

ソフトウェアテスト容易化のための経験則

James Bach, Satisfice, Inc.

可制御性

より良い制御ができれば、もっと多くのテストの自動化・最適化が可能になります。

- スクリプト化可能なインターフェースもしくはテスト装置が利用できる。
- ソフトウェア及びハードウェアの状態・変数をテストエンジニアが直接制御可能である。
- ソフトウェアのモジュール、オブジェクト、または機能層を別々にテストできる。

可観測性

見えるものがテスト可能なものです。

- 過去のシステムの状態及び変数が可視化されている、或いは照会可能である。(トランザクション・ログが見える。)
- それぞれの入力データに対し、識別可能な出力データが生成される。
- 実行時にシステムの状態及び変数が可視化されている、或いは照会可能である。
- 出力データに影響を与える全ての係数が可視化されている。
- 不正な出力データを容易に特定可能である。
- 内部エラーはセルフテスト機構により自動検出され、報告される。

可用性

テストを行うには、まず対象となる製品が利用可能でなければなりません。

- システムにはほとんどバグがない。(バグがあるとテスト時に分析・報告の間接費が追加で発生する。)
- テストの実行を妨げるようなバグはない。
- 製品は機能段階毎に開発される。(つまり、開発とテストの平行実施が可能。)
- ソースコードにアクセス可能である。

簡潔性

簡潔であればあるほど、テスト対象も少なくなります。

- 設計に自己矛盾がない。
- 機能が簡潔である。(一連の機能は要件を満たす上で必要最低限なものである。)
- 構造が簡潔である。(モジュール強度が高く、粗結合である。)
- コードが簡潔である。(外部検査官には有効なレビューができないほどコードが複雑でない。)

安定性

変更が少なければ少ないほど、テストでの混乱も減ります。

- ソフトウェアの変更があまりない。
- ソフトウェアの変更は制御・連絡共行き届いている。
- ソフトウェアの変更により自動テストが無効になってしまう事はない。

情報

情報が多ければ多いほど、テストも効率的に行えます。

- 設計が他の既知の製品と似ている。
- 製品のベースとなっている技術については十分に理解されている。
- 内部、外部、共有コンポーネント間の依存関係について十分に認識されている。
- ソフトウェアの目的が十分に認識されている。
- ソフトウェアのユーザーが十分に認識されている。
- ソフトウェアの使用環境について十分に認識されている。
- 技術文書は入手しやすく、内容は正確でよくまとめられており、具体的かつ詳細である。
- ソフトウェアの要件は十分に理解されている。