

第IV章 テストおよび検収

第IV章 テストおよび検収

学習目標

1. テストを行い設計されたシステムの機能性・正確性を検証し、業務上で使用できるか検証する。
2. テスト前のシステムには、エラーが必ず存在するため多くのエラーを検出し、実稼働で影響がないようにテストする。
3. テストの検収に際して、信頼度、正確性から終了を計る。

内容のあらまし

この章では、テストの手法を選び、データを作成し、テストを実施する。SADが行うテストには機能テスト、負荷テスト、退行テストがある。

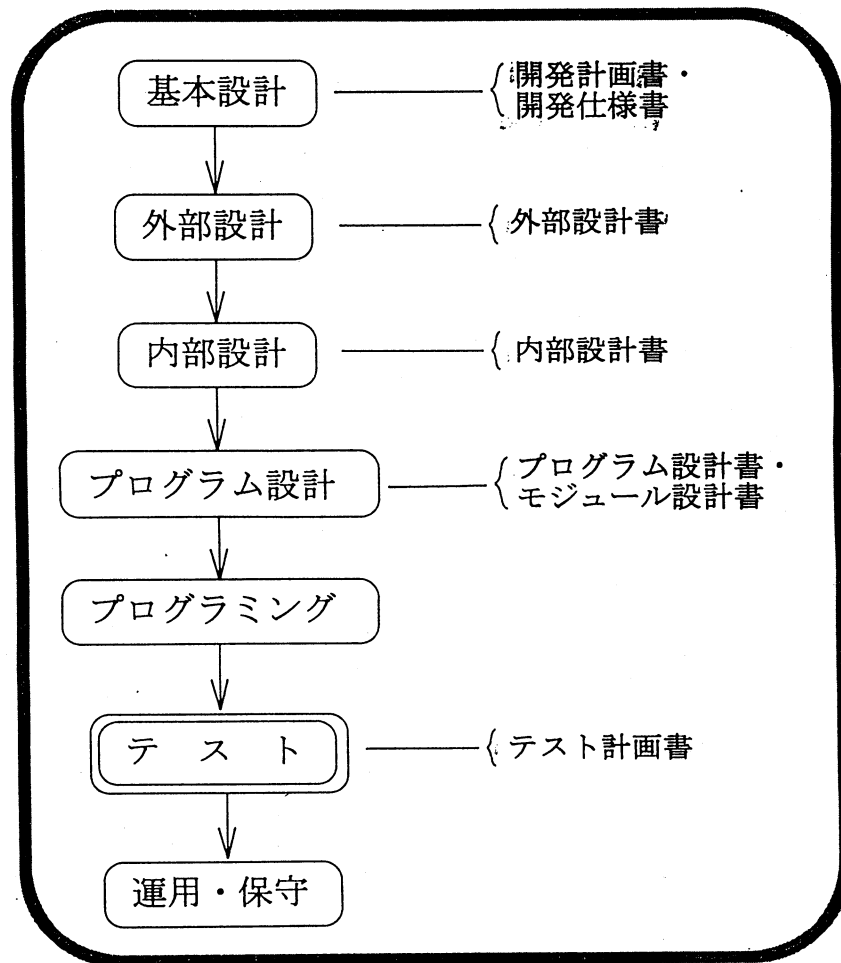
節 項	内 容
1. テストの目的	テストの位置づけと設計書・仕様書 システムアドミニストレーターの行うテスト テストによって検出されるエラー テスト工程 テストの種類 その他のテスト
2. テストデータの作成	テストケースの設計 データの設定
3. テストの実施	
4. テスト結果の検収	信頼性 正確性 安全性

1. テストの目的

テストは、構築制作したシステムが設計どおり正常に動作し、求められている機能を満たしているか検証する作業である。プログラムを機能単位に動かし、その結果の妥当性、システム全体や装置の正当性を事前に設計した設計書、仕様書と一致しているか確認する。

テストの実施においては、必ずシステムにはエラーが存在することを念頭に置いて、実稼働に入ってから障害が発生しないようにエラーを取り除き、品質の高いシステムを作り上げる。

(1) テストの位置づけと開発工程で作成される設計書・仕様書



図表IV-1 テストの位置づけ

(2) システムアドミニストレータの行うテスト

テストは大きく分けて開発過程でのテストと製品システムテストに分けられる。

- ① 開発過程でのテスト（制作者側のテスト）
 - a. プログラム単体テスト
 - b. 結合テスト
- ② 製品システムテスト（利用者側のテスト）
 - a. システム（総合）テスト
 - b. 運用テスト

利用者の立場であるシステムアドミニストレータは、②の総合テストと運用テストを行う。
 テストの評価項目として、

- ・信頼性（正しい動作）
- ・操作性（使いやすさ）
- ・機能（要求された機能を満たしているか）
- ・性能（処理速度の速さ）
- ・保守性（保守や拡張が容易か）

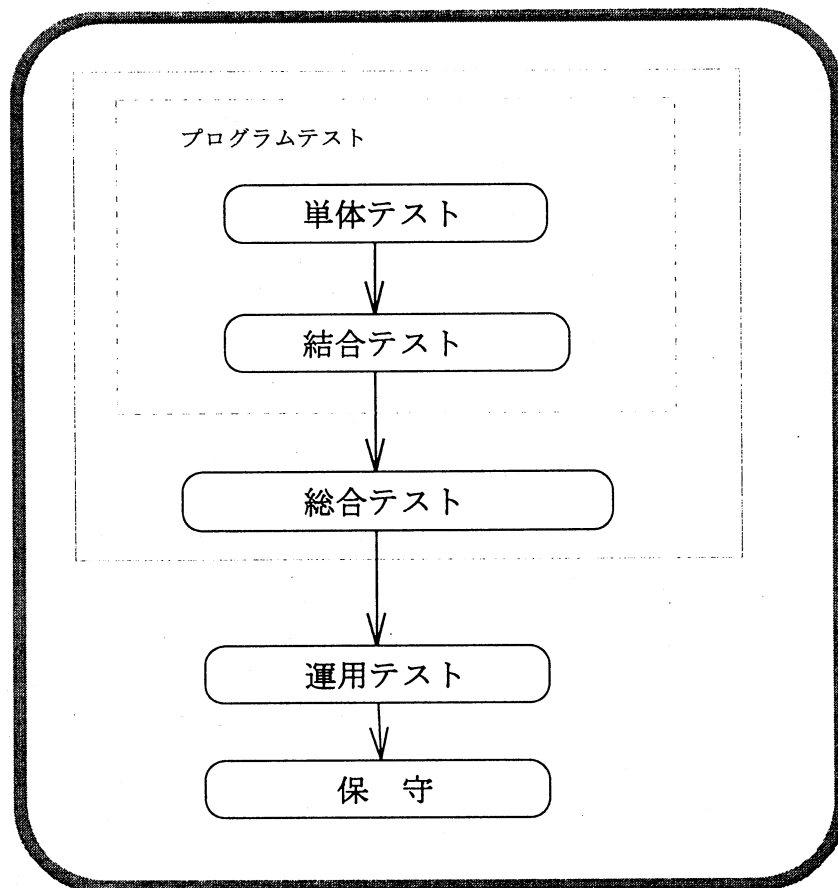
などがあるが、これらをふまえてテストする。

システムアドミニストレータとしての参画は、ウォーターフォール型の開発システムでは、総合テスト以降で、プロトタイプ型の開発システムではいつでも良い。システムアドミニストレータは、システムを利用する側の立場から機能性、インタフェースの設計の整合性を検証するが、システム開発でのテスト行程と内容の把握をしておく、エラー発生時の有効な対処ができる。

(3) テストによって検出されるエラー

- | | |
|-------------|------------------|
| プログラムエラー | ・・・コーディング時のエラー |
| 設計エラー | ・・・設計時のエラー |
| 性能エラー | ・・・システムの制約の考慮漏れ |
| 運用エラー | ・・・運用上の制約の考慮漏れ |
| 障害・回復エラー | ・・・リカバリ機能のエラー |
| オペレーションエラー | ・・・操作ミスによるエラー |
| テストケース作成エラー | ・・・テストケース設計時のエラー |
| テストデータの不足 | |
| テスト環境の不備 | |

(4) テスト工程



図表IV-2 テスト工程

① 単体テスト

モジュールごとに、仕様書どおり動作するか検査する。テストの範囲がモジュール単位なので、エラー箇所の特定が容易である。1つのモジュールで構成される場合、プログラムのテストが単体テストとなる。

② 結合テスト

単体テスト後、関連するモジュールやプログラムを連結して、仕様書通り動作するか検査する。モジュール間インタフェース（モジュール間のデータの受け渡し、入出力のタイミング）の機能の検証も行う。

- a. ボトムアップテスト
- b. トップダウンテスト
- c. ビックバンテスト
- d. 折衷テスト

③ 総合テスト

結合テストの終了したモジュールを結合して、システム全体が正しく機能するか確認する。システムとして要求された機能がすべて仕様書どおりに動作し、求められた機能を満たしているか検証する。安全性、操作性についても検証する。

- a. 機能テスト
- b. 性能テスト
- c. 負荷テスト

④ 運用テスト

実際の運用環境と同じ条件でのシステムの機能、操作の検査をする。ユーザ側を中心として、実データの使用やオペレータの参加など、仮運用テストとして行う。

- a. 受け入れテスト
- b. 導入テスト

⑤ 保 守

システムの性能を維持し、また障害が発生した場合、システムを回復させること。

(5) テストの種類

システムアドミニストレータの立場として必要なテストは性能テスト、負荷テスト、退行テストである。

① 性能テスト

応答時間などのシステムの性能を評価する。

② 負荷テスト

実際の運用時に最大に稼働させることを想定して、システムの性能をテストする。データ量、リクエスト数、稼働時間を評価項目とする。

③ 退行テスト

システムの保守段階で、仕様変更やプログラム変更を行った場合、今までの機能が正しく動作することを確認する。変更や修正で新たなエラー発生を生む場合があるので注意する。

(6) その他のテスト

① 受け入れテスト

ユーザ主体のテストデータを用意して、システムが十分に機能しているかテストする。ユーザが実際にテストし、機能面と操作面をチェックする。

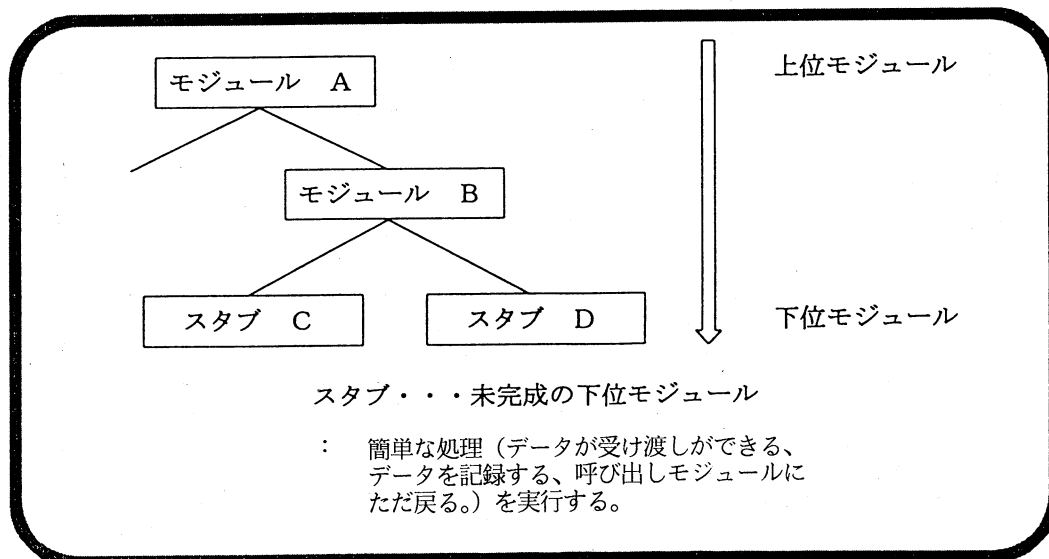
② 導入テスト

実際の環境でシステムを運用してテストする。

③ トップダウンテスト

上位モジュールから順にテストとしていく方法、スタブ（未完成の下位モジュール）を使用してテストを進める。

・モジュールとスタブの間にインタフェースのテストができる。

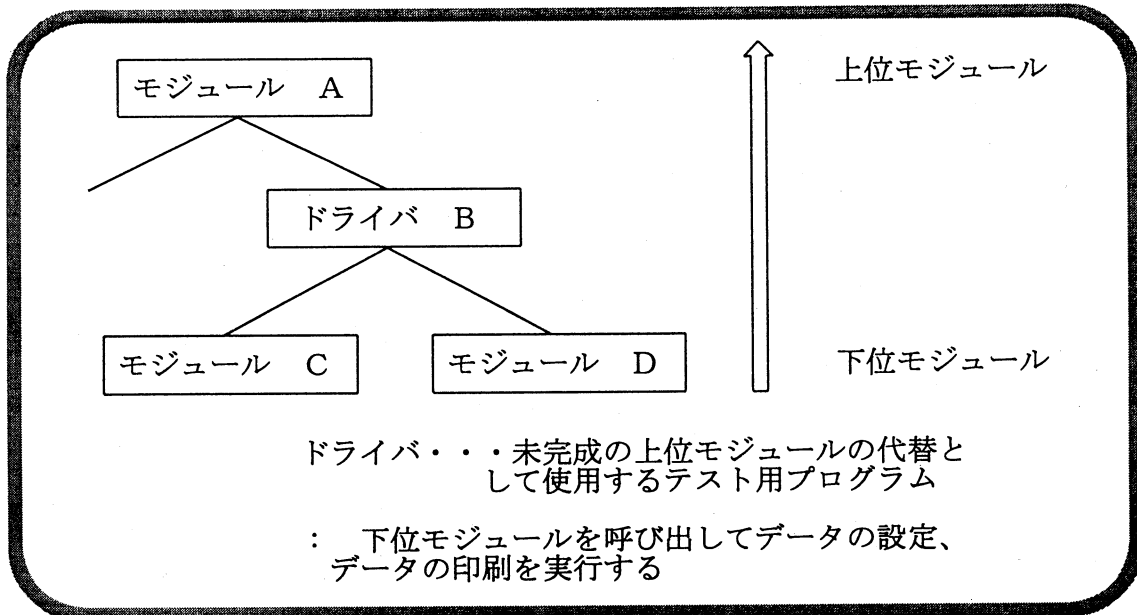


図表IV-3 トップダウンテスト

④ ボトムアップテスト

下位モジュールから上位モジュールに向かってテストしていく方法。ドライバ（未完成の上位モジュール）を使用してテストを進める。

- ・下位モジュールの開発ができる。

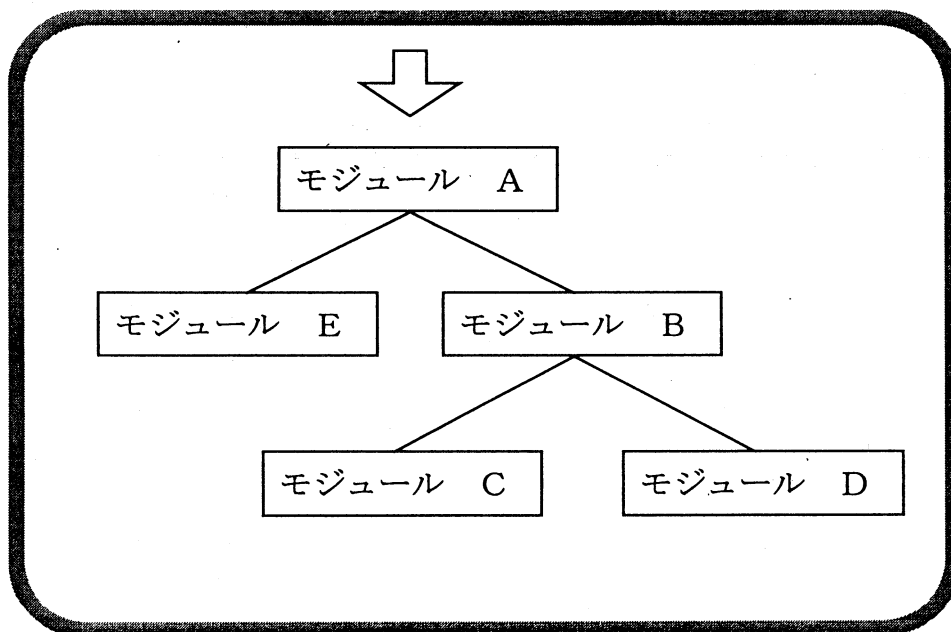


図表IV-4 ボトムアップテスト

⑤ ビッグバンテスト

各モジュールの単体テスト後、全て結合し、一斉に結合テストを行う。各単体モジュールごとのテストの段階でドライバやスタブを用意する必要がある。

- ・全モジュールのテストができる。

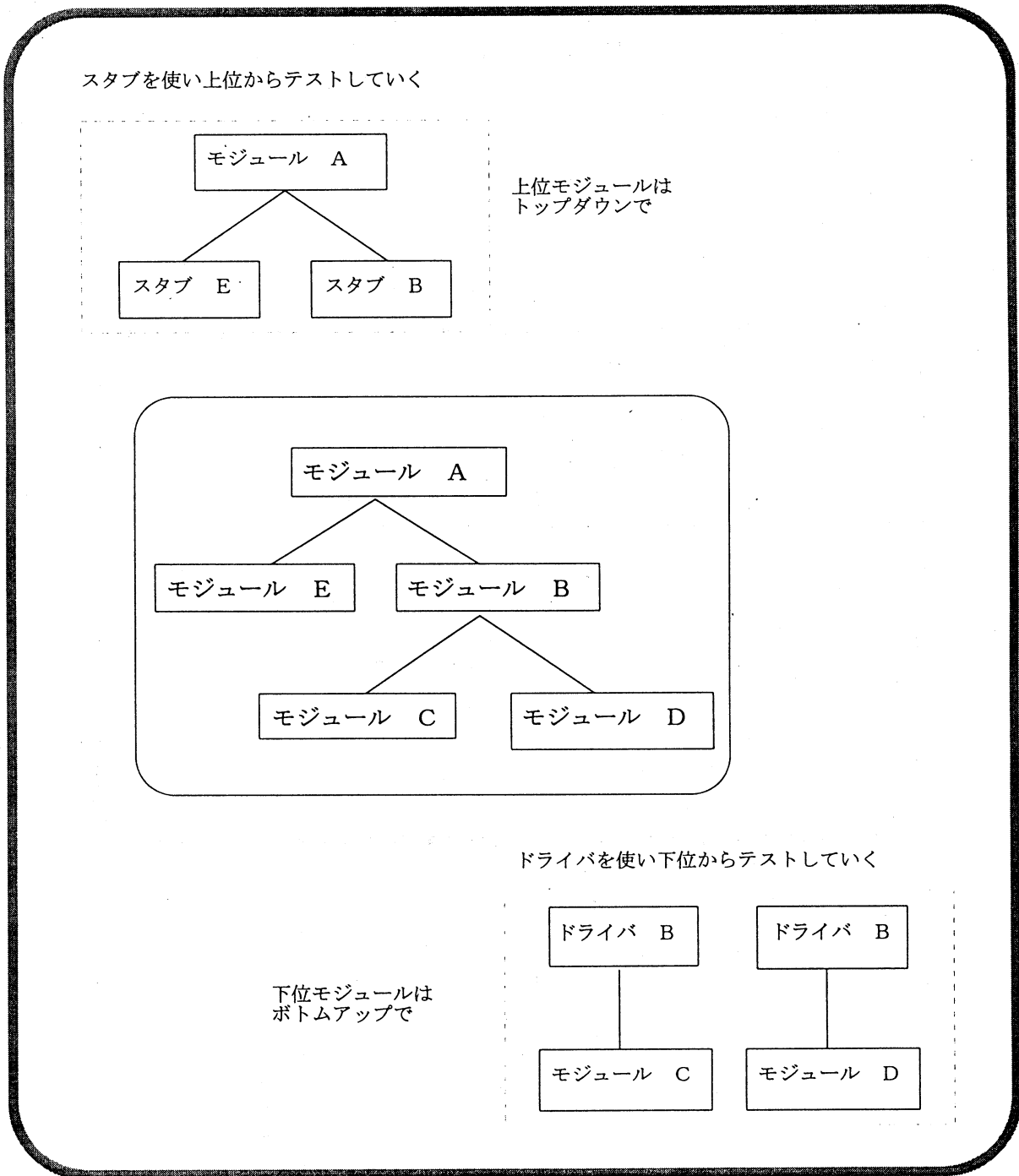


図表IV-5 ビッグバンテスト

⑥ 折衷テスト

ボトムアップテストとトップダウンテストを組み合わせるテストを行う。

- ・テストを平行してできる。



図表IV-6 折衷テスト

2. テストデータの作成

(1) テストケースの設計

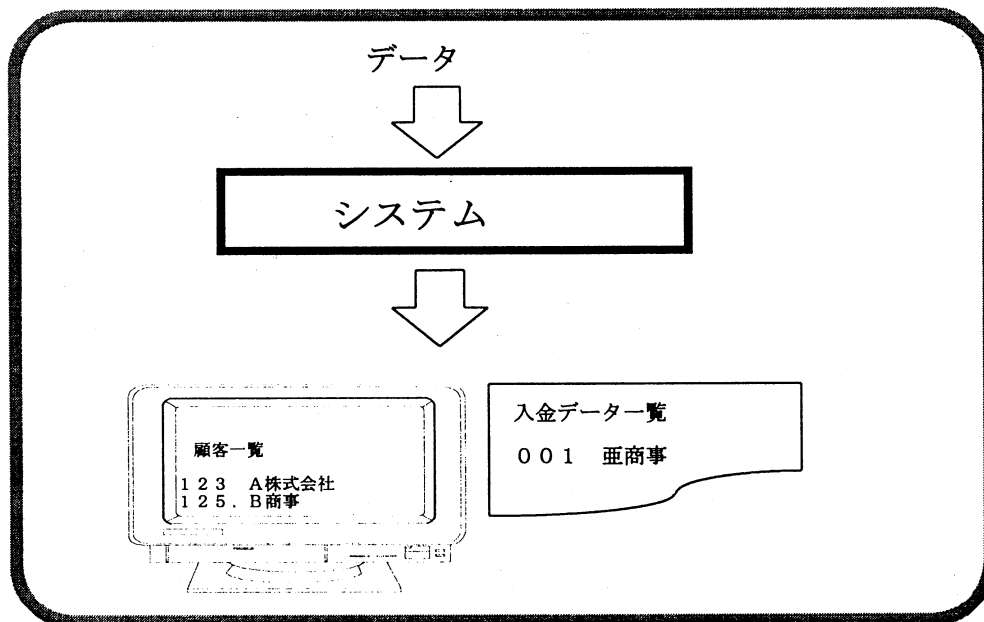
システムが稼働し始めてからエラーの発生がないように十分なテストケースの設計を行う。早期のテスト段階で、エラーを取り除き品質の高いシステムを作り上げる。テストケースの設定は、どのようなデータを使い、どのような方法でテストするか設定することである。

システムアドミニストレータの立場としては、ブラックボックス法の方法によるテストを行うのが一般的である。

① ブラックボックス法

プログラム設計書や内部設計書で設定した入力・出力の機能を満たしているかテストケースを設計する。

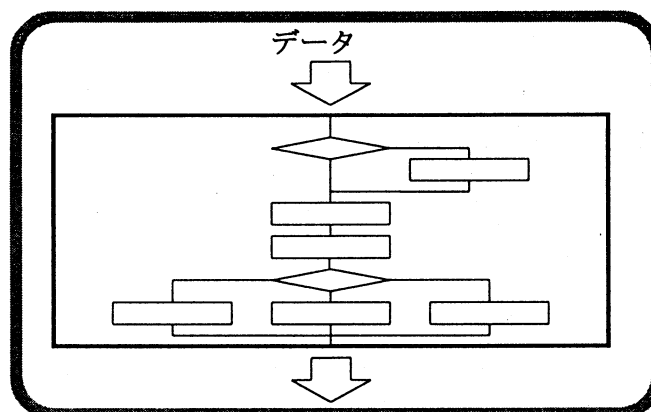
処理手順は問題にせず、設計通りの形式の出力や機能を確認する。



図表IV-7 ブラックボックス法

② ホワイトボックス法

プログラムのロジック（論理）をすべて、または重要な部分を通るようにテストケースを設計する。プログラム内部の実行が正しく行われているか確認する。プログラム内部の実行ルートに従い、処理手順を確認する。



図表IV-8 ホワイトボックス法

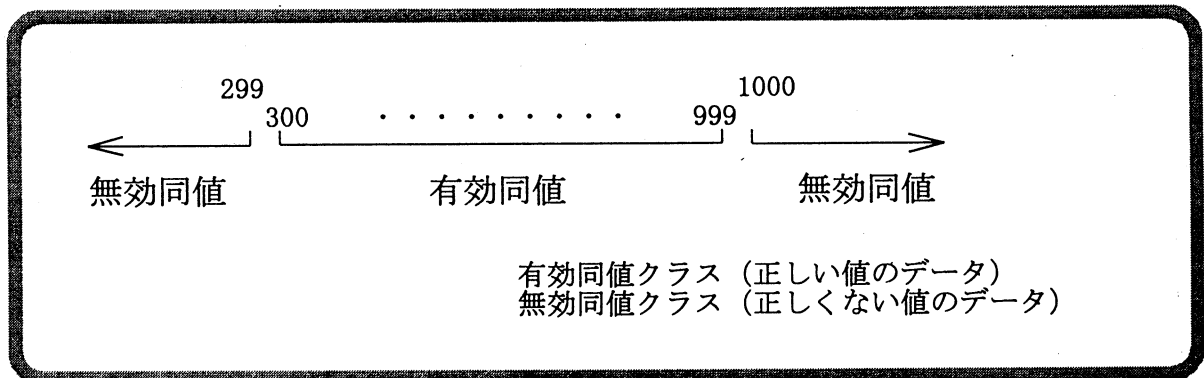
(2) データの設定

テストケースに合わせデータを設定していくが、正常データの他、異常データを用意する。

- ・重複データ
- ・データ量の不足
- ・エラー条件を発生させるデータ
- ・特殊処理データ
- ・異常終了をさせるデータ

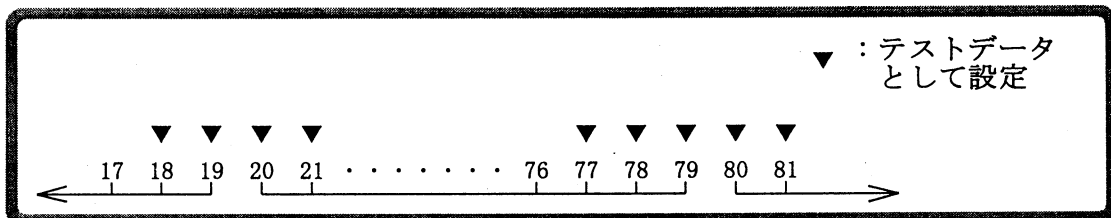
① ブラックボックス法

- a. 同値分割・・・テスト用の入力データの取りうる値を想定していくつかのクラスに分け、代表的な値をテストデータとして用いる。



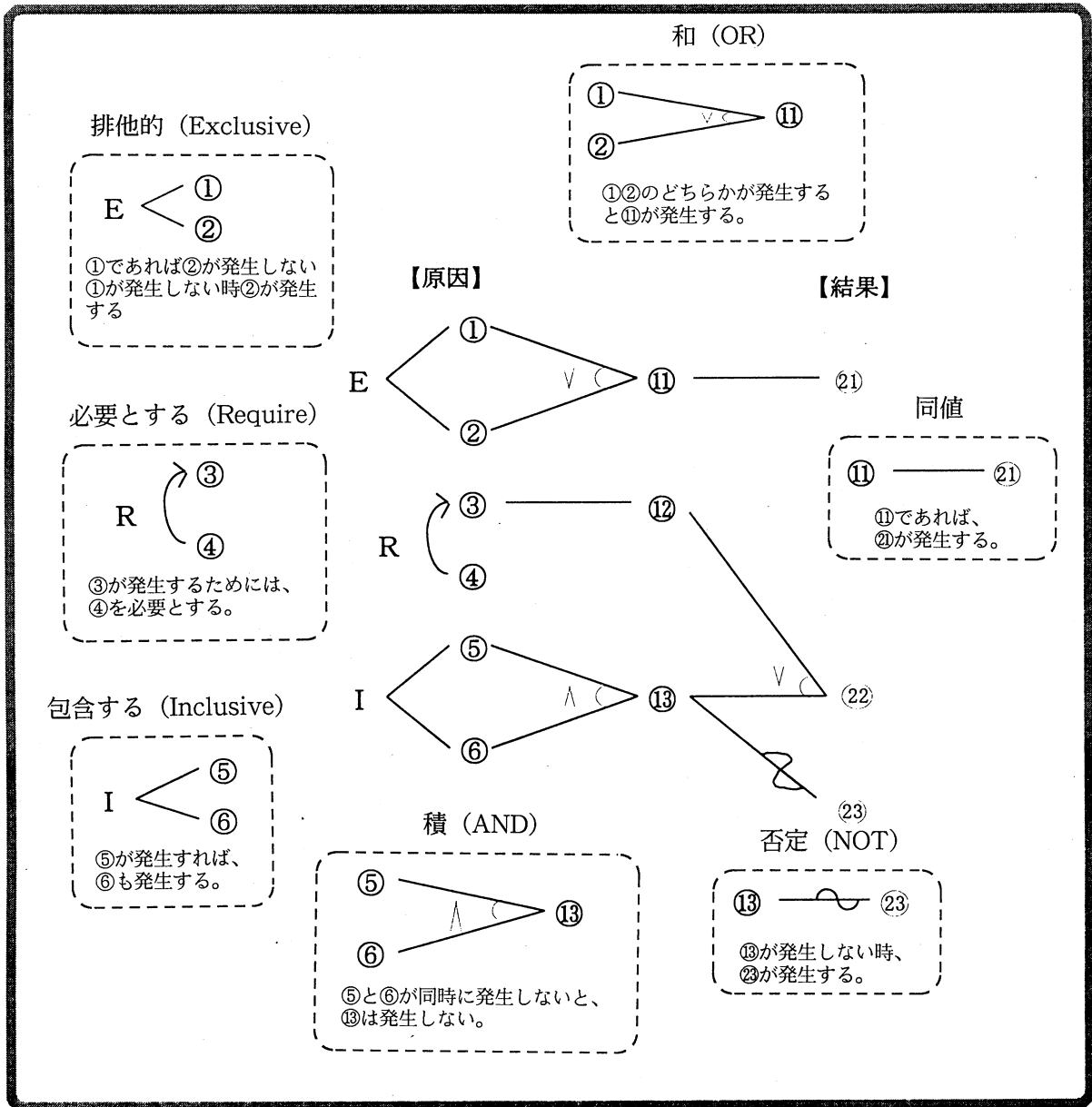
図表IV-9 同値分割

- b. 限界値分析・・・プログラムが許容する限界値付近の値をテストデータとして用いる。



図表IV-10 限界値分析

c. 因果グラフ・・・細かい機能単位に相互の因果関係をグラフに表す。データがクラス分けできないときに有効。



図表IV-11 因果グラフ

d. エラー推測法・・・過去に発生したデータのうち頻度の高いものをテストデータとして用いる。

② ホワイトボックス法

- a. 命令網羅法・・・すべての命令を最低1回は実行させる。
- b. 判定条件網羅法・・・ある条件の真・偽の両方を最低1回は実行させる。
- c. 条件網羅法・・・判定条件のすべての可能な結果を最低1回は実行させる。
- d. 複数条件網羅法・・・条件の組み合わせをすべて1回は実行させる。

3. テストの実施

① 機能テスト

外部設計の仕様どおり機能しているかテストする。ユーザー側の要求する機能を満たしているか、インタフェースの関連などを検証する。帳票類の出力、入出力の仕様、操作性など主要な部分のテストから始めて全機能をテストする。

② 性能テスト

システムとして要求された性能をテストする。多数の利用者の使用を前提としたテストを行い、処理速度、応答時間、レスポンスタイムが仕様通りかテストする。

③ 安全性テスト

万一の障害に備えて、バックアップの保存、停電などによるシステムダウンを想定して、リカバリ機能が正常に動作するかテストする。バックアップデータからシステムを再起動できるかテストする。

④ 操作性テスト

ユーザー側に立っての操作のしやすさ、使いやすさをテストする。カーソルコントロール、カーソルガイドなど操作面で問題がないかテストする。

4. テスト結果の検収

テストは、テスト仕様書に基づいてシステムの品質・性能をテストしていくが、進捗状況や信頼度から目標に対する達成度を把握しておくことが大切である。

評価項目として信頼性、品質の正確性、安全性、操作性などがある。

(1) 信頼性

システムには安定した運用が求められている。運用中の処理の中断は、システムの信用がなくなるだけでなく、経済的な損失も発生する。チェック機能、障害発生防止の機能などの評価する。信頼度を測るものとして、稼働率がある。

① 稼働率

ある時点でシステムが正常に機能している確率。

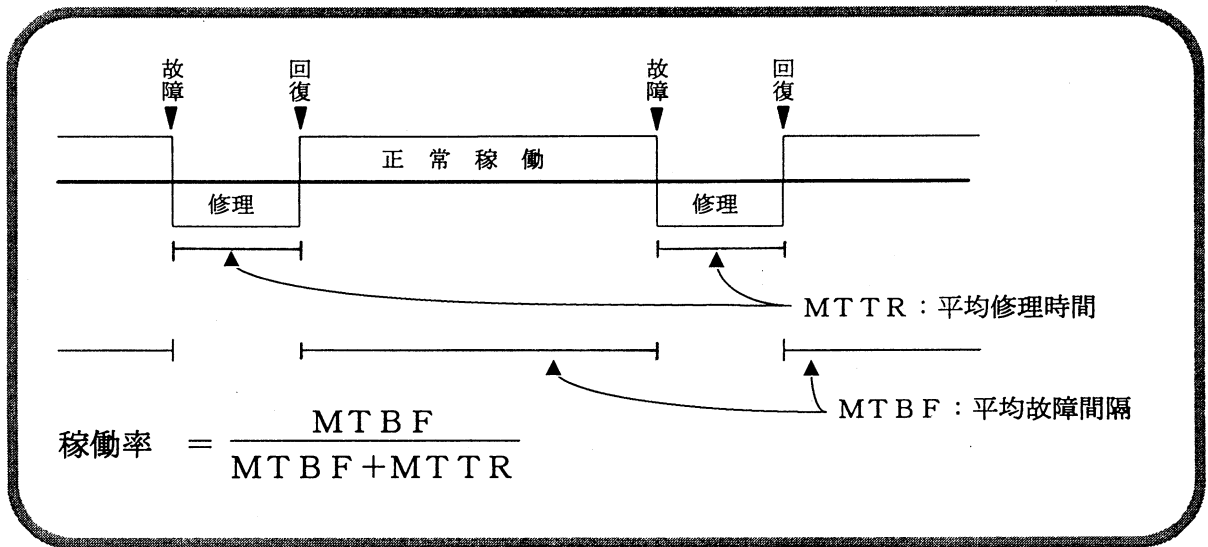
稼働率 = 平均故障間隔 / (平均故障間隔 + 平均修復時間) で表される。

② 平均故障間隔 (MTBF)

稼働中のシステムでの、ある故障の回復後から次の故障までの時間の平均。大きいほど信頼性が高い。

③ 平均修理時間 (MTTR)

故障後回復するまでの平均時間。小さいほど保全性が高い。



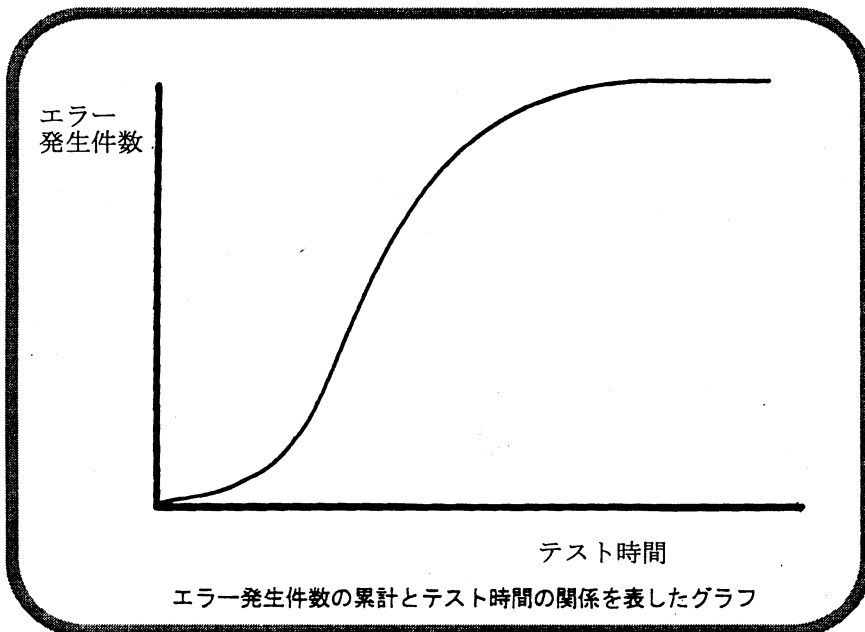
図表IV-12 稼働率

(2) 正確性

正確な処理はシステムの信用維持に欠かせない。正確性は、処理時のエラー件数、誤差などの把握で程度がわかる。総合テストやシステム移行時の障害記録、エラー分析表を活用する。

① エラー発生率

テストでは、事前の予想と実際の結果の違いがエラーとなって現れ、システムの修正を必要としたり、テスト環境を修正する必要がでてくる。テスト時間の経過に対してのエラーの発生率の累計で、テストの達成度を把握することができる。



図表IV-13 ゴンベルツ曲線

(3) 安全性

システム運用の際の安全対策を評価する。社内でコンピュータの故障時のバックアップ、回復法、再スタートの手順などを規定しておく。

第IV章のまとめ

設計したヒューマンインタフェースをテストし、入出力の動作やシステムの中での受け渡しの動作が正しく行われているか確認する。

テスト行程の流れの把握とテストの種類を選択、テストに必要なテストケースの設定、テストデータの作成、テストの実施とよりよいシステムの作り上げるための知識と技術を修得する。

テスト時においてSADは、

- ・システムを利用する立場で・・・機能性や操作性の確認をする。
 - ・・・テストのデータを設定する。
 - ・・・問題点や改善点を開発者側に伝達する。
- ・システムの開発者の立場で・・・プログラムの動きを把握する。

主要用語

テスト、単体テスト、結合テスト、運用テスト、ブラックボックステスト、ホワイトボックステスト、ボトムアップテスト、トップダウンテスト、性能テスト、負荷テスト、対抗テスト、テストケース、テストデータ、エラー発生率、稼働率、MTTR、MTBF

練習問題

問1 次の a, b, c に入る言葉を選びなさい

性能テストにおいては、、、レスポンスタイムなどが評価項目になる。負荷テストにおいては、データ量、、システム稼働時間を評価項目とする。

解答群

ア. 処理速度 イ. アクセスタイム ウ. 応答時間

問2 次の a, b に入る言葉を選びなさい。

テストの方法には 、 あり、前者はプログラムを対象とした内部構造やロジックの動きをテストする方法で、後者は入力と出力だけ明確にしてテストする方法である。

解答群

ア. ブラックボックス法 イ. ボトムアップテスト
ウ. ビッグバンテスト エ. ホワイトボックス法

問3 次の a～d をテストの実施の順番に並べ正解を選びなさい

a 結合テスト b 運用テスト c システムテスト d 単体テスト

解答群

ア. a-b-c-d
イ. a-d-c-b
ウ. d-b-a-c
エ. d-a-c-b
オ. c-b-d-a

問4 次の a～e に入る言葉を選びなさい。

テストにおいては、 または を用意し、それぞれのモジュールテストを行う。最後に各モジュールを結合して、一斉にプログラムテストを行う。

テストでは最下位のモジュールから順次モジュールテストを行っていく方法である。次に、その上位のレベルのモジュールに結合してテストしていく。下位のモジュールからテストしていくため を用意する必要がある。

テストでは最上位のモジュールから順次モジュールテストを行っていく方法である。次に、その下位のレベルのモジュールに結合してテストしていく。上位のモジュールからテストしていくため を用意する必要がある。

テストは テストと テストを組み合わせる方法である。一般的には、最上位に近いモジュールは、 テスト、最下位に近いモジュールは、 テストを用いる。

解答群

ア. スタブ イ. ボトムアップ ウ. 折衷 エ. トップダウン
オ. ビッグバン カ. ドライバ

問5 システムテストにおいて、実際の運用を想定して、ピーク時の予想最大負荷をかけ、応答時間や処理時間をテストするのはどれか選びなさい。

解答群

- ア. 機能テスト イ. 回帰テスト ウ. 負荷テスト エ. 性能テスト
カ. 回帰テスト キ. レグレッションテスト