

【品質保証力】

Excelツールの テスト手法①

2022年10月
株式会社セルネッツ

本書の範囲

開発規模に応じたテスト

システム開発ではフェーズに応じた「テスト」があります。様々な本にテストの話が出てきますが「これが正解」と言うものは提示されていません。それは、開発同様 **テストも「規模」に応じて変わる**ためです。

本資料では「Excel開発におけるテスト」を範囲として紹介します。

事後アンケートについて

1か月後に対するアプローチ

本講座では皆様の今後の活動に役立てるため「振り返り」を用意しています。
「振り返り」はアンケート方式で行いますのでご協力ください。

- ▶ 新たに習得できたこと
- ▶ 習得まで至らなかったこと
- ▶ 今後、気を付けたいこと
- ▶ 1か月後に実践できていたいこと

Section 1

[テストとは]

- ◆ テストの種類
- ◆ ツール開発の向こう側
- ◆ テストを意識した「設計」
- ◆ 品質特性図

某ECサイトで「プログラミング」の本を探すと2万書がヒットします。しかし「ソフトウェアテスト」とすると1000を切ります。開発分野の中でも大変不人気な工程で、知識としての認知度も低いです。

一方で、品質管理は「**最後の砦**」と言われる重要な部分を担っています。

「最後の砦」を制覇すべく、まずは種類を学びましょう。

ツール開発の向こう側

(山の向こう) 相手を想像する

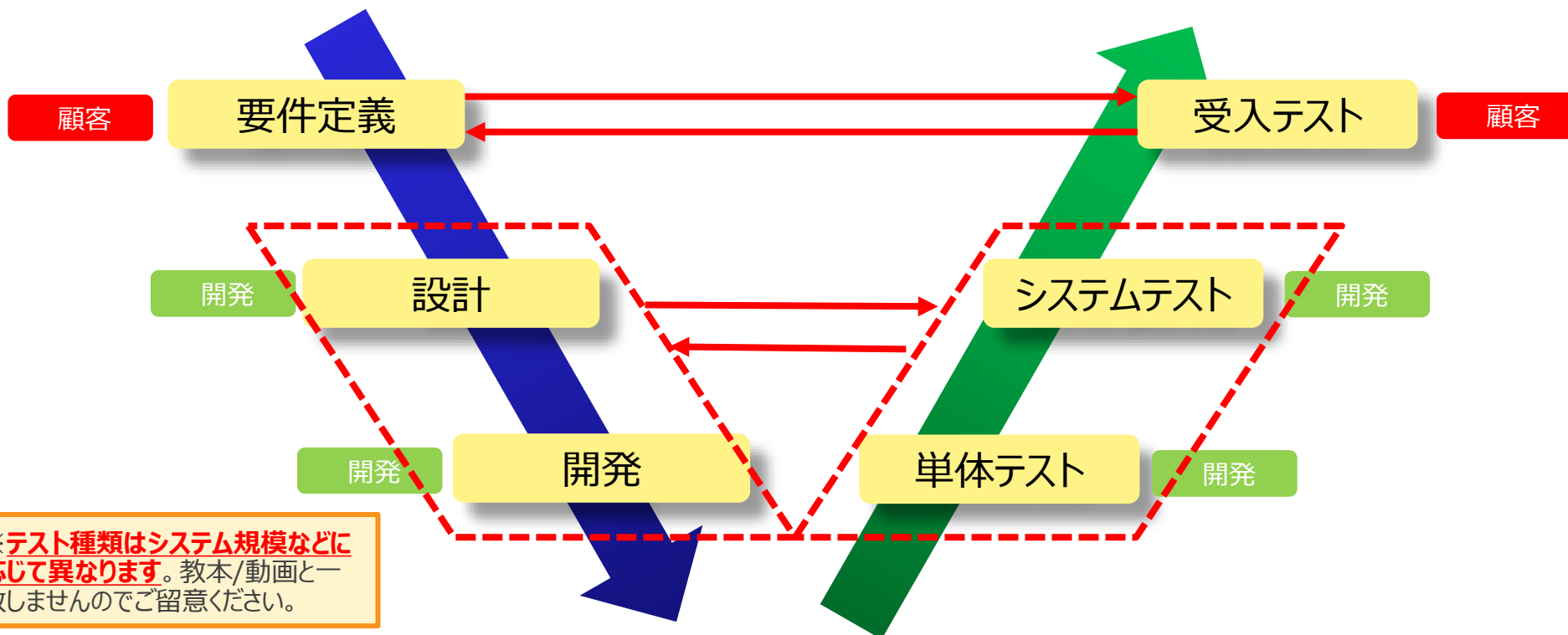


テストの種類

目的に応じたテスト（Excelツール編）

開発には「フェーズ」があり、「フェーズ」に応じて関係性が強くなる人が異なります。

「テスト」は「フェーズ」と遂になるため、テストに応じて担当する人が変わります。



テストを意識した「設計」

「逆算」による準備が必要

テスト設計は「システム設計」時点から始まっています。「**どのようにテストしたら保証できるのか？**」を意識して「システム設計」を行う必要があります。



品質特性図 (1/2)

体系化された「ソフトウェア品質」

(凡例： ■ 重要箇所)

品質特性	副特性	説明
機能性	(functionality)	必要な機能が実装されていること
	合理目的性	使用上適切な機能が提供されていること
	正確性	必要な精度で正しい結果が得られること
	相互運用性	他のシステムと正しく相互作用すること
	セキュリティ	許容されていない人が情報を見たり修正したりできないこと
	機能性/標準適合性	機能性に関する法律、規約、規格が遵守されていること
信頼性	(reliability)	機能が正しく動作し続けられること
	成熟性	故障を回避できること
	障害許容性	障害が発生した場合でも指定された機能を維持できること
	回復性	障害発生後、データの復旧を含めて正常状態に戻れること
	信頼性/標準適合性	信頼性に関する法律、規約、規格が遵守されていること
使用性	(usability)	利用者に使いやすく魅力的であること
	理解性	利用者が使い方を理解しやすいこと
	習得性	利用者が習得しやすいこと
	運用性	利用者が運用できること
	魅力性	利用者に魅力的であること
	使用性/標準適合性	使用性に関する法律、規約、規格が遵守されていること

品質特性図 (2/2)

体系化された「ソフトウェア品質」

(凡例： ■ 重要箇所)

品質特性	副特性	説明
効率性	(efficiency)	資源の量に対比して適切な性能であること
	時間効率性	応答時間、処理時間が適切であること
	資源効率性	実行時の資源量、種類が適切であること (メモリー、DISK、通信回線など)
	効率性/標準適合性	効率性に関する法律、規約、規格が遵守されていること
保守性	(maintainability)	維持、変更がしやすいこと
	解析性	欠陥の診断、故障の追求、修正箇所の識別が行いやすいこと
	変更性	修正がしやすいこと
	安定性	修正により影響を受けにくいこと
	試験性	妥当性の確認がしやすいこと
	保守性/標準適合性	保守性に関する法律、規約、規格が遵守されていること
移植性	(portability)	別環境に移しやすいこと
	環境適応性	異なる環境で修正なしに動くこと (Officeのバージョンなど)
	設置性	指定された環境に設置しやすいこと
	共存性	他の独立したソフトと動くこと
	置換性	同一環境で同じ目的のソフトと、入れ替えて使えること
	移植性/標準適合性	移植性に関する法律、規約、規格が遵守されていること

Section2

[各種テスト]

- ◆ 用語
- ◆ 単体テスト
- ◆ システムテスト
- ◆ 受入テスト

テストには各々の目的・着眼点があるため
それらを理解したうえで、テストを行う必要があります。
本章では各テストの目的・着眼点を紹介します。

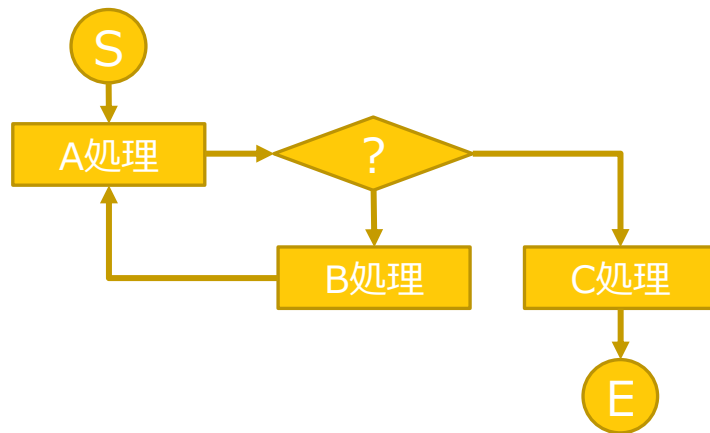
用語 (1/2)

「内部構造」に着目したテストのカテゴリ

テストには「内部構造」を意識したテスト、意識しないテストがあります。
どちらを用いたテストが必要か、テスト工程に関係するため覚えましょう。

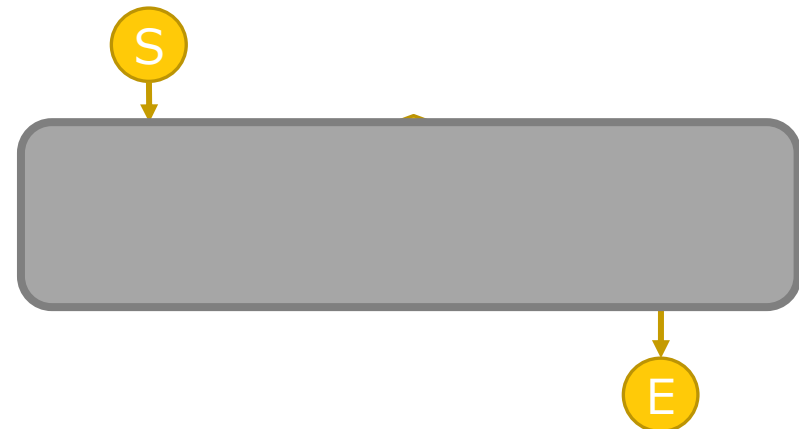
ホワイトボックステスト

論理構造に注目したテスト。処理や分岐命令、データ処理などが正しく行われるか確認する。



ブラックボックステスト

論理構造は参照せず、**入力と出力のみに注目して動作を確認するテスト**。入出力の間にある処理が見えないことから「ブラックボックス」と呼ぶ。



用語 (2/2)

「実施方法」に着目したテスト分類

日常的に出てくる用語ではありませんが、違いは理解しておきましょう。

静的テスト

動作させることなく行うテストの総称。
ソースのコードレビューがこれに該当する。

机上



動的テスト

実際に動作させて確認するテストの総称。
ツール作成時におけるテストがこれに該当する。

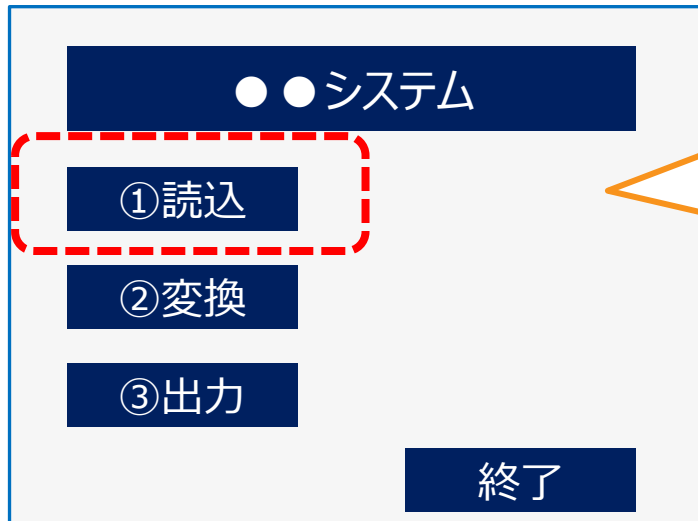
動作



単体テスト（1/3）

最小単位での確認テスト

コードとして記載した一つ一つの挙動が正常に動作することが必要です。
一つ一つの処理に着目し、つぶさに確認するのが「単体テスト」です。



POINT!!



1つの機能に着目してテストすることで、
1つ1つの条件が正しく動作することなどを保証する。
これにより、**後続する処理は「保証されたデータ」より
動作の確認を行うことができる**ようになる。

単体テスト (2/3)

単体テストの技法

ホワイトボックステスト/ブラックボックステストより、最小単位での機能を確認する。

機能確認テスト

仕様通りに動作しているか確認するテスト。
ブラックボックステストとして実施する。

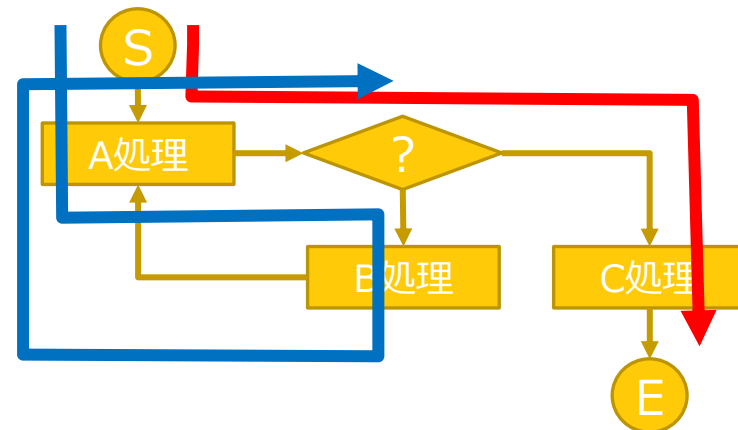
POINT!!



ツールには「**当たり前品質**」が存在する。
本テストはその部分による事項が多い。
詳しくは次のセミナーで紹介する。

制御フローテスト (カバレッジテスト)

処理や分岐命令、データ処理など論理構造に着目したテスト。ホワイトボックステストとして実施する。



単体テスト (3/3)

カバレッジテスト

カバレッジテストの実施方法の一つを紹介する。

①処理、分岐条件に
ブレークポイントを設定

```
Sub 条件に応じた値を表示してメッセージに表示する()  
    Dim セルの値 As Variant  
    Dim メッセージ As String  
    ● セルの値 = 0_11.Range("A1")  
    ● If IsDate(セルの値) = True Then  
    ●     メッセージ = "日付です."  
    ● Else  
    ●     メッセージ = "日付ではありません."  
    ● End If  
    ● MsgBox "A1の値は" & メッセージ  
End Sub
```

②処理を実行し、命令文が呼び
出されたらブレークポイントを解除

```
Sub 条件に応じた値を表示してメッセージに表示する()  
    Dim セルの値 As Variant  
    Dim メッセージ As String  
    ● セルの値 = 0_11.Range("A1")  
    ● If IsDate(セルの値) = True Then  
    ●     メッセージ = "日付です."  
    ● Else  
    ●     メッセージ = "日付ではありません."  
    ● End If  
    ● MsgBox "A1の値は" & メッセージ  
End Sub
```

POINT!!



すべての処理（条件内の処理含む）が実行できれば、ブレークポイントがすべて解除される。

単体テスト (3/3)

カバレッジテスト

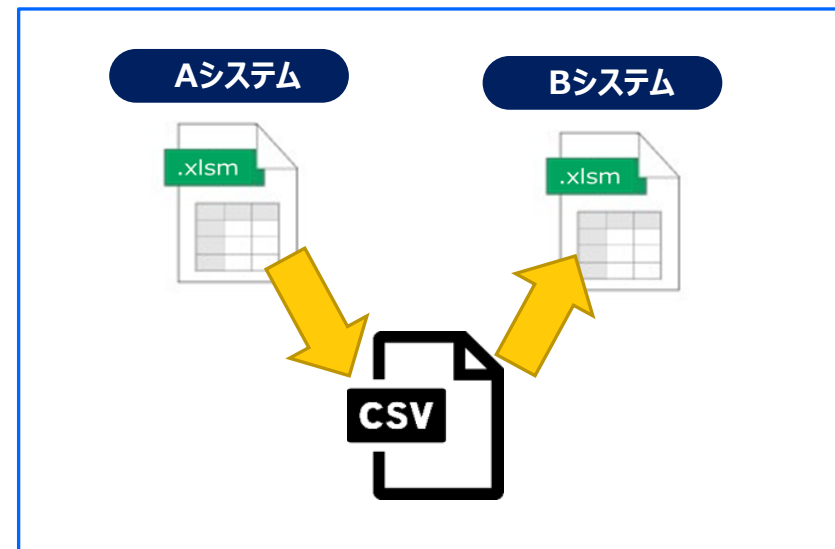
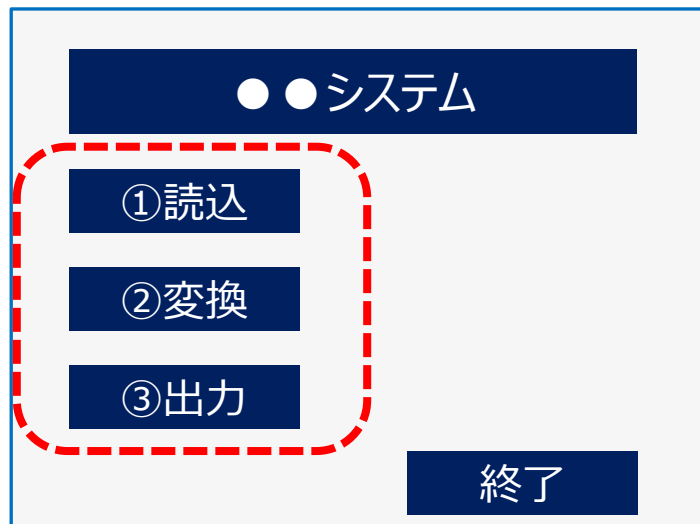
Sample

```
Sub 条件に応じた値を表示してメッセージに表示する()  
  
    Dim セルの値 As Variant  
    Dim メッセージ As String  
  
    セルの値 = 0_11.Range("A1")  
  
    If IsDate(セルの値) = True Then  
        メッセージ = "日付です。"  
    Else  
        メッセージ = "日付ではありません。"  
    End If  
  
    MsgBox "A1の値は" & メッセージ  
  
End Sub
```


結合テスト

機能間/システム間のつながりを確認する

単体では機能毎にテストを行いました。「結合テスト」では**各機能の結果が正しく結びつくか否かをテスト**します。これは機能だけでなく、システムの間をつなぐ場合もあります。



システムテスト（1/2）

高い視点に立ったテスト

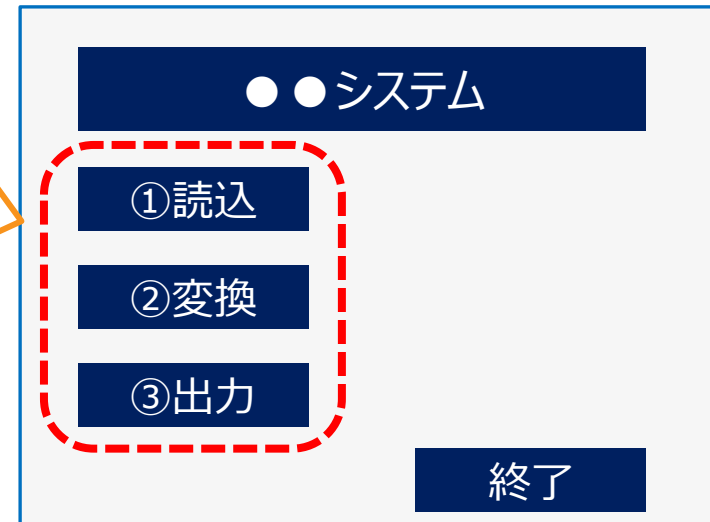
単体テストでは「正常に動作すること」を保証するため、つぶさに確認をします。見る視点が低いため「全体的に問題がないか」を確認することはできません。

「運用した際に問題がないか」を確認するために「システムテスト」を行います。

POINT!!



「運用に関連する」ものとして「**負荷テスト**」や「**環境テスト**」があります。「環境」はOfficeのバージョンやOSの違い、画面サイズの違いなどを指すため、利用者が多い場合、環境別のテストも必要となります。



システムテスト（2/2）

運用に耐え得る品質の保証

車の性能テストと同じく、ツールも負荷をかけたテストを行います。

単体テストで細かい面が網羅されていることを確認した後にテストします。

負荷テストの種類

- ▶ **性能テスト**：処理能力（レスポンスタイムなど）が仕様を満たしていることを確認するテスト
- ▶ **ロングランテスト**：長時間の連続稼働によって処理能力が稼働率に問題が生じないか確認する
- ▶ **大容量テスト**：容量の大きいデータや大量のデータを処理できることを確認するテスト
- ▶ **高頻度テスト**：一定時間内に繰り返し大量の処理を行った場合に問題が生じないことを確認する
- ▶ **ストレステスト**：極端に負荷をかけた状況下（大量データを連投する等）で動作を確認するテスト

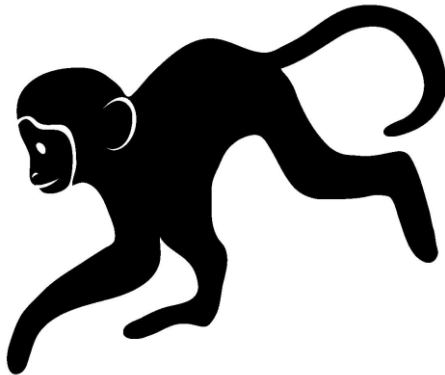
モンキーテスト（ランダムテスト）

テストケースに縛られないテスト

テストケースに沿ってテストは行います。開発者は開発に集中しているため

見落としているテストケース残されている可能性があります。そういったものを

除くために「モンキーテスト（ランダムテスト）」があります。



POINT!! 

場当たり的に行うテスト。

テスト担当者の経験則などから不具合を
あぶりだす。

受入テスト（1/3）

顧客側でしか発生しない問題がある！

「**顧客環境下で顧客が意図した通り動作するか**」を確認するテストです。

顧客環境下で起こる問題/詳細

①画面サイズ 問題

ユーザーフォーム/MAINシートのボタンが画面内に入らない。

⇒詳細を聞いたところ、営業担当者は13インチのノートPCを利用しており、本ツールは営業担当も利用するため、このままでは業務にならないとの報告を受けた。また、プルダウンにおいては経理担当者のPCでも下の方が選択できないとの連絡があった。

②ネットワーク /権限問題

ネットワークドライブにある読込用ファイルが参照できない。

⇒読込用ファイルをネットワークドライブに配置したところ、担当者であるAさんはツールから参照することができるが、Bさんは参照できないとの連絡があった。話を伺ったところ、AさんとBさんは部署が異なり、ネットワークドライブの権限より、そもそもBさんはAさんのファイルを参照できないということが判明した。

③速度問題

開発PCでは30秒なのに、顧客環境では5分かかると言われた。

⇒開発PCは性能が高いものを購入しており、メモリは32GB（16GB×2）が積まれていた。一方、顧客のPCは2GBの物を利用しており、性能に大きな差が生じていた。また、ネットワーク越しに読取用ファイルを取得しており、その部分でも時間がかかっていることが発覚した。

受入テスト（2/3）

早めの把握、早めの対応

システムテストは最後のフェーズですが、本当に最後にすると危険しかありません。

顧客環境下で起こる問題に対する対策

①画面サイズ 問題

早い時点で顧客に確認いただく。

⇒ユーザーフォームを用いる場合、操作性のほとんどはユーザーフォームに依存します。操作性が変わるとプログラムも変わるため、早めに画面サイズ等を確認しましょう。また、上記理由を伝え「**担当者だけでなく、利用する先の方数名に確認いただく**」よう、努めましょう。

②ネットワーク / 権限問題

相手のITリテラシーの理解。サーバー知識の習得。

⇒相手のITリテラシーの低さからくる問題が強いです。相手のITリテラシーを探りながら話を進めましょう。開発者側も**ネットワークドライブ特有の問題（権限やセキュリティ）**の知識が必要です。サーバーに対する知識を身につけ、問題発生時に対処できるように努めましょう。

③速度問題

時間計測および業務量を想定したテストを実施。

⇒自分のPCで問題なければ大丈夫、と言う事ではないということに意識を置きましょう。また、違いを把握するために「**計測**」できる状態を構築することが重要です。「計測」できたとしてもテスト時のデータ数が少なければ、運用時点までに違いに気付くことはできません。「**業務を想定したデータ量**」を顧客に提示してテストいただきましょう。

受入テスト (3/3)

受入テスト成功の秘訣：「丸投げしない！！」

「**相手はテスト初心者**」です。そのため「テストしてください」の一言で、正確にテストできると考えてはいけません。そのフォローも含めて「受け入れテスト」となります。

テスト内容の提示



テスト用のデータ



POINT!! 

仕様齟齬確認の面も持っています。
簡単なデータで、意図した計算結果が
できるか、確認いただきましょう。
その他に大量データテストも行い
速度の齟齬などがいないか確認しましょう。

Section3

[テスト方法]

- ◆ テストケースの洗い出し
- ◆ テストケースの作成
- ◆ デモデータの作成
- ◆ テスト実施
- ◆ テスト密度/バグ密度

ここまでテストの流れや注意点を紹介してきました。
ここからは実際のテスト方法や注意点を紹介します。

本章では「テストの方法」について紹介します。

テストケースの洗い出し

(正常系) レギュラーケース/ (異常系) イレギュラーケース

「何をテストするべきか」を明らかにしてテストすると漏れなくテストすることができます。
その際「レギュラーケース」「イレギュラーケース」に対する意識が必要です。

(正常系) レギュラー ケース

想定通りの動作を確認するためのテストケース



(異常系) イレギュラー ケース

想定外動作に対する振る舞いを確認するためのテストケース



テストケースの作成

パターンを洗い出す

「なんとなく考えながらテストをする」というのは網羅性に欠ける行為です。
 テストパターンを洗い出した上でテストを行きましょう。

テスト仕様書(format1_MENU)									
システム名		機能名			項目数	OK数	NG数	残数	
CN在庫表更新ツール		format1_MENU			25	0	0	25	
No.	分類	テスト・確認項目	テスト・確認内容	実施日	実施者	実施結果	不具合No.	備考	
1	初期表示	初期画面	[MENU]シートがアクティブになっていること。						
2			スクロール位置が先頭で、かつ、左端であること。						
3			「処理時間」「Path」「件数」が空欄であること。						
4			「処理時間」「Path」「件数」に前回実行した情報が残されていないこと。						
5			「前回保存先」に前回実行した情報が残されていること。						
6			システム名が「CN在庫表更新ツール」であること。						
7			ボタン名が適切であること。						
8		シート表示	以下のシートが表示されていること。 MENU、11D、12D、13D、14D、21対象製品、						
9			以下のシートが表示されていないこと。 80TMP (※「非表示」にも表示されない)						
10		シート保護	以下のシートにシート保護がかけられていること。 MENU						
11			以下のシート以外にシート保護がかけられていないこと。						

【カレンダー更新パターン】									
No.	チェック対象	対象項目	出力項目/条件	結果	実施日	実施者	実施	不具合	備考
1	カレンダー更新	基準日 (月始めの週)	月曜日 (2021年5月3日)	2W 2021年5月3日を先頭に1~6に日付 (計算式) がセットされる。					
2			2W	3W 2021年5月10日を先頭に1~6に日付 (計算式) がセットされる。					
3			3W	4W 2021年5月17日を先頭に1~6に日付 (計算式) がセットされる。					
4			4W	5W 2021年5月24日を先頭に1~6に日付 (計算式) がセットされる。					
5			5W	6W 2021年5月31日がセットされる。					
6			6W	7W 2021年6月7日がセットされる。					
7			7W	8W 2021年6月14日がセットされる。					
8			8W	9W 2021年6月21日がセットされる。					
9			9W	3M 2021年7月1日がセットされる。					
10			3M	4M 2021年8月1日がセットされる。					
11			4M	5M 2021年9月1日がセットされる。					
12			5M	6M 2021年10月1日がセットされる。					
13			6M	2W 2021年5月6日を先頭に1~6に日付 (計算式) がセットされる。					
14			2W (2021年5月6日)	3W 2021年5月17日を先頭に1~6に日付 (計算式) がセットされる。					

デモデータの作成

デモデータ作成サイトの活用

テストでは「大容量」でも正しく動作することを確認する必要があります。「大容量」データは顧客から提供されない場合があります、その際は**自分で作成しなくて**はいけません。



参考：TM-WebTools

テストデータ生成ツールの使い方

1. 列名と出力されるデータ形式を入力します。必要な数だけ列数を追加して入力してください。
2. 出力行数と出力フォーマットを入力します。
3. テストデータ生成ボタンを押下することで、テスト用のダミーデータが表示されます。
4. ダウンロードリンクを押下することで、ファイルとしてダウンロードできます。（※IEでは不安定です）

出力行数： 出力フォーマット：

#	列名	データ形式	削除
1	<input type="text" value="氏名"/>	姓名(半角スペース区切り) <input type="text"/>	<input type="button" value="削除"/>
2	<input type="text" value="生年月日"/>	生年月日 <input type="text"/>	<input type="button" value="削除"/>

氏名,生年月日
中平 涼香,2002/8/6
湯野 春香,1994/8/2
成瀬 良平,1983/1/6
宮部 達夫,1980/6/9

POINT!!



自動的にデモデータを作成するサイトがあります。こちらなどを活用し、自分でデモデータを作成し、テストに使用してください。

テスト実施

必要環境は事前に確認！

テストはツール/データだけでなく設備も必要になります。ツールの要件に応じて異なりますので、事前に必要な設備を確認しておきましょう。

ネットワークドライブの利用



排他制御の確認

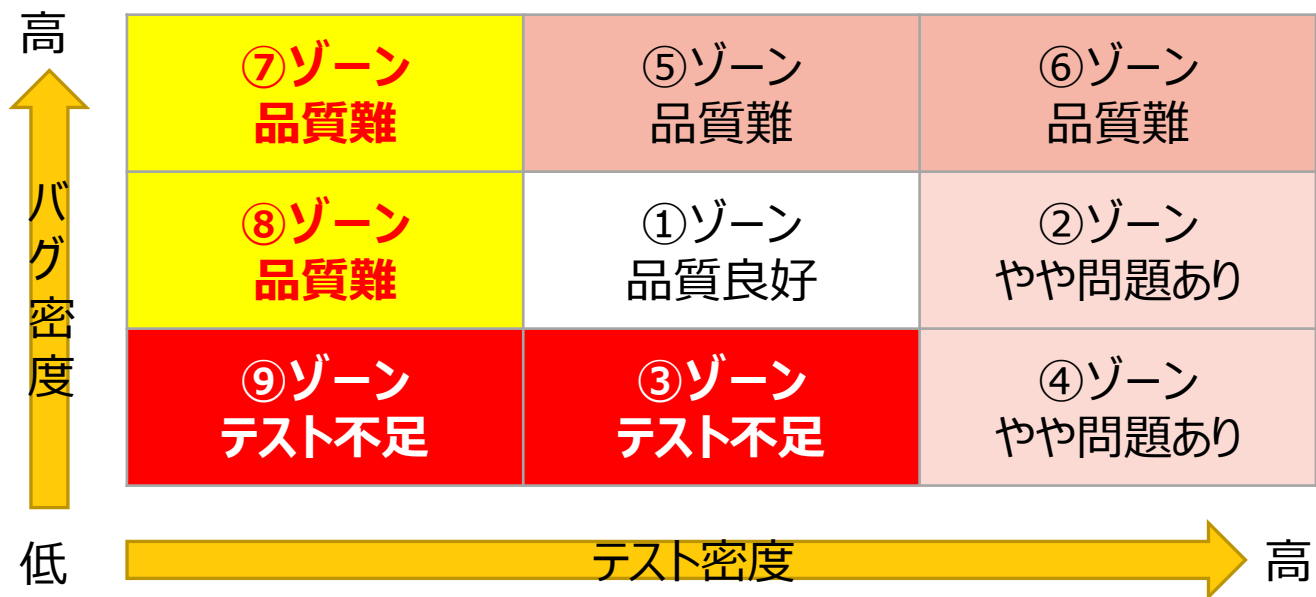


テスト密度/バグ密度

不具合がないのが「不具合」

開発では「バグ密度」を用いた品質判断が成されており、以下表で表されます。

テスト密度：機能数やプログラムコード数よりテスト密度を測る
バグ密度：機能数やプログラムコード数より不具合の発生度を測る



(参考：[Shift様\(バグ密度とは?\)](#))

テスト密度/バグ密度

①品質良好（テスト件数OK、不具合も検出されている）



(参考 : [Shift様\(バグ密度とは?\)](#))

テスト密度/バグ密度

②④ やや問題あり (テストしている割に、不具合が出ていない)



(参考 : [Shift様\(バグ密度とは?\)](#))

テスト密度/バグ密度

⑤⑥品質難（テストは一定以上しているが不具合も多い）



(参考 : [Shift様\(バグ密度とは?\)](#))

テスト密度/バグ密度

⑦⑧品質難（テストが少ない。しかし不具合は多い）



(参考 : Shift様(バグ密度とは?))

テスト密度/バグ密度

⑨⑩テスト不足（テストが少なく不具合があまり出ていない）



(参考 : [Shift様\(バグ密度とは?\)](#))

目的を理解したテストが重要

本資料では「**目的に応じたテスト**」についてを紹介しました。

必要な時点で必要な事項を確認しないと運用した際に不具合が発生します。

お客様に迷惑をかけないために、プロ品質なツールを提供しましょう。



