

■ チーム紹介

- F A関係の仕事をしている有志メンバーです
- FactorySoftTestWorkingGroupの略です

■ コンセプト

- テストの網羅性と効率性(テストと開発)の最大化
- 開発への思いやり

■ 開発への思いやり

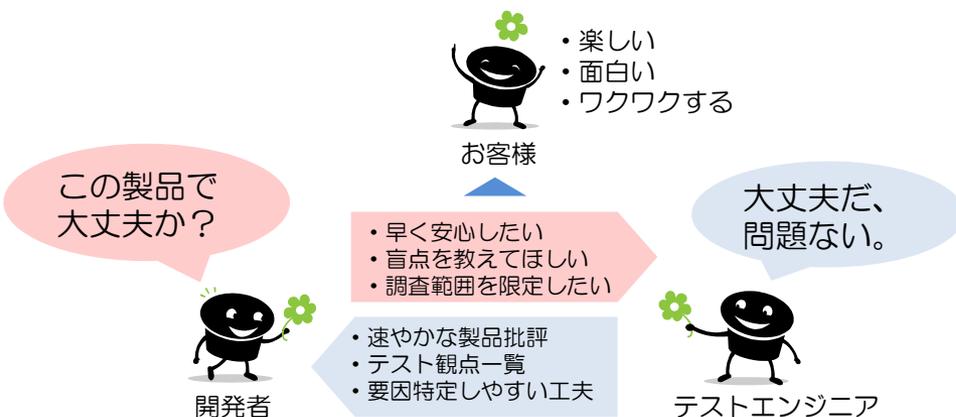
- モデリング手法を用いて各機能の分解&仕様整理を実施し、仕様の欠点、要望を抽出しました
- 開発側の不具合調査の探索範囲を小さくするため、完成すべき順でこまめにテストするアーキテクチャを作りました
- 過去不具合から発見の難しいOSの互換性を重視してアーキテクチャ設計に盛り込みました。
- 機能テストの優先順位を魅力度の視点から行い、納得性の高い手法を生み出しました

■ 官能評価への挑戦

- 製品の魅力を顧客アンケートから作成し、平面図で俯瞰できるようにしました
- 人の感性であるワクワク感などの判定難しいテスト手段を数値化できるよう画像認識や心拍数測定など従来のアンケート手法にとられない発想で広げました
- 採点機能には人の歌声にこだわらず、ボーカロイドにて正確に自動化できるよう効率化を図りました

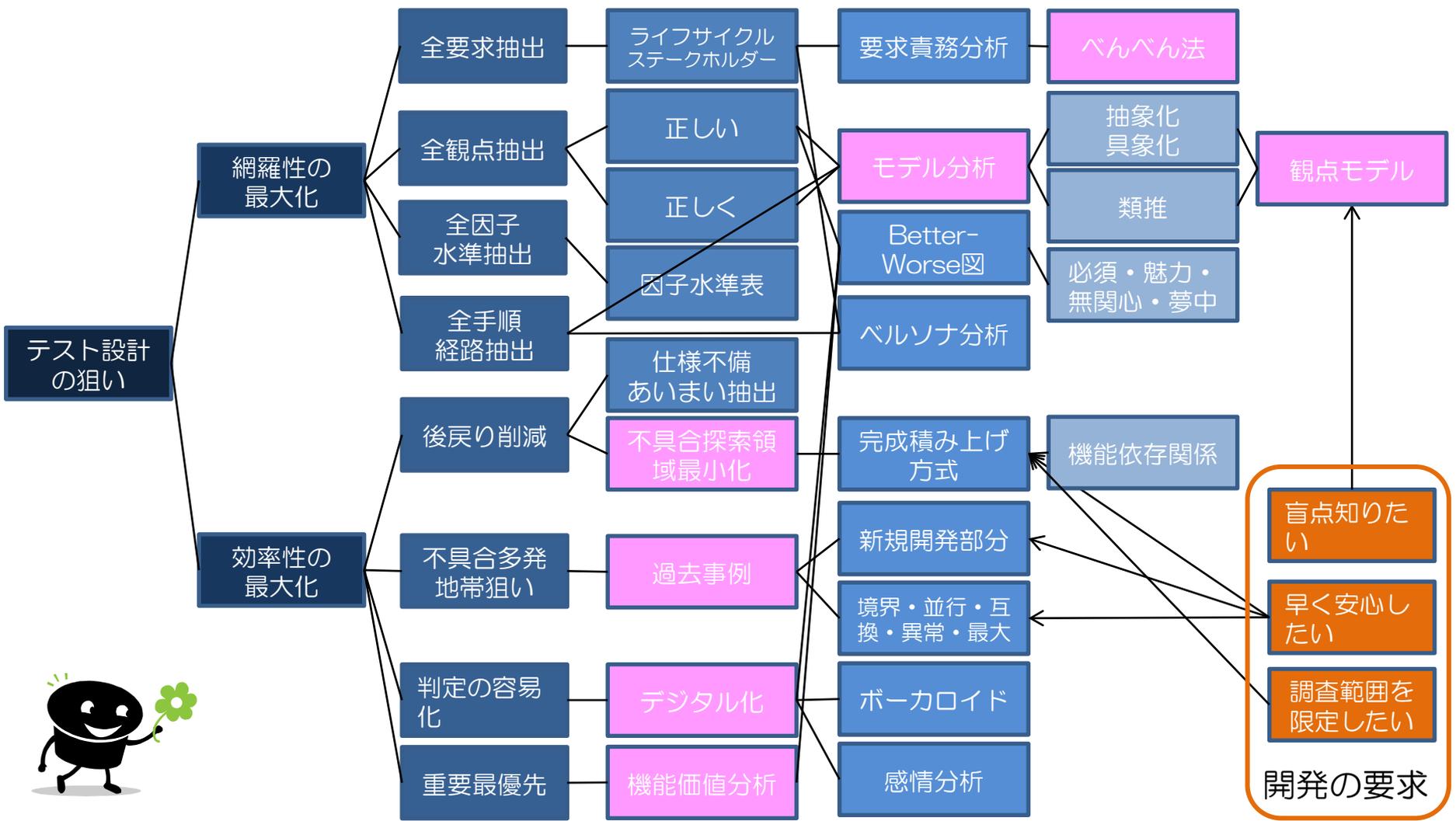
■ ひらめきの連発

- テスト対象を複数の視点で捉える「べんべん法」を発明し、求められている品質を出しました
- 観点モデリングにて抽象化、階層化することで見えなかった観点を出して網羅性を高めました



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

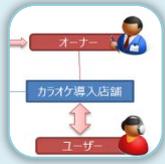
正しいものを作ってる?



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

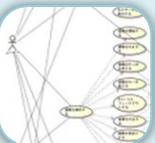


テストの切り口を見つけるよ

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



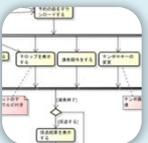
利用手順



状態

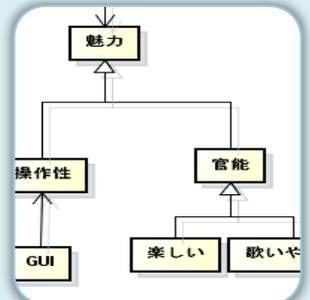


データ構造



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



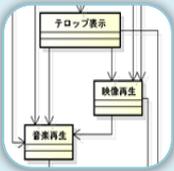
過去に学ぶよ

アーキテクチャ設計

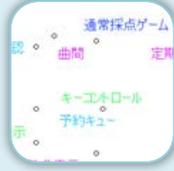
順番



優先順位を決めるよ



機能依存

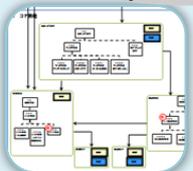


商品的魅力

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

テスト観点

全ての観点を洗い出すよ



正しいものを作っているか

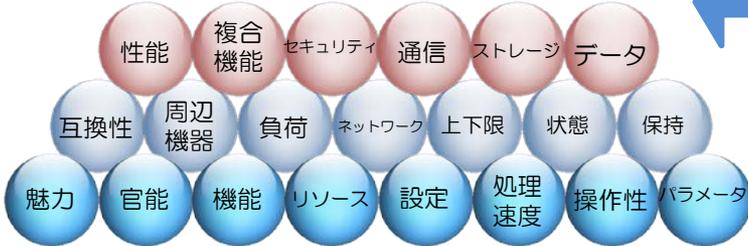
ステークホルダー	責務	要求大項目	要求(ステークホルダーがカラオケシステムに要求すること) 1. 機能から 2. 市場要望から 3. ビジネス視点から 4. 市道不具合から	あなたにとってカラオケシステム(機械)とは	テスト対象?	テスト観点		
ユーザー	カラオケを楽しむ	気持ちよく歌いたい	音質が良い方がいい	ストレス発散のための機械 楽しむための機械	する	性能	パラメータ	魅力
					する	性能	互換性	周辺機器
			歌いやすい		しない			
					する	機能	官能	処理
							管理	負荷
							監視	タイミ
							設定	設定

①責務を考える

③別名を考える

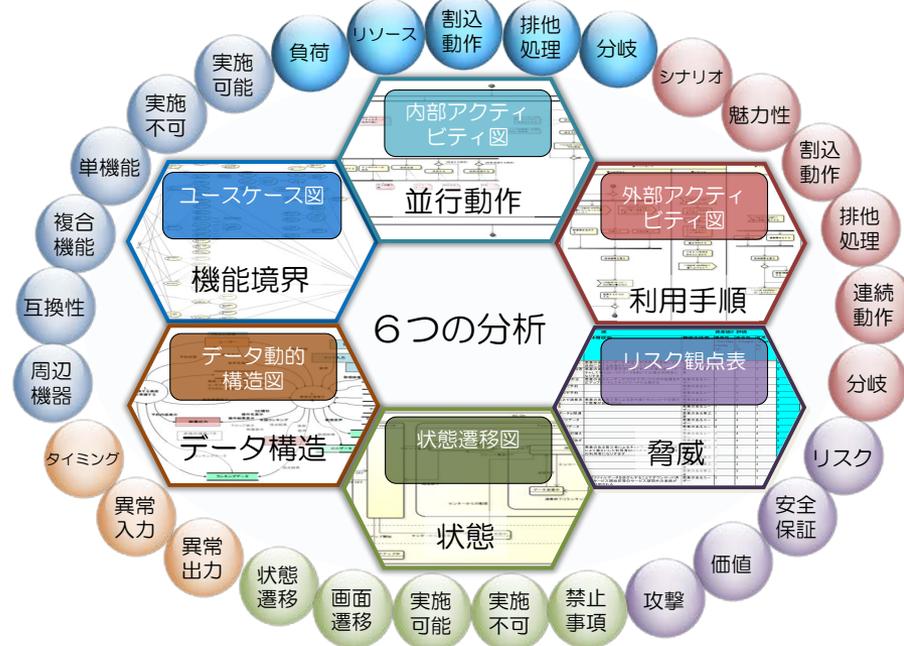
②要求を考える

④観点をみつける



ステークホルダーの要求ごとにテスト観点を抽出した

正しく作っているか



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

機能の魅力関係

優先順位を決めるよ



■ 手順

要件を元にアンケートを年代、性別、
責務別に取り

		ペルソナ	BOX 20代女性		
機能名	属性	有ればどうか	無いとどうか	判定	
1-1	楽曲編集	当然である	困る	必須	karaoke_miryoku\$Better
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	
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

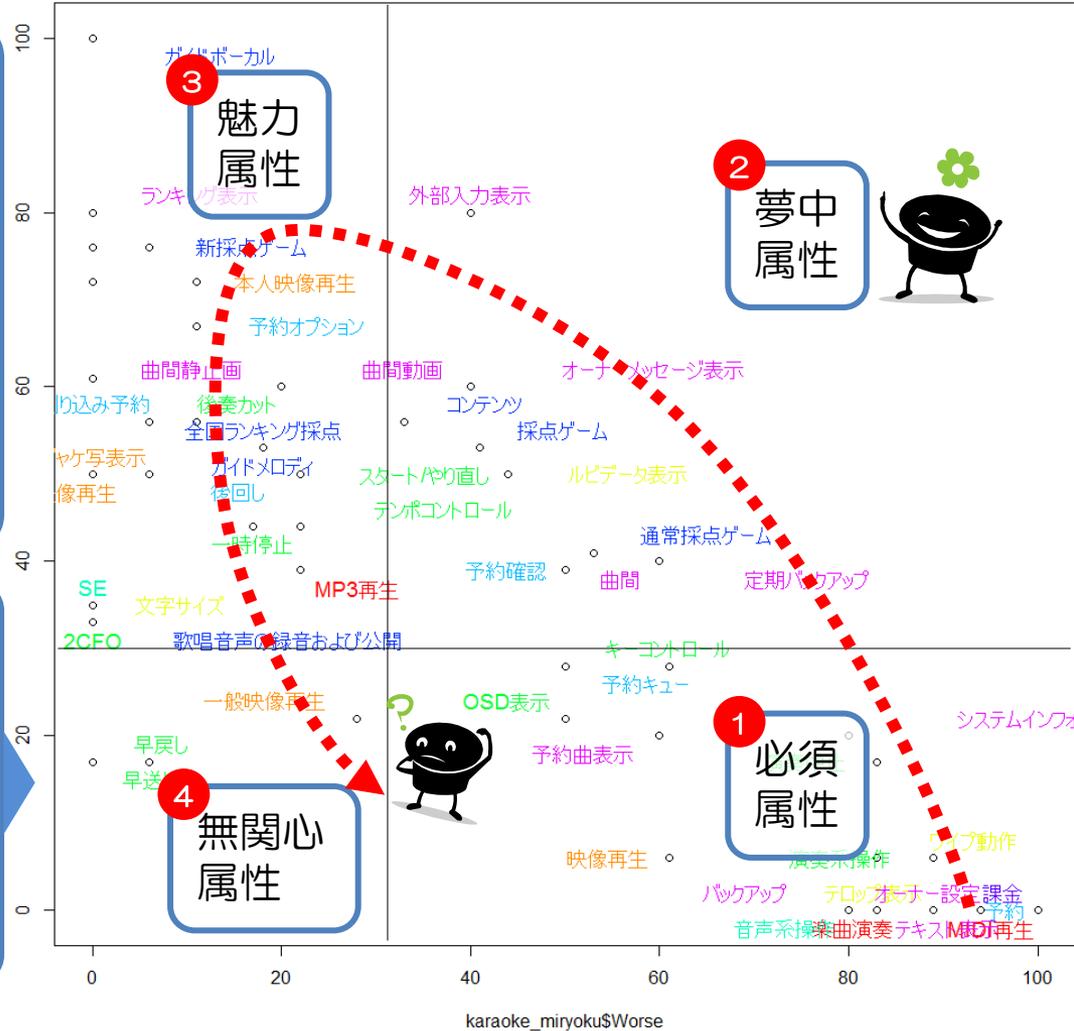
	O	A	M	I	R	S
属性1	x	x	x	x	x	x
属性2	x	x	x	x	x	x
...						

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

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

魅力的属性 (A, O) / 一元的属性 (M, I) / 当たり前属性 (M, I)

出所：『属性をより定量的に特徴づける方法』(Walden, 1933)



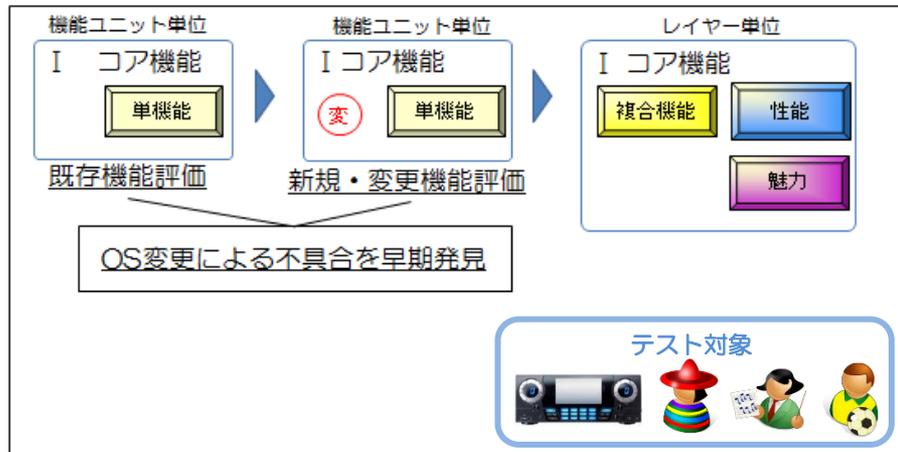
求められる機能の魅力度の必須→夢中→魅力→無関心で優先順位を決定

テストアーキテクチャ

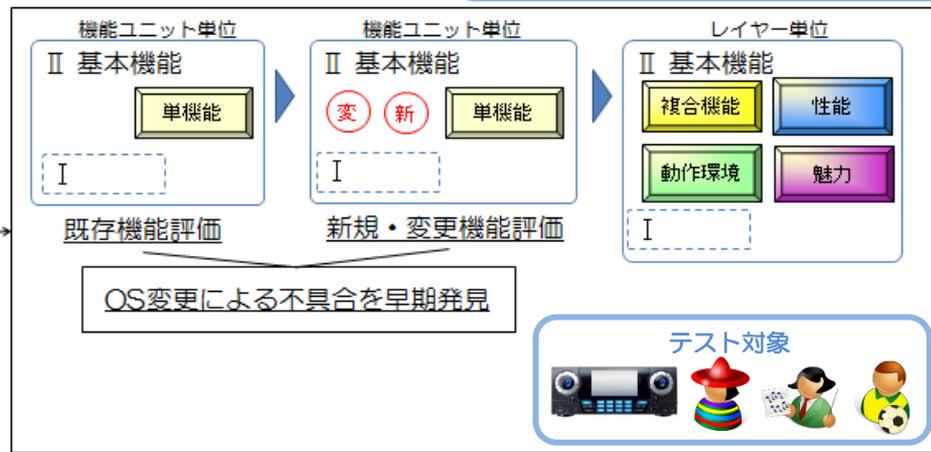
テスト対象アイコン説明



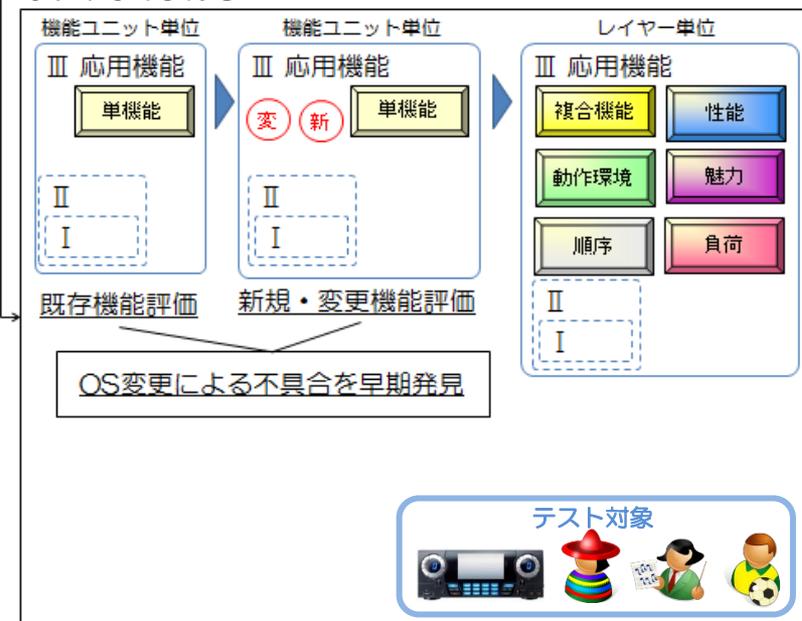
テストサイクル1



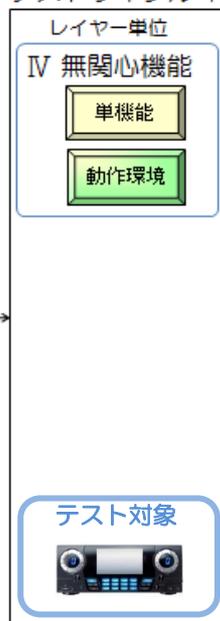
テストサイクル2



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