リファクタリングとセキュリティ

リファクタリング

必要な理由

- ソフトウェア（アプリケーション）の設計を改善するため
- ソフトウェアを理解しやすくするため
- バグを見つけるため
- プログラムをより早く実行するため
- 既存のレガシーデータベースを修正するため
- 革新的な開発をサポートするため
- ユーザの一貫性を高めるため

効　果

- コードが読みやすくなる
- コードをきれいに整頓
- 重複するコードとコメントの削除
- パフォーマンスの改善
- 一般化する
- 繰り返しを無くす
- 類似コードを組み合わせて破棄
- 長いコードの分割
- 再利用可能なコードの生成
- よりよいクラスと関数の結合

最重要ポイント：セキュリティ

セキュリティの脅威は時代と共に変化

- 脆弱性の撲滅
- アクセス制限強化

長期保存しているワインには、天然素材であるコルクがもたらす弊害を防ぐために「リコルク」が行なわれる場合がありますが、長期に渡って使用しているプログラムでも、多くの場合「リファクタリング」が有効に働きます。
リファクタリングとは、プログラム実行時の外部的な動きは変えず、内部のソースコードの構造を改善することです。
その一般的には、プログラムソースを見やすく改善することで、修正が簡単になり、パフォーマンスが上がることを目的としています。
一方、一度運用に乗ったプログラムに手を加えると、かえって動作がおかしくなったり、他の連携しているプログラムに影響を与える恐れもあるため、バックアップやテストを十分に行う必要が生じます。そのため、リファクタリングをすることに時間を費やすことがおっくうになってしまうのですが、リファクタリングをうまく行うことができれば、担当が変更になってもプログラムを長く維持することができ、再利用もし易くなります。
そこで、見落としがちなのがセキュリティの問題です。サイバー攻撃の手法の巧妙化など、セキュリティの脅威は時代と共に変化し、プログラム作成当時のセキュリティ情報で最善の対策をとっていたとしても、リスクが内在しているプログラムになってしまう可能性が大きいです。プログラムは、ハードウェアではないので経年劣化はしませんが、最新の環境に常に最適化できているわけではないということで、ある意味劣化します。
このプログラムの劣化への対策のひとつが、セキュリティを意識したリファクタリングです。
開発当時に、コード内に一時保管していたデータが、他人からみれば悪意ある第三者から狙われかねない脆弱な部分になることもあります。コード内にパスワードなどを変数に代入してクリアしていない記述があったりするとなおさらです。
ネットワークの利用者が拡大したことで、プログラム内で使用するファイルや、DB にコード内でアクセス制限をかける必要も生じるかもしれません。
いつリファクタリングを行うかということは、いろいろなケースが考えられますが、プログラムへの機能追加やバク修正とは分けて考え、リスク対応計画の中に組み入れ、適宜リスクが感じられたときに行うべきです。
セキュリティを意識して対応することの重要性を認識して、リファクタリングを実行に移すことは、事業継続にも貢献しシステムの劣化を防ぐことができます。

© CEC Customer Service Ltd.