デザイン思考に基づく
新しいソフトウェア開発手法
EPISODE

____データ分析，人工知能を活用した____
小規模アジャイル開発

理学博士　西野　哲朗　著

コロナ社

ま　え　が　き

本書の目的は，イノベーティブなシステム開発における自律的，実践的能力を養成することである。そのための手法として，デザイン思考やアジャイル開発の手法に基づく EPISODE という新たなシステム開発手法を紹介する。

この EPISODE の枠組みを理解するために，最初に，通常のソフトウェア工学やアジャイル開発の基礎について学ぶ。つぎに，EPISODE の枠組みについて解説した後に，EPISODE を用いたシステム開発の実践例として，開発ツールとして IBM Watson Assistant を用いた人工知能アプリケーション（チャットボット）の開発事例を紹介する。さらに，データ分析や論文執筆などへの EPISODE の応用法についても紹介する。

具体的には，本書では以下のような学習を通じて，イノベーティブなシステム開発における自律的，実践的能力を養成していく。

(1) **基　　礎**：　最初に，ソフトウェア工学，アジャイル開発，およびデザイン思考の基礎を実践に生かせる形で学ぶ。

(2) **原　　理**：　つぎに，筆者が提案する新たなシステム開発手法である EPISODE の枠組みについて解説する。

(3) **応　　用**：　人工知能開発ツール IBM Watson Assistant を使用し，人工知能アプリケーション（チャットボット）を開発する事例を通して，EPISODE という開発手法を実践的に習得する。さらに，EPISODE の多様な応用例を紹介するために，データ分析，就職のためのエントリーシート作成や論文執筆などへの EPISODE の適用方法についても紹介する。

ここで，筆者が提案しているシステム開発手法 EPISODE の概略を紹介しておく。EPISODE では，本書 7 章図 7.1 のようなサイクルで開発を行う。1 回のサイクルでつくり上げる機能は小さなものに限定し，複数回このような小規

模開発を行うことで徐々にシステムの機能を拡張していく（これはアジャイル開発の考え方を踏襲した方式である）。

(1) 図 7.1 の企画フェーズではブレインストーミングや親和図作成を行い，新しいアイディアを発見する（この部分はデザイン思考の手法を用いて行う）。

(2) つぎに，得られたアイディアからストーリー抽出を行い，コンセプトを明確化する。

(3) 設計フェーズではストーリーを実現するために流れ図（アクティビティ図）を作成する。つづく実装フェーズでは動作が確認できるソフトウェアなどを開発する。

(4) さらに，評価フェーズで価値分析やユーザ自身の検証によって，達成すべき目的が実際に達成できているか否かを確認する。

本書では，システム設計に関する以下のドキュメントの作成方法について解説する。

・企画書
・ストーリーカード
・アクティビティ図
・タスク分割カード
・広　告

これらのドキュメントは，EPISODE の枠組みに従ってシステムを開発する際に使用するものだが，システムの開発後には，これらのドキュメントを開発したシステム（ソースコードなど）と一緒に公開しておく。そうすると，後日，第三者がソースコードを解読する前に，付随したこれらのドキュメントを読むことで，開発されたシステムの概要を迅速に把握することができる。

Education の語源は，ラテン語のエデュカーレ（「引き出す」の意）である。つまり，Education とは，学生の潜在的な能力やさまざまな可能性を引き出すことを意味している。そのため欧米では，教師は学生の能力を引き出す役割を担うものとされている。ところが日本では，Education を「教育」（教え

はぐくむ）と訳したため，「知識をもっている者がもたない者に教える」という知識的側面が強調されてきたようだ。

もちろん，教師は学生に対して新たな知識を伝授するわけだが，本書では，そのことだけにとどまらず，その新たな知識を用いて，読者の潜在的なシステム開発能力を引き出していくことを目標にしたいと考えた。そのことが，多少なりとも，読者諸兄に感じ取っていただけたとすれば，筆者にとっては望外の喜びである。

末筆ながら，8章の実装事例の掲載をご快諾下さった，電気通信大学大学院情報理工学研究科の長谷川勝彦氏と平尾佳那絵氏に感謝いたします。また，本書の刊行にあたっては，コロナ社に，遅れがちな原稿執筆を大変辛抱強くサポートしていただきました。この場をお借りして，深く感謝申し上げます。

2022年1月

西野　哲朗

目　　次

1. ソフトウェア工学

2. 要 求 定 義

3. ソフトウェア設計

4. ソフトウェア構築

5. ソフトウェアテスト

6. アジャイル開発

7. EPISODE

8. チャットボット

9. EPISODE の応用

1 ソフトウェア工学

1.1 学 習 の 目 的

従来のソフトウェア工学[1)†1]は，銀行のオンラインシステムや列車の座席予約システムの開発のような大規模プロジェクトによるソフトウェア開発を対象としている。そのため，従来のソフトウェア工学の手法を用いて，少人数グループによるソフトウェア開発を行おうとすると，以下のような問題が発生する。

問題点 1： 通常，開発するソフトウェアの企画（注文）は顧客がもち込むため，従来のソフトウェア工学には，開発するソフトウェアの企画を立てるための手法があまり含まれていない。

問題点 2： 大人数で開発することが前提となるため，開発したソフトウェアのドキュメント（マニュアルなど）が膨大になり，その把握に時間がかかり過ぎる。

最近は，スマートフォンのアプリ[†2]などを，1 人，または少人数で開発するプロジェクトも多い。しかし，そのような小規模プロジェクトに，従来のソフトウェア工学の手法をそのまま適用しようとすると上記のような問題が発生する。そこで，本書では，少人数によるソフトウェア開発の手法として，筆者の研究室で考案した EPISODE という新たなシステム開発手法を紹介していく。

---

†1 肩付数字は，巻末の参考文献の番号を表す。

†2 以下，アプリケーションを略してアプリという場合がある。

先に結論を申し上げておくと，EPISODE では，上記二つの問題点を，それぞれ以下のような形で解決する。

解決策 1：　EPISODE では，開発するソフトウェアの企画を立案するために，デザイン思考の手法を用いてアイディア出しを行う。具体的には，後述の方法でアイディア出しを行い，ブレインストーミングも行って，開発しようとするシステムの概要を固める。さらに，そのシステムの利用シーンをユーザ目線でストーリーカードの形で書くことで，実現しようとするシステムの企画を具体化していき，企画書を作成する。

解決策 2：　EPISODE において，システム開発のためにソフトウェアを作成する際には，企画書，ストーリーカード，アクティビティ図，タスク分割カード，広告などのドキュメントを作成していくが，システムが完成した際には，ソースコードと一緒に，これらのドキュメントも公開する。これらのドキュメントは理解しやすいので，ソースコード自体を読まずとも，そのプログラムの概要を比較的短時間で把握することができる。さらに，ソースコードを読むことが必要になった場合にも，これらのドキュメントによってプログラムの概要が理解できていると，ソースコードを解読する時間を短縮することができる。

上記のような EPISODE の枠組みを深く理解してもらうためには，従来のソフトウェア工学の内容を一通り理解しておく必要がある。そこで，EPISODE が提案された背景を理解してもらうために，本書の 2 章から 6 章までででは，従来のソフトウェア工学の要点を実践に生かせる形で概観していく。それと同時に，通常のソフトウェア工学の手法を，少人数グループによるソフトウェア開発に適用した場合の問題点についても考察していく。

1.2　コンピュータの発展の歴史

まず，最初に，これまでのコンピュータの発展の歴史を簡単に振り返っておこう。

1.2.1　メインフレーム時代

1950 年ごろに商用コンピュータが初めて登場して以来，コンピュータの歴史は 70 年ほどの歳月になる。当初のコンピュータは高価で大型だったため，企業が保有するコンピュータの台数は少なかった。その後，コンピュータで処理すべき業務は増加の一途をたどり，それに伴ってシステムの規模はどんどん大きくなっていった。

しかし，ハードウェア技術の急速な進歩によりコンピュータの処理能力も飛躍的に向上したため，増えつづける業務を 1 台もしくは数台のコンピュータで処理することができた。このようなコンピュータを大型汎用機（メインフレーム）と呼ぶが，その当時は，コンピュータからの処理結果のみが表示される端末を使ってメインフレームを利用していた。

筆者の学生時代には，大学の大型計算機室に IBM 社製のメインフレームコンピュータが設置されていた。週に一度のプログラミングの授業の際には，出された課題の解答のプログラムを，タイプライタのような穿孔機（穴あけ機）が多数並んでいる穿孔室という部屋で，多数の厚紙のカードにパンチして穴を開ける形で作成する。プログラムの 1 行に対して 1 枚のカードを作成し，それらのカードをプログラムの行の順番に重ねたカードの束（カードデックという）をつくる。さらにその上に，ジョブコントロールカードという「おまじない」のような 10 枚程度のカードを載せて，落とさないようにして慎重に大型計算機室に持参する。もし，不注意でジョブコントロールカードの順番が入れ替わっていたりすると，そのカードデックを読み込ませたコンピュータが止まってしまうということで，ティーチングアシスタントの先輩学生にこっぴどく叱られた。そのため，筆者はメインフレームコンピュータを利用するのが億劫（おっくう）だったことをいまでもよく覚えている。

首尾よく自分のカードデックを受け付けてもらえても，翌週に計算結果を大型計算機室に見にいくと，筆者の学籍番号が表示された棚に入れられたプリンタ用紙に，「エラー」の文字が表示されていて，計算結果が出ていないことがわかる。そこで，慌てて穿孔室に直行してプログラムを修正し，カードデック

（課題の解答）を出し直すというような作業を行っていた。その当時は，コンピュータを利用するということは，そのような作業を行うことを意味していたので，そういった作業に馴染めなかった筆者は，コンピュータ利用は非常に手間がかかるという印象を強くもっていた。そのため学部時代は，コンピュータを使用する授業は極力受けないようにしていたのだが（コンピュータが嫌いだったといっても過言ではない），それがパーソナルコンピュータ（PC）の出現によって，コンピュータに興味をもつようになり，コンピュータサイエンスを専攻することになったのだから，人生というのはわからないものである。

1.2.2　クライアントサーバ時代

その後も，コンピュータで処理すべき業務は急速に増加しつづけ，コンピュータシステムを開発するスピードが追いつかない状況になっていった。1990 年代初めになると，ワークステーションやパーソナルコンピュータの低価格化・高性能化と，ネットワークの標準化が進んだ。その結果，従来のメインフレームによる中央集権的な処理方式から，ワークステーションなどを組み合わせて処理を分散させる方式に移行していった。

この処理方式は，ユーザが使うクライアント（入出力を表示する端末）と，クライアントからの要求を処理するサーバがネットワーク上で結合される構成になっていたため，クライアントサーバ方式と呼ばれた。例えば，電子メールを使用する場合には，各ユーザが使用する PC（クライアント）からメールサーバにアクセスしている。クライアントサーバ方式のシステムを開発する場合には，個々の業務に対するシステムを独立に開発しておき，それらのシステムにサーバを追加していけばよい。そのため，メインフレームのように 1 台のコンピュータですべての業務を処理する場合と比べると，技術面や運用面の難しさがかなり軽減された。これにより，コンピュータ化に要する時間が大幅に短縮された。

これよりも少し前の時代を思い返すと，筆者自身は，コンピュータを個人で占有できるようになったことが，とても嬉しかったことを鮮明に覚えている。

日本初の本格的なパーソナルコンピュータ NEC PC−8000 が，1981 年ごろに大学の研究室に納入されたため個人的に使用していた。いまでは考えられないくらい小規模な 8 ビットパソコンだったが，それでも，自分 1 人でコンピュータを占有でき，カードデックを大型計算機センターに運ぶ必要もなく，カードの並びの間違いを TA に注意されずにすみ，しかも，計算結果がその場で確認できることに，非常に大きな利便性を実感した。そのことで，コンピュータが身近で便利な道具であることが認識できたことから，筆者自身，コンピュータサイエンスを専門分野にしたいと考えたのだった。

1.2.3 インターネット時代

コンピュータの処理能力向上とともに，システムに大きな影響を与えたのが 1990 年代半ばからのインターネットの普及であった。インターネット上では，世界中の数多くのネットワークが網状につながっている。そのため世界の隅々まで，メールで情報の送受信を行ったり，ホームページで情報発信を行えるようになってきた。

同時に PC の小型化，高性能化，低価格化がさらに進んだため，1 人 1 台の PC をもつことが当たり前の時代になった。その結果，企業もインターネットと PC を使ったシステム（Web システム）を採用することにより，より低価格で広範囲なサービスを展開することが可能となった。さらに最近では，スマートフォンを利用したアプリケーションも広く一般に利用されている。

従来のソフトウェア工学は，メインフレームコンピュータによるソフトウェア開発が行われていた時代に，その大枠が構築された。しかし，その後，上記のようにコンピュータとそれを取り囲むインターネット環境が大きく変化したため，インターネット時代の現在に適合するような，新たな形のソフトウェア工学が必要になってきた。そのようなソフトウェア工学における新たな流れを模索する試みとして，本書では，EPISODE をご紹介していきたいと思う。

インターネット時代になり，各個人が世界に向けて情報発信できるようになってきた。そのような時代の流れの中で，個人，あるいは小グループが，自

索　　引

——著 者 略 歴——

1982 年　早稲田大学理工学部数学科卒業
1984 年　早稲田大学大学院博士前期課程修了（数学専攻）
1984 年　日本アイ・ビー・エム株式会社勤務
1987 年　東京電機大学助手
1991 年　理学博士（早稲田大学）
1992 年　北陸先端科学技術大学院大学助教授
1994 年　電気通信大学助教授
2006 年　電気通信大学教授
　　現在に至る

デザイン思考に基づく新しいソフトウェア開発手法 EPISODE
—データ分析，人工知能を活用した小規模アジャイル開発—

EPISODE, A New Software Development Method based on Design Thinking
—Small–scale Agile Development using Data Analysis and Artificial Intelligence—

2022 年 3 月25日　初版第 1 刷発行

検印省略

著　者　西野哲朗（にしの てつろう）
発行者　株式会社　コロナ社
　　代表者　牛来真也
印刷所　壮光舎印刷株式会社
製本所　株式会社　グリーン

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　https://www.coronasha.co.jp

ISBN 978-4-339-02925-3　C3055　Printed in Japan　（金）