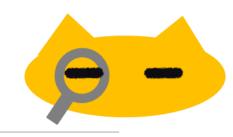
## チーム紹介

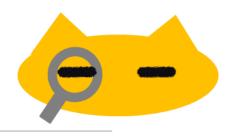




東海/関西のテスト設計の有志。



# 前提条件



私たちは、オープンソースソフトウェア開発コミュニティのQAチームとして、個人の限られた期間を利用して、プロダクトのリリースを止めるバグを減らすために、どう取り組むべきか、テストでどのように対処すべきかを検討しています。

#### 問題・課題とそのつながり フィードバックが多いバグ 票が大量に出る) バグがなかなか治らない 課題の全体を整理する人がいない バグが見過 提起課題の全体像が分 ごされる からない ソフトウェア構造 ソフトウェア構造の全体像を が分からない 簡単に把握できる手段がない 修正箇所の影響範 コードをいじると 囲が分からない 不具合が出る

→ テストする人員が少ない

メーカーとのコミュニティの連携



テストのモチベー

ションが低し

テストが十分に

できていない

## テスト開発コンセプトとリリース基準

### テスト開発コンセプト

「プロダクトのリリースを止めるバグ を許容範囲まで減らす!」



【メリット】 :プロダクトのリリースを止める事象に注力して対処できる。

【デメリット】: リスク分析の質に依存する。

### <u>リリース基準</u>

#### 【項目】

- ユーザーがリスクを受容できる
- 身体への障害を及ぼさない
- 外部要請:法律など、どの程度要請を満たさない

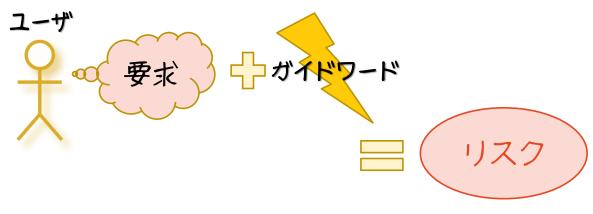
多くのユーザにとって、ソフトウェアの価値が根本的に損なわれる問題がある例: 起動しない。音楽が流れない。



## 対処する品質と本テスト設計の関係。

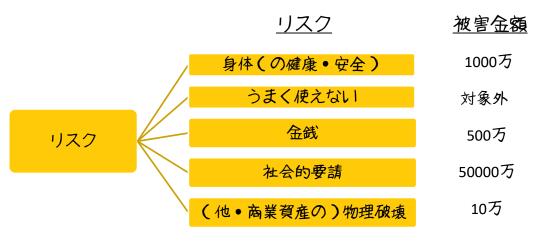


要求+HAZOPによるリスクの導出



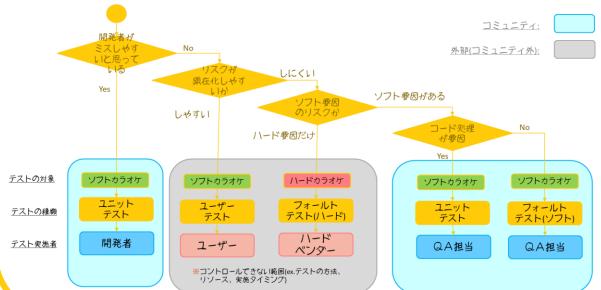
起きてほしくない事象

#### リスクカテゴリによる抜け漏れチェック



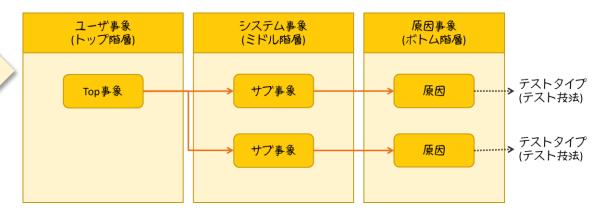
※リスクカテゴリ、被害金額は、コミュニティで定義

### 事象ごとのテスト担当決めフロー



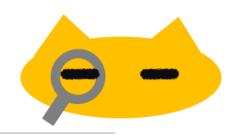
### フォールトツリー図によるテスト設計

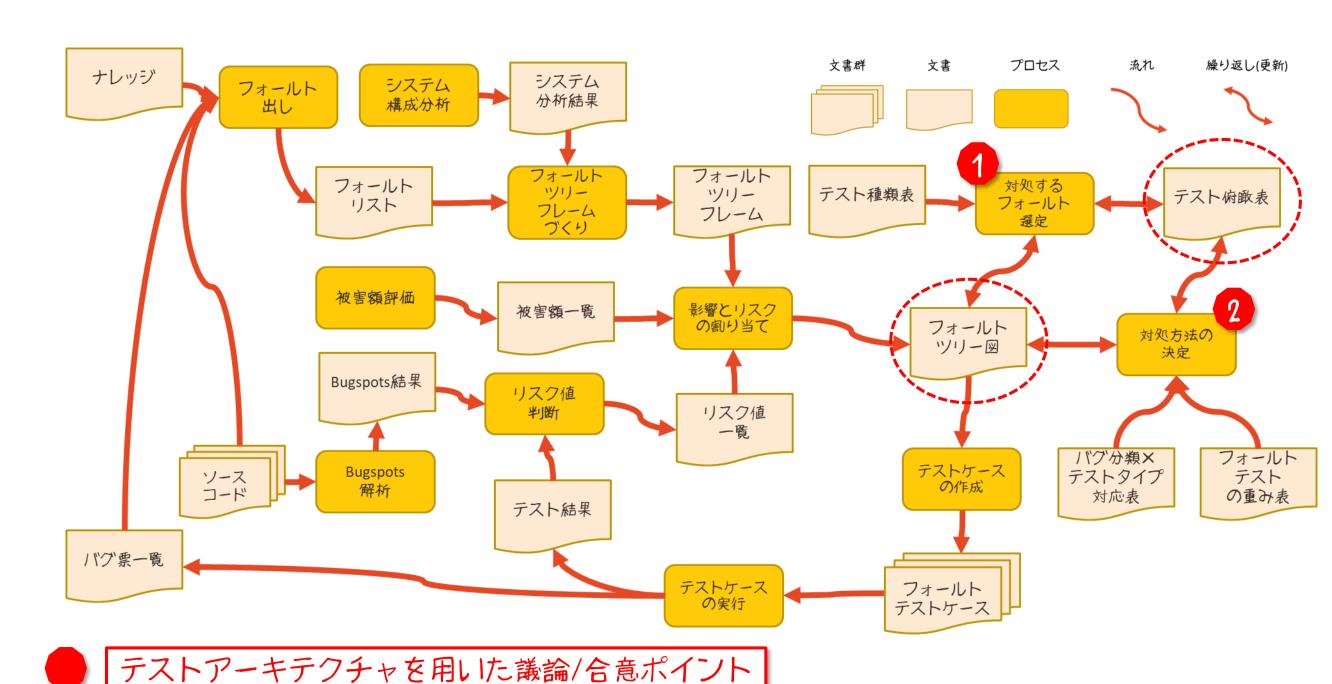
リスクアセスメント(リスク特定、リスク分析、リスク評価)を行う。





## テストプロセスフロー





テストアーキテクチャの定義:ステークホルダーとテストで議論/合意したいことを示すもの。



# Q さすにゃん B テストアーキテクチャ

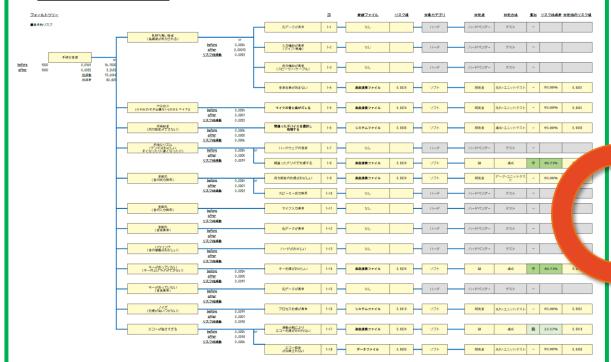
### フォールトビュー

対応プロセス:

対処する フォールト 强定

事象の影響度(被害金額)と発生確率(リスク値)から対処すべき事象(フォー ルト)を特定し、対応者について合意する。

#### フォールトツリー図



### テスト設計方針ビュー

対応プロセス:

対処方法の 決定

テストで対処する起こってほしくない事象のために、テスト設計方針を合意

項目	内容
ユーザーテスト	コミュニティのユーザーが実施する。ユーザーの通常利用に 耐えられない事象が発生しないことを確認することが目的。 コミュニティのスタンスとしては、実施要否はユーザーに任 せる。
フォールトベースド テスト	コミュニティのQA担当が実施する。ユーザーの通常利用では 顕在化しない、かつ、影響度(被害額)と発生確率が高い事象に 対処することが目的。
ユニットテスト	コミュニティの開発者自身が実施する。コードの品質を最低 限担保することが目的。

#### テストタイプ⇔テスト技法対応表

テストタイプ	(キーワード)	テスト技法		(テストケース帯出モデル
<ul><li>データ (領域、境界、サイズ)</li><li>性能</li></ul>	テストタイプ+ 抜け/温れ ズレ/浜び 選動/不足 矛輡/衝突 未反映/初期値/なし	同値分割 境界値分析	例	11 11
<ul><li>- 流れ (シナリオ、処理)</li></ul>	同上	フローチャート	例	
<ul><li>イベント/タイミング (状態、モード、選移)</li></ul>	同上	状態遷移図/表	例	•
<ul><li>構成 (環境、機器、モジュール)</li></ul>	同上	分類ツリー法	ØJ	<u> </u>
<ul><li>条件 (入出力、制約)</li></ul>	同上	デシジョンテーブル	<b>Ø</b> IJ	A

#### フォールトテストの重み表

		テストタイプ <mark>※1</mark>					
		シナリオ N-カバレッジ	組み合わせ N-wise	状態遷移 N-switch	同値・境界値		
	軽 最低限のテスト	CO	1	0	同値		
	中	C1	2	1	同値+境界値(2値)		
	重	C2	3	2	同億十境界億(3億)		

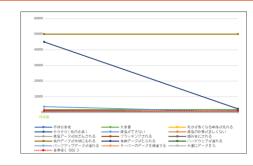
	テストタイプ <u>※</u> 2				
	シナリオ	組み合わせ	状態遷移	同値・境界値	
	N-カバレッジ	N-wise	N-switch	1017世 東北小川田	
整 最低限のテスト	30%	67%	50%	30%	
整 最低限のテスト 中	50%	95%	80%	50%	
(大)	65%	99%	90%	75%	

#### 【議論/合意内容】

- ①事象の被害金額とリスク値
- ②対処する起こってほしくない事象
- ③起こってほしくない事象の対応者
- ④テストの重み(カバレッジ)
- ⑤テストタイプ(対応方法)

#### テスト俯瞰表





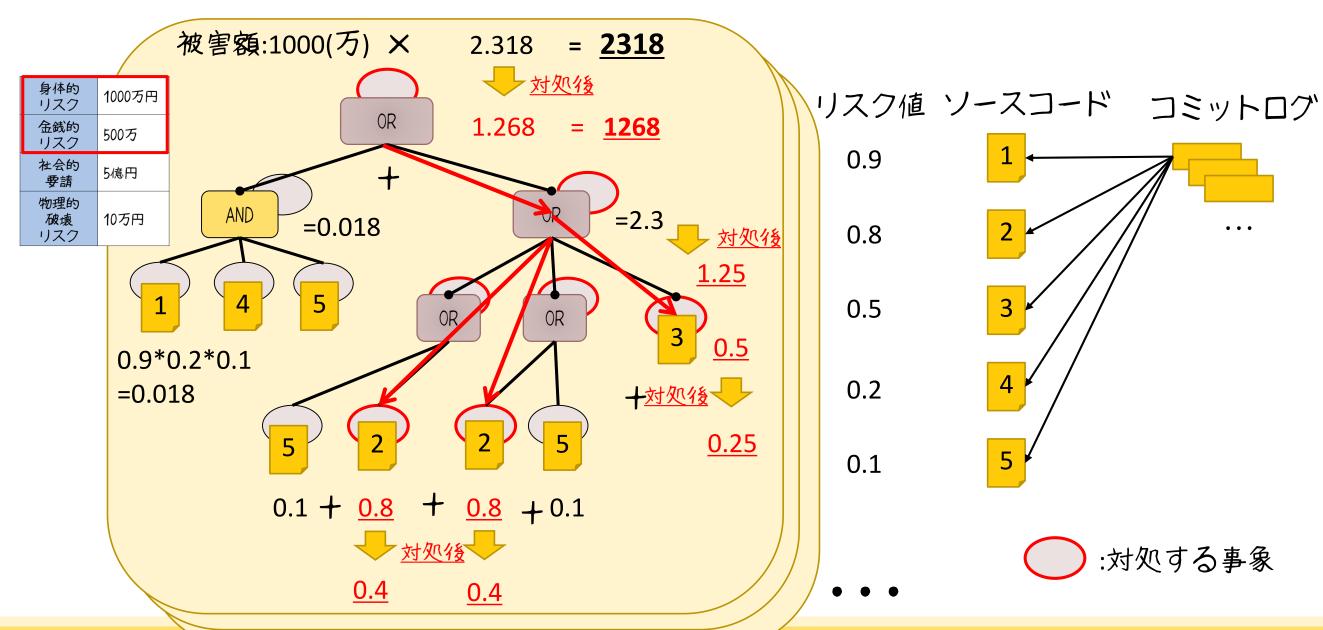
やばさの傾向グラフ





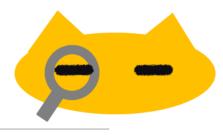
## 対処する事象(フォールト)の識別と評価方法

- 1. ToP事象からフォールトツリー図をそれぞれ作成する
- 2. Bugspotsというツールを用い、コード変更およびバグ修正から導出されるリスク値をソースコード毎に算出する
- 3. 作成したフォールトツリー図の事象にソースコードを割り当てる
- 4. 被害額×リスク値が高いTop事象に対して、リスク値を下げるために、テストやそれ以外の対応を検討する



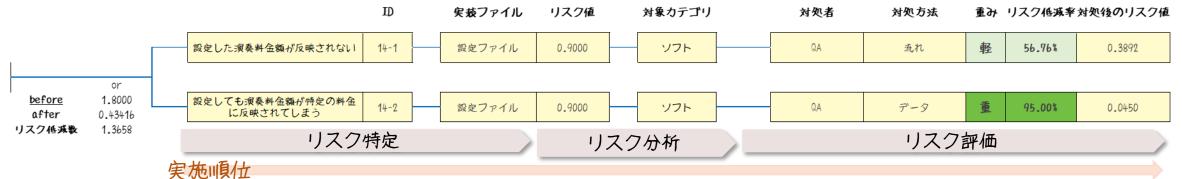


## テスト設計サンプル



#### ①フォールトツリー図

この図を用いて、リスクアセスメント(リスク特定、リスク分析、リスク評価)を行う。



#### ②テスト俯瞰表

複数のフォールトツリー図の結果を俯瞰し、優先順位、対応内容などを合意する。



### <u>③フォールトテストケース</u>

リスクアセスメント(リスク評価)結果 から対応する事象のテスト等出モデルを 作成し、テストケースを作成する。

