Engineer

欠陥

製品が要件を満たすのを妨げる不備

どのようなものか

欠陥は、成果物の内部にある問題で、意図と異なる動作をさせたり、本来の役割の一部を果たせなくしたりするものです。プログラマーの多くはバグと呼び、テスト担当者や標準では、faultやflawとも表記します。

重要なのは、_成果物_という語です。欠陥は、ソースコードだけでなく、別の要件と矛盾する要件、想定した負荷を支えられない設計、設定ファイル、誤った値を正解とするテスト、二つ前のリリースで消したボタンを説明するマニュアルにも存在します。どこにあっても、その物自体に属し、誰かが気付いたかどうかに関係なく、存在しています。

テストの用語は、似た三つの語を区別します。この区別には意味があります。人が、仕様の読み違い、間違った前提、キーの打ち間違いといった誤りを犯します。その誤りが、作った物に欠陥を残します。欠陥のある部分を、それが表に出る条件で動かすと、製品が誤った動作をし、その目に見える誤動作が故障です。連鎖は一方向に進み、どの段階でも止まり得ます。一度も実行されない分岐の欠陥や、誰も送らない入力を必要とする欠陥は、一度も故障を起こさず、何年もコードベースに残る場合があります。

何を問題と見なすかは、期待したことによって決まり、期待は、書かれた仕様より広いものです。ISO/IEC 25010の製品品質モデルは、機能適合性を、製品が「指定された条件で使用するとき、明示的および暗黙的なニーズを満たす機能を提供する」程度と説明します。これを、完全性、つまり指定した作業と想定利用者の目標を機能が網羅するか、正確性、つまり結果が正しいか、適切性、つまり作業の完了を機能が助けるかに分けます。誤った合計は正確性の問題で、誰もエクスポートできないレポートは完全性の問題です。同じモデルは、信頼性を別の特性とし、その要素に、無故障性、可用性、耐故障性、回復性を含めます。正しい答えを計算しても、その後、一日に二回クラッシュする製品には、やはり欠陥があります。

リグレッション、つまり退行は、履歴のある欠陥です。動いていたものが、後の変更で壊れた状態を指します。独自の名前があるのは、最初からあった欠陥にはない手掛かり、最後に正常だと分かっているバージョンを持つためです。

概念を判断に生かす

「欠陥とは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 3社の間で受け渡す代わりに、1つのチームがアイデアのスケッチから、お客様がログインして使える製品まで担当します。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 GIS地図アプリの開発を主導し売上を10倍に — 10x 地図製品の開発を主導して実現した売上成長

GIS地図アプリの開発を主導し売上を10倍に お問い合わせ →

なぜ重要なのか

欠陥の負担は、主に、どこで見つかるかによって決まります。コードを書いている本人が見つければ、一分で済み、跡も残しません。レビューで見つければ、会話が必要です。顧客が見つければ、故障自体、サポートの問い合わせ、調査、修正、信頼の一部を失う負担がかかります。実害があれば、インシデントの一部にもなっています。型システム、リンター、テスト、コードレビューは、どれも同じ方向で働き、発見の時点を、欠陥を書いた時点へ近づけようとします。

欠陥には、同じ系列のものがあります。一つの背後にある間違いには、たいてい仲間があります。読み違えた同じインターフェースを六か所で同じように呼ぶ、同じ単位の取り違いを新しいモジュールへ毎回コピーする、同じ未検証の前提を各リクエストハンドラーに置く、といった例です。発見後に役立つ問いは、壊れた行がどこかというより、何がその行を書きやすくしたのか、同じことがほかのどこで起きたかです。

欠陥は、戻ってくることもあります。コードは常に変わり、修正した欠陥が、リファクタリング、マージ、依存関係の更新で再発する場合があります。欠陥があれば失敗し、修正すれば通るテストを作ると、その修正を、ビルドが継続して守る性質にできます。それでも退行が現れたら、バージョン履歴を探索範囲にします。`git bisect` は、正常だと分かっているバージョンと、異常だと分かっているバージョンの間のコミットを、二分探索します。千コミットの範囲でも、約十回テストすれば、バグを入れた一つへ絞れます。

欠陥には、セキュリティー上の問題でもあるものがあります。MITREのCommon Weakness Enumerationは、弱点を列挙し、それぞれを「特定の状況で、脆弱性を生じさせる一因となり得る、ソフトウェア、ファームウェア、ハードウェア、

またはサービスの構成要素にある状態」と説明します。一覧には、攻撃者が利用できる普通の欠陥も、多数含まれます。境界チェックの欠落、エスケープせずにクエリーへ渡す利用者の入力、一つの経路で飛ばした認可処理などです。

実際の活用例

このサイトで最も負担の大きかった欠陥は、自動の指標がすべて正常と判断したものでした。監査で、自動ルールが見逃していた十一件のアクセシビリティーの欠陥が見つかりました。どれも、マークアップの記述ではなく、ページがどう描画されるかに関わる性質だったためです。うち二件は、検査自体にありました。レイアウト検査が開くウィンドウは、すべて高さ900ピクセルでした。そのため、二つの要素の間隔が七ピクセルしかない、852×393の横向きのスマートフォンの画面は、一度も見ていませんでした。また、ルールエンジンは、105のルールのうち、70だけを実行していました。各欠陥を、それによって作ったルールと一緒に台帳へ記録しました。一件だけを直すのではなく、同じ種類の間違いを防ぐためです。レイアウト検査そのものを直すと、三言語にわたり、128回失敗しました。

第二の欠陥には、まったく症状がありませんでした。サイトは、既定でダークテーマを使い、システム設定でライトテーマを指定している読者には、ライトテーマを使います。ページ全体を覆う背景は、ダークテーマでは三層で構成します。各テーマが描く層を数える、小さな構造検査は、最初の実行で、ライトテーマが二層しか描いていないことを見つけました。重ねる層は、ライトテーマを書いた当初から欠けていました。エラーは出ず、すべてのページは描画されました。背景が一層少なくても、意図した選択に見えるため、何か月も調べる理由がありませんでした。修正は一つの規則でしたが、直った状態を保つのは、その検査です。

修正した状態を守るのは、リポジトリのガードエンジンの仕事です。一つの規則が、すべてのバグ修正を証拠へ結び付けます。修正と記したコミットには、それがなければ失敗するテストが必要です。ガード自身も、146のテストを持ちます。欠陥のあるガードは、拒否すべき変更を通してしまい、正常に働いているように見える場合があるためです。

問題が起きるところ

一件だけを直し、同じ種類の問題は残ります。 報告された行を修正し、チケットを閉じても、同じ間違いの仲間が、コードの別の場所に三件残っています。修正のたびに、それを生んだパターンを探します。機械で認識できるパターンなら、その検索を検査に変えます。

直したバグが、戻ります。 半年後のリファクタリングで、元の動作が復活してしまいます。修正が何を守っていたかを、何も記録していなかったためです。修正前にテストを書き、古いコードで失敗するのを確かめ、テスト群へ永続的に残します。

検査器が、欠陥を持っています。 何も知らせずに一部しか読み込まないルール群、すべて同じ高さのテスト用ウィンドウ、すべき動作ではなく、現在の動作を確かめるアサーションは、どれも欠陥が目の前にあっても、成功を報告します。検査にも専用のテストを持たせ、実際に動いたルールは、道具自身の出力から確認します。

網羅率を、正しさの代わりにします。 全行を実行するテスト群でも、結果を調べるアサーションがなければ、誤った結果を通してしまうことがあります。仕様どおりのコードも、暗黙のニーズを満たしていない場合があります。網羅率は、どのテストも触れないコードを見つけるために使い、テストは、どの欠陥を発見できるかで評価します。

「設計どおり」で、正当な報告を閉じます。 動作が仕様に合うとして報告を却下しますが、欠陥のある成果物は、最初から仕様そのものでした。問題が要件や設計にあるなら、その要件や設計に対して欠陥を登録し、根本から修正できるようにします。

参考資料

- 欠陥 — ソフトウェアテストで使うISTQBの定義です。同義語と、関連する誤り、故障という語も説明します。
- ISO/IEC 25010 — 製品品質モデルと、各特性の副特性の概要です。機能適合性と信頼性も含みます。
- git-bisect — バグを入れたコミットを二分探索で見つける、Gitのマニュアルです。テストスクリプトでの自動実行も含みます。
- CWEについて — Common Weakness Enumerationの意味、ソフトウェアとハードウェアの弱点一覧の構成、運営者を説明します。



用語集

実際の制約に合わせて選ぶ

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「欠陥とは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/defect/