

第 39 年度 研究コース 3 (ソフトウェアテスト)

付録 1 アンケート

【アンケートの目的】

ベテランの視点を意識し、研究員の経験からテスト観点を決めてマトリックスにまとめました。このマトリックスを見て、テストが思いつくかどうかを確認するためにアンケートをお願いいたします。

【アンケート】※該当する箇所に“○”または記入をお願いいたします。

開発/テスト経験年数は何年ですか? 「 _____ 」

あなたの立場は? 「 テストマネージャ ・ プログラマー ・ テスト担当者 ・ その他 (_____) 」

【1】 資料 (マトリックス) から、これまで考え付かなかったテストが思いつきそうですか?

「 けっこう思いつきそう ・ 少し思いつきそう ・ 思いつかない 」

【2】 資料 (マトリックス) に追加したほうがよいベテラン特有のテスト観点があれば教えてください。

※設計書/仕様書に記載されているようなテスト観点は除きます。

「 _____ 」

【3】 資料 (マトリックス) のボリューム感はどうでしょうか?

「 多すぎる ・ 多い ・ ふつう ・ 少ない ・ やや少ない 」

【4】 自身で資料 (マトリックス) を使ってみたいですか?

「 使いたい ・ 使いたくない ・ どちらでもない 」

【5】 テストマネージャの立場の方向け : プロジェクトに適用できそうですか?

「 適用できそう ・ 適用できなそう ・ わからない 」

【6】 テストマネージャの立場の方向け : ご意見があればご記入をお願いします。

「 _____ 」

【7】 本アンケートを通しての御意見ををお願いします

「 _____ 」

付録 2 アンケート結果

【回答者数: キャリア別】

新人 5 名・中堅 8 名・ベテラン 37 名

【回答者数: 立場 (複数選択可)】

PM/PL16 名・プログラマー7 名・テスト担当者/テスト担当者 65 名

【1】

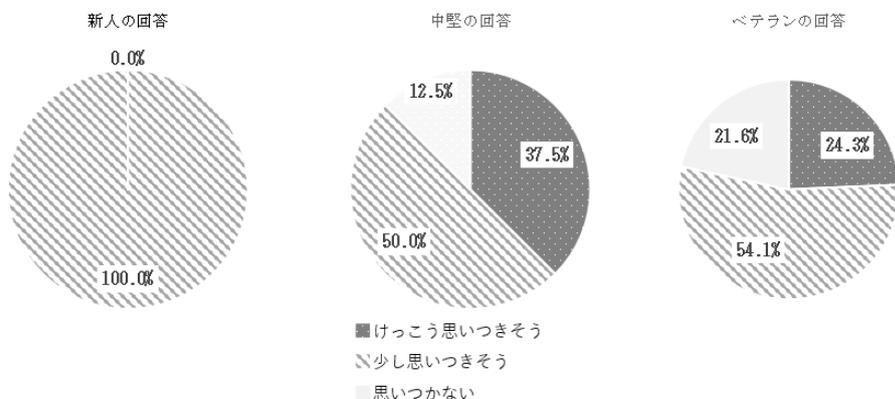


図 1 「アンケート【1】の結果」

第 39 年度 研究コース 3 (ソフトウェアテスト)

【2】

【機能適合性】

- ・ 0 件データ入力時の動作確認
- ・ 初期化漏れの確認の追加
- ・ 複数エラーの発生後のシステムの振る舞い
- ・ 旧 Ver で作成したデータに対する動作確認
- ・ OS や開発環境が上がった場合、動作リプレイスなので作り直す場合、既存機能と異なる部分がないか
- ・ パラメータの複合

【性能効率性】

- ・ データの削除、更新、追加の各パターンの性能テスト
- ・ 日次・週次・月次・年次処理のサイクルでの確認
- ・ 複数イベントの処理や、繰り返しでのメモリチェック

【使用性】

- ・ 条件にもよるがウィンドウサイズの変更、ファンクションキーの押下、テキストボックスにコピーした文字列を貼り付け
- ・ エラー時のメッセージの妥当性 (エラー箇所が特定できること、エラーの原因を特定できること、対処方法を判断できること)
- ・ システム全体の統一感 (デザイン、メッセージ、操作性、エラーチェック等、どこ機能に触っても違和感がないこと)
- ・ OS 仕様に基づく操作 (ショートカット、プロンプト経由等)
- ・ 予約文字や、先頭/末尾のスペースなど文字バリエーション

【信頼性】

- ・ 非同期処理間のタイミングに着目した観点
- ・ 結果、データ確認ができるものが複数あるならば、すべてで確認を行う
- ・ 異なるログが出たならば、結果に関係なく内容を確認する
- ・ 実際には難しいがタイミングテスト
在庫減算→在庫移動などの連携更新手順が逆に来た場合 (メッセージキュー) 案件
- ・ 提供初期状態での動作や不足提供物 (最初から不足している場合、異常終了するようなファイルやフォルダを提供しているか)
- ・ 正常 (想定) とは異なる操作順
- ・ 特定のタイミングでの操作
- ・ 実現可能操作の洗い出し
- ・ 他ソフトとの同居
- ・ 投入データの生成元
- ・ 並列/排他処理の整理
- ・ 対向先のパラメータと合わせた観点
- ・ 特定のタイミングでのイレギュラーデータの投入
- ・ 一つの異常データが、他の表示等影響を及ぼさないか

【セキュリティ】

- ・ セキュリティとして過剰な情報提供が行われていないか

【その他】

- ・ 開発を他者に依頼した場合に、処理ロジック・他プロセスとの整合性が合っていないのではないかと、中身を見て思う事があり、そのような場合にシステム全体の処理として正しいかパターンテストを行い確認
- ・ 類似製品との比較
- ・ 拡張子の改ざん

【3】

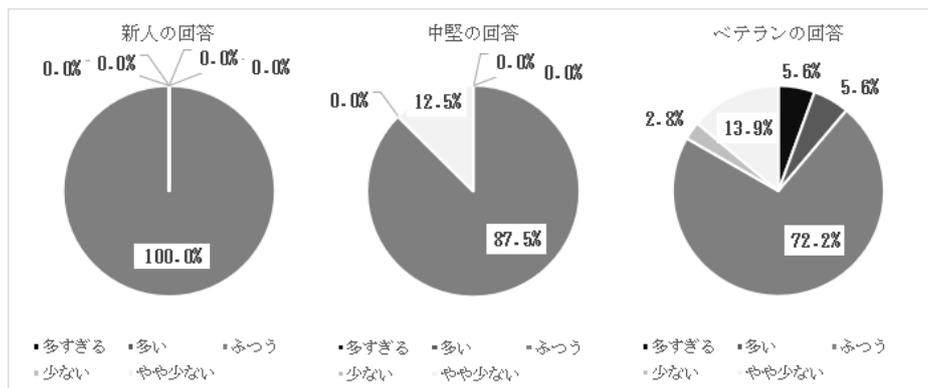


図 2 「アンケート【3】の結果」

【4】

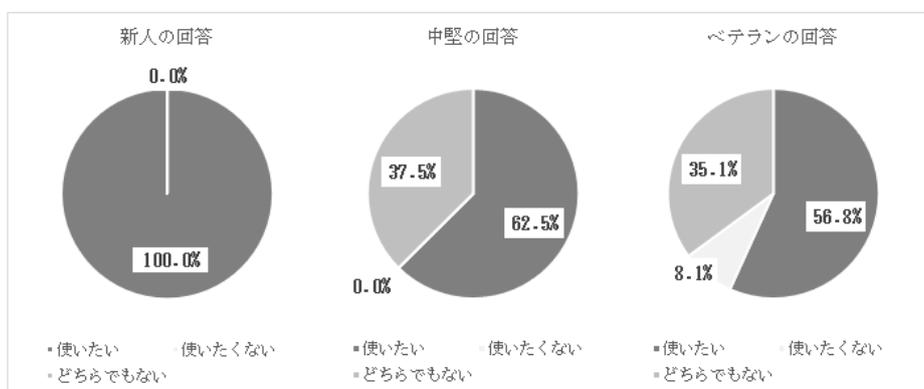


図 3 「アンケート【4】の結果」

【5】

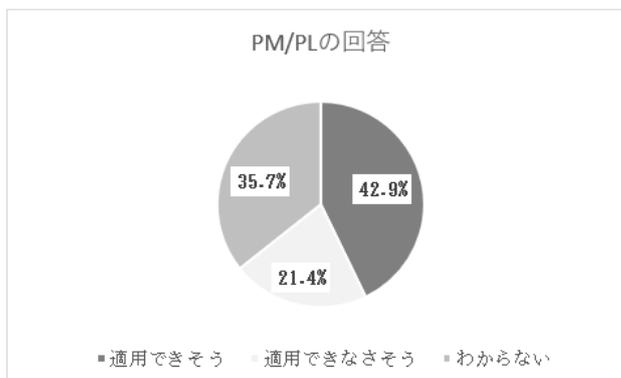


図 4 「アンケート【5】の結果」

【6】

「適用できそう」の理由

- ・担当者のセルフチェック観点として利用できそう。
- ・ベテランは製品知識が豊富であり、過去の類似製品のバグの経験があるのでその点において設計書や仕様書等のインプット以外の観点があるとだと思います。
- ・チーム体制次第ではあると思いますが、パートナーに委託しているテーマではパートナーがこういったテスト観点を理解し適用できるかどうかのポイントになると思います。
- ・C23 セルのシステムの動作に、違和感があった時のベテラン視点の項目が紐づいていないような気がします。

第 39 年度 研究コース 3 (ソフトウェアテスト)

・資料が少し見にくいです。若手からどのような回答を期待しているのかよく読み取れない。

・探索的テストのテストチャーターでベテランの視点を取り入れるのは良いと思います。逆に全くの素人だからこそあり得ない操作や組合せをつくって異常を起こすケースって市場障害から傾向出せたりすると面白いマトリックスができるかもしれないですね。

「適用できなそう」の理由

- ・実効性のありそうな部分は既に実施しており、効果が薄そう。
- ・新規に開発を実施する場合は有効だと思うが、複数案件を行う上では、すべての項目はテストが難しいのでコンポーネント開発環境で吸収したほうが良いと思う。
- ・若手をターゲットとした場合、例えば「重いデータ」のような主観的な表現だと、結局意味のないテストをするだけになるリスクがある。そういった主観部分がベテランの勘所でもあると思うのでそれを具体的に表現できた方が良いと思います。

「わからない」の理由

- ・使った方が良いと思うが、工数が合わない。
- ・テストはやりすぎて損ということはないと思うので、余裕があればやった方が良い
- ・工事種別に応じて項目をカスタマイズして欲しい。
- ・テストを実施している時間とリリース後トラブル件数に相関性は無いということは何となく認識はしているが。理想的な試験方案を探すところからやってみては？
- ・機能テスト+探索的テストなら効果があると思うが、探索的テストのみはちょっと不安。
- ・新規開発では適用し確認する事が大切。現状の見積では工数・日数が含まれていないので見積段階からテストを計画したうえで実施する事がよい。リプレース案件では、テスト工数・日数が少ない状況が多いため適用が難しい。

【7】

- ・テストチャートを使ってみたいですが、もう少し手軽に使えるもの（入りやすいもの）があるとうれしいです。
- ・周りのベテランの方は以下のようなことを想定していることが多い印象です
 - 製品が発売されてから具体的にユーザが使い始めるところをイメージする
 - 過去製品で発生したバグとテスト対象の類似点からバグを推測する
 - OS やミドルウェアがテスト対象に与える影響を考える
 - 見つかったバグの内容や対策の内容からさらに想定される影響を考察する
- ・また、マトリックスには色々な観点が盛り込まれていて、利用する側が場面に合わせて選んだりアイデアを想起させるような使い方を想定しています。なので情報量は多ければ多い方が良いかと思います。
- ・【2】の追加した方が良い観点にいろいろ記載してみましたが、汎用化するのが難しいものがあるかもしれません。
- ・探索的テストはやみくもに触るものではなく目的を持たせてやるものという認識なので実施する場合は何らかのテストチャーターは使用していると思います。
- ・アンケート対象のようなマトリックスではなく、単に機能一覧だったり、過去障害の一覧だったりする場合がありますよね。

付録 3 テストパターンマトリックスの要素に取り込んだ 39 件

【機能適合性】

- ・評価対象システムと対向システムの間で境界値パラメータの受け渡しを行い、対向システムが境界値パラメータを正しく活用できることを確認する
- ・正規の手順とは異なる機能の使用手順を行い、手順の過不足によりシステムが故障しないことを確認する
- ・評価対象システムに対してパラメータの矛盾や読み捨て対象となるデータの投入を行い、想定外のパラメータを受け取った際に期待するエラーメッセージが表示される（エラーハンドリングができています）ことを確認する
- ・全角/半角、接頭接尾のスペース、拡張子の小文字/大文字、サロゲートペア等の利用可能なファイルの命名規則に従ってユーザやシステム側で使用される文字の組み合わせを行い、特殊なデータが使用されてもシステムが動作することを確認する
- ・検索機能に対し、検索条件の組み合わせ、組み合わせ方法（AND や OR 等）、部分一致の位置変更、検索する文字列の過不足等を行い、条件に一致するデータが取得できること（または合致しないデータが取得されないこと）を確認する
- ・正常に登録できるデータとエラーになるデータが混在した状態でデータ登録を行い、エラー発生時にシステムの停止や続行が正しいタイミングで行われるかを確認する
- ・拡張子を改ざんしたデータを使用し、想定外のデータが登録された場合にシステムが異常な状態にならないことを確認する
- ・エラーを発生させてから復帰するまでの操作を行い、その後の正常な処理（後続データ等）に影響がないことを確認する
- ・エラーを発生させてから復帰するまでの操作を行い、その後にエラーが発生した箇所から他のデータ（関係データ等）に影響がないことを確認する
- ・連携システム間でエラーを発生させてシステムを停止してから復帰するまでの操作を行い、その後の正常な処理で内部カウンタ値の不正による誤課金や処理停止等が発生しないことを確認する
- ・使用するボタン等に対して Tab キーでフォーカスを合わせてから Enter キーの長押しやボタン連打を行い、システムが異常な状態にならないことを確認する
- ・ボタンを複数回連打し、ウィンドウやメッセージが複数起動しないことやシステムに負荷がかかっていないことを確認する”
- ・旧 Ver のソフトや外部連携先とデータの送受信を行い、システムの互換性や相互連携性が保持されていること（データの継承/取扱いに問題がないこと）を確認する
- ・評価対象システムと類似システム（同様の機能やデータを扱っているシステム）を共存させた状態で操作を行い、動作中のシステムやサービスが正しく動作を続けられること（競合により停止しないこと）を確認する
- ・データの生成元（または製造元）の種類を変更して様々なデータを使用し、汎用的な条件で記載している動作環境で正しく動作することを確認する
- ・現在動作中のユーザ権限を別の権限に変更し、評価対象システムが適切なタイミングで権限を失った動作や処理を止められることを確認する

【性能効率性】

- ・極端に大きなファイルサイズのファイルを使用した際にシステムが予期せぬエラーやタイムアウト等を起こさないことを確認する
- ・特定処理に影響を及ぼすような特徴的なデータを使用し、システムが不得手とするデータが投入された際に誤作動や障害を起こさないことを確認する
- ・複数環境から同時に評価対象システムに負荷をかけ、並列処理が正しく動作すること（リソースの解放、リソースの割り当て等が行えること）を確認する

【使用性】

- ・複数のインプット処理を同時実行後に処理中断や再開を繰り返し行い、割り込み処理が円滑に行えることを確認する
- ・登録に時間のかかるデータを大量に使用し、登録が完了するまでの間にユーザが進捗状況を正しく把握できることや処理に時間がかかりすぎていないことを確認する
- ・エラーを発生させてから復帰するまでの操作をメッセージやマニュアルに基づいて行い、ユーザが本当にエラーから復帰できることを確認する
- ・マニュアル等を利用せずに直感的な操作を行い、ユーザが直感的に動作完了まで到達できることを確認する
- ・新規の運用に合わせた様々なデータの登録年月日とデータの使用状態（未使用、使用中、使用済み、削除済み等）を用意し、保存したデータが過剰に蓄積されていないことを確認する
- ・新規の運用に沿った操作を行って各種メッセージを表示させ、表示されたメッセージに妥当性や統一性があることを確認する
- ・評価対象システムを動作途中の様々なタイミングで停止し、その際にユーザが状況を把握できること（今何が起きているかを認識できること）を確認する
- ・新規の運用に沿った操作をマウスなし（ショートカット等）で行い、時短を求めているユーザの利便性の観点から要求が実現できることを確認する”
- ・評価対象システムにログインした状態で放置し、別の端末から他ユーザ等が同じユーザ ID（または異なるユーザ ID）を使用してログインや処理を適切に実施できることを確認する
- ・先行 Ver のシステムや前身となるシステムや、より売れている製品との比較を行い、ユーザが使いにくさを感じていないことを確認する

【信頼性】

- ・複数の異なるエラーを同時に発生させ、停止すべき機能が正しく停止しているか（他に影響していないか）を確認する
- ・システムを操作中に想定外の事態（停電やネットワーク切断）を発生させてから復帰させ、その後に作業の継続（または中断）が正しくできることを確認する
- ・評価対象とは別のシステムから割り込み操作を行い、システムが安定稼働できることを確認する（別のシステム例：OS 等）
- ・OS の変更やモニタの解像度の変更を行い、評価対象システムの表示（警告やエラーの表示等を含む）や機能が正しく動作することを確認する”
- ・処理中のシステムに対してキューの追加を行い、機能が想定している順序/内容で処理できていることを確認する
- ・時刻同期している連携システムの一方向のシステム時間のみを変更し、システム間の同期が正しく取れていることを確認する”
- ・仕様未定義の未制限因子（ウィンドウ数、ブラウザ数等）を複数起動（限界値まで）させてそれぞれの画面から操作を行い、排他処理や競合処理により不整合が起きないことを確認する

【セキュリティ】

- ・データ入力箇所に対して HTML 文やコードの入力を行い、システムの乗っ取りや、暴走を防いでいることを確認する
- ・エラーを発生させ、その後に開示される情報（エラーメッセージやログ）に過剰な情報が表示されていないことを確認する
- ・URL やコード内に表示されている文字列の確認を行い、開発名称や個人を特定できる情報等が入っていないこと（機密や機微な情報に容易にアクセスできるようになっていないこと）を確認する

【付録4】

表1 テストパターンマトリックス (アンケート実施前)

		機能適合性	性能効率性	使用性	信頼性	セキュリティ	
		<ul style="list-style-type: none"> システムが正しく動作しているか 利用者の作業・目的に対して機能は適切に働いているか 	<ul style="list-style-type: none"> システムの処理能力は利用者の要求を満たせるか 使用するリソース (CPU、メモリ、通信量) やデータ許容量 (HDD、DB、データストレージ容量、クエリ回数、アクセス頻度) は十分か 	<ul style="list-style-type: none"> 直感的に操作可能か 誤操作時にシステムは適切に対応できるか 	<ul style="list-style-type: none"> システムは容易にクラッシュしないか システムがバックアップからリカバリ後に正常動作するか 	<ul style="list-style-type: none"> システムは改ざんや偽装がされないか システムは機密情報を漏洩していないか 	
テスト方針 「ベテランの視点をテストに取り入れる」		○	○	○	○	○	
ベテランの視点	仕様になり、仕様が曖昧な点	該当 テスト観点：ここで何を確認するかの簡単な説明	○ ・仕様を確認する意味で限界値、境界値、Null値などを使ったテストをする	○ ・重いデータ、苦手なデータを処理させる	○ ・データ一部キャンセル状態で正常に完了できるか (画面からユーザーが判断できるか)	○ ・複数エラーを同時に発生させる	○ ・サニタイズ処理、SQLインジェクション対策の確認 ・仕様外のデータが混入した際にシステムが改算されないことを確認する
	データ登録処理が途中で止まった	該当 テスト観点：ここで何を確認するかの簡単な説明	○ ・正常に登録されるデータと異常となるデータが混在するパターンを確認する ・データの投入順序を変えて確認する ・サポート外データの投入	○ ・【負荷テスト】本番想定最大のデータ量を投入し、メモリリークが発生していないか確認する	○ ・データ登録処理中にアプリがフリーズした際に利用者は適切な対応が可能かどうか確認する	○ ・一部分でエラーが発生させた状態でシステムとして動くか確認する (機能の経過) ・【連続運転テスト】メモリリークが発生していないか確認する	○ ・仕様外のデータが混入してもシステムをすぐに復旧できることを確認する
	エラー発生後のシステムの振る舞い	該当 テスト観点：ここで何を確認するかの簡単な説明	○ ・エラー発生後にシステム/機能が正しく復旧できるか確認する ・エラー発生後にデータの整合性が保たれていることを確認する	○ ・エラー発生後に性能が劣化しないことを確認する	○ ・エラー発生後に利用者は適切な対応 (エラーを正して再実行すること) が可能かどうか確認する	○ ・【エラーの復旧手段を確認する】 ・作業中の停電による強制終了後の再起動で作業継続できるか確認する ・作業中の抜線等の異常発生時の復旧確認、作業継続できるか確認する ・システムの停止 (一部、全部の強制停止) 後に作業再開できるか確認する	○ ・エラー発生後に表示されるメッセージに内部情報 (アカウント名、サーバー名、データベース名など) が表示されていないか確認する
	システムの動作に違和感を感じた	該当 テスト観点：ここで何を確認するかの簡単な説明	○ ・操作を連続して確認する (同じ機能や画面遷移の操作等) ・繰り返しの操作が適切に処理されているか確認する (Enterキーの押しっぱなしやボタン連打等)	○ ・繰り返し処理ができることを確認する (メモリ/ディスク容量等)	○ ・ユーザー要求の実現手段/フローを確認する (直感的に操作できなかった場合)	○ ・スマートデバイスでOS割り込み (着信、スリープ、端末回転) を確認する	○ ・URLが暗号化 (https処理) されているか確認する ・URLに内部情報が含まれていないか確認する
	【新規/追加機能】 新たな操作	該当 テスト観点：ここで何を確認するかの簡単な説明	○ ・既存機能どううまく連携できるか確認する ・旧Ver/既存機能のデータが投入できるか確認する	○ ・複数の端末から同じ機能を連続実行しても問題ないか確認する ・リソース枯渇を考慮 (メモリ/CPU) した確認をする	○ ・保存した項目の削除タイミングを意識した確認をする (データやログの保持期限が切れたら正しく処理されること) ・エラーメッセージの妥当性や統一性を確認する ・作業をキャンセル/中断フローを確認する	○ ・処理中に関連する機能が使用できることを確認する ・処理中に連続してデータを投入できることを確認する ・設備との連動処理における異常状態から復旧できる事を確認する ・スケジューラ (サービスやバッチ) の起動時間が正しく設定されているか確認する ・システム時刻が正しく同期されているか確認する	N/A
	その他 (他のシチュエーションに当てはまらない障害)	該当 テスト観点：ここで何を確認するかの簡単な説明	○ ・処理中にアカウント情報を更新しても問題がないか確認する ・複数機能を同時に実行しても問題ないか確認する ・古いプログラムを利用しているところに16bit問題がないか確認する	○ ・時間経過の資源消費量をチェックしてメモリリークが発生していないことを確認する	○ ・タイムアウト発生時/発生後の動作を確認する	○ ・仕様で定義されていない未制限因数の限界値を確認する (画面数、窓数など)	N/A

○：該当、-：非該当、N/A：適用外、●：テスト済み

【付録6】

表2 テストパターンマトリックス (アンケート実施後)

		機能適合性	性能効率性	使用性	信頼性	セキュリティ
バグ発見時の状況	仕様として定義されなかった(または仕様が無意味だった)	<ul style="list-style-type: none"> 各機能が要件どおりに動作しているか 利用者の作業、目的に対して機能は適切に働いているか 	<ul style="list-style-type: none"> システムの処理能力は利用者の要求を満たせるか 使用するリソース(CPU、メモリ、通信量)やデータ許容量(HDD、DB、データストレージ容量、クエリ回数、アクセス頻度)は十分か 	<ul style="list-style-type: none"> 直感的に操作可能か 誤操作時にシステムは適切に対応できるか 	<ul style="list-style-type: none"> システムは容易にクラッシュしないか システムがバックアップからリカバリ後に正常動作するか 	<ul style="list-style-type: none"> システムは改ざんや偽装がされないか システムは機密情報を漏洩していないか
	データ登録処理が途中で止まった	<ul style="list-style-type: none"> 評価対象システムと対向システムの間で境界値パラメータの受け渡しを行い、対向システムが境界値パラメータを正しく活用できることを確認する 正規の手順とは異なる機能の使用手順を行い、手順の過不足によりシステムが故障しないことを確認する 評価対象システムに対してパラメータの矛盾や読み捨て対象となるデータの投入を行い、想定外のパラメータを受け取った際に期待するエラーメッセージが表示される(エラーハンドリングができていない)ことを確認する 全角/半角、接頭接尾のスペース、拡張子の欠文字/小文字、サポートタイプ等の利用可能なファイルの命名規則に従ってユーザやシステム側で使用される文字の組み合わせを行い、特殊なデータが使用されてもシステムが動作することを確認する 検索機能に対し、検索条件の組み合わせ、組み合わせ方法(ANDやOR等)、部分一致の位置変更、検索する文字列の過不足等を行い、条件に一致するデータが取得できること(または合致しないデータが取得されないこと)を確認する 	<ul style="list-style-type: none"> 極端に大きなファイルサイズのファイルを使用した際にシステムが予期せぬエラーやタイムアウト等を起こさないことを確認する(例:市場で利用される大きなファイルの2倍、3倍、10倍等のファイル) 特定処理に影響を及ぼすような特徴的なデータを使用し、システムが不得手とするデータが投入された際に誤作動や障害を起こさないことを確認する(例:顔検知処理に双子の写真データ、描画処理に複数レイヤのデータ等) 	<ul style="list-style-type: none"> 複数のインプット処理を同時実行後に処理中断や再開を繰り返して、割り込み処理が円滑に行えることを確認する 	<ul style="list-style-type: none"> 複数の異なるエラーを同時に発生させ、停止すべき機能が正しく停止しているか(他に影響していないか)を確認する 	<ul style="list-style-type: none"> データ入力箇所に対してHTMLやコードの入力を行い、システムの乗っ取りや、暴走を防いでいることを確認する
	エラー発生後のシステムの振る舞いを想定していなかった	<ul style="list-style-type: none"> エラーを発生させてから復帰するまでの操作を行い、その後の正常な処理(後続データ等)に影響がないことを確認する エラーを発生させてから復帰するまでの操作を行い、その後エラーが発生した箇所から他のデータ(関係データ等)に影響がないことを確認する 連携システム間でエラーを発生させてシステムを停止してから復帰するまでの操作を行い、その後の正常な処理で内部カウンタ値の不正による誤演算や処理停止等が発生しないことを確認する 		<ul style="list-style-type: none"> エラーを発生させてから復帰するまでの操作を行い、ユーザが直感的に動作してまで到達できることを確認する 	<ul style="list-style-type: none"> システムを操作中に想定外の事象(停電やネットワーク切断)が発生させてから復帰させ、その後作業の継続(または中断)が正しくできることを確認する 	<ul style="list-style-type: none"> エラーを発生させ、その後に表示される情報(エラーメッセージやログ)に過剰な情報が表示されていないことを確認する(例:エラー箇所を特定できる情報やユーザの個人情報等からハッキングが容易ではないこと)
	操作していたシステムが予想外の動作をした	<ul style="list-style-type: none"> 使用するボタン等に対してTabキーでフォーカスを含わせてからEnterキーの長押しやボタン連打を行い、システムが異常な状態にならないことを確認する(例:特定のタイミングで処理の重複が発生しないこと、処理のスキップが発生しないこと、システムがダウンしないこと) ボタンを数回連打し、ウィンドウやメッセージが複数起動しないことやシステムに負荷がかかっていることを確認する 		<ul style="list-style-type: none"> マニュアル等を利用せずに直感的な操作を行い、ユーザが直感的に動作してまで到達できることを確認する 	<ul style="list-style-type: none"> システムを操作中に想定外の事象(停電やネットワーク切断)が発生させてから復帰させ、その後作業の継続(または中断)が正しくできることを確認する OSの変更やモニタの解像度の変更を行い、評価対象システムの表示(警告やエラーの表示等を含む)や機能が正しく動作することを確認する 	<ul style="list-style-type: none"> URLやコード内に表示されている文字列の確認を行い、開発名称や個人を特定できる情報等が入っていないこと(機密や機微な情報が容易にアクセスできるようにしていないこと)を確認する
	新規機能または追加機能に関連した新たな操作を想定できていなかった	<ul style="list-style-type: none"> 旧Verのソフトや外部連携先とデータの送受信を行い、システムの互換性や相互運用性が保持されていること(データの継承/取扱いに問題がないこと)を確認する 評価対象システムと類似システム(同様の機能やデータを扱っているシステム)を共存させた状態で操作を行い、動作中のシステムやサービスが正しく動作を続けられること(競合により停止しないこと)を確認する データの生成元(または製造元)の種類を変更して様々なデータを使用し、汎用的な条件で記載している動作環境で正しく動作することを確認する(例:使用前のファイル名やデータサイズを使用後と比較してデータが壊れていないことを確認する等) 	<ul style="list-style-type: none"> 複数環境から同時に評価対象システムに負荷をかけ、並列処理が正しく動作すること(リソースの解放、リソースの割り当て等が行えること)を確認する 	<ul style="list-style-type: none"> 新規の運用に合わせた様々なデータの登録年月日とデータの使用状態(未使用、使用中、使用済み、削除済み等)を用意し、保存したデータが過剰に蓄積されていないことを確認する(例:保存したデータの保持期限や削除契機や削除条件等を確認する) 新規の運用に沿った操作を行って各種メッセージを表示させ、表示されたメッセージに妥当性や統一性があることを確認する(例:エラーメッセージの内容が新規の運用に対して妥当であることや表現が統一されていること) 評価対象システムを動作途中の様々なタイミングで停止し、その際にユーザが状況を把握できること(今何が起こっているかを認識できること)を確認する(例:表示されたメッセージの内容やその後の処理を含めて妥当性があることを確認する) 新規の運用に沿った操作をマウスなし(ショートカット等)で行い、時間を求めているユーザの利便性の観点から要求が実現できることを確認する 		
その他(他のシチュエーションに当てはまらない障害)	<ul style="list-style-type: none"> 現在動作中のユーザ権限を別の権限に変更し、評価対象システムが適切なタイミングで権限を失った動作や処理を止められることを確認する(例:権限切替わりの契機が一度ログアウトしてログインし直すまでの場合、ログアウトする前まで変更前の権限が適用されていることを確認する等) 例:権限切替わりの契機が即時の場合、現在動作中の動作を強制的に止めさせられることを確認する等) 		<ul style="list-style-type: none"> 評価対象システムにログインした状態で放置し、別の端末から他ユーザ等が同じユーザID(または異なるユーザID)を使用してログインや処理を適切に実施できることを確認する(例:放置によりユーザIDがロックされないことや他ユーザ等がログインできなくなることを確認する等) 先行Verのシステムや前身となるシステムより売れている製品との比較を行い、ユーザが使いにくさを感じていないことを確認する(例:作業等が煩雑になっていないことを確認する等) 			