

GS LetterNeo Vol.128

2019 年 03 月

2010 年代レトロスペクティブ（回顧）（2）

土屋 正人

はじめに

「〇〇年代レトロスペクティブ(回顧)」と題して、Vol.89 で 1980 年代、Vol.90 で 1990 年代、Vol.106 と Vol.108 で 2000 年代(～2009 年)、それぞれの 10 年間に熟読した本を紹介しました。2010 年代を回顧するにはまだ少し早いですが、平成も終わろうとしているこの時期、ふりかえてみたいと思います。

以前の「〇〇年代レトロスペクティブ(回顧)」に書きましたが、専門書を読む際、気付きをマインドマップでメモするようにしています。マインドマップであれば短時間でふりかえることができます。

今回もマインドマップを羅針盤にして、2010 年以降に出会った本で、今後も読み継がれてほしいものを綴ってみます。

既に Vol.125 でアジャイル関連書を取り上げたので、本号ではモデリングとオブジェクト指向を取り上げてみます。

モデリング

一昔前に比べて、ソフトウェア開発におけるモデリングの重要性が論じられなくなってきたような気がしています。アジャイルでもモデリングは重要だと思いますし、DX(デジタルトランスフォーメーション)では必須の要素になっていくと考えます。

UML を使うことだけがモデリングではないことは論を俟ちません。モデリングは思考方法であり、世界を現す手段であり、問題やソリューションを発見する道具であり、何より楽しいものです。そんな楽しさや新たな発見を与えてくれる書籍を。

・ オントロジエ学の理論と実践

溝口 理一郎 著 オーム社

Vol.108「2000 年代レトロスペクティブ(2)」で紹介した同著者による「オントロジエ学」の続編に相当するもの。タイトル通り理論だけではなく医療分野等における実践例が解説されており、参考になりました。ドメインモデルを構築する上で、同じく Vol.108 で紹介したエリック・エヴァンスの「ドメイン駆動設計」とともに座右においておきたい本です。

- **モデルベース要件定義テクニック**

神崎 善司 著 秀和システム

リレーションシップ駆動要件分析「RDRA(ラドラ)」の教科書。システム価値から機能とデータまでのつながりを意識して要件を定義していく手法で、UMLとその拡張図を採用しています。システム価値→システム外部環境→システム境界→システムの順にスコープを絞り込んで行きますが、その際の各ステップにおける「ヒアリング用の問い」が非常に参考になりました。

- **ピクト図解 — ビジネスモデルを見える化する**

板橋 悟 著 ダイアモンド社

ものごとを捉える視点について分かりやすく説明しています。例えばサッカーの試合。普通の選手は目の前のボールの動きに注意が集まるのに対し、ファンタジスタと呼ばれる選手は、それに加えてピッチ全体を把握しています。試合を中継するテレビカメラは、ピッチ全体から一人ひとりの選手の動きまで追いかけます。ファンタジスタの視点も同じ。著者は、ビジネスでも同様の視点が必要だと説き、アイデア発想法として、ダイアグラム発想法とアナログ発想法を説明しています。ビジネスモデルだけではなく、問題解決全般に適用できるのではないかと思います。

オブジェクト指向

「いまさらオブジェクト指向なんて」と言う前に、「なぜオブジェクト指向なのか？」とふりかえてみるのも良いのではないのでしょうか。コンパクトながら充実した内容の書籍が翻訳されたことは嬉しいです。

- **オブジェクト指向設計実践ガイド — Ruby でわかる進化しつつける柔軟なアプリケーションの育て方**

Sandi Metz 著 高山 泰基 訳 技術評論社

原題(Practical Object-Oriented Design in Ruby)を考慮したサブタイトルが付けられています。その名の通り ruby によるサンプルを掲げ、オブジェクト指向設計原則の中の SOLID を中心に解説しています。これらの

重要さを改めて認識しました。SOLID とは、

- **単一責任原則**

(SRP: Single Responsibility Principle)

クラスを変更する理由は 1 つ以上存在してはならない

- **オープン・クローズド原則**

(OCP: Open Closed Principle)

ソフトウェアの構成要素(クラス、モジュール、関数など)は拡張に対して開いていて(オープン)、修正に対して閉じていなければならない(クローズド)

- **リスコフ置換原則**

(LSP: Liskov Substitution Principle)

派生型はその基本型と置換可能でなければならない

- **インタフェース分離原則**

(ISP: Interface Segregation Principle)

クライアントに、利用しないメソッドへの依存を強制してはならない(インタフェースはクライアント側に属するもの)

- **依存性逆転原則**

(DIP: Dependency Inversion Principle)

抽象化した部分が実装の詳細に対して依存してはならない(実装の詳細が抽象化した部分に依存すべき)

の 5 つの設計原則の頭文字です。オブジェクト指向設計原則を詳細かつ網羅的に解説している書籍としては、Vol.106「2000 年代レトロスペクティブ」で紹介したロバート・C・マーチン「アジャイルソフトウェア開発の奥義」があります。

一番良かったオブジェクト指向本

「そもそもオブジェクト指向とはどのような考え方なのか?」「何が良いのか?」「なぜ必要なのか?」「なぜ良さが感じられないのか?」といった根本的な疑問をわかりやすく解消し、より良い設計に導いてくれるのが次の書籍です。

- **現場で役立つシステム設計の原則 — 変更を楽で安全にするオブジェクト指向の実践技法**

増田 亨 著 技術評論社

オブジェクト指向の長所が生かされない現実が多々ある中、なぜそのような事態になるのかを考察し、リファクタリングやバリューオブジェクト、オブジェクト指向設計原則を取り入れることで改善して行く方法を解説した実践的教科書。コードレビューを行う際に大いに役に立ちました。タイトルに「オブジェクト指向」はありませんが、2010年代の一番良かったオブジェクト指向本です。

GSLetterNeo Vol.128

2019年3月20日発行

発行者 株式会社 SRA 先端技術研究所

編集者 土屋正人

バックナンバー <http://www.sra.co.jp/gsletter>

お問い合わせ gsneo@sra.co.jp



株式会社SRA

〒171-8513 東京都豊島区南池袋 2-32-8

夢を。



夢を。Yawaraka Innovation
やわらかいのべーしょん