

テスト設計コンテスト '16

ASTER 通信カラオケシステム



チーム

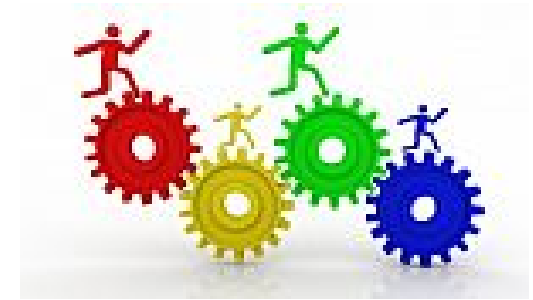
FSTWG

2016年3月9日

- 1. チーム紹介
- 2. 前提
- 3. 全体像
- 4. 各プロセスのポイント
 - テスト計画
 - テスト要求分析
 - アーキテクチャ設計
 - テスト詳細設計
- 5. まとめ

チーム紹介

■ チーム名：FSTWG



- F A関係の仕事をしている有志のメンバーです
- メンバー紹介：「あなたにとってカラオケとは」
 - 石川（リーダー）：ストレス発散のつもりが翌日に疲れを残すモノ
 - 中村：発声練習装置
 - 小森：現実逃避したい時は宙船で浪漫飛行！
 - 清水：ボックスよりナイト派
 - 横田：元カノを思い出すトリガ
 - 星野：気持ちよくシャウト＆最悪ホテル代わり
 - 溝上：敬愛するサザンのKKに近づく手段
 - 岩田：唯一マイクを持つ機会

今回のコンテストでテスト道の黒帯目指します

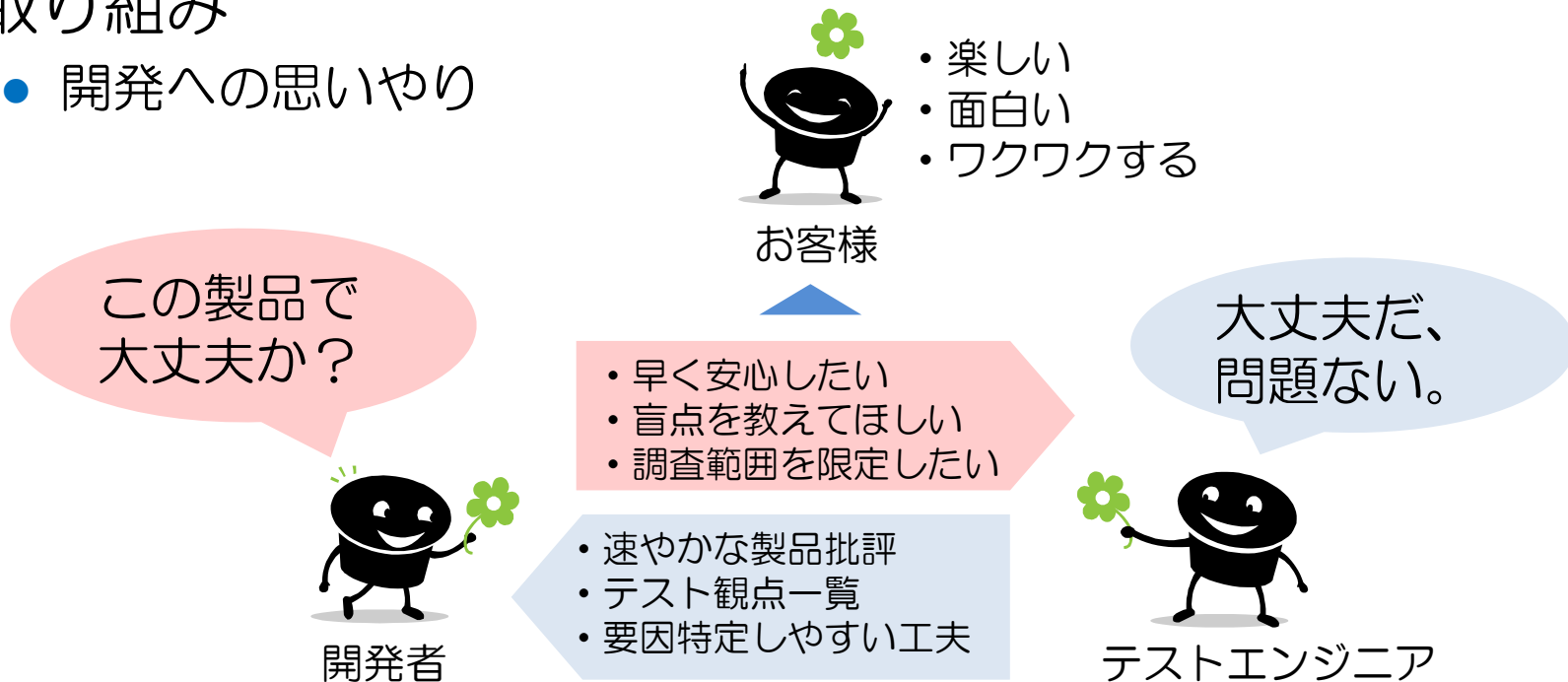
前提

■ 我々の立場

- カラオケ開発メーカーに所属するテストチーム
- 現在開発が始まった段階で開発と平行してテスト設計を進める
- テストレベルはシステムテストとする

■ 取り組み

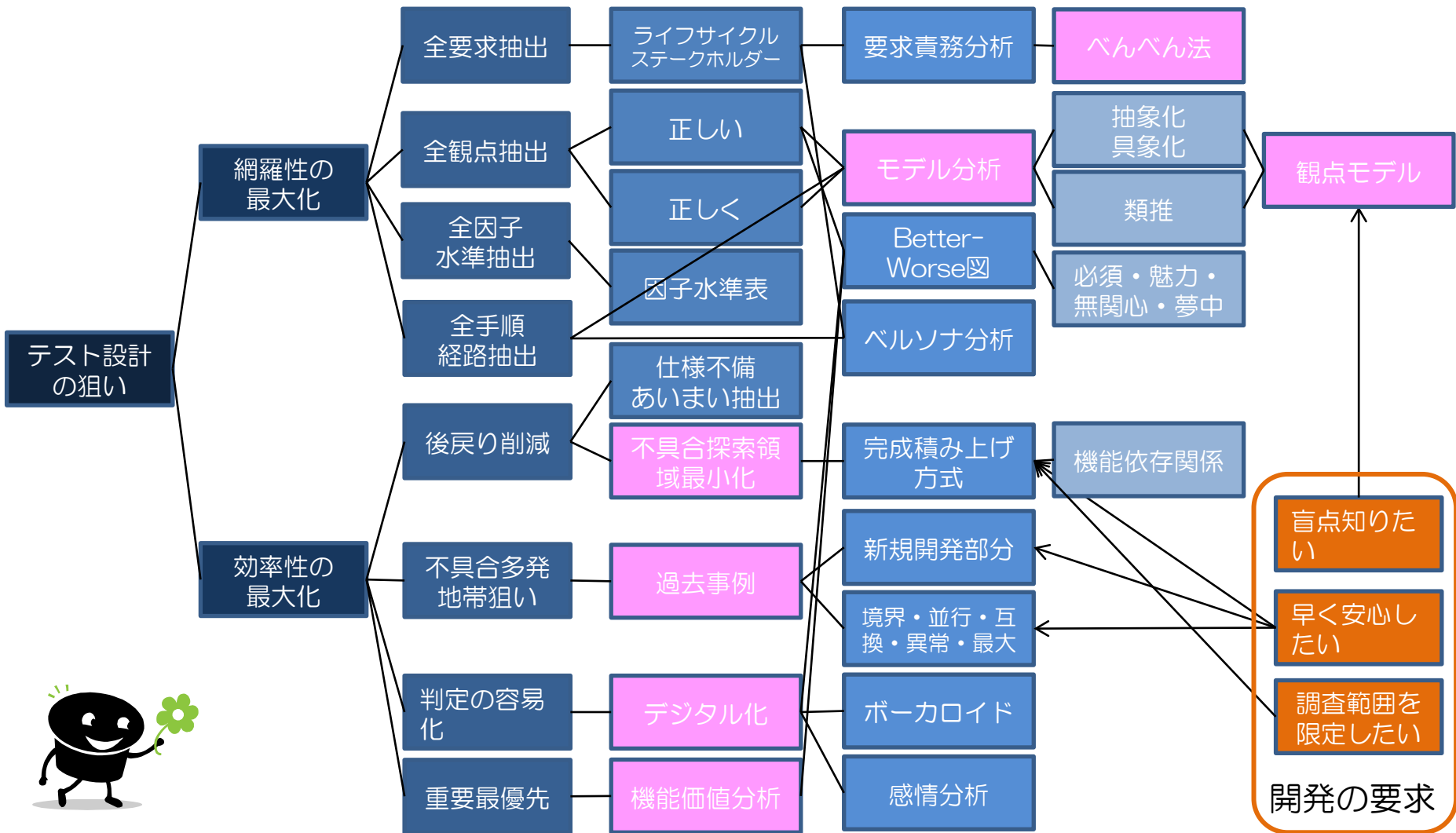
- 開発への思いやり



開発とテストとのシナジー効果で三方が幸せな関係を構築する

設計の全体像

目的 ← → 手段



テストと開発双方の要求を満たす設計を実現する



テスト計画

テスト設計の流れ

テスト要求分析

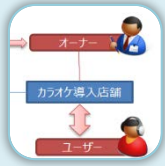
正しいものを作ってる?



商品価値



商品の価値がわかるよ



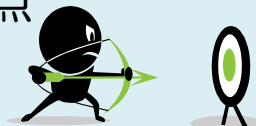
利害関係者抽出

要望抽出



関係者の望みを分析するよ

観点



テストの切り口を見つけるよ

正しく作ってる?



モデルで可視化
曖昧性除去するよ



機能境界



利用手順



状態

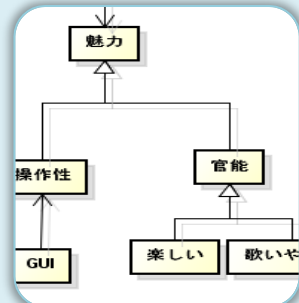


データ構造



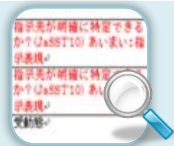
並行動作

脅威



全観点

不備はない?

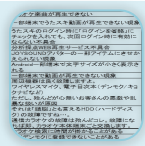


仕様不備抽出



あいまいや
矛盾をなくすよ

予測



過去状況確認



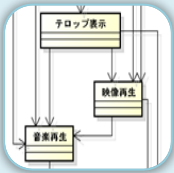
過去に学ぶよ

アーキテクチャ設計

順番



優先順位を決めるよ



機能依存



商品的魅力

境界



テスト課題に
対して構えるよ



構造構築



テスト手段



テスト単位分割

テスト詳細設計

技法



テストケース

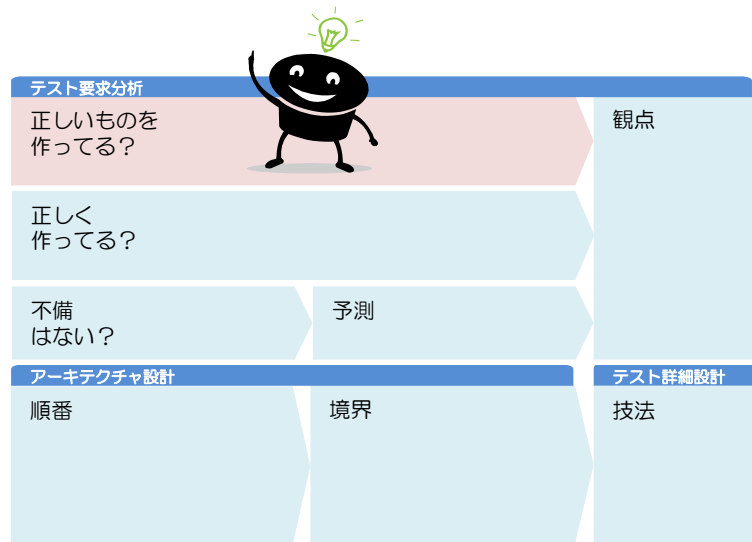


効率を考えてテスト
技法を適用するよ

正しいものと正しくの切り口から観点で合成し、アーキテクチャを構築する

テスト要求分析 1

正しいものを作っているか視点



商品価値判定

商品の価値がわかるよ



9/32

■ 手順

要件を元に年代、性別、責務別にアンケートを取る

		ペルソナ	BOX 20代女性		
ユーザー操作					
コンテンツ管理機能					
機能要求	機能名	有ればどうか	無いとどうか	理由	
1-1	新曲演奏	当然である	困る	必須	
	1-1-1	MIDI再生	当然である	困る	必須
	1-1-2	MP3再生	何とも感じない	何とも感じない	無関心
	1-1-2	MP3再生	何とも感じない	何とも感じない	無関心

下記の式で機能の魅力度を計算する

	O	A	M	I	R	S
属性1	x	x	x	x	x	x
属性2	x	x	x	x	x	x
...						

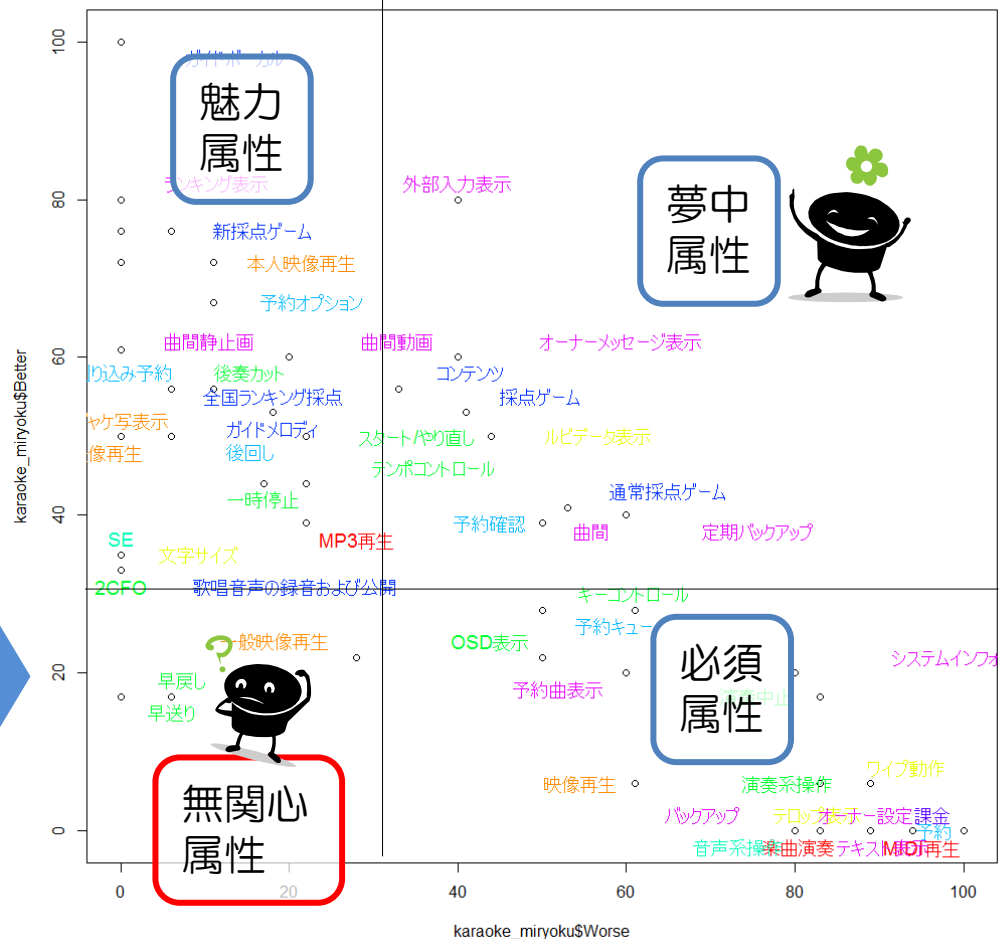
$$\text{Better} = \frac{A+O}{A+O+M+I}$$

$$\text{Worse} = \frac{M+O}{A+O+M+I}$$

魅力的属性 (A, O) と 一元的属性 (M, I) の比率を示す。


出所：『属性をより定量的に特徴づける方法』(Walden,1933)

要件毎にプロット (Better-Worseチャート)



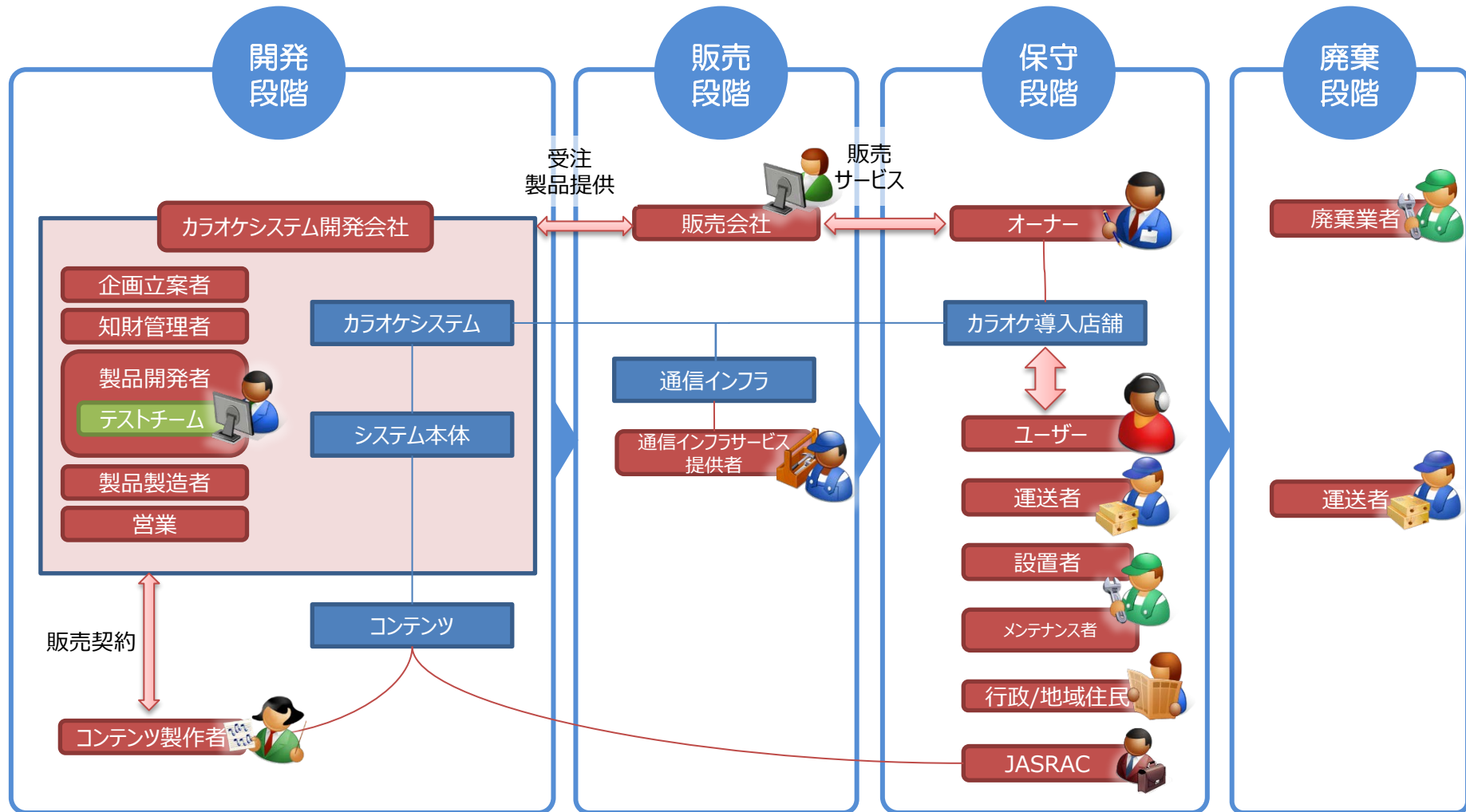
必須、魅力属性に機能が多く無関心属性が少ないので価値が高い

要求の源泉獲得

関係者の望みを分析するよ 

■ ステークホルダー

自分の立ち位置 **ステークホルダー** カラオケに必要なもの



製品ライフサイクルより考えるステークホルダーを洗い出す

テスト対象の視点拡張

■ 機能以外の発見

ステークホルダー	責務	要求大項目	あなたにとってカラオケシステム(機械)とは
ユーザー	カラオケを楽しむ	気持ちよく歌いたい	ストレス発散のための機械 楽しむための機械

テスト対象を「カラオケシステム」と捉えていると機能面しか見えない

ステークホルダーの要望を実現するモノ化する(名前付け)

名付けたモノにふさわしいかを考えることで、評価すべき視点が広がる



「べんべん法」って名前にしようかな




カラオケシステムのようにカラオケシステムでない、と考える

テスト観点

関係者の望みを分析するよ



■ 観点抽出とテスト対象選別

ステークホルダー	責務	要求大項目	要求(ステークホルダーがカラオケシステムに要求すること) 1. 機能から 2. 市場要望から 3. ビジネス視点から 4. 市場不具合から	あなたにとってカラオケシステム(機械)とは	テスト対象?	テスト観点		
						性能	パラメータ	魅力
ユーザー 	カラオケを楽しむ	気持ちよく歌いたい	音質が良い方がいい	ストレス発散のための機械 楽しむための機械	する	性能	パラメータ	魅力
					する	性能	互換性	周辺機器
					しない			
					する	機能	官能	処理
			歌いやすい					
			画質が良いほうがいい		する		処理速	負荷
			曲にマッチした画像がでてほしい		する	機能	機	タイミ

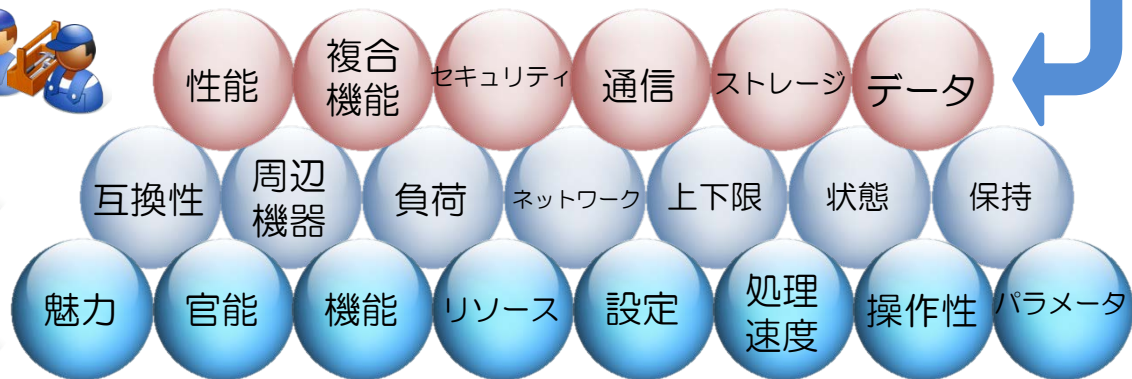
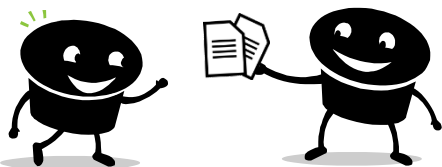
①責務を考える

②要求を考える

③別名を考える

④観点をみつける

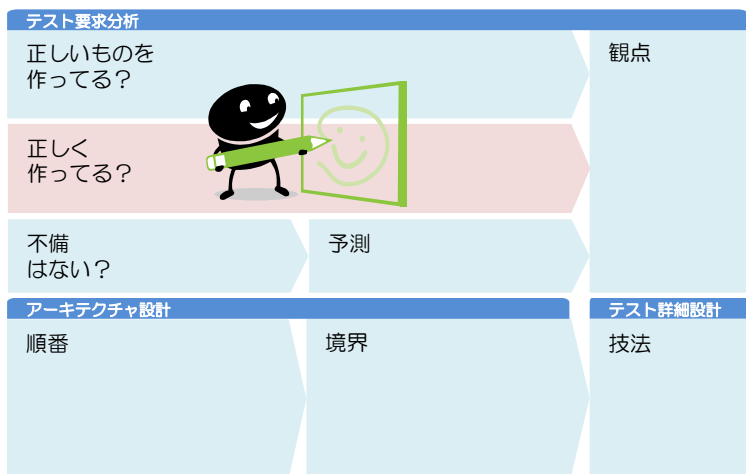
要求で技術的に困難なものは
次回の開発への要望として集約



ステークホルダーの要求ごとにテスト観点を抽出した

テスト要求分析 2

正しく作っているか視点

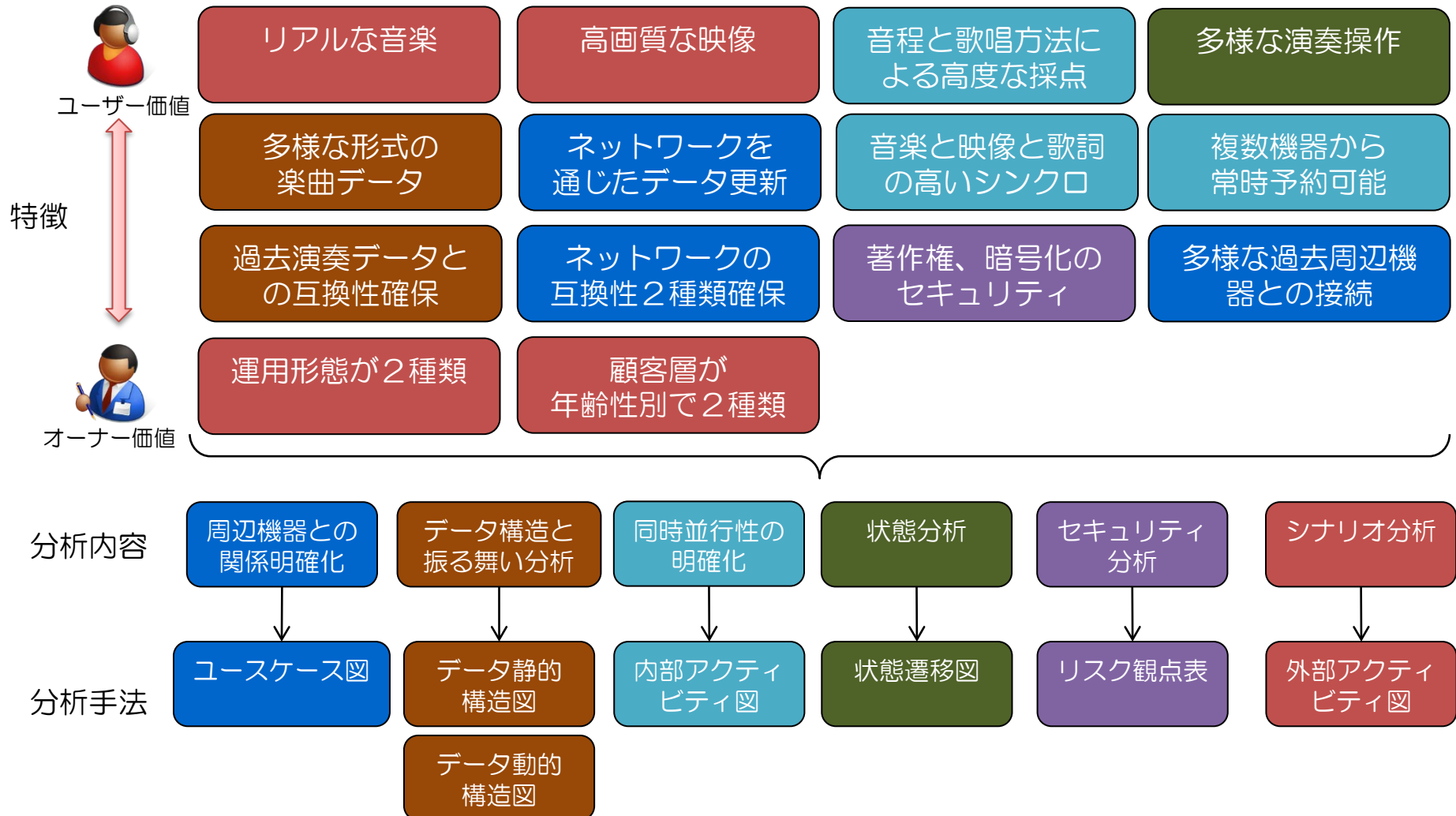


分析手法の選択

モデルで可視化、曖昧性を除去するよ



■ カラオケの特徴



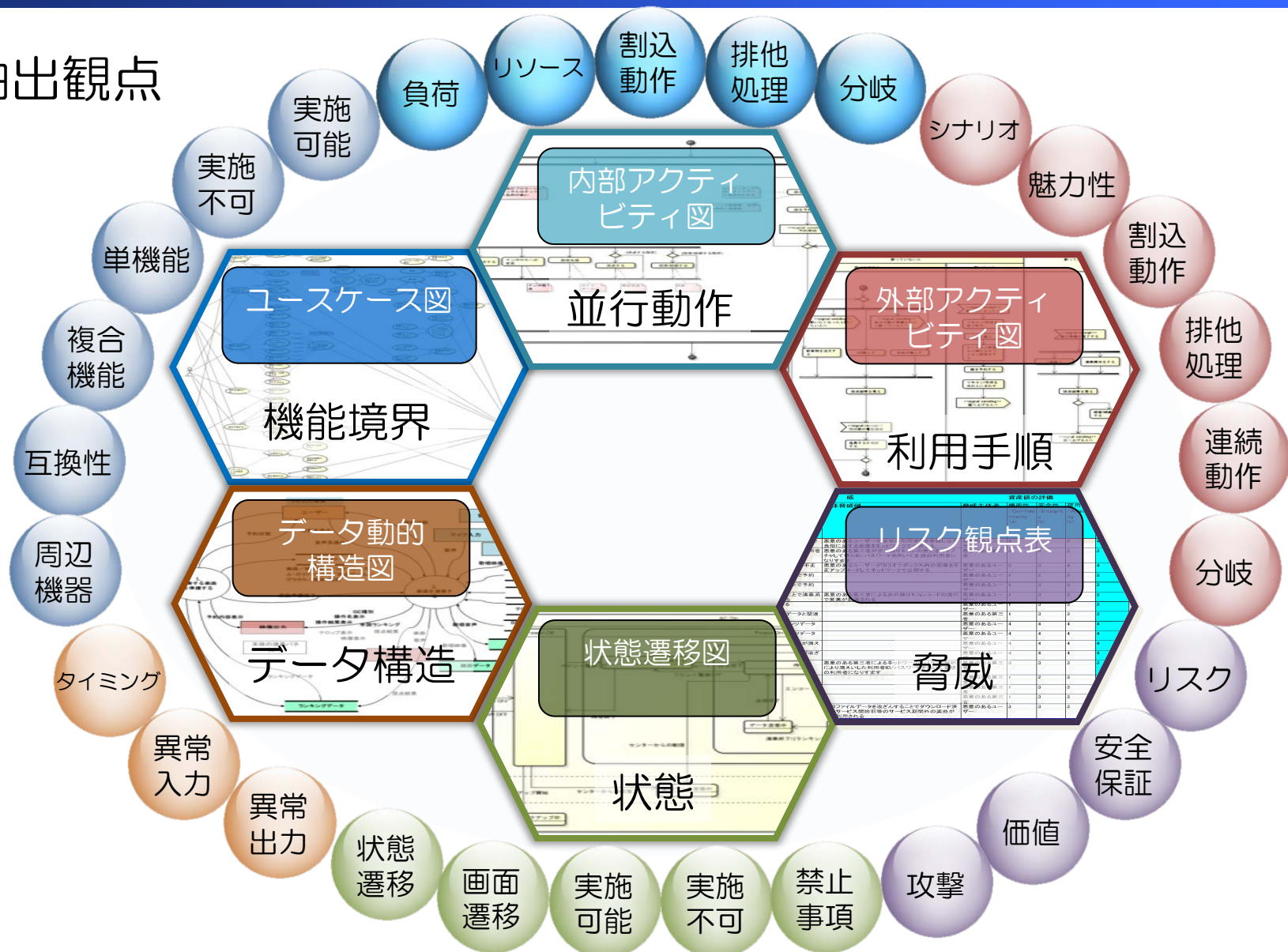
要件の特徴から分析すべきことを決める

要求分析

モデルで可視化、曖昧性を除去するよ



抽出観点




要求分析はモデルで整理し、理解促進と仕様の不備を見つける

テスト要求分析 3

仕様不備、予測



仕様不備・予測

■ あいまい分析 あいまいや矛盾をなくすよ 

要件のあいまいさ検出

文章のあいまいさには理解のズレが発生しやすく不具合につながるため
校閲機能を使い、特定の「その」や「～時」の文字を検出して抽出する（ツールで自動化）

Sheet1!A334	セル	連曲モードの場合でも演奏を中止し、次曲予約がある場合はその曲の	指示先が明確に特定できるか？(JaSST10) あいまい:指示表現
Sheet1!A349	セル	演奏中のその楽曲に対してのみ有効で、次曲演奏時には初期値に戻る。	指示先が明確に特定できるか？(JaSST10) あいまい:指示表現
Sheet1!A352	セル	標準ボタンを押下されると楽曲のテンポの初期値となる。	変動態
Sheet1!A355	セル	テンポボタン操作時には設定された値を OSD として画面に表示する。	指定された条件以外の動作については記述があるか？(JaSST10) あいまい:指示表現

不明なものは開発への
質問リストとして集約

あいまい

■ 予測（過去不具合・経験） 過去に学ぶよ 

カラオケの過去不具合

カラオケ楽曲が再生できない
一部端末でうたスキ動画が再生できない現象
うたスキのログイン時に「ログインを省略」にチェックを入れても、次回ログイン時に有効にならない不具合
分析種別WEB再生サービス不具合
JOYSOUNDサーバーの一部アイテムにきせかえられない現象
Android一部端末で文字サイズが小さく表示される
一部端末で動画が再生できない現象
周辺機器は良く故障しますよ。
ワイヤレスマイク、電子目次本(デンモク/クナなど)
ただし、殆んどが心無いお客様の悪戯や乱暴な扱いが原因
それは「頭脳」とも言えるHDD(ハードディスク)の故障ですね…
通信カラオケの故障は殆んどコレ。故障になれば、カラオケ本体壊れごとと交差します。
カラオケ検索に動画が掛かることがある
MV/デンモクに登録できないことがある

→主にネット
ワーク障害、
機器破損

過去の経験

互換性

性能

脅威

OSの移植

- 性能低下、互換、GUI
- OS吸収層の不具合
- OSの不具合

→要因が表面化しにくい

セキュリティ

- 楽曲データの流出
 - 海賊版のインストール
 - 楽曲解禁日破り
- 既存手口の確認が大事

あいまいさによる後戻りを減らす。過去不具合より未来を予測する。

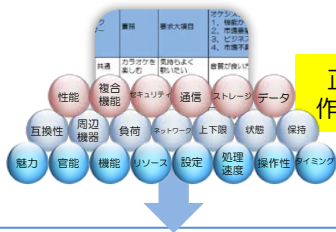
テスト要求分析 4

テスト観点の合成

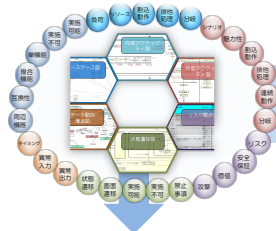


テスト観点合成

テストの切り口を見つけるよ

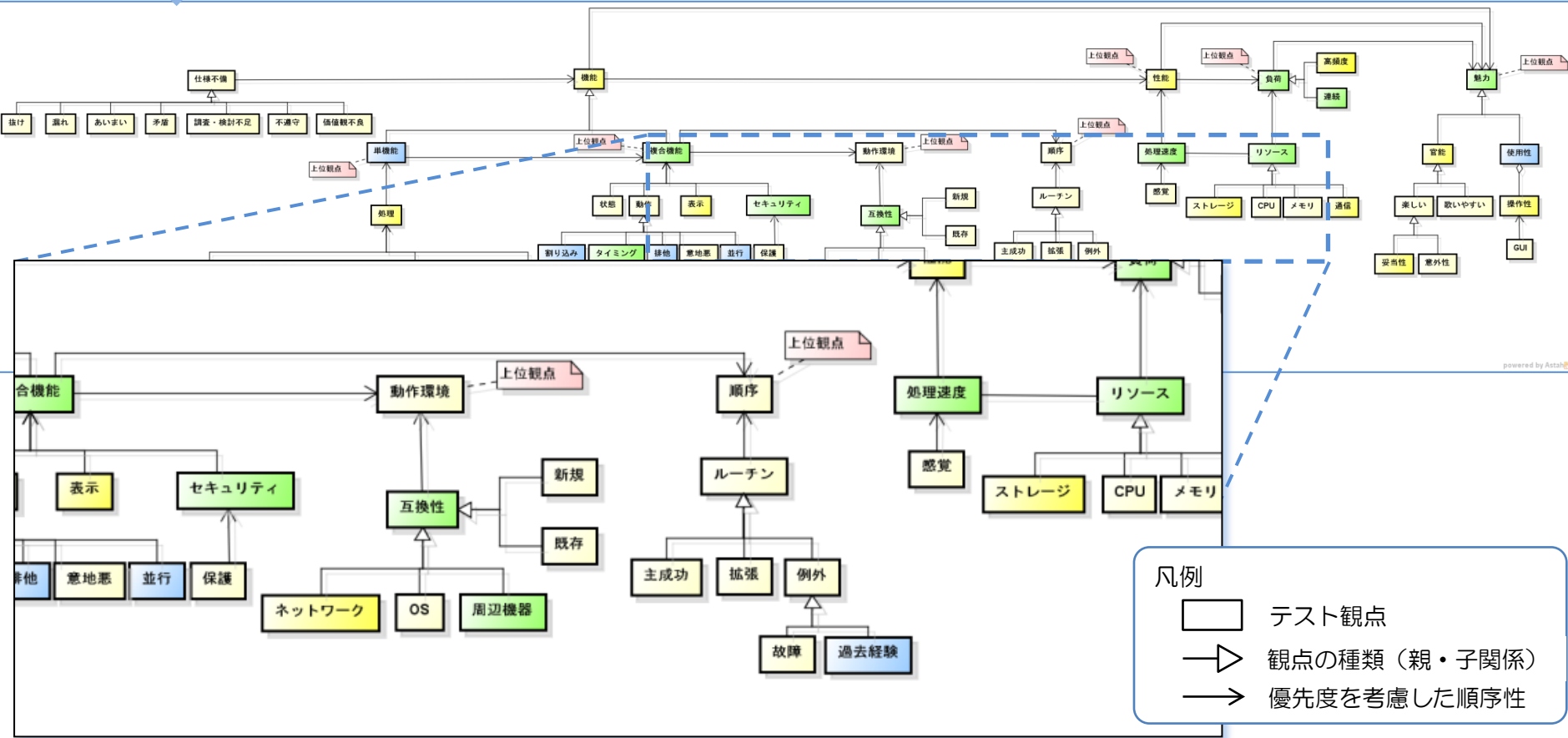


正しいものを作っているか？



正しく作っているか？

1. 観点が同じ種類はまとめる。
2. 不足している観点を追加する。
3. 関係のあるところは関連線一をひく。
4. 流れを→で順序付ける。



凡例

- テスト観点
- 観点の種類 (親・子関係)
- 優先度を考慮した順序性

すべてのテスト観点の階層化と抽象化を行い、新たな観点を抽出し反映する



テストアーキテクチャ設計 優先順位



機能の依存関係

優先順位を決めるよ



21/32

■ 手順

1. 要件定義書、ユースケース図、カラオケの進化を元に作る

2. 機能全体に関わるものは右レーンに振り分ける

3. 機能を重要性で三階層に分ける

カラオケ進化

2011年
ひとりカラオケ

2003年
ブロードバンド化

2000年
HDD搭載

1994年
採点システム

1993年
CD動画カラオケ

1992年
通信カラオケ

1987年
カラオケBOX普及

1984年
オートチェンジャー

1982年
LD (映像つき)

1977年
8トラ (音声のみ)

Ⅲ:応用機能

配信サーバーにデータ送信

歌う姿の録画

カメラ入力

歌の録音

曲間動作

Ⅱ:基本機能

ランキング

バックアップ

データ受信

SE

採点

マイク入力

曲の予約

Ⅰ:コア機能

テロップ表示

映像再生

演奏操作

音楽再生

スクリーン出力

音声出力

映像出力

構造の下層から順にテストをするため機能依存関係を整理する

機能の魅力関係

優先順位を決めるよ



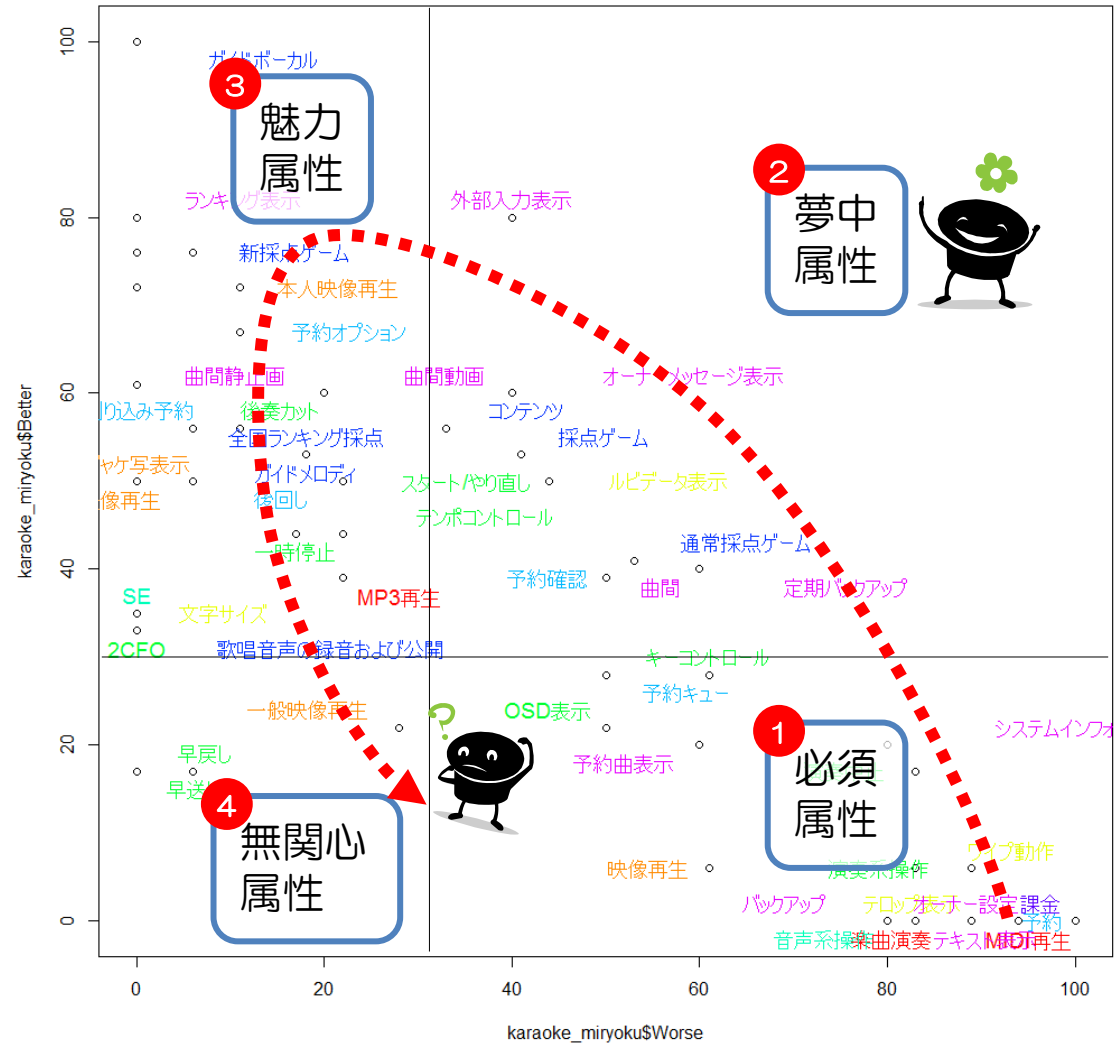
■ 手順

機能が多いと順位をつけるのが大変

お客様が重要視する価値に着目する

大事なものを優先してテストをし、万が一スケジュールが遅れそうになっても大事にならない。

価値を判定した図より左図の順で実施する。



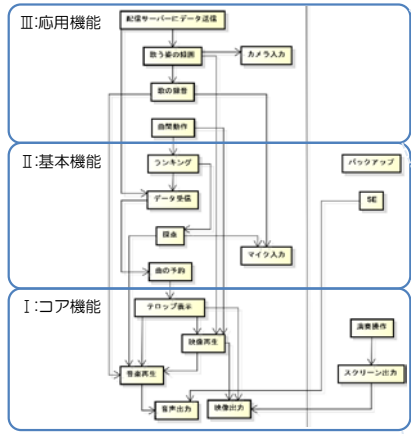
求められる機能の魅力度の必須→夢中→魅力→無関心で優先順位を決定

テストアーキテクチャ設計 境界



テストアーキテクチャ設計

課題に対し構えるよ



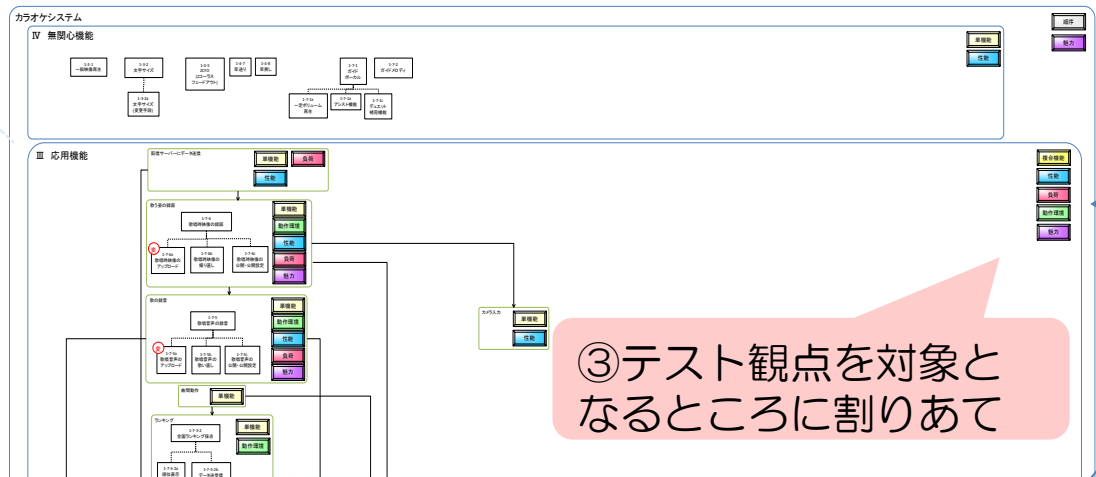
機能依存

①階層は機能依存のまま



機能魅力関係

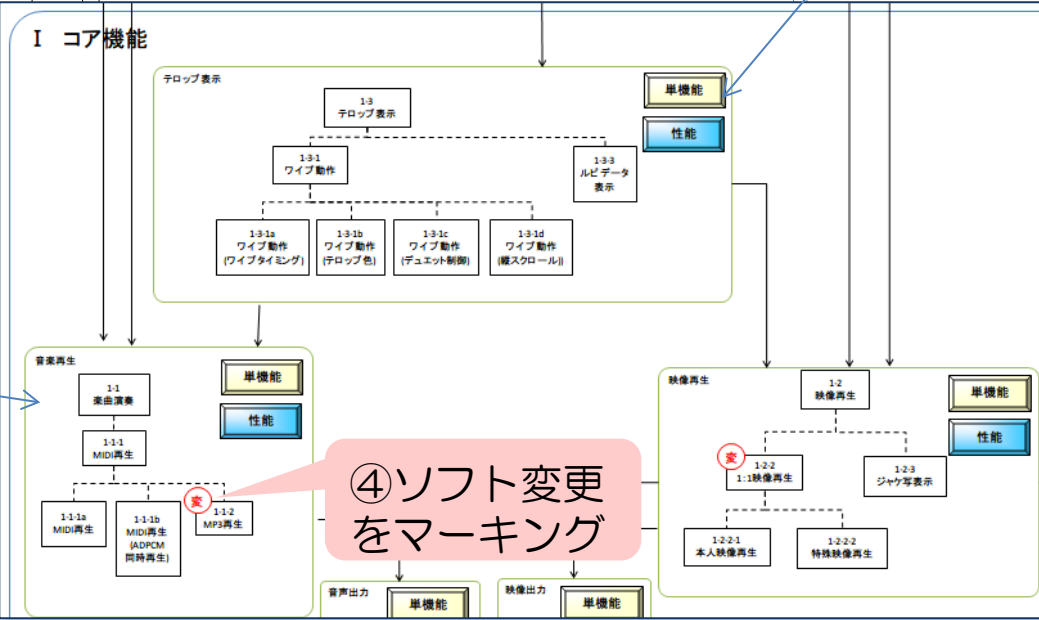
②各大項目のテスト順を魅力から決定



③テスト観点を対象となるところに割りあて



テスト観点



④ソフト変更をマーキング

テストの全体像を作り、上下関連、優先順位、必要な観点を可視化した



テスト手段検討

■ 発想を豊かにテスト手段と対象機能を考える

テスト対象 ステークホルダー要求抽出の "カラオケシステム(機械)"より	テスト観点						関係する主な機能	測定方法 (将来の測定方法も含む)
カラオケシステム(機能) 	単/複機能	動作環境	順序	性能	魅力	負荷	とくになし(全ての機能)	<ul style="list-style-type: none"> 録音、<u>ボーカロイド</u>による採点の正確性測定 <u>バイオメトリクス</u>による指標判定 一対比較表による比較測定 機能についての満足度アンケート(全ステークホルダー対象) OK/NG判定による機能・性能その他観点の判定
みんなで盛り上がる機械 	単/複機能	魅力					とくになし(全ての機能)	<ul style="list-style-type: none"> <u>動きのシンクロ率測定、ハイトッチの回数測定、拍手回数測定</u> <u>画像認識型感情計測で幸福度を測定</u> <u>満足度のアンケート(ユーザー対象)</u>
自己陶醉するための機械 	単/複機能	魅力	性能				レイヤー1 音声出力、音楽再生、演奏操作 レイヤー2 採点、マイク入力、 レイヤー3 歌う姿の録画 レイヤー4 -	<ul style="list-style-type: none"> <u>歌っている間の目を閉じている時間の測定</u> 画像認識型感情計測で幸福度を測定 満足度のアンケート(ユーザー対象) 各レイヤーの機能・性能テストにてOK/NG判定
金儲けの機械 	単/複機能	魅力					レイヤー1 スクリーン出力 レイヤー2 オーナー設定 レイヤー3 曲間動作 レイヤー4 -	<ul style="list-style-type: none"> 満足度のアンケート(オーナー対象) 各レイヤーの機能テストにてOK/NG判定
歌の練習装置 	単/複機能	魅力	性能				レイヤー1 音声出力、音楽再生、演奏操作 レイヤー2 採点、マイク入力、 レイヤー3 ランキング レイヤー4 -	<ul style="list-style-type: none"> 満足度のアンケート(ユーザー対象) 各レイヤーの機能・性能テストにてOK/NG判定
暇つぶしの機械 	単/複機能	魅力					とくになし(全ての機能)	<ul style="list-style-type: none"> <u>予約時間と実使用時間の調査、歌っている時間の割合調査</u> 満足度のアンケート(ユーザー対象)

判断を容易にするよう見える化、デジタル化できるテスト手段を検討

テスト手段 効率を考えて手段を選択するよ



■ 機能テスト（曲採点）

曲の採点には音程やリズムの一致性と歌唱法の評価があるので、手法について分けた

	人の声（生歌）		人の声（録音）		ボーカロイド	
内容	人が直に歌う 		人が歌ったものを録音して再生する 		ボーカロイドに歌ってもらおう 	
再現性	×		○		○	
準備	○	人材が必要	△	録音必要	×	データの作成
音程	△	人による	△	人による	○	完璧
歌唱法	△	人による	△	人による	○	調教次第
自動化	×		○	何回でもできる	○	何回でもできる
実施時期		最終評価		最終評価前		採点機能確認時、最終前




ボーカロイドを使用して繰り返し再現可能な自動テストを採用した

テスト手段 効率を考えて手段を選択するよ



■ 官能テスト（歌いやすさ、楽しさ）

官能的なテストを定量化するために、評価内容にあわせて組み合わせて使用する

	一対比較		バイオメトリクス	
内容	2つのうちどちらが良いかを選択する比較方法 		被験者の心拍数で興奮度合いと、カメラで笑顔の回数と時間を計測する方法  	
厳密さ	◎	2つの比較は容易	○	
真偽度合い	○		◎	体は正直
準備	×	対象を2種類用意する必要あり	△	
万能性	△		△	
テスト時期		サイクル1~4		最終評価前の歌の部分

Anger	0.0000
Contempt	0.00001
Disgust	0.0000
Fear	0.0000
Happiness	0.9999
Neutral	0.00009
Sadness	0.0000
Surprise	0.0000

出処：Microsoft Emotion API

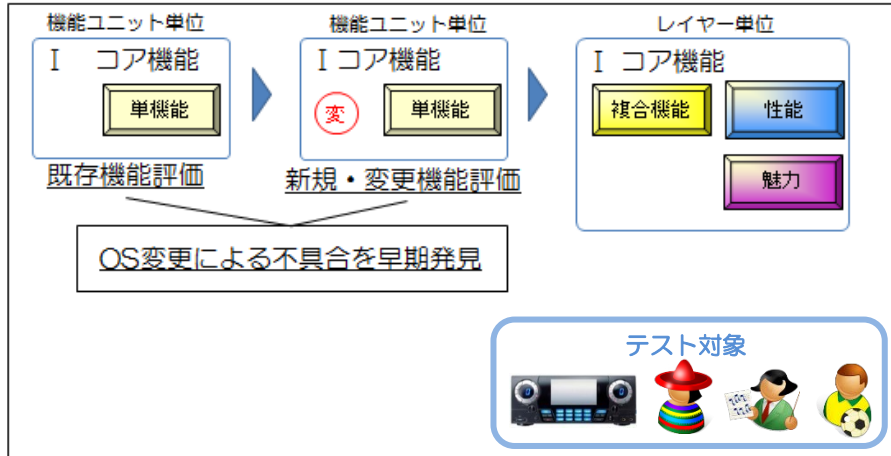
あいまいになりがちな評価も数値化して判断できるように工夫した

テストアーキテクチャ

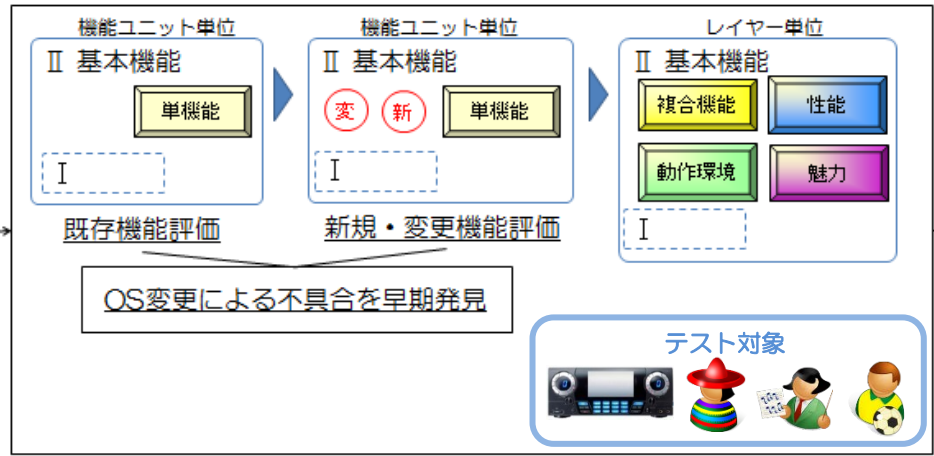
テスト対象アイコン説明



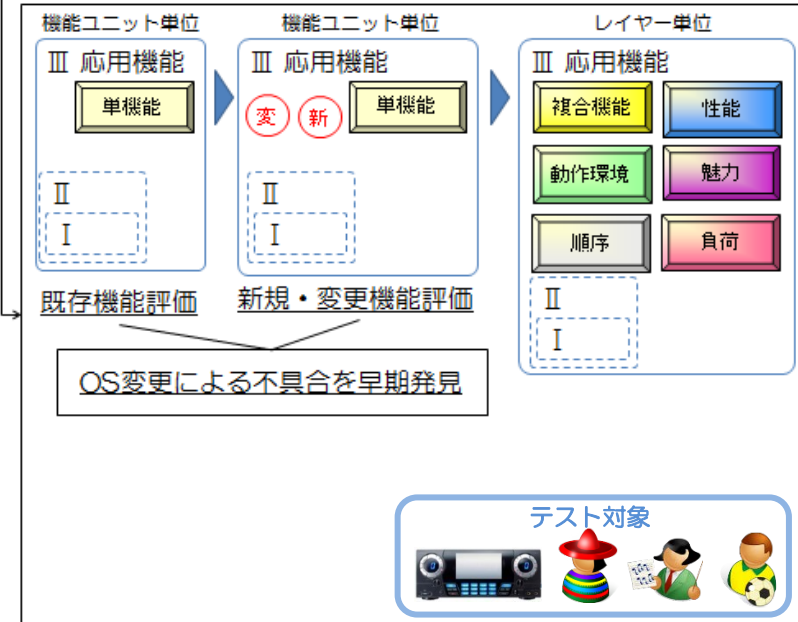
テストサイクル1



テストサイクル2



テストサイクル3



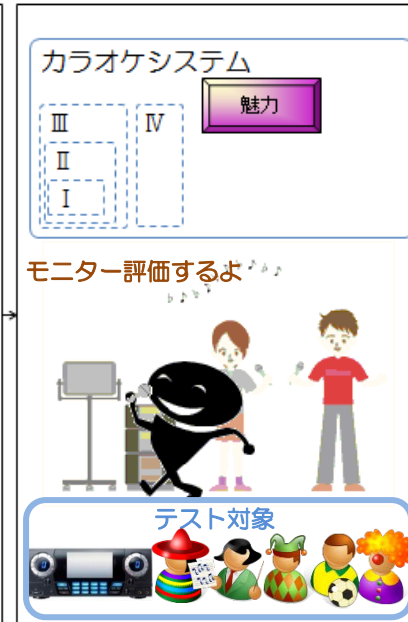
テストサイクル4



テストサイクル5



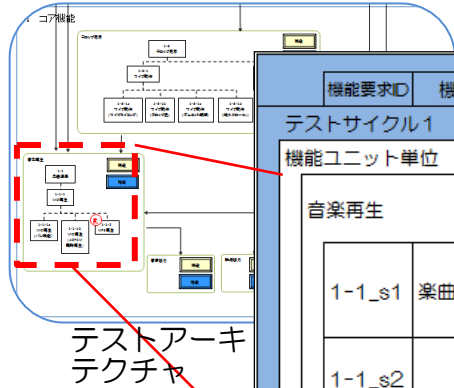
テストサイクル6



効率よく下から積み上げる切り口を時間軸でまとめてサイクルにした

テスト設計方針

機能と観点



機能要求ID	機能名	概要	因子	観点
テストサイクル1 対象メインレイヤー : I コア機能				
機能ユニット単位				
音楽再生				
1-1_s1	楽曲演奏	以下のフォーマットの音声データが再生できること。 ・MIDI ・MP3	音声データフォーマット	単機能 性能
1-1_s2	↑	↑	演奏マーク	単機能
1-1_s3	↑	↑	演奏可能楽曲時間	単機能

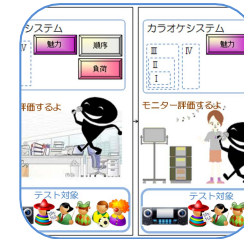
変換表を元に
方針決定するよ



観点方針変換表

上位観点	観点	例	方針	技法
単機能	設定	初期値	機能テスト	OK/NG判定
	固定値			
	上限下限			
複合機能 ※機能の切り分けにより単機能と同様になる場合もあり	状態	並行	機能テスト	OK/NG判定
	並行	排他		
	セキュリティ			
	感覚			
性能	処理速度	リソース	性能テスト	OK/NG判定
	リソース			
	機能			

テストサイクル



テスト対象と方針・技法

因子水準表

機能要求ID	機能名	概要	因子	テスト設計方針			
				方針1	技法1	方針2	技法2
テストサイクル1 対象メインレイヤー : I コア機能							
機能ユニット単位							
音楽再生							
1-1_s1	楽曲演奏	以下のフォーマットの音声データが再生できること。 ・MIDI ・MP3	音声データフォーマット	単機能テスト→正確性の確認	OK/NG判定	性能テスト→妥当性判断	OK/NG判定
1-1_s2	↑	↑	演奏マーク	単機能テスト→正確性の確認	OK/NG判定	性能テスト→表示タイミングの妥当性(感覚)判断	OK/NG判定
1-1_s3	↑	↑	演奏可能楽曲時間	単機能テスト→正確性の確認	OK/NG判定		

ユーザー向け機能	因子	水準			
カラオケ演奏機能	機能要求ID	赤字表記負荷テスト時の固定する水準			
カラオケ演奏機能	1-1 s1	音声データフォーマット	MP3	MIDI	
	1-1 s2	演奏マーク	歌唱開始位置	歌唱終了位置	各コーラスごとのサビ開始位置 各コーラスごとのコーラスエンド
	1-1 s3	演奏可能楽曲時間	30分(MAX)	5秒(MIN)	

テストケース抽出



テスト観点を元にすべてのテスト対象のテスト設計方針を決定

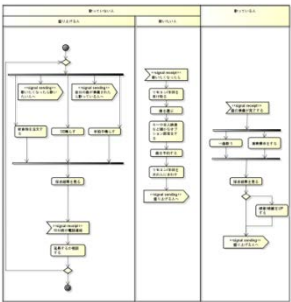
テスト詳細設計



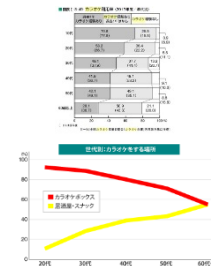
テスト詳細設計

■ テストケース シナリオテスト

ユーザー視点のアクティビティを作成



ボリュームゾーンの利用者のユーザー層のペルソナを作成する (プラグマティックペルソナフォーマット)



Aさん

年齢: 42歳
性別: 男性
職業: 会社員
年収: 750万円
住居: 賃貸 東京都中央区
家族構成: 妻 長男 長女
趣味: 音楽鑑賞
PCスキル: 中級

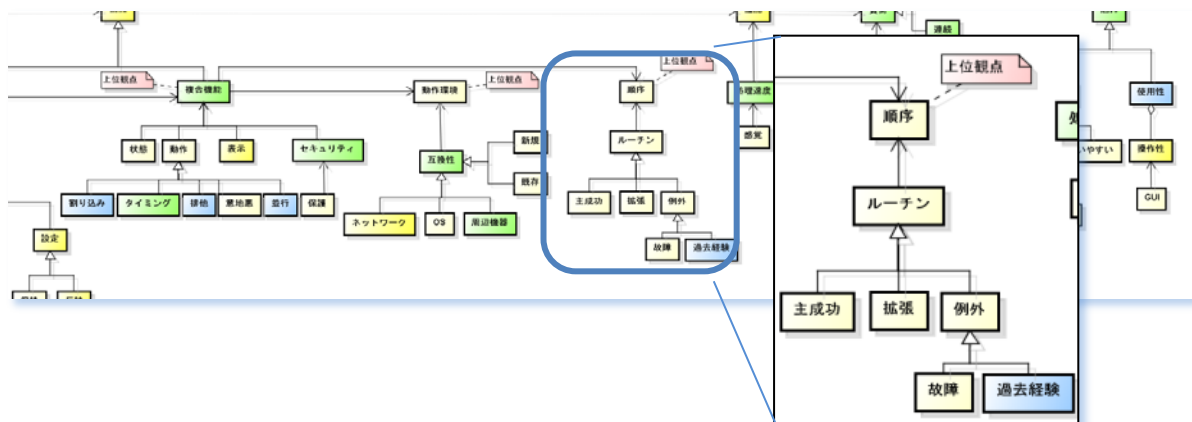
利用状況
今日は、8月のある連休日の深夜、いつものように帰省先である父の家で音楽鑑賞をする。父の部屋には、3000ccのオーディオシステムが設置されている。父は、音楽鑑賞が好きで、いつも、何でもいい曲を聴かせる。父の部屋には、父の趣味の音楽が溢れている。父の部屋には、父の趣味の音楽が溢れている。父の部屋には、父の趣味の音楽が溢れている。

3-1 (可憐な)
演奏の良し、種類が豊富に揃っている。その中でも、父の趣味の音楽が溢れている。父の部屋には、父の趣味の音楽が溢れている。父の部屋には、父の趣味の音楽が溢れている。父の部屋には、父の趣味の音楽が溢れている。

Better-Worse表や抱えている問題点の高いものを優先にテストケースを作成する

ID	シナリオ種類	ユーザー	内容	問題番号
Scene1	主成功+拡張シナリオ	ペルソナケース1	・楽曲となる 先鋒さんの選別会、その二次会としてカラオケBOXへ行ってきた。	
			・途中、後輩の心算が、口火を切って場を盛り上げる	
			・一曲を演奏、歌い始めた。	
			・通常の楽曲 (MIDI再生) で、音がいい曲の演奏 (一般演奏再生) も特に聞かない。フロッピーディスクの音も聞かない。	
			・カラオケボックスに詳しい同僚のCさんばかりもコンをやって、手拍子などのBEを鳴らす。場を盛り上げていく。	
			・同僚のDさんは気が付くタイプ。皆さんの歌う曲を聴きつつ飲み物注文。	
			・歌が終わるとメンバーの誰かがゲームで勝負！歌中に行くという事で、新録ゲームで勝負。	1-1-1
			・1曲終わったところで場は大盛り上がり。みんな酒に酔い酔いの音を聴く。	1-2-1
			・付属の無料LANリモコン2台に加え、本体操作も念のため3台の予約が完了 (私Aは糖音持ち。)、予約確認で3曲の演奏をOSD表示で確認の後、自分の曲を演奏。	1-3
			・2曲目は主役の好きな、大ファン女性歌手の曲を本人録音再生で選択。ただ、キーが高めな為、キーコントロールを調整して聴く。	1-4-1
			・3曲目はDさんだったが、携帯電話の鳴り響きも聞いた。後継としてDさんの曲へ。	1-4-2
			・Dさんは、アコースティック、演奏もスペシャルな1曲で、録音再生で再生 (MP3再生)、世界に入らぬで録音、でもアコースティックの、.. . 音を聴いてCFOを使って2曲までで終了。	1-1-2
・ここで、後継になったD子さんの曲、十八番の曲を演奏。新録ゲームで、これまででDさんの最高音。みんな酔い酔いとD子さんの音を聴く。	1-1-2			
・方々にこれだけ聴いて、最高の聴き手もおり、近頃はずいぶんここで大興奮。最後にみんなで歌える一曲を思ったが、多岐手キーは高解の2000、.. . 予約解除をと言えだが、録音も無し。.. . まで、録音に詳しいさんが割り込み予約で曲を入れる。それでも、時間は聴き手も、リベハンション機能で一気にかん、みんなで聴きつつ楽しい時を過ごすことが出来たのでした。	1-4-5 1-4-6 1-4-7 1-4-8 1-4-9 1-4-10			

シナリオの観点に主成功、拡張、例外を入れる



効率よくシナリオテストを作成した

まとめ

■ テストの網羅性と効率性の最大化

- 開発への思いやり
 - 不具合調査対象の最小化
 - 魅力的機能からのテスト
- 官能評価への挑戦
 - 製品の魅力の俯瞰
 - 人の感性の数値化
- ひらめきの連発
 - 観点モデリング
 - べんべん法



ワクワクする商品開発を実現するテスト設計ができた