

第1章

テストの進め方

基本から学ぶソフトウェアテスト

Cem Kaner, Jack Falk, Hung Quoc Nguyen

発表者 吉富 楓雅

この本が対象にするのは、必ずしもみんなが
ルールを守らない、または守るつもりがない、
もしくは守る必要がないときの進め方だ。

目次

- テスト第1段階：ステップ1～5
- テスト第2段階：ステップ1～3
- まとめ

テスト第1段階



第1段階のステップ

1. 簡単なテストから始める
2. 他にテストすべきことを書き出す
3. 正常値のケースを試し、何が起こるか確認する
4. その場で考えたテストを行う
5. プログラムとその障害についてわかっていることをまとめる

対象とするプログラム

- 入力された2つの数の和を計算するプログラム
 - * 入力できる数は1桁か2桁とする
- 入力の手順
 1. プログラムを開始する際はADDERと入力する
 2. プログラムは入力をそのまま表示する
 3. 次に和を表示する
 4. 各入力後には<ENTER>を押す

ステップ 1

- まずは簡単な数値から
- 2 と 3 の和を求めている
- * テスト終了時の画面の表示状態

?		2
?	3	
5		
?	—	

障害の報告

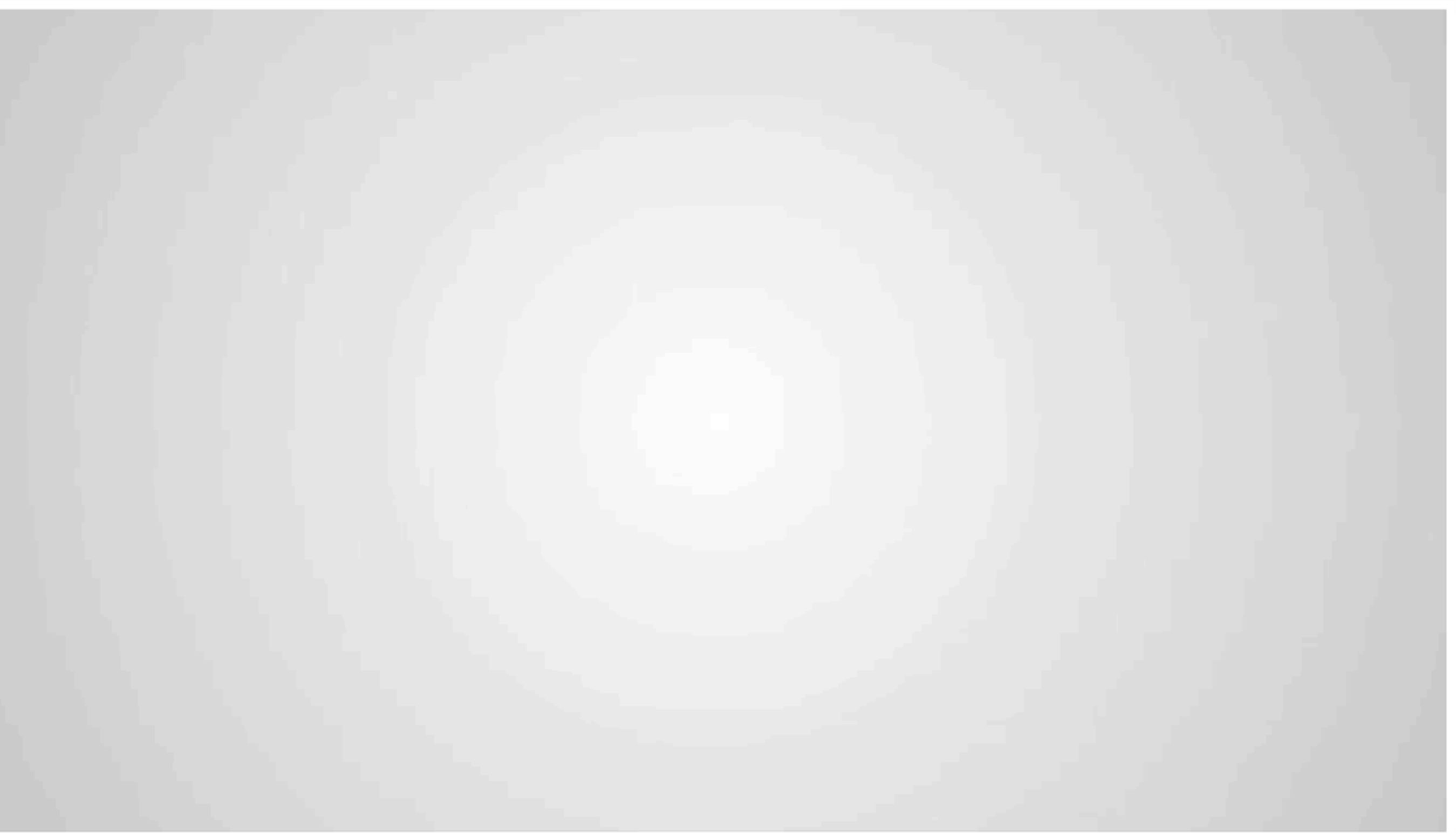
- 2 と 3 を入力して 5 を返したという意味では正常に動作している

* しかし以下のような不具合があると考えられる

1. 設計上の問題：何のプログラムかが分からない. このプログラムを確かに使っていることをどうやって知るのか？
2. 設計上の問題：画面に操作の指示が出ない. 間違った数字を入力したらどうなるの？動作の指示をどこかしらに出したい
3. 設計上の問題：どうやってプログラムを止めるのか？
4. コーディングエラー：合計値(5)は表示されている他の数と一列に並んでいない

ステップ2

- 他にテストすべきことを書き出す
- 全数テストをするのは不可能…
- 全数テストがいかに恐ろしいかは, 次のページの動画で説明されています



境界条件を探す

- 2つのテストに同じ結果が期待されるなら, そのうちの1つだけを使用する
 - * 例: 「22+33」, 「54+87」は同じテストである
- テストの代表を選ぶとき, 常に, 最もプログラムが誤動作しそうだと思うものを採用する

ステップ3

- 正常値のケースを試し, 何が起こるか確認する
- プログラムをテストする理由は, それが正しく動作しないのではないか? というからである
- ある障害が出たケースと同じようなケースを試しても, 大量の時間を無駄に浪費してしまう

ステップ4

- その場で考えたテストを行う
 - * テストの行程が終了すると、次の行程までテストされることはない
 - ▶ しかしテストを行うことはもちろん可能
- 己の本能に従って思いつくままのテストをしましょう

ステップ5

- プログラムとその障害についてわかっていることをまとめる
 - * 必ずしも行わなければならないわけではないが、のちに役立つ
- プログラムについて考えていることをまとめてリスト化するなど

少し練習してみましよう！

対象とするプログラム 2

- 入力された3つの整数で三角形を作成するプログラム
 - * 不等辺三角形, 二等辺三角形, 正三角形のいずれかを表示したい
 - * 作成はC, C++, Javaいずれでも良い
- では, このプログラムをテストするのに十分と思われる一連のテストケースを答えてください

解答 (一例)

- 有効な不等辺三角形をテストする
- 有効な正三角形をテストする
- 有効な二等辺三角形をテストする
- 有効な二等辺三角形で三種類の辺の組み合わせをテストする
- 1 辺の長さがゼロの値をテストする
- 1 辺の長さが負の値をテストする
- 2 辺の和がもう 1 辺と同じである
- 2 辺の和がもう 1 辺より小さい
- 全ての辺の長さがゼロ
- 入力の個数が間違っている
- それぞれのテストケースに対しその出力が正しいかをチェックする

テスト第2段階



第2段階のステップ

1. テスト開始前に障害レポートの回答を検討し、次にすべきことを決める
2. 「修正しない」というコメントの再検討. テスト追加の必要はないか？
3. 前回のメモを取り出し、新しいメモを付け加え、テストを開始する

テスト第一段階の障害レポートに対する回答

1. 設計上の問題：画面上にプログラムのタイトルがない
回答：修正予定なし
2. 設計上の問題：操作指示がない
回答：修正予定なし。コメント：いい指摘だが、処理が遅くなる
3. 設計上の問題：処理を止める方法は？
回答：修正済み。画面に「終了するにはCtrl-Cを押してください」を表示
4. バグ：合計値(5)が他の数字とずれて表示される
回答：修正済み。
5. バグ：負数を入力するとクラッシュする
回答：修正済み。負数も加算できるようになった
6. バグ：3文字目が無条件に<Enter>押下と見なされる
回答：保留(未修正)
7. バグ：数値以外を入力するとクラッシュする
回答：障害ではない。コメント：これはやってはならない

回答の問題点

- 「数値以外を入力するとクラッシュする」
 - * 回答: 障害ではない。これはやってはならない
 - ▶ プログラマにとっては当たり前だが, ユーザはそう感じていない
- プログラマに修正を促すにはどうすれば良いのか?
 - * 例: ブレーンストーミングによって, 項目と理由を書き出す

まとめ



まとめ

- まずは考えられる中で最も簡単なテストから始める
 - * 一通り通れば正式にテスト項目を作成する
- プログラムの品質状態を確かめる
 - * それを書き留め, 次回のテストの参考にする
- 障害レポートの回答を検討する
 - * プログラマが当たり前と想着いても, ユーザは当たり前ではないかもしれない