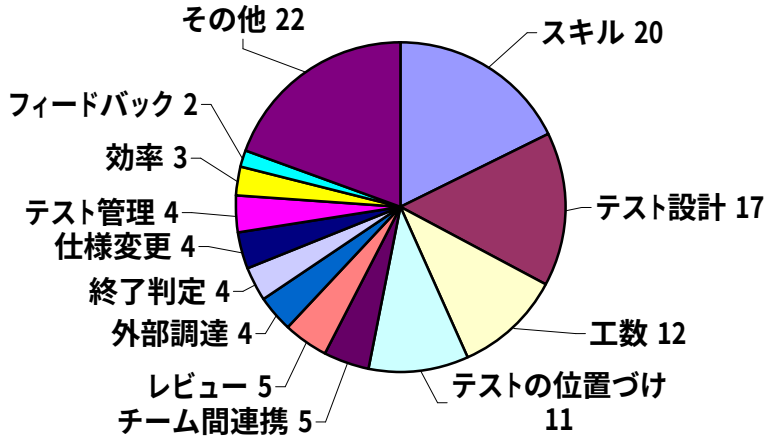
JaSST‘06札幌　参加型ワークショップ「ソフトウェアテストの現状」

この資料は、JaSST‘06札幌でのワークショップ「ソフトウェアテストの現状」の中で、約50名の参加者の皆様に挙げて頂いた、ソフトウェアテストに関する課題を集計したものです。

ソフトウェア・テストにおける課題



*以下は集計前のデータです。表現はニュアンスを失わない範囲で要約している場合があります。

スキル　20

・テストのスキル・ノウハウの共有できてない。
・新人の教育
・人によってテストの品質にバラツキが大きい
・未経験者に対する教育の方法・立ち上げのスピードアップ
・プロジェクトごとに独自のやり方を取っていて、共有化がされない。次のプロジェクトでも生かされない。
・スキル不足のある人材ないし新人をいかに教育・指導するか。
・チーム内のテスターのスキルに差がある。
・テスターの質が低い。使える人は開発に取られる。
・知識と実践の乖離
・テスト専門の要員を作る人的余裕がない
・受託開発で短期かつ多分野の開発が多く、テストエンジニアのスキルを蓄積しにくい
・テストエンジニアのなり手が少なく、外注に頼らざるを得ない。
・評価技術者の教育について、これといった良い方法にたどりつけておらず、スキルがバラバラ。経験ベースから抜け出せない。
・担当者によってチェックの結果にバラツキがある。
・テスト作成のノウハウの共有ができていない。
・テストに対する学習方法
・テストは経験に頼る部分が多いが、比較的経験の浅い者が担当する事がある。
・テストケースの基準が個人によりばらつく。
・有識者が持っているスキルの可視化。
・全体を知っている人／詳細を知っている人　の中間の人がいない。認識が合わない。

テスト設計　17

・テストに戦略性がない。
・確認項目の抜き取り方法
・短期間開発で試験工程を統合しており、テスト下流での不具合検出ができない傾向がある
・条件の組み合わせをどこまでテストするかの加減が難しい
・サブシステムごとのテストで問題がなくても、組み合わせ時に不具合が発生する事がある
・テストだけ見て改善してもいつかは止まってしまう。要件定義や設計とのつながりをどう考えていくかが課題。
・試験仕様書に書かない部分で意外とバグが多い。（タブ移動順、ドット単位のデザインなど）
・テストの計画はスケジュールのみで戦略がない

・テスト技法について書籍などに色々出ているが、自分たちの開発に適した方法を考えるのに苦労している。
・テストケースの設計が正しいかどうか検証できない。
・テスト項目の洗い出しがうまく行かない。詳細な部分まですべて出してしまうと、仕様書作成の工数が大きくなる。
・検査項目をどこまで細分化すると効率的かつ効果的なのか。
・テスト項目の妥当性が分からない。
・何万項目もテストしていても不具合が出てしまう。無駄な項目もあるのでは？
・テストケースの効果的な抽出方法が確立できていないこと。
・試験仕様書の粒度が作業担当者によってバラバラ
・テストケースの質・量ともに不十分。

工数 12

・テストファーストの普及を目指すが、余裕ない。
・テスト仕様書作成の工数が大きい。
・テストケースの質・量を上げる工数を確保できない。
・開発期間が短く、総合テストに十分時間を取れない。
・工数圧縮のためにテストを自動化したいが、その時間をタスクとして確保する事が難しい。
・ユニットテスト１００％を目指しているが、工数の関係で１００％書く時間が取れない。
・作業の量に対して工数が少ない。結果的に品質が悪くなる。
・納期が厳しいとどうしてもテスト工数を減らしてしまう。
・厳密性の向上に伴う工数の増大。
・テスト仕様書作成に掛かる工数が大きい
・実装に時間をとられ、工数が食いつぶされていく。
・コードが大きすぎて単体テスト仕様書作成に時間が掛かりすぎてしまう。

テストの位置づけ 11

・テストの重要性が上司に理解されない。
・テストは簡単な仕事であると言う認識がある
・テストの専任メンバーがいない。
・テストエンジニアの地位が低い
・テスト計画のプライオリティが低い。
・テストはプロジェクトごとに異なった戦略が必要。従って戦略を立案するには相応の工数が必要だが、会社はその重要性を認めていない。
・テストに予算が割けない。
・開発者と比べテスト担当者または品証の人数が少ない。
・テスト技術者に対する評価
・単体テストが十分でなく、結合・総合テストで拾っていく進め方。（納期ありき）
・テスト専任チームが早くから存在しているが機能していない。寄せ集めでスキル不足、テスト仕様書を作らない、テスト設計がない、テスト計画がない、テストを理解していない→その結果、実績がない、立場が低い、給料が低い、やる気がない。

チーム間連携 5

・設計者との連携
・開発チームとテストチームの連絡不足。テスト環境やテスト項目などのやり取りが少ない。
・開発とテストチームの間に障害検出や切り分けのやり取りが無く、テスト環境や項目の連絡不足があり障害修正の遅れや再評価に時間が掛かる。
・テストチームが離れたところにあるため、不具合として上げられた内容を理解する事が難しい事がある、。
・チームの拠点が2つあり、テストの進め方にバラツキがある。

レビュー 5

・レビュー方法が確立していない。
・レビューの質
・試験項目に対するレビューが不足している。
・レビューア不足で、レビューアの作業負荷が高い。
有識者へのレビュー負荷の集中。

テスト管理 4

・テストデータの管理に悩んでいる
・テストの目標が曖昧。試験密度、バグ検出密度などの数値目標を持たずに行っている。
・テスト実施時のバグ、実施、分析において評価内容の難易度の異なる状態で、進捗を判断する事が課題。
・テストに掛かる工数の見積りが正確に出来ない

外部調達 4

・OEMソフトの品質が低く、摘出バグに対する修正も遅い。
・開発が外注のため開発レベルでの対策が取れない。
・オフショアの品質問題
・外部発注したプログラムの品質

終了判定 4

・どこまでテストしてよいのか分からない
・どこまでテストをすればよいか。
・どこまでテストしていいか分からない＋妥当性。
・不具合の収束判断のタイミングが難しい。

仕様変更 4

・顧客からの要望がコロコロ変わったり、追加があり、何が正しい仕様なのかタイミングにより違ってしまう場合がある。
・プロジェクト終盤での仕様変更が多い。
・仕様の凍結と言う言葉があったが、最近では仕様の凍結をすると売れる製品が出来ない等あり、仕様の変更をうけないといけない。その場合の効率的なテスト技法は？
・製造～単体までに要求が曖昧な点がややあり、調整が間に合わない。

効率 3

・流用母体の規模が大きい開発の場合の、効率的なテスト方法
・複数の組み合わせ条件によるテストを効率的にテストしたい。
・少人数・短時間で行う開発での有効なテスト手法について。

フィードバック 2

・バグトラッキング、レビューのフィードバックをうまく回すのが難しい。
・テスト結果の分析が未実施／生かされない。分析方法が良くないのか。

その他 22

・仕様書、コードなど相当数のレビューを実施しており、多くの工数が掛かっている。品質向上も感じるが費用対効果のバランスが難しい。どのような測定が適しているのか。
・テストの自動化が難しい
・要求の意図
・障害度の基準をどの程度細分化すべきか
・一人での開発においてテストを手抜きしてしまう
・DB内のデータが変わるとユニットテストを再実行したときにエラーとなる。
・テスト仕様書の作成がテスト前になる事がある。
・第3者テストのチームを作ると、開発側のデバッグの質が下がる。
・モチベーションを維持できない人がいる。
・UnitTestを導入しているものの、後回しになりがち。
・納期が近くなるとドキュメント不足になり不具合が起こる可能性が高くなるが、見直すためのドキュメントが不足していて修正工数も大きくなる。→ドキュメントを簡略化したい。
・自分たちが作ったシステムではない。
・機能が多く複雑で全部を把握できない。
・テスト仕様書を書いていない。Unitテストでカバーしようとしているが、Unitテストもあまりかけていない。
・不具合帳票の内容が不正確
・評価スキルの可視化がうまくできない。
・ユニットテストの有効性をなかなか他のメンバーに伝える事ができない。
・短期開発の案件が多いため、単体テスト仕様書が残らない事がある。
・回帰テストができていない
・一人で設計、試験仕様、コーディングをやっているため、試験項目が妥当か、不足が無いか分からない。また、試験に都合のいいようにコーディングしてしまう。
・リスク判定基準が欲しい。
・品質の見極め