

**「テストプロセス、プロダクト品質評価のための実用的な
ODC分析とCATによる管理方法」**

**株式会社 SHIFT
山腰 直樹
石井 優**

**「製品の品質が落ちているが、どこから手をつければ？」
「新しい分析手法を試したいけどあきらめがち」。
こんなお悩みはありませんか？**

欠陥分析手法「ODC分析」を用いてテストプロセスやプロダクト品質を評価するための、具体的なメトリクスとその分析事例を紹介。

**「テスト活動は妥当か？」「製品に致命的な欠陥はない？」
よくあるお悩みを定量的に評価し、対策につなげていきます。**

弊社のテスト管理ツール「CAT」での情報収集と分析例も実演。

デモ、対談、QAタイムにてよりODC分析の利用方法について深めていただけるセッションです。

- 登壇者紹介（5分）
- ODC分析で始める、品質分析（25分）
- CATを使った品質分析 実践方法（25分）
- Q&A（5分）

スピーカー紹介



大手ITベンダーにて、約38年の製品開発（SWエンジニア）およびITシステム開発（ソフトウェアテスト事業）を経験後、2022年2月SHIFTに参画。

金融サービス部門にてお客様支援の傍ら、ライフワークのソフトウェアテスト事業の拡大の一環として「欠陥分析サービス」を立ち上げる。
また、我々自身すべてのメンバーに、SHIFTのもつテストオフリングをよく知ってもらい、お客様のニーズに応じたご支援ができるよう活動中。

Executive Test Specialist

山腰 直樹 (やまこし なおき)



プロフィール ～石井 優～

株式会社SHIFT
サービス&テクノロジー本部 技術統括部
開発支援プロダクト推進部 CAT開発室
セールスサポート CATエヴァンジェリスト
石井優 (Suguru ISHII)

経歴

2009/4 - 2014/11 倉庫系システム部門
2014/12 - 2015/7 東中野のPCサポート屋さん・石井屋
2015/8 - SHIFT・CATサポート

好きなもの

- ・ビール・酒・うまいもの
<http://blog.livedoor.jp/sugubon-otsumami/>
- ・Crazy Ken Band / 日本の合唱 / アレサ・フランクリン
/ チャラン・ポ・ランタン
- ・東中野



ODC分析で始める、品質分析

1. 品質評価における共通のお悩み ～ 品質の定量評価
2. 欠陥分析による品質の定量評価手法のODC分析とは？
3. 品質の定量評価、さらに踏み込んで「採点」をできないかを考えてみる
4. 今後の可能性と展望について

品質の定量分析って難しいですね

よくテスト完了報告書における品質評価のページで、以下のように評価されていますよね？

例) バグ密度：5.0/KLOC，前回と比較して向上しており適切なテスト活動と評価する

でもこれってよく考えると、、、

1. 品質評価における共通のお悩み ～ 品質の定量評価

でもこれってよく考えると、、、

例) バグ密度：5.0/KLOC，前回と比較して向上しており適切なテスト活動と評価する

この部分は**定量的**に
測定されていますが、

この部分は**定性的**な
評価ですよね？

できれば、適切であるとした部分も定量的、かつ客観的に評価したいですね。

たとえば上の例で言うならば、

今回の測定値の品質評価は80点です。前は60点でしたので、20点向上しました。

みたいなの。。。

1. 品質評価における共通のお悩み ～ 品質の定量評価

テスト工程の完了報告書における品質評価で、最も重要な要素の1つである障害分析では

検出されたバグの結果から、

- どのような傾向があり、
- 品質的には何点なのか、
- もっと良くするためには、（次は）何をすればいいのか

を述べたいですね。。。

本日は、ODC分析という技術を使ってそのようなことを実現するアプローチについてご紹介いたします。

2. 欠陥分析による品質の定量評価手法のODC分析とは？

ODC分析とは…

- ・ODCとは、1992年に米IBM社 ワトソン研究所で確立された「欠陥分類手法」
- ・ODC分析とは過去のODC分類データとの比較により品質を定量的に評価する手法で、RCA（なぜなぜ分析）とは異なる「欠陥の統計分析」

統計的分析

- ・ 履歴データとの比較
- ・ 成長曲線モデル

是正措置につながらない



原因分析（RCA、なぜなぜ分析）

- ・ 限定・特定不具合の調査

多大な労力・コストがかかる

ODC 分析手法

個々の不具合の意味論を定量的に捉え、
プロセスの進歩と品質成熟への橋渡し

効率よく原因・傾向が分析できる

「ソフトウェア不具合改善手法 ODC分析」
日科技連ODC分析研究会編より抜粋

ODC: Orthogonal Defect Classification（直交欠陥分類）

参考文献： <https://www.ipa.go.jp/files/000040829.pdf>

検出された欠陥を2つの軸（直交）で分類する手法です。

1. バグ報告者の観点 (どんなテストで見つかったか)
2. バグ修正者の観点 (どんな問題があったか)

代表的な分類属性

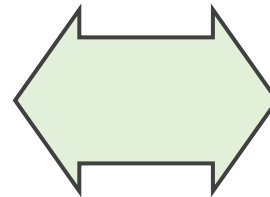
1. バグ報告者情報

トリガー (どのようなテストでバグを発見したか)

- 単一機能の基本操作
- 単一機能の特定条件
- 操作手順に依存
- 複数機能の相互作用
- 負荷・ストレス

など

インパクト (ユーザーに与える影響)



2. バグ修正者情報

欠陥タイプ (どのような修正をしたか)

- 値の代入・初期化
- 値のチェック
- アルゴリズム
- 機能／クラス
- インターフェース／メッセージ
- 処理の順番
- データやオブジェクトの関連付け

実装タイプ (実装のパターン)

- 修正
- 欠落

基本的に分類の粒度は大きいです。これによりある程度のデータが集まれば、統計分析が可能になります。
(あまり粒度が細かいと、統計分析を行うためには大量のバグデータが必要になる)

粒度の例 1 (バグ報告者)

1. バグ報告者情報

トリガー (どのようなテストでバグを発見したか)

- 単一機能の基本操作
- 単一機能の特定条件
- 操作手順に依存
- 複数機能の相互作用
- 負荷・ストレス

など

インパクト (ユーザーに与える影響)

- **単一機能の基本操作 (カバレッジ)**
特定な条件に依存せずバグが発生する

例：十分な預金がある口座からATMを使って引き出したが預金不足のエラーとなった。

- **単一機能の特定条件 (バリエーション)**
ある特定な条件でバグが発生する

例：普通預金と定期預金の合計が100万円、普通預金口座に50万円、の条件で、普通預金口座からATMを使って55万円を引き出す操作をしたら、エラーとなった。

基本的に分類の粒度は大きいです。これによりある程度のデータが集まれば、統計分析が可能になります。
(あまり粒度が細かいと、統計分析を行うためには大量のバグデータが必要になる)

粒度の例 2 (バグ修正者)

2. バグ修正者情報

欠陥タイプ (どのような修正をしたか)

- 値の代入・初期化
- 値のチェック
- アルゴリズム
- 機能／クラス
- インターフェース／メッセージ
- 処理の順番
- データやオブジェクトの関連付け

実装タイプ (実装のパターン)

- 修正
- 欠落

- **実装タイプ：修正**
ソースコードの一部を修正した

例：仕様書には20歳未満と記載されていたのに、20歳以下として実装してしまった。

- **実装タイプ：欠落**
ソースコードを一部追加した

例：仕様書には20歳未満で、性別が男性の場合と記載されていたのに、性別の判定ロジックが抜けており、その部分を追加した。

ODCの分類された欠陥情報を集計し、その結果について考察することがODC分析です。
 たとえば以下のように、ODCの分類で、開発プロセスの改善点が見える化できます。

例：バグ修正者情報と欠陥混入過程の関係

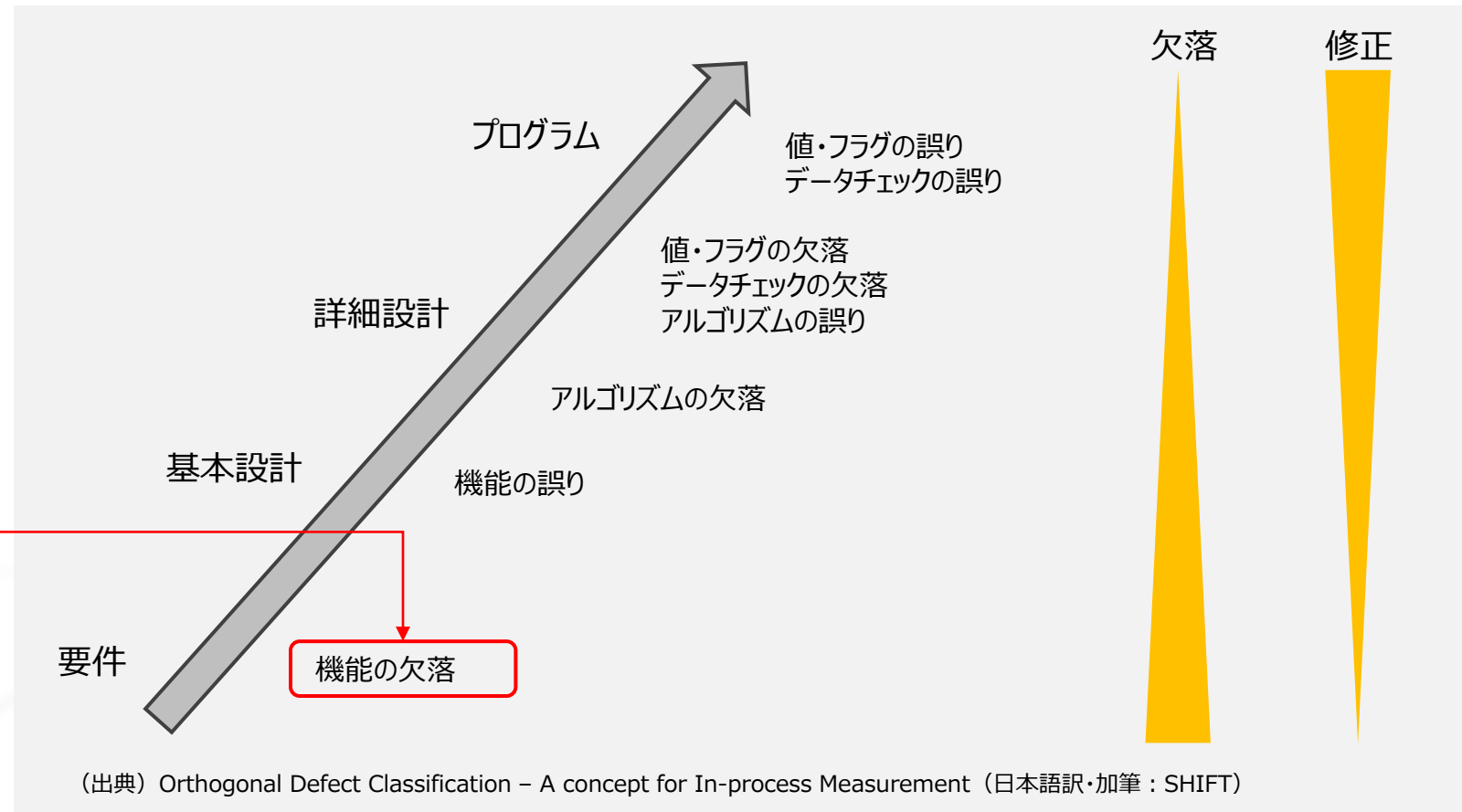
2. バグ修正者情報

欠陥タイプ（どのような修正をしたか）

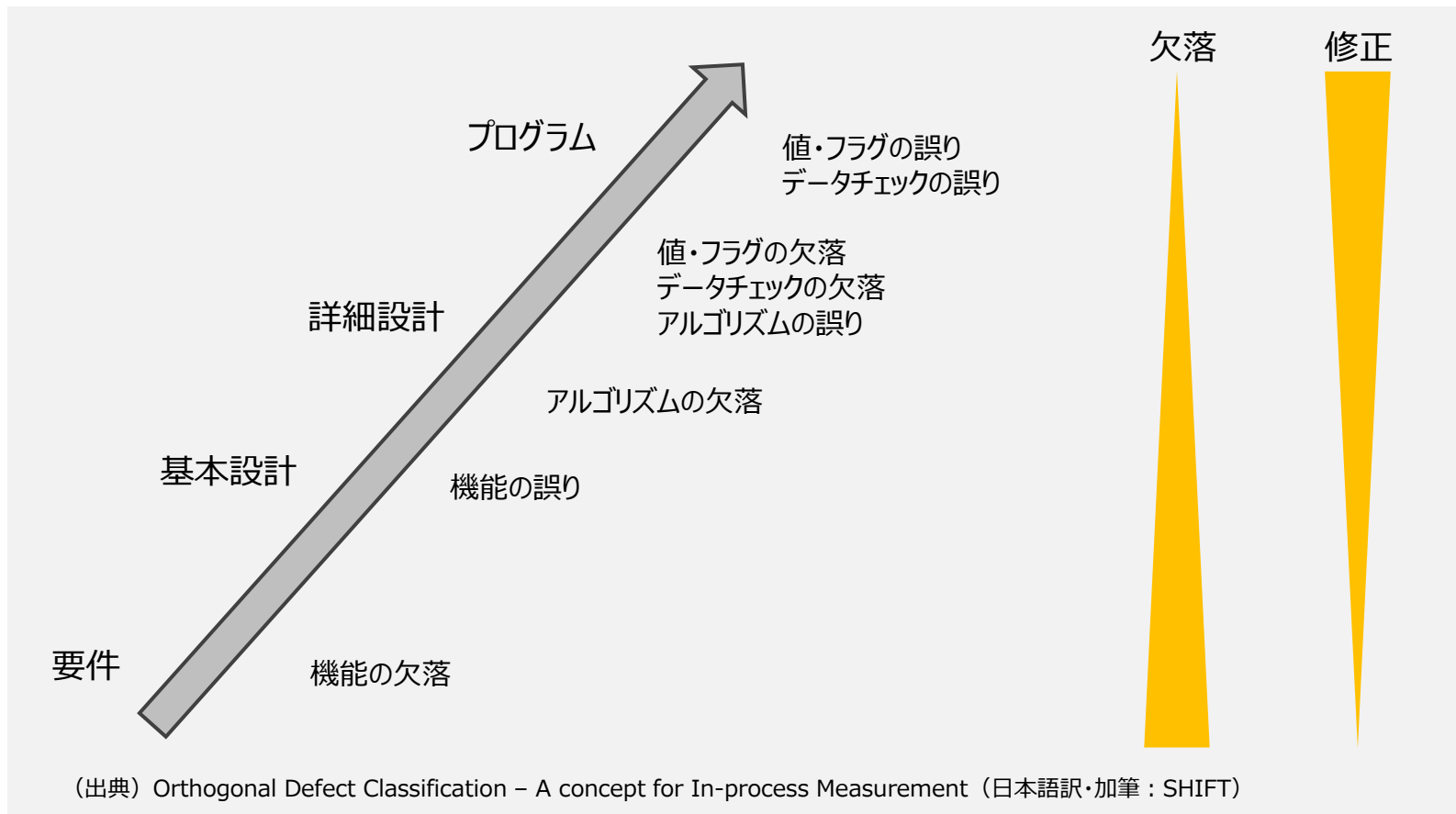
- 値の代入・初期化
- 値のチェック
- アルゴリズム
- **機能／クラス**
- インターフェース／メッセージ
- 処理の順番
- データやオブジェクトの関連付け

実装タイプ（実装のパターン）

- 修正
- **欠落**



それでは前ページの分析手法を使い、あるプロジェクトにおける分類結果を分析してみます。



値・フラグの誤り	112
データチェックの誤り	
値・フラグの欠落	29
データチェックの欠落	
アルゴリズムの誤り	
アルゴリズムの欠落	24
機能の誤り	0
機能の欠落	4

【分析結果サマリ】

一般的には欠陥混入工程が上流であるほど、課題は大きいとされています。

今回のケースでは詳細設計・プログラム工程における欠陥混入が多くなっていたので、担当チームへの連携あるいは教育が重要であると考えられます。

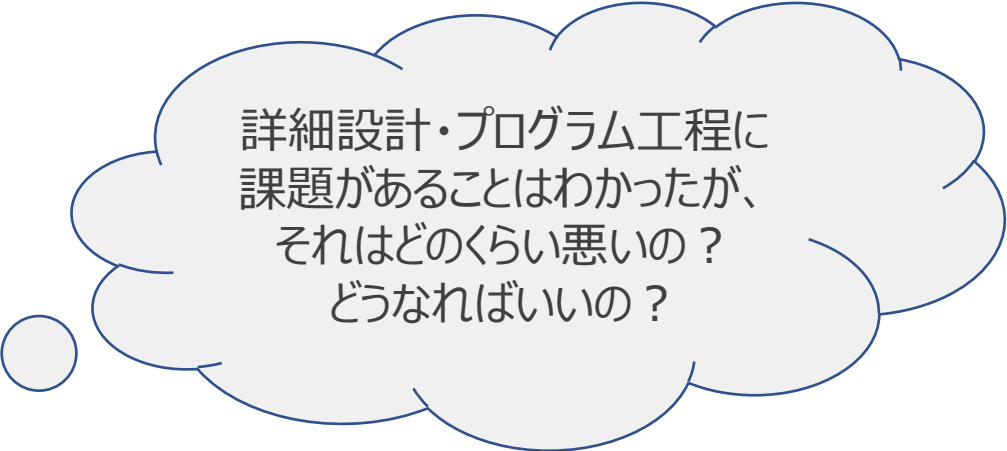
前ページの例では「バグ修正者情報と欠陥混入過程の関係」から、欠陥混入過程で改善の余地があると考えられる工程に対する対策を講じています。

これは定量的な評価と言えますが、いわゆる「絶対的」な評価であり、これをより「相対的」な評価、すなわちどのくらいの悪さ加減なのか、を述べることはできないでしょうか？

【分析結果サマリ】

一般的には欠陥混入工程が上流であるほど、課題は大きいとされています。

今回のケースでは詳細設計・プログラム工程における欠陥混入が多くなっていましたので、担当チームへの連携あるいは教育が重要であると考えられます。

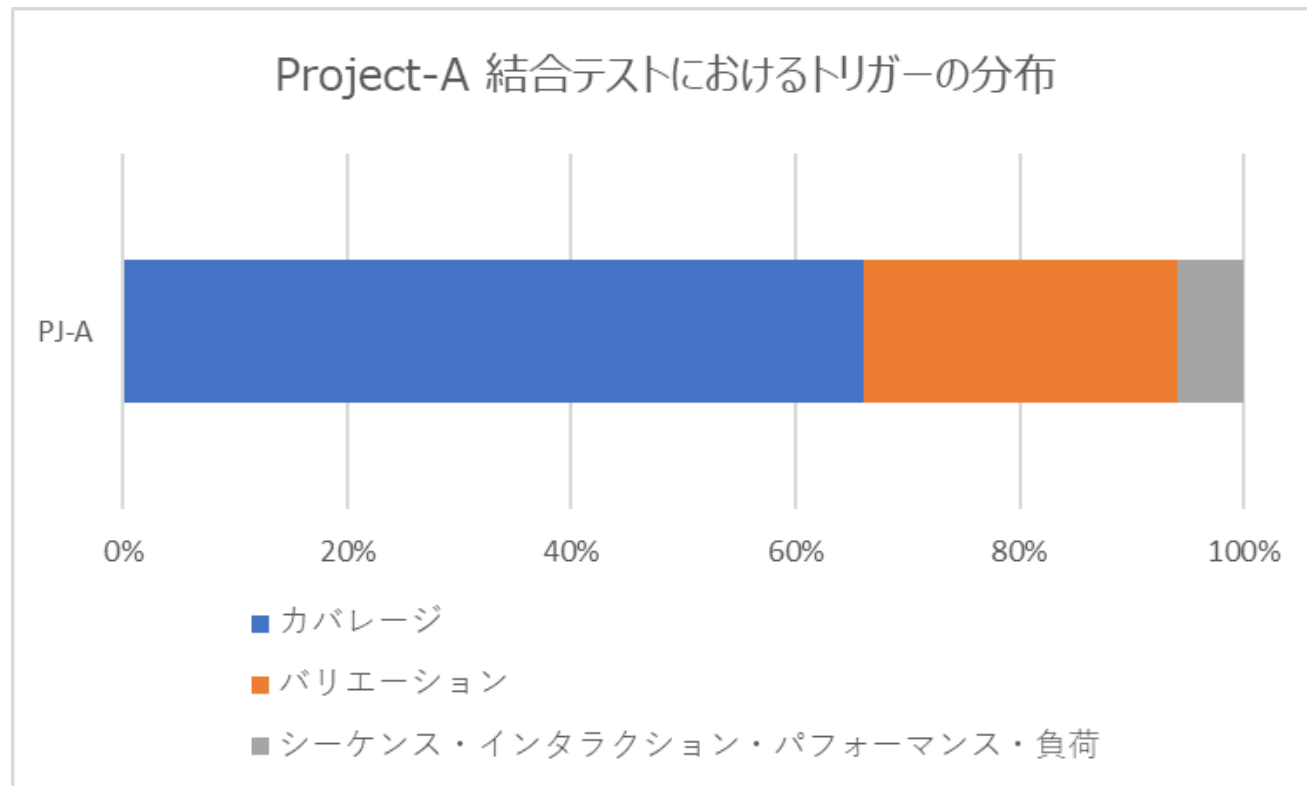


詳細設計・プログラム工程に課題があることはわかったが、それはどのくらい悪いの？
どうなればいいのか？

ここである事例における、結合テスト工程でのODC分類「トリガー（どのようなテストでバグを発見したか）」の分類結果を見てみます。

一般的にトリガーの分布は、どのテスト工程でどのようなテストを実施してバグを発見したかの分析になりますので、すべてのテスト工程を俯瞰して、それぞれの工程で適切なテストを実施できているかを分析するのに有効です。

つまり「**全体テスト方針の有効性**」についての評価になります。



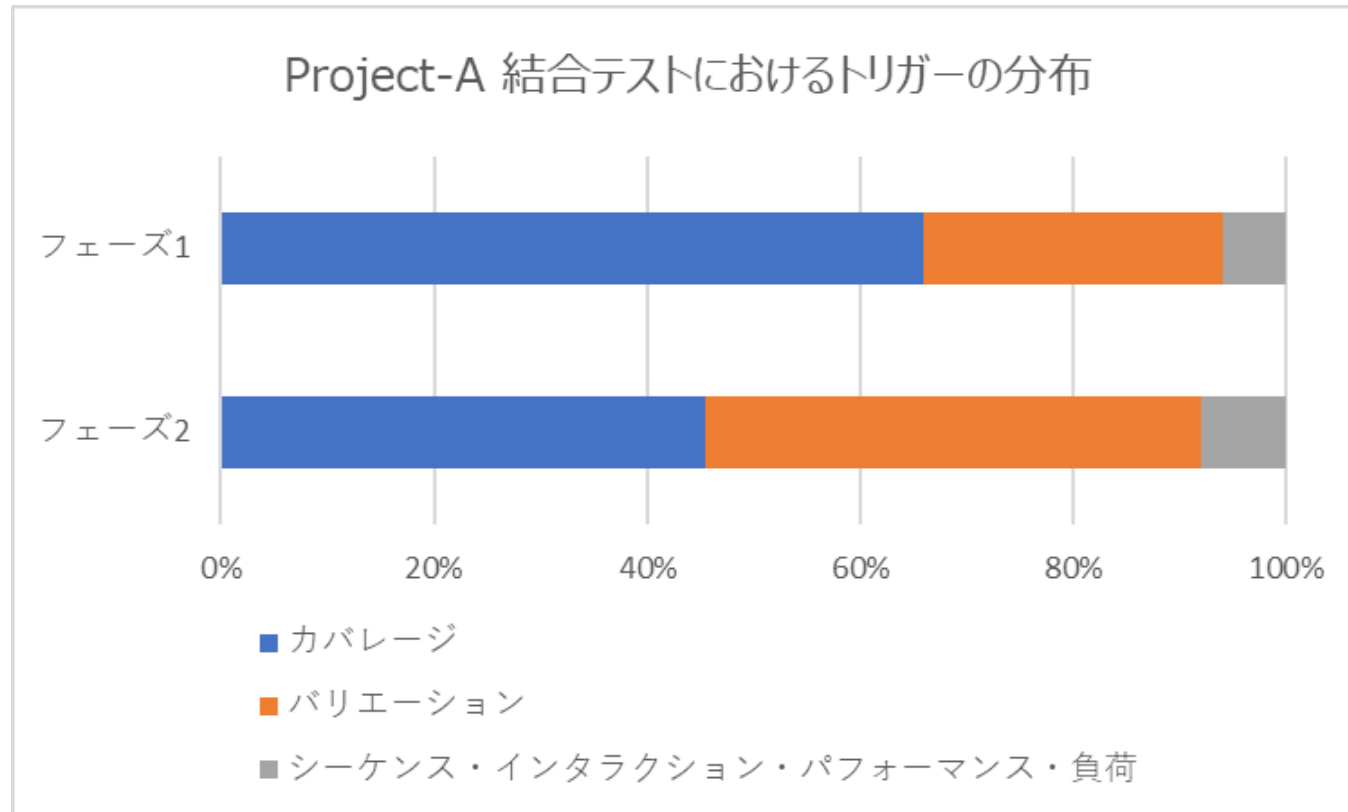
左のグラフは前ページの事例プロジェクトにおける結合テスト工程でのトリガーの分布です。

実はこのプロジェクトは2フェーズに分け実施しており、フェーズ1で様々な課題が浮き彫りになったため、フェーズ2ではその反省をもとに各種対策を打ち、様々な品質改善活動を取り入れ、実施しました。

その結果、フェーズ1での左の分布がフェーズ2でどのように変わったか見てみましょう。

⇒ 次のページ

適切な品質改善活動を取り入れた結果、フェーズ2での分布はこのようになりました。



このように適切な品質改善活動を取り入れたフェーズ2では、トリガーの分布が大きく変わりました。

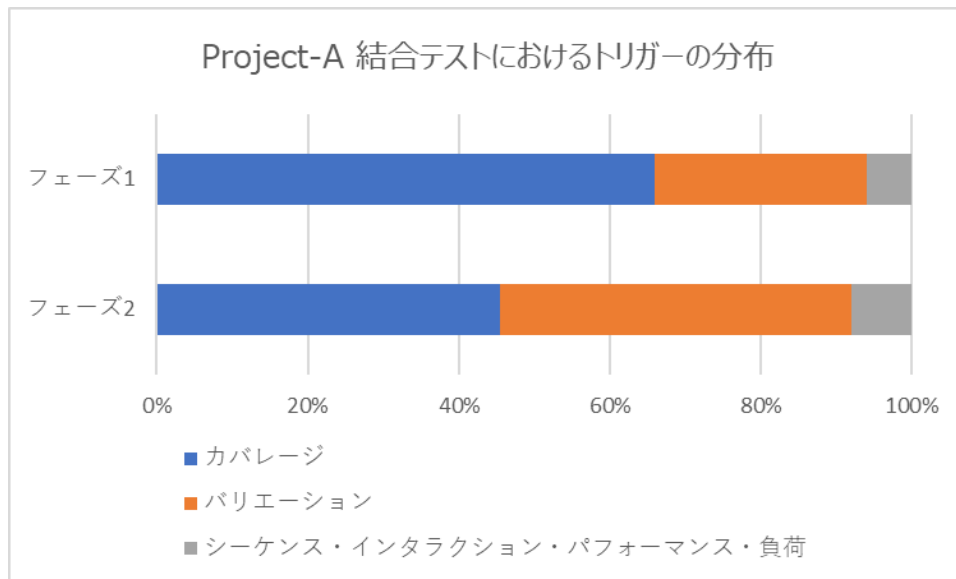
実はこの傾向はProject-Aだけではなく、すべての案件においてこのような傾向を示すということが経験上わかっています。

つまり、類似の特性を持つプロジェクトでは、結合テストにおけるトリガーの分布はこうあるべきである、という**ベストプラクティス指標値を設定**できるということです。

それでは上記の仮説に従い、実際にフェーズ1のトリガー分布の状態を「採点」してみましょう。

3. 品質の定量評価、さらに踏み込んで「採点」をできないかを考えてみる

今回の場合、適切な品質改善活動を取り入れたフェーズ2での分布結果をベストプラクティス指標値と考えます。それでは「全体テスト方針の有効性」について、0~5の6段階で「採点」してみます。



カバレッジは基本操作で検出されたものであり、フェーズ2では46%に抑えられていますが、フェーズ1では66%もあります。

これまで多くのデータを分析してきた経験上、この数値は5%刻みに段階を評価するのが適切と考えます。

ベストプラクティス値が46%であるのに対し、フェーズ1では20%の開きがあるので、4段階減点されます。

すなわちフェーズ1における「全体テスト方針の有効性」の評価は、“1”（0~5の6段階のうち）であるということになります。

トリガー	フェーズ1	フェーズ2
カバレッジ	66%	46%
バリエーション	28%	47%
シーケンス・インタラクション ・パフォーマンス・負荷	6%	8%

ODCの分類結果には、開発・テストプロセスの傾向が現れます。

では開発・テストプロセスにおける**品質管理の観点**からどのようなメトリクスを設定すればよいのでしょうか？

上流工程では、

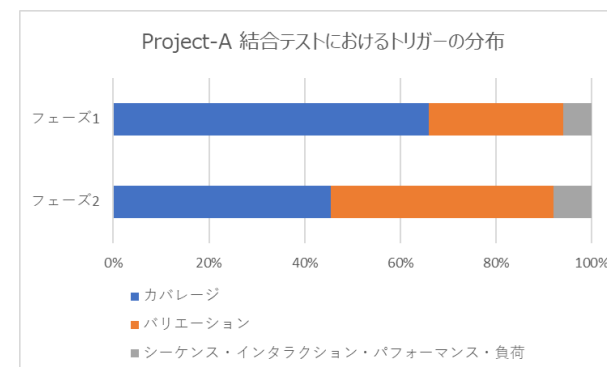
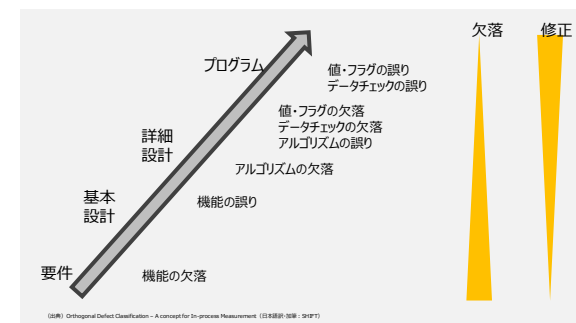
通常欠陥が混入されるのは上流工程（要件定義～基本・詳細設計～製造）です。
欠陥混入の工程別分布を分析すれば、特にどの工程で課題があるか、が見えてくるはずです。
そして前述の通り、ODCのバグ修正者情報から欠陥混入工程を紐づけることが可能なので、このメトリクスは測定可能です。

下流工程では、

通常V字モデルの下流工程（単体テスト～内部・外部結合テスト～総合テスト）においては、それぞれの工程の目的に即したテストが行われます。

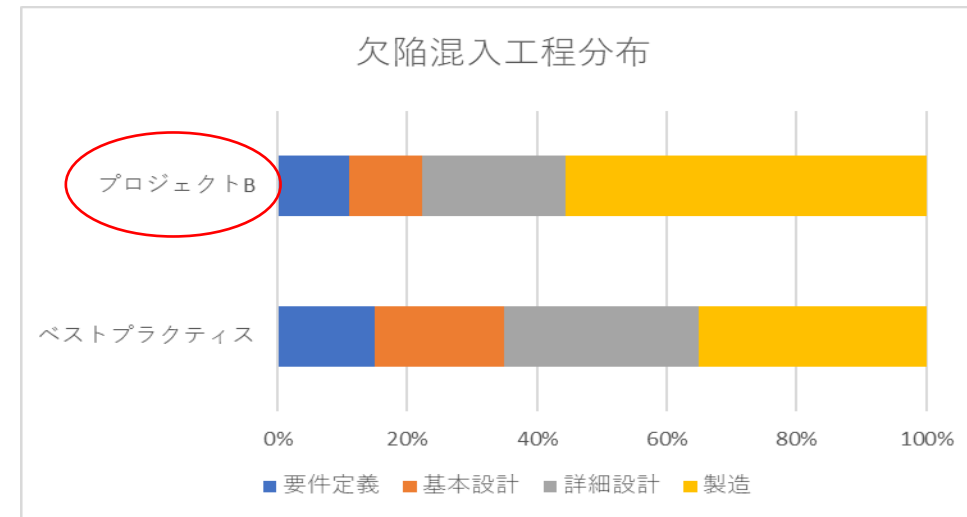
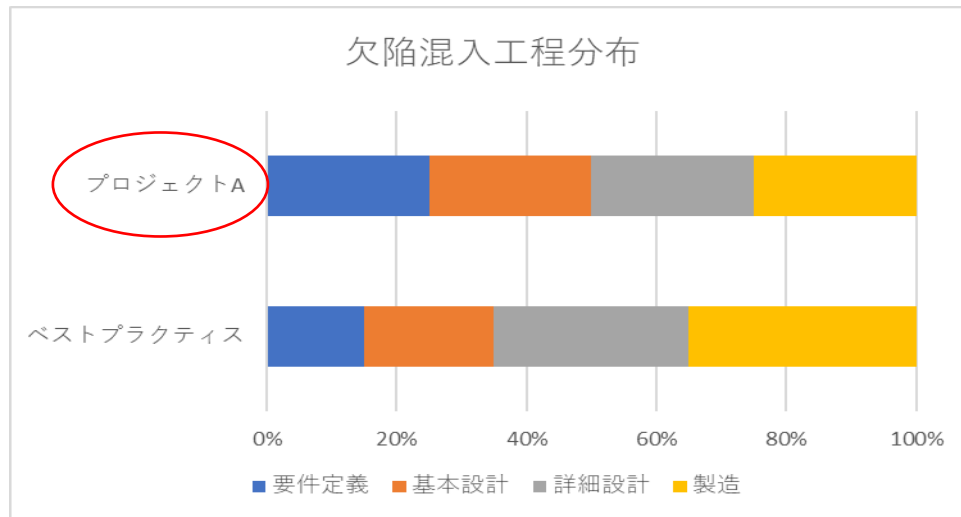
（例：単体テストで画面単機能テストを重点的に実施する、など）

たとえば前述の通り、結合テストにおいて**検出されたバグのトリガー分布**を分析すれば、その工程およびその工程に至るまでに実施すべきテストがしっかりされていたかどうかを評価することが可能です。



欠陥混入の工程別分布を分析すれば、特にどの工程で課題があるか、が見えてくるはずで、課題解決のための有効な対策を講ずることができます。

たとえば欠陥混入の工程別分布がベストプラクティスに比較して以下であったとします。



※ ベストプラクティス値は仮のものです。

プロジェクトAでの欠陥混入は平均的に分布されており、一見すると問題ないように見えますが、ベストプラクティス値と比較することにより、とくに要件定義工程での品質活動に課題があることがわかります。

プロジェクトBでは、明らかに製造（プログラミング）工程に課題があることがわかります。

したがってプロジェクトAとプロジェクトBでは、取るべき対策は異なるものになるはずです。

4. ODC+アルファの品質評価の可能性について

SHIFTのテスト管理ツール：CATに蓄積された莫大な欠陥データの情報を活用すれば、今後さらに踏み込んだ品質評価ができるようになるはずです。

CATにはテスターが検出したバグについて、たとえば次のような分類情報が紐づいています。

1. テスト観点

SHIFTでは工程別で多く検出される（すべき）標準テスト観点が定義されています。

⇒ 標準テスト観点の分布のベストプラクティス値を使えば、「本来検出すべき工程で検出できたのか」が見えるはずです。

例：画面I/Fの 카테고리

画面 I/F	表示
画面 I/F	レイアウト
画面 I/F	遷移
画面 I/F	入力
画面 I/F	操作

例： 카테고리: 表示のテスト観点

- 階層の確認
- 推奨画面サイズでの表示
- 推奨外画面サイズでの表示
- 推奨解像度での表示
- 推奨外解像度での表示

2. プログラムのバグではなかった場合の原因の分類

テストの本来の目的はソフトウェア（プログラム）のバグを検出することです。

しかしテスターがバグとして報告したものが、ソフトウェアのバグと認識されなかった場合は、別の原因があるはずです。

たとえば……

- テスターの誤操作
- テストケースの不備（期待値の誤りなど）
- 与えられたテストデータの不備
- 与えられたテスト環境の不備

などです。

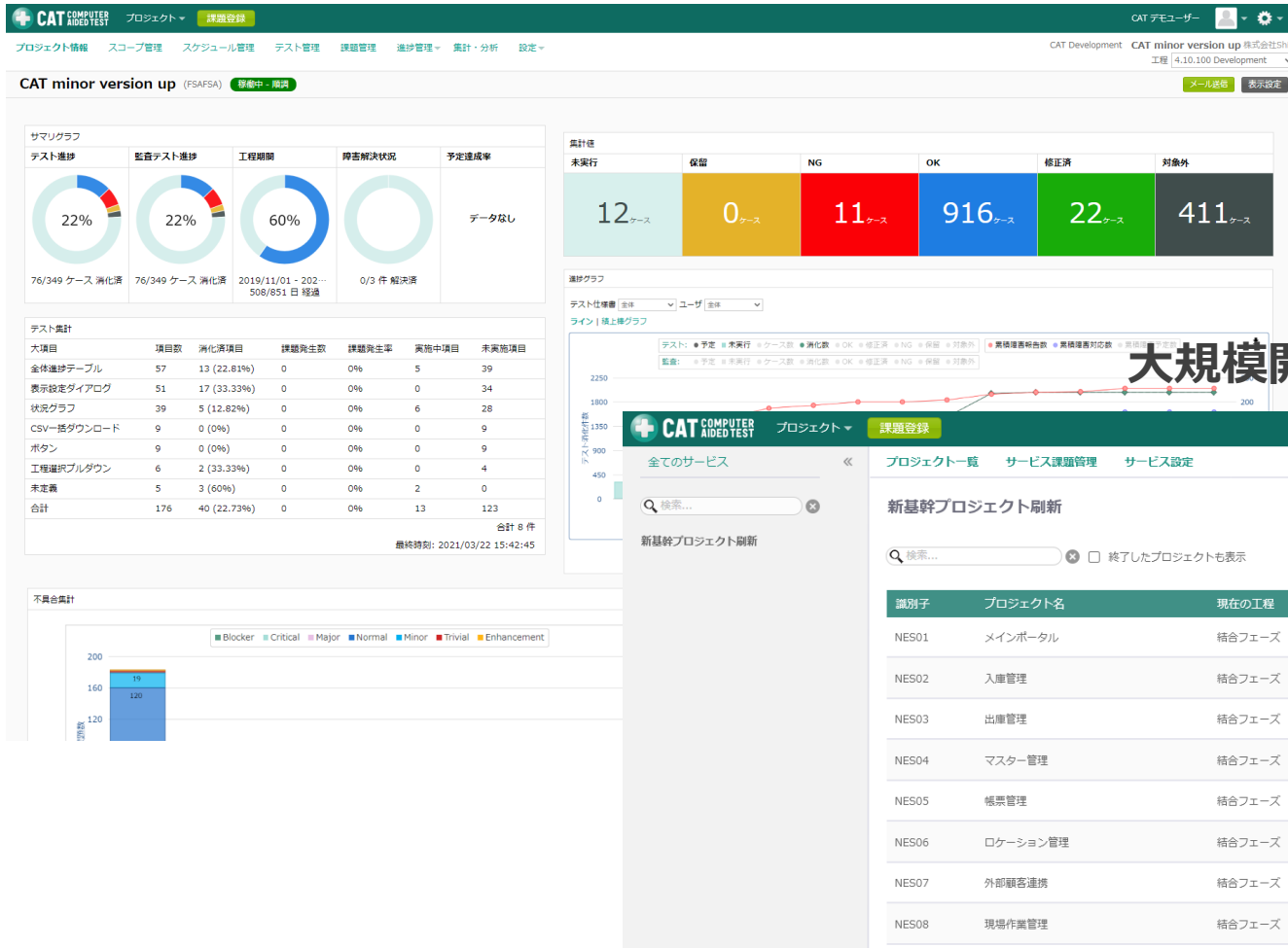
もしこれらの分布がベストプラクティス値からかけ離れている場合は、対策が必要となります。

- 書籍：ソフトウェア不具合改善手法 ODC分析
杉崎 眞弘 (著), 佐々木 方規 (著), 日科技連ODC分析研究会 (編集)
- IPA公開資料：SEC高信頼化技術適用事例
ODC分析による欠陥除去と品質成熟度の可視化 ～医療機器の安全性・高品質を担保するために～
オリンパスソフトウェアテクノロジー株式会社 山崎 隆
<https://www.ipa.go.jp/files/000040829.pdf>
- Web資料：Backlog 管理図分析
<https://backlog.com/ja/blog/what-is-a-control-chart/>

CATを使った品質分析 実践方法

テスト・欠陥情報を一元的に管理し、テストフェーズをリアルタイムに可視化！

プロジェクトの品質・進捗を一元管理できるダッシュボード



大規模開発でもサブシステムごとに進捗・パフォーマンスを把握

中規模～大規模プロジェクト管理におけるブラックボックス化を防止、
次のような課題を解決し、品質の高いプロジェクトを実現します。

- **正確な進捗や障害の発生・対応状況が見えない**

テスト全体のボリューム、進捗、障害発生状況、などがチームからの報告を元にぼんやりとしか把握できない。

- **課題の発見が遅くなる**

手作業で情報を収集しているため、報告に上がってこない課題・リスクなどが発生する。

- **対応コストの上昇、品質の低下を招く**

テストの遅延や、数日のテスト実施不可期間などの発見が遅れ、それに影響した後続のテスト実施や障害

対応も遅れ、結果即時での対応に比べ、対応コストが高くなる。

影響範囲も大きくなるため、品質の低下に繋がる。

CAT導入時の定量的効果例

プロジェクト内で発生する作業に対するコスト削減例

チーム体制、マネージャ2名、テスト実行者13名、週2回集計するチームの例

ロール	人数	単価(時間)	作業	現状の作業時間(週)	CAT利用時の作業時間(週)	削減可能な時間(週)	削減可能なコスト(週)
マネージャ	2名	1.25万円/時	Excelファイルの回収・確認	0.5時間/日×2人×2日 =2時間/週	0時間/日	2時間/週	(2時間/週+4時間/週) ×1.25万円 = 7.5万円/週
			テストと障害の情報を集計	1時間/日×2人×2日 =4時間/週	0時間/日	4時間/週	
			状況の分析、課題の抽出	2時間×5人×2日 =8時間/週		0時間	
テスト実行者	13名	0.625万円/時	結果・エビデンス登録	5時間/日×13人×5日 =325時間/週	4時間/日×13人×5日 =260時間/週	65時間/週	(65時間/週+13時間/週) ×0.625万円 = 約48.8万円
			進捗状況のまとめ、報告	0.5時間/日×13人×2日 =13時間/週	0時間/日	13時間/週	
削減可能なコスト合計							約56.25万円/週 約225.00万円/月

CAT DL版ライセンス費用(買い切り・25ユーザー÷12ヶ月) : 7.5万円/月・25ユーザー

CAT クラウド版サブスクリプション(15ユーザー・月額) : 5.2万円/月・15ユーザー

求められる進捗管理の細かさに応じて、削減効果が大きくなります
貴社での効果についてもお確かめください

プロジェクトで頻繁に発生する課題事例と、対応例①

大規模プロジェクトにおいては、数百万単位の損失を防止

プロジェクトで頻繁に発生する課題

課題	事例	CATを導入した場合の効果	CATコスト例	課題に関するお客様からの声
<p>チーム単位でテスト中断が発生、発見と解決が遅れ、稼働工数のロスが発生</p> <p>(大規模プロジェクト・600人/月)</p>	<p>状況</p> <ul style="list-style-type: none"> 1サブシステムのチーム（20人）でテストが中断。 進捗の可視化・エスカレーションが遅く、3日後に課題を本件を把握、その後対応 <p>影響</p> <ul style="list-style-type: none"> 当該チームの稼働が対応開始まで4日程度停止、工数の400時間のロス（20人×4日×5時間） 後続テストの別チーム（10人）のテストも2日停止、稼働時間損失が雪だるま式に増え、工数100時間のロス（10人×2日×5時間） 	<p>改善例</p> <ul style="list-style-type: none"> テストの中断を、半日で検知。ロスを50時間で抑え、350時間のロスを阻止（20人×0.5日×5時間） 後続のテストチームへの影響は防げたため、100時間のロスを阻止 合計225万円の損失を阻止（0.5万円/時 ×（350+100）時間） 	<p>104万/月（600L）</p>	<p>大手エネルギー小売企業</p> <ul style="list-style-type: none"> 正確な状況把握がいつでも、だれでも（経営層、PM、P他）できる 進捗の異常にいち早く対応し、明日以降の計画の反映できる 600人規模のプロジェクトでもスムーズに導入できる

大規模プロジェクトでは、このような事例は何度も発生します。

積み重なると、大きな損失だけでなく、プロジェクトの大幅な遅延・信用失墜に繋がります。

ブラックボックス化を阻止、いつでも課題をキャッチできる状況にすることは、テスト管理の基礎となります。

プロジェクトで頻繁に発生する課題事例と、対応例②

短期、中規模プロジェクトにおいても効果を発揮

プロジェクトで頻繁に発生する課題

課題	事例	CATを導入した場合の効果	CATコスト例	課題に関するお客様からの声
<p>1メンバーのテスト中断が発生、発見が遅れ稼働工数のロスと納期遅延が発生</p> <p>(短期プロジェクト・10人/月)</p>	<p>状況</p> <ul style="list-style-type: none"> 5営業日の短期プロジェクト テスター1人の稼働が、スキル不足により止まっており、翌日発覚。 リカバリに上位者も一緒にテストを実施 <p>影響</p> <ul style="list-style-type: none"> テスター8時間(1日分)のロス 上位者のテスト実施4時間が発生 納期が1日遅れる 	<p>改善例</p> <ul style="list-style-type: none"> テスターの課題を3時間で検知、5時間分のロスを阻止 上位者のテスト実施4時間分のロスを阻止 合計約5万円の損失を阻止 (0.5万円/時 × 5時間、0.62万円/時 × 4時間) 納期通りに納品完了 	<p>2.65万円/月 (10L)</p>	<p>スタートアッププロジェクト</p> <ul style="list-style-type: none"> 時間単位で進捗を確認し、短期プロジェクトにでも安心できる 開発チームからの進捗問い合わせにすぐに対応できる 再テスト対象なども把握でき、テストの効率もあがる
<p>メンバーの集計コストがかさむ</p> <p>(中規模プロジェクト・50人/月)</p>	<p>状況</p> <ul style="list-style-type: none"> 外部ベンダーのPMOチーム6人のうち、1名は進捗集計係として稼働 <p>影響</p> <ul style="list-style-type: none"> 1名分の作業費用が発生、月80万のコスト その上、集計が遅く不正確なため、正確な情報が把握できない 	<p>改善例</p> <ul style="list-style-type: none"> PMOチームを5人に削減、月80万のロスを阻止 正確な情報をリアルタイムに把握できるように 	<p>12.7万円/月 (50L)</p>	<p>流通系開発企業</p> <ul style="list-style-type: none"> 無駄な集計コストをなくせる 人による報告粒度のばらつきや誤りをなくせる

中規模、小規模なプロジェクトにおいても、モニタリングが重要であり、コストの削減にも効果があります。

品質分析に必要な複数の項目を、適切なタイミングで、適切なメンバーに、ヌケモレなく入力してもらおう

品質管理メンバーがタグ付け

- タグ付けに時間と労力を要する
- このメンバーだけではわからない情報もある

Excelで各メンバーに欠陥情報を入力してもらおう

- ステータスごとに入力してほしい項目がもれる
- 起票日付や、担当者などもいちいち手入力が発生し、正確かつ大量の情報を集めづらい

ツールを使って欠陥情報を入力してもらおう

- ステータスごとに必要な項目を適切なメンバーに入力強制できヌケモレを防ぐ
修正完了→開発担当者・ソース属性・タイプ属性
- 起票日付や発生環境などはほぼ自動的に設定でき、**入力の負担を軽減**

起票時に必要な項目
だけ必須可する

設定する項目の意味
については、事前にメ
ンバーに説明を行う

必須項目はなるべく1
画面で見渡せる範囲
に配置する

自動で入力される項
目も活用する

障害報告-新規 隠す

工程	* 結合試験
種別	* バグ
タイトル	* ダッシュボードアイテムを追加後、ウィンドウを狭めると大きさが追従しない
ステータス	* 未着手
Spec owner	i --選択--
報告者	* CAT デモユーザー
概要	<p>・ダッシュボードアイテム編集画面 ・ウィンドウを狭めたときに大きさが追従しない</p> 書式
操作手順	i * 1. アイテム追加を行う 2. ウィンドウの幅を、300px以下にする
再現性の有無	i * <input checked="" type="radio"/> あり <input type="radio"/> なし <input type="radio"/> 未確認
機能	* ダッシュボード
発生環境	i Win10 Edge
(ODC)実施したテスト (システムテスト)	i * 復旧テスト/例外テスト
(ODC)ユーザーへの影 響	i * ユーザービリティ
添付ファイル	ファイルを選択 i
担当者	* SH_川口
発生座	...

1回に入力させる項目は
10項目程度に抑える

ダッシュボードアイテムを追加後、ウィンドウを狭めると大きさが追従しない

編集 共有 ▼ その他 ▼ < 前へ 次へ > 1 / 12

ユーザー
22:49
ユーザー
22:49

作業ログ

工程 : 結合試験	種別 : バグ
ステータス : 未着手	再現性の有無 : あり
機能 : ダッシュボード	発生環境 : Win10 Edge
(ODC)実施したテスト : 復旧テスト/例外テスト (システムテスト)	(ODC)ユーザーへの影響 : ユーザービリティ
優先度 : P3 (中)	重要度 : Normal
環境依存性 : 未確認	
概要 : ・ダッシュボードアイテム編集画面 ・ウィンドウを狭めたときに大きさが追従しない	
操作手順 : 1. アイテム追加を行う 2. ウィンドウの幅を、300px以下にする	

ステータスに応じた必須項目を設定する

ステータス変更

課題の次の担当者及びステータスを設定して、更新を押下してください。

ステータス 担当者

必須フィールド

開発担当者 *

修正バージョン *

修正内容 *

(ODC)欠陥の原因 ⓘ *

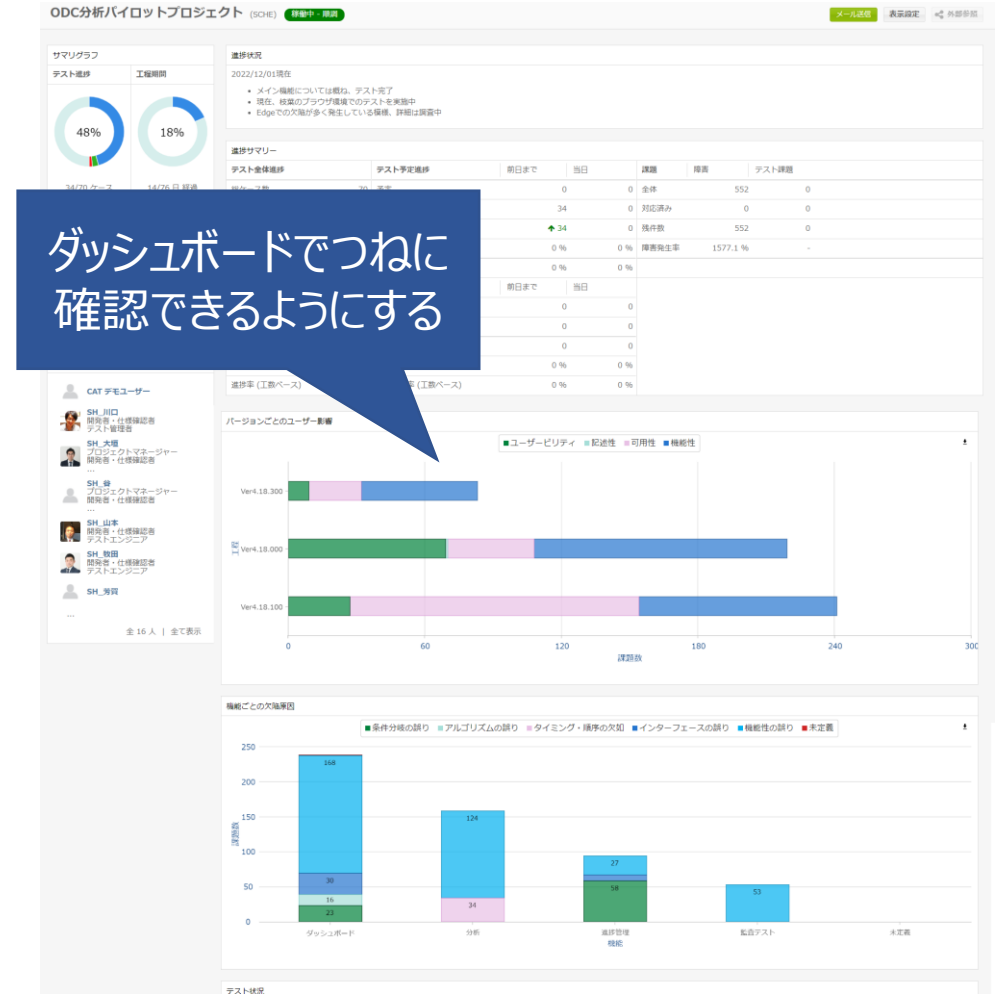
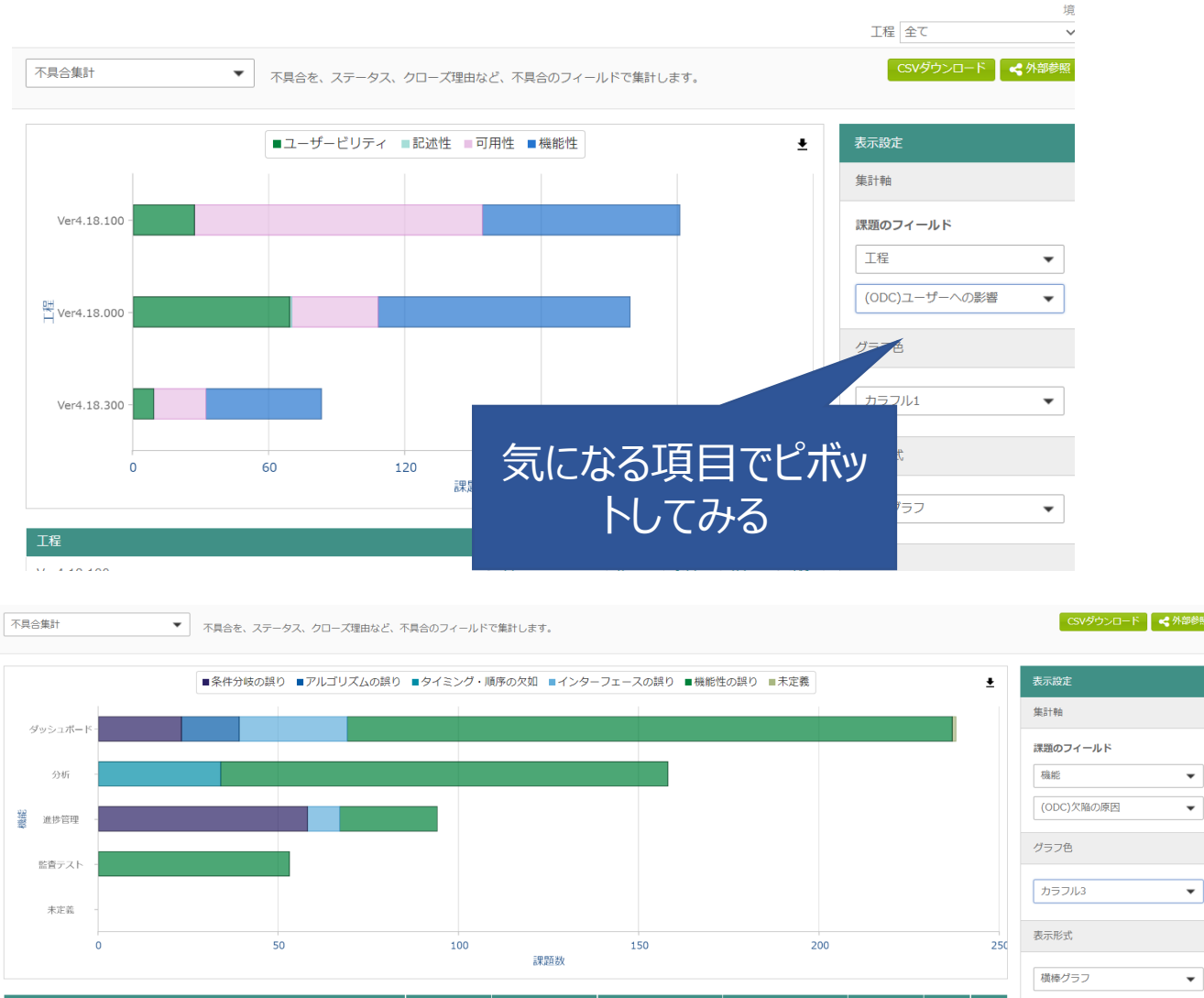
(ODC)ソースコードの性質 ⓘ *

見積工数(時間) *

コメント

ⓘ

運用を実装する上でのポイント(レポート)



CAT フリープラン リリース！

小規模なチーム向けの**永続的に無償**でご利用いただけるプランをご用意しました！

特徴

- **主要機能（テスト管理、課題管理、進捗管理、分析）を制限なく無償で利用可能**
- 最大ユーザー数10ユーザーまで
- 申込みから利用まで**最短3分で環境構築、稟議もカードも不要！**
- マニュアル、サポート（ベストエフォート）が利用可能
- **サンプルデータが入っている**ので、機能の利用イメージがしやすい

こんなお客様へおすすめ

- **小規模なチーム・プロジェクトでテスト管理・課題管理を利用したい**
ノーマルプランは少し予算が合わないという方も大歓迎
- **ノーマルプランの利用検討のために長期で利用したい**
大規模プロジェクトなどを念頭に数ヶ月の利用検討にも活用可能
- **CATに興味があるのでとりあえず触ってみたい**
テストのフォーマットや進捗情報などサンプルデータの登録もあり

ノーマルプラン（有償）との主な差異

- 同時プロジェクト管理数：5個まで
 - 最大ユーザー数10ユーザーまで
 - 一部機能の制限あり（外部BTS連携、ガントチャート（スケジュール管理）など）
 - 90日利用しない場合、環境を自動でクローズ
 - ノーマルプランへの移行はできません（エクスポートしたデータの利用は可）
- その他の差異は[価格ページ](#)をご確認ください

利用開始はこちら（クレジットカード情報不要、最短5分で利用開始）

<https://www.catcloud.net/>

CAT テスト管理ツール

で検索！

製品HPイメージ

統合型ソフトウェアテスト管理ツール「CAT」
フリープラン リリース！

22年7月、テスト管理に必要な機能を無期限・無料で利用できるプランをリリース。小規模な開発チームや長期で試したいPMOメンバーに最適なプランです。



サンプルデータに触って各種機能をお試しいただけます

サンプルサービス

課題ID	プロジェクト名	現在の工程	工程開始	総ケース数	現在のケース数(進捗率)	予定達成率
SAMP01	ECサイトテスト(サンプル)	初期結合テスト	100% 経過	300 ケース	150 ケース (50%)	50% (↓-150)
SAMP02	ECサイト会計機能強化テスト(サンプル)	総合テスト	43% 経過	10 ケース	0 ケース (0%)	-
SAMP03	ECサイト強化テスト(サンプル)	総合テスト	0% 経過	10 ケース	1 ケース (10%)	-



種別	タイトル	担当
問い合わせ	一括登録の仕様についての問い合わせ	
要望	一回で5000ケースアップできるようにします。	サ:
不具合	ログインページからユーザーがログインできない	サ:
不具合	[テスト管理] Excel形式でテストケース500件の…	サ:
不具合	[テスト管理] ユーザは課題のマッピングができ…	サ:

とことん使い倒す技

- 1組織内でもメールアドレスが異なる場合は複数の環境も構築可能
- 設計支援ツール「TD」から出力したテスト仕様書（CSV）をそのままアップロード可能
- 課題管理のカスタマイズも無制限
カスタムフィールドやフローで課題管理・分析基盤を構築
- テスト仕様書や課題の登録は無制限
- 高度な進捗管理、分析機能でプロジェクトを見える化

CATをご利用際のサポートはこちらをご利用いただけます。

マニュアル

すべての機能マニュアルをWebで公開しています。

サポートサイト・FAQ

メンテナンス及びよくあるお問い合わせなどを掲載しています。

問い合わせ（メール）

いつでもお気軽にご連絡ください。

cat_tech_support@shiftinc.jp

※通常、2営業日以内に返信致します。

※フリープランの場合はベストエフォートとなります

CAT note 公式ブログ

細かい活用方法、TIPSやCATが開発されてきた歴史などをブログで紹介しております。

お役立ちメール

もっと効率的に利用するためのTIPS・活用事例を月1で配信。

興味をお持ちのお客様にもぜひご購読いただきたい内容です。

[購読フォームから登録](#)

CAT製品HP

製品の最新情報などを掲載しています。

CAT導入事例

ご利用頂いているお客様の声を掲載しています。

TD製品HP

TDの概要、機能について掲載しています。

🔍 **CAT テスト管理ツール** で検索！

その常識、変えてみせる。
SHIFT

>コーポレートサイトはこちら >サービスサイトはこちら

CAT COMPUTER AIDED TEST 導入事例 機能一覧 アップデート情報 価格 サポート マニュアル お問い合わせ **無料で利用する**

統合型ソフトウェアテスト管理ツール“CAT”
主要機能を無償で公開

22年7月、テスト管理に必要な機能を無期限・無償で利用できるプランをリリース。きめ細やかなテスト、課題、進捗管理をすぐにご利用いただけます。

フリープランの情報を見る

今すぐ無料で試す

今すぐ無料で試すからお試ください！

お知らせ >> 2022/7/14 基礎機能は制限なし10人まで無料のフリープランをリリース！

テスト設計支援ツール「TD」遂にリリース
設計支援システムと蓄積したナレッジをもとに
誰でもカバレッジの高いテスト設計が可能に。

詳しい情報を確認する

TD
TEST DESIGNER

いかがでしたでしょうか