

テスト設計コンテスト OPEN クラス - アピールシート

※全体でA4縦 1 ページに収まるように記述してください。

地域名 東京

チーム ID S270519003

チーム名(ふりがな) リリカル (りりかる)

チーム紹介

金融系事業会社で QA エンジニアをしています。

コンセプト

システムテストですべてを担保するという発想には、次の問題があると考えました。

- ・実際のシステム構成でなくても確認できることまで実構成で行うため、コストが過剰になる傾向にある
- ・要求や設計の段階で取り除くべき問題への対応が遅れ、テストでは判定できない事項も残るため、確認漏れが生じやすい

そこで、リスクの全体を開発ライフサイクル全体で捉え、適切な箇所、適切なタイミングに、適切なアプローチで対処することにした。
テストを減らすことではなく、リリース可否の判断に必要な証拠を、最も早く、確実に、効率よく得ることを目的としています。

工夫点

1. QA アーキテクチャとして、リスクの全体と対処箇所を提示

V03 の変更内容とテスト依頼書の依頼事項からリスクを洗い出し、その 1 件ごとに、開発ライフサイクルのどの工程で、どのように対処し、どのような証拠で確認するのかを対応させました。この対応関係の全体を、ここでは QA アーキテクチャと呼んでいます。リスクの一覧やテストケースの分類ではなく、リスクと対処箇所と証拠の対応として示すことで、どのリスクがどこで担保され、担保しきれない部分がどこに残るのかを見渡せます。

2. リリース判断に使う読み手に合わせた成果物の編成

納品する成果物を、テストを依頼したプロジェクトマネージャがリリース可否を判断するために読むものと位置づけました。冒頭に、リリースしてよいと判断するために何が揃っている必要があり、現時点でどこまで揃っているのかを置いたうえで、依頼事項 1 件ごとの状態と、根拠を確認できる場所を示しています。テスト設計技法の選定、テストケースごとの手順、実施環境、実施記録の取り方は、テストの実施者が参照する内容であるため、中間成果物の側へ分けました。対応をお願いしたい事項も本文へ散らさず、末尾へ一覧として集約しています。

3. テストケースの設計・実行にかかる工数の削減

システムテストで設計・実行するテストケースを、必要な範囲へ絞り込みました。実際のシステム構成や、複数のシステムを通した一連の利用でなければ判断できないものだけをシステムテストへ残し、それ以外は要求・設計、設計レビュー、単体テスト、結合テスト、運用設計で対処します。下位の工程で確認できることを実構成で繰り返さないため、設計と実行の工数を抑えられます。