

レッスン③…リファクタリング 運用マニュアル

中林 智之

表1 各疑問のショート・アンサー

疑 問	ショート・アンサー
なぜリファクタリングするのか？	ソフトウェアを継続的に速く作り続けるため、AIからの支援を引き出すため
どこをリファクタリングするのか？	頻繁に変更を実施するところ、凝集度が低く結合度が高いところ
いつリファクタリングするのか？	ほとんどを開発の流れの中で常にやる、たまに計画された大規模リファクタリングをする
どのくらいリファクタリングするのか？	プロジェクトが必要とするメンテナンス性を得るために、妥当な時間で完了する範囲のリファクタリングをする
どうやってリファクタリングするのか？	手動でのリファクタリングとツール (IDEやAI) でのリファクタリングを使い分ける

本章では、なぜ/どこを/いつ/どのくらい/どうやって、リファクタリングするのかを明らかにしま

す。最初に、それぞれについてショート・アンサーを表1に示します。

3-1 「なぜ」リファクタリングするのか？

技1 リファクタリングは 速く作り続けることを可能にする

組み込みソフトウェア開発は、かつて安定性と信頼性が最優先とされる領域でした。もちろんセキュリティ・クリティカルな分野では今もそうだと思います。しかし、近年では市場競争の激化やソフトウェア主導の製品開発が進むに連れ、継続的に速く作り続けることがトレンドになっています。この背景には次の要因が存在します。

▶1、市場投入までのスピード競争

近年ではIoT機器やスマート・デバイスのように、数カ月単位で新機能の追加や改良が求められる製品が増えています。場合によっては半年～1年で新しいハードウェア製品自体を開発することもあります。

▶2、組み込みソフトウェアの複雑化とアップデートの常態化

組み込みソフトウェアの規模は年々拡大し、スタンダードアローンで動くだけでなく他のデバイスと通信する機能を搭載することが増えています。

その中でユーザー体験の向上、不具合修正、SDKに含まれるものを含めた脆弱性への対処を目的とした

ファームウェア・アップデートを提供することが当たり前になっています。

スマートホーム用のIoTプロダクトはもちろん、2024年にはおもちゃのたまごっちユニ (バンダイ) もファームウェア・アップデートを提供していることが話題になりました。組み込み製品であっても作ってリリースして終わりではなく、継続的なファームウェア・アップデートが求められています。

▶3、アジャイルやDevOpsの影響

従来の組み込みソフトウェア開発はV字モデルが主流でしたが、近年ではアジャイルの文化やDevOps (開発と運用) の手法を取り入れる動きが加速しています。これにより、継続的インテグレーション (CI) や継続的デリバリー (CD) の導入が進み、短いサイクルでのリリースが求められるようになっていきます。

技2 継続的なリファクタリングで 構造の劣化を防げる

● ソフトウェアの構造は開発の中で劣化していく

継続的に速く作り続けるためには既存のコードを理解しやすく保つことが鍵になります。仕様が追加されていくファームウェア・アップデートを提供する開発