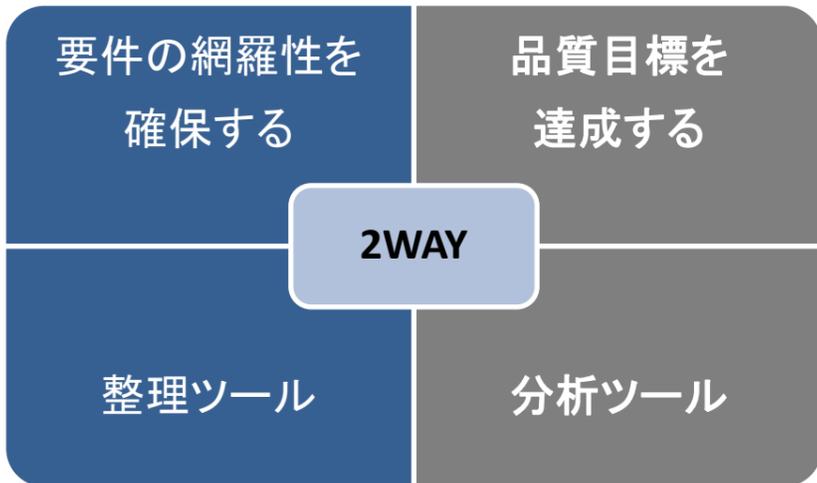


DMM (機能分析表) の活用法
 ~さまざまな関連性の整理と品質を追求する要求分析~
 DMMとはDiamond Mandala Matrix の略称です。



国際規格ISO25010 の品質特性を枠組みに利用して機能要求の漏れを防ぎたい

<1> ISO1-1 完全性 ISO1-2 正確性 ISO1 機能適合性	<2> ISO2-1 時間効率性 ISO2-2 資源効率性 ISO2-3 収容効率性	<3> ISO3-1 共存性 ISO3-2 相互運用性 ISO3 互換性
<8> ISO8-1 適合性 ISO8 移換性	<0> ISO1 機能適合性 ISO8 移換性 ISO7 保守性	<4> ISO4-1 適切適応性 ISO4-2 簡潔性 ISO4-3 運用性 ISO4-4 ユーザーエラー防止性 ISO4-5 UIの快便性
<7> ISO7-1 モジュール性 ISO7-2 再利用性 ISO7-3 解析性 ISO7-4 実装性 ISO7-5 テスト容易性	<6> ISO6-1 機能保持性 ISO6-2 整合性 ISO6-3 西証防止性 ISO6-4 セキュリティ ISO6-5 責任追跡性	<5> ISO5-1 成熟性 ISO5-2 可用性 ISO5-3 障害特性性 ISO5-4 回復性

品質全体が俯瞰できる。

マインドマップを整理したい

機能要求を網羅し、全体を俯瞰したい

テストケースの抜け漏れを予防したい

テストケースから効率的にバグ抽出したい

テストスコープの関連性を見たい

ユーザー向け	オーナー向け	サブライカー向け	拡張性	移換性	性能目標	新たな様式システムを構築する	高度な映像データに対応する	生産データに対応する
ハードウェアインターフェース	ソフトウェアインターフェース	通信インターフェース	機能要求 (要件定義書)	要件定義書	要件定義書	ハードウェアを構築する	ソフトウェアの機能を構築する	暗号化を行う
システム構成	構成外構成	画面構成	テストスコープ	要件定義書	要件定義書	ハードウェアを構築する	ソフトウェアの機能を構築する	暗号化を行う

機能要求の関連性を見たい

1-0 音声再生	1-1 音声再生	1-2 音声再生	1-3 音声再生	1-4 音声再生	1-5 音声再生	1-6 音声再生	1-7 音声再生	1-8 音声再生	1-9 音声再生	1-10 音声再生	1-11 音声再生	1-12 音声再生	1-13 音声再生	1-14 音声再生	1-15 音声再生
1-16 音声再生	1-17 音声再生	1-18 音声再生	1-19 音声再生	1-20 音声再生	1-21 音声再生	1-22 音声再生	1-23 音声再生	1-24 音声再生	1-25 音声再生	1-26 音声再生	1-27 音声再生	1-28 音声再生	1-29 音声再生	1-30 音声再生	1-31 音声再生

親階

ISO1 機能適合性	ISO2 性能効率性	ISO3 互換性	ISO4 運用性	ISO5 可用性	ISO6 責任追跡性	ISO7 保守性	ISO8 移換性
ISO1 機能適合性	ISO2 性能効率性	ISO3 互換性	ISO4 運用性	ISO5 可用性	ISO6 責任追跡性	ISO7 保守性	ISO8 移換性
ISO1 機能適合性	ISO2 性能効率性	ISO3 互換性	ISO4 運用性	ISO5 可用性	ISO6 責任追跡性	ISO7 保守性	ISO8 移換性

DMM間でトレーサビリティを確保できる。

①: プロジェクトのテスト目的に合った具体的な要求(品質)を定義する。

②: ①で作成したDMMの品質特性毎にテストタイプやテスト技法を割り当てる。

テスト要求モデルとして委託元に提供し、テストの保証範囲や動作を保証する機能の概要の位置付けで確認してもらえばボリューム感が伝わる。

システム構成図など既存の図解資料があれば外部(データの発生源/行き先)との入出力を整理でき、テスト設計の精度が向上できる。

なぜか見逃してしまったり、バグやインパクトの高い機能についてさらに深い分析を行い、テストケースを増やしたい。

アイデアを絞り出す。

DFDを作図すればプロセス(機能)間の関連性を整理できる。

DMMを設計図としてさらにDFDを作成できる。

アイディアを絞り出す

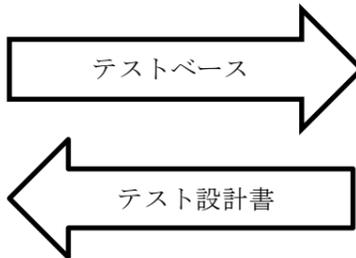
ISO1 機能適合性	ISO2 性能効率性	ISO3 互換性	ISO4 運用性	ISO5 可用性	ISO6 責任追跡性	ISO7 保守性	ISO8 移換性
ISO1 機能適合性	ISO2 性能効率性	ISO3 互換性	ISO4 運用性	ISO5 可用性	ISO6 責任追跡性	ISO7 保守性	ISO8 移換性
ISO1 機能適合性	ISO2 性能効率性	ISO3 互換性	ISO4 運用性	ISO5 可用性	ISO6 責任追跡性	ISO7 保守性	ISO8 移換性

テスト設計プロセスの全体図

～低コストで高品質かつ短納期を実現する～
 テストを実行するテスター(自分)がイイと思えるものを目指して

テストの十分性を確認したい。

委託元
社内の開発部門



自部門で実施可能なテスト内容の全容を知ってもらうことで、テストで保証できる範囲と動作を保証できる内容を明確に伝えたい。

テスト方針	テスト計画						
	テスト要求分析			テストアーキテクチャ設計		テスト詳細設計	
	①	②	③	④	⑤	⑥	⑦
プロセスの見える化	●	●	●	●			
分析に無駄がない		●	●	●			
要件とテストケースがマッチしている					●	●	●

テスターのテスト技術も確認したい。

DMMを整理ツールとして使う。

テスト要求分析

DMM(機能分析表)
 ・テストスコープのDMM
 ・データ辞書(テストスコープ用)
 テストコンテナ
 ・テスト観点のベース(枠組み)

資料
 ・要件定義書
 ・補足資料
 ・用語集
 ・QA管理表

①テストスコープを確認する

DMM(機能分析表)
 ・機能要求のDMM

②DFDを作図する

1 > DFD_テスト条件を抽出済み
 ・事前条件、事後条件
 ・テスト観点
 ・データ辞書(機能要求用)

③テストフレームを作成する
 ※テスト条件の検討

<前工程にフィードバックする>
 DFD(データフローダイアグラム)
 ・プロセス(機能)間の関連性を整理済み
 ・外部(データの発生源/行き先)を洗い出し済み
 ・コンテキストダイアグラム

DMMは工夫次第でいろいろ使える。

完成した機能要求のDMMをテスト設計書へ添付する

2 > DMM_テストレベルを抽出済み
 ・単体テスト
 ・外部結合テスト、内部結合テスト
 ・システムテスト

テストコンテナ
 ・テスト条件作成済み

テストコンテナ
 ・テストフレーム作成済み

④テスト条件を作成する

テストアーキテクチャ設計

⑤テストコンテナを整理する

テストコンテナ
 ・整理済み

⑥テストデザインパターンを検討する

テスト詳細設計

テストコンテナ
 ・コンテナ分割済み

⑦テストケースを作成する

3 > DFD_要素を抽出済み(原因・結果・中間)
 ・原因結果グラフ
 ・デシジョンテーブル
 DMMを分析ツールとして使う。

テスト詳細設計

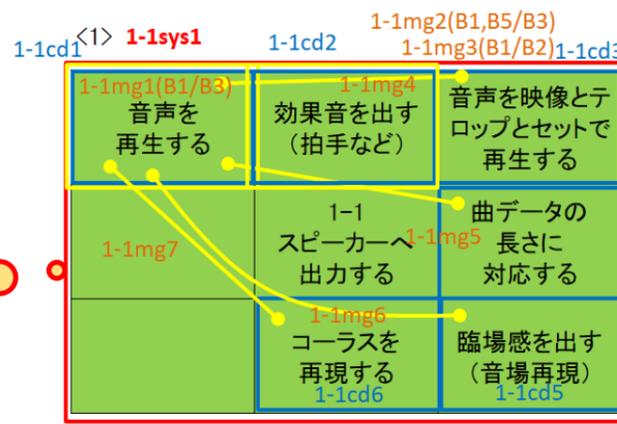
テスト実装作業へ

第1~2階層

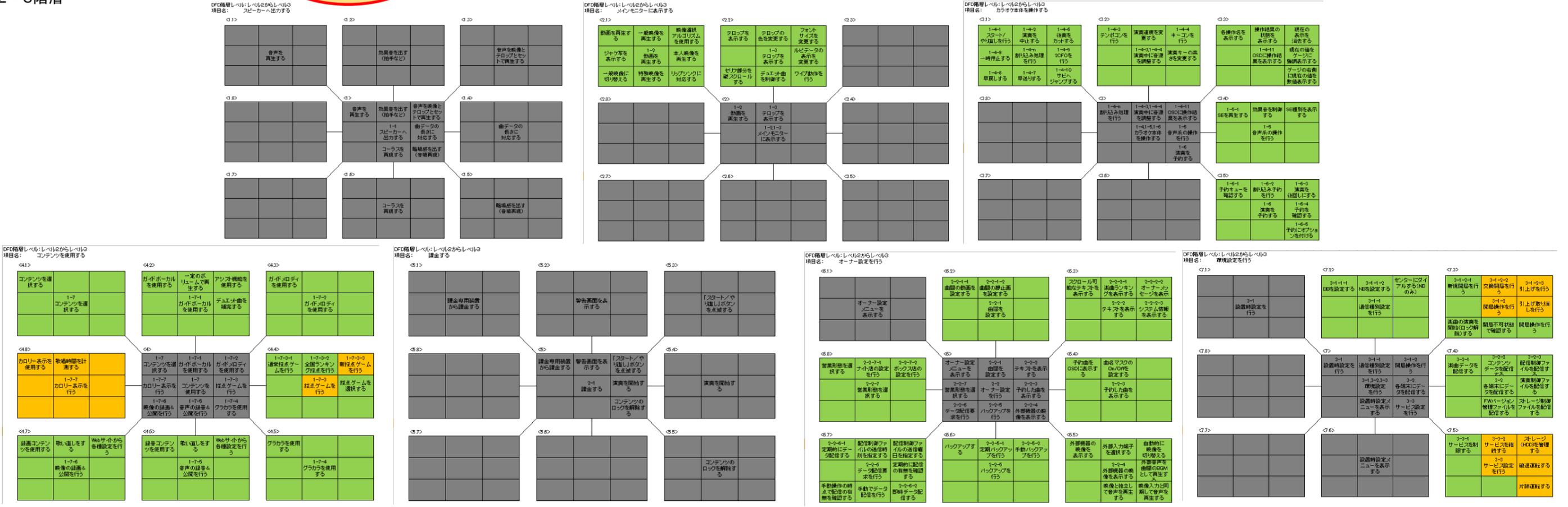
機能要求のDMM

~機能要求のすべてを俯瞰する~
要求漏れ0件を達成するため

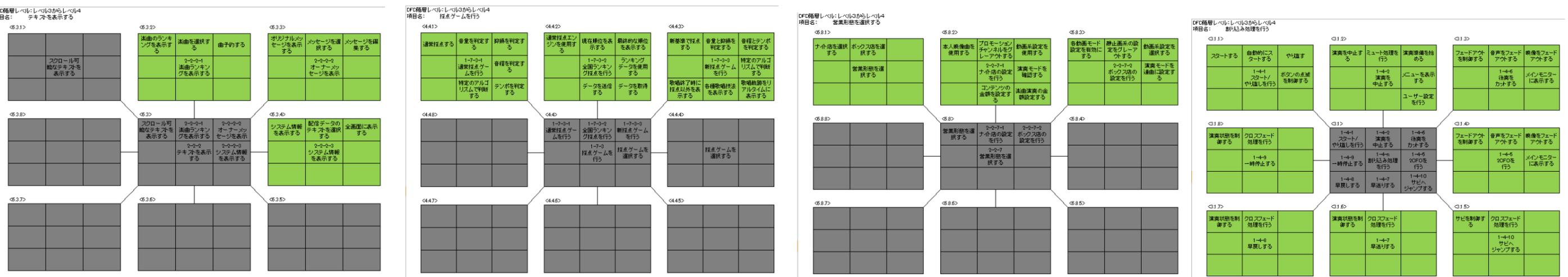
単体テストは青色の枠で囲む。
外部結合は黄色の枠で囲む。
内部結合は黄色の野線で結ぶ。
システムテストは赤色の枠で囲む。



第2~3階層



第3~4階層



DFD(データフローダイアグラム)は品質のかなめ
 ~新たなテストケースを生み出すモデリングを~
 リバースする目的

DFDはまずトップダウン式にプロセス(機能)間の関連性の整理を行い、テスト観点として必要な外部(データの発生源/行き先)を洗い出す。

外部(データの発生源/行き先)との入出力を整理する目的でDFDの作成に使用する。

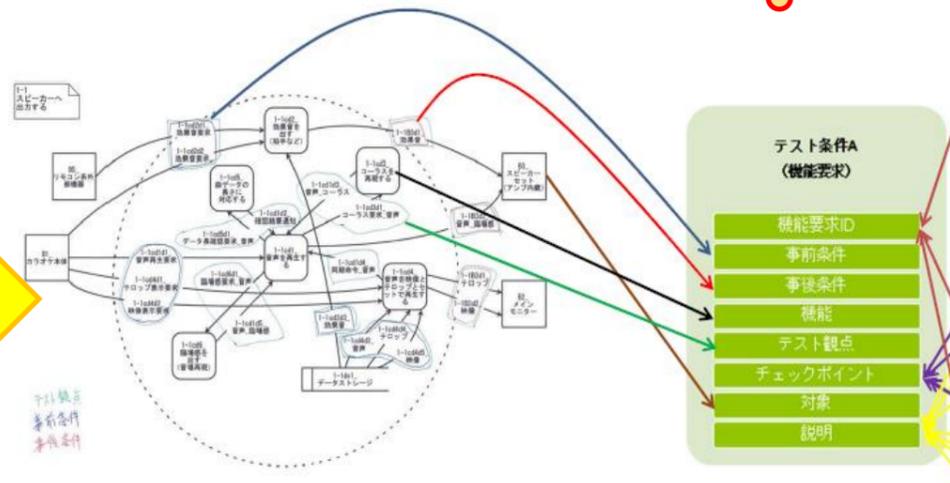
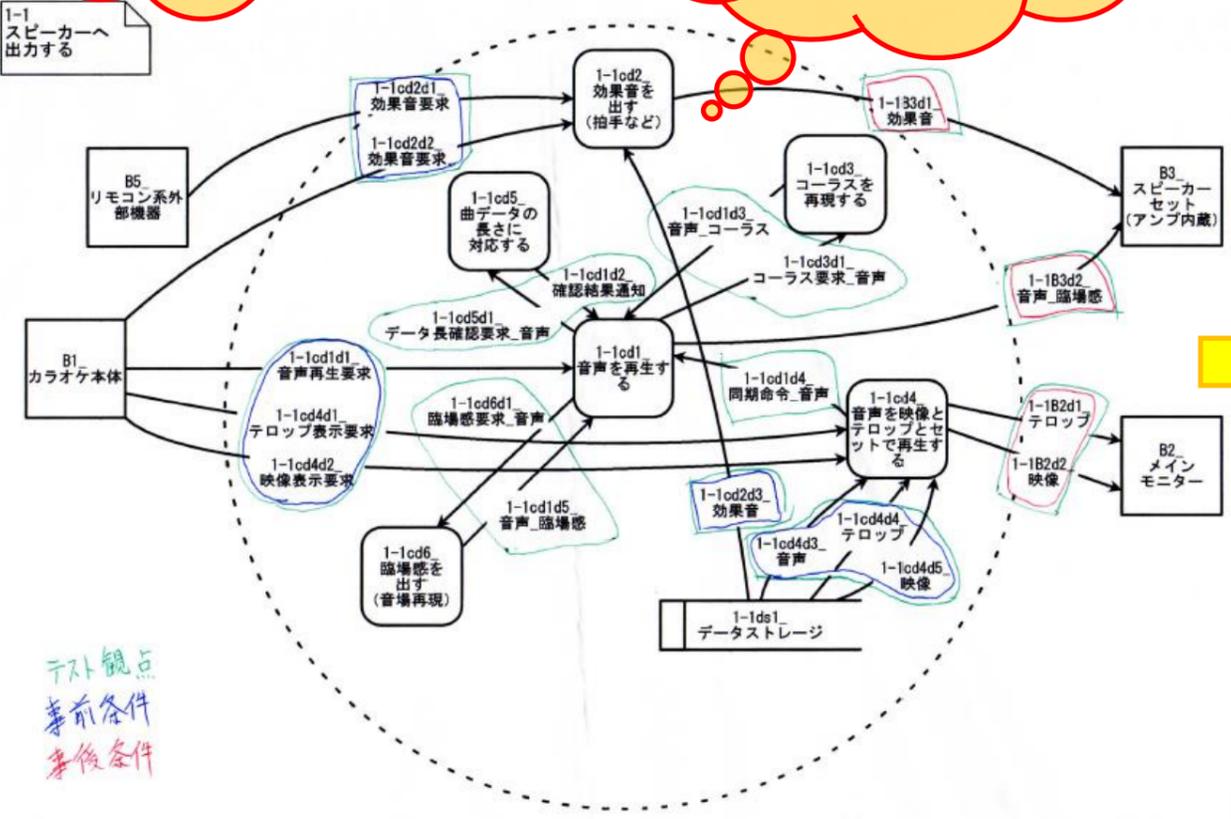
外部(データの発生源/行き先)

システム構成図より抽出したテスト観点(親)		機能要求の分析により抽出したテスト観点(子)		
No	センター側	No	分類1	分類2
S1	認証サーバ	S1-1	Webブラウザ	
S2	コンテンツサーバ			
S3	配信サーバ			
S4	端末管理サーバ			
S5	専用VPNボックス			
S6	アナログモデム			
BB設定店舗側			分類1	分類2
BB1	専用VPNボックス			
BB2	無線LANアクセスポイント			
NB設定店舗側			分類1	分類2
NB1	アナログモデム			
NB4	ビルコインボックス	NB4-1	コインボックス	
BB&NB共通機器			分類1	分類2
B1	通信カラオケ本体(本機)	B1-1	DSP	
B2	メインモニター			
B3	スピーカーセット(アンプ内蔵)			
B4	マイク(複数あり)	B4-1	マイク1	B4-2
B5	リモコン系外部機器(複数機種あり)			

データの流れに抜けを漏れを出さないために
 コンテキストダイアグラムを書こう!

関連性を整理したら次は最下層からボトムアップ式に外部(データの発生源/行き先)との入出力を整理する。

作成したDFDから事前条件、事後条件、プロセス(機能)、テスト観点、対象となる外部(データの発生源/行き先)を抽出し、テストベース(機能要求)と合わせてテストフレームを作成する。



テスト条件A (機能要求)

機能要求ID	1-1
事前条件	
事後条件	
機能	
テスト観点	
チェックポイント	
対象	
説明	

カラオケ業務機能

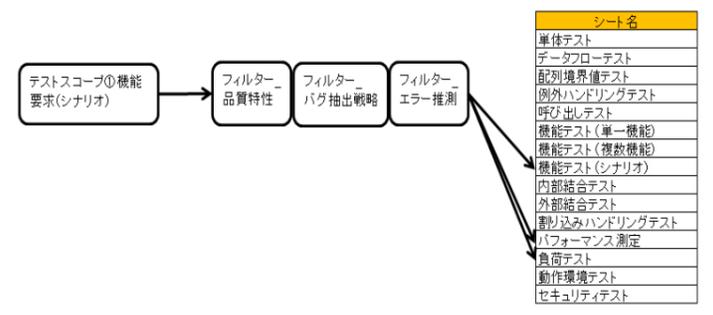
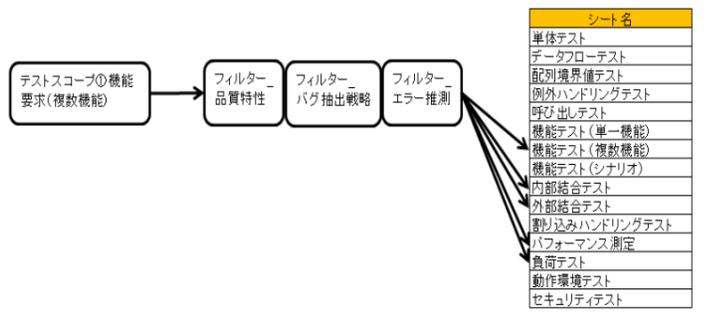
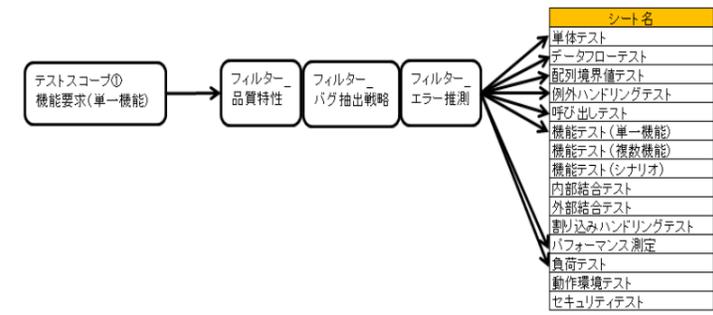
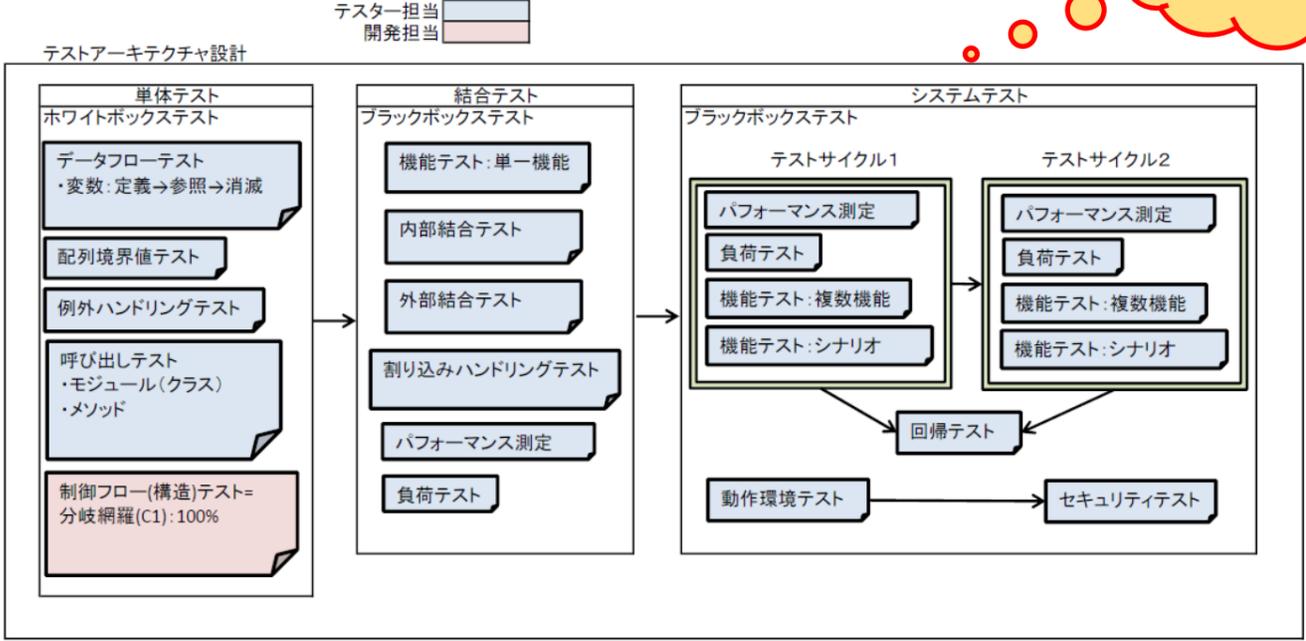
機能要求ID	機能名	概要
1-1	音楽再生	以下のフォーマットの音声データが再生できること。 ・MIDI ・MP3
1-1-1	MIDI再生	また再フォーマットとも楽曲データにはガイドメロディ、採点用リフ、アレンジなど演奏マークを含んでいる。 演奏マークは以下の種類がある。 ・各コーラスのコースエディット ・各コーラスのサブ開始位置 ・歌唱開始位置 ・歌唱終了位置
1-1-2	MP3再生	再生時により各種設定で指定された映像データ、テロップデータとセットで再生される。 これをカラオケ機能の基本機能とする。
1-1-3	MP3再生	楽曲再生可能なMP3ファイルの再生は再生される楽曲ファイル(最小10分)の再生時間に対応する必要があり、配信データの状況の条件は可変である(例: 最長30分)の再生が可能である。
1-1-4	MP3再生	楽曲データはサブライナーから提供されるため、その扱いは問わないものとする。
1-1-5	MP3再生	再生可能なMP3ファイルの再生は再生される楽曲ファイル(最小10分)の再生時間に対応する必要があり、配信データの状況の条件は可変である(例: 最長30分)の再生が可能である。
1-1-6	MP3再生	楽曲の音声データのフォーマットとして「標準音」と呼ばれる。
1-1-7	MP3再生	楽曲の音声データのフォーマットとして「高音質音」と呼ばれる。
1-1-8	MP3再生	楽曲の音声データのフォーマットとして「低音質音」と呼ばれる。

テスト観点
 事前条件
 事後条件

テストアーキテクチャ設計 ～テストプロセス全体をデザインする～ 理想的な職場環境をイメージする

テストプロセス全体をモデリングする。

テストコンテナAの関連図



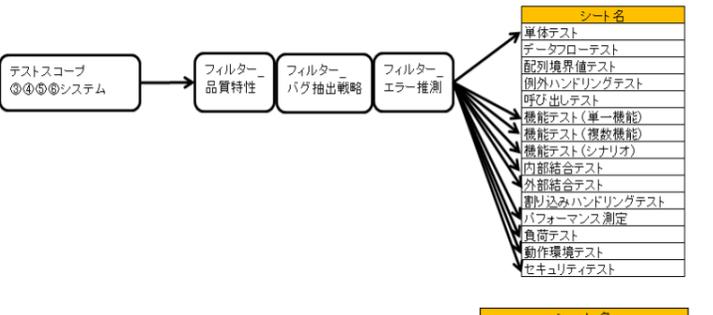
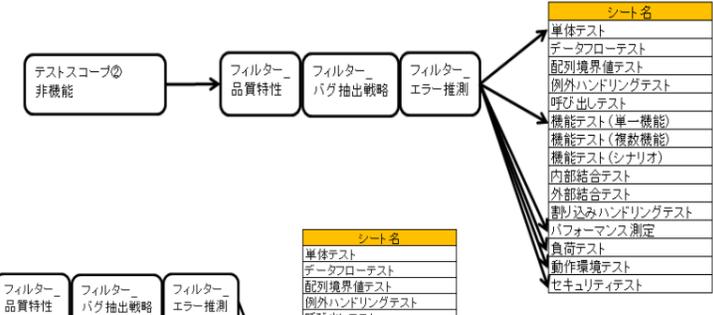
テストコンテナAと重複する項目についてどうするかを明確にする。重要なテスト観点はリズムもテストに必要です。歯抜けにするとテスターが迷います。。

テストコンテナBの関連図

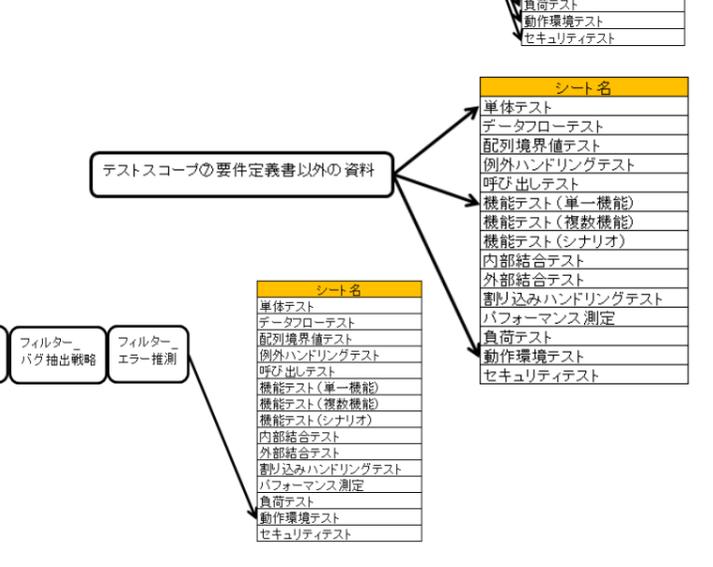
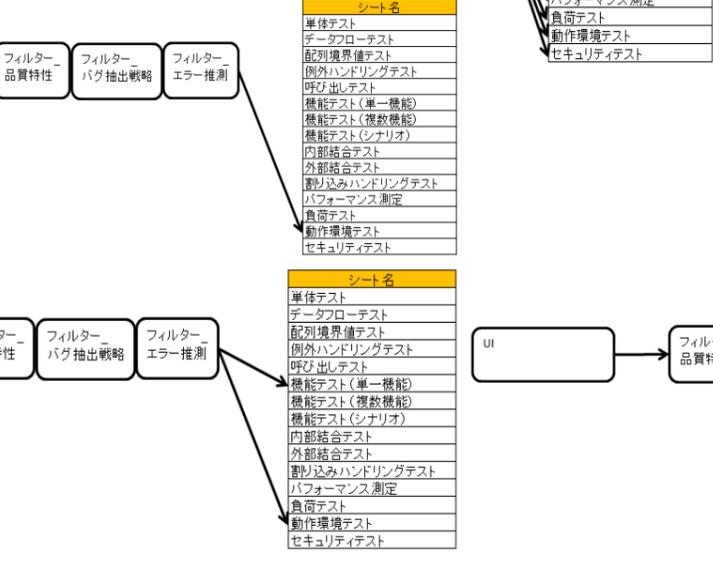
テストコンテナとテスト詳細設計の関連図

コンテナタイプ	シート名	備考
テストコンテナA	テストスコープ①機能要求(単一機能)	テスト条件A
	テストスコープ①機能要求(複数機能)	テスト条件A
	テストスコープ①機能要求(シナリオ)	テスト条件A
テストコンテナB	テストスコープ②非機能	テスト条件B
	テストスコープ③,④,⑤,⑥システム	テスト条件B
	テストスコープ⑦要件定義書以外の資料	
	テストスコープ⑧インターフェース	テスト条件B
	ふるまい	テスト条件B
	UI	テスト条件B
	外部(データの発生源・行き先)	
テストコンテナF	フィルター 品質特性	
	フィルター バグ抽出戦略	
	フィルター エラー推測	

テスト詳細設計タイプ	シート名	備考
テスト詳細設計 開発用	単体テスト	テスト構造のみ
テスト詳細設計 テスター用	データフローテスト	テスト構造のみ
	配列境界値テスト	テスト構造のみ
	例外ハンドリングテスト	テスト構造のみ
	呼び出しテスト	テスト構造のみ
	機能テスト(単一機能)	
	機能テスト(複数機能)	
	機能テスト(シナリオ)	
	内部結合テスト	
	外部結合テスト	
	割り込みハンドリングテスト	
	パフォーマンス測定	
	負荷テスト	
動作環境テスト		
セキュリティテスト		



テストコンテナ内を同じようなテスト観点の組み合わせで整理し、実施するテストプロセス全体図を参照してテスト詳細設計を目的別に分類する。



3つのフィルターでユーザーに安心と安全を提供する
 ~本来のテスト目的とは~
 品質に抜け漏れなく、バグも抽出する

ISO25010の品質特性を抜け漏れ防止のため、参考に使用する。

ISO25010品質特性(副特性)	通信カラオケシステムにおける品質特性の定義	担当者	実施可否	備考
ISO1 機能適合性	ISO1 機能適合性を確認			
ISO1-1 完全性	ISO1-1-1 要件を満たしている ISO1-1-2 要求を満たしている	テスター	可	結合テスト システムテスト
ISO1-2 正確性	ISO1-2 入出力が正確である	テスター	可	
ISO1-3 適切性	ISO1-3 目的に合っている	テスター	可	
ISO2 性能効率性	ISO2 性能効率性を確認			
ISO2-1 時間効率性	ISO2-1 パフォーマンスを測定する	テスター	可	結合テストから計測を開始する
ISO2-2 資源効率性	ISO2-2 連続で楽曲演奏を再生するなど、再利用時に問題がない	テスター	可	メモリーリーク等がないこと
ISO2-3 収容効率性	ISO2-3-1 大容量データ使用時に問題がない ISO2-3-2 音声映像とテロップとセットで再生時などの並行処理時に問題がない	テスター	可	通信の切断、音声のノイズ、映像の乱れ等がない
ISO3 互換性	ISO3 互換性を確認			
ISO3-1 共存性	ISO3-2-1 現行の外部機器に対応されている	テスター	可	必要性なし
ISO3-2 相互運用性	ISO3-2-2 過去の楽曲データに完全互換対応されている ISO3-2-3 従来機に搭載されていたコンテンツが搭載されている	テスター	可	
ISO4 使用性	ISO4 使用性を確認			
ISO4-1 適切認識性	ISO4-1 適切に機能の目的が伝わる	テスター	可	
ISO4-2 習得性	ISO4-2 表記が分かりやすい	テスター	可	
ISO4-3 運用性	ISO4-3 待機時間は妥当である	テスター	可	楽曲開始まで 次曲開始まで ログインされるま
ISO4-4 ユーザーエラー防止性	ISO4-4 2度押し対応するなどエラー防止処置は妥当である	テスター	可	
ISO4-5 UIの快美性	ISO4-5 画面に乱れがなくUIは快美である	テスター	可	
ISO4-6 利便性	ISO4-6 ボックス店とナイト店のターゲット層が使いやすい	テスター	可	高齢層層のため 文字の大きさや ハイライト等の必
ISO5 信頼性	ISO5 信頼性を確認			
ISO5-1 成熟性	ISO5-1 高負荷時も安定して稼働できる	テスター	可	テストサイクル
ISO5-2 可用性	ISO5-2 営業中は連続して稼働する	テスター	可	耐久テスト
ISO5-3 障害許容性	ISO5-3 イリーガル操作の対策は妥当か	テスター	可	
ISO5-4 回復性	ISO5-4 トラブル発生しても早期に復旧	テスター	可	
ISO6 セキュリティ	ISO6 セキュリティを確認			
ISO6-1 機密保持性	ISO6-1 データや通信は暗号化されるか	テスター	可	
ISO6-2 整合性	ISO6-2 暗号化後は正しく復号化できる	テスター	可	
ISO6-3 否認防止性	ISO6-3 カラオケの利用履歴を残している	テスター	否	要件に履歴に関する記述がない
ISO6-4 真正性	ISO6-4 外部から不正アクセスできない	テスター	可	
ISO6-5 責任追跡性	ISO6-5 登録されたユーザー情報に対する操作履歴は残している	テスター	否	要件に履歴に関する記述がない
ISO7 保守性	ISO7 保守性(モジュールに対して)を確認			
ISO7-1 モジュール性	ISO7-1 機能的なモジュール化を行い、相互依存性は最小化されている	開発	可	
ISO7-2 再利用性	ISO7-2 モジュールは他製品でも再利用可能で	開発	可	
ISO7-3 解析性	ISO7-3 ソースコードはデバックして解析できる	開発	可	
ISO7-4 変更性	ISO7-4 ソースコードは変更を考慮して整理している	開発	可	
ISO7-5 テスト容易性	ISO7-5 コードカバレッジ計測のためにテストを自動化している	開発	可	分岐網羅:C1
ISO8 移植性	ISO8 移植性(モジュールおよびシステム)を確認			
ISO8-1 適合性	ISO8-1-1 他の製品でもロジックの使用が可能である ISO8-1-2 オリジナルOSからLinuxOSへ移植後の適合度合いを確認する	開発 テスター	可	新採点システム カロリー表示機能
ISO8-2 設置性	ISO8-2-1 他の製品でもロジックは呼び出すだけで簡単に使用できる ISO8-2-2 システムを別のハードウェアへ交換可能である	開発 テスター	可	新採点システム カロリー表示機能
ISO8-3 置換性	ISO8-3-1 異なる言語にもロジックを置き換えやすい ISO8-3-2 Linux以外のOS環境にも置き換えられる	開発 テスター	可 否	新採点システム カロリー表示機能 具体的な要件の記述がないため確認しない

要件をテストケースと紐づける!

効率的にバグを取る技術!

バグを抽出するための戦略の参考とする。

バグ抽出戦略	備考
BUG1 ブラックボックステスト技法の活用	
BUG1-① 同値分割(時間帯)	
BUG1-② 同値分割(制限値)	
BUG1-③ 境界値分析(2値)	
BUG1-④ 境界値分析(3値)	
BUG2 性能テストを行う	
BUG2-① 同時実行可能な複数機能の並列処理	
BUG2-② 単一機能内で発生する複数の処理の並行処理	
BUG3 組み合わせテスト技法の活用	
BUG3-① 利用可能な機能の組み合わせテストを行う	
BUG3-② 利用可能な機器類の組み合わせテストを行う	
BUG3-③ デシジョンテーブルテストを行う	
BUG3-④ nワイステストを行う	
BUG4 シナリオテスト技法の活用	
BUG4-① ユースケーステストを行う(汎用的なもの)	
BUG4-② シナリオテストを行う(具体的なもの)	
BUG4-③ ユーザビリティテストを行う	
BUG5 エラー推測技法の活用	DFDから原因結果グラフを作成
BUG5-① デシジョンテーブルを作成する	
BUG5-② 原因結果グラフを作成する	
BUG6 セキュリティテストを行う	
BUG6-① 通信経路は侵入テストを行う	
BUG6-② 通信経路はポート開放状態を確認する	
BUG6-③ 暗号化される場合は暗号化の強度を確認する	
BUG6-④ 情報資産は操作した内容をログで確認する	
BUG6-⑤ 利用した履歴をログで確認する	
BUG7 保守性テストを行う	開発担当者が確認すること
BUG7-① 命令網羅(C0)の容易性を確認する	開発担当者が確認すること
BUG7-② 分岐網羅(C1)の容易性を確認する	開発担当者が確認すること
BUG7-③ 条件網羅(C2)の容易性を確認する	開発担当者が確認すること
BUG7-④ テスト自動化の容易性を確認する	
BUG8 移植性テストを行う	開発担当者が確認すること
BUG8-① 別のハードウェアへの移植のしやすさを確認	
BUG8-② 別のOS環境への移植のしやすさを確認	

無理やりでもバグを見つけるアイデアを絞り出そう!

より深いエラー推測を行うための参考値とする。

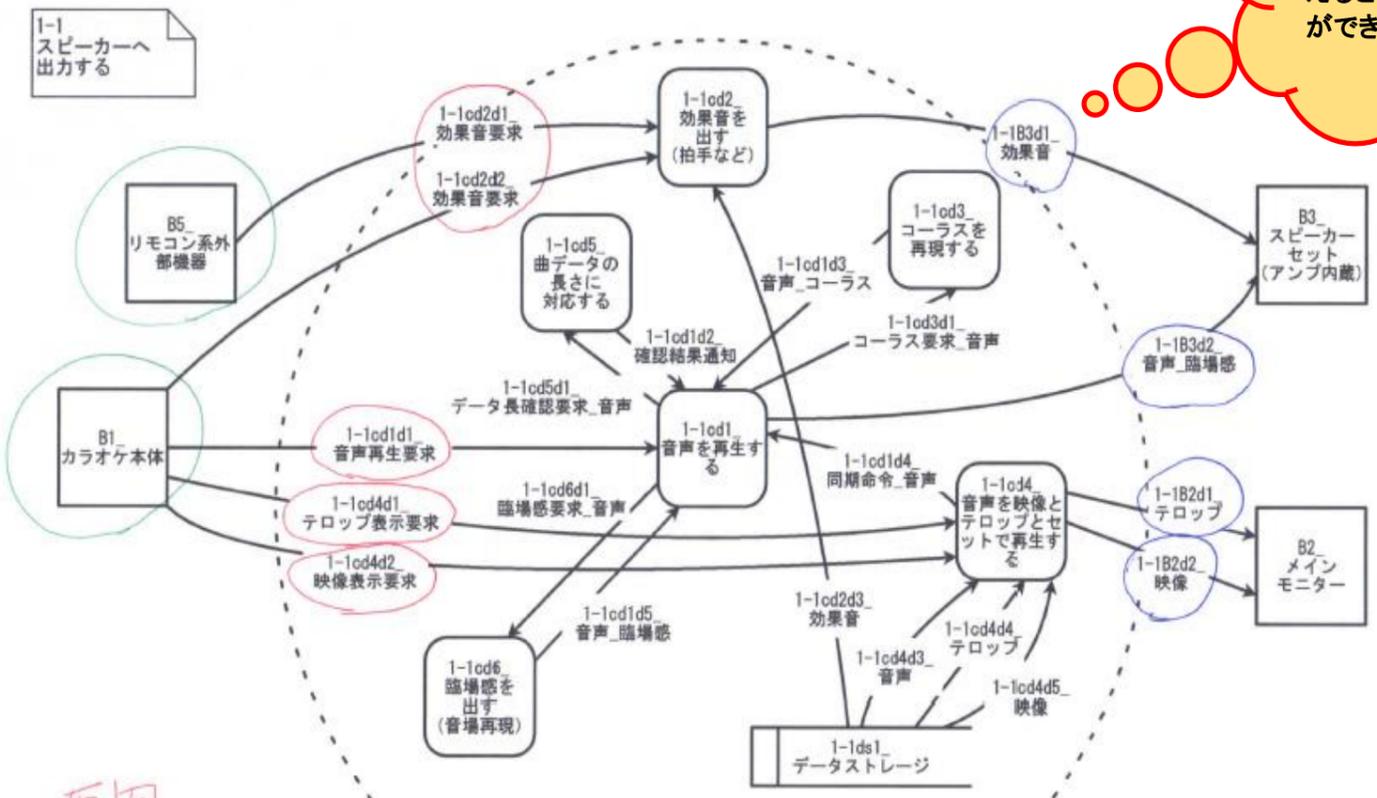
エラー推測(ピンポイント的に抽象度を変更する)	備考
G1 デシジョンテーブルを作成する	粒度を変更してみる
G2 原因結果グラフを作成する	粒度を変更してみる
G3 経験	過去に携わった製品との類似点
G4 直感	その場の思い付きでもよい
G5 過去事例	本製品で抽出されたバグの傾向
G6 ありそうなバグ	ネット上で検索するなどして探す
G7 同値分割	粒度を変更してみる
G8 境界値分析	粒度を変更してみる

本通信カラオケシステムに関する機能要求(テスト条件)とフィルターとするテスト観点との関連性を示すマトリクス(テストケース一覧)である。
 抜け漏れ防止フィルター(ISO25010)で使用するバグ抽出戦略をフィルターとして対応付けている。バグ抽出フィルターを使用した場合は番号(①②③...)を入力する。
 別途エラー推測フィルターはバグ抽出フィルターの補填であり、テスターに依存するためテストケースとのトレーサビリティは記載せず、○のみを入力する。(記載は番号を優先とする)

エラー推測 バグ抽出	G1~G8																															
	BUG1				BUG2				BUG3				BUG4				BUG5				BUG6				BUG8							
抜け漏れ防止(ISO25010)																																
1-1cd1 音声再生する		○																														
1-1cd2 効果音を出す(拍手・歓声・口笛)		○																														
1-1cd3 コーラスを再現する		○																														
1-1cd4 音声映像とテロップとセットで再生する		○																														
1-1cd5 曲データの長さに対応する		○																														
1-1cd6 臨場感を出す(音場再現)		○																														
1-1cdm1																																

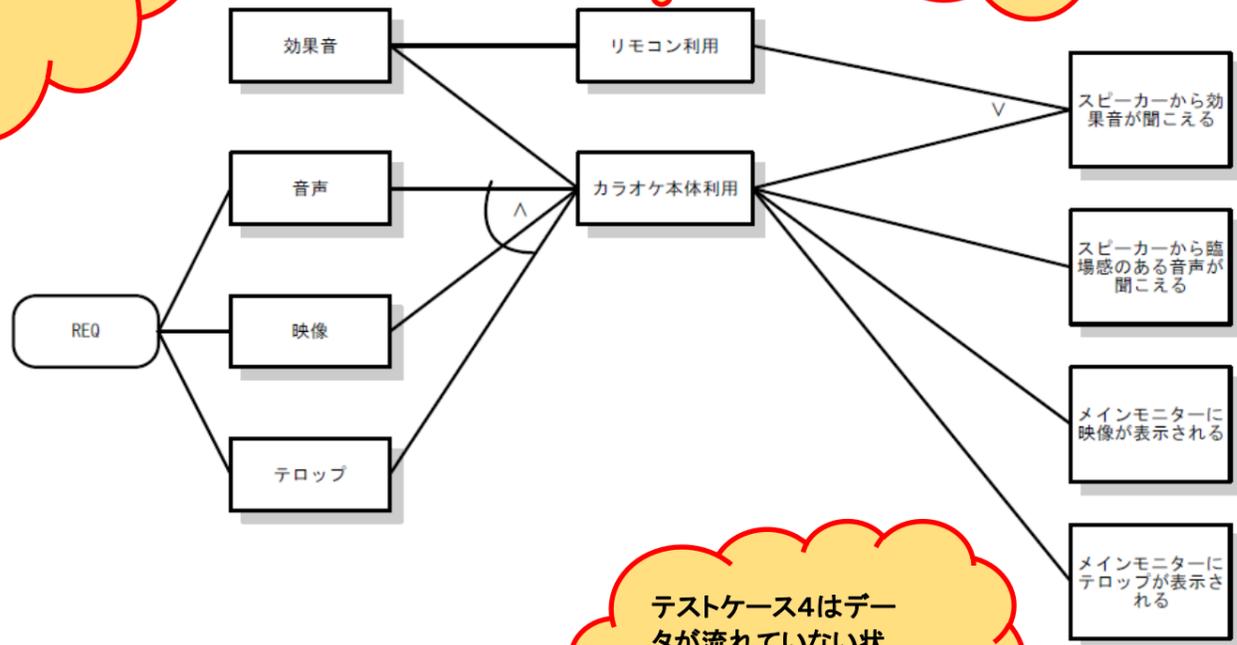
3つのフィルターを通せば機能要求と要件を網羅するためのマトリクスが完成する!

シナリオテストの組み合わせテストケースを作る
 ~テスト技術の応用へ~
 DFDの正しさがカギ



文章から読み取るのは難しい原因結果グラフだが視覚的に見ると容易に作ることができる。

データの流れの観点で抜け漏れがないように設計する。



テストケース4はデータが流れていない状態のため調査不要です。

データの種類の意識せず、データの流れや処理の違いに注目。

偶発的に発生するデータの衝突など想定外のエラーも見つけて取り除く。

原因
結果
中間

「中間」は外部(データの発生源/行き先)だけとは限らないため、慎重に分析しよう!

	ノード名	テストケース			
		1	2	3	4
原因	効果音	T	T	F	F
	音声	T	T	T	F
	映像	T	T	T	F
	テロップ	T	T	T	F
中間	リモコン利用	T	F	F	F
	カラオケ本体利用	T	T	T	F
結果	スピーカーから効果音が聞こえる	T	T	F	F
	スピーカーから臨場感のある音声が聞こえる	T	T	T	F
	メインモニターに映像が表示される	T	T	T	F
	メインモニターにテロップが表示される	T	T	T	F

シナリオID	テスト条件	ISO25010	バグ抽出戦略	エラー推測	テストケース	テスト合格条件	優先順位
1-1 sys1	1-1 cds1	ISO1-1-2	-	G1/G2_テストケース1	ユーザーが楽曲演奏を開始するとすぐに臨場感のある音声再生され、映像とテロップがメインモニターに表示された後、歌い始めている。視聴者が途中でリモコンから効果音を選択して場を盛り上げている。	スピーカーから効果音と臨場感のある音声が再生されていること。またメインモニターには映像とテロップがセットで表示されていること。	A
1-1 sys1	1-1 cds1	ISO1-1-2	-	G1/G2_テストケース2	ユーザーが楽曲演奏を開始するとすぐに臨場感のある音声再生され、映像とテロップがメインモニターに表示された後、歌い始めている。視聴者が途中でカラオケ本体から効果音を選択して場を盛り上げている。	スピーカーから効果音と臨場感のある音声が再生されていること。またメインモニターには映像とテロップがセットで表示されていること。	A
1-1 sys1	1-1 cds1	ISO1-1-2	-	G1/G2_テストケース3	ユーザーが楽曲演奏を開始するとすぐに臨場感のある音声再生され、映像とテロップがメインモニターに表示された後、歌い始めている。視聴者は静かに音声に耳を傾けている。	スピーカーから臨場感のある音声が再生されていること。またメインモニターには映像とテロップがセットで表示されていること。	A

イイですとチームの特徴

～低コスト+短納期+高品質～

モデリングやバグを取り除く早さの秘密

DMMを作る過程はExcelが速い！

①機能要求の一覧化。

②分析→分解。

③階層化。

機能要求ID	機能名	上位要求ID	利用者グループ	備考
1-1	楽曲演奏		ユーザー	
1-1-1	MIDI再生	1-1	ユーザー	
1-1-2	MP3再生	1-1	ユーザー	
1-2	映像再生		ユーザー	

機能要求ID	機能名	上位要求ID	利用者グループ	備考	備考2
1-2-1	一般映像再生	1-1	ユーザー	楽曲演奏 ガイドメロディ、採点用リファレンス、演奏マークをフォーマットに含む。マークには種類がある	
1-2-2	1:1映像再生				
1-2-2-1	本人映像再生				
1-2-3	特殊映像再生				
1-2-4	ジャケ写表示				
1-3	テロップ表示			マークの種類 ・各コーラスのコーラスエンド ・各コーラスのサビ開始位置 ・歌唱開始位置 ・歌唱終了位置 MIDIとMP3フォーマットがある	
1-3-1	ワイプ動作				
1-3-2	文字サイズ				
1-3-3	ルビデータ表示				
1-4	演奏系操作				
1-4-1	スタート/やり直し				
1-4-2	演奏中止				
1-4-3	テンポコン(テンポコントロ				
1-4-4	キーコン(キーコントロール				

DMM(機能構成図)							機能名	上位要求ID	利用者グループ	備考
階層レベル0	階層レベル1	階層レベル2	階層レベル3	階層レベル4	階層レベル5					
通信カラオケシステム	1-1 スピーカーへ出力する	1-1 楽曲を演奏する				楽曲演奏		ユーザー		
		1-1-1 MIDIを再生する				MIDI再生	1-1	ユーザー		
		1-1-2 MP3を再生する				MP3再生	1-1	ユーザー		
		1-2 映像を再生する				映像再生		ユーザー		
		1-2-1 一般映像を再生する				一般映像再生	1-2	ユーザー		
		1-2-2 1:1映像を再生する				1:1映像再生	1-2	ユーザー		
		1-2-2-1 PV等の本人映像を再生する				本人映像再生	1-2、1-2-2	ユーザー		
		1-2-2-2 特殊な映像を再生する				特殊映像再生	1-2、1-2-2	ユーザー		
	1-n,2-2,3-n メインモニターに表示する	1-2-4 ジャケ写を表示する				ジャケ写表示	1-2	ユーザー		設定のみ オーナー
						テロップ表示	1-1	ユーザー		

テストはバグを念頭に実行する！

結合ID	テスト条件	ISO25010	バグ抽出戦略	エラー推測	テストケース	テスト合格条件	優先順位
1-1 mg1	1-1 cdm1	ISO1-1-2	-	G6_音声と映像とテロップがズレている可能性がある	B1_カラオケ 本体から楽曲演奏を開始する	楽曲の音声と映像とテロップがメインモニターに表示されること	A
1-1 mg3	1-1 cdm3	ISO1-1-2	-	G4_効果音が操作者の思い通りに再生されていない可能性がある	B1_カラオケ 本体から効果音を選択する	効果音がスピーカーから再生されること	A
1-1 mg2	1-1 cdm2	ISO1-1-2	-	G4_効果音が操作者の思い通りに再生されていない可能性がある	B5_リモコン系外部機器から効果音を選択する	効果音がスピーカーから再生されること	A
1-1 mg6	1-1 cdm5	ISO1-1-2	-	G6_臨場感があまり再現されていない可能性がある	臨場感を要求される楽曲を選曲する	臨場感のある音声と映像がスピーカーから再生されること	A
1-1 mg7	1-1 cdm6	ISO1-1-2	-	G6_コーラスがあまり再現されていない可能性がある	コーラスを要求される楽曲を選曲する	コーラスが再現された音声と映像がスピーカーから再生されること	A

1つのテストケースに1つのバグ抽出戦略を立てられないときはありそうなバグを推測したり、直感を働かせて潜んでいるバグを見つけるアイデアを絞り出します。あとで直感なら直感と分かるようにしておくことでメンテナンスもし易くなり、新たなエラー推測を取り入れられる仕組みを作ることが大切です。