# ASTER通信カラオケシステム テスト設計のご提案



2017年2月24日(土)

いわた、すはら、なかはら、

えのき、ホーだん:※3/※:



# 私たちのチーム紹介





東海/関西のテスト設計の有志。



# アジェンダ



- 1. 提案の概要
- 2. 前提条件と課題
- 3. テストの全体像
- 4. まとめ



# テスト開発コンセプト

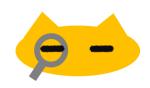


「製品のリリースを止めるバグ を許容範囲まで減らす!」





# 前提条件



開発製品 :オープンソースソフトウェアの通信カラオケシステム

・立場 :コミュニティメンバ QA担当

・リソース :2人

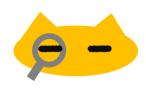
●リリース周期 :2週間(最新版)、6か月(安定板)

製品のフィーチャーはコミュニティ開発者の意思で、適宜実装される。





# リリースについての課題



- 1. 製品のリリースを止める事象およびバグをどう特定するか?
- 2. テストで対処する/しない事象はどのように決めるか?
- 3. 対処するためにどんなテストをどれくらい行うか?





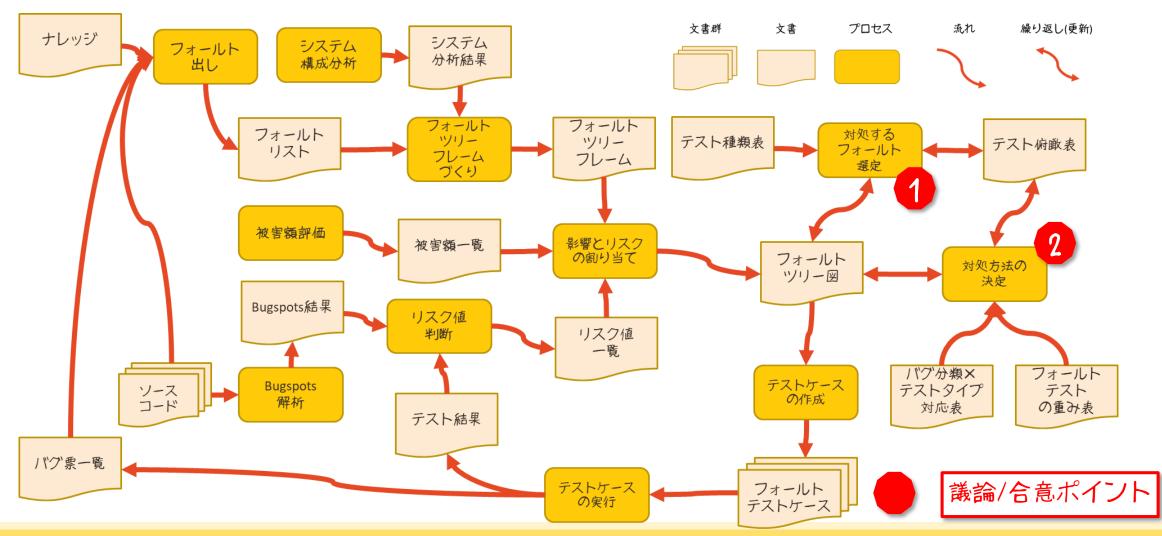




トの全体像まとめ

# テストの全体像







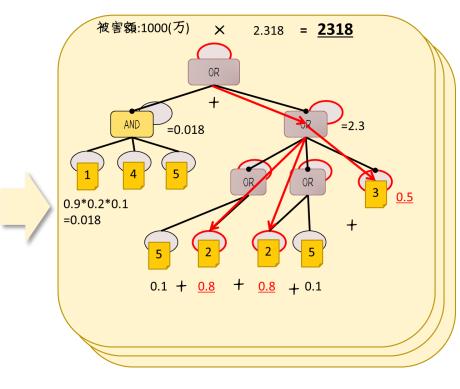
## テストアーキテクチャ(フォールト)



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。

身体的 リスク	1000万円
<b>金銭的</b> リスク	500万
社会的要請	5億円
物理的破壊 リスク	10万円

	不快な音波
身体に	大音量
影響	気分が悪くなる映像が流れる
	チカチカ(光の点滅)
	課金ができない
課金	課金の計算が正しくない
	課金データが改ざんされる
	クラッキングされる
社会的	踏み台にされる
要請	操作データが外部にもれる
	楽曲データがとられる
	ハードウェアが壊れる
物を	バックアップデータが壊れる
破壊	サーバーのデータを破壊する
	大量にデータをDL





## テストアーキテクチャ(フォールト)

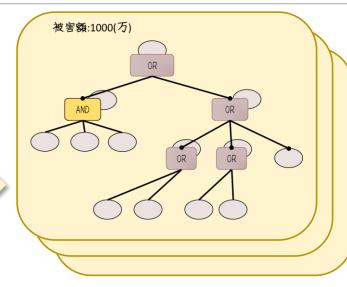


	不快な音波
身体的	大音量
リスク	気分が悪くなる映像が流れる
	チカチカ(光の点滅)
<b>金銭</b> 的 リスク	課金ができない
	課金の計算が正しくない
///	課金データが改ざんされる
	クラッキングされる
社会的	路み台にされる
要請	操作データが外部にもれる
	楽曲データがとられる
	ハードウェアが壊れる
物理的破壊	バックアップデータが壊れる
リスク	サーバーのデータを破壊する
	大量にデータをDL

#### フォールトリスト

#### 合意事項:

- 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

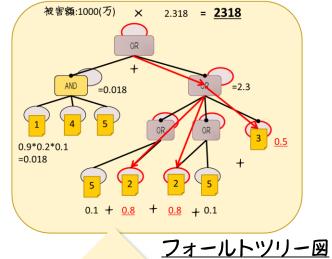


フォールトツリーフレーム

身体的リスク	1000万円
金銭的リスク	500万
社会的要請	5億円
物理的破壊 リスク	10万円

被害額一覧





No.	File Name	Path	Bugspot s (0.0~1. 00)	前回の 値	更新の 有無
1	pykdb.py	\pykdb.py	0.0134	None	更新
2	pykaraoke.py	\pykaraoke.py	0.0093	None	更新
3	ChangeLog	\ChangeLog	0.0078	None	更新
4	pykar.py	\pykar.py	0.0024	None	更新
5	pycdg.py	\pycdg.py	0.0019	None	更新
6	performer_prompt.py	\performer_prompt.py	0.0018	None	更新
7	pympg.py	\pympg.py	0.0010	None	更新
8	_pycdgAux.c	\_pycdgAux.c	0.0009	None	更新
9	setup.py	\setup.py	0.0009	None	更新
10	pykmanager.py	\pykmanager.py	0.0006	None	更新
11	.cvsignore	\.cvsignore	0.0003	None	更新
12	pykaraoke_mini.py	\pykaraoke_mini.py	0.0002	None	更新
	pykplayer.py	\pykplayer.py	0.0002	None	更新
14	README.txt	\README.txt	0.0002	None	再新

リスク値一覧



案の概要 前提条件と体

テストの全体像

## フォールトツリー作成方法

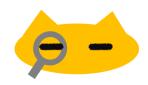


### 三層に分けてフォルトツリーを考える

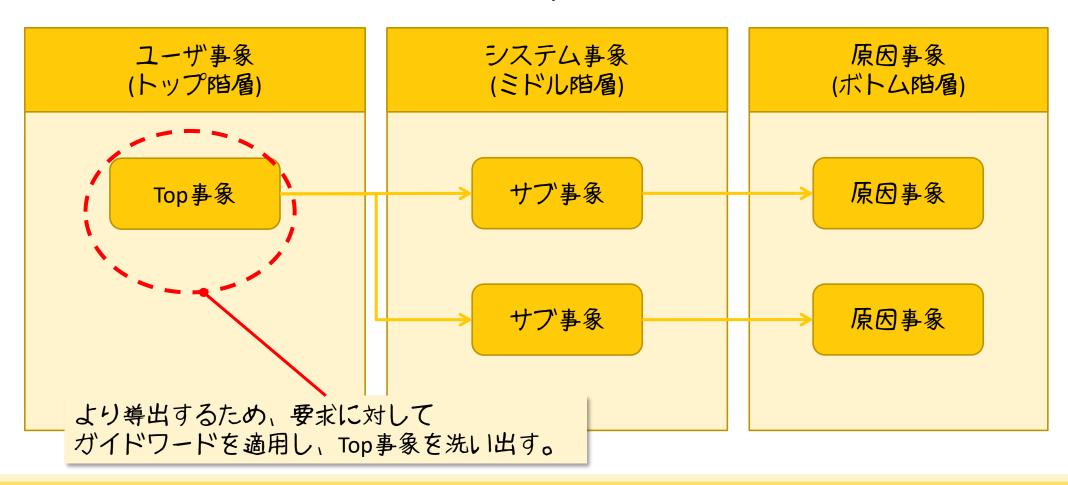




## フォールトツリー作成方法



ステークホルダーの要求からTop事象を出す。





# 要求からのTop事象(フォールト)だし

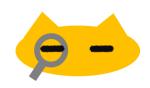


要求+HAZOPガイドワードにより、Top事象を効率よく導出する。

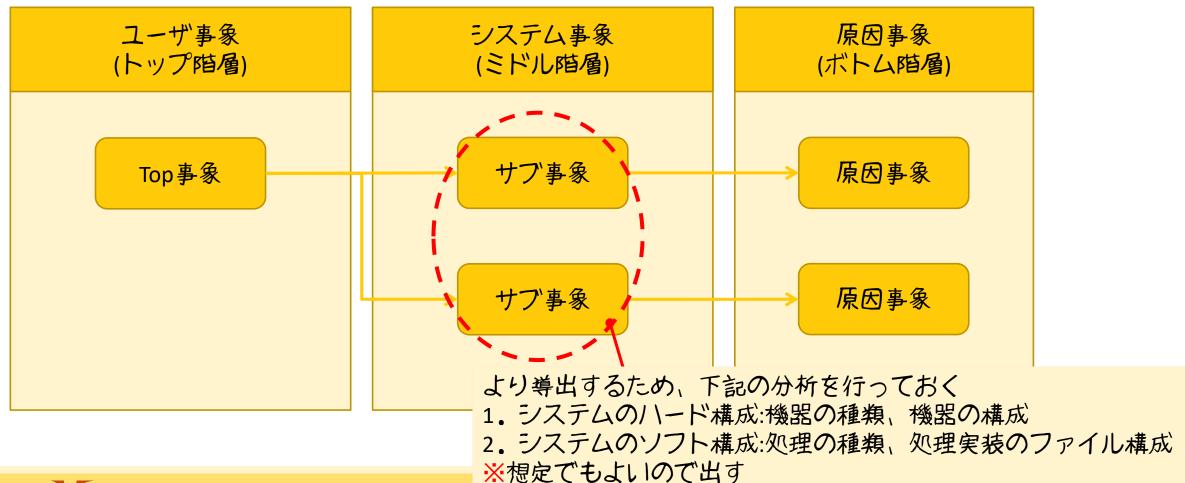




# フォールトツリー作成方法



また他にハードとソフトの分析を行ってから作成する。

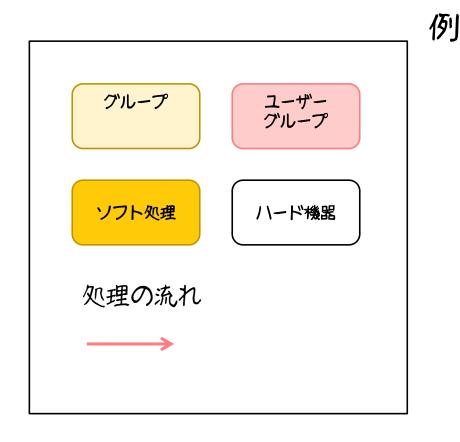


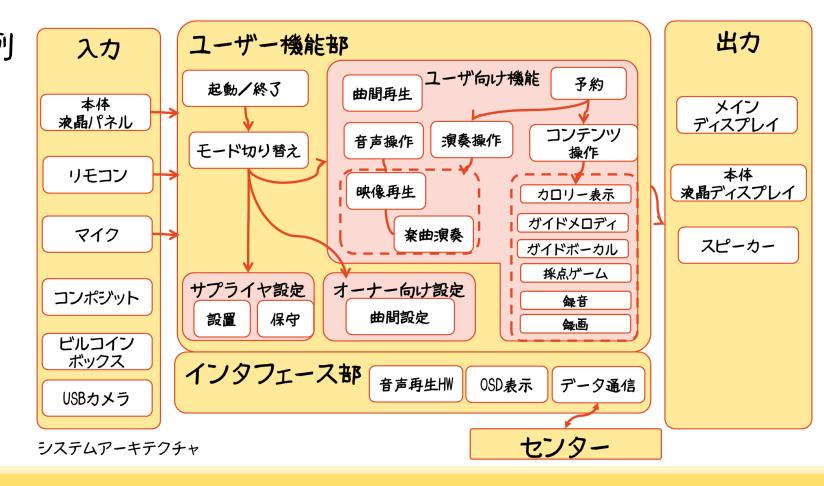


# ハードの分析:機器の構成



### ハードウェア機器の構成を考える。





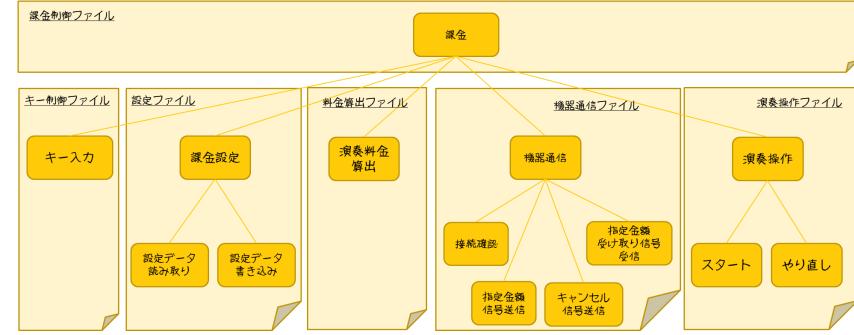
## ソフトの分析:処理実装のファイル構成



ソフトウェア処理実装のファイル構成を考える。

ファイル 課金制御ファイル 処理 ソフト処理 関連

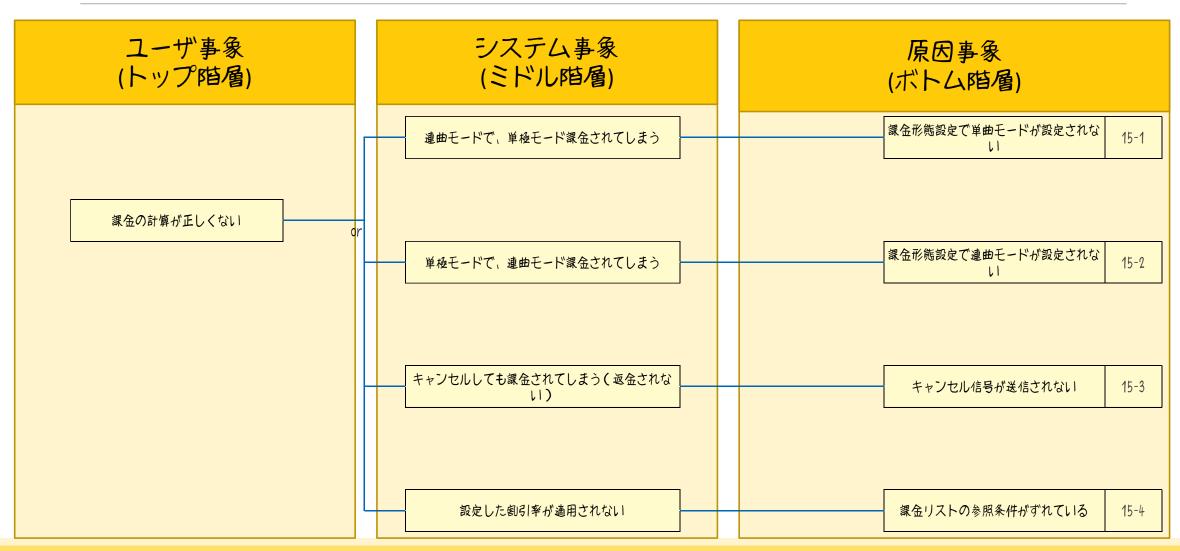
例



※可能であれば、処理の流れも書く。



# フォールトツリーフレームサンプル・ラー





# テストアーキテクチャ(テスト設計方針) ラー



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 2 計方針を合意する。

#### 合意事項:

- テストにおける対応者
- テストの重み(カバレッジ)
- テストタイプ(対応方法)

テス	<u>、卜対象</u>	テスト設計方針	テスト実施者
事象	顕在化しやすい事象	<b>ユーザーテスト</b>	ユーザー
*	顕在化しにくい事象	<b>→</b> フォールトベースドテスト	QA担当
ロード	コードの正しさ	ユニットテスト	開発者

				テストタイプ※	1	
		データ	条件	シナリオ	イベント /タイミング	構成
せづけ	軽 最低限のテスト	有効/無効	重複結果なし	CO	0-switch	1-wise
、や重の、	中	+同値クラス	無効なし	C1	セル網羅	2-wise
テスト	重	+境界値	完全	C2	1-switch	3-wise

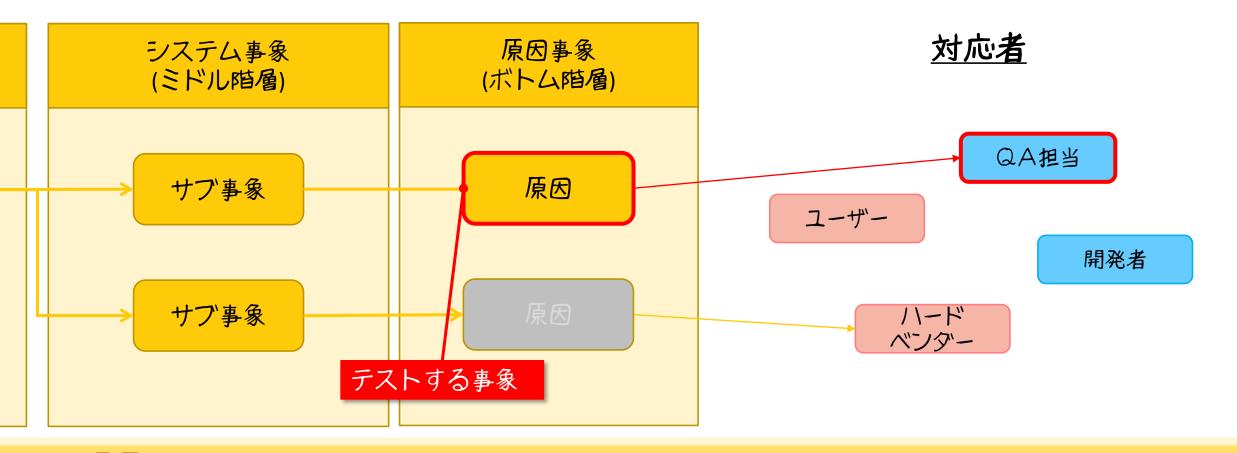
				テストタイプ		
		データ	条件	シナリオ	イベント /タイミング	構成
值低減度	軽 最低限のテスト	31.00%	25.00%	56.76%	61.67%	23.57%
ク値低	中	71.00%	45.00%	77.35%	83.89%	80.71%
-17	睡	95.00%	95.00%	95.00%	95.00%	95.00%



## 対処する事象(フォールト)の選定と対処方法割り当て



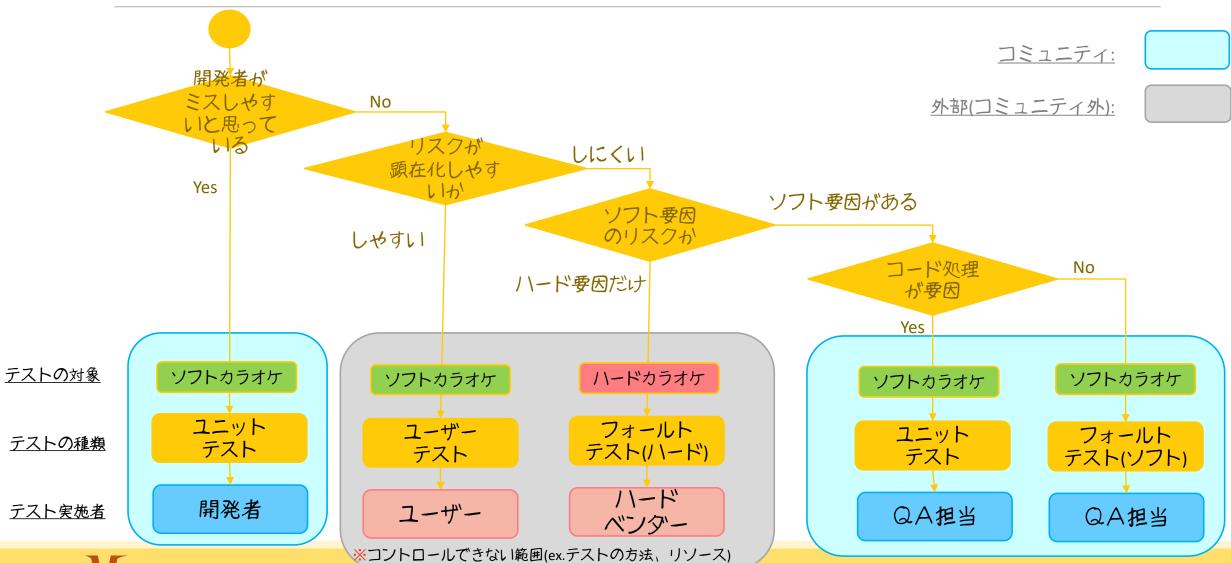
QAが対処すべきかを合意する。





# 事象ごとのテスト担当決めフロー



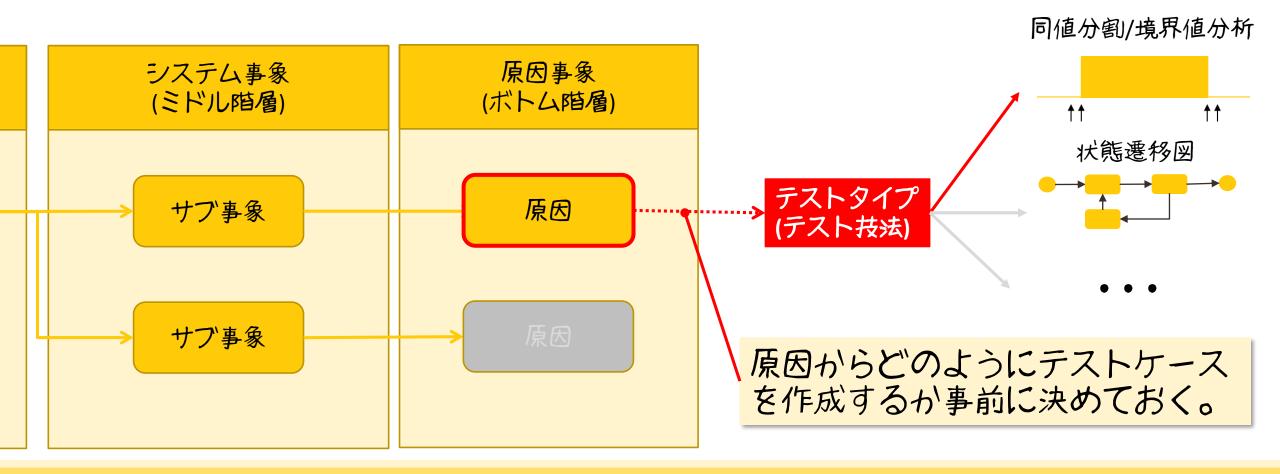




## 対処する事象(フォールト)の選定と対処方法割り当て



QAが対処すべき事象に対処方法(テストタイプ)を割り当てる。

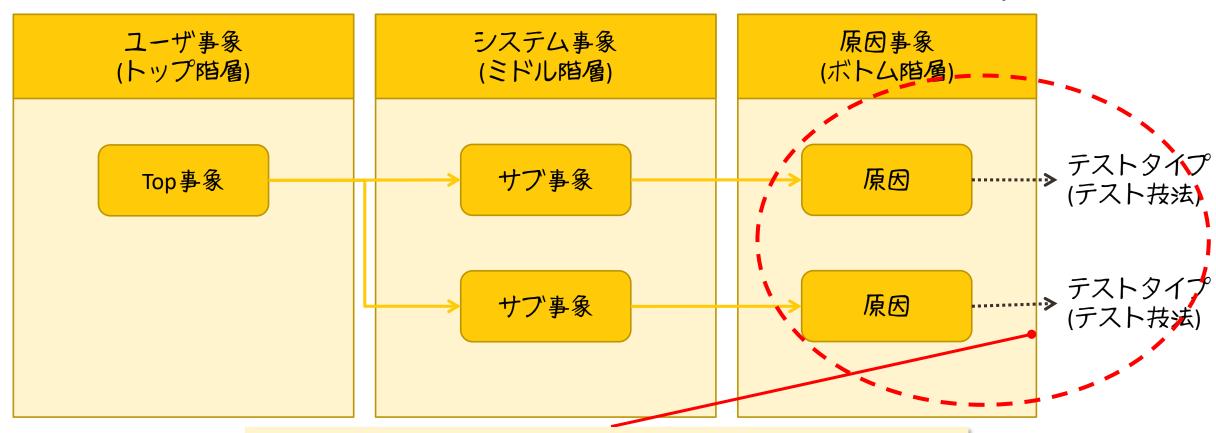




## テストケース作成方法



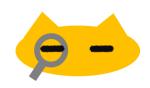
原因からどのようにテストケースを作成するか決めておく。



原因に対して、必要があれば、新たなテストタイプを定義する。



# テストタイプ⇔テスト技法対応表



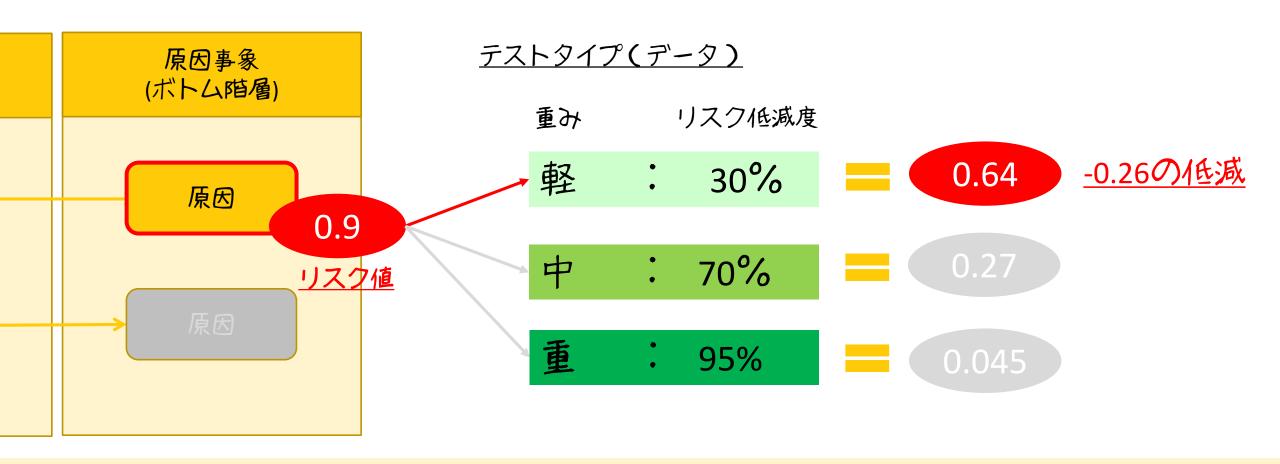
テストタイプ	(キーワード)	テスト技法		(テストケース導出モデル)
<ul><li>データ (領域、境界、サイズ)</li><li>性能</li></ul>	テストタイプ+ 抜け/漏れ ズレ/跳び 過剰/不足 矛盾/衝突 未反映/初期値/なし	同値分割 境界値分析	例	<u>††</u>
<ul><li>流れ (シナリオ、処理)</li></ul>	同上	フローチャート	例	
• イベント/タイミング (状態、モード、遷移)	同上	状態遷移図/表	例	
• 構成 (環境、機器、モジュール)	同上	分類ツリー法	例	
• 条件 (入出力、制約)	同上	デシジョンテーブル	例	1 2 3 4 A F F T T B F T F T Result O O O



### 対処する事象(フォールト)の選定と対処方法割り当て



テストタイプの重みを割り当て、事象全体の「やばさ」を下げる。





# フォールトツリー図サンプル



	ĪD		実装ファイル	リスク値	対象カテゴリ	<u>対処者</u>	対処方法	仮重	リスク低滅率	対処後のリスク値
	設定した演奏料金額が反映されない 14-1	╟	設定ファイル	0.9000	ソフト	QA	流れ	軽	56.76%	0.3892
or <u>before</u> 1.8000 <u>after</u> 0.43416 リスク修滅数 1.3658	設定しても演奏料金額が特定の料金 に反映されてしまう		設定ファイル	0.9000	ソフト	QA	データ	重	95.00%	0.0450

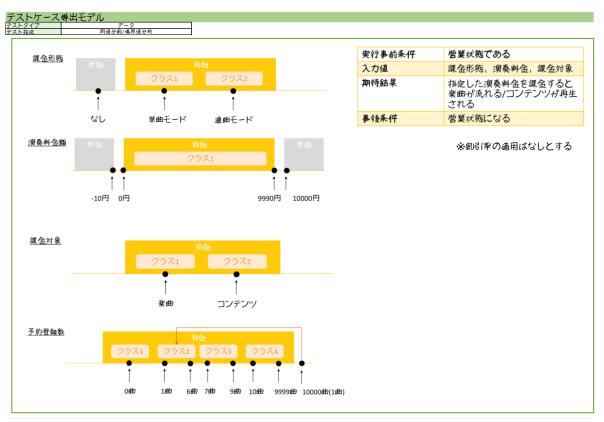
実装ファイル	<u>リスク値</u>	対象カテゴリ	<u>対処者</u>	<u>対処方法</u>	ん重	リスク低滅率	対処後のリスク値
設定ファイル	0.9000	ソフト	QA	流れ	軽	56.76%	0.3892
設定ファイル	0.9000	ソフト	QA	データ	重	95.00%	0.0450
リスク特定		リスク分析		リスク	フ評価		



# テストケースサンプル



#### テストケース導出モデルからテストケースを作成する。



テストケース等出モデル

テストケース	Z				
194			重		
ドレッジ		効/無効+同	「値クラス+#	類相值	
施テストケース数	女		168		
ストケースNo.		入力条件	「事前状態)		
.,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	課金形態	演奏料金		予約登録数	(1)
	1連曲モード	-10円	コンテンツ	1曲	設定
	2 単曲モード	-10円	コンテンツ	1曲	設定
	3連曲モード	-10円	コンテンツ	10000曲	設定で
	4なし	10000円	楽曲	9999曲	設定でき
	5連曲モード	10000円	コンテンツ	10000曲	設定できた
	6なし	0円	コンテンツ	6曲	課金された
	7 単曲モード	10000円	楽曲	9曲	設定できな
	8 単曲モード	0円	コンテンツ	7曲	指定金額で される
	9 単曲モード	0円	楽曲	6曲	指定金額で される
	10 連曲モード	0円	コンテンツ	7曲	指定金額で される
	11 単曲モード	9990円	楽曲	0曲	指定金額で記される される 神足金額で記
	12 連曲モード	9990円	コンテンツ	10000曲	<b>神史 金額</b> しま
	13なし	9990FJ	楽曲	6曲	課金されな
	14 単曲モード	0円	コンテンツ	9999曲	指定金額で記される
	15なし	-10円	コンテンツ	6曲	設定できない
	16 単曲モード	10000円	楽曲	0曲	設定できない
	17 単曲モード	10000円	コンテンツ	9999曲	設定できない
	18 単曲モード	-10円	コンテンツ	0曲	設定できない 和定位額し国
	19 単曲モード	9990円	コンテンツ	10000曲	ヤガス
	20 なし	0円	コンテンツ	7曲	課金されない
	21 単曲モード	-10円	楽曲	9999曲	設定できない
	22 なし	-10円	コンテンツ	10000曲	設定できない
	23 連曲モード	0円	楽曲	6曲	指定金額で される
	24 連曲モード	-10円	コンテンツ	9曲	設定できない
	25 単曲モード	0円	楽曲	1曲	指定金額で課 される
	26 なし	9990円	コンテンツ	1曲	課金されない 神変金額であ
	27 単曲モード	9990円	コンテンツ	6曲	神変・重視 ( i
	28 なし	0円	楽曲	10000曲	課金されな
	29 なし	9990円	楽曲	7曲	課金されな

テストケース

• • •



# リスク•テスト全体を俯瞰



「やばさ」に対して、許容範囲になっているか、何をテストするかを俯瞰する。

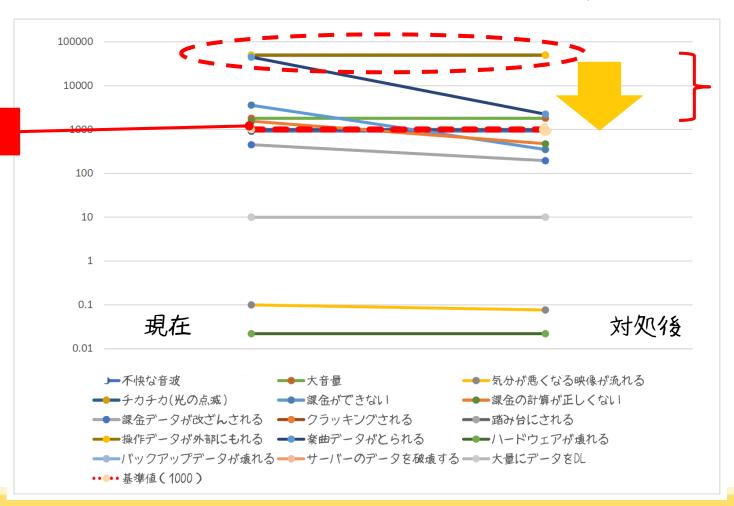
				かは	ざさ			テストタイプ						
カテゴリ	リスクカテゴリ	No. フォールト項目	Rank	現在	⇒ Ran	k		低減数	低减率	データ	流れ	条件	イベント/タイミ ング	構成
顕在化しにくい	身体的リスク	1 不快な音波	9	917.8	⇒	7	917.8	0	0.00%					
		2 大音量	6	1809.32	⇒	5 18	809.32	0	0.00%	-				
		3 気分が悪くなる映像が流れる	14	0.10	⇒	14	0.10	0	0.00%	-				
		4 チカチカ(光の点滅)	8	1000	⇒	6	1000	0	0.00%	-				
	金銭的リスク(課金)	14課金ができない	5	3600.00	⇒	9 3	352.08	3247.92	90.22%	0	0	0	0	
		15課金の計算が正しくない	7	1544.58	⇒	8 4	472.5	1072.08	69.41%	-			0	0
		16課金データが改ざんされる	10	450.00	⇒	10 1	194.58	255.42	56.76%	-	0			
	社会的要請	17 クラッキングされる	1	50000	⇒	1 5	50000	0	0.00%					
		18 踏み台にされる	1	50000	⇒	1 5	50000	0	0.00%	-				
		19 操作データが外部にもれる	1	50000	⇒	1 5	50000	0	0.00%					
		20 楽曲データがとられる	4	45000	⇒	4	2250	42750	95.00%					
	物理的破壊リスク	21 ハードウェアが壊れる	15	0.022	⇒	15 (	0.022	0	0.00%					
		22 バックアップデータが壊れる	11	10	⇒	11	10	0	0.00%					
		23 サーバーのデータを破壊する	11	10	⇒	11	10	0	0.00%					
		24大量にデータをDL	11	10	⇒	11	10	0	0.00%					
顕在化しやすい	うまく使えないリ スク	5ハウリング	-		⇒									
		6音が出ない			⇒					-				
		7映像が出ない			⇒									
		8字幕がずれる			⇒									
		9途中で止まる			⇒					-				
		10 テンポが変わる			⇒									
		11 字幕が切れる			⇒					-				
		12 となりの音が割り込む			⇒					-				
		13 起動しない			⇒									
				204351.82	⇒	1!	57026.4	47325.4	23.16%					



# リスク•テスト全体を俯瞰



「やばさ」がどうなるかをそのリリース毎に確認していく。



対応必須なもの



基準値:1000



テスト開発コンセプト

「製品のリリースを止めるバグを許容範囲まで減らす!」

- 製品のリリースを止める事象およびバグをどう特定するか?
- 2. テストで対処する/しない事象はどのように決めるか? ⇒
- 3. 対処するためにどんなテストをどれくらい行うか? ⇒





テスト開発コンセプト

「製品のリリースを止めるバグを許容範囲まで減らす!」

- 1. 製品のリリースを止める事象およびバグをどう特定するか? ⇒フォールトツリー図を用いた事象の分解
- 2. テストで対処する/しない事象はどのように決めるか? ⇒
- 3. 対処するためにどんなテストをどれくらい行うか? ⇒





テスト開発コンセプト

「製品のリリースを止めるバグを許容範囲まで減らす!」

- 1. 製品のリリースを止める事象およびバグをどう特定するか? ⇒フォールトツリー図を用いた事象の分解
- 2. テストで対処する/しない事象はどのように決めるか? ⇒被害金額とコード変更履歴からのリスク値による優先順位
- 3. 対処するためにどんなテストをどれくらい行うか? ⇒





テスト開発コンセプト

「製品のリリースを止めるバグを許容範囲まで減らす!」

- 1. 製品のリリースを止める事象およびバグをどう特定するか? ⇒フォールトツリー図を用いた事象の分解
- テストで対処する/しない事象はどのように決めるか?
   ⇒被害金額とコード変更履歴からのリスク値による優先順位
- 3. 対処するためにどんなテストをどれくらい行うか? ⇒テストタイプのリスク低減表による事象の許容範囲への調節





テスト開発コンセプト

「製品のリリースを止めるバグを許容範囲まで減らす!」

- 1. 製品のリリースを止める事象およびバグをどう特定するか? ⇒フォールトツリー図を用いた事象の分解
- 2. テストで対処する/しない事象はどのように決めるか? ⇒被害金額とコード変更履歴からのリスク値による優先順位
- 3. 対処するためにどんなテストをどれくらい行うか? ⇒テストタイプのリスク低減表による事象の許容範囲への調節



## 以上で、本提案のご説明を終わります。 ご清聴ありがとうございました。

