひとつひとつの複雑さを抑えます。