DIGITAL SERIES

未来へつなぐ
デジタルシリーズ

Connection to the future with Digital series

ソフトウェア システム工学入門

五月女健治
工藤　司
片岡信弘
石野正彦　著

22

共立出版

DIGITAL SERIES

未来へつなぐ
デジタルシリーズ

ソフトウェア システム工学入門

五月女健治
工藤　司
片岡信弘
石野正彦　著

22

共立出版

Connection to the Future with Digital Series

未来へつなぐ デジタルシリーズ

編集委員長：　白鳥則郎（東北大学）

編集委員：　水野忠則（愛知工業大学）
高橋　修（公立はこだて未来大学）
岡田謙一（慶應義塾大学）

編集協力委員：片岡信弘（東海大学）
松平和也（株式会社 システムフロンティア）
宗森　純（和歌山大学）
村山優子（岩手県立大学）
山田圀裕（東海大学）
吉田幸二（湘南工科大学）

（50 音順）

未来へつなぐ デジタルシリーズ　刊行にあたって

デジタルという響きも，皆さんの生活の中で当たり前のように使われる世の中となりました．20 世紀後半からの科学・技術の進歩は，急速に進んでおりまだまだ収束を迎えることなく，日々加速しています．そのようなこれからの 21 世紀の科学・技術は，ますます少子高齢化へ向かう社会の変化と地球環境の変化にどう向き合うかが問われています．このような新世紀をより良く生きるためには，20 世紀までの読み書き（国語），そろばん（算数）に加えて「デジタル」（情報）に関する基礎と教養が本質的に大切となります．さらには，いかにして人と自然が「共生」するかにむけた，新しい科学・技術のパラダイムを創生することも重要な鍵の 1 つとなることでしょう．そのために，これからますますデジタル化していく社会を支える未来の人材である若い読者に向けて，その基本となるデジタル社会に関連する新たな教科書の創設を目指して本シリーズを企画しました．

本シリーズでは，デジタル社会において必要となるテーマが幅広く用意されています．読者はこのシリーズを通して，現代における科学・技術・社会の構造が見えてくるでしょう．また，実際に講義を担当している複数の大学教員による豊富な経験と深い討論に基づいた，いわば“みんなの知恵”を随所に散りばめた「日本一の教科書」の創生を目指しています．読者はそうした深い洞察と経験が盛り込まれたこの「新しい教科書」を読み進めるうちに，自然とこれから社会で自分が何をすればよいのかが身に付くことでしょう．さらに，そういった現場を熟知している複数の大学教員の知識と経験に触れることで，読者の皆さんの視野が広がり，応用への高い展開力もきっと身に付くことでしょう．

本シリーズを教員の皆さまが，高専，学部や大学院の講義を行う際に活用して頂くことを期待し，祈念しております．また読者諸賢が，本シリーズの想いや得られた知識を後輩へとつなぎ，元気な日本へ向けそれを自らの課題に活かして頂ければ，関係者一同にとって望外の喜びです．最後に，本シリーズ刊行にあたっては，編集委員・編集協力委員，監修者の想いや様々な注文に応えてくださり，素晴らしい原稿を短期間にまとめていただいた執筆者の皆さま方に，この場をお借りし篤くお礼を申し上げます．また，本シリーズの出版に際しては，遅筆な著者を励まし辛抱強く支援していただいた共立出版のご協力に深く感謝いたします．

「未来を共に創っていきましょう．」

編集委員会
白鳥則郎
水野忠則
高橋　修
岡田謙一

はじめに

コンピュータの急速な技術発展と普及拡大に伴い，様々なシステムが開発され，生活の中に多くの利便性と有用性をもたらしてきた．これらのシステムではソフトウェアの役割が大きな位置を占めることとなってきた．本書の目的は，ソフトウェアを中心とするシステムの開発について基本的な概念を学ぶことである．

最近のシステムは，ソフトウェアの役割の比重が以前よりも高まってきており，結果としてコンピュータシステムの開発はソフトウェア開発が大部分を占めている．本書も，ソフトウェアの開発を中心に説明し，必要に応じてハードウェアの説明を行うに留める．ソフトウェアの重要性が増す一方で，ソフトウェアは大規模化，複雑化しており，これに対応するためソフトウェア工学の必要性が増大し，その技術が築き上げられてきた．ソフトウェアの開発プロセスは，要件定義，設計，構築，テストから構成されるが，本書では，要件定義と設計の上流工程ついて，基本的な知識の習得と，演習問題を通じた開発の体験ができることを目指している．

本書の特徴として次のものがあげられる．

(1) ソフトウェアシステムの開発の経験のない学生を前提として，わかりやすく具体的な事例をベースに説明し，学習をしやすくしている．
(2) プログラムを開発せずに，ノンプログラミングツールでソフトウェアシステムを構築することで上流工程の成果物を確認できる．
(3) オブジェクト指向アプローチ (OOA) とデータ中心アプローチ (DOA) を統合した開発手法を，一貫した事例に基づき解説している．

本書の構成では，第 1 章で，ソフトウェアシステム工学の全体像とソフトウェアシステムの開発の特徴，および本書の範囲を述べている．第 2 章は，ソフトウェア開発のプロセスとプロダクトについて説明し，本書で扱うソフトウェア開発プロセスの概要を示している．さらに，第 3 章から第 5 章までを要件定義プロセスとし，第 3 章で，UML（Unified Modeling Language；統一モデリング言語）のアクティビティ図を用いて，現状分析を行う方法を説明し，第 4 章で要件分析の方法を，第 5 章で要件定義にユースケース図を利用した方法を説明する．第 6 章から第 10 章までを設計プロセスとし，第 6 章のシステム分析（ロバストネス分析）では，UML のロバストネス図を利用してソフトウェアの構造の設計法を示し，第 7 章のシステム設計（クラス設計）で，これをさらにクラス図に展開する方法を説明する．第 8 章のデータベース設計（正規化）と第 9 章のデータベース設計（ER モデリング）では，OOA によって抽出したクラスの属性をインプットとして，DOA に基づきデータベースの最終的な設計図である ER 図の

導出方法を説明している．第 10 章では，画面などのユーザインタフェースについて，設計上の改善ポイントを解説する．また，構築プロセスでは，プログラミング経験のない方でも設計したソフトウェアが簡単に確認できるように，クラウドコンピュータ上のソフトウェア開発ツールの force.com を利用する．第 11 章，第 12 章，第 13 章では，このツールを使って，設計したソフトウェアの動作を確認する．また，第 14 章でソフトウェアの開発管理の考え方について解説している．

本書は，大学の 15 回の授業に合わせて各章の内容をまとめ，1 章ごとに 1 回の授業で講義できるようにしている．さらに，例題や演習問題として，身近な図書館業務や，会議や催しなどでの参加者の出席管理を行うイベント管理システムを取り上げることで，学生に理解しやすくしている．また，各章に学習のポイントや演習問題をあげると共に，第 15 章に全体の演習問題をあげ，さらに巻末に用語集を入れて活用しやすくした．なお，本書の例題、演習に基づく成果物のサンプルを用意しており，教員が出版社のサイトからダウンロードして活用できるようにしている．

ぜひ，本書を多くの大学の情報系や経営系などの学科での講義や演習などで，ご活用いただけるよう，何卒よろしくお願い申し上げます．

また，本書をまとめるにあたって，大変ご協力を戴きました，未来へつなぐデジタルシリーズの編集委員長の白鳥則郎先生，編集委員の水野忠則先生，高橋修先生，岡田謙一先生，および編集協力委員の松平和也先生，宗森純先生，村山優子先生，山田圀裕先生，吉田幸二先生，ならびに共立出版の編集部の島田誠氏，他の方々に深くお礼を申し上げます．

2014 年 3 月

執筆者　五月女健治
工藤　　司
片岡　信弘
石野　正彦

目　次

第1章
ソフトウェアシステム工学とは

□ 学習のポイント

本章では，本書の対象としているソフトウェアシステム工学とは何かと，ソフトウェア開発の特徴・本書の範囲について述べる．これにより，以降の章に対するイントロダクションとする．

- ソフトウェアシステム工学とはソフトウェアを中心としたシステムを開発する技術であることを理解する．
- ソフトウェアのライフサイクルとは何かを理解する．
- ソフトウェアの特徴およびソフトウェア開発の特徴を理解する．
- 本書の対象とする範囲を理解し，上流工程を主に対象としていることを理解する．

□ キーワード

ソフトウェアシステム工学，ソフトウェア開発，ソフトウェアライフサイクル，上流工程，下流工程

1.1 ソフトウェアシステム工学とは

本書が対象としているソフトウェアシステム工学とは，ソフトウェアを中心としたコンピュータシステム開発に対する技術である．

コンピュータシステムは，企業による金融や生産，あるいは物流などの経済活動，官公庁における行政サービス，電気や交通システムさらに通信などの社会インフラなどの様々な分野で利用されている．また，自動車や家電製品などの制御のためにも利用されている．スマートフォン（以降スマホと略す）やパソコンで利用するインターネットもコンピュータシステムと通信ネットワークで成り立っている．

これらのコンピュータシステムは，ハードウェアとソフトウェアの組合せで構成されるが，ソフトウェアの役割の比重が以前よりも高まってきている．その理由は，システムでの処理内容が複雑化してきており，その複雑化した多くの処理をソフトウェアで行っていること，また，画像圧縮のように従来ハードウェアで処理されていたものもソフトウェアで処理を行うことが多くなってきたことなどによる．

結果としてコンピュータシステムの開発はソフトウェア開発が大部分を占めている．本書も

ソフトウェアの開発を中心に説明し，必要に応じてハードウェアの説明を行うに留める．また，ソフトウェアは，ソフトウェア工学として，幅広い分野をカバーする学問であるが，本書では，このソフトウェア工学のうち，コンピュータシステム開発に焦点を当てた領域について説明を行う．

1.2 ソフトウェアとは

前に述べたように，コンピュータシステムは，ハードウェアとソフトウェアの組合せで構成される．ソフトウェアは，ハードウェアを動作させるプログラムの集まりである．ソフトウェア工学には，開発，運用，保守のライフサイクル全体に対する開発技法や開発手順（プロセス）が存在する．ソフトウェアの開発，運用，保守は，建築物の構築などと比較して，手工業的であるとか，属人性が強いといわれている．これは，一般の工学は科学知識を基本としているのに対して，ソフトウェア工学の体系では，永年に渡って培った，知識や手順を基本としているからである．ソフトウェア工学は，これら先人の培った知識や手順を整理して，誰でもが利用できるような体系にしたものであるといえる．

ソフトウェア工学には，ユーザがどのようなものを必要としているかの要件を分析し定義する要件定義から，具体的にどのようものを開発するかの設計，プログラムの作成や動作確認である構築，構築されたソフトウェアがユーザの要求と合致しているかのテスト，ソフトウェアを実際に動作させる運用，正常に運用するための保守などすべての作業を対象としている．これらの対象領域を図 1.1 に示す．

また，ソフトウェア工学は，それぞれの作業においてどのようなアウトプット（成果物）を作成するかも規定している．すなわち，作業途中，および最終品としてのアウトプットの規定と同時に，これらをどのような手順で作り出すかを規定している．

ソフトウェアのライフサイクルを構成する各プロセスや，個々の作業内容，用語の意味などの枠組みは国際標準化機構により標準化がなされている．多数の関係者の間の認識に差異が生

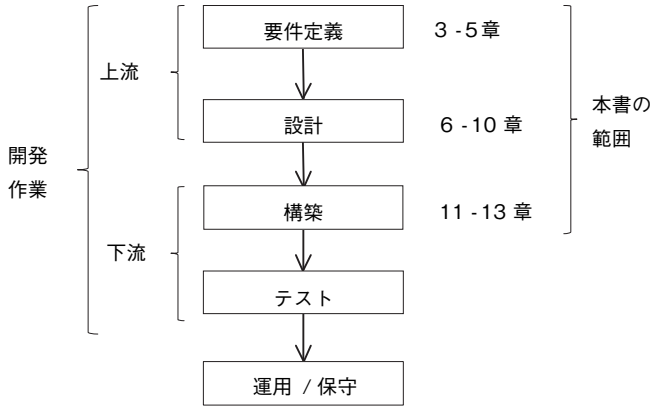

図 **1.1** ソフトウェアライフサイクルの全体像

じないようにすることが目的である．本書も基本的には，これに従っているが，わかりやすさを重点に記述している．

1.3 ソフトウェアの構成

情報処理用語–基本用語 (JIS 0001) では，ソフトウェアを「情報処理システムのプログラム，手続き，規則及び関連文書」と定義している．つまり，システム上で動作するプログラムとそれを作り出す手続き，規則，技法に加え設計文書であるドキュメントを含めてソフトウェアと定義している．プログラムは，コンピュータが行うべき具体的な処理内容を定義した命令語の集まりである．ハードウェアは，このプログラムに従って動作し，演算や比較などのデータ処理，入出力装置を介したデータの入出力を実行し，システムとしての役割を果たす．

これらのソフトウェアは，大きく分類すると次の 3 つに分類できる．ハードウェアを制御するオペレーティングシステム (OS)，ユーザの特定の仕事を行うアプリケーションソフトウェア (以降アプリケーションと略す)，それらの中間に位置するミドルウェアである．OS の例としては，Microsoft 社の Windows があり，ミドルウェアには，第 8 章で説明するデータベース管理ソフトウェアがある．これらを図 1.2 に示す．

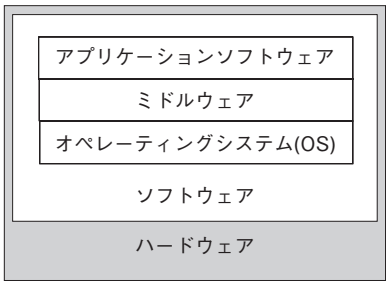

図 1.2 ソフトウェアの構成図

1.4 アプリケーションソフトウェア（アプリケーション）の分類

アプリケーションは大きく分類すると，経済活動や日々の活動のための業務ソフトウェア，発電所制御や新幹線制御のための制御ソフトウェア，自動車や家庭電化製品などの組込みソフトウェアに分類される．これを表 1.1 に示す．

この中で，業務ソフトウェアは，もっとも幅が広く，企業や行政などで利用される銀行業務，生産管理，販売管理，市役所業務などがある一方，日々の生活で利用する，ネットショッピング，電子マネー決済，銀行や郵貯の ATM での処理などが存在する．また，パソコンで利用している会計ソフトや Acrobat のようなパッケージソフトや日々利用しているスマホや携帯電話などのアプリもこれに分類される．

表 **1.1** アプリケーションの分類

分 類	事 例
業務 ソフトウェア	銀行業務，生産管理，販売管理，市役所業務 ネットショッピング，電子マネー決済，ATM処理 PC用パッケージソフト，スマホアプリ，携帯アプリ
制御 ソフトウェア	発電所制御，新幹線制御，人工衛星制御
組込み ソフトウェア	家電組込み，携帯電話組込み，自動車組込み， 自動販売機，

1.5 システムの仕組み

システムは，ハードウェアとソフトウェアで構成されるが，本書で取り上げるソフトウェアは業務ソフトウェアであるため，業務ソフトウェアでの動作の仕組みについて説明する．多くの業務ソフトウェアは，サーバ上に存在し，インターネットや LAN を介してパソコンなどの端末のもとで動作する．端末上では通常ブラウザが動作し業務ソフトウェアに対するデータの入出力を行う．業務ソフトウェアが動作するためには，それが扱うデータが必要であり，これはミドルウェアであるデータベース管理システムにより管理される．このデータには，商品情報や顧客情報などが存在する．これを図 1.3 に示す．

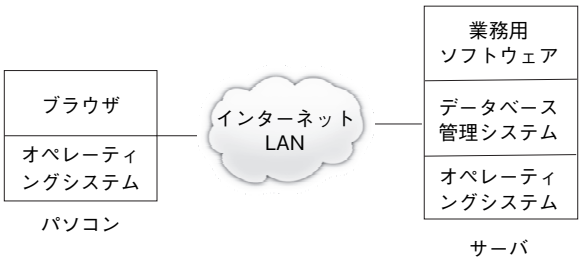

図 **1.3** システムの仕組み

1.6 ソフトウェアの特徴

ソフトウェア製品は物理的な形状をもつ工業製品などと比較して次のような特徴をもっている．

(1) 形状が見えない．

ソフトウェアは，形状が見えないため，規格化が難しく，ソフトウェアの仕様や品質を外観からチェックすることができない．

(2) 開発の実体が掴みにくい．

開発がどこまで，進展しているかは，途中の成果物であるドキュメントにより確認することとなるが，工程が計画通りであるか，仕様が予定のものであるかのなどの確認は難しい．

(3) ユーザごとに開発されるオーダメイド品が多い．

多くの工業製品は，工場で生産され，それがそのまま利用される．同様に，パソコン上で動作する多数のソフトウェアやスマホアプリ，携帯アプリは，パッケージソフトウェアとして開発されたものが変更されることなくそのまま既製品として利用される．一方，企業や行政で利用している業務用ソフトウェアや社会インフラ制御ソフトウェアもパッケージソフトウェアを利用することもあるが，ユーザごとのニーズに応えるため，オーダメイドとして開発されるソフトウェアも多い．

1.7 ソフトウェア開発の課題

ソフトウェアの特徴から出現する課題とその対応策は次のものである．

(1) 形状が見えない．

ソフトウェアの仕様や品質を開発中に外観からチェックすることができないため，最終成果物のテストだけではなく，開発途中でのレビューによる仕様の確認や品質の確認が重要となる．

(2) 開発の実体が掴みにくい．

開発工程を細かく設定し，開発の進捗を途中の開発成果物でチェックする方法などが行われている．

(3) ユーザごとに開発されるオーダメイド品が多い．

有効なシステム開発のために，ユーザの課題や要求を引き出すのが難しい．また，ソフトウェアが対象としている領域も，ますます複雑になってきており，この要件定義の作業を難しくしている．要件定義で確実な定義ができないまま，設計や構築の作業に入り，これらの時点で，要件定義の変更が発生すると設計や構築の作業に大きな手戻り作業が発生することとなる．

要件定義の作業では，ユーザからいかに確実に要求や課題を引き出すかが重要となる．このため，要件定義の作業では，ユーザと確実なコミュニケーションのもとで作業を進めることが必要であり，場合によっては，ユーザ側に入り込み，要求や課題をユーザ側の立場で提示することが必要である．

1.8 ソフトウェアの開発プロセスと本書の範囲

図 1.1 に示すように，本書では要件定義，設計の上流工程を中心に説明を行う．設計したソフトウェアが要件通りのものかどうかの確認のために，クラウド上の開発，運用環境である“force.com”を利用する．これにより，プログラムを作成することなく，設計したソフトウェアの動作確認を行うことができる．

本書の例題，演習問題，総合演習で利用しているソフトウェアは，業務ソフトウェアであり，特に身近な図書館業務，受講登録業務，レポート提出業務などを利用することにより読者の理解を得やすくしている．また，これらの業務はコンピュータシステムとして動作するが，以降の各章では，単にシステムと表現し，業務ソフトウェアも単にソフトウェアと表現する．

演習問題

設問 1 ソフトウェアの特徴を 2 点あげ簡単に説明せよ．

設問 2 なぜ，システムでソフトウェアの位置づけがより重要となりつつあるかを述べよ．

参考文献

[1]（株）NTT データ ソフトウェア工学推進センタ（編著）:『実例で学ぶソフトウェア開発』オーム社 (2008)

[2] 神長裕明他 :『ソフトウェア工学の基礎』（未来へつなぐデジタルシリーズ 13）共立出版 (2012)

[3] 西康晴，他（監訳），Pressman, R. S.（著）:『実践ソフトウェアエンジニアリング—ソフトウェアプロフェッショナルのための基礎知識』日科技連出版社 (2005)

[4] 白鳥則郎（監修）:『コンピュータ概論』（未来へつなぐデジタルシリーズ 17）共立出版 (2013)

第2章
ソフトウェア開発のプロセス

□ 学習のポイント

第1章に示したように，様々な分野でソフトウェアは大きな役割を果たしている．一方で，ソフトウェアに要求される機能は複雑化，大規模化しており，多くのソフトウェアが連携して1つのシステムとして動くようになっている．このため，ソフトウェアの開発は単にプログラムを作成するだけでなく，要求の明確化，実現方式の決定などの様々な作業を，一定の手順に従って行うことが必要になっている．本章は次のことを理解することを目的とする．

- ソフトウェア開発のプロセスとプロダクトについて理解する．
- モデル化によってソフトウェアの機能，構成を把握できることを理解する．
- 本書で扱うソフトウェア開発プロセスの概要を理解する．

□ キーワード

プロセス，プロダクト，クラス，オブジェクト，モデル化， DOA，OOA，UML

2.1 ソフトウェアの開発プロセス

近年ではソフトウェアの利用範囲が拡大している．図書館システムを考えた場合には，図書検索や，図書の貸出し，返却，予約，返却の督促，あるいは会員（利用者）情報の管理など，図書館の運営業務で幅広くソフトウェアが利用されている．これに伴いソフトウェアは複雑化，大規模化している．このため，ソフトウェアをより単純な複数の部品に分割し，これらの部品を連携させることによって複雑な機能を実現するようになってきた．このようなソフトウェアの部品はコンポーネントとよばれ，現在では，ソフトウェアは多数のコンポーネントで構成されるようになっている．

ここで，身の回りの簡単な開発の事例として，家を建てる場合を考えてみる．家が建ってしまってから，実はユーザが希望するものと異なっていたことがわかった場合には，困ったことになる．このため事前に，平面図で間取りを，立面図で家の外観を確認し，どのような家になるのかという情報を関係者の間で共有した上で建築が行われる．同様に，ソフトウェアの開発においても，ソフトウェアにもたせる機能や動きの情報を共有し，その上でプログラム作成を

行うことが重要になる．このため，ソフトウェア開発は段階的に進められ，各段階で情報の確認と共有が行われる．

2.1.1 プロセスとプロダクト

プロセスは互いに関連をもった作業の集合であり，ソフトウェアの開発においては，インプットをアウトプットであるプロダクトに変換させるものである．プロダクトは成果物ともよばれる．ここで，ソフトウェアはユーザの何らかの要求を実現することを目的として開発される．たとえば，図書館の図書検索は必要な図書の有無や，所在を素早く知ることが目的といえる．したがって，ソフトウェア開発プロセスとは，ソフトウェアに対するユーザの要求をインプットとして，この要求を実現するソフトウェアを作り上げる一連の作業といえる．また，プロダクトであるソフトウェアは製品としてユーザに提供される．

一方で，規模が大きくなると，プロセスを分割して段階的にプロダクトを作り上げる必要があり，図 2.1 に示すように，あるプロセスで作成されたプロダクトをインプットとして次のプロセスが実行される．このような中間段階で作成されたプロダクトを中間成果物とよぶ．たとえば，家を建てる場合には設計というプロセスで中間成果物である平面図や立面図が作成され，これらをインプットにして家を実際に建築するというプロセスにより最終的なプロダクトとして家が作られることになる．

私たち自身も日常の生活の中で様々なプロセスを実行している．たとえば，食事を作るのも 1 つのプロセスである．メニューを決めて必要な材料を準備し，材料を切ったり煮込んだりして食事を作る．このとき，慣れた人であればレシピに従って必要な材料を準備し，示された手順に沿って調理するだろう．逆に，レシピを準備せずに作り始めると，途中で材料の不足や手順の誤りによって失敗することになる．このように，良いプロダクトを作るためにはプロセスが重要である．

ソフトウェアの開発も同様であり，ユーザが使用するソフトウェアを開発するのに，ユーザの要求を十分に確認せずに開発を行うと，せっかく作ったソフトウェアがユーザの要求とまったく違うものになっていた，ということが起こってしまう．これは一見，自明のように見える．しかし，実際のソフトウェア開発においては，これが失敗の原因の非常に大きな部分を占めていることが知られている．

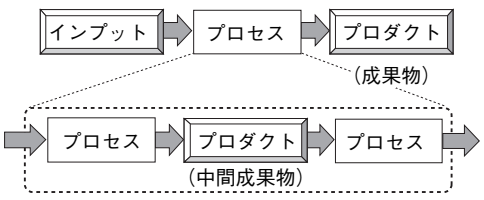

図 **2.1** プロセスとプロダクト

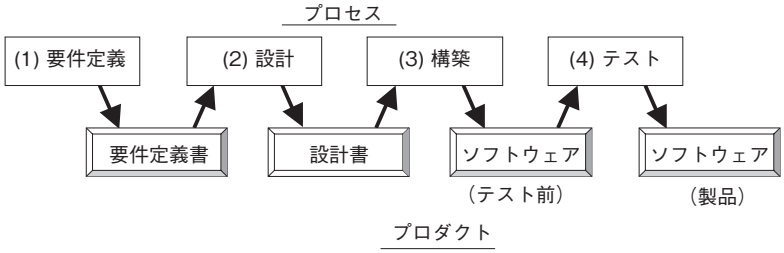

図 2.2 ソフトウェア開発プロセスとプロダクト

2.1.2 ソフトウェア開発プロセスの構成

ソフトウェアの開発プロセスに関しては，対象とするソフトウェアの規模や目的により様々なものが提案されている．開発プロセスを作業の段階ごとに区分する場合，基本的な構成として図 2.2 に示すような以下の 4 つのプロセスに分けることができる．

(1) 要件定義：ソフトウェアはユーザの要求を実現するために開発される．要件定義では，ユーザが抱える課題と解決方法を明確にし，解決方法の実現のためにソフトウェアが備えなければならない機能や性能などの要件を明らかにして，要件定義書としてまとめる．

(2) 設計：要件を満足するためのソフトウェアの仕様を明らかにする．仕様とは要件を満足するための方法といえる．ソフトウェアは複数のコンポーネントから構成されており，要件定義書に基づきソフトウェアの構造や各々のコンポーネントの機能などを設計書としてまとめる．

(3) 構築：設計書に基づき実際に動作するプログラムを作成し，コンポーネント単位あるいは複数のコンポーネントを組み合わせて動作を確認する．

(4) テスト：ソフトウェアが設計書通りに正しく動作することを検証し，さらに要件定義書に示された要件をすべて網羅していることを確認する．

こうして，テストによって確認されたソフトウェアが，最終的な成果物である製品としてユーザに提供されることになる．

2.2 ソフトウェアのモデル化

ソフトウェア開発プロセスでは，開発関係者の間でソフトウェアに関する理解を共有することが重要である．たとえば，要件定義では開発者がユーザの課題を理解し，逆にユーザは開発者が考えているソフトウェアの要件を理解して課題解決に十分なものかを判断する必要がある．ところが，ソフトウェアは目に見えないため，何らかの手段により相互の考えを理解する必要がある．この手段として使用されるのがモデルである．

ここで，モデルとは現実を単純化したものと定義できる．たとえば，家の建築における平面図や立面図は，実際の家をモデル化したものということができる．これによって，開発者とユーザは実際に建築する前に，平面図で家の間取りの，立面図で窓の位置や外観の理解を共有し，同

じ認識の上で家の建築を行うことができる．モデルを作成する作業をモデル化（モデリング；modeling）とよぶ．ソフトウェアでも建築と同様に，モデル化により関係者が理解を共有することができる．また，建築で平面図と立面図があるように，ソフトウェアもまた，様々な視点からモデル化が行われる．なお，本書では，ソフトウェアのモデル化のために，以下の 2 つのアプローチを利用している．

2.2.1 オブジェクト指向アプローチ (OOA: Object Oriented Approach)

図書館システムでは，図書やユーザである会員は実世界に存在するモノである．このような，モノをモデル化したものはオブジェクトとよばれる．このようなオブジェクトに注目してモデル化を行う手法がオブジェクト指向アプローチ (OOA) である．図 2.3 に，図書をオブジェクトとしてモデル化した，図書オブジェクトの事例を示す．図書は，図書番号，書名，棚番号をもっており，図書の管理のためには図書番号の確認や，図書の配架の移動に伴う棚番号の変更が行われる．このような，オブジェクトのもつ性質が属性である．オブジェクトは属性と操作からなり，属性のデータは操作を通じてアクセスされる．図 2.3 の事例では操作として図書が配架されている場所を確認する「所在確認」と，配架場所を移動する「所在変更」の操作をもつ．このように，オブジェクトは属性と，そのデータに対する操作を一体化したものと考えることができる．

OOA では，ソフトウェアは相互に連携したオブジェクトで構成される．たとえば，会員が図書の所在の確認を行う場合には，画面を操作して棚番号の情報を画面に表示する．この場合には，画面を表示する画面オブジェクトは直接図書オブジェクトのデータにアクセスするのではなく，図書オブジェクトの操作である「所在確認」を通じてデータの取得を要求する．このように，他のオブジェクトへの処理は，そのオブジェクトの提供する操作だけを通じて行われる．したがって，あるオブジェクトの処理の内容や方法を変更しても，操作を変更しない限りは他のオブジェクトへの影響がないという利点がある．たとえば，図 2.3 において，画面の操作性を向上するために画面オブジェクトを変更したとしても，図書オブジェクトには影響しない．逆に，図書オブジェクトに，「著者」のような新たな属性を追加しても，操作を変更しなければ画面オブジェクトには影響しない．

ここで，図書館には様々な図書があるが，図書というオブジェクトは図 2.3 に示すように共通の属性と操作をもっている．このような，共通の属性と操作をもつオブジェクトの集合をク

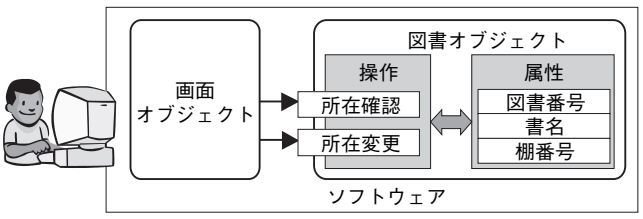

図 **2.3** オブジェクトの事例