

残席わずか

保守性を高めるアプリケーション設計の勘所と留意点【会場・オンライン同時開催】

(4123003)

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	2023年12月13日(水) 10:00-17:00
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・ピー・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、構想を支援。
参加費	J U A S 会員/ITC : 33,800円 一般 : 43,000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	一般社団法人日本情報システム・ユーザー協会 (日本橋堀留町2丁目ビル2階)
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	30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

【選べる受講形態】

A. 会場にてご参加 : [【Co-lab-po \(2階会議室\) 施設利用にあたっての取り組み】](#)B. オンラインにてご参加 : [【セミナーのオンライン受講について】](#)

■テキスト

A. 会場にてご参加 : 当日配布

B. オンラインにてご参加 : 開催7日前を目途に発送 (お申込時に送付先の入力をお願いします)

※開催7日前から開催前日までにお申込の場合、テキストの送付は開催後になることがあります。ご了承ください。

■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。

◆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だけはととても良かった。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に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
- ・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
- ・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