# ソフトウェア設計論

## ―役に立つ UML モデリングへ向けて―

博士（情報科学） 松浦　佐江子 著

コロナ社

# ま　え　が　き

ソフトウェア開発とはなにか，考えてみたことはありますか？

コンピュータは私たちの生活のあらゆる場面に登場し，ソフトウェアを工夫することで私たちの暮らしを便利にかつ豊かにするサービスを提供しています。こうした役に立つ，使いやすいサービスを提供するためには，どのようなソフトウェアを創れば，どんな場面でどのように役立つのかを明らかにする必要があります。そして，明らかにしたことをコンピュータが実行できるように，コンピュータが理解できる指示書を皆さんがつくるわけです。私たちが役に立つサービスを考える技術や，コンピュータへそれを正しく指示できる技術をもつことが必要です。

ソフトウェアをつくるためには，プログラミング言語を知っていれば大丈夫と考えている人もいるかもしれません。でも，本当に必要なものを適切につくれなければ誰も使ってはくれません。リッチな機能をもつシステムは大規模で複雑です。人々の要求は限りないので，素早く対応しなければなりません。一人の力でつくれるものは限られます。本書では，役に立つソフトウェアをどのように考えながらつくっていくのかを，モデリング言語 UML（Unified Modeling Language）を用いて説明します。

モデルはアイデアをスケッチするのには有効だが，それ以上のものではない，と思っている人もいるでしょう。確かに，モデルは，要求を分析し，設計のアイデアを整理し，表現するものですが，そのアイデアをコンピュータへの指示書に正しく反映できなければ役に立ちません。モデリング言語の使い方次第で，役に立つモデルになるかどうかが決まるのではないでしょうか。ソフトウェア開発に携わりたいと考えている皆さんには，役に立つモデルをつくり，役に立つソフトウェアを考える力をつけてほしいと思っています。

モデリングとは，対象を表す模型（モデル）によって，対象のある側面をわかりやすくとらえる作業です。ソフトウェア開発ではなんのためにモデルをつくるのでしょうか？ ソフトウェアは，最終的にはプログラミング言語で書かれた命令列をコンピュータがその順番どおりに実行することで，人々が要求していたことを直接的あるいは間接的に実行するものです。人々の要求という，曖昧で，不正確で，実現できるかもわからないものを，その人々が「思っていることができる！」と思えることを実現する指示書としてつくり込むわけです。指示書ができ上がるまでには，要求項目を決める，要求項目を実現する方法を決める，この方法を効率よく特定の環境で実現する方法を決定する，決定に従ってプログラミング言語を使って指示書をつくる，指示書がこれらの決定に従っているかどうかをテストする，といった段階を踏みます。このような段階形式で作業を進めていくためには，そのときには考えなくてもよいことを捨てて必要な側面だけを明らかにする，という**モデリングの観点**を学習することが，役に立つソフトウェアをつくることにつながります。

観点をとらえてうまく対象の性質をモデリングできたとしても，それが適切であるかどうかを確認することも重要です。この適切であるかどうかの基準が，ソフトウェアの**品質**です。

- 課題文を読み，要求をとらえてUMLで定義する。定義したものが妥当かどうかを検討する。定義した要求はどのような品質を保証できるかを考える。
- 定義した要求をプログラミング言語Javaで定義できるようにUMLで定義する。定義されたものが要求を満たしているかを検討する。

本書では，こういったプロセスをシンプルな事例を用いて解説し，モデリングの観点と品質について一緒に考えていくことにします。

ソフトウェア関連の用語については，広義・狭義も含め，いろいろな使われ方をすることが多いと思います。ここでも，特定の概念には妥当な定義を与えて解説していきたいと思います。

本書では，例えばWebアプリケーションのフレームワークや，データベー

スとの接続などの具体的な方法には言及しません。それらを利用して，いかにつくりたいソフトウェアを設計していくかが本書の主な目的です。

ソフトウェア開発に携わりたいと考えている学生の皆さんが，本書を通じて，人・社会・環境の中で役に立つソフトウェアをつくってみたいと思うようになっていただければ幸いです。

2016 年 8 月

松浦　佐江子

# 目　　次

## 1. ソフトウェアとは



## 2. 事　　例



## 3. 要求分析

## 4. 設　　計

## 5. 設計から実装へ



## 6. 役に立つ UML モデリングへ向けて

# 1 ソフトウェアとは

本章では，ソフトウェア開発の流れと，モデリングの観点について解説し，ソフトウェアをつくるときに，なにを考えればよいかについて整理する。

## 1.1 ソフトウェア開発とは

### 1.1.1 ソフトウェアの役割

コンピュータのできることは「あらゆる計算可能な数を人より速く正確に計算する」ことであるといえる。この能力を最大限に生かすために，ソフトウェアがあり，人間の暮らしを豊かにするためのさまざまな**サービス**を提供している。ここでのサービスとは「コンピュータを使用して社会におけるさまざまな問題を解決する手段の提供」を意味する。

サービスはコンピュータを利用した**システム**によって提供される。システムは単独のパーソナルコンピュータ（以下 PC）であったり，ネットワークで連結したコンピュータであったり，特定の装置に組み込まれたコンピュータであったりする。

**ソフトウェア**は，「サービスを実現するために，コンピュータがどのように動作すればよいかの指示を記述したコンピュータへの指示書」である。そして，コンピュータへの指示書を記述する言語がプログラミング言語であり，「プログラミング言語で書かれたコンピュータへの指示書」を**プログラム**と呼ぶことにする。

サービスを「コンピュータを使用して社会におけるさまざまな問題を解決す

る手段の提供」と考えると，サービスは，これらのシステムが「＊＊できる」という言葉で表現することができる。「＊＊できる」ということは，ソフトウェアの「機能」と呼ばれている。例えば，インターネットショッピングでは，いろいろな種類の商品を選ぶことができる，気に入った商品を注文することができる，注文をキャンセルすることができるなどなど，商店へ行かなくても，いつでも好きなときに，気に入った商品を購入することができる，という一つの解決策を消費者に提供している。

コンピュータを使用して社会におけるさまざまな問題を解決することにより，人，社会，環境へ有益な効果をもたらすことができる。しかし，有益な効果をもたらすには，開発者が本当に必要なことを，ソフトウェアとして正しくつくらないと意味がないことに注意してほしい。サービスが，人，社会，環境に有益な効果をもたらせるかどうかの鍵を握るという点で，ソフトウェアは大きな役割を担っている。

### 1.1.2　ソフトウェアのライフサイクル

先に述べたインターネットショッピングは，ほしい品物を手軽に手に入れたいという要求に対して，一つの解決策を与えているが，ビジネスの世界は競争が激しいので，いろいろな付加価値を付けて顧客を引き付けたり，増やしたりしなければならない。また，利用者もさらなる利便性を求める。一つのサービスも利用者の要求やハードウェアの進化で変化し続け，それに伴い，ソフトウェアも変化しなければならない。

このように，ソフトウェアは人々の要求から出発し，その解決策をプログラムとして実現した後も，その役目を終えるまで新たな要求を取り入れて進化し続ける。**図 1.1** の外側のサイクルがこれを表している。大規模で複雑なものを考えるときの一つの手段として，観点を変えて段階的に物事を考えるという方法がある。サービスの開発では，**要求分析**，**設計**，**実装**という異なるモデリングの観点の段階を経てプログラムを作成する。各段階において，作成されるものを**プロダクト**と呼ぶ。これらは段階ごとに，仕様書，設計書，プログラムと

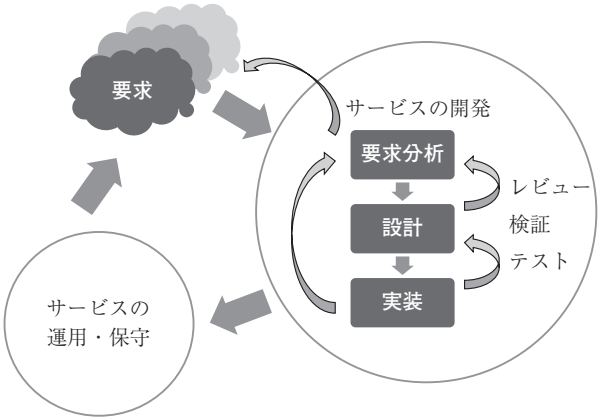


**図 1.1**　ソフトウェアのライフサイクル

呼ばれる。プロダクトがその前段階のプロダクトを満たしているかを確認する方法として，レビュー，検証，テストがある。図 1.1 のサービスの開発の内側のサイクルがこれを示している。要求へ向かった矢印は，要求分析のプロダクトがそもそもの要求を満たしているかを確認する作業である。これらを合わせて，ソフトウェアの**ライフサイクル**と呼ぶ。

要求の変化に伴い，プロダクトを適切に変更しなければならない。そこで，ソフトウェアが変更しやすいという性質をもっていることは重要である。このソフトウェアの性質を**保守性**と呼ぶ。この性質はソフトウェアのライフサイクルにおいて大きな意味をもち，サービスの開発の間につくり込む必要がある。

## 1.2　ソフトウェアの品質

### 1.2.1　品質をつくり込むとは

1.1.1 項で述べたように，ソフトウェアは「サービスを実現するために，コンピュータがどのように動作すればよいかの指示を記述した，コンピュータへの指示書」であり，システムが「サービスを実現するために＊＊できる」というソフトウェアの「機能」をプログラムとして定義する。1.1.2 項で述べた要

求分析，設計，実装の段階を経て，要求を満たすプログラムをつくるわけであるが，「要求を満たす」とは「＊＊できる」というだけでよいのだろうか？

インターネットショッピングの機能には，商品を選ぶことや，商品を注文することがある。しかし，ただ選べればよい，とにかく注文できればよい，というわけではない。商品を選ぶ場合に，例えば「今日は目的の商品があり，それを探したい」とする。商品名が正確にはわからないときには，近いキーワードで探したり，カテゴリーやジャンルといった分類で探したりして，「簡単に」目的の商品にたどり着けると便利である。たくさんの商品が検索できたら，今度は「見やすい」ように商品を絞り込んだり，順番にページを「サクサクとたどってみたり」できるとさらによい。また，商品の写真を拡大してよく見ることができると「わかりやすい」だろう。お気に入りが見つかったら，「スムーズに」買い物かごに入れられるとよいし，支払いをカードでするならば，大事な情報を「安全に」相手に送れないと心配である。

このように，ソフトウェアには，価値のある，役に立つサービスとするために満足すべき性質がいくつかある。例えば，国際標準化機構 ISO（International Organization for Standardization）で策定されている規格の一つであるソフトウェア品質特性の規格では，下記のように六つの性質が定義されている。これらの性質は，ソフトウェアをつくる際に念頭に置くべき事項であり，またつくられたソフトウェアがこれらの性質を満足しているかどうかについて，つねに検討する必要がある。

〔1〕 **機　能　性**　ソフトウェアは，「＊＊できる」ということを，ユーザの要求や開発システムが遵守すべき規格・法律・規則などに従って適切に処理し，期待される正しい結果をもたらすことが必要である。また，ソフトウェアやその対象となるデータに関して不当なアクセスを排除できるように，セキュリティに関する要件も満たさなければならない。

〔2〕 **使　用　性**　ソフトウェアが役に立つ機能を提供していても，難しくて使いこなせなければ，結局は役に立たない。わかりやすく，操作がしやすく，誰でもすぐに使えるようになることが大切である。

〔3〕 **信 頼 性** 交通管理や銀行などのシステムのように，故障による社会的影響の大きいシステムの場合には，ソフトウェアがその機能をきちんと維持できることが大切である。また，障害が発生してもその影響を最小限に食い止め，短時間に容易に復旧できるようにするための工夫が必要になる。

〔4〕 **効 率 性** ソフトウェアは限られたコンピュータの資源，例えばメモリなどを利用するため，実行時には，効率よくその資源を利用しなければならない。さらに，大量のデータに対しても，処理時間や応答時間を短くする工夫が必要である。

〔5〕 **保 守 性** コンピュータを利用したサービスが増加・拡大しているということは，サービスを実現する膨大な量のソフトウェアを開発しなければならないということである。そのために，すべてを新規に開発するのではなく，これまでのソフトウェア資産を再利用することによって，開発が行われる必要がある。すなわち，ソフトウェアは一度つくられたら絶対に変わらないのではなく，つねに変更されるという意識が不可欠である。そこでソフトウェアは，新たな要求に対して変更しやすいように，その内容がわかりやすく，変更によって予期せぬ問題を引き起こさないことを保証するような性質をもつ必要がある。

〔6〕 **移 植 性** ソフトウェアはさまざまな異なる種類のコンピュータの上で動作する。したがって，同じサービスを提供するソフトウェアが，さまざまな異なる種類のコンピュータでも容易に動作するようにできることが必要である。

役に立つサービスを開発するには，このようなさまざまな環境の中で，そのサービスに必要な性質を満たすようにソフトウェアをつくり込む必要がある。

### 1.2.2 機能要求と非機能要求

サービスは「社会におけるなんらかの問題を解決する手段の提供」であるので，ソフトウェアはその実現の目的や目標をもつ。目標を達成するために「＊＊できる」という要求は**機能要求**と呼ばれる。その他の要求をまとめて**非機能**

# 索　　引

―― 著 者 略 歴 ――

1979 年　津田塾大学学芸学部数学科卒業
1982 年　津田塾大学大学院修士課程修了（数学専攻）
1985 年　津田塾大学大学院博士課程単位取得退学（数学専攻）
1985 年　株式会社 管理工学研究所研究員
～2002 年
2001 年　博士（情報科学）（早稲田大学）
2002 年　芝浦工業大学助教授
2006 年　芝浦工業大学教授
　　現在に至る


**ソフトウェア設計論 ―役に立つ UML モデリングへ向けて―**

Software Design Methodology ―Towards Useful UML Modeling―




2016 年 10 月 13 日　初版第 1 刷発行　★

検印省略

著　者　松浦（まつうら）佐江子（さえこ）
発 行 者　株式会社　コロナ社
　　代 表 者　牛来真也
印 刷 所　萩原印刷株式会社

112-0011　東京都文京区千石 4-46-10
**発行所**　株式会社　**コ ロ ナ 社**
CORONA PUBLISHING CO., LTD.
Tokyo Japan
振替 00140-8-14844・電話（03）3941-3131（代）
ホームページ　http://www.coronasha.co.jp

ISBN 978-4-339-02681-8　（金）　（製本：愛千製本所）
Printed in Japan