

チーム名：イイてすと

イ	イ	
て	す	と

1メンバー：飯沼

1. チーム紹介

- 普段はソフトウェア検証に携わっており、要件がテストケースとマッチしていると感じられることでテストする自分がイイと思えるようなテストを目指している。

2. 概要

- ASTER様の開発部門からテスト設計を依頼された。実施可能なテスト内容の全容が分かるテスト設計書を作成し、開発部門がテストの十分性を確認する。



3. 方針その1

- 本テスト設計の導入により、従来よりも全体のコストを抑え、短い納期で品質が良いものを目指す。



具体的には業務プロセスの見える化を行うことで効率化すること。また分析に無駄のないことである。

4. 方針その2

- 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す。



具体的にはソフトウェアの要件とテストケースがマッチしていると感じられるように設計することである。

5. テスト設計の特徴その1

- DMM（機能分析表）を取り入れたモデリングへのアプローチ方法とソフトウェア品質やバグを分析する力である。



DMMとは3 × 3 マスを一つのDMMとする整理ツールと分析ツールの2つの役割を持ち、モデリングやアイデア出しなどを助けてくれるものである。

(5-1)なぜDMMなのか

- テスト対象をモデル化するためにDFDを使いたいが、階層化するためには抽象化の度合いを揃える技術など、習得するのが難しいという課題があった。そこでDMM（機能分析表）を用いて先に階層化を行い、DFDを作図するための設計図を作ることにしたのが導入するきっかけだった。

(5-2) 良かった点/悪かった点 ～DMMとは何なのか～

○ 良かった点



思わぬ発見があった。

整理だけでなく分析目的にも向いていた。

○ 悪かった点



事前にExcel上で分割することで解決した。

1つのDMMは8つまでに分割が制限されるため、分割目的では使いにくかった。

6. テスト設計の特徴その2

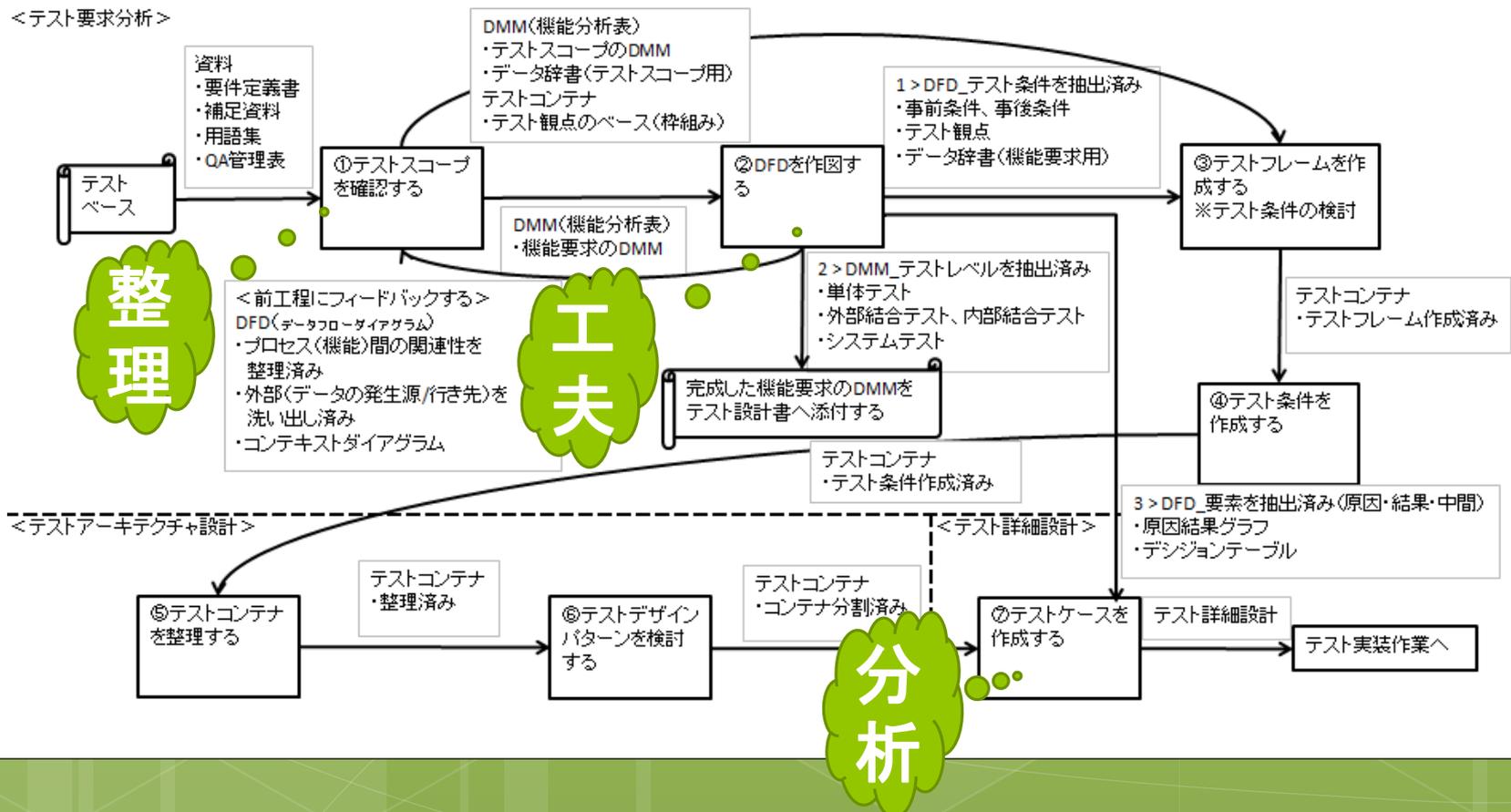
- DFD（データフローダイアグラム）から生み出されるテストケースを使っている。



テスト対象をモデル化することで可視化し、システム理解を深め、プロセス（機能）間の関連性の整理と外部（データの発生源/行き先）との入出力の整理を行うことによって、高い精度で抜け漏れのないテストケースを効率よく作成できる。

7. テスト設計プロセス ～DMM活用ポイント～

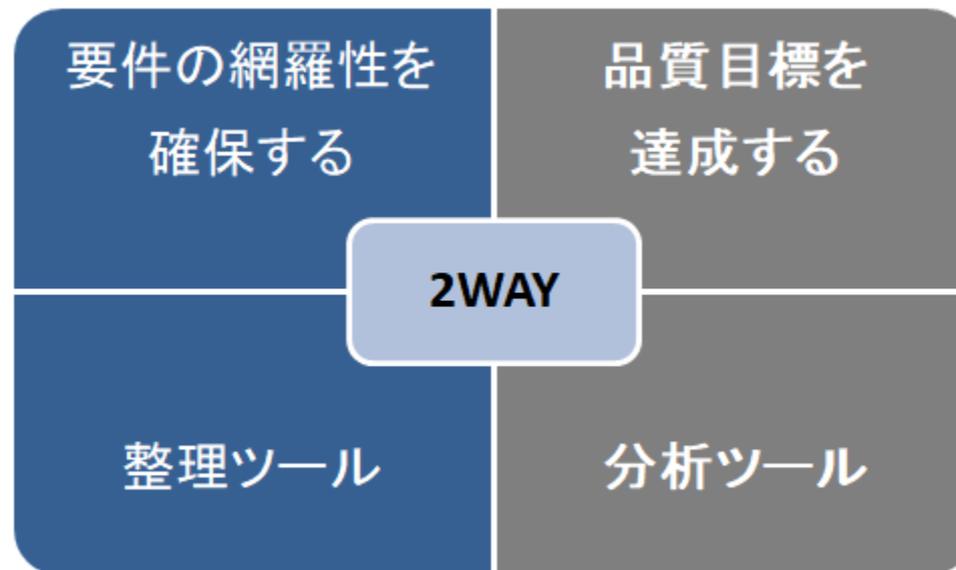
<テスト要求分析>



8. DMM（機能分析表）

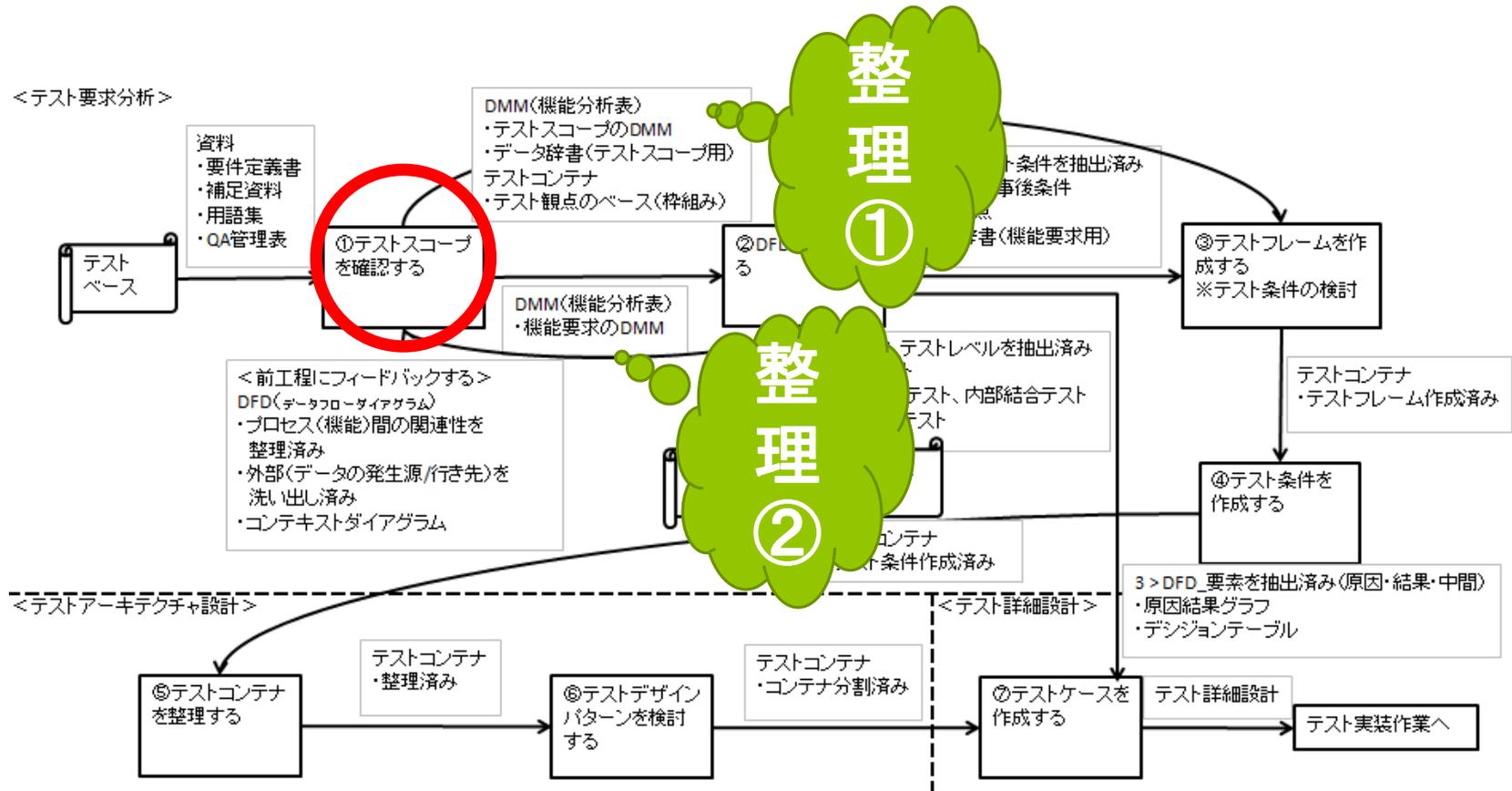
～Diamond Mandala Matrix～

- DMMは中央のマス目にコンテキスト名を書き、周囲の8つのマス目には整理/分析した対象の名前を書くというものである。



(8-1) DMMでプロセスを見える化

～テストスコープと機能要求の整理～



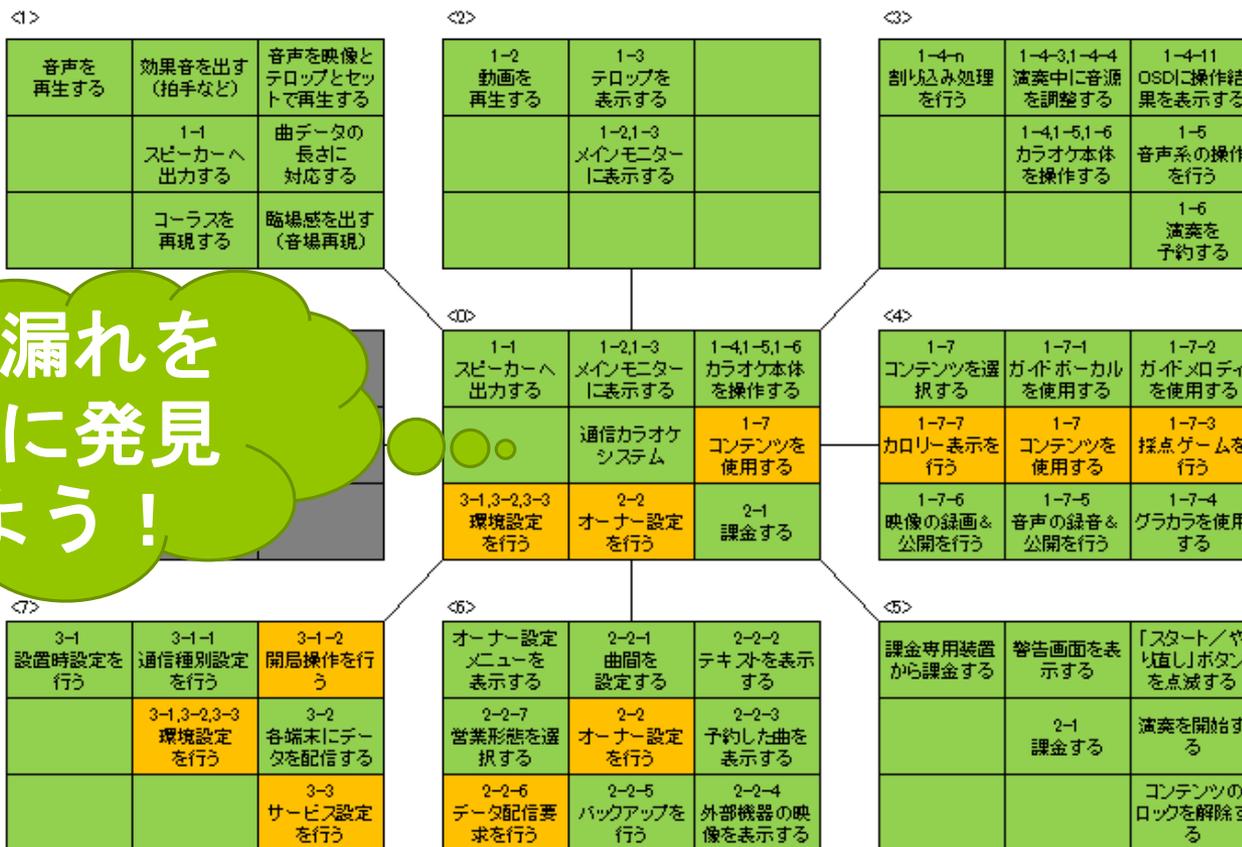
(8-2)最上位のDMM ～機能要求の全体を俯瞰する～

1-1 スピーカーへ 出力する	1-2,1-3 メインモニター に表示する	1-4,1-5,1-6 カラオケ本体 を操作する
	通信カラオケ システム	1-7 コンテンツを 使用する
3-1,3-2,3-3 環境設定 を行う	2-2 オーナー設定 を行う	2-1 課金する

中央から
見ていく
のがコツ。

インパ
クトの
高いテ
スト領
域はオ
レンジ
色に！

(8-3) どんどん階層化していく ～機能要求の整理～



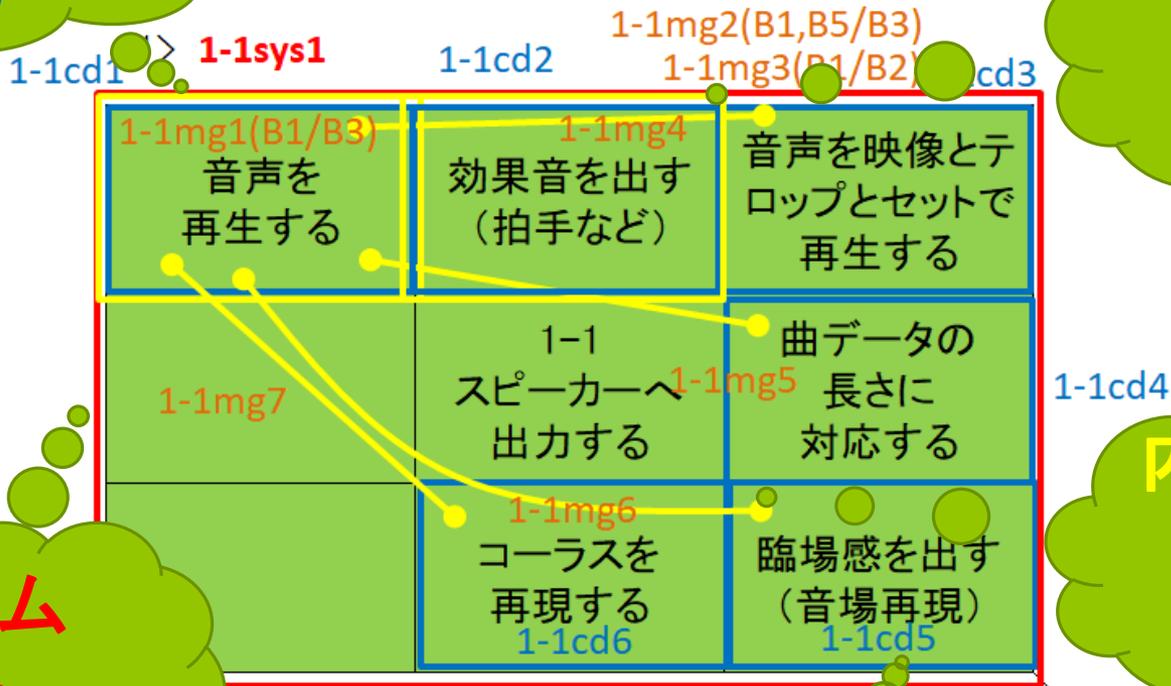
抜け漏れを
早期に見つけよう！

(8-5) こんな使い方もできるDMM

～工夫した点～

単体テスト
(枠)

外部結合
テスト
(枠)

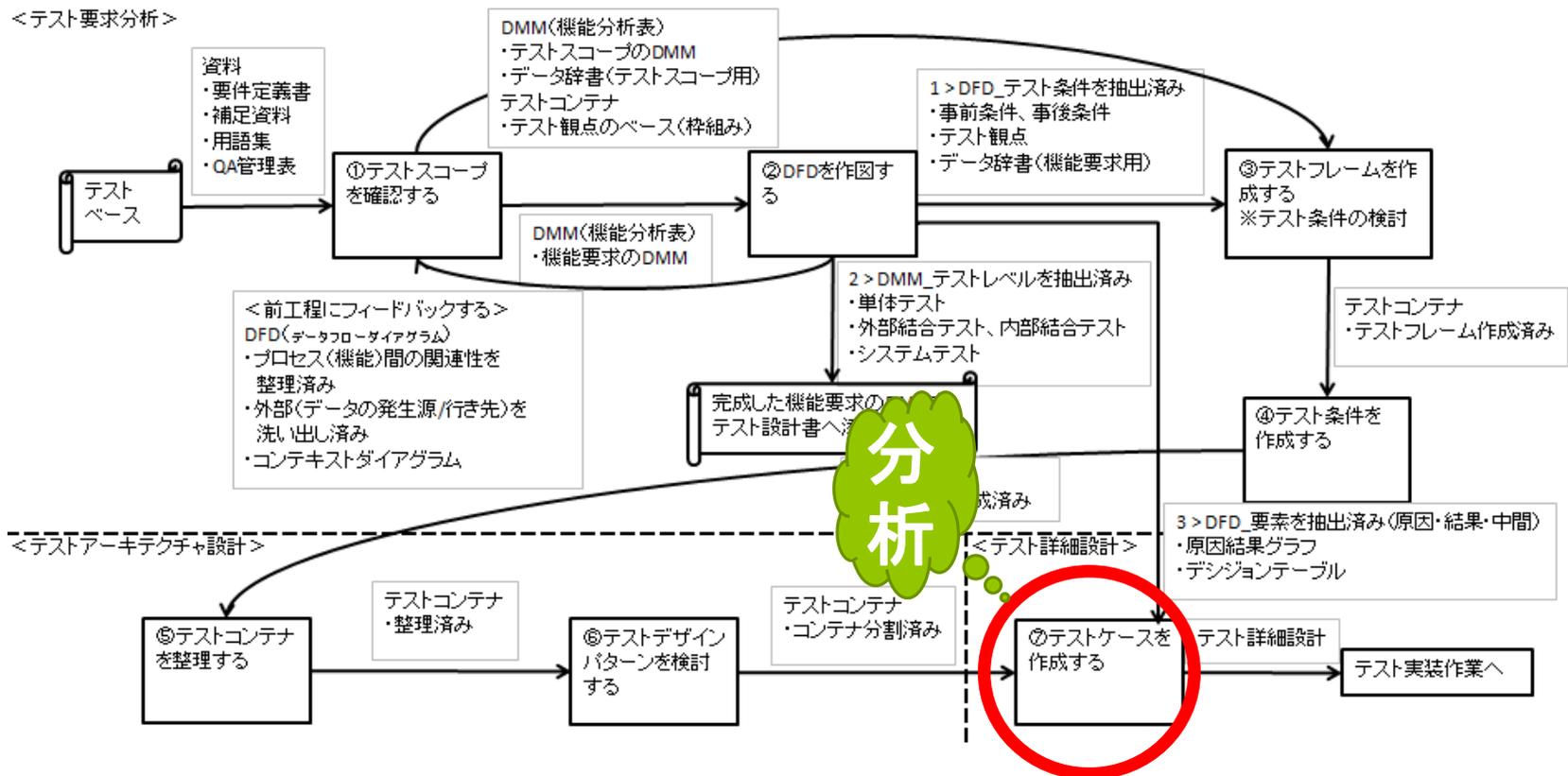


システム
テスト
(枠)

内部結合
テスト
(罫線)

すべてナンバー管理!

(8-6)DMMで秀でたエラー推測 ～テストに必要な分析力～



(8-7) ストイックなテスターへ！ ～分析には気合も～

不具合を絶対に見つけたい気持ち
がピンポイントでエラーを推測する。

結果グラフ 作成する	経験
エラー推測	直感
同値分割	あり

アイデアを無理やり
絞り出す！

9. テスト効率を考えたDFD ～理解し易さ～

- テストし易い大きさのDFDを作る。



DMM（機能分析表）なら丁度良い大きさのDFDが自然にできるようになる。

(9-1) リバーースするプロセス ～DFDの作り方～

- ①上層からトップダウン式に作る。



関連性を整理し、必要な外部を洗い出す。

- ②下層からボトムアップ式に作り直す。

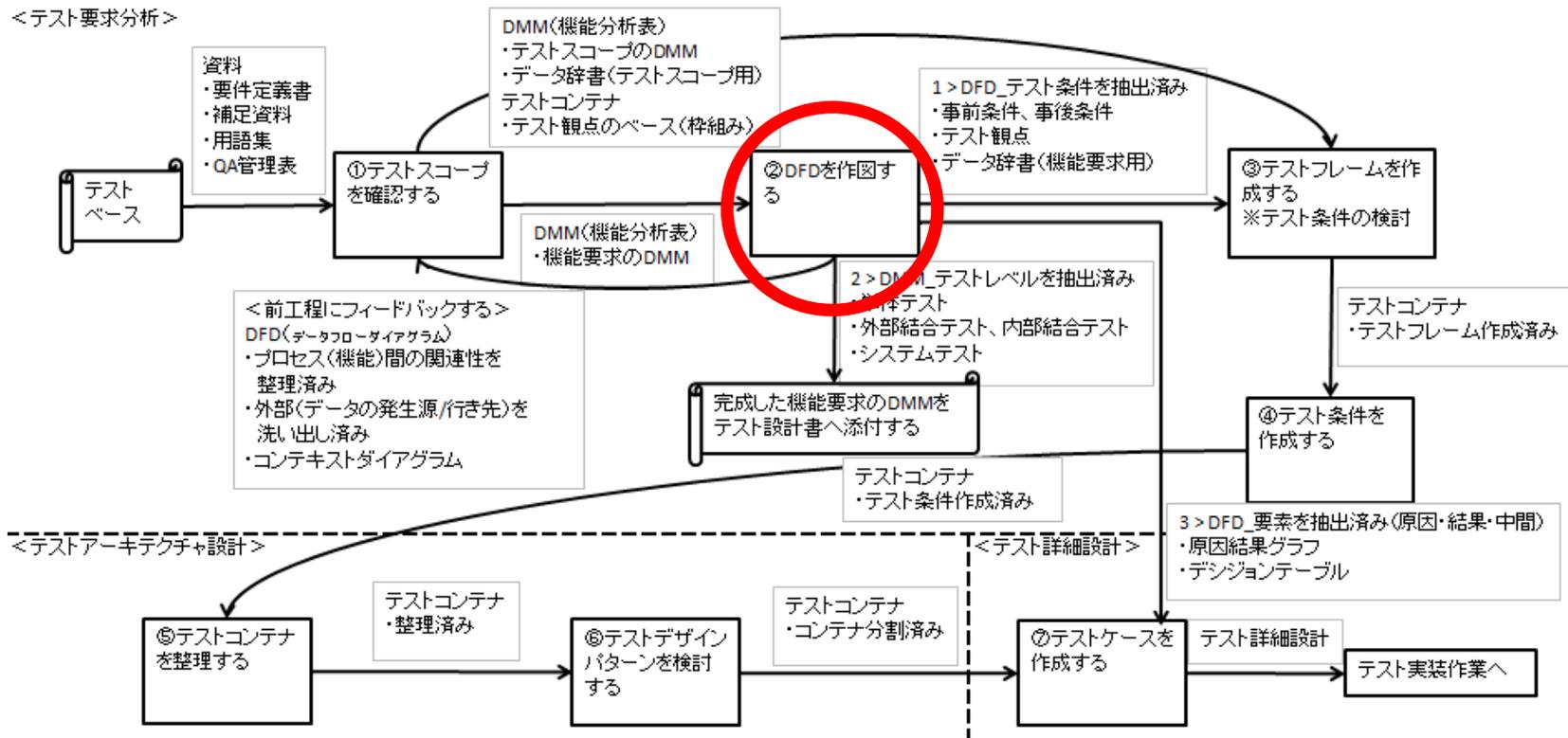


システム構成図と洗い出した外部とで入出力の整理を行う。

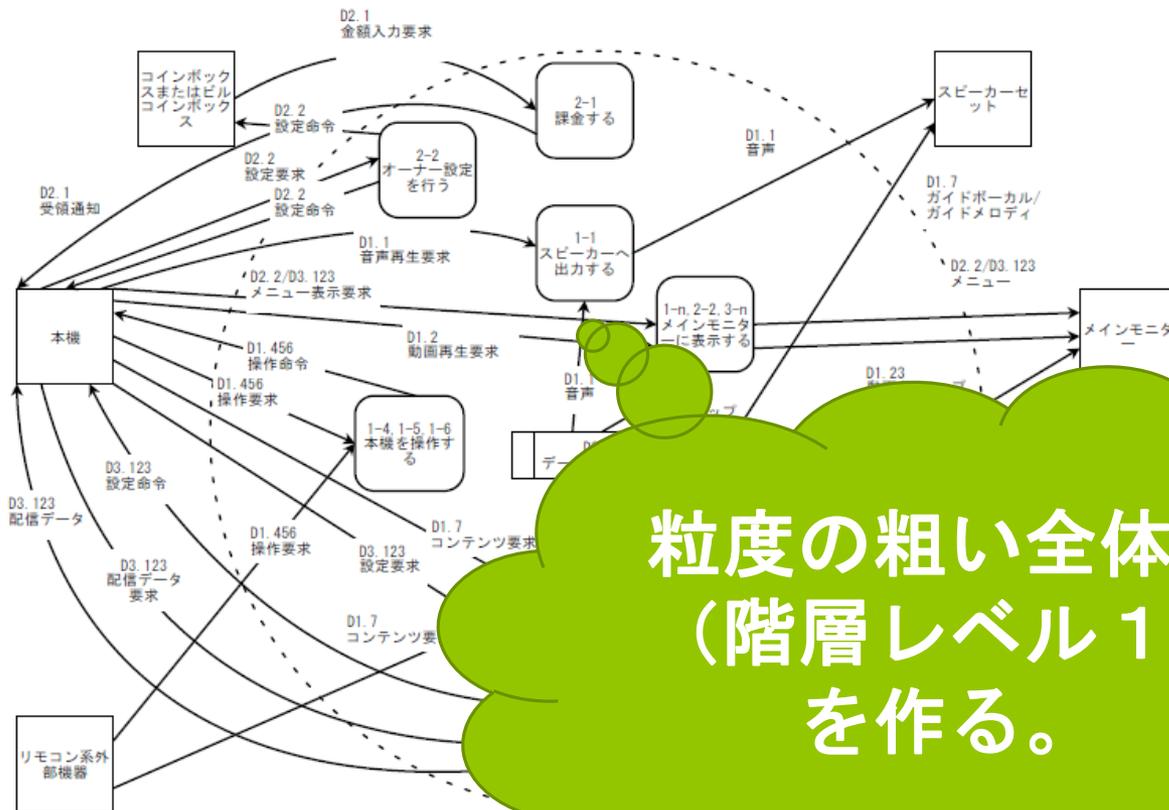
(9-2) DFDを作図する

～データフローダイアグラム～

<テスト要求分析>

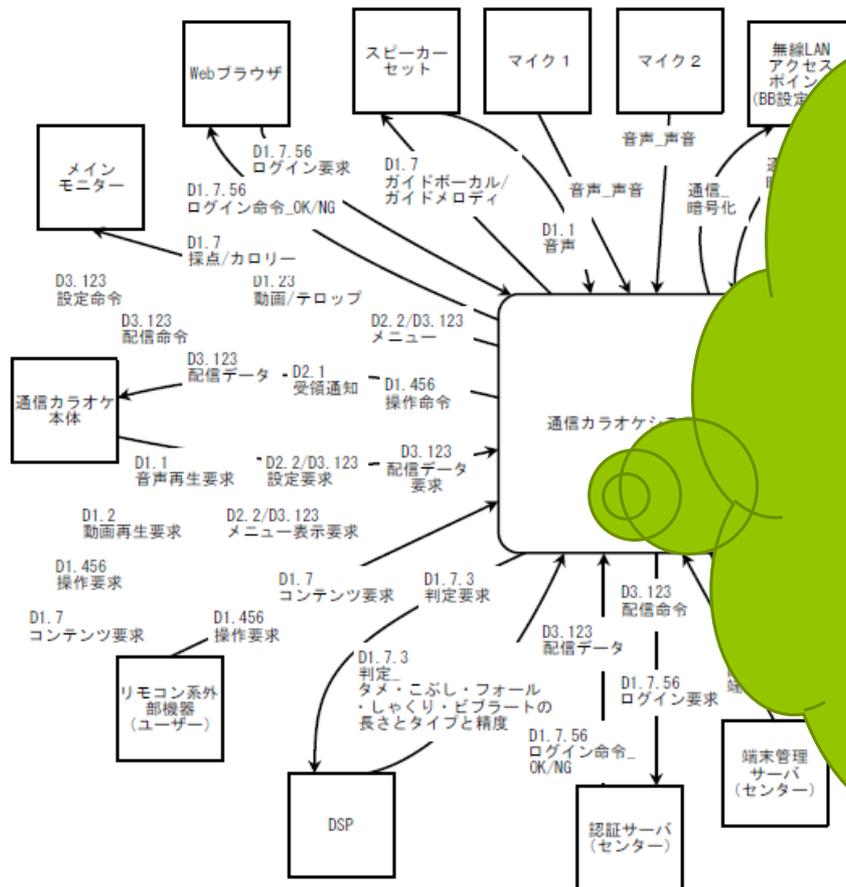


(9-3) トップダウン型



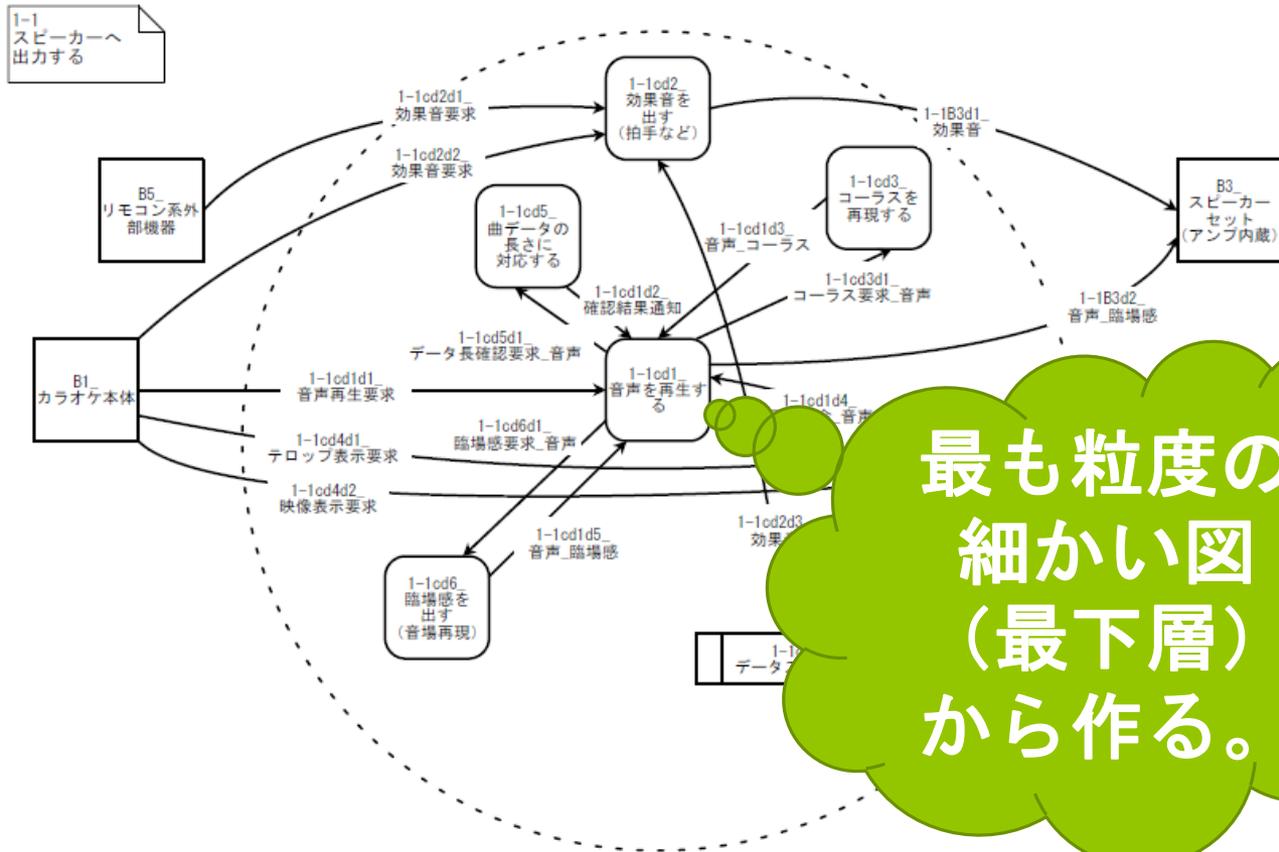
粒度の粗い全体図
(階層レベル1)
を作る。

(9-4) コンテキストダイアグラム ～システム全体の文脈を表す～



抜け漏れ防止
に外部と通信
カラオケシス
テムとの文脈
を表すデータ
の流れを書
く！

(9-5)ボトムアップ型



(9-6) データ辞書を作成する ～DMMとDFDを読むために～

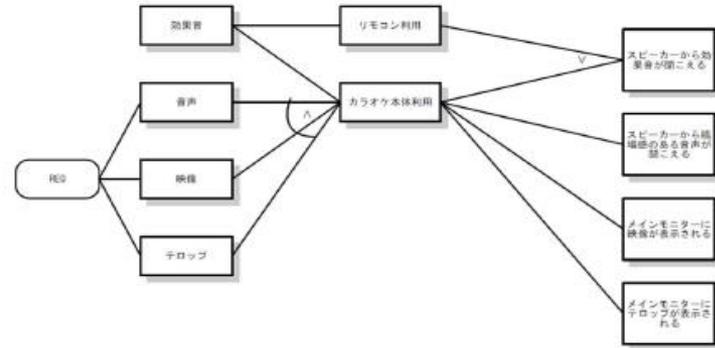
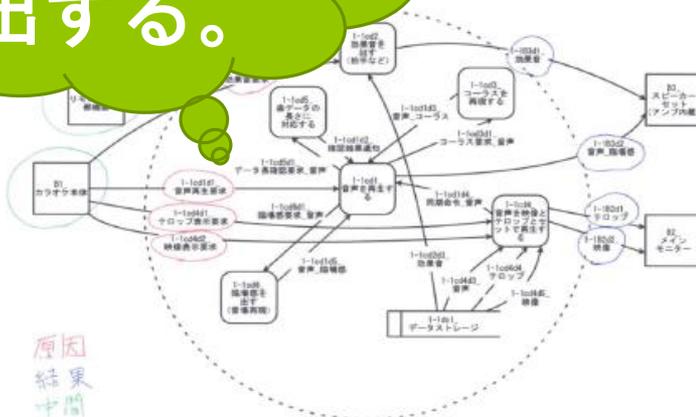
コンテキスト名	1-1 スピーカーへ出力する	作成日	2017年12月5日	
DFD名	1-1 スピーカーへ出力する	更新日	2017年12月5日	
		更新者	藤沼	
プロセス名	入力(データの発生源)	処理の概要	出力(行き先)	備考
1-1cd1 音声を再生する	<ul style="list-style-type: none"> B1. 通信カラオケ本体(本機) 曲データの長さに対応する コーラスを再現する 音声を映像とテロップとセットで再生する 臨場感を出す(音場再現) 	<ul style="list-style-type: none"> 音声再生要求を受信し、音声再生を開始する 曲データの長さを制御する コーラスを制御する 音声を映像とテロップと同期し、セットで再生する 臨場感を制御する 	<ul style="list-style-type: none"> B3. スピーカーセット(アンプ内蔵) 曲データの長さに対応する コーラスを再現する 臨場感を出す(音場再現) 	
1-1cd2 効果音を出す(拍手・歌声・口笛)	<ul style="list-style-type: none"> B1. 通信カラオケ本体(本機) B5. リモコン系外部機器 	拍手・歌声・口笛の効果音を出す	B3. スピーカーセット(アンプ内蔵)	
1-1cd3 コーラスを再現する	音声を再生する	コーラス(ADPCMとMIDIを同時再生)を再現する	音声を再生する	
1-1cd4 音声を映像とテロップとセットで再生する	<ul style="list-style-type: none"> B1. 通信カラオケ本体(本機) データストレージ 	<ul style="list-style-type: none"> 音声を映像とテロップと同期し、セットで再生する データストレージから音声と映像を受信する 	B2. メインモニター	同期処理とは音声は映像とテロップとセットで再生される要件から想定している
1-1cd5 曲データの長さに対応する	音声を再生する	音声を再生し、曲の長さが30分以内であることを確認する	音声を再生する	
1-1cd6 臨場感を出す(音場再現)	音声を再生する	臨場感を出し、音場を再現する	音声を再生する	

DMMやDFDは見るために、
データ辞書はそれを読むために必要になる。

(9-8) テストケースの抽出 ～DFDと原因結果グラフ～

要素を抽出する。

DFDから要素を抽出し、原因結果グラフを作成する。



作成した原因結果グラフからデシジョンテーブルを作成し、組み合わせの整合性を担保する

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	スピーカから効果音が聞こえる	T	T	F	F
	スピーカ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

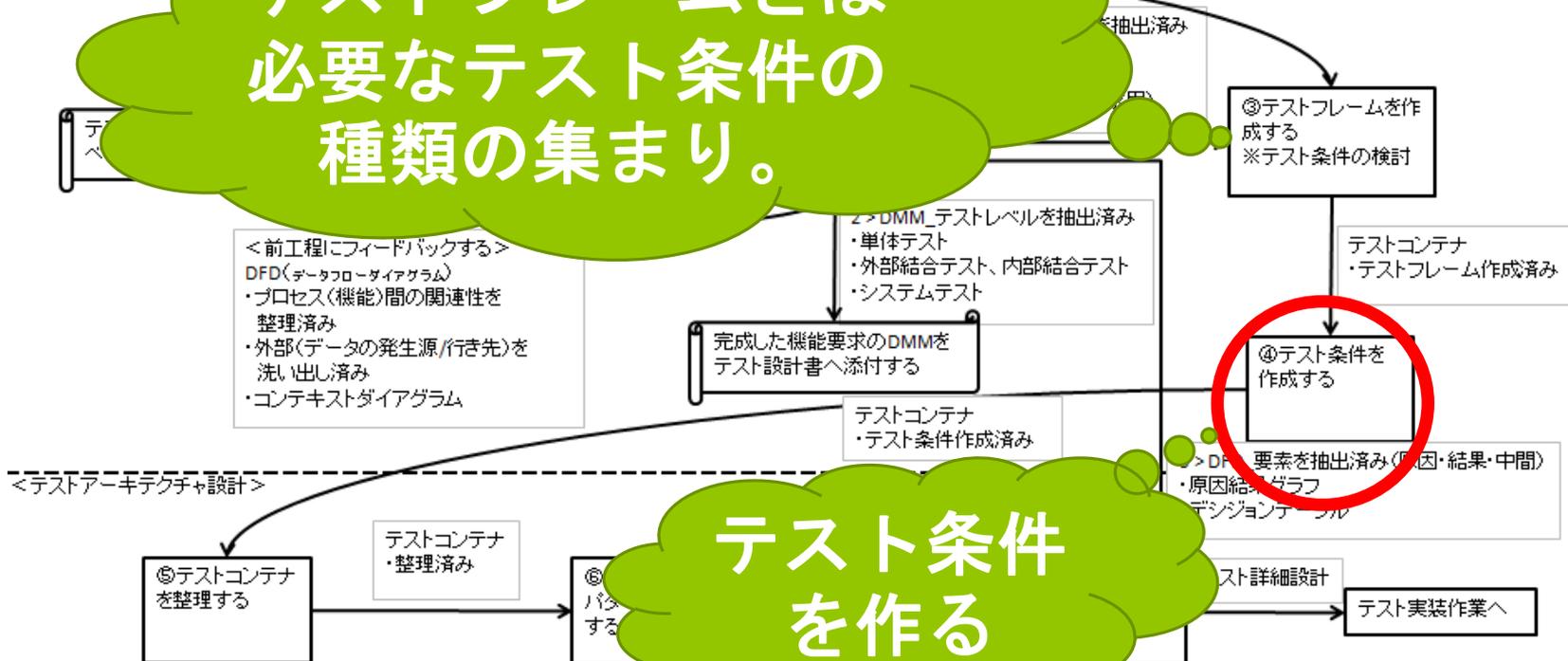
抜け漏れの無い
組み合わせを作成する。

(9-9) テスト条件を作成する

～テストフレームを作成した後で～

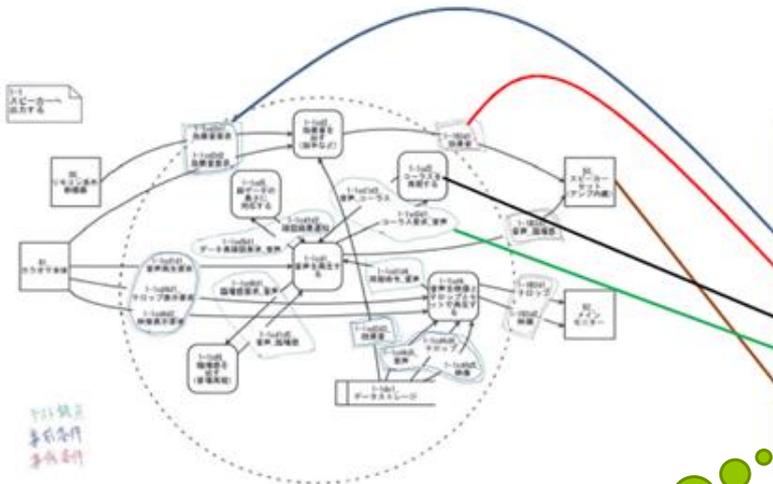
<テスト要求分析>

テストフレームとは
必要なテスト条件の
種類の集まり。



(9-10)DFDとテストベースの関係

～テストケースを作成するために～



カラオケ演奏機能	
機能要求ID	1-1
機能名	楽音再生
概要	以下のフォーマットの音源データが再生できること。 ・MIDI ・MP3
	また両フォーマットとも楽音データにはガイドメロディ、視点用リアレンジ等および演奏マーズを含んでいる。
	演奏マーズは以下の種類がある ・各コーラスのロータスシンク ・各コーラスのサビ開始位置 ・歌開始位置 ・歌終了位置
	楽音再生時には各コーラスで指定された楽音マーズ、マロップマーズとセットで再生される。
	楽音再生可能なマーズの長さは1分以内である。マーズ再生開始時刻(再生開始)に対する必要があり、楽音データの相対的時間は楽音再生開始時刻の100%以内である。
	楽音マーズはサブライヤーから提供されるため、その扱いは別項とする。
上階要求ID	
利用者グループ	ユーザー
備考	
機能要求ID	1-1-1
機能名	MIDI再生
概要	SMF(スタンダード MIDI フォーマット)形式のデータが再生可能なマーズ。
	音源の再生するマーズは1分以内である。
	マーズ再生開始時刻(再生開始)に対する必要があり、楽音データの相対的時間は楽音再生開始時刻の100%以内である。
	楽音の音声データのフォーマットとして、「標準音」と呼ばれる。
上階要求ID	1-1
利用者グループ	ユーザー
備考	
機能要求ID	1-1-2
機能名	MP3再生
概要	MPEG-1 Audio Layer3のオーディオフォーマット形式のデータが再生可能なマーズ。
	楽音の音声データのフォーマットとして、「標準音」と呼ばれる。
上階要求ID	1-1
利用者グループ	ユーザー
備考	

テスト条件に
インプット!

10. DMMのまとめ

～方針の回収～

- DMM化するプロセスをExcelで見える化できた。



Excelで！

機能要求を一覧化→分析→分解→階層化



DMMへ！

ExcelからDMMへスムーズに移行でき、
工程に無駄がなかった。

1 1. DFDのまとめ ～方針の回収～

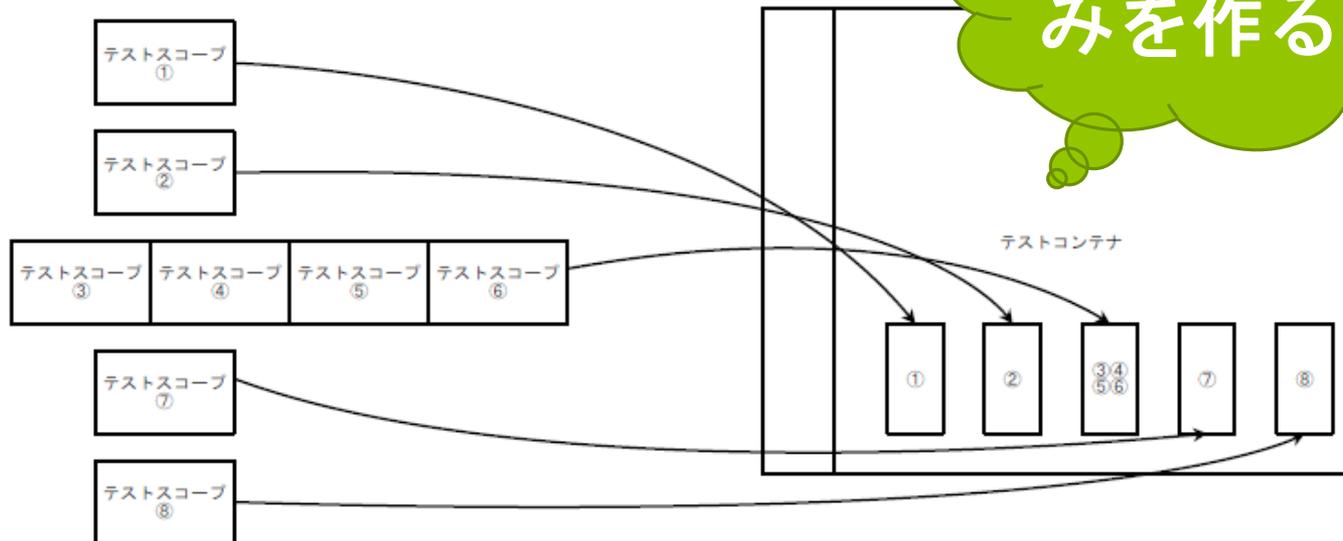
- DFDの作図はDMMを設計図として見える化したことでスムーズに行うことができ、またDFDを作図するにあたってはプロセス（機能）間の関連性の整理と外部（データの発生源/行き先）との入出力の整理の工程を分けることで、作り直しの手間を最小限にできた。また図の精度もシナリオテストまでのテストケースを作成可能なほど高かった。

12. テストコンテナ

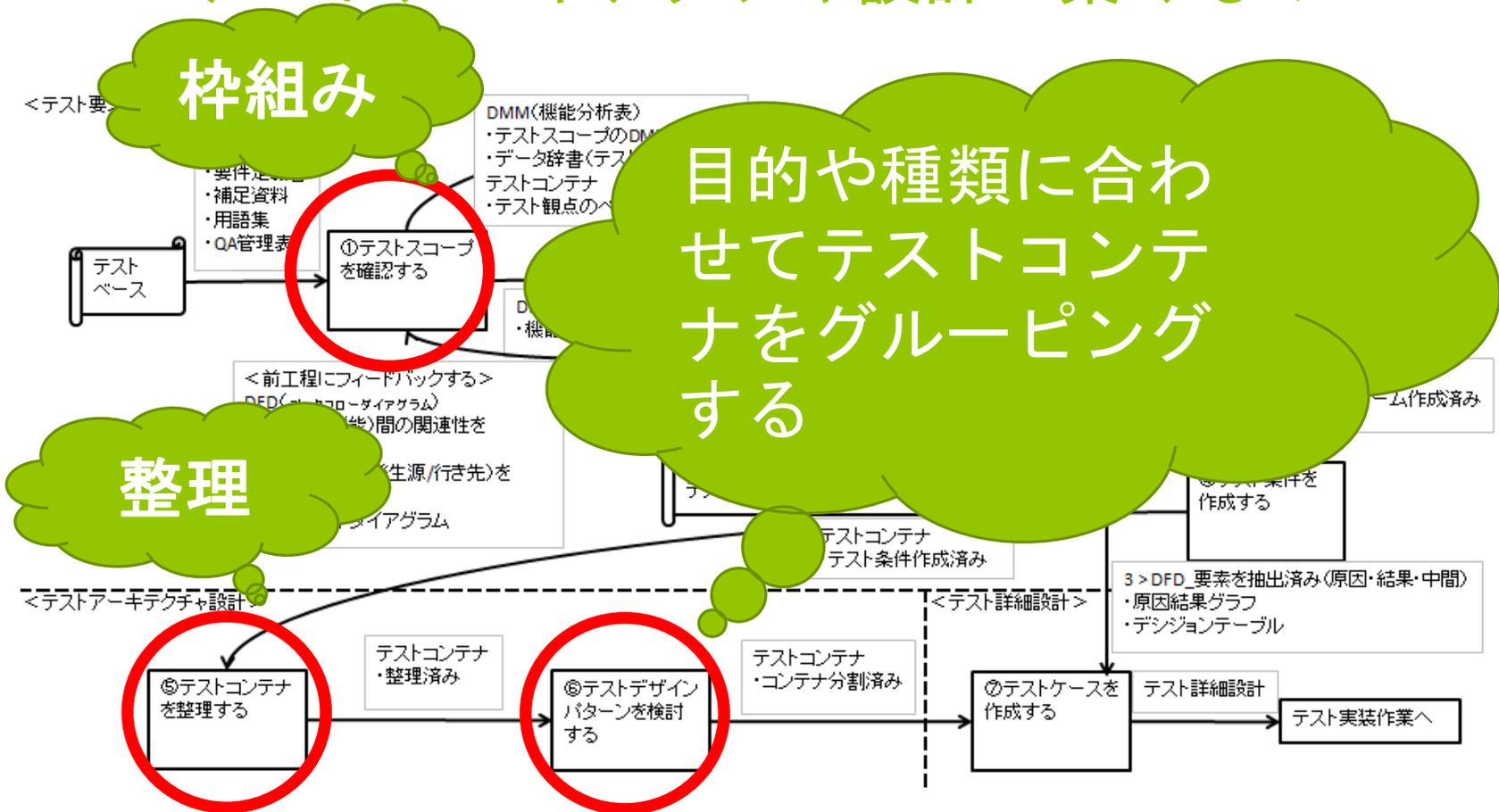
～テスト観点のグループ～

- 同じまとまりとしてテストすべきテスト観点のグループを作る。

まず枠組みを作る。



(12-1) テストコンテナへの導き ～テストアーキテクチャ設計へ繋ぐもの～



(12-2) テストコンテナの整理 ～テストのデザインパターンを検討する～

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③④⑤⑥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	

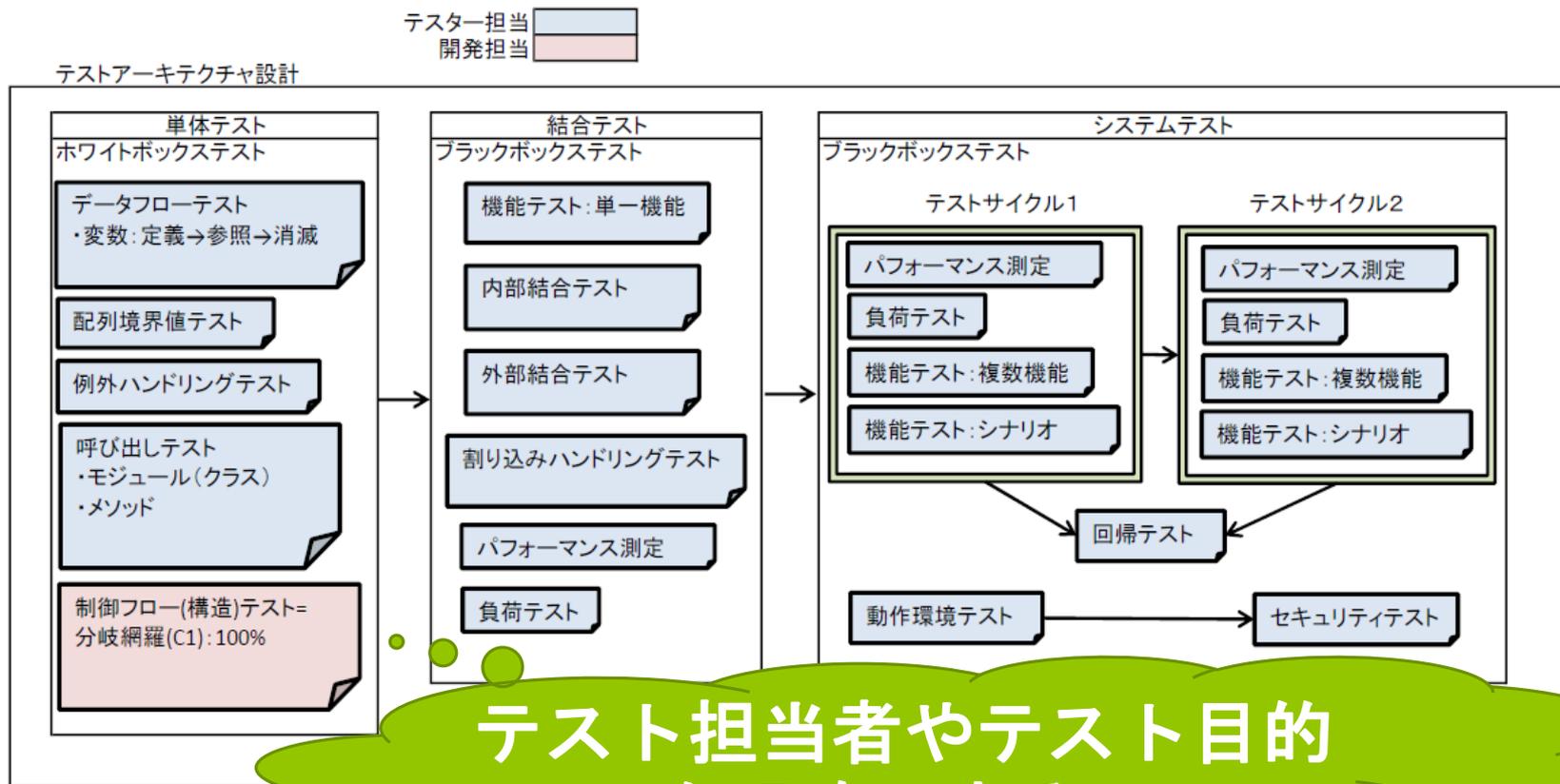
テスト詳細設計タイプ	シート名	備考
テスト詳細設計_開発用	単体テスト	テスト構造のみ
テスト詳細設計_テスター用	データフローテスト	テスト構造のみ
	配列境界値テスト	テスト構造のみ
	例外ハンドリングテスト	テスト構造のみ
	呼び出しテスト	テスト構造のみ
	機能テスト(単一機能)	
	機能テスト(複数機能)	
	機能テスト(シナリオ)	
	内部結合テスト	
	外部結合テスト	
	割り込みハンドリングテスト	
	パフォーマンス測定	
	負荷テスト	
	動作環境テスト	
	セキュリティテスト	

テストコンテナを目的別に分割する。

(12-3) テストコンテナの関連図 ～重複するテストの行方～



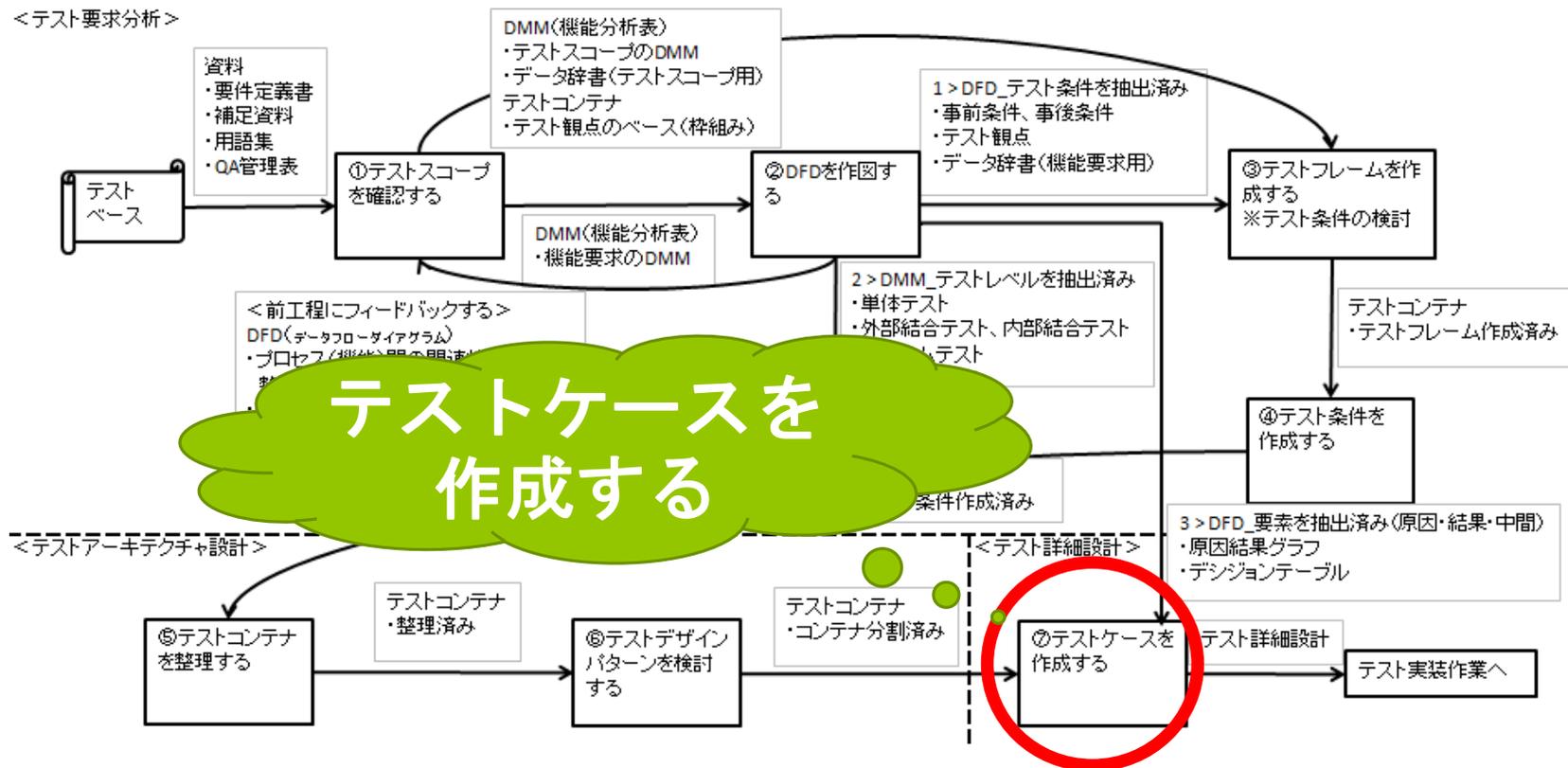
13. テストプロセスの全体図 ～テストアーキテクチャ設計へ入る前に～



テスト担当者やテスト目的
を明確にする。

14. テスト詳細設計

～機能要求や要件の抜け漏れを防ぐ～



15. 自分でイイと思えたか ～方針の回収～

- プロセスを見える化したことによって要件からテストケースまで高いトレーサビリティを保つことができ、作成したテストケースは取り除きたいバグを念頭に要求ともマッチしてイイテストが実行できると思えるものが作れた。

16. まとめ

- テスト設計コンテストを通じ、テストでは要件や要求の抜け漏れだけでなく、限られた時間で潜在するバグをどうすれば取り除けるかを追及することが必要だと気づいた。

以上です。
ご清聴ありがとうございました。