

プロローグ

- ①開発スピードの向上, ②AI支援の最大化,
- ③一生ものの開発スキルが身につく

コードをきれいに保つだけじゃない! リファクタリングのススメ

[ご購入はこちら](#)

中林 智之



図1 建て増しで複雑怪奇になっていく建造物…たび重なる機能追加でソースコードにも同じことが起こる

ソフトウェア開発者なら 誰しも経験する負のスパイラル

ソフトウェアを開発していて、次のような経験をしたことはありませんか。

- ソースコードを読んでも設計が把握しにくい
- ちょっとした機能追加のはずが、幾つもの場所に変更を加えないといけない
- ソースコードが複雑でバグの原因を特定するのが難しい
- 開発が進むにつれて実装速度が下がっていく

筆者はソフトウェア開発をしている間、常にこのような経験をしています。他人の書いたソースコードだけでなく、自分で書いたコードで経験することもあります。

これらの経験はソースコードを上記のような状態ではなく、きれいな状態に保てていないことが原因です。どのように対処すればよいのでしょうか。筆者も以前はその方法が分かりませんでした。ソースコードの振る舞いが壊れるのが怖くて、目の前のソースコードをきれいにすることができなかつたのです。

そこから先は泥沼です。きれいな状態を保てていないソースコードにさらに機能追加すると、ソースコードはより複雑怪奇になっていきます。この負のスパイ

ラルを繰り返すと、手の施しようがないソースコードが出来上がってしまいます(図1)。

負のスパイラルを断ち切る 「リファクタリング」

● ソースコードをきれいに保つための活動

幸運なことに、筆者はこの負のスパイラルを断ち切る対処方法に出会うことができました。それからは、冒頭のような経験で苦しむことが減りました。「ソースコードを読んでも設計が把握しにくい」ことはあっても、途方に暮れたり、手のつけようがない、と感じたりすることが少なくなっていったのです。その対処方法が本特集で紹介するリファクタリングです。

リファクタリングは一言で言うとソースコードをきれいに保つための活動です。しかし本稿で説明するリファクタリングは、ソースコードをきれいに保つありとあらゆる活動を意味するわけではありません。明確な定義「外部から見たときの振る舞いを保ちつつ、理解や修正が簡単になるように、ソフトウェアの内部構造を変化させること」があり、バグが入り込まないような実施手順も確立されています。

図2のイメージに示す通り、ソースコードの内部構造を変化させますが、外部から見たときの振る舞い(出力)は保ったままにします。