保守性を高めるアプリケーション設計の勘所と留意点【オンラインライブ】(4124150)

DX時代では、基幹系アプリケーションの保守もスピーディな対応が求められています。より迅速、かつ、正確なリリースを繰り返していくためには、アプリケーションのブラック・ボックス状態を解消し、保守性を高める必要があります。本講座は、現場視点の事例を通じて「ブラック・ボックスではない状態」を理解し、アプリケーションの設計・保守に必須の具体的な勘所と留意点を習得する機会を提供します。これによって、スピーディーなリリースを妨げる最大の要因である変更影響と、回帰テストの最小化された高度な保守性を備えるアプリケーション設計を実現することができるようになります。

開催日時	2025年2月14日(金) 9:00-16:00ライブ配信
カテゴリー	IS導入（構築）・IS保守 専門スキル
講師	天羽正道 氏 （フリーランスITアーキテクト ） 元日本アイ・ビー・エム株式会社エグゼクティブアーキテクト チーフアーキテクトとして様々な環境でのビジネス・アプリケーションの開発をリード。現在、アーキテクチャ策定支援などに従事。
参加費	ＪＵＡＳ会員/ITC：35,200円 一般：45,100円（１名様あたり　消費税込み、テキスト込み） 【受講権利枚数1枚】
会場	オンライン配信（指定会場はありません）
対象	保守性の向上に関心のある方 受講前提条件：アプリケーション開発と保守の３年程度の経験を持っていること 中級
開催形式	講義、グループ演習
定員	25名
取得ポイント	※ITC実践力ポイント対象のセミナーです。（2時間1ポイント）
ITCA認定時間	6

主な内容

■受講形態
ライブ配信（Zoomミーティング）【セミナーのオンライン受講について 】
■テキスト
開催７日前を目途にマイページ掲載
■開催日までの課題事項
特になし

DX時代では、基幹系アプリケーションの保守もスピーディな対応が求められています。
より迅速、かつ、正確なリリースを繰り返していくためには、アプリケーションのブラック・ボックス状態を解消し、保守性を高める必要があります。

本講座は、現場視点の事例を通じて「ブラック・ボックスではない状態」を理解し、
アプリケーションの設計・保守に必須の具体的な勘所と留意点を習得する機会を提供します。
これによって、スピーディーなリリースを妨げる最大の要因である変更影響と、
回帰テストの最小化された高度な保守性を備えるアプリケーション設計を実現することができるようになります。

特徴
・様々な設計手法を背景とした現場の保守性向上の経験に基づいた実際的な内容です。
・現場目線で保守性とは何か、保守性の向上とは何か明確にしている
・多くの現場で見られる保守上で不利な設計と是正の事例
・データやアプリの設計を行う際の保守性の工夫
・性能などの非機能要件への保守性を高める対応の考え方
・稼働中のアプリの保守性を高めるリファクタリングなどの方法

・テスト容易性概念の提供とその実現のための着眼点
・保守性を実現する上でのヒューマンファクター
・保守性を向上させる設計の演習を行います。

◆受講者の声◆
・保守性を高める設計手法について具体的な事例を紹介していただきつつ、どこが悪い・良いを明確にしていただけたことはとても良かった。自身チェック時の観点をどう変えれば良いのか見直すことができた。
・保守性を観点とした研修、書籍はあまりなく、勉強になった。
・現場の経験を生かした資料になっており、分かりやすく、かつ、新しい発見や共感できる内容になっていた。
・要件定義時点でも保守性、テスト容易性などを考慮すべき新しい視点を得られた。
・話の内容や資料がわかりやすく、保守性とはなんなのかという点について理解しやすかった。
・日々の業務に追われなかなか保守性について考える時間が取れなかったので良い機会になった。また改めて保守性の重要性を再確認できた。
・資料の出来が非常に良く復習しやすいと感じた。
・どうしたら保守性が高い構造になるのか、具体例を紹介しながら経験に基づいて解説いただいたため、大変わかりやすかった。どうしたら保守性が高いシステムになるのか悩んでいたが、指針を示していただきモヤモヤが晴れた。

1　保守性の価値
・企業がITに求めるもの
・保守性の重要性

2　保守性とは何か
・保守性向上の阻害要因
・構造
・テスト
・その他の施策

3　良い構造、悪い構造
・導出のタイミング
・引数のスコープ
・分岐のタイミング
・共通モジュール
・検査のタイミング
・モジュール分割
・データの正規化・非正規化
・エンティティの分割
・エンティティの属性付与
・補足）主キーの選択

4　分析・設計の手法の適用上の考慮点
・データモデリング
・非機能要件
・性能対応

5　テスト容易性
・テスト容易性とは
・環境関連
・アプリケーション資源
・データとアプリの準備
・アプリの実行条件
・テストの再利用
・問題分析
・機械化・自動化など

6　保守性の改善
・保守性改善の手法

・リファクタリング
・最適影響法
・大規模リファクタリングの考え方

7 保守性の周辺トピック
・保守性についての役割
・要求の扱い
・アーキテクチャー
・成果物と点検
・環境
・エンジニアリング

プロジェクトオーナー、利用者、アーキテクト、開発者、インフラ担当、プロジェクトマネージャーは保守性の役割のサブ項目であり不要です。