



テスト設計コンテスト プレゼンテーション



STUDIO IBURI

STUDIO IBURI紹介



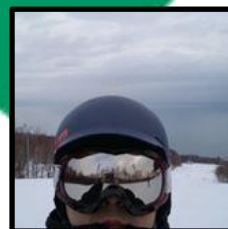
北海道の胆振地区が誇る
(架空の) テスト会社 (の想定) です。



↓だいひよー
みずのり

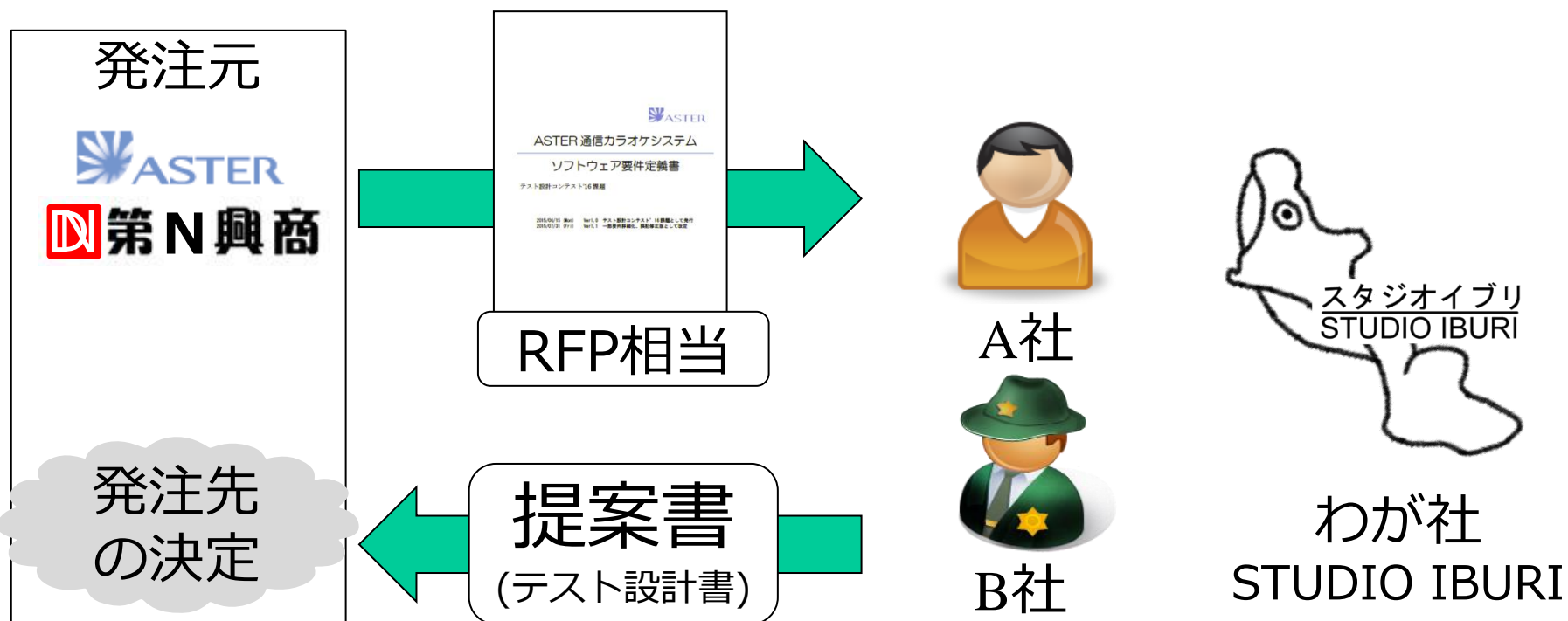


せーの



くつきー

本発表はASTER及び第N興商さんが
テスト業務の発注先を選定をするための提案書
という想定で内容紹介します。

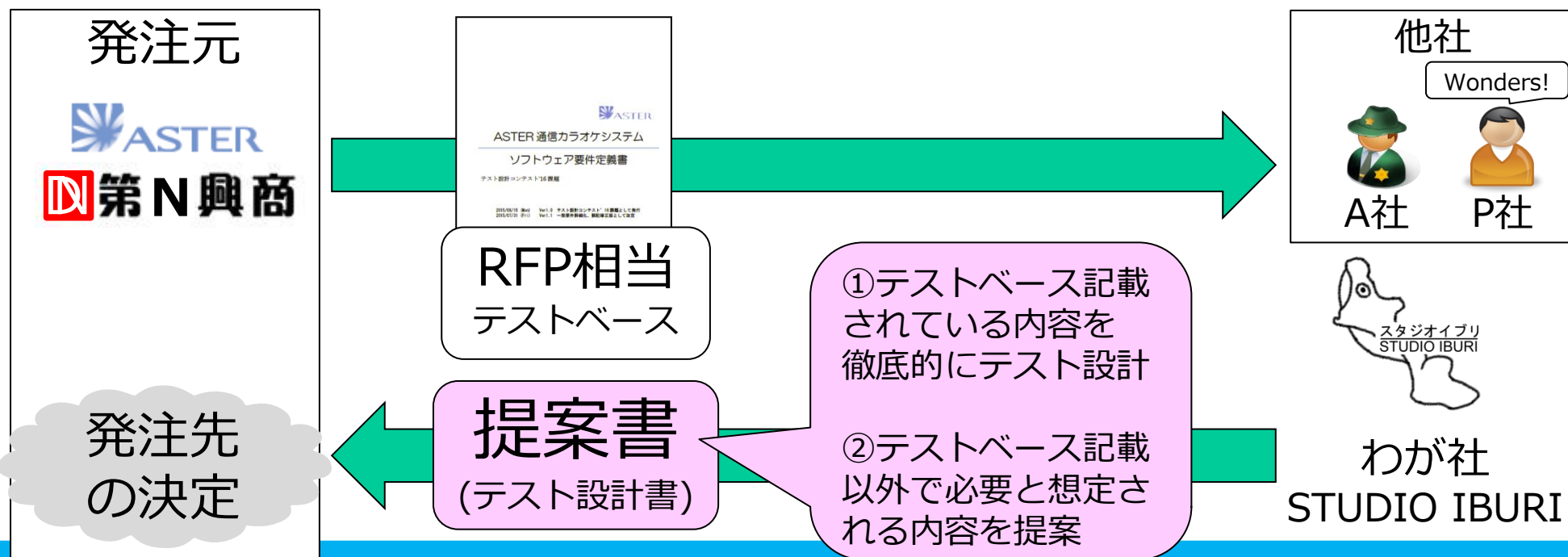


※注意事項

発注のためにテストの実力を見る、という想定であるため、

- ①テストベースを徹底的にテストする ことを優先します。※
- ②テストベース記載外の内容を「追加提案」という形式にします。

※ユーザをペルソナ分析などで想定してテストベース外の提案を多数行うことも可能ですが、マーケティングやプロダクト戦略が既に存在しており被る（余計なお世話となる）可能性が有るため、追加提案はテストベースに記載の「ねらい」から想定される最低限の内容としております。



ASTER/第N興商様向け テスト設計業務の 提案書



STUDIO IBURI

テスト上位設計の結果を3つの「ビュー」として表すことで、顧客との議論を行いやすい。

⇒スライドP34、P35、P41にそれぞれのビューを記載。

単純にテストベースからは出てこない内容のテストを「参照モデル」を活用することで導出。

⇒スライドP48～50で紹介。

ねらいを達成するために必要と考えた内容を提案ベースのテストとして追加。

⇒スライドP57～65で紹介。

顧客がテストをカスタマイズできる各種情報を用意。

⇒スライドP68～70で紹介。

①テストベースを
徹底的にテスト

②追加価値の提案
カスタマイズ可能

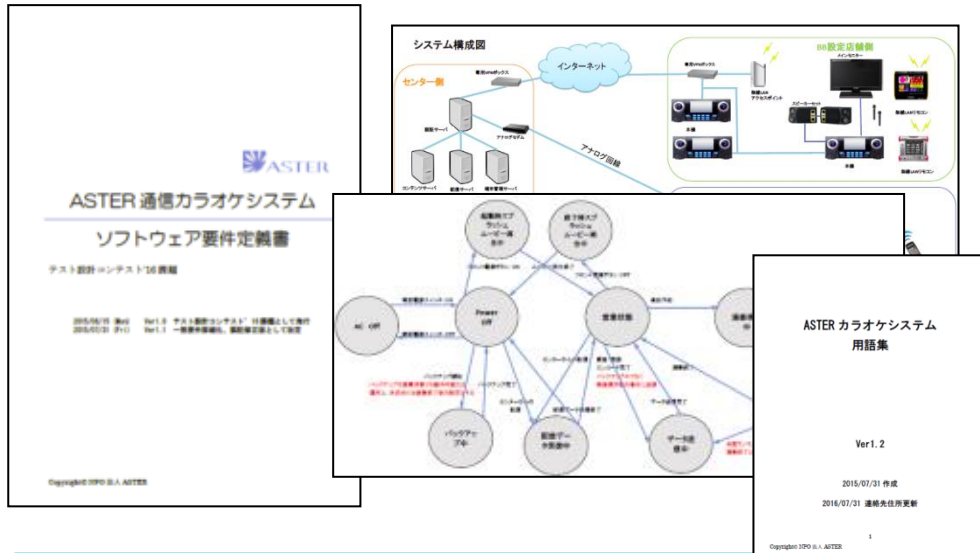
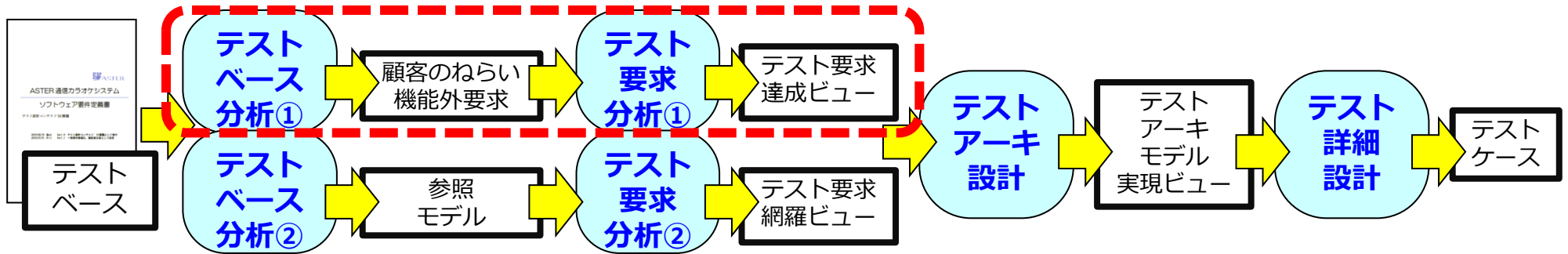
テストの技術力

その① : テストベースを徹底的にテスト



① テストベースを
徹底的にテスト

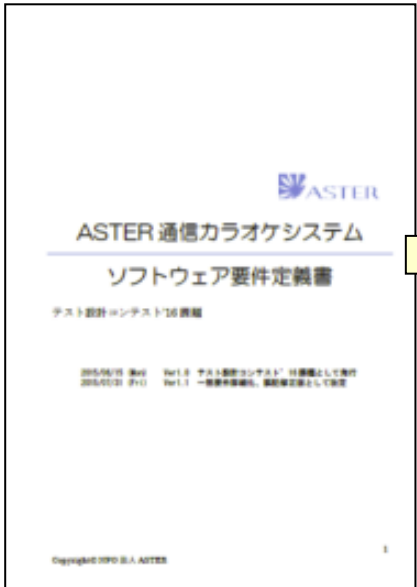
1. なぜ、テストが必要か？



御社のねらいや
達成すべきものを
を検討

※本来はインタビュー等で合意を取りながら進めます

1. テストベース記述から、テストの必要性を抽出



まとめております

顧客のねらい
機能外要求

【ねらい】

抜粋元	No	顧客	分類
ソフトウェア要件定義書(1)			
1-1	1	コン	背景
システム	2	通信	背景
開発の	3	高速	背景
背景-趣旨	4	新機	変更
	5	ストレージ容量の増加による高画質映像データ	変更
	6	生音データの追加	変更
	7	過去の楽曲データへの完全互換(互換性)	機能外要求
	8	現行の外部機器への対応(互換性)	機能外要求
1-2	1	現行機リプレイス、他社機との入れ替えを促すボックス、ナイト店への導入を前提	ねらい
システムの	2	快適でリッチな体験の提供	ねらい
目的	3	ターゲット向けのコンテンツを搭載	変更
	4	快適に使用できるスペックを持つ機器	変更
	5	現行インフラ増強、データ量増加、リアルタイム通信対応	変更
1-3	1	現行のインフラ継承	制約
システム	2	一部光回線での高速化	変更
全体像	3	センター側のHW更新と同線増強	変更
開発方針	4	今後利用可能なハードウェアの追加を考慮	ねらい
展望	5	楽曲データ、外部機器への完全互換性(互換性) ※1-1の7.8と同じ なので除外	機能外要求
	6	従来機のコンテンツも可能な限り搭載(互換性)	機能外要求
	7	OSをLinuxに変更	変更
	8	ハードウェアを刷新	変更
2-1	1	ハードウェアを刷新、OSをLinuxへ変更(互換性) ※1-3の7.8と同じ なので除外	変更
システム	2	著作権、肖像権保護	ねらい
開発	3	データ、ストレージの暗号化、通信トンネル化(セキュリティ)	変更
制約条件	4	HW国内法準拠 ※HWはテストスコープ外とする なので除外	機能外要求
2-2	1	サプライヤ系過去不具合:引き上げ処理	不具合
	2	サプライヤ系過去不具合:プログラム更新	不具合
ソフトウェア要件定義書(3章:機能要求) ※以下記載の番号は機能要求ID			
1-2-2-1	-	映像とのタイミング(リップシンク)	気がかり
1-3-1	-	歌いやすさ(テロップタイミング)	気がかり
1-4-1、1-4-2	-	演奏やり直し、演奏中止時のノイズの影響	気がかり
1-4-7、1-4-9	-	早送り早戻し、サビへジャンプ時のノイズの影響	気がかり
1-4-8、1-4-10	-	一時停止時のノイズの影響	気がかり
1-4-2	-	誤動作防止(二度押し)	気がかり
1-4-3	-		
1-4-4	-	曲間BGMへのキー、テンポコントロールの影響	不具合
1-7-2	-	ガイドメモリ有効、無効の即時反映	気がかり
3-1-1-2	-	NB使用時のオーバー通信費軽減	気がかり
ソフトウェア要件定義書(4章:機能外要求)			
4-1	-	現行の外部機器が全て動作する(互換性) 予約機器及び課金専用装置のプロトコル100%互換 ※1-1の8と同じのため除外	機能外要求
4-2	-	新採点エンジンが次世代機種に移植しやすい(移植性) ソフトのモジュール化、ドキュメントの準備	機能外要求
4-3	-	前回のフラグシップモデルを全ての性能で上回ること 配信、圧縮データ展開時の負荷の高い場合でも、新探 ルタイム表示の遅延や描画崩れの乱れが無い事(性能)	
4-4	-	現状のシステム、通信の仕組みに変更がなくてはなら ない ※1-3の1と同じのため除外	
4-5	-	現行機種が持つセキュリティに関わる仕組みを全て継承 同等以上のセキュリティを確保(セキュリティ)	機能外要求

過去不具合

気がかり

【機能外要求】

1. テストベース記述から、テストの必要性を抽出



まとめております

顧客のねらい
機能外要求

【ねらい】

- 【ねらい】
- ・ 現行機リプレイス、他社機との入れ替えを促す
 - ・ 快適でリッチな経験の提供
 - ・ 今後利用可能なハードウェアの追加を考慮

抜粋元	No.	顧客	分類
ソフトウェア要件定義書(1)			
1-1	1	コン	背景
システム	2	通信	背景
開発の	3	高速	背景
背景-趣旨	4	新機	変更
	5	ストレージ容量の増加による高画質映像データ	変更
	6	生音データの追加	変更
	7	過去の楽曲データへの完全互換(互換性)	機能外要求
	8	現行の外部機器への対応(互換性)	機能外要求
1-2	1	現行機リプレイス、他社機との入れ替えを促す	ねらい
システム	2	ボックス、ナイト店への導入を前提	ねらい
目的	3	快適でリッチな経験の提供	変更
	4	ターゲット向けのコンテンツを搭載	変更
	5	快適に使用できるスペックを持つ機器	変更
	6	現行インフラ増強、データ量増加、リアルタイム通信対応	変更
1-3	1	現行のインフラ継承	制約
システム	2	一部光回線での高速化	変更
全体像	3	センター側のHW更新と同線増強	変更
開発方針	4	今後利用可能なハードウェアの追加を考慮	ねらい
展望	5	楽曲データ、外部機器への完全互換性(互換性)	機能外要求
	6	※1-1の7.8と同じ なので除外	機能外要求
	7	従来機のコンテンツも可能な限り搭載(互換性)	機能外要求
	8	OSをLinuxに変更	変更
	9	ハードウェアを刷新	変更
2-1	1	ハードウェアを刷新、OSをLinuxへ変更(互換性)	変更
システム	2	※1-3の7.8と同じ なので除外	変更
変更の	3	著作権、肖像権保護	ねらい
要件	4	データ、ストレージの暗号化、通信トンネル化(セキュリティ)	変更
	5	HW国内法準拠	機能外要求
	6	※HWはテストスコープ外とする なので除外	機能外要求
	7	サプライヤ系過去不具合:引き上げ処理	不具合
	8	サプライヤ系過去不具合:プログラム更新	不具合
ア要件定義書(3章:機能要求) ※以下記載の番号は機能要求ID			
	-	映像とのタイミング(リップシンク)	気がかり
	-	歌いやすさ(トップタイミング)	気がかり
-4-2	-	演奏やり直し、演奏中止時のノイズの影響	気がかり
-4-9	-	早送り早戻し、サビへジャンプ時のノイズの影響	気がかり
-4-10	-	一時停止時のノイズの影響	気がかり
	-	誤動作防止(二度押し)	気がかり
	-	曲間BGMへのキー、テンポコントロールの影響	不具合
	-	ガイドメモリ有効、無効の即時反映	気がかり
	-	NB使用時のオーナー通信費軽減	気がかり
ア要件定義書(4章:機能外要求)			
	-	現行の外部機器が全て動作する(互換性)	機能外要求
	-	予約機器及び課金専用装置のプロトコル100%互換	機能外要求
	-	※1-1の8と同じのため除外	機能外要求
	-	新採点エンジンが次世代機種に移植しやすい(移植性)	機能外要求
	-	ソフトのモジュール化、ドキュメントの準備	機能外要求
	-	前回のフラグシップモデルを全ての性能で上回ること	機能外要求
	-	配信、圧縮データ展開時の負荷の高い場合でも、新採	機能外要求
	-	ルタイム表示の遅延や描画崩れの乱れが無い事(性能)	機能外要求
	-	現状のシステム、通信の仕組みに変更があっても	機能外要求
	-	※1-3の1と同じのため除外	機能外要求
要件			
	-	現行機種が持つセキュリティに関わる仕組みを全て継承	機能外要求
セキュリティ目標			
	-	同等以上のセキュリティを確保(セキュリティ)	機能外要求

過去不具合

気がかり

【機能外要求】

1. テストベース記述から、テストの必要性を抽出

まとめております



顧客のねらい
機能外要求

【ねらい】

抜粋元	No	顧客	分類
ソフトウェア要件定義書(1)			
1-1	1	コンテンツの追加	背景
システム	2	通信速度の向上	背景
開発の	3	高速動作の実現	背景
背景-趣旨	4	新機能の追加	変更
	5	ストレージ容量の増加による高画質映像データ	変更
	6	生音データの追加	変更
	7	過去の楽曲データへの完全互換(互換性)	機能外要求
	8	現在の外部機器への対応(互換性)	機能外要求
1-2	1	現行機リプレイス、他社機との入れ替えを促す	ねらい
システム	2	ボックス、ナイト店への導入を前提	ねらい
目的	3	快適でリッチな体験の提供	変更
	4	ターゲット向けのコンテンツを搭載	変更
	5	快適に使用できるスペックを持つ機器	変更
	6	現行インフラ増強、データ量増加、リアルタイム通信対応	変更
1-3	1	現行のインフラ継承	制約
システム	2	一部光回線での高速化	変更
全体像	3	センター側のHW更新と同線増強	変更
開発方針	4	今後利用可能なハードウェアの追加を考慮	ねらい
展望	5	楽曲データ、外部機器への完全互換性(互換性)	機能外要求
	6	※1-1の7.8と同じ なので除外	機能外要求
	7	従来機のコンテンツも可能な限り搭載(互換性)	機能外要求
	8	OSをLinuxに変更	変更
	9	ハードウェアを刷新	変更
2-1	1	ハードウェアを刷新、OSをLinuxへ変更(互換性)	変更
システム	2	※1-3の7.8と同じ なので除外	変更
	3	著作権、肖像権保護	ねらい
	4	データ、ストレージの暗号化、通信トンネル化(セキュリティ)	変更
	5	HW国内法準拠	機能外要求
	6	※HWはテストスコープ外とする なので除外	機能外要求
	7	プレイヤ系過去不具合:引き上げ処理	不具合
	8	プレイヤ系過去不具合:プログラム更新	不具合
	9	3章:機能要求) ※以下記載の番号は機能要求ID	
	10	歌とのタイミング(リップシンク)	気がかり
	11	いやさ(テロップタイミング)	気がかり
	12	歌やり直し、演奏中止時のノイズの影響	気がかり
	13	送り早戻し、サビへジャンプ時のノイズの影響	気がかり
	14	待停止時のノイズの影響	気がかり
	15	動作防止(二度押し)	気がかり
	16	前BGMへのキー、テンポコントロールの影響	不具合
	17	リモコン有効、無効の即時反映	気がかり
	18	使用時のオーナー通信費軽減	気がかり
	19	4章:機能外要求	
	20	外部機器が全て動作する(互換性)	機能外要求
	21	外部機器及び課金専用装置のプロトコル100%互換	機能外要求
	22	※1-1の8と同じのため除外	機能外要求
	23	単点エンジンが次世代機種に移植しやすい(移植性)	機能外要求
	24	各モジュールのモジュール化、ドキュメントの準備	機能外要求
	25	目的のフラグシップモデルを全ての性能で上回ることを	
	26	言、圧縮データ展開時の負荷の高い場合でも、新探	
	27	タイム表示の遅延や描画崩れの乱れが無い事(性能	
	28	次のシステム、通信の仕組みに変更がなくてはなら	
	29	※1-3の1と同じのため除外	
	30	外部機器を持つセキュリティに関わる仕組みを全て継承	機能外要求
	31	※1-3の1と同じのため除外	機能外要求
	32	外部機器を持つセキュリティに関する仕組みを全て継承	機能外要求
	33	※1-3の1と同じのため除外	機能外要求

過去不具合

気がかり

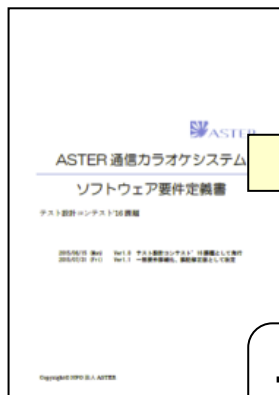
(一部)

- ・映像とタイミング (リップシンク)
- ・歌いやすさ (テロップタイミング)
- ・演奏やり直し、演奏中止時のノイズの影響
- ・...

気がかり

【機能外要求】

1. ねらいから「達成するもの」を整理



上位の「ねらい」からの
関連性を整理しています。

機能名	No.	顧客の欲しい 上位要求(機能以外)	分類
1-1-1	1	コンテンツのHD化要望の裏取り	背景
1-1-2	2	通信環境の飛躍的向上	背景
1-1-3	3	低遅延環境での快適なリスニング	背景
1-1-4	4	新採点システムの導入	変更
1-1-5	5	ストリーミング環境による高音質映像データ	変更
1-1-6	6	高音質データの完全互換(互換性)	機能外要求
1-1-7	7	高音質データの完全互換(互換性)	機能外要求
1-2	8	現行機リプレイス、他社機との入れ替えを促す	ねらい
1-2-1	8-1	ボツマス、ナレ音への導入を前提	ねらい
1-2-2	8-2	音楽リソース管理の操作性	変更
1-2-3	8-3	ターゲット向けのコンテンツを掲載	変更
1-2-4	8-4	設置に際してはスベールを考慮	変更
1-2-5	8-5	現行のインフラ環境	制約
1-2-6	8-6	新採点システムの導入	変更
1-2-7	8-7	センター側の設置と接続確認	変更
1-2-8	8-8	将来利用可能なハードウェアの汎用性を考慮	ねらい
1-2-9	8-9	将来利用可能なハードウェアの存在互換性(互換性)	機能外要求
1-2-10	8-10	OS1の互換性(互換性)	機能外要求
1-2-11	8-11	OS2のLinux化	変更
1-2-12	8-12	ハードウェア更新	変更
1-2-13	8-13	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-14	8-14	OS1の互換性(互換性)	変更
1-2-15	8-15	OS2の互換性(互換性)	変更
1-2-16	8-16	映像、音声品質	ねらい
1-2-17	8-17	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-18	8-18	OS1の互換性(互換性)	変更
1-2-19	8-19	OS2の互換性(互換性)	変更
1-2-20	8-20	映像、音声品質	ねらい
1-2-21	8-21	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-22	8-22	OS1の互換性(互換性)	変更
1-2-23	8-23	OS2の互換性(互換性)	変更
1-2-24	8-24	映像、音声品質	ねらい
1-2-25	8-25	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-26	8-26	OS1の互換性(互換性)	変更
1-2-27	8-27	OS2の互換性(互換性)	変更
1-2-28	8-28	映像、音声品質	ねらい
1-2-29	8-29	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-30	8-30	OS1の互換性(互換性)	変更
1-2-31	8-31	OS2の互換性(互換性)	変更
1-2-32	8-32	映像、音声品質	ねらい
1-2-33	8-33	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-34	8-34	OS1の互換性(互換性)	変更
1-2-35	8-35	OS2の互換性(互換性)	変更
1-2-36	8-36	映像、音声品質	ねらい
1-2-37	8-37	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-38	8-38	OS1の互換性(互換性)	変更
1-2-39	8-39	OS2の互換性(互換性)	変更
1-2-40	8-40	映像、音声品質	ねらい
1-2-41	8-41	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-42	8-42	OS1の互換性(互換性)	変更
1-2-43	8-43	OS2の互換性(互換性)	変更
1-2-44	8-44	映像、音声品質	ねらい
1-2-45	8-45	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-46	8-46	OS1の互換性(互換性)	変更
1-2-47	8-47	OS2の互換性(互換性)	変更
1-2-48	8-48	映像、音声品質	ねらい
1-2-49	8-49	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-50	8-50	OS1の互換性(互換性)	変更
1-2-51	8-51	OS2の互換性(互換性)	変更
1-2-52	8-52	映像、音声品質	ねらい
1-2-53	8-53	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-54	8-54	OS1の互換性(互換性)	変更
1-2-55	8-55	OS2の互換性(互換性)	変更
1-2-56	8-56	映像、音声品質	ねらい
1-2-57	8-57	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-58	8-58	OS1の互換性(互換性)	変更
1-2-59	8-59	OS2の互換性(互換性)	変更
1-2-60	8-60	映像、音声品質	ねらい
1-2-61	8-61	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-62	8-62	OS1の互換性(互換性)	変更
1-2-63	8-63	OS2の互換性(互換性)	変更
1-2-64	8-64	映像、音声品質	ねらい
1-2-65	8-65	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-66	8-66	OS1の互換性(互換性)	変更
1-2-67	8-67	OS2の互換性(互換性)	変更
1-2-68	8-68	映像、音声品質	ねらい
1-2-69	8-69	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-70	8-70	OS1の互換性(互換性)	変更
1-2-71	8-71	OS2の互換性(互換性)	変更
1-2-72	8-72	映像、音声品質	ねらい
1-2-73	8-73	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-74	8-74	OS1の互換性(互換性)	変更
1-2-75	8-75	OS2の互換性(互換性)	変更
1-2-76	8-76	映像、音声品質	ねらい
1-2-77	8-77	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-78	8-78	OS1の互換性(互換性)	変更
1-2-79	8-79	OS2の互換性(互換性)	変更
1-2-80	8-80	映像、音声品質	ねらい
1-2-81	8-81	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-82	8-82	OS1の互換性(互換性)	変更
1-2-83	8-83	OS2の互換性(互換性)	変更
1-2-84	8-84	映像、音声品質	ねらい
1-2-85	8-85	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-86	8-86	OS1の互換性(互換性)	変更
1-2-87	8-87	OS2の互換性(互換性)	変更
1-2-88	8-88	映像、音声品質	ねらい
1-2-89	8-89	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-90	8-90	OS1の互換性(互換性)	変更
1-2-91	8-91	OS2の互換性(互換性)	変更
1-2-92	8-92	映像、音声品質	ねらい
1-2-93	8-93	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-94	8-94	OS1の互換性(互換性)	変更
1-2-95	8-95	OS2の互換性(互換性)	変更
1-2-96	8-96	映像、音声品質	ねらい
1-2-97	8-97	ハードウェア更新、OSをLinuxへ変更(互換性)	変更
1-2-98	8-98	OS1の互換性(互換性)	変更
1-2-99	8-99	OS2の互換性(互換性)	変更
1-2-100	8-100	映像、音声品質	ねらい

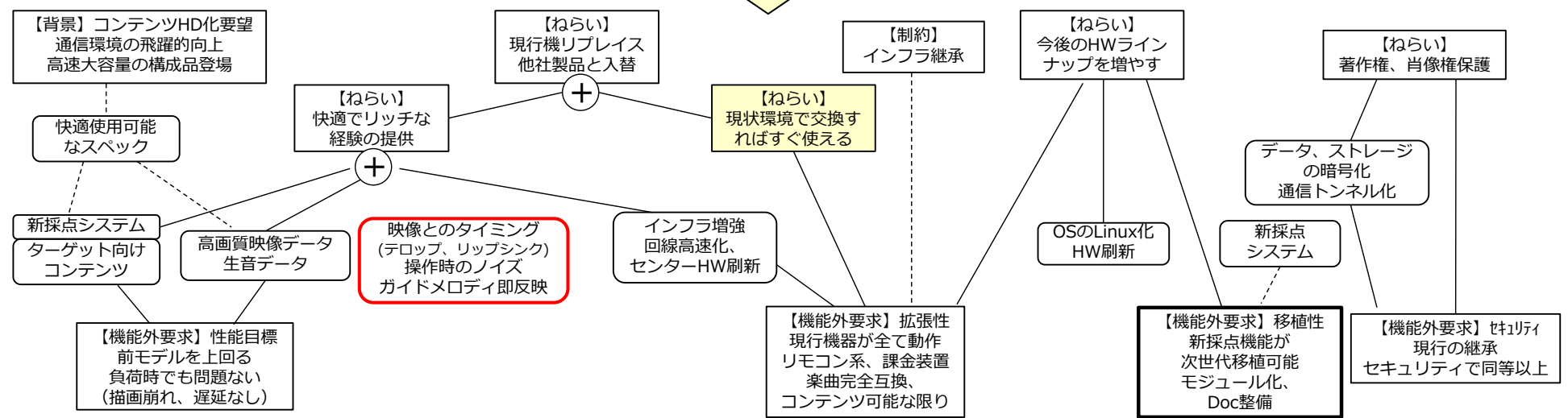
【ねらい】

過去不具合

気がかり

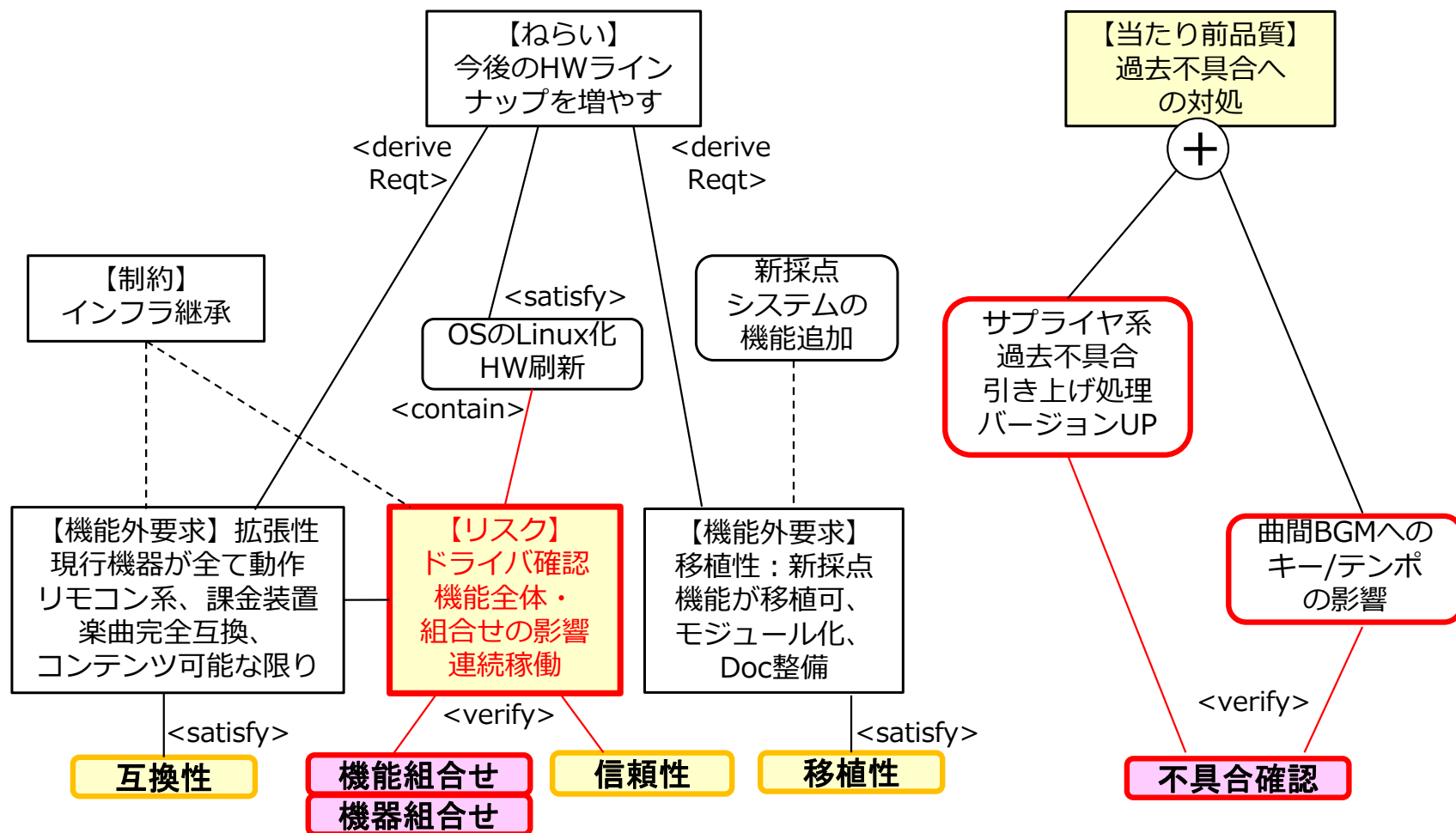
【機能外要求】

顧客のねらい
機能外要求

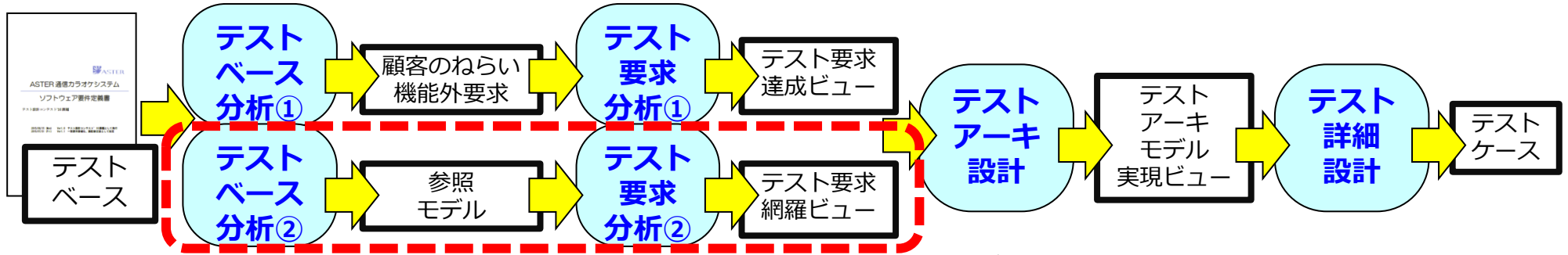


1. ねらいから「達成するもの」を整理

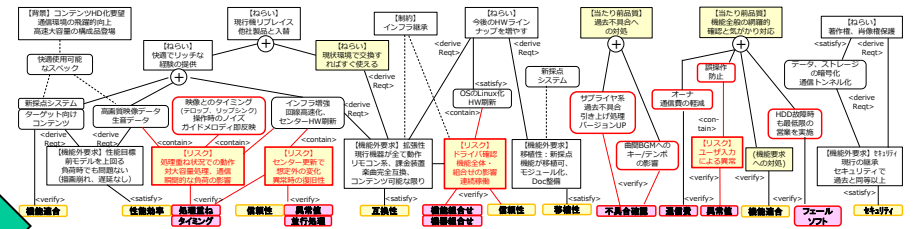
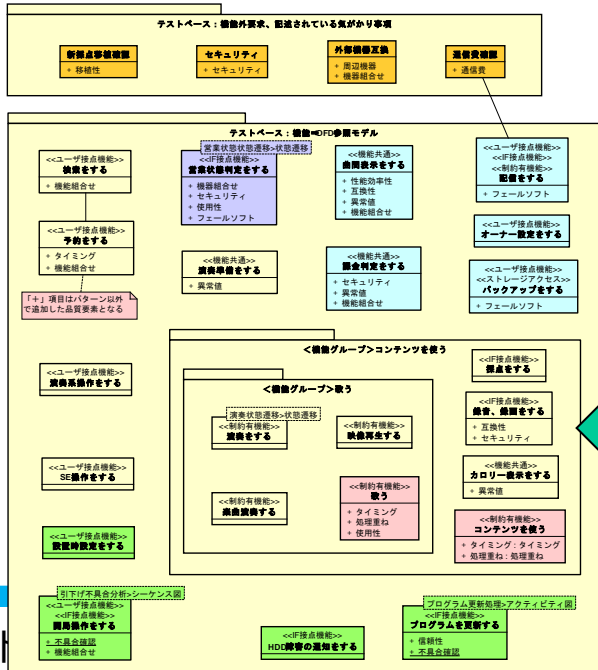
拡大：今後のHWラインナップを増やす「ねらい」



2. 何をテストすべきか？



2つのビューで
網羅的な検討をします。

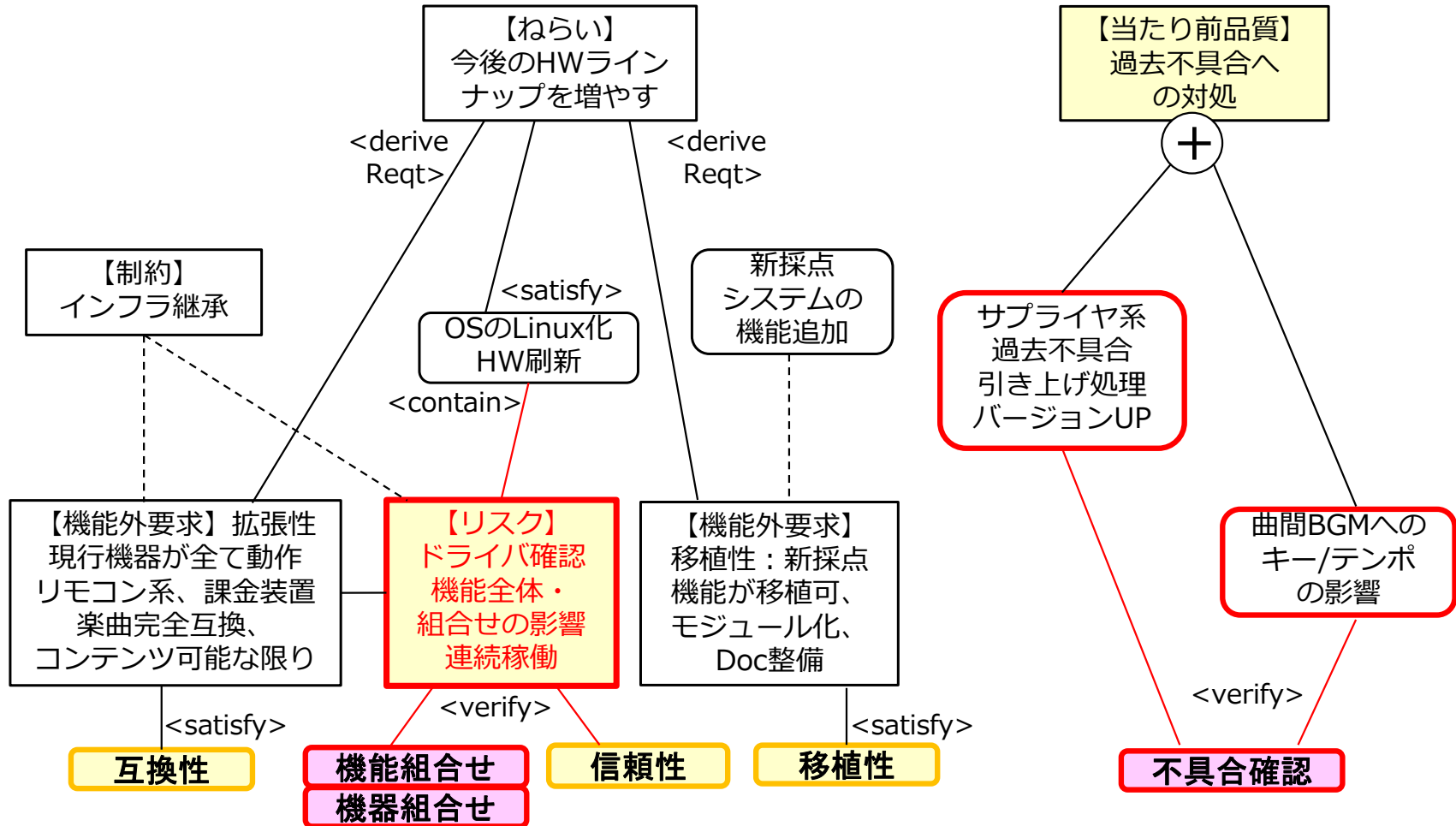


KQF (Key Quality Factor) × 各種テスト

機能適合 信頼性	性能効率	信頼性 処理遅延 タイムリ	異常値	信頼性 並行処理	互換性 機能 組合せ	機能 組合せ	移植性	セキュリティ	不具合確認	フェール ソフト	運賃費
テストベース ユーザ接点 新機軸等価値観	ユーザ接点 機能	ユーザ接点 機能	ユーザ接点 機能	ユーザ接点 機能	ユーザ接点 機能	ユーザ接点 機能	ユーザ接点 機能	ユーザ接点 機能	ユーザ接点 機能	ユーザ接点 機能	ユーザ接点 機能

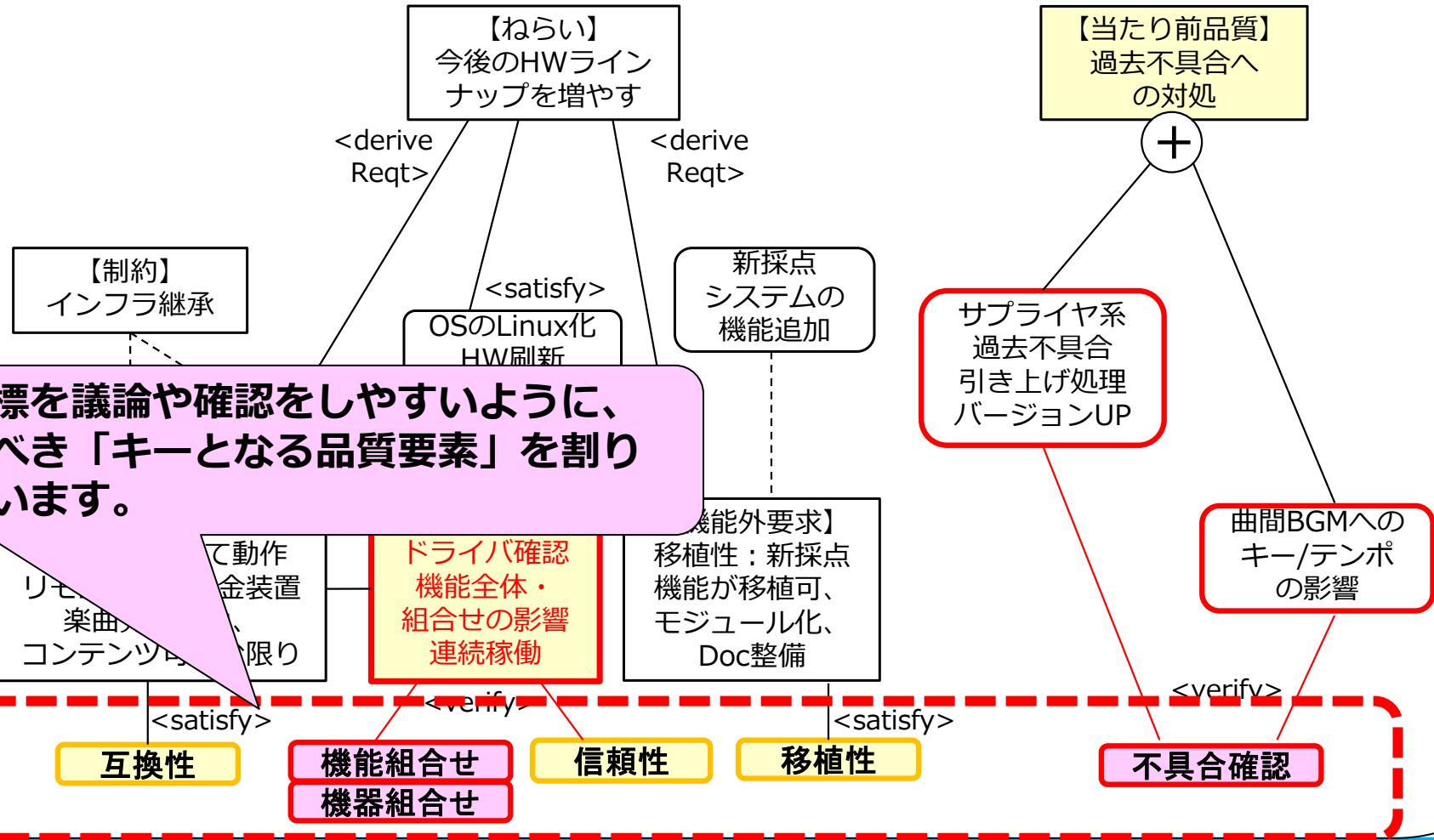
2. 何をテストすべきか？

拡大：今後のHWラインナップを増やす「ねらい」



2. 何をテストすべきか？

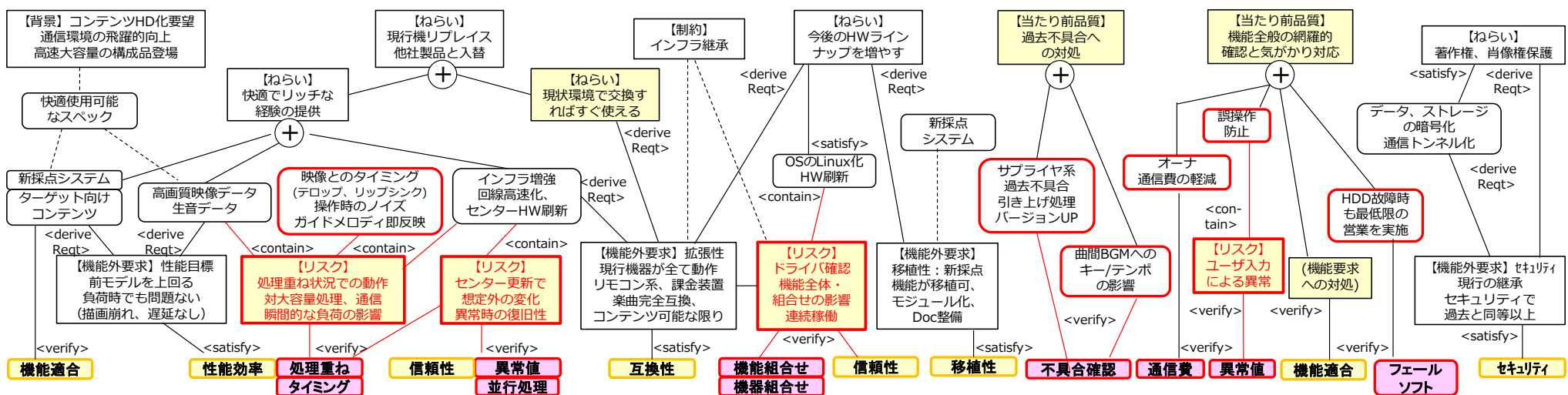
拡大：今後のHWラインナップを増やす「ねらい」



達成目標を議論や確認をしやすいように、達成すべき「キーとなる品質要素」を割り当てています。

2. 何をテストすべきか？テスト達成マップ全体

テスト達成マップ



※想定されるリスクや、「ねらい」から考えると除外されることの多い「当たりの品質」の要素を追加することで、テストで達成すべき全体像を整理しております。

注：成果物1には、追加提案を加えたテスト達成マップを載せております。

2. 何をテストすべきか？：キーとなる品質要素（参考）



分類/KQF項目	説明
品質特性 (SQuaRE: ISO/IEC25010ベース)	
機能適合	各機能に対する入出力のロジックや正確さを確認する。
性能効率	主に負荷時処理や処理速度、データ量などに関わる部分についての確認及び評価を行う。
信頼性	処理量が多い場合、特殊な操作、連続動作や復旧性に問題発生が無いかを確認する。
使用性	操作に対する分かりやすさや見栄えの良さについて確認を行う。
互換性	現行機器及び楽曲・コンテンツに互換性が有り使用できることを確認する。
移植性	新採点機能に対する移植性についての確認を行う。
セキュリティ	暗号化や通信のトンネル化処理の正しさ、不正利用への対応を確認する。
欠陥検出戦略	
異常値	異常値や異常な操作に対するシステムの影響を確認する。
並行処理	複数処理が並行して発生する場合における影響を確認する。
機能組合せ	機能や入力パラメータを組合わせ、動作に異常がないことを確認する。
機器組合せ	使用が予想される周辺機器を組合わせ、動作に異常がないことを確認する。
処理重ね	一部カタマリ特化の観点。音声や映像などの複数の処理を重ねて実施される場合における機能や性能面への影響を確認する。
タイミング	一部カタマリ特化の観点。複数の処理が同時に動いている処理やシビアなタイミングが要求される処理に対して確認を行う。
不具合確認	一部カタマリ特化の観点。過去発生した不具合として提示されている箇所に対して、不具合の可能性を検討し、再現しないことを確認する。
通信費	ナローバンドを用いるナイト店に対して、通信費をおさえることが出来ていること、必要最小限の通信が行われることを確認する。
フェールソフト	主にメインストレージ故障や片肺運転時に最低限の営業ができることを確認する。

機能適合

性能効率

信頼性

使用性

互換性

移植性

セキュリティ

異常値

並行処理

機能組合せ

機器組合せ

処理重ね

タイミング

不具合確認

通信費

フェールソフト

※参考：ISO/IEC25010における品質特性用語を用いている理由

テストベース「ASTERカラオケシステムソフトウェア要件定義書」にて機能外要求を中心に品質特性と一致する用語を使用していることが多いことが理由の1つである。

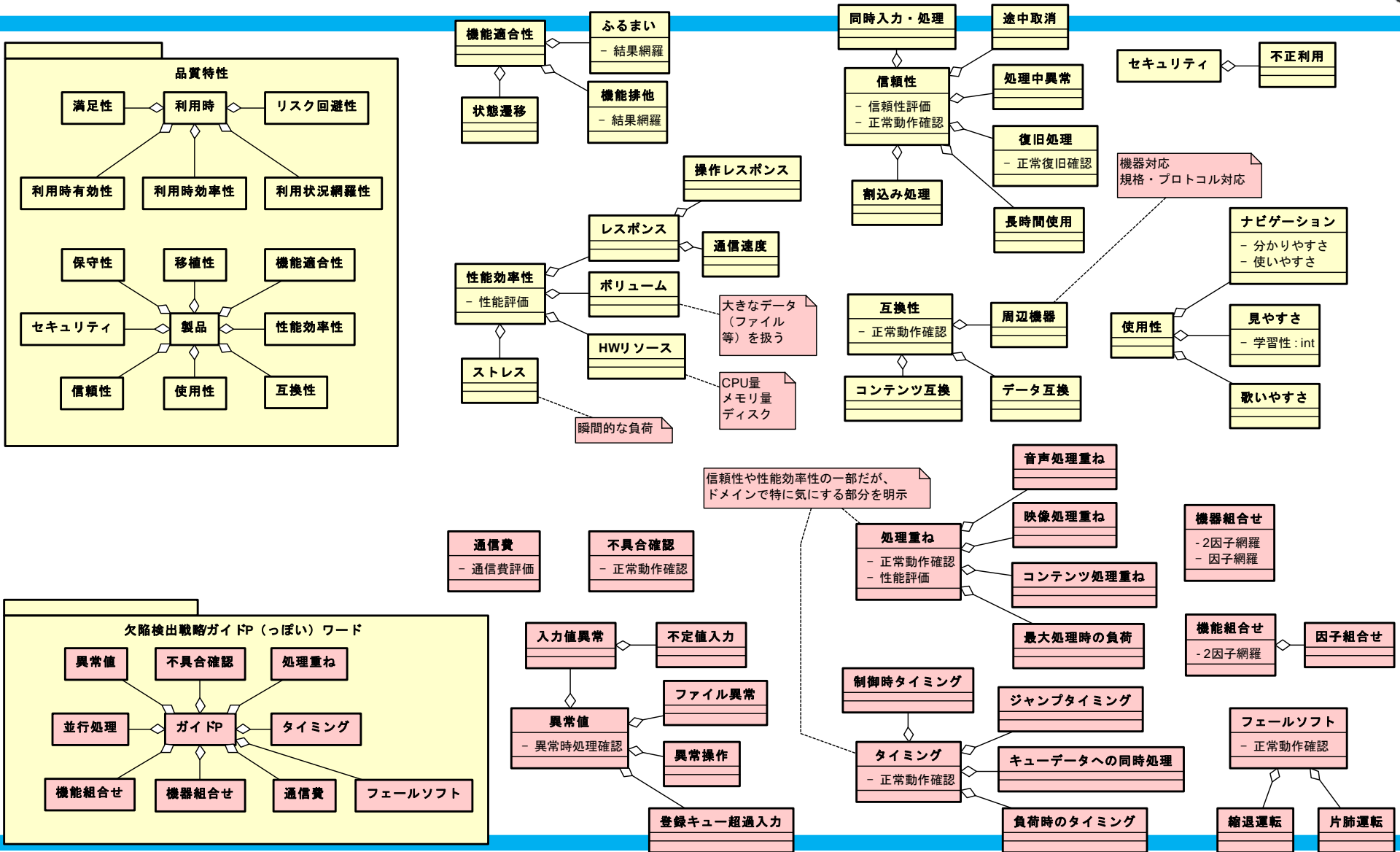
また、品質特性の持つMECEな特性を用いて抜けが無いことを確認しやすくするためである。

他にも、今回のような事前に顧客側と直接議論・交渉できない状況では一般化された用語の方が良いと判断している。

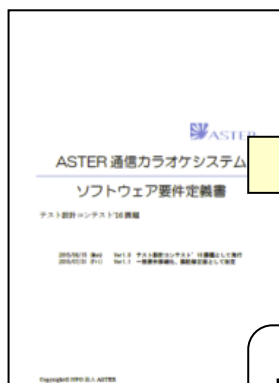
使用する用語や分類名はASTER/第N興商とインタビュー・議論することにより、

ドメインの関係者がより分かりやすい言葉に置き換えることも可能である。

2. 何をテストすべきか? : テストカテゴリとの階層構造



2. 再掲：ねらいから「達成するもの」を整理



上位の「ねらい」からの
関連性を整理しています。

品名	No.	顧客の欲しい 上位要求(機能以外)	分類
ASTER	1-1	コンテンツのHD化要望の裏取り	背景
システム	2	通信環境の飛躍的向上	背景
機種の	3	従来モデルの機種のリリース	背景
世界・地域	4	新採点システムの導入	変更
	5	ストレーン(録音の増加)による高音質映像データ	変更
	6	高音質データの増量	変更
	7	高音質の楽曲データの完全互換(互換性)	機能外要求
	8	機種の互換性への対応(互換性)	機能外要求
1-2	1	機種の互換性(リプレイス、他社機との入れ替えを促す)	ねらい
システム	2	ボタンス、ナビ等への導入を前提	ねらい
目的	3	音楽ソフトの複製の防止	変更
	4	ターゲット向けのコンテンツを掲載	変更
	5	設置に際してはスベールを考慮	変更
	6	機種の互換性(ユーザー追加、リアルタイム通信対応)	変更
1-3	1	機種の互換性	制約
システム	2	新採点システムの導入	変更
8年後	3	センター側の仕様変更と接続確認	変更
開発方針	4	将来利用可能なハードウェアの汎用性を考慮	ねらい
機種の	5	機種の互換性(複製防止の存在互換性(互換性))	機能外要求
	6	機種の互換性(互換性)	機能外要求
	7	OSのLinux化	変更
	8	ハードウェアの更新	変更
システム	9	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	10	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	11	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	12	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	13	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	14	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	15	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	16	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	17	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	18	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	19	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	20	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	21	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	22	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	23	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	24	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	25	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	26	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	27	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	28	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	29	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	30	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	31	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	32	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	33	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	34	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	35	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	36	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	37	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	38	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	39	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	40	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	41	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	42	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	43	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	44	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	45	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	46	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	47	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	48	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	49	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	50	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	51	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	52	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	53	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	54	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	55	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	56	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	57	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	58	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	59	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	60	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	61	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	62	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	63	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	64	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	65	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	66	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	67	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	68	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	69	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	70	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	71	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	72	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	73	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	74	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	75	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	76	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	77	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	78	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	79	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	80	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	81	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	82	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	83	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	84	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	85	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	86	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	87	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	88	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	89	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	90	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	91	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	92	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	93	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	94	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	95	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	96	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	97	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	98	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	99	ハードウェアの更新、OSのLinux化(互換性)	変更
機種の	100	ハードウェアの更新、OSのLinux化(互換性)	変更

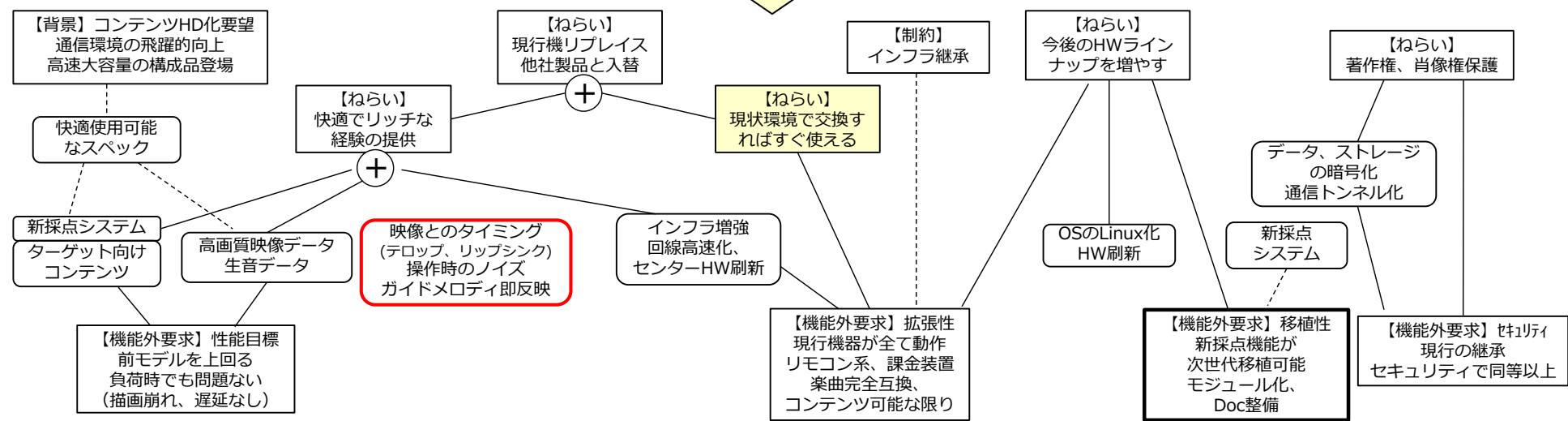
【ねらい】

過去不具合

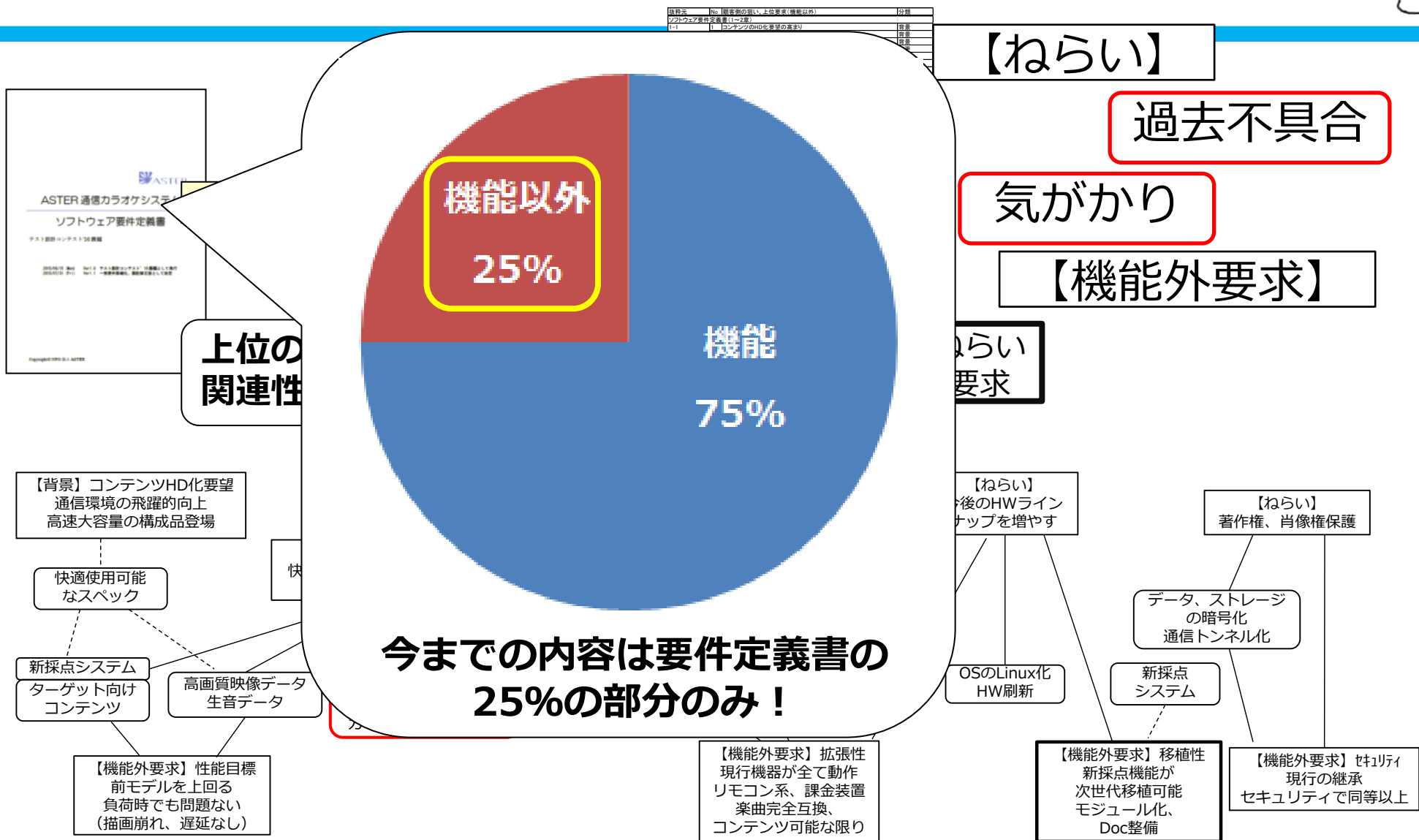
気がかり

【機能外要求】

顧客のねらい
機能外要求



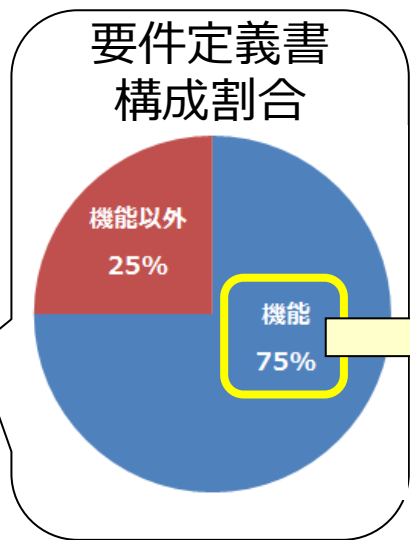
2. 再掲：ねらいから「達成するもの」を整理



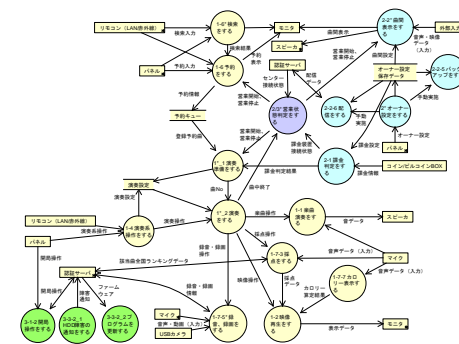
2. 何をテストすべきか？ 参照モデルへ

- 目的1：テストベース理解のため
- 目的2：機能やふるまいの関連明確化のため

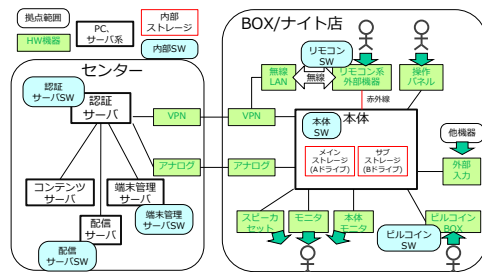
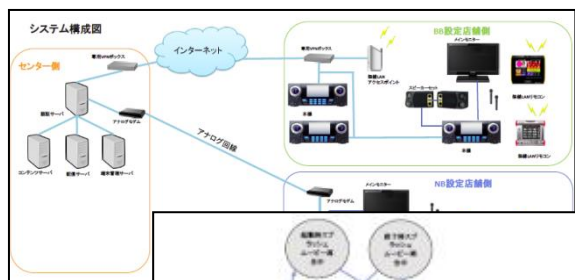
「参照モデル」で整理



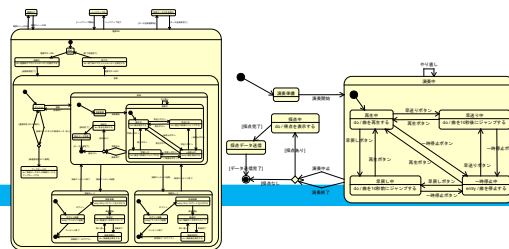
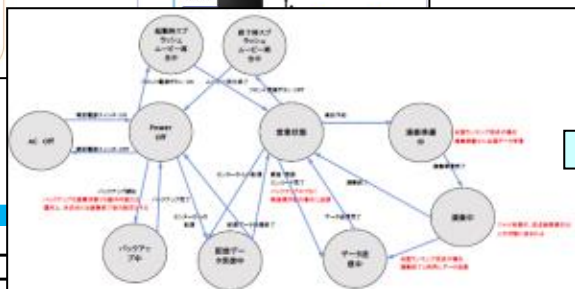
※ページ数より
当社比



【機能】
DFD

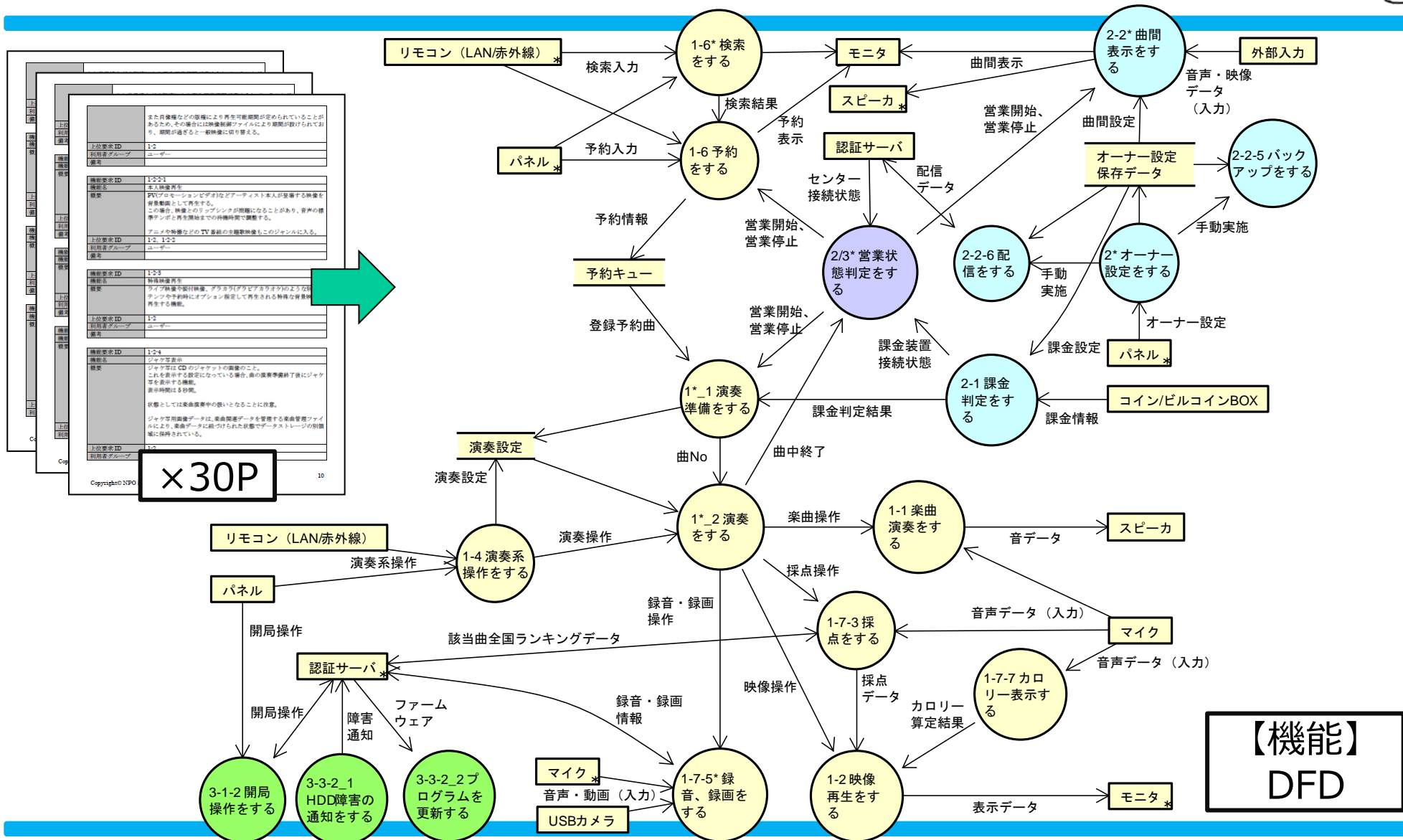


【構造】
システム
構造

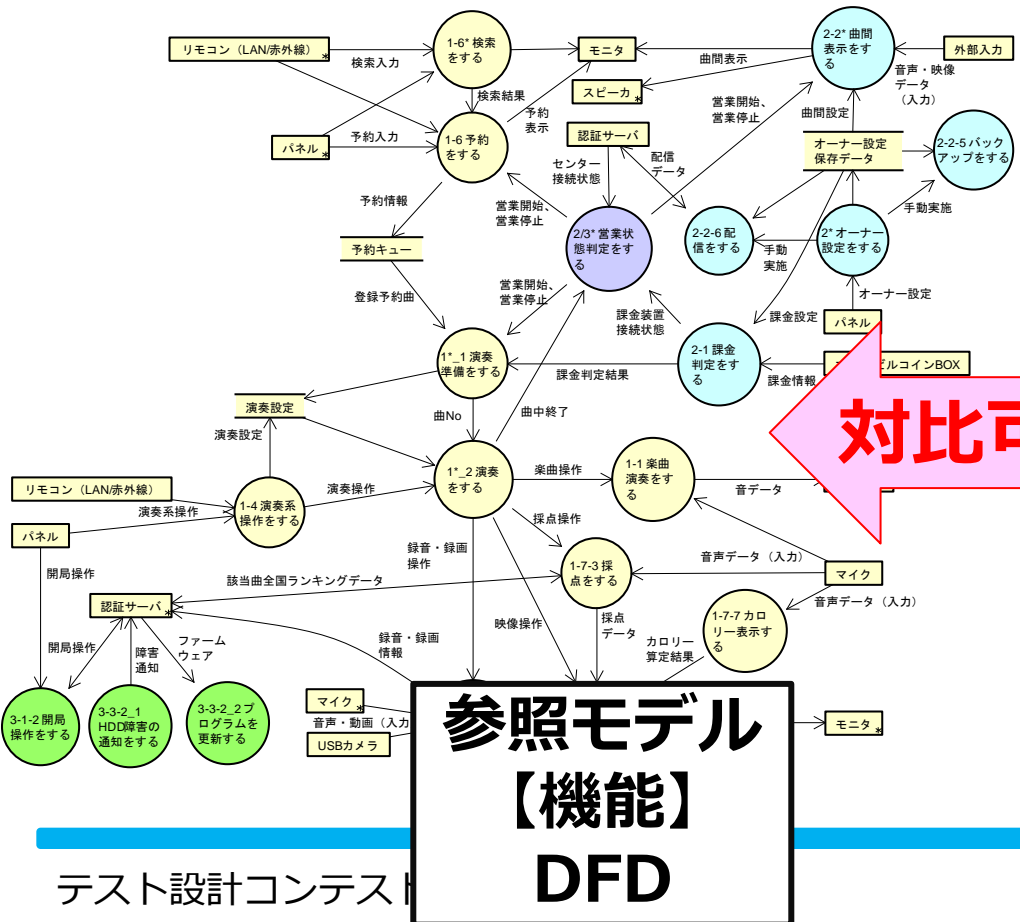


【振舞】
状態
遷移図

2. 参照モデルの例 : DFD (Data Flow Diagram)

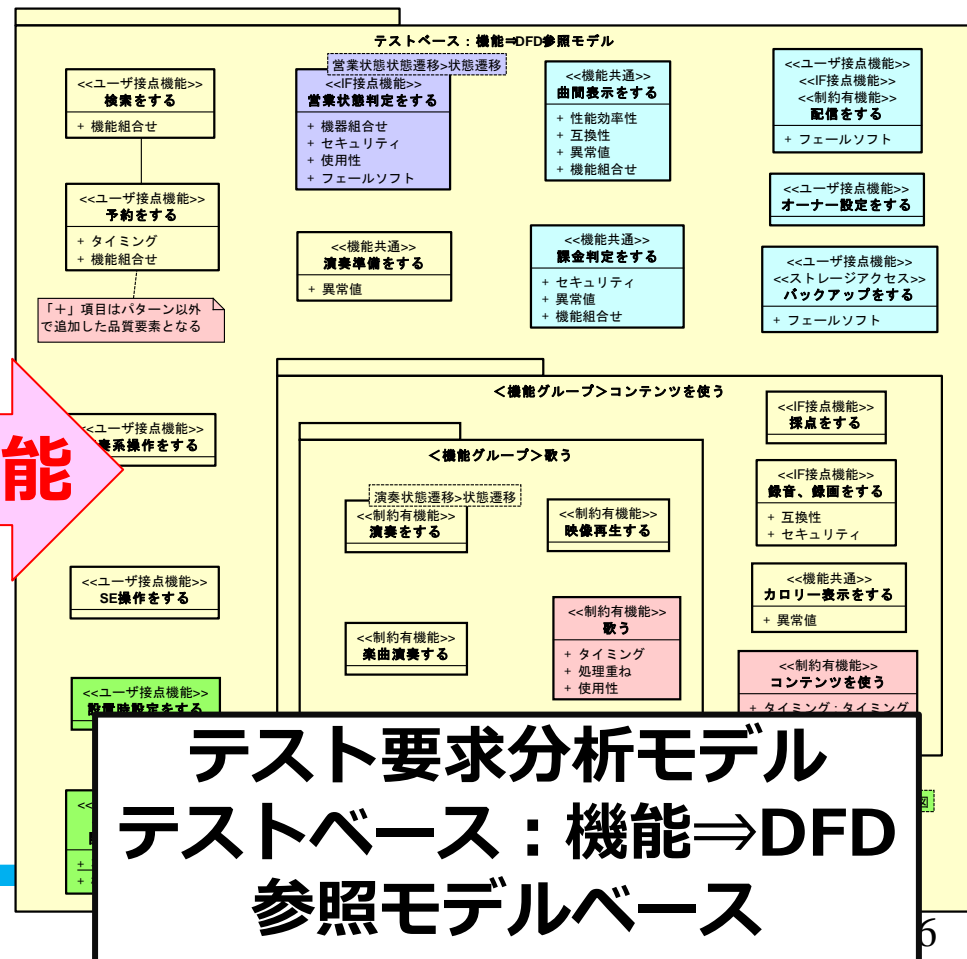


2. DFDと対応するテスト要求：機能⇒DFD参照モデルベース



**参照モデル
【機能】
DFD**

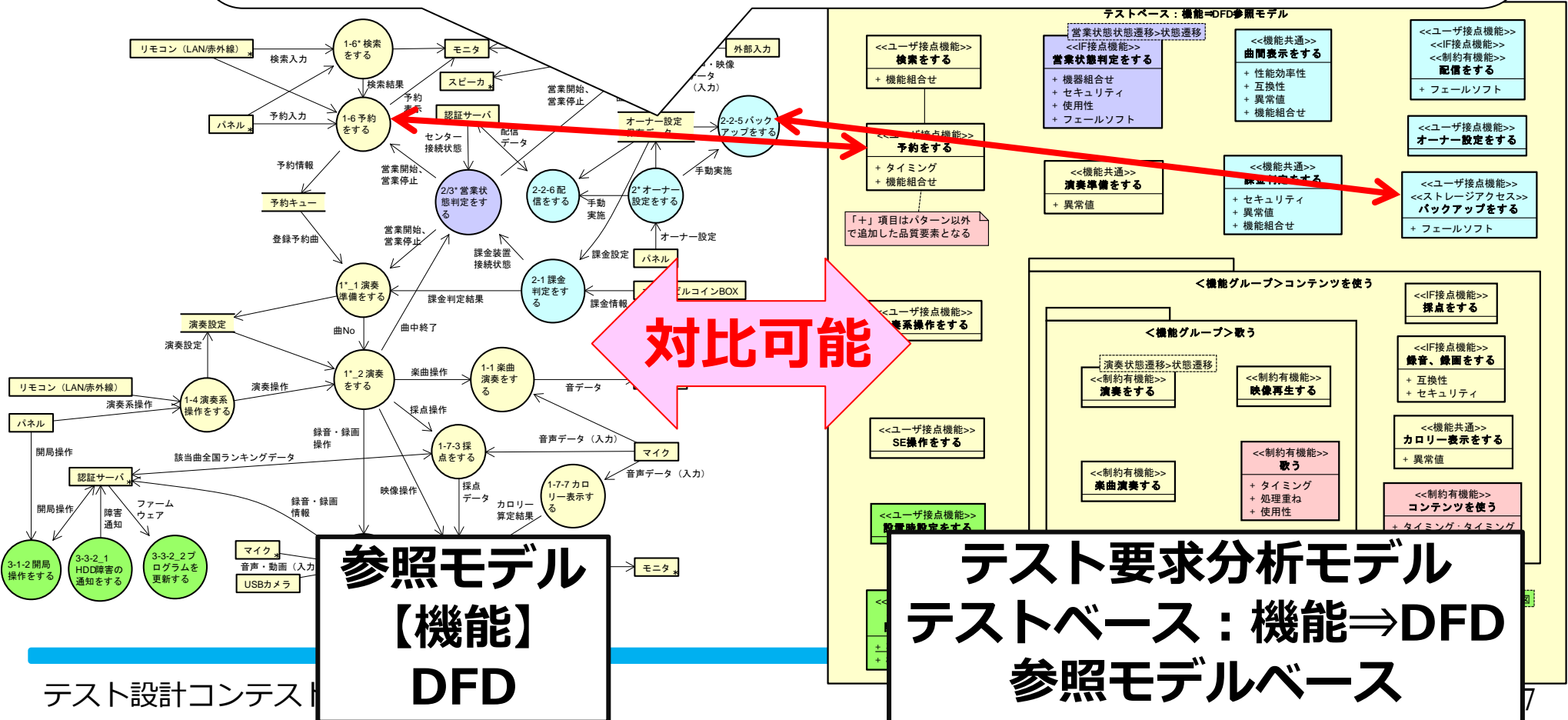
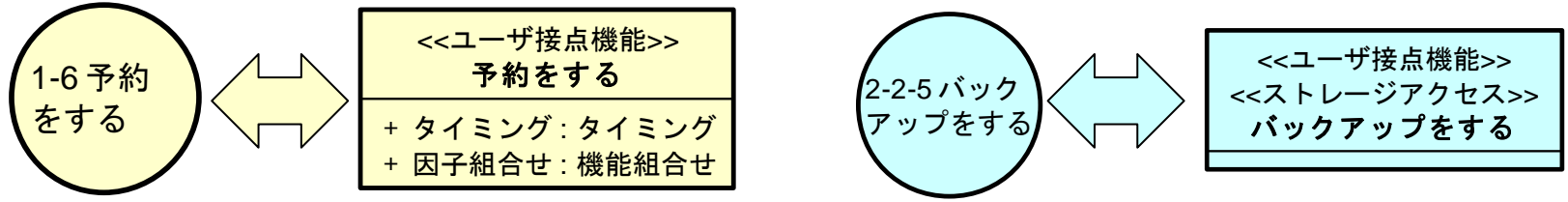
対比可能



**テスト要求分析モデル
テストベース：機能⇒DFD
参照モデルベース**

2. DFDと対応するテスト要求：機能⇒DFD参照モデルベース

DFDと対応することで、機能との対応を分かりやすくしております。



2. キーとなる品質要素の割り当て

- 機能適合
- 性能効率
- 信頼性
- 使用性
- 互換性
- 移植性
- セキュリティ

- 異常値
- 並行処理
- 機能組合せ
- 機器組合せ
- 処理重ね
- タイミング
- 不具合確認
- 通信費
- フェールソフト

機能の特徴にあわせて
キーとなる品質要素の
割り当てをする
「テスト要求パターン」
を設定してます。

例：

ユーザ接点機能

- ~ 機能適合
- ~ 異常値
- ~ 性能効率
- ~ 使用性
- ~ 信頼性

ユーザ接点機能

- ~ 機能適合
- ~ 異常値
- ~ 性能効率
- ~ 使用性
- ~ 信頼性

IF接点機能

- ~ 機能適合
- ~ 異常値
- ~ 並行処理
- ~ 性能効率
- ~ 相互運用
- ~ 信頼性

ストレージアクセス

- ~ 機能適合
- ~ 並行処理
- ~ 性能効率
- ~ 信頼性
- ~ 相互運用

**テスト要求
パターン**

機能共通

- ~ 機能適合
- ~ 信頼性

制約有機能

- ~ 機能適合
- ~ 性能効率
- ~ 信頼性

**<<ユーザ接点機能>>
予約をする**

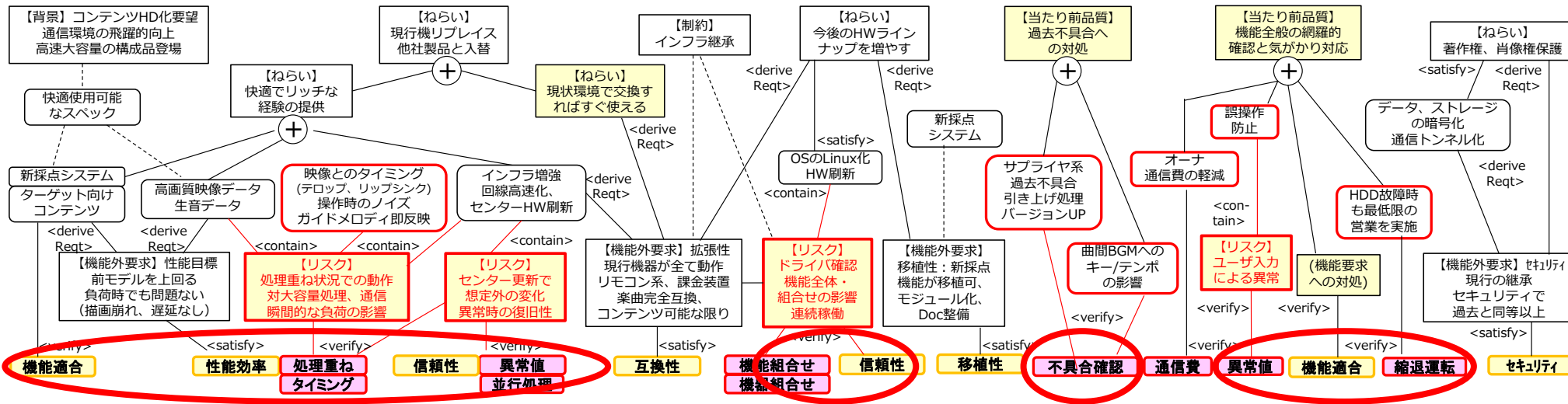
- ~ 機能適合性：機能適合性
- ~ 異常値：異常値
- ~ 性能効率：性能効率性
- ~ 使用性：使用性
- ~ 信頼性：信頼性
- + タイミング：タイミング
- + 因子組合せ：機能組合せ

+ 予約登録因子組合せ(2因子網羅)：因子組合せ

2. テスト達成マップからテスト状況をチェック

機能面からの切り口のテストでは、テスト達成マップ上のすべての品質要素を確認できていたわけではありません。
 ※赤で○のついている部分のみを確認しております。

テスト達成マップ

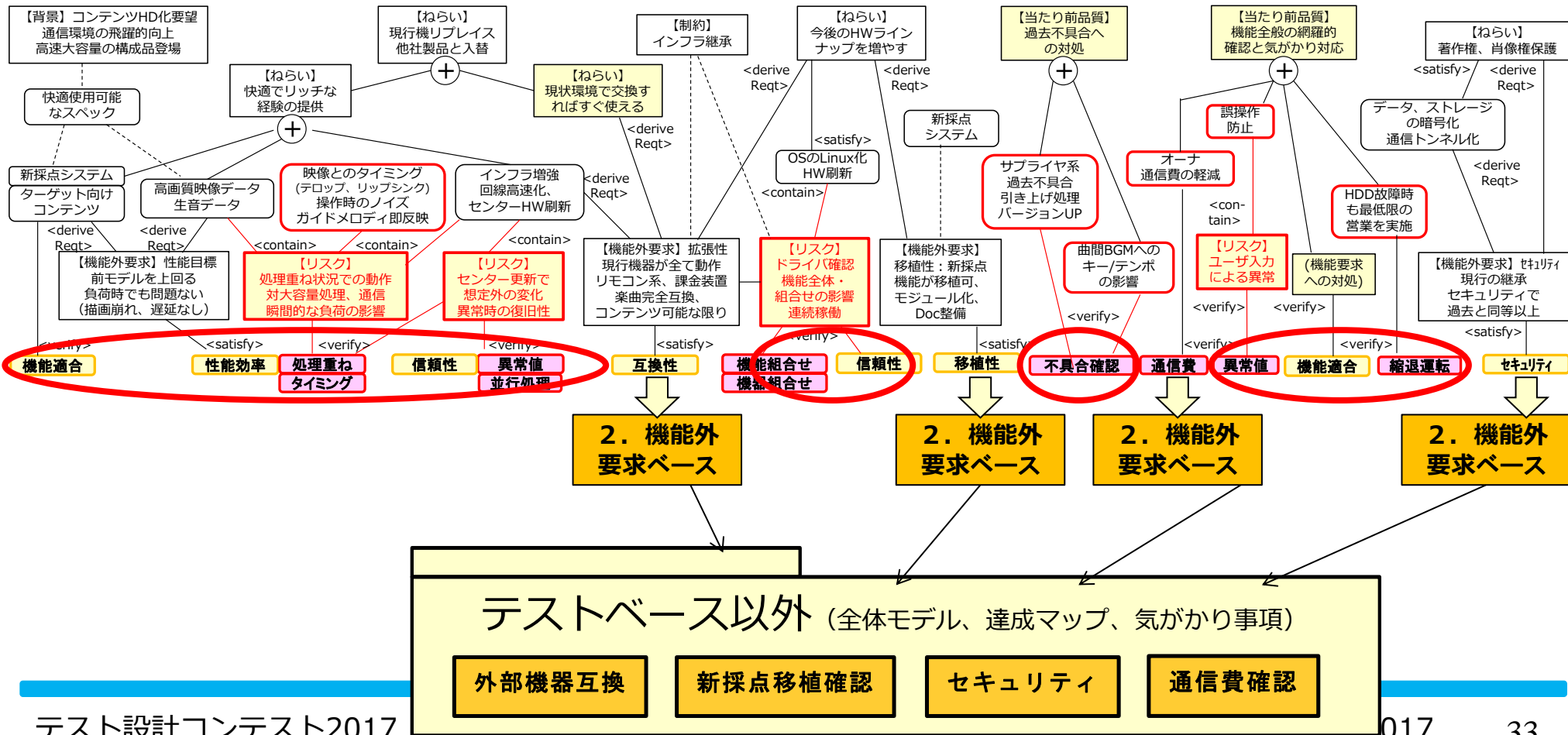


確認！ (Confirmation!) (with arrows pointing to the highlighted areas)

テスト要求分析モデル
 テストベース：機能⇒DFD
 参照モデルベース

2. テスト要求分析：テスト達成マップから確認

抜けが多いのは、「機能外要求」として記述されていたものが多いです。
これを、テストベースの「機能外要求」を満足するテストで定義しています。



2. テスト要求分析：網羅ビューによる全体像

今回のテストベース
準拠のテストの
全体像を示します。
(網羅ビュー)
テストの大枠として
以下2つに分けています。

■ テストベースの網羅
①機能：
DFD参照モデルベース

②機能外要求：
テストベース記載の
機能外以外の内容

注：成果物1には、追加提案を加えた
網羅ビューを載せております。

テスト要求モデル (網羅ビュー)

②機能外要求のテスト

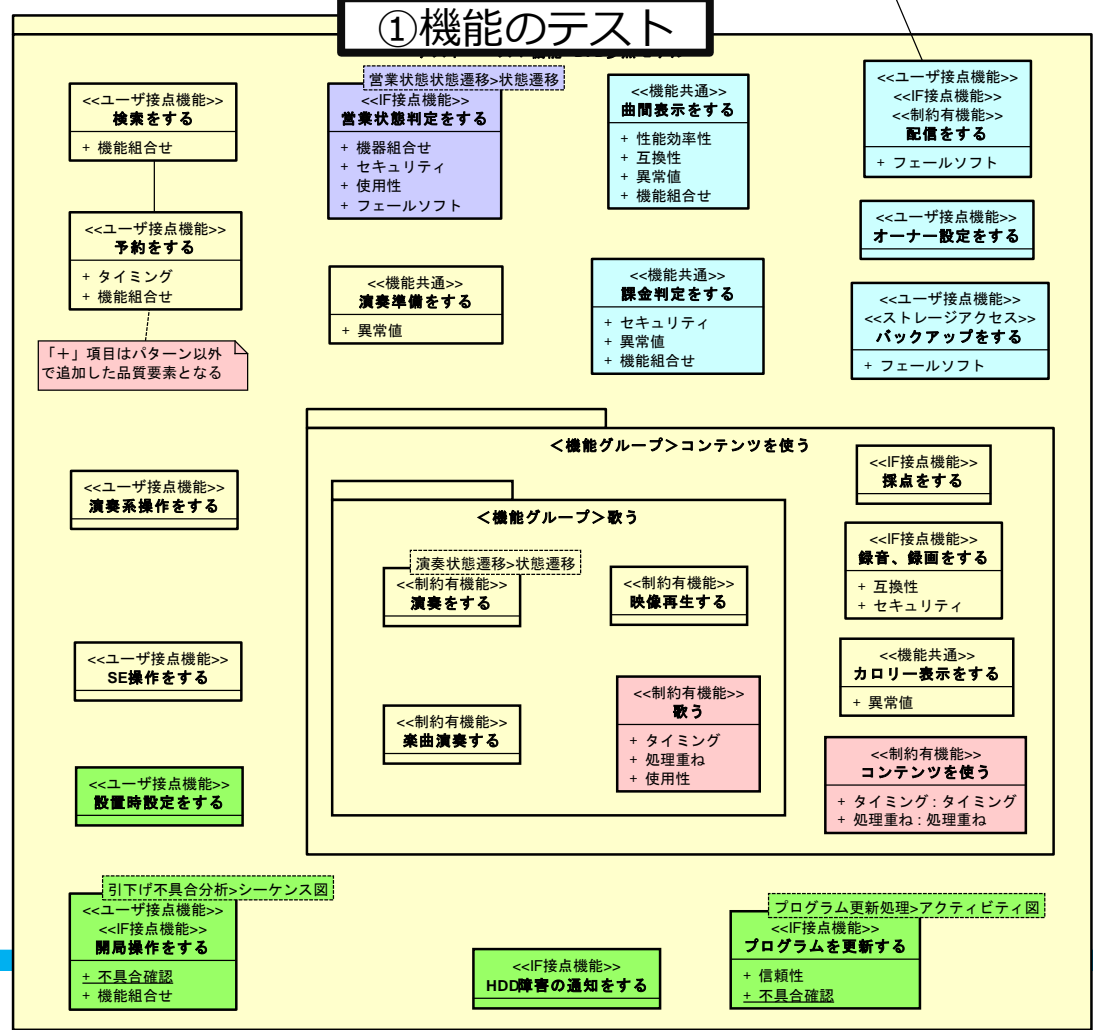
新採点移植確認
+ 移植性

セキュリティ
+ セキュリティ

外部機器互換
+ 周辺機器
+ 機器組合せ

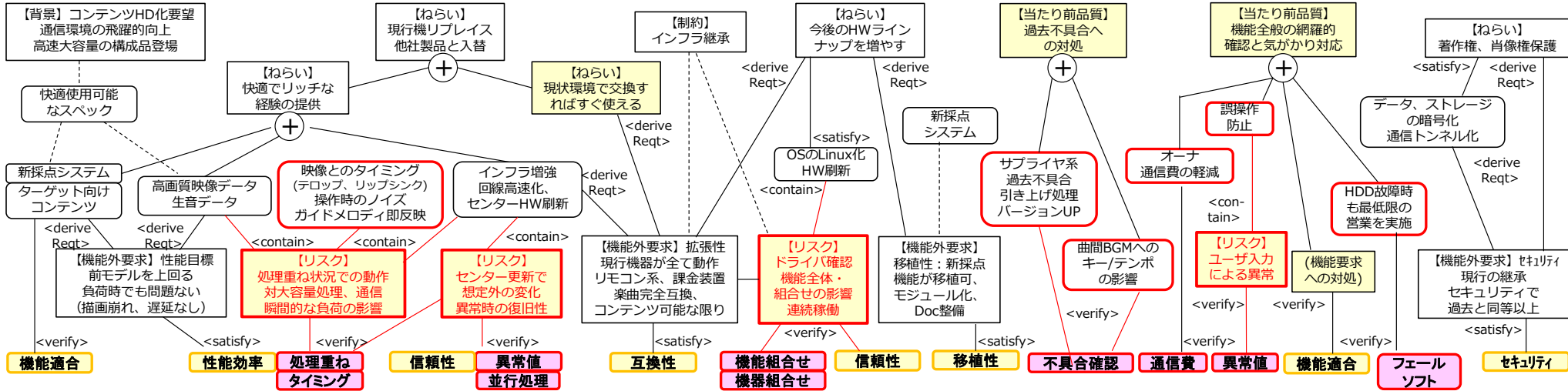
通信費確認
+ 通信費

①機能のテスト



2. テスト要求分析：達成ビューによる全体像

テスト達成マップ上で、テスト割り当て状況の確認が出来ます。(達成ビュー)



KQF (Key Quality Factor) × 各種テスト

機能適合 信頼性	性能効率	信頼性 処理重ね タイミング	異常値	信頼性 並行処理	互換性 機器 組合せ	機能 組合せ	移植性	セキュリティ	不具合確認	フェール ソフト	通信費
テストベース 機能⇒DFD 参照モデル	ユーザ接点機能 IF接点機能 制約有機能	歌う コンテンツを 使う	ユーザ接点機能 IF接点機能	IF接点機能	IF接点機能 外部機器互換	テストベース 機能⇒DFD 参照モデル	新採点移植確認	セキュリティ 録音・録画 をする 営業状態 判定をする	<<IF接点機能>> プログラムを更新する <<ユーザ接点機能>> <<IF接点機能>> 開局操作をする	バックアップをする 配信をする	通信費確認

テスト要求モデル
(達成ビュー)

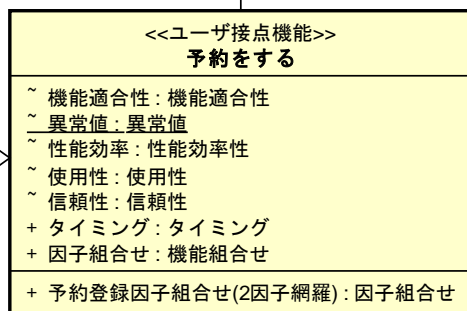
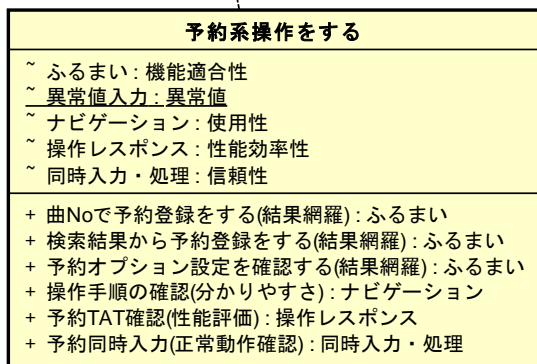
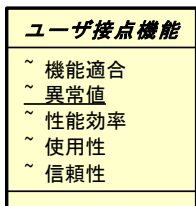
注：成果物1には、追加提案を加えた達成ビューを載せております。

2. テスト要求分析：テスト要求モデル詳細

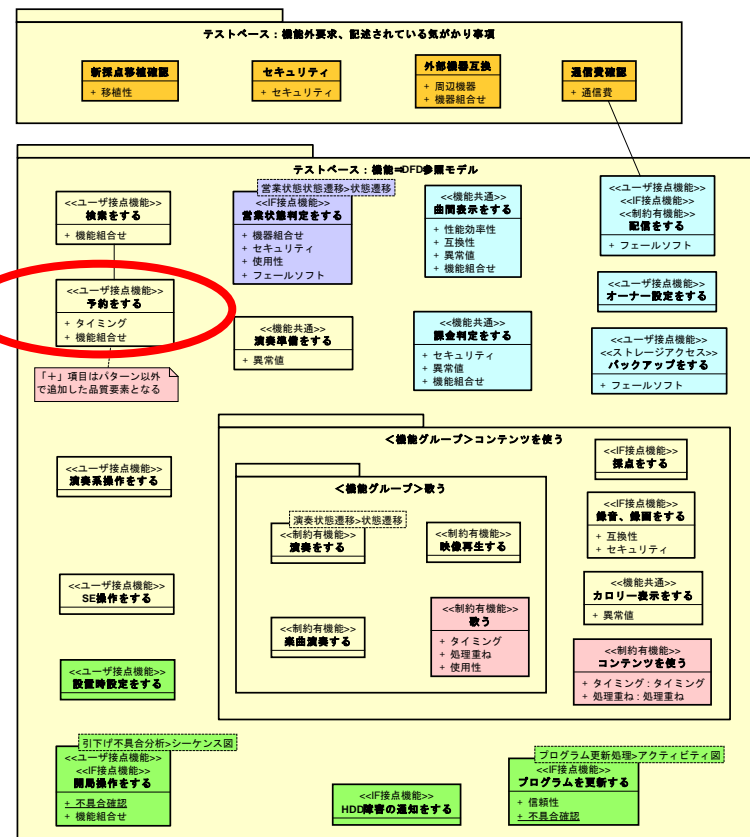
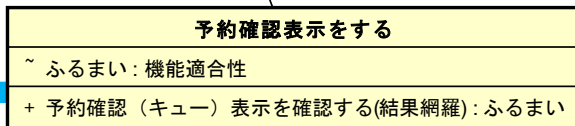
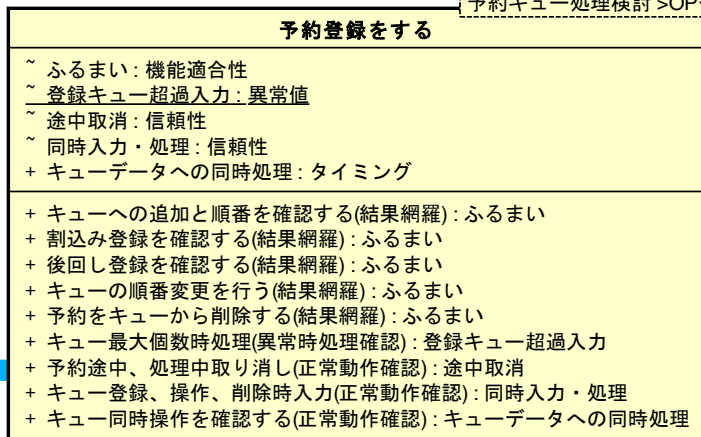
テスト要求モデル詳細を、網羅ビュー上の各テストに対応して作成しております。

テスト要求詳細図 ≡各テストの構造

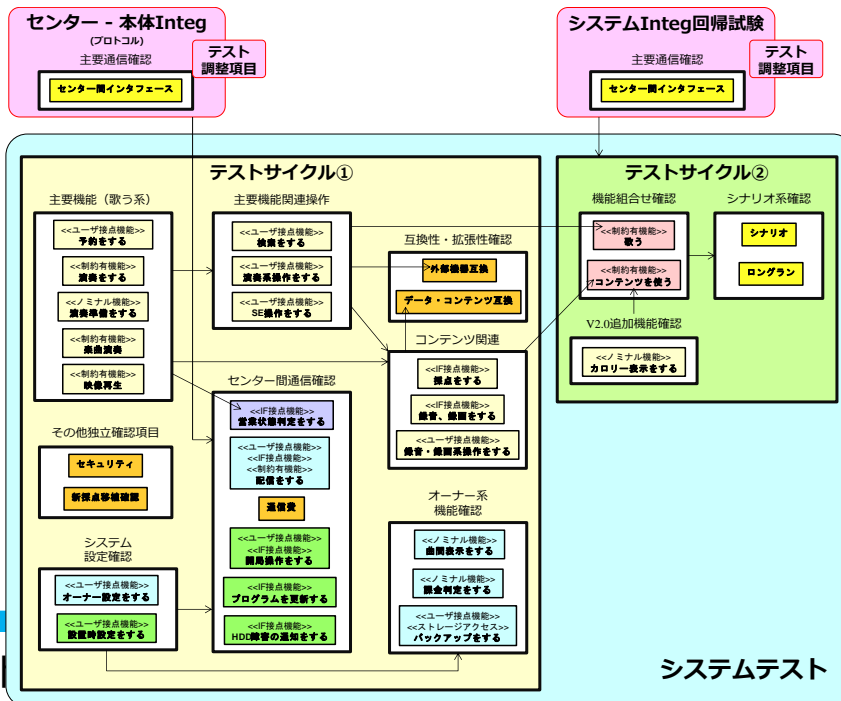
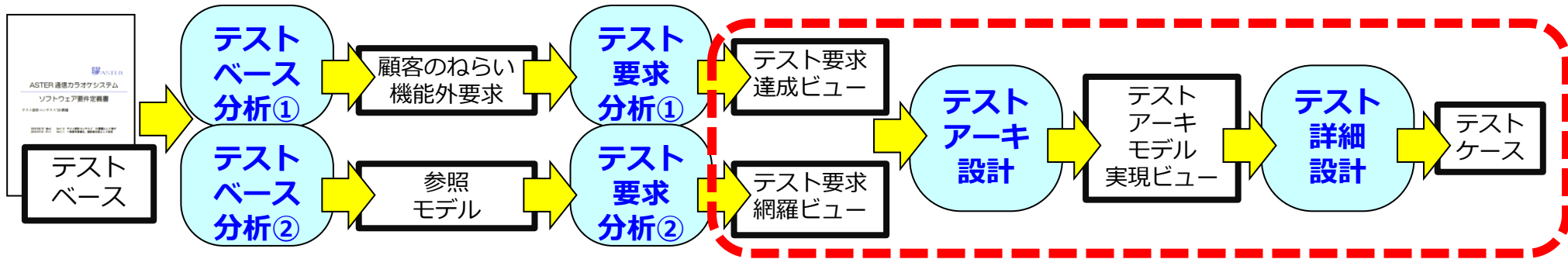
異常値入力の確認は機能適合のテストに含まれる



予約キュー処理検討>OPモデル



3. どうやってテストすべきか？

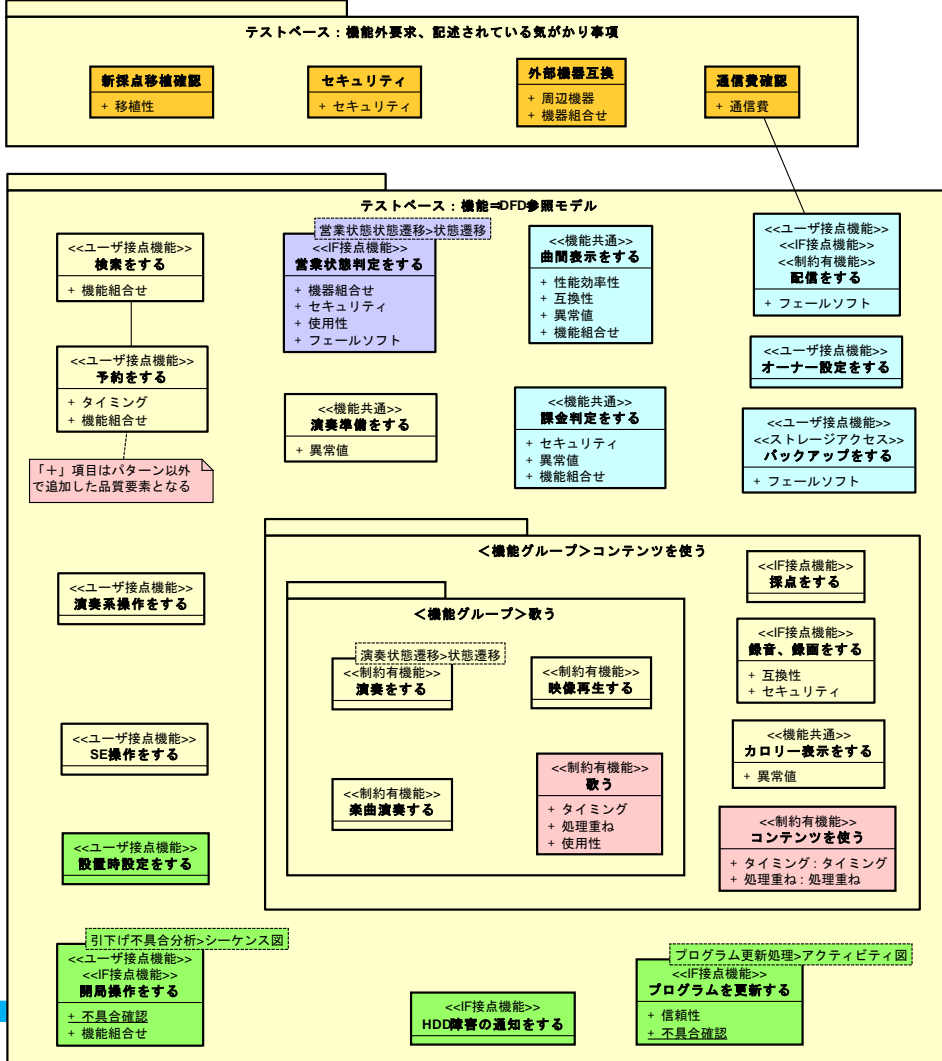


テストを実現するための方法構築と具体的なテストケースを設計します

3. テストの実現方法を決める

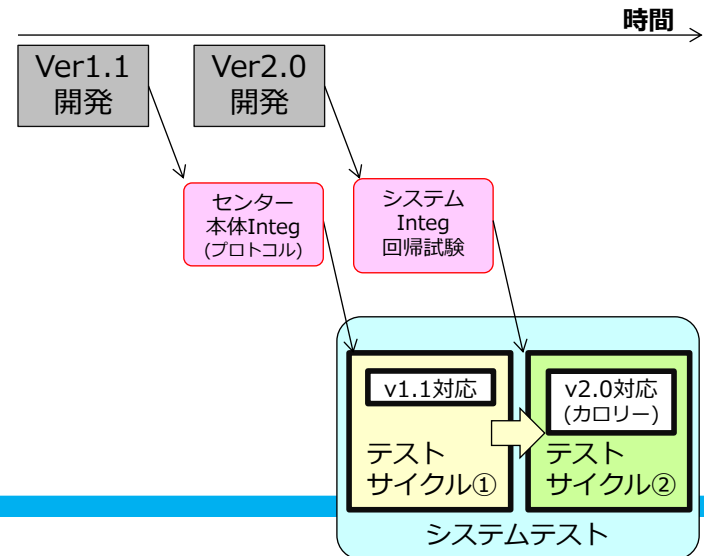
テスト要求モデル
(網羅ビュー)

実現へ

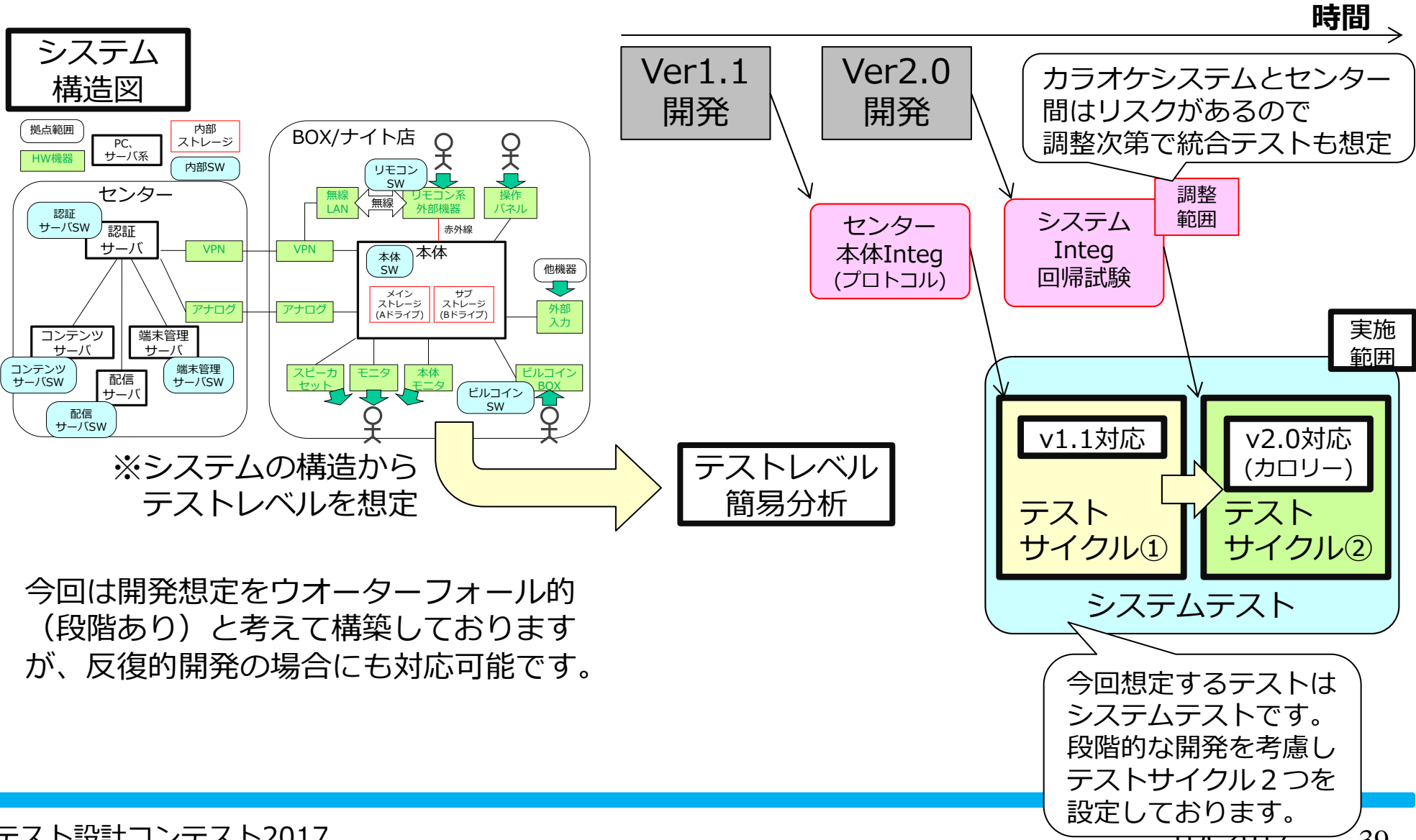


テスト要求の検討結果を実施する際には、テストの順序を考える必要があります。

(本来はお客様側の状況を確認して決めるべきですが)今回は、弊社側の想定を入れてテストの実現方法を考えました。



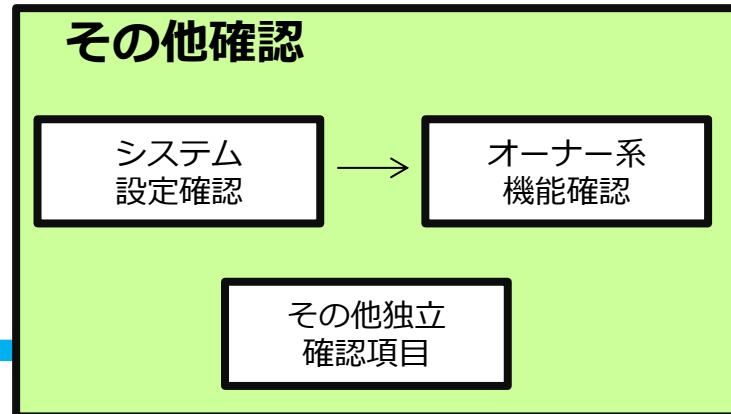
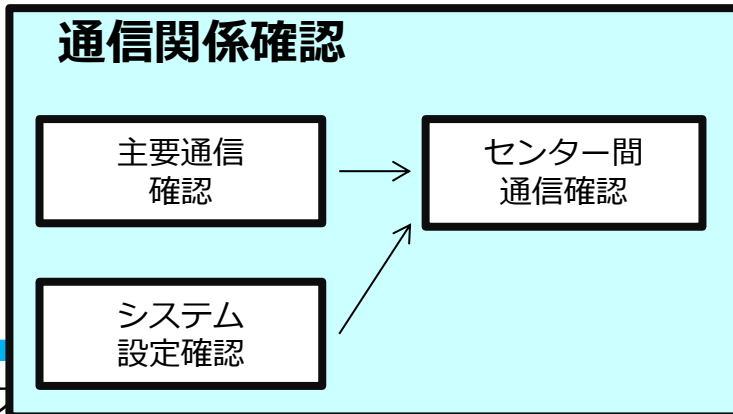
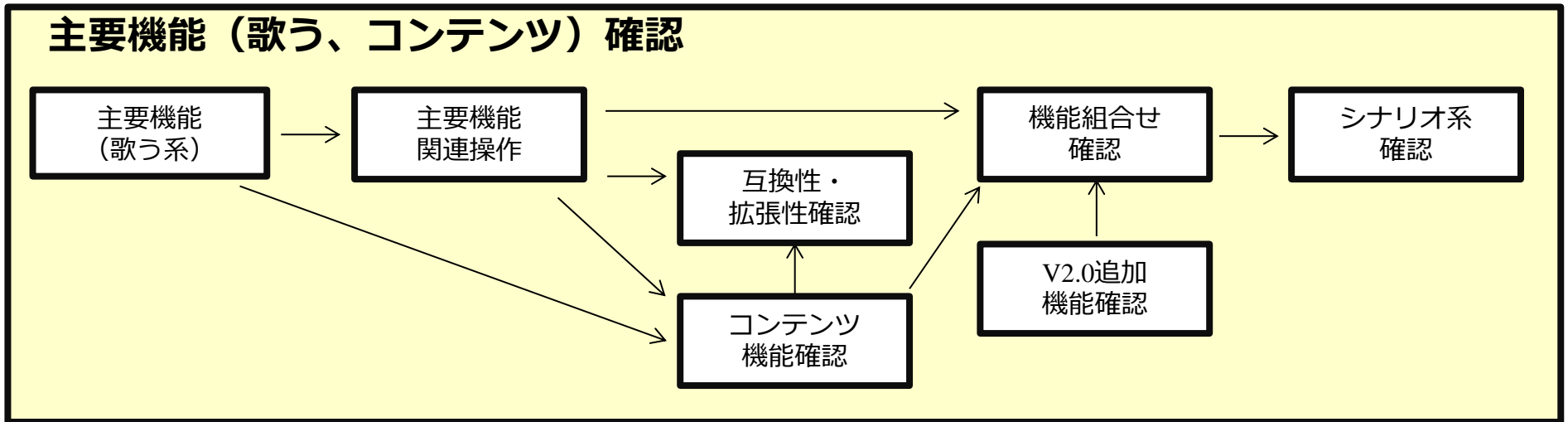
3. 想定のテストレベル



今回は開発想定をウォーターフォール的（段階あり）と考えて構築しておりますが、反復的開発の場合にも対応可能です。

3. テストの順序を決める

テストの実現方法を考えるため、テストの実現に向けた分類と順序関係を整理。
 順序を決める要素は「テスト対象の粒度（例：単体機能⇒複数機能）」の考慮、
 「同時に検討及び実施のやりやすさ」、「その他制約（バージョン単位）」。



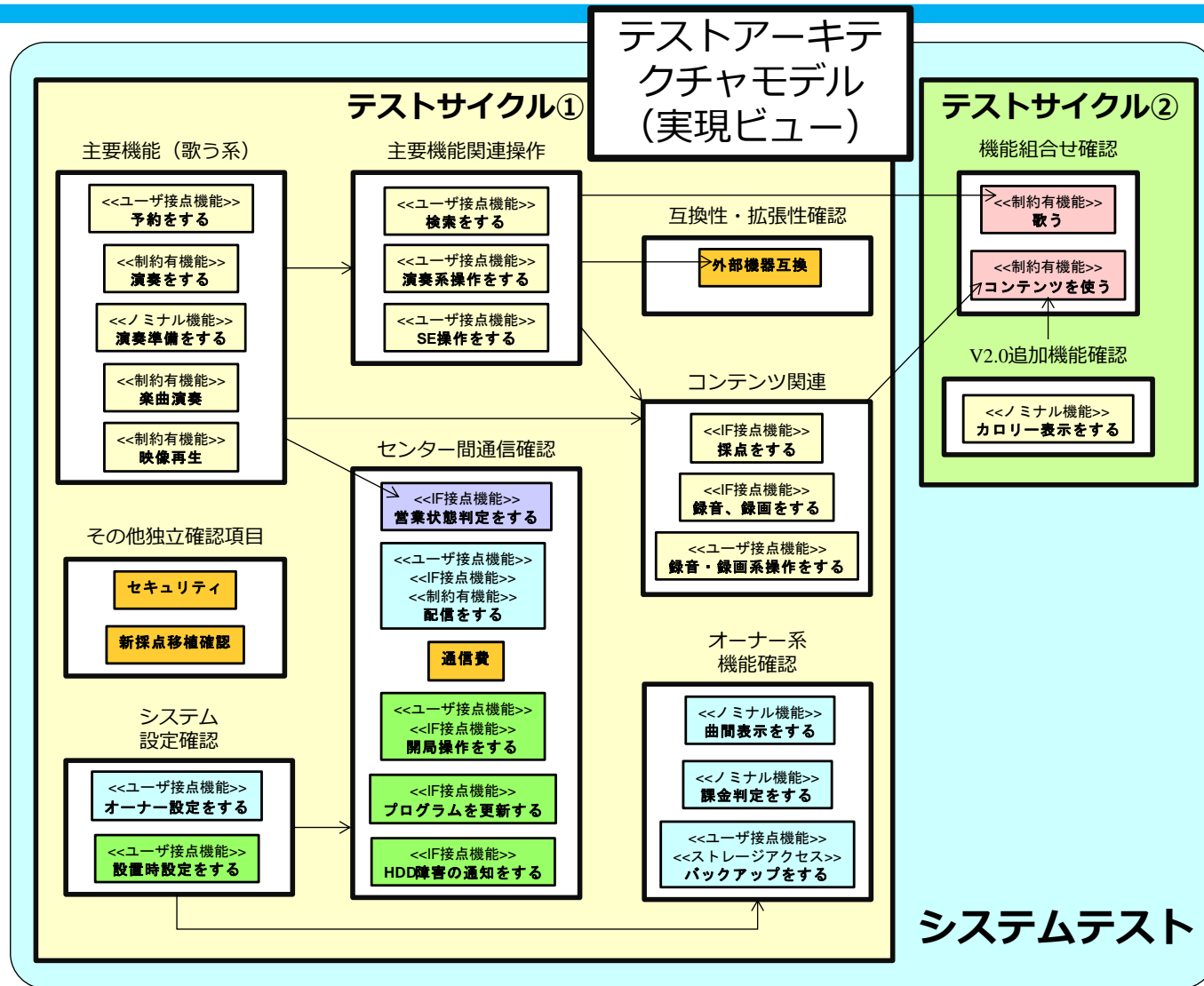
3. 各テストサイクルへ割り当て⇒実現ビューへ

具体的にテストを実現するためのモデルです。
(実現ビュー)

検討した順序をテストサイクルに割り当てることで、テストを実現するための順序関係を明確化しております。

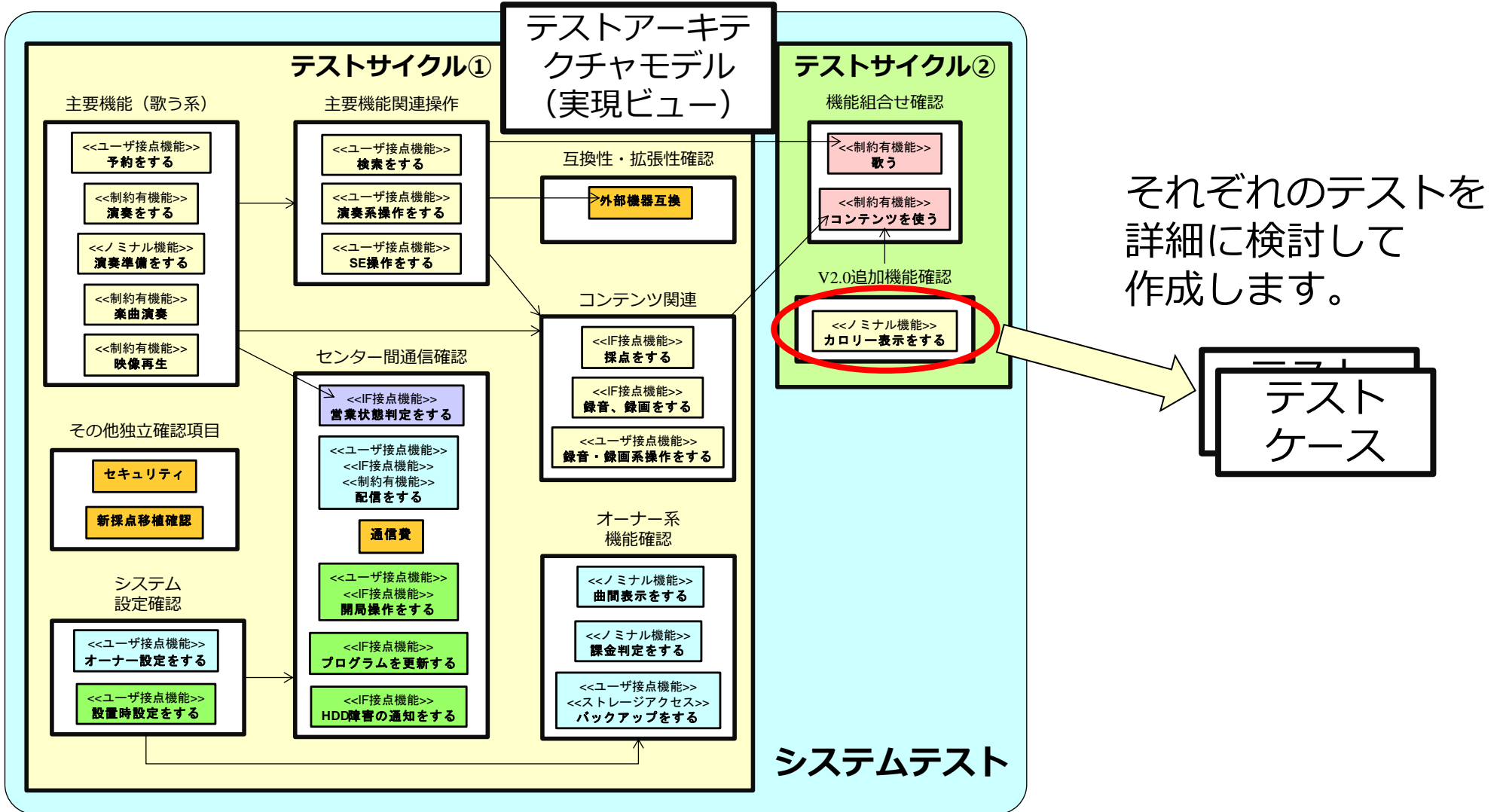
テストサイクルには、リスクが高い*1ものはテストサイクル①に入れています。

*1：リスクは各テストにおいて、不具合の発生率と不具合の重大性を考慮して決定しています。



注：成果物1には、追加提案に対する達成ビューを載せております。

4. 具体的なテストケースをつくる

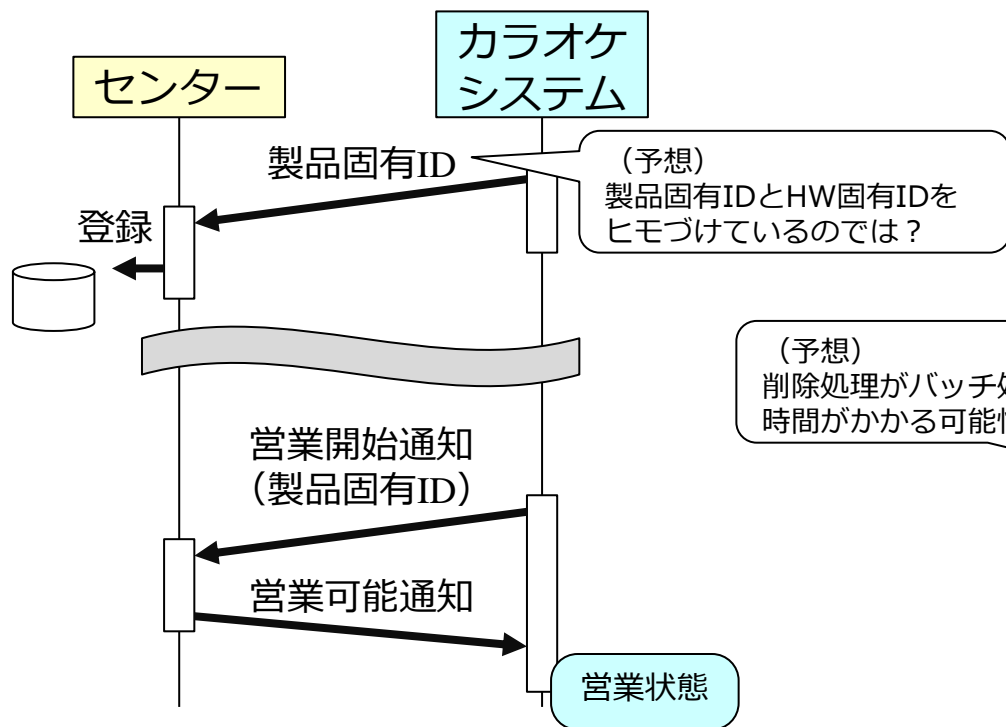


4. テスト詳細設計：仕組みが複雑な対象の検討：例

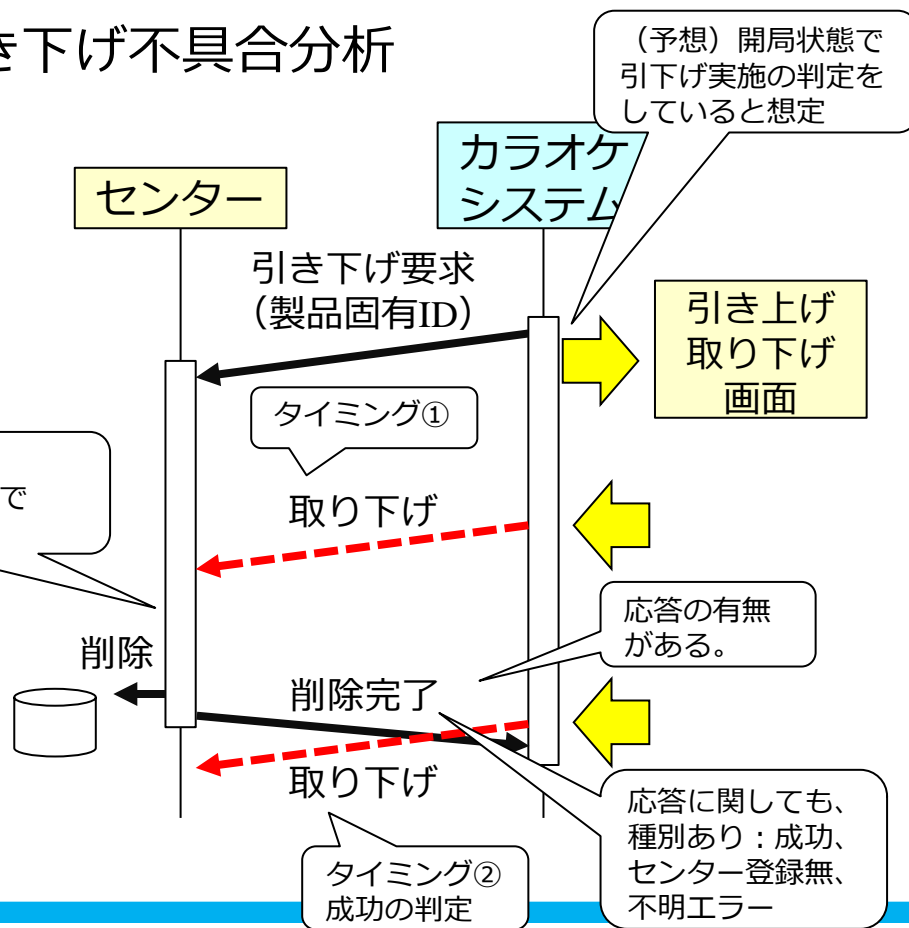


対象の仕組みが複雑な場合の例：「開局をする」に対してのシーケンス検討
⇒過去不具合の可能性を含めた「想定」をモデルとして作ります。

開局処理（開局要求～配信実施）



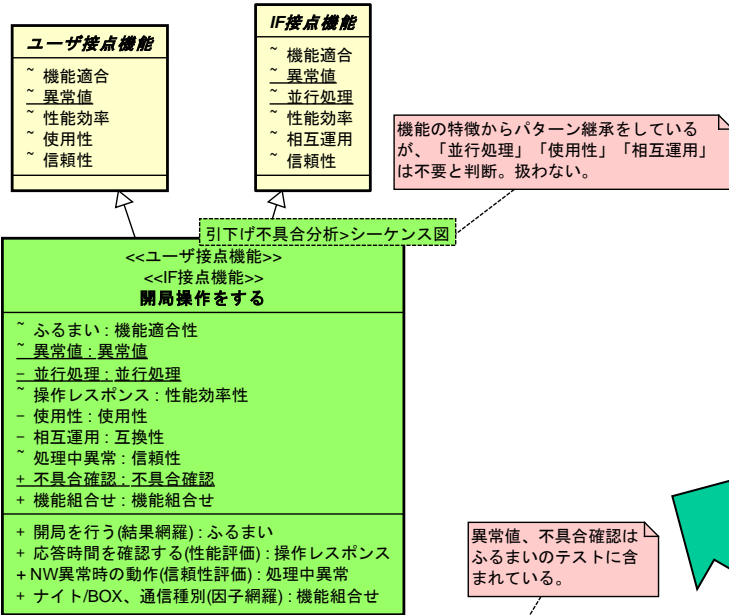
引き下げ不具合分析



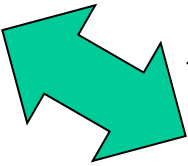
4. テスト詳細設計：仕組みが複雑な対象の検討：例



対象の仕組みが複雑な場合の例：「開局をする」
 全体のテストの内容をテストカテゴリ検討マトリクスで考えます。



テストカテゴリ検討マトリクスと対応を取りながら、テスト検討を具体化していく。



異常値、不具合確認はふるまいのテストに含まれている。

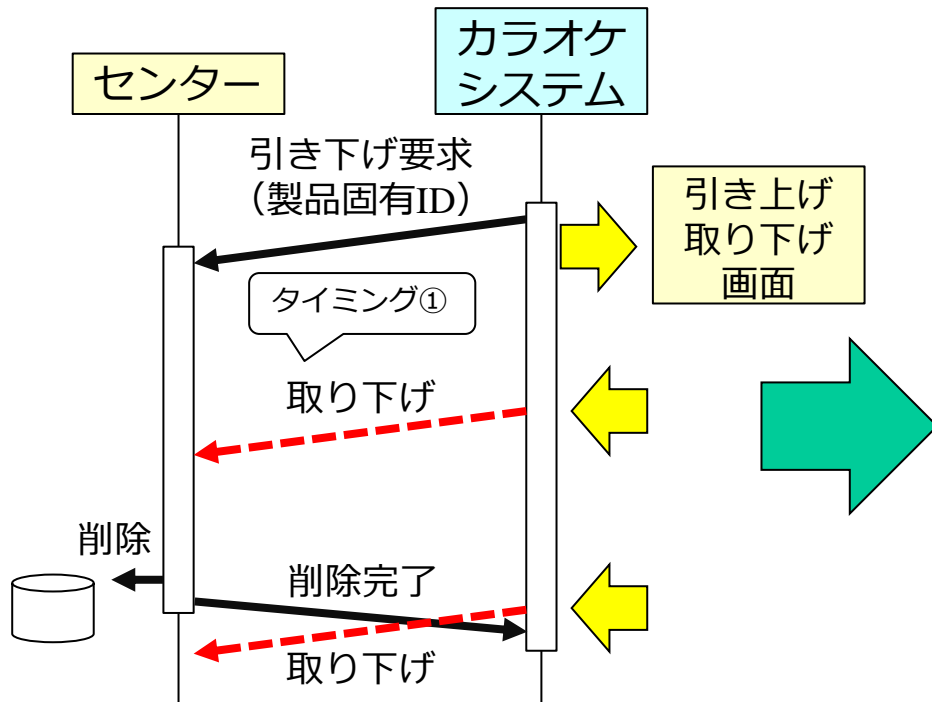
新規開局をする		品質特性				欠陥検出戦略/ガイドワード		
Targetカタマリー	機能適合性	性能効率性	信頼性	異常値	機能組合せ	不具合確認		
開局操作をする 分割機能	ふるまい	操作レスポンス	同時入力・処理	処理中異常	入力値異常	因子網羅	不具合確認	
開局操作をする	・開局を行う	・応答時間を確認する		・NW異常時の動作		・ナイト/BOX、通信種別		
新規開局をする	・新規開局を行う ・開局状態を確認する			・処理中LAN切断確認	・切断後再接続をする			
引下げ・中止指示をする	・引下げ及び中止処理を行う		・センターバッチ処理中の処理	・処理中LAN切断確認		※ふるまいで確認		※ふるまい、処理中異常で確認

4. テスト詳細設計：モデルからパラメータの検討

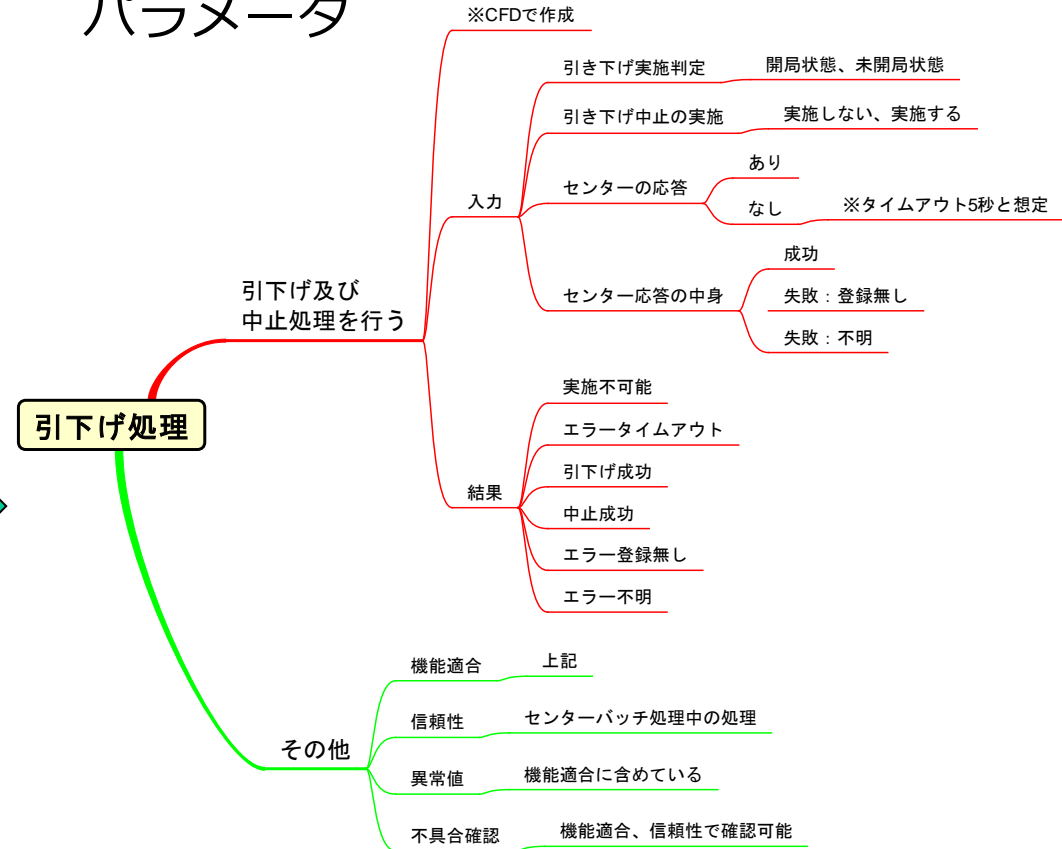


(必要に応じて) 作成したモデルからパラメータの検討を実施します。

引き下げ不具合分析



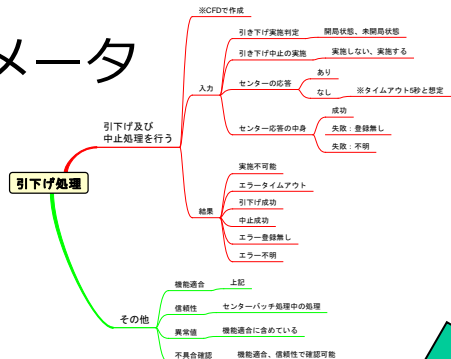
パラメータ



4. テスト詳細設計：パラメータから組合せ整理

パラメータのロジックをCFDで整理を行い、デシジョンテーブルを作ります。

パラメータ

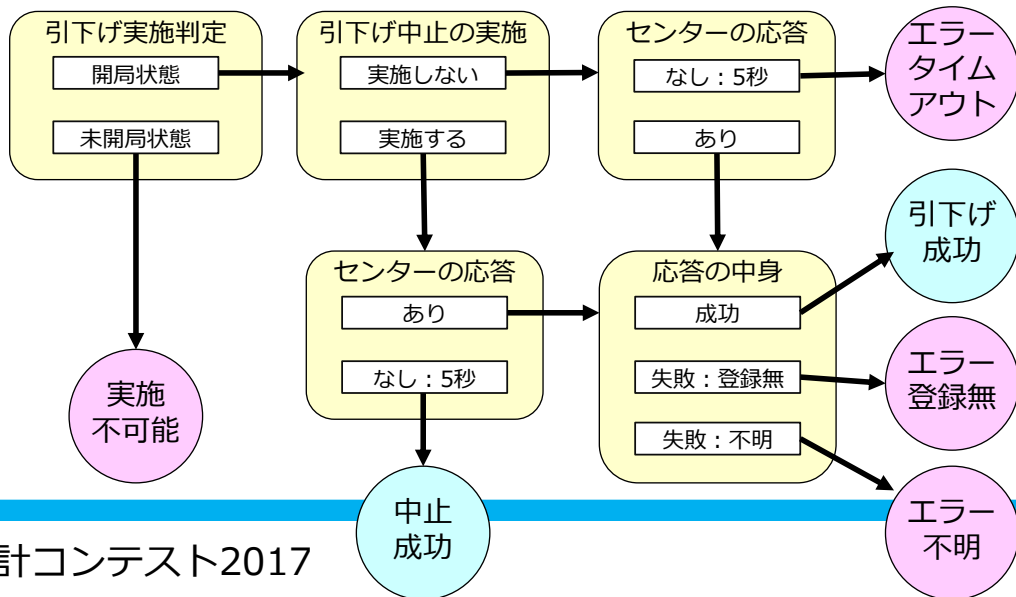


デシジョンテーブル

ID	DT3-1-2D_03001	#1	#2	#3	#4	#5	#6	#7	#8	#9
入力条件										
1	引下げ実施判定									
1	1 開局状態	-	○	○	○	○	○	○	○	○
2	2 未開局状態	○	-	-	-	-	-	-	-	-
2	引き下げ中止の実施									
1	1 実施しない	-	○	○	○	○	-	-	-	○
2	2 実施する	-	-	-	-	-	○	○	-	-
3	センターの応答									
1	1 あり	-	-	○	-	-	-	○	-	○
2	2 なし	-	○	-	-	-	○	-	-	-
4	応答の中身									
1	1 成功	-	-	○	-	-	-	○	-	-
2	2 失敗：登録無し	-	-	-	○	-	-	-	○	-
3	3 失敗：不明	-	-	-	-	○	-	-	-	○
期待結果										
1	実施可能、不可能									
1	1 実施可能	-	○	○	○	○	○	○	○	○
2	2 実施不可能	○	-	-	-	-	-	-	-	-
2	成功・失敗									
1	1 引下げ成功	-	-	○	-	-	-	○	-	-
2	2 中止成功	-	-	-	-	-	○	-	-	-
3	3 失敗	-	○	-	○	○	-	-	○	○
1	エラー通知									
1	1 タイムアウトエラー	-	○	-	-	-	-	-	-	-
2	2 登録無しエラー	-	-	-	○	-	-	-	○	-
3	3 不明エラー	-	-	-	-	○	-	-	-	○

CFDによる整理

DT3-1-2D_03001



4. テスト詳細設計：テストケースへの展開

最終的にテストケースのひな型へ落とし込んでおります。

※テスト管理ツールでも管理可能なフォーマットになっております。

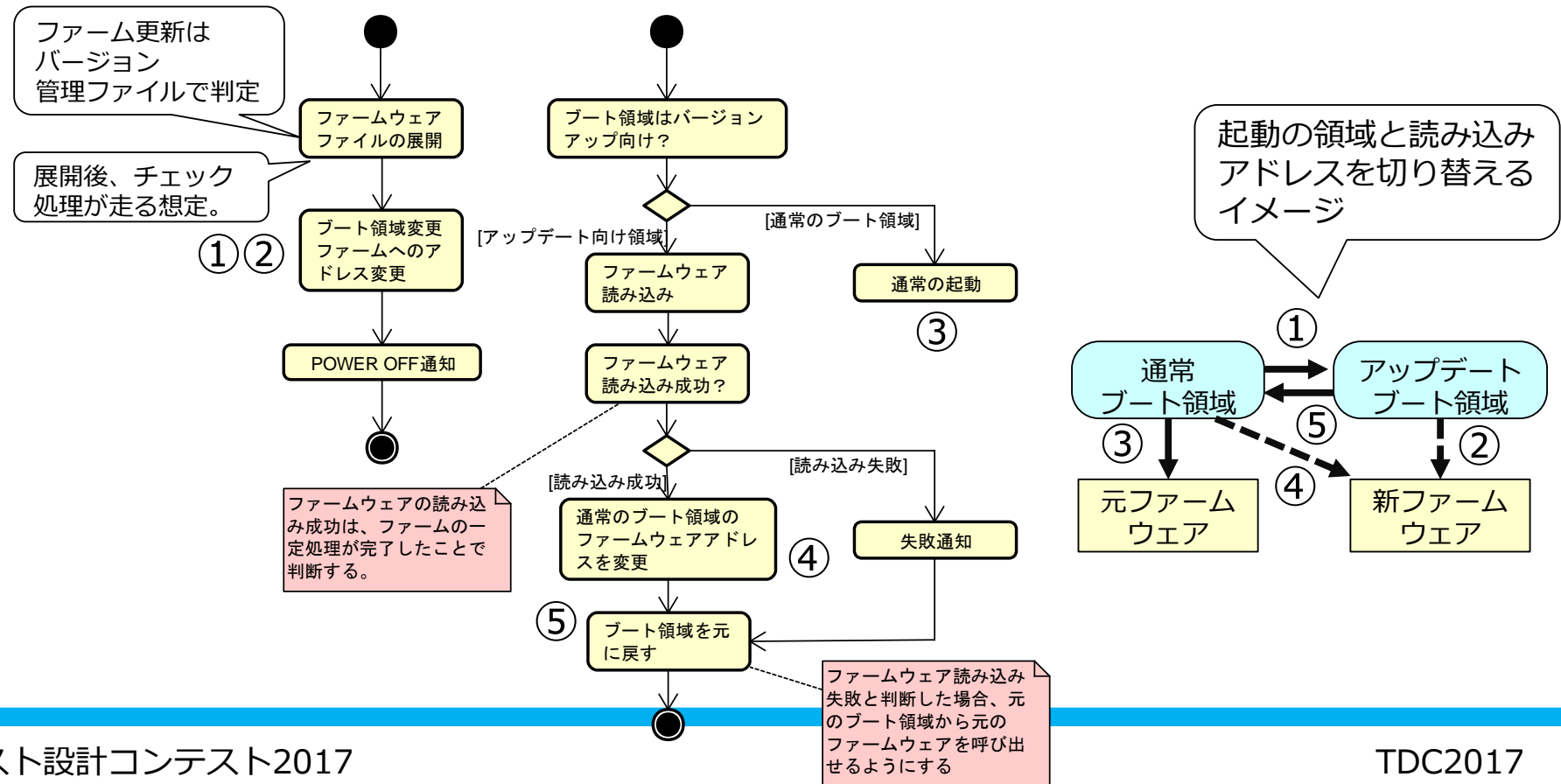
試験カテゴリ/ID/属性	試験項目名	試験概要	試験実施方法(案)	合否判定基準	ケース/設定	備考
3-1-2D: 開局操作をする						
3-1-2D 01XXX 開局操作をする						
3-1-2D 010XX 開局操作をする: 機能適合性						
3-1-2D_01001	交換開局: 新規開局及び引き下げを実施する	新規開局及び引き下げの処理を行う交換開局の手順を実施を行う。 以下記載の順番を実施する。 3-1-2D_03001 引下げ及び中止処理を行う 3-1-2D_02001 開局を行う	1. 引下げを実施する 2. 開局を実施する	引下げ、開局共に成功すること。	-	テスト自動化が出来るが良い
3-1-2D 011XX 開局操作をする: 性能効率性						
3-1-2D_01101	応答時間を確認する	以下2つのテストに対して、応答時間を確認して、長時間(30秒以上: 仮設定)時間がかからないことを確認する。 対象テスト 3-1-2D_02001 開局を行う 3-1-2D_03001 引下げ及び中止処理を行う	以下テストの手順に従う。 3-1-2D_02001 開局を行う 3-1-2D_03001 引下げ及び中止処理を行う	30秒以上など、極端に長い時間にならないこと。	-	長時間の目的は別途ステークホルダと調整
3-1-2D 012XX 開局操作をする: 信頼性						
3-1-2D_01201	NW異常時の動作	以下2つのテストに対して、(LANを抜いて)NW異常を発生させ、復旧不可能な状況に陥らないことを確認する。 対象テスト 3-1-2D_02001 開局を行う 3-1-2D_03001 引下げ及び中止処理を行う	各処理中にLANケーブルを抜く。	復旧不可能な状況に陥らないこと。	開局時 引き下げ時 引下げ中止時	
3-1-2D 013XX 開局操作をする: 互換性						
3-1-2D_01301	ナイト/BOX、通信種別	ナイト/BOXの営業店種別、NB/BBの通信種別の設定に対して、以下2つの成功ケースが成功すること。 対象テスト 3-1-2D_02001 開局を行う 3-1-2D_03001 引下げ及び中止処理を行う	1. ケース参照に記載の各種設定を実施する 2. 開局及び引き下げを行う	各環境で処理が成功すること。	BOX(BB) ナイト(BB) ナイト(NB)	
3-1-2D 02XXX 新規開局をする						
3-1-2D 020XX 新規開局をする: 機能適合性						
3-1-2D_02001	開局を行う	カラオケシステム、センター側の状況に対して開局処理が成功・失敗することを確認する。	(DTに記載の応答が来る準備をする) 1. カラオケシステムから開局処理を行う 2. センターからの応答によるふるまいを確認	2. DT記載のパターンの受理・棄却結果となること。	DT3-1-2D_02001 参照	テスト自動化が出来るが良い
3-1-2D_02002	開局状態を確認する	カラオケシステムとセンター間で開局状態を確認する処理に対して、以下の処理を確認する。 ・正常通知: 開局状態が正常に通知される ・未開局通知: 未開局というセンター応答 ・応答無し: センターからの応答がない ・エラー通知: センターからエラー応答	(準備: センターから該当の応答が来る状況にする) 1. カラオケシステムとセンターを接続する 2. センターからの応答によるふるまいを確認	以下の結果となること。 ・正常通知: 開局状態となる ・未開局通知: 未開局のまま ・応答無し: タイムアウトでエラー、未接続状態となる ・エラー通知: エラー、未開局となる	正常通知 応答無し エラー通知 未開局通知	
3-1-2D 021XX 新規開局をする: 信頼性						
3-1-2D_02101	切断後再接続をする	開局の処理実施後、センターからの応答が来る前の段階で通信が切断した場合の処理を確認する。	1. カラオケシステムとセンターを接続する 2. カラオケシステムから開局をセンターに通知したタイミングで通信を切断する。 3. 再度接続して開局操作を実施、もしくは開局にすることが出来るかどうかを確認する。	センターとの齟齬が発生しないこと、もう一度開局を実施して開局されていること、若しくはセンター側で処理成功している場合には開局状態になること。	-	
3-1-2D 03XXX 引下げ・中止指示をする						
3-1-2D 030XX 引下げ・中止指示をする: 機能適合性、異常値						
3-1-2D_03001	引下げ及び中止処理を行う	カラオケシステム、センター側の状況に対して引下げ処理及び引き下げ中止が成功・失敗することを確認する。	(DTに記載の応答が来る準備をする) 1. カラオケシステムから引下げ処理を行う 2. センターからの応答によるふるまいを確認 ※手順によっては引下げ中止を行う	2. DT記載のパターンの受理・棄却結果となること。 引下げ中止の画面が開いたままにならないこと。	DT3-1-2D_03001 参照	テスト自動化が出来るが良い
3-1-2D 030XX 引下げ・中止指示をする: 信頼性						
3-1-2D_03001	センターバッチ処理中の処理	センターがバッチ処理中の状況下で、引下げ中止を行う場合の処理を確認する。	(DTに記載の応答が来る準備をする) 1. カラオケシステムから引下げ処理を行う 2. センターからの応答によるふるまいを確認	センター側での応答に対して、3-1-2D_03001記載の結果と同じ内容が得られていること。	-	

テスト詳細設計：特殊テスト (STUDIO IBURIならではの)



以下のようなテストは、単純にテストベースから出てきません。
※STUDIO IBURIへの依頼だからこそ出てくるテストです！

プログラム更新時の不具合をアクティビティ図で想定したテスト

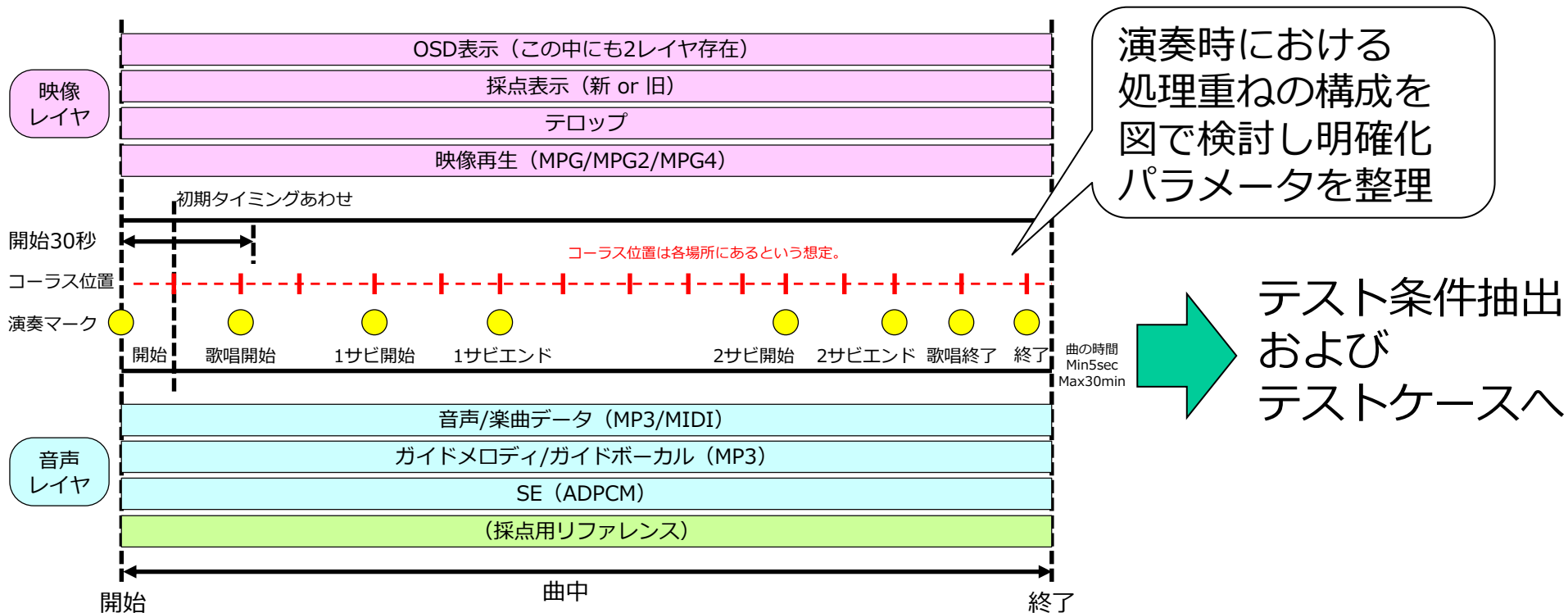


テスト詳細設計：特殊テスト (STUDIO IBURIならではの)



以下のようなテストは、単純にテストベースから出てきません。
※STUDIO IBURIへの依頼だからこそ出てくるテストです！

「歌う」際における気がかりな処理不可に対するテストの検討



テスト詳細設計：特殊テスト (STUDIO IBURIならではの)



以下のようなテストは、単純にテストベースから出てきません。
※STUDIO IBURIへの依頼だからこそ出てくるテストです！

「歌う」際におけるテロップの表示タイミングの確認

1. テロップ表示単位を想定⇒「表示グループ」と呼ぶ

ひか くも つ め
光る雲を突き抜け
フライアウェイ
Fly Away (Fly Away)



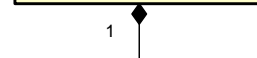
2語/1sec
→
からだじゅうに
広がるパノラマ
→
4語/1sec

2. 表示の中で
速度変わる
「再生歌詞」
単位で動作

テロップの構造は
以下のように想定。

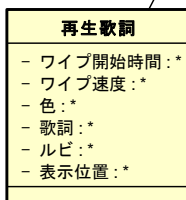
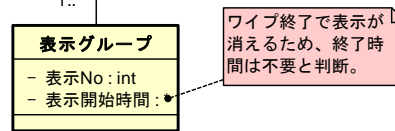
テロップのデータ・ふるまい
構造をリバースしてテストの
ための条件・パラメータ整理

(1曲分) テロップデータ



YAH YAH YAH YAH YA YA YA
Tシャツを脱ぐように

3. 再生歌詞単位は
同時動作可能



テスト条件抽出
および
テストケースへ

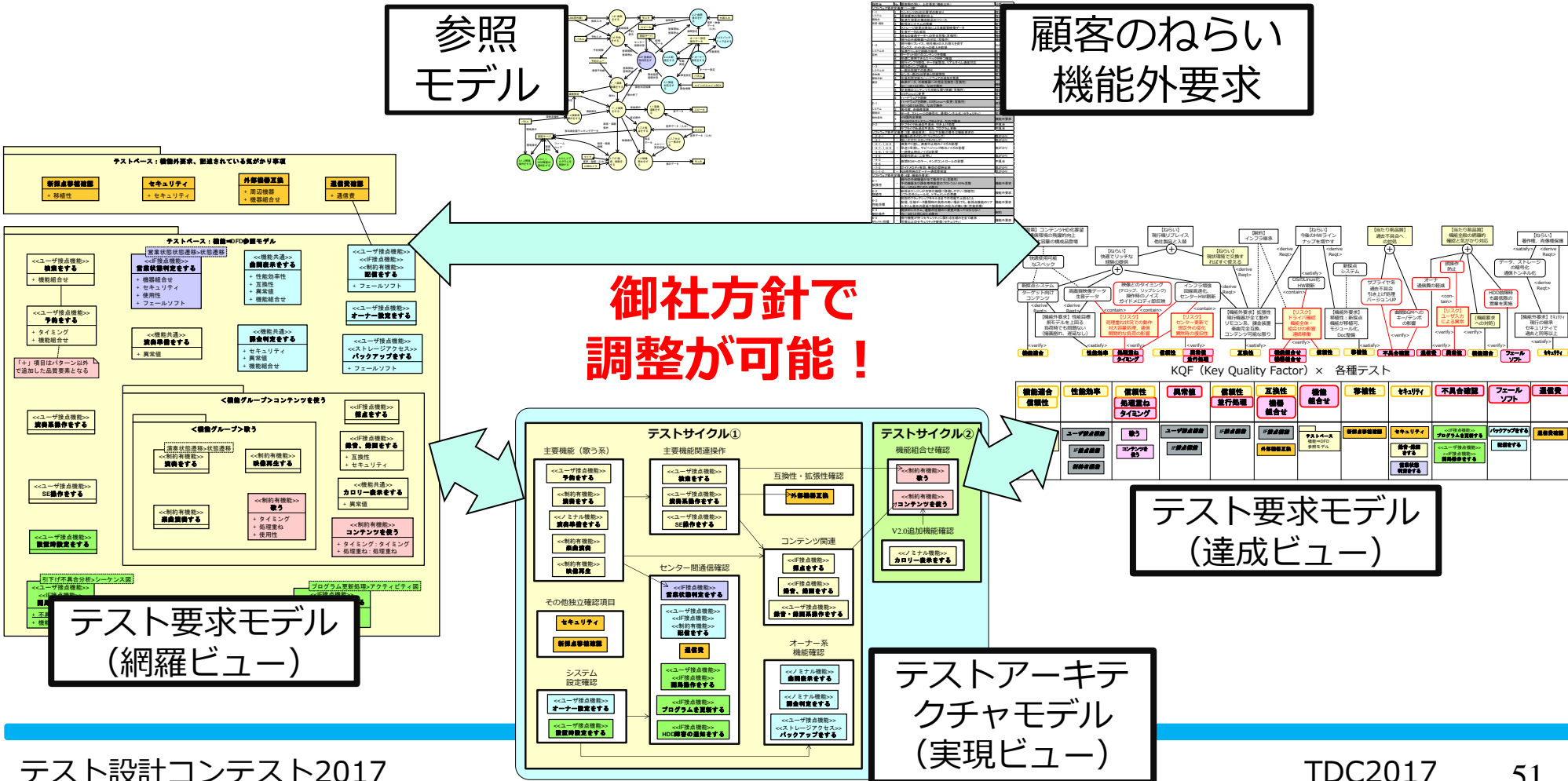
※記述DSLの例

```
<表示グループ: No1、開始時間: 0:08sec>  
<開始位置: 0:10sec、速度: 1語/sec、色: 黒>  
<rb>光[ひか]</rb><rb>雲[くも]</rb>を  
<rb>突[つ]</rb>き<rb>抜[め]</rb>け<br>  
<開始位置: 0:14sec、速度: 2語/sec、色: 黒>  
<rb>Fly[フライ]</rb><rb>Away[アウェイ]</rb>  
<開始位置: -, 速度: 0語/sec、色: 黒>  
(Fly Away)  
</表示グループ>
```

※速度0語/secの設定
はワイプしない。

調整は可能です

今までの内容は弊社の「想定」が多く入っていますが、御社の想定にあわせて達成内容や実現内容の調整は可能です！



その②：追加提案・カスタマイズ可能

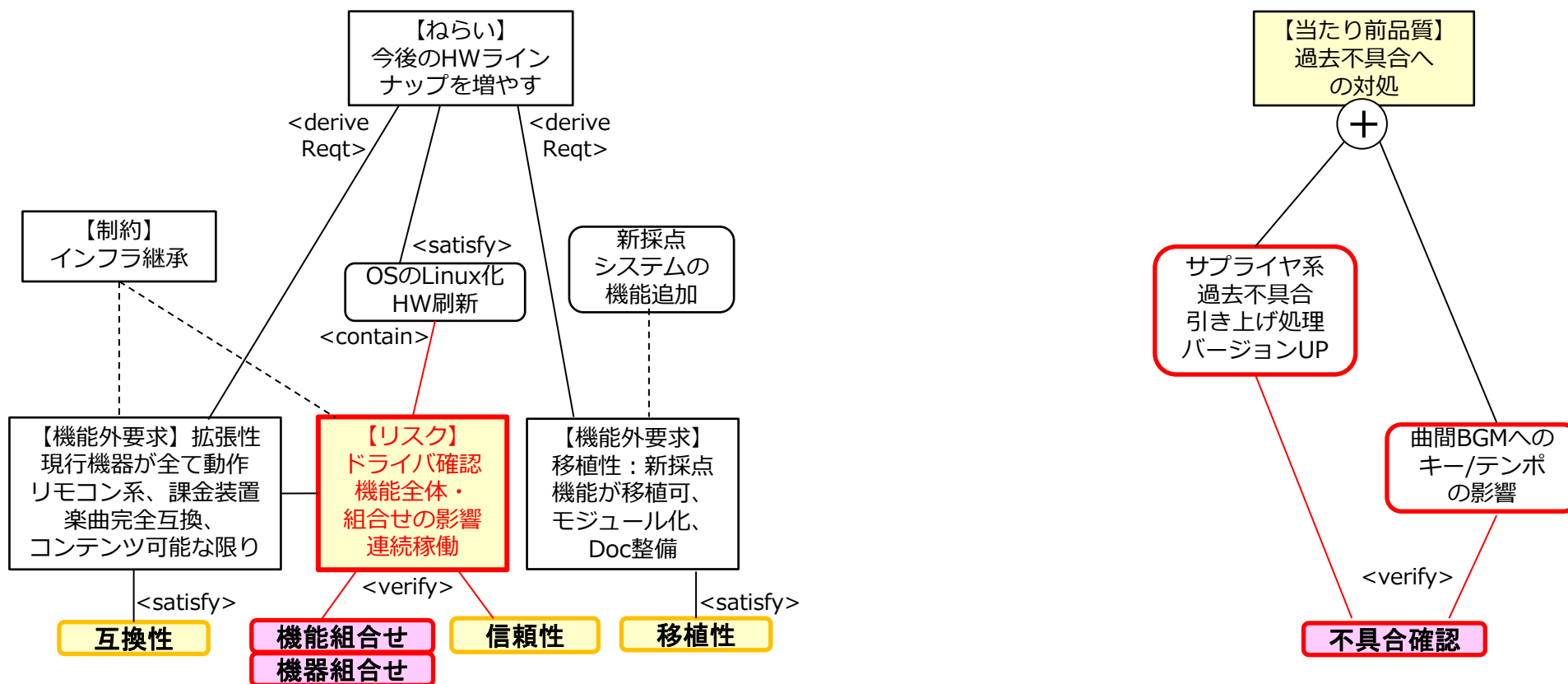
以下、顧客側の「ねらい」を達成するために
推奨するテストの提案となります。

②追加価値の提案
カスタマイズ可能

追加提案項目

「ねらい」を達成するための提案となります。

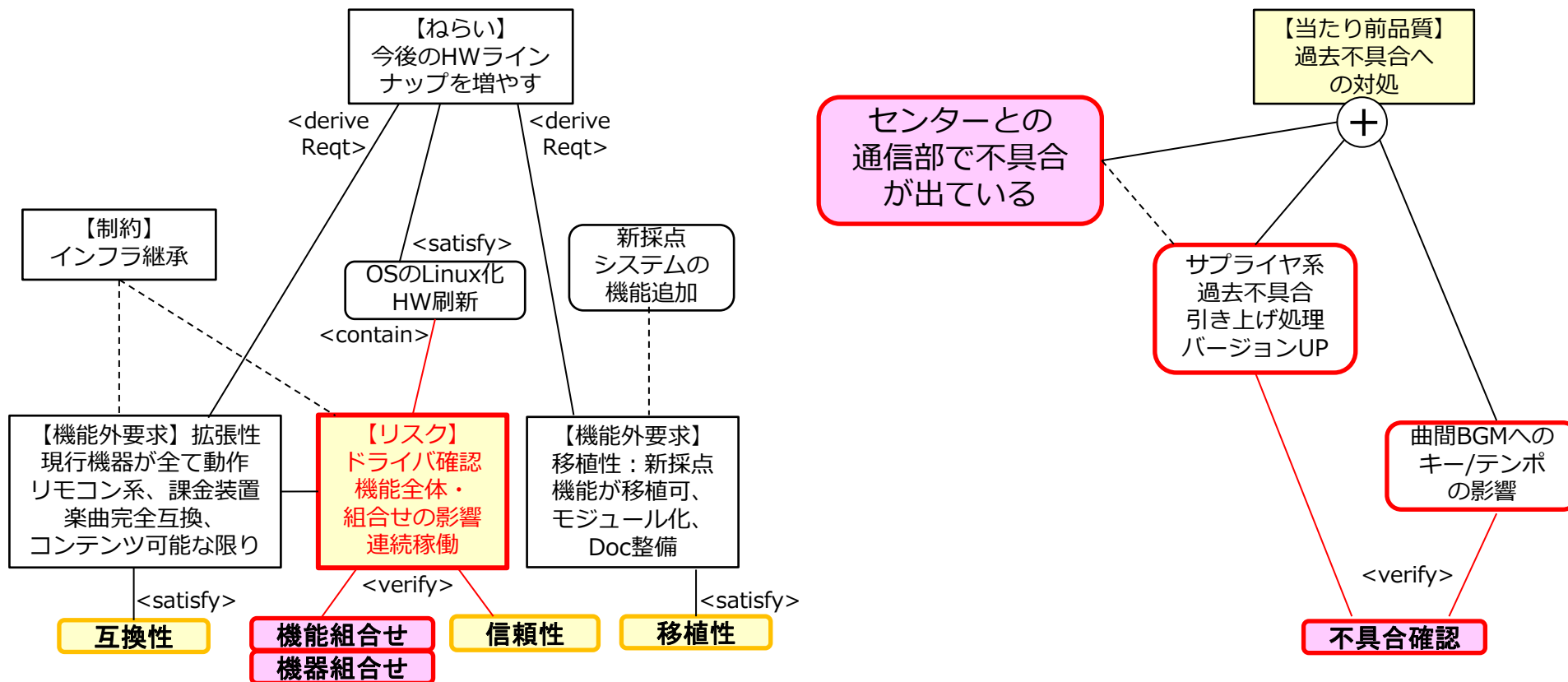
※達成マップ上の関連性で紹介



追加提案項目

「ねらい」を達成するための提案となります。

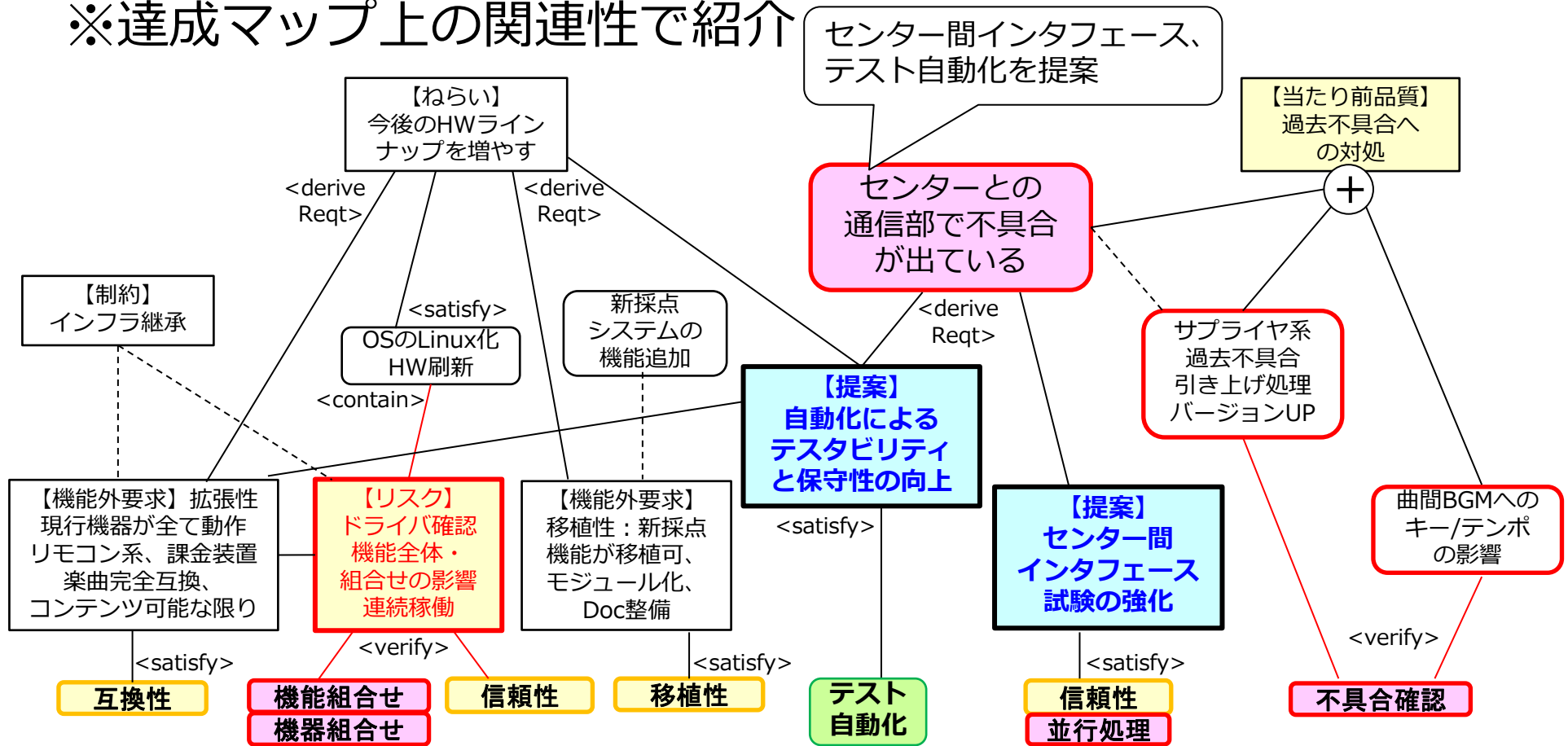
※達成マップ上の関連性で紹介



追加提案項目

「ねらい」を達成するための提案となります。

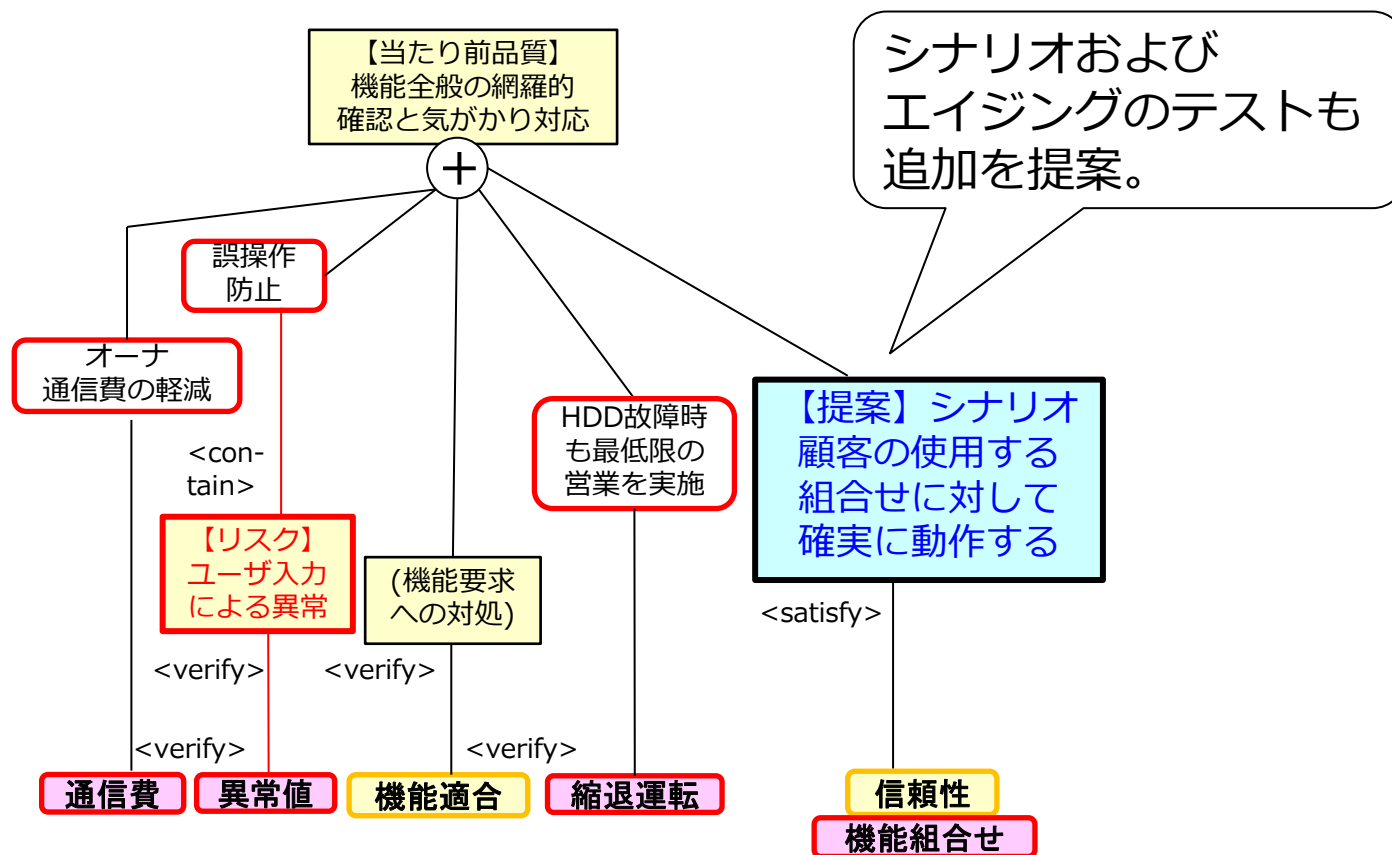
※達成マップ上の関連性で紹介



追加提案項目

「ねらい」を達成するための提案となります。

※達成マップ上の関連性で紹介

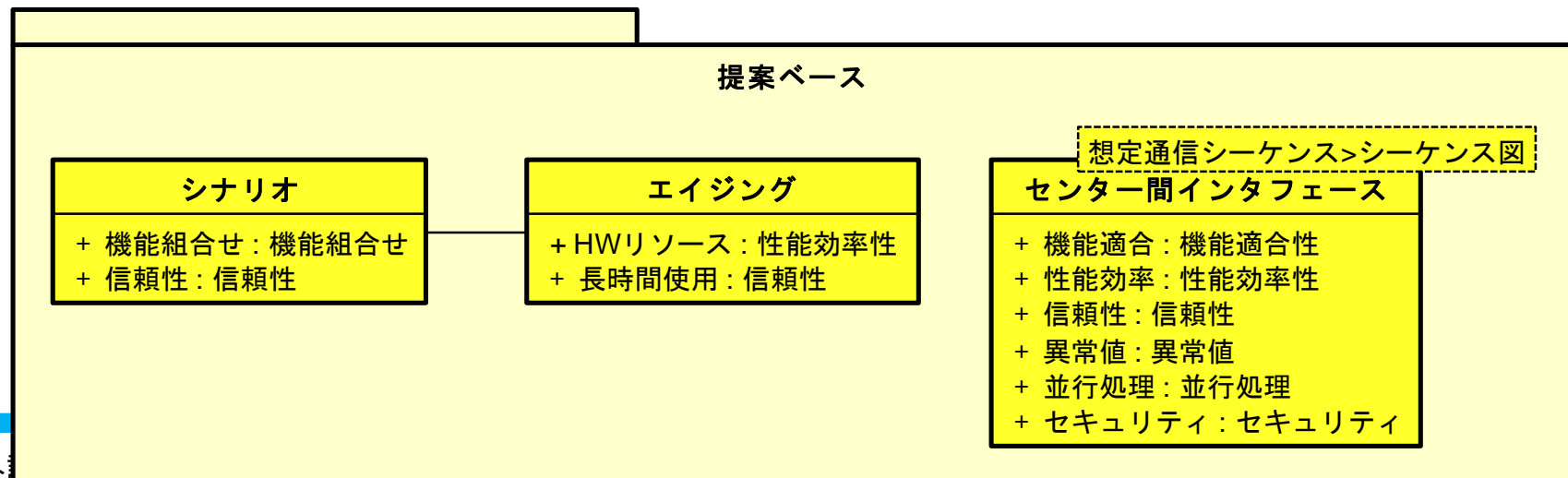


追加提案項目（3点）

追加提案するテストとして、
以下のテスト+テスト自動化を提案します。

（追加テスト）

- ・シナリオ
- ・エイジング
- ・センター間インタフェース

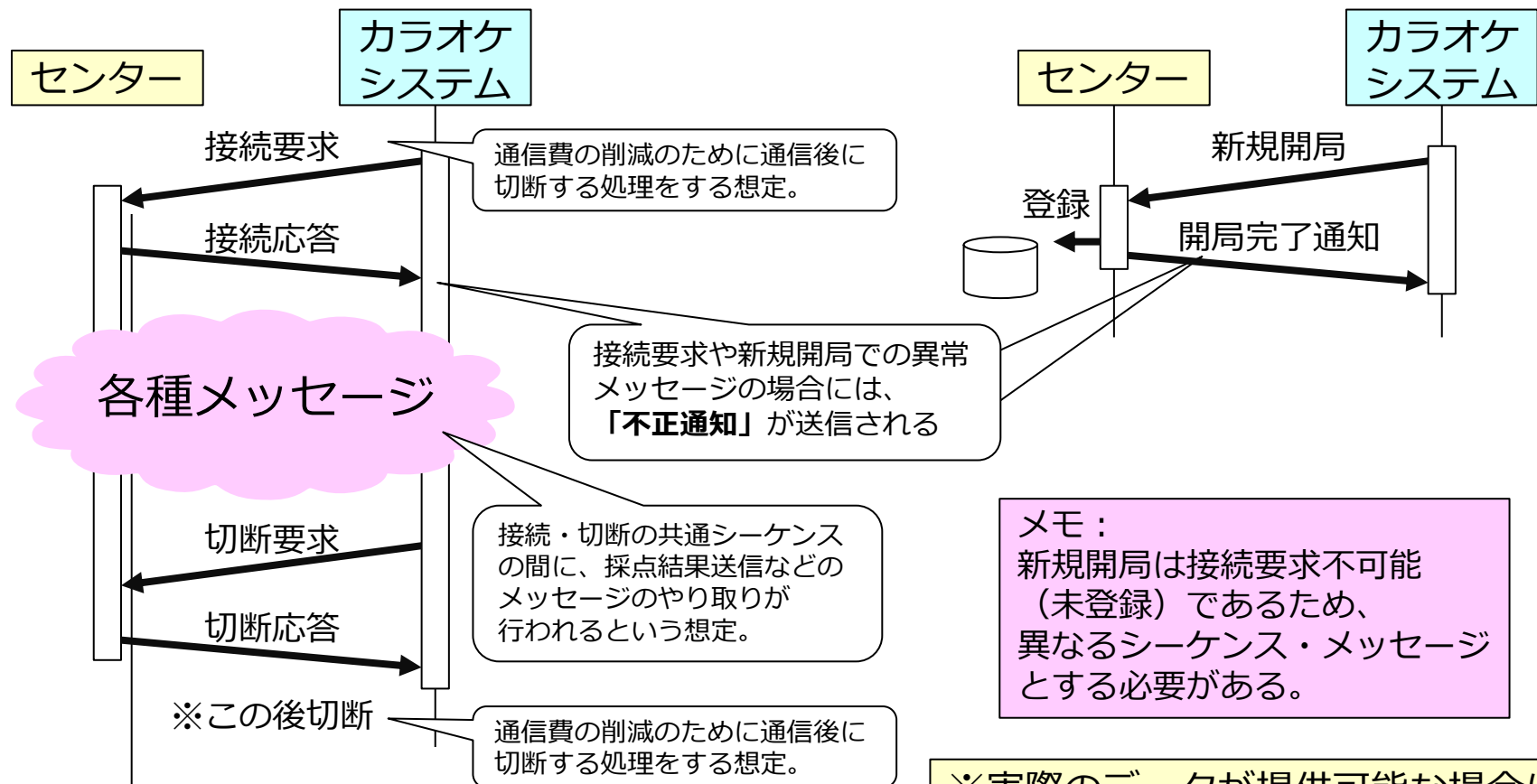


追加テスト：センター間インタフェース

想定インタフェースでテストを検討。信頼性を中心に向上を狙う。

接続・切断型共通シーケンス

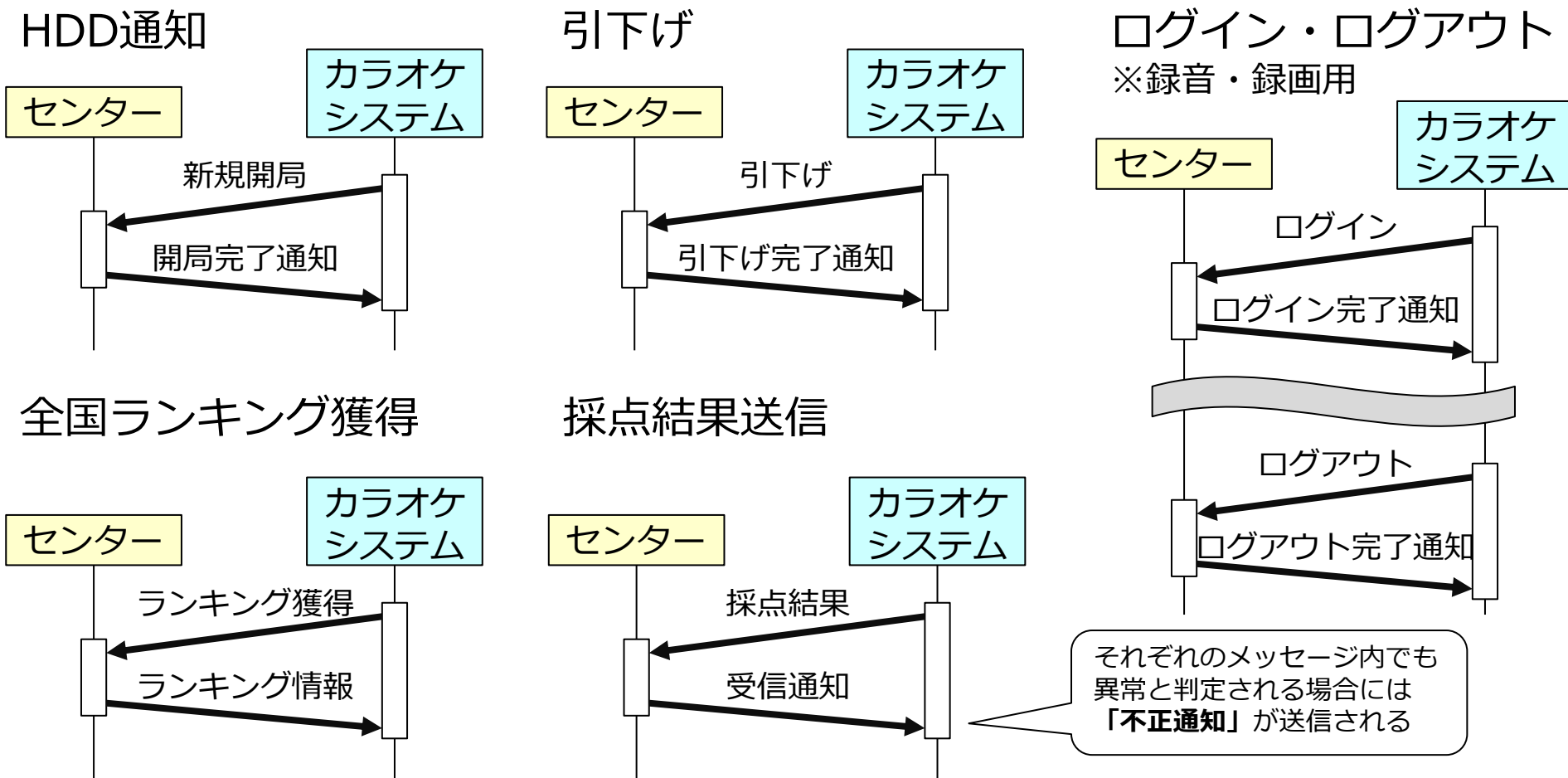
新規開局シーケンス



※実際のデータが提供可能な場合には、そのデータでテストを行うことも可能

追加テスト：センター間インタフェース

想定インタフェースでテストを検討。信頼性を中心に向上を狙う。

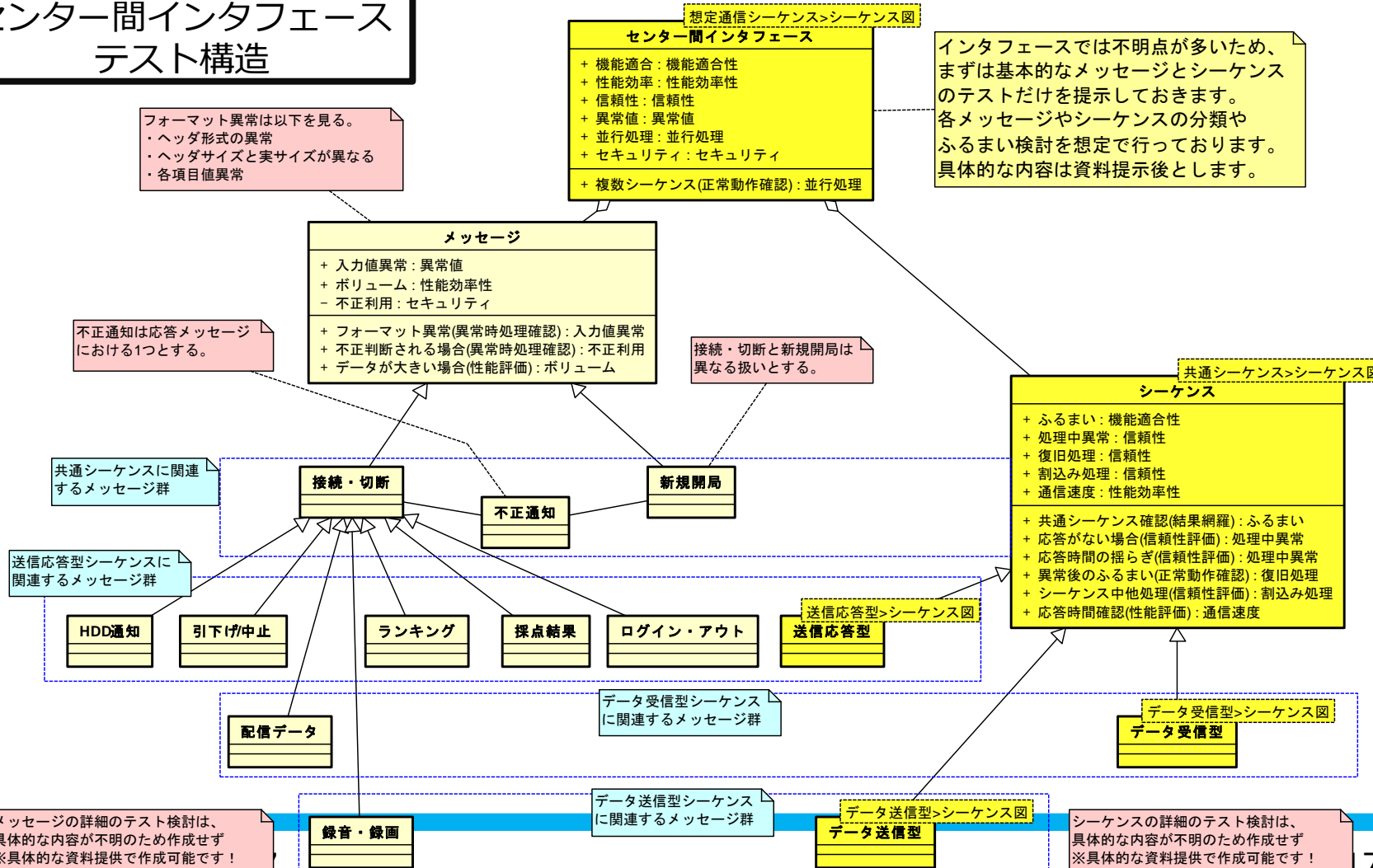


※実際のデータが提供可能な場合には、そのデータでテストを行うことも可能

追加テスト：センター間インタフェース

想定インタフェースでテストを検討。信頼性を中心に向上を狙う。

センター間インタフェース テスト構造

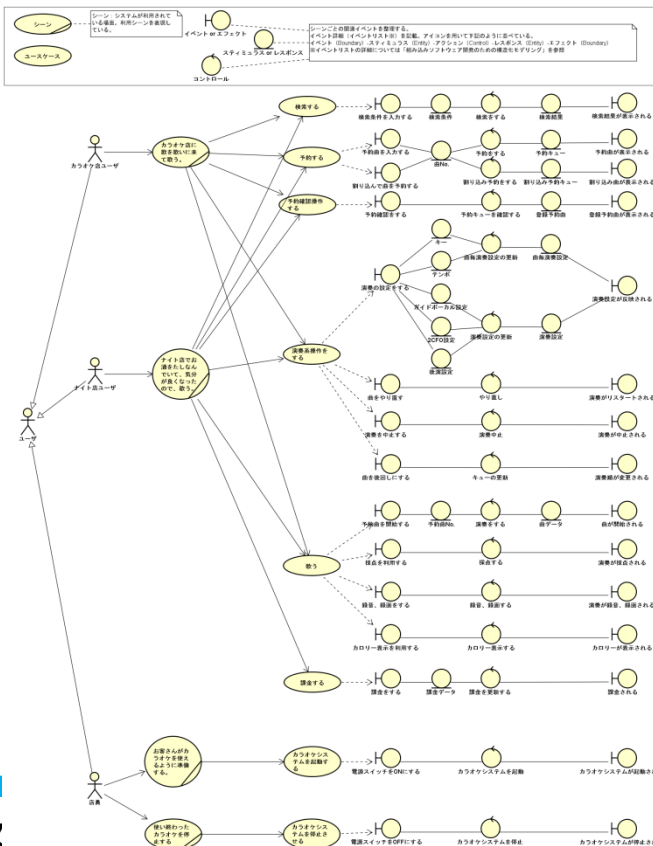


追加テスト：シナリオ・エイジング

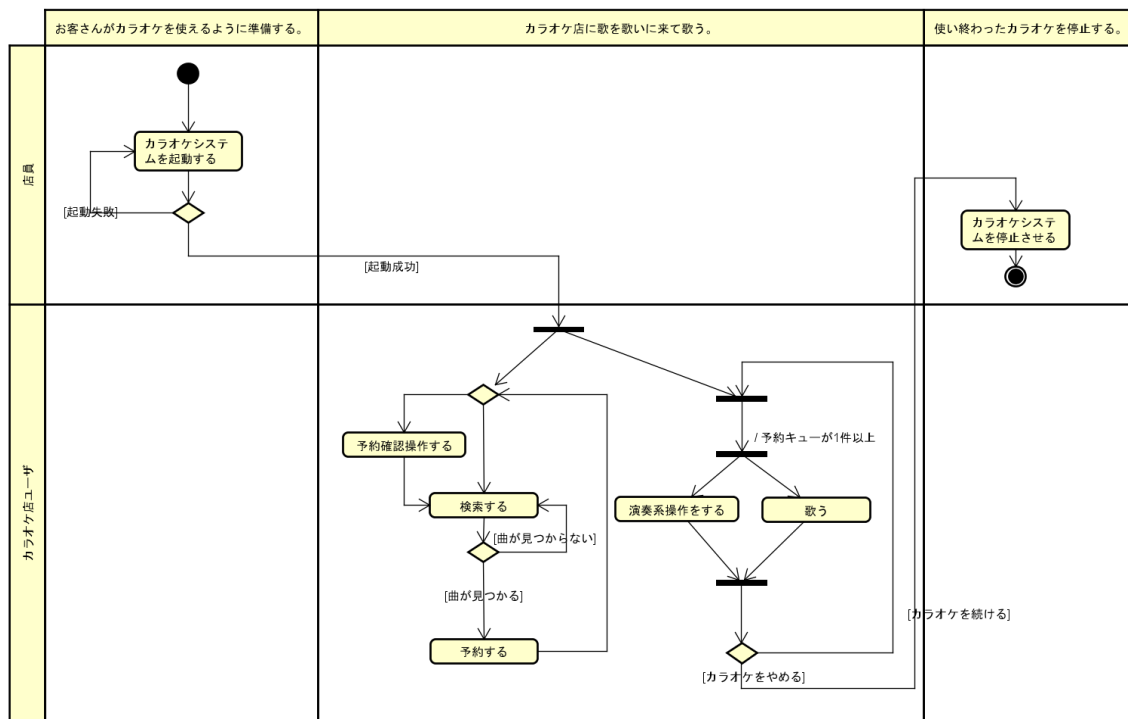


顧客のユースケース、機能に関連するイベントを割出して、機能組合せの確認や信頼性の向上を狙う。

シナリオテスト参照モデル



シナリオテスト向けアクティビティ図

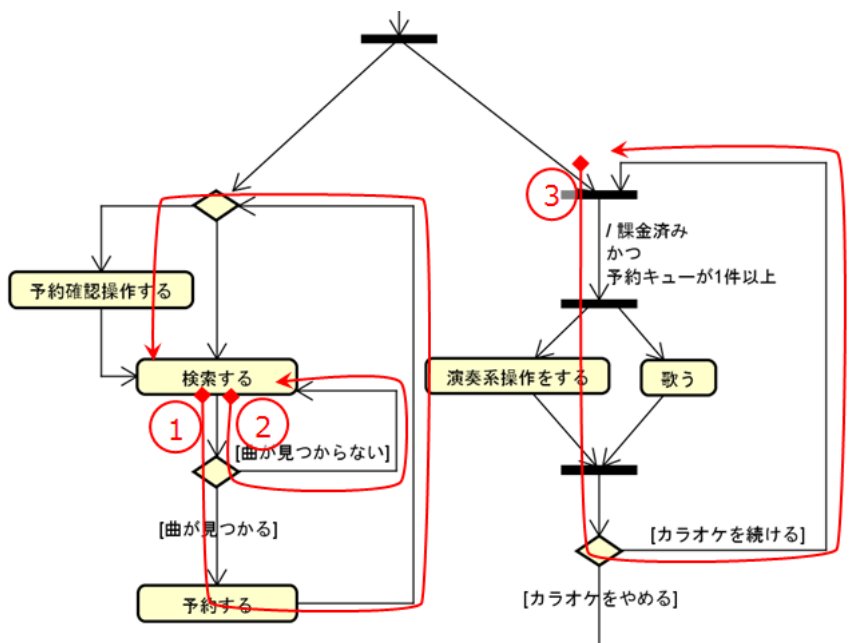


※具体的な顧客想定が提供可能な場合には、そのデータでテストを行うことも可能

追加テスト：シナリオ・エイジング

顧客のユースケース、機能に関連するイベントを割出して、機能組合せの確認や信頼性の向上を狙う。

エイジングテストシナリオ



エイジングテストシナリオ

ID 9-40_00_0101
予約し続ける

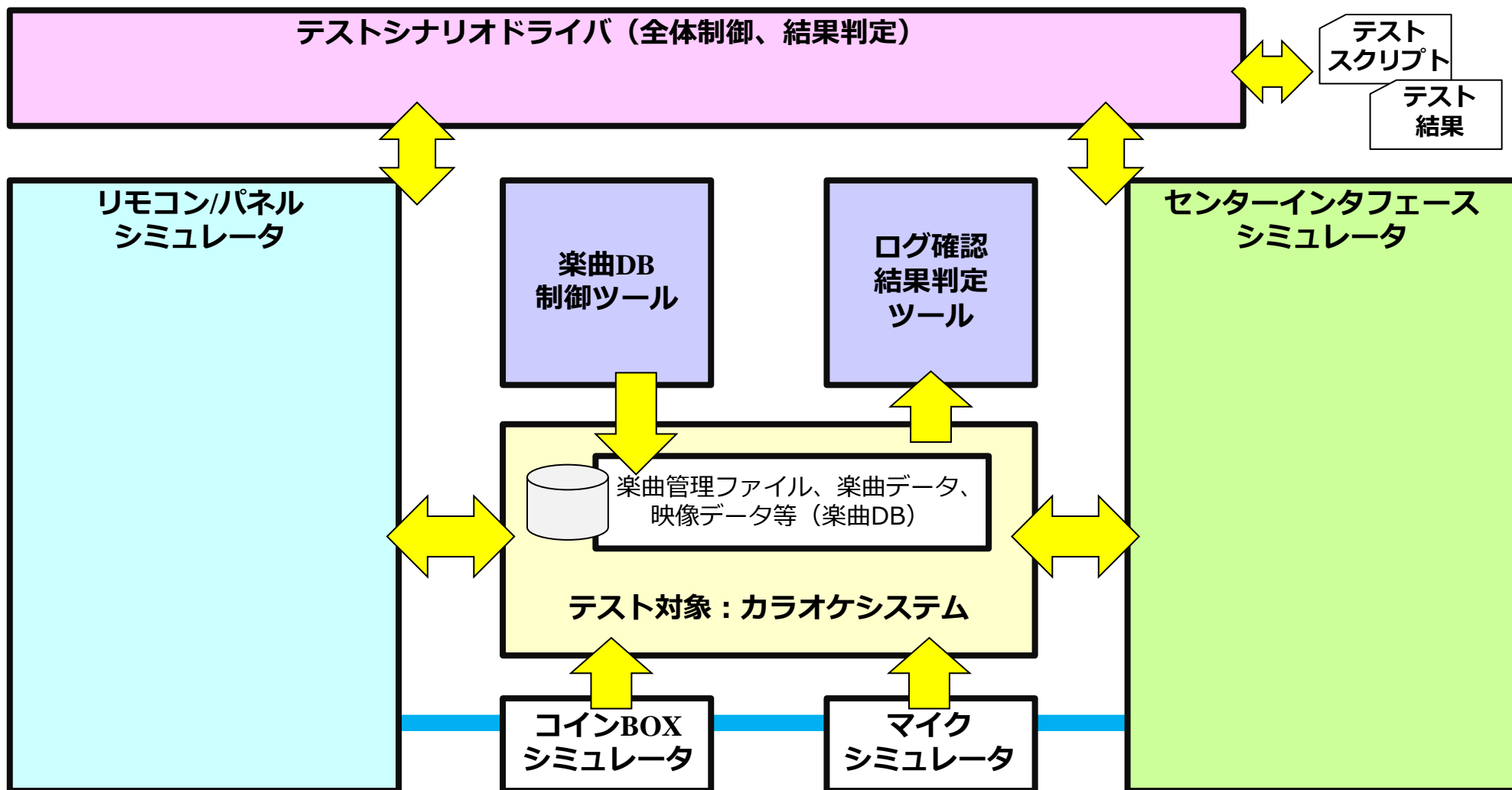
イベント	パラメータ	値
検索する	検索条件を入力する	
予約する	予約曲を入力する	通常予約
予約確認操作する	予約確認をする	

ID 9-40_00_00121
曲の後回しにより歌い続ける

イベント	パラメータ	値
歌う	予約曲を開始する	
	採点を利用する	機能OFF
	録音、録画をする	機能OFF
	カロリー表示を利用する	機能OFF
演奏系操作する	演奏の設定をする	
	曲をやり直す	やり直さない
	演奏を中止する	中止しない
	曲を後回しにする	後回しにする

提案：テスト自動化

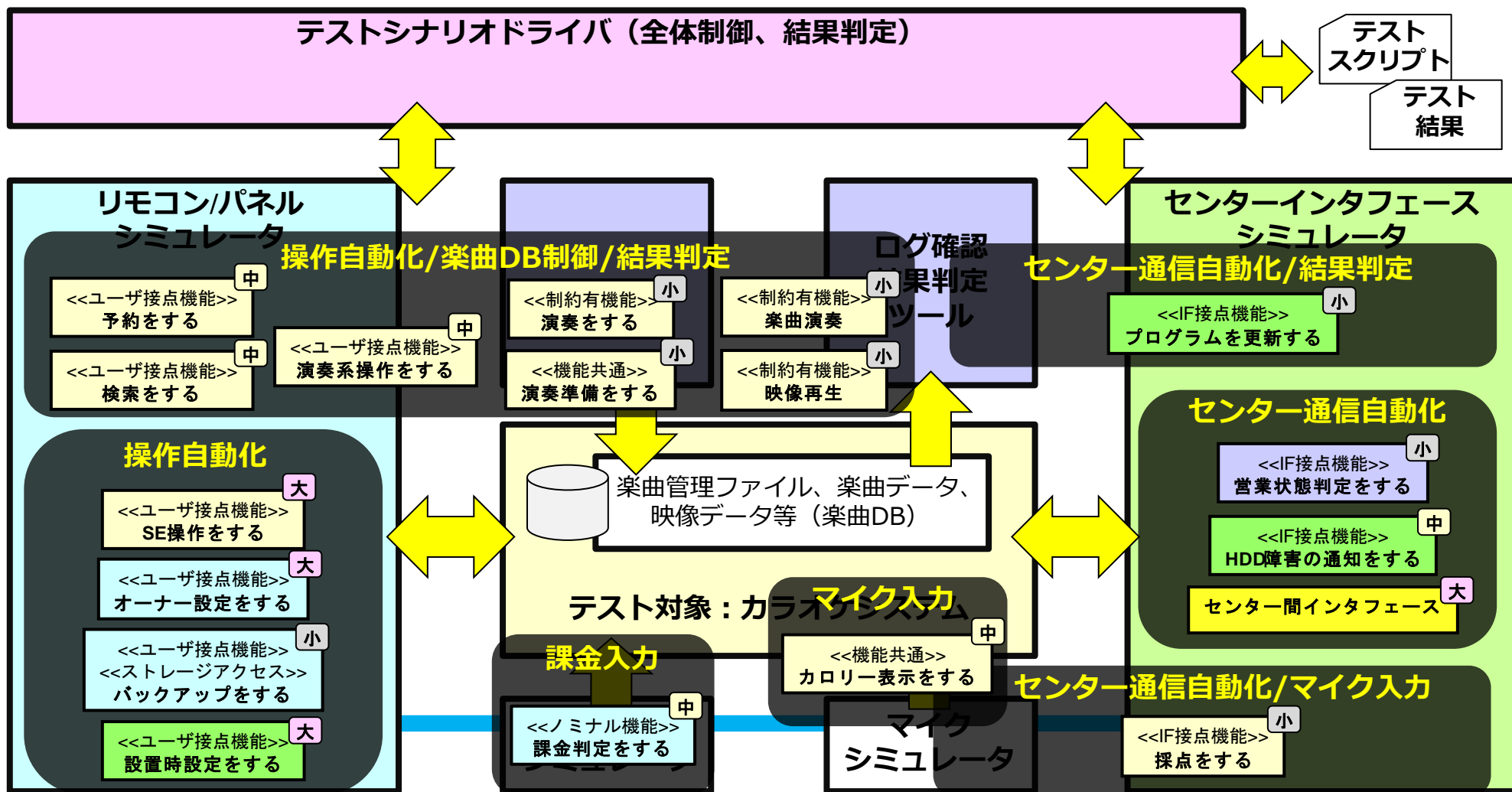
テストバリエーションと保守性の向上のためテスト自動化を提案する。
テスト自動化の仕組みを以下のように想定する。



提案：テスト自動化向け実現ビュー（テストアーキテクチャ）



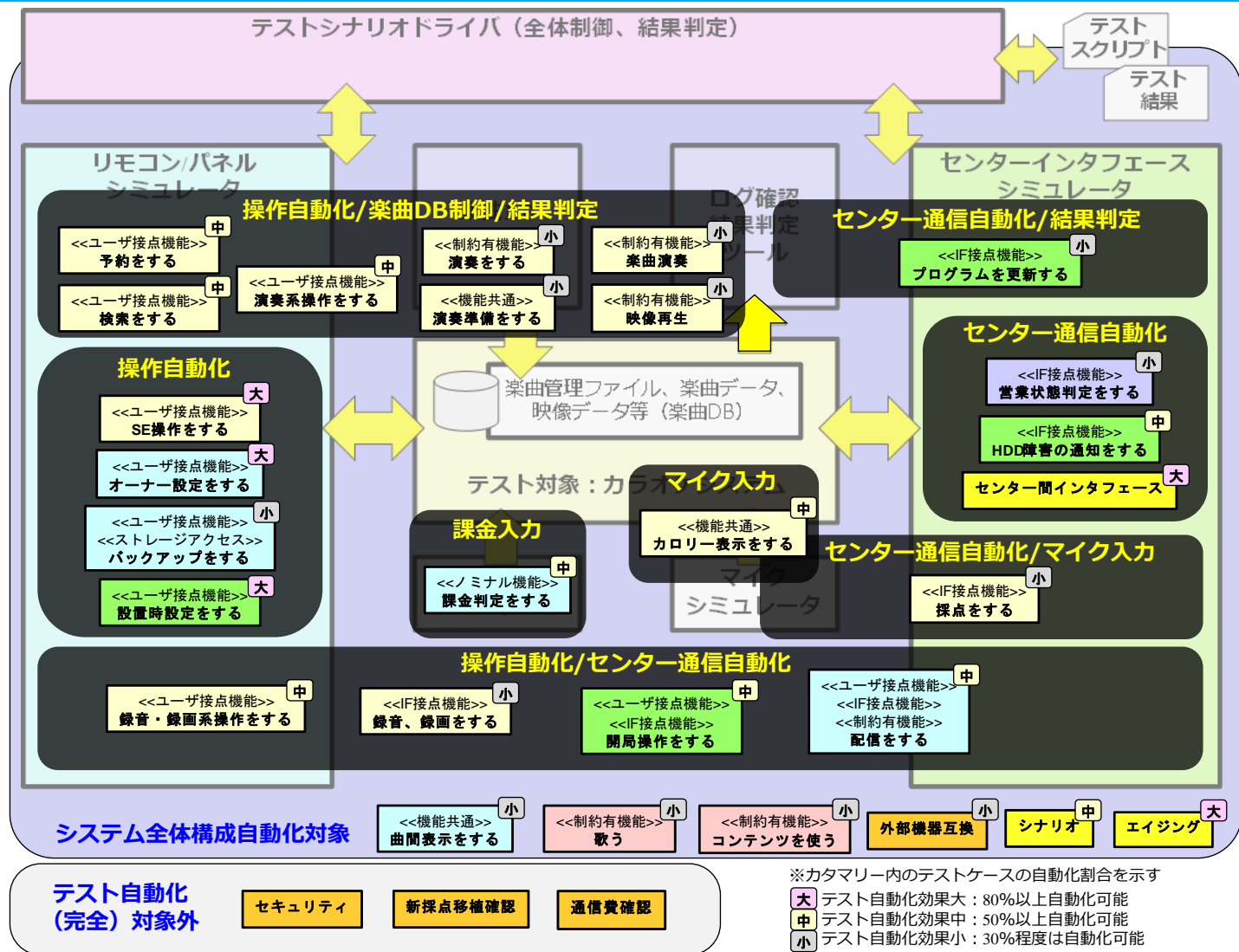
どのテスト自動化の仕組みにより自動化が出来るようになるか？
を別の「実現ビュー」として整理している。



提案：テスト自動化向け実現ビュー（テストアーキテクチャ）



「テスト自動化向け実現ビュー」を用いることで、テスト自動化による効果と必要な投資を比較し考慮しながら、テスト自動化の具体的な実現に向けた検討・調整が可能となります。



各モデルの変化を確認できるようにする@網羅ビュー



今回のテストの提案込みの全体像（網羅ビュー）。
テストの大枠として以下3つに分けています。

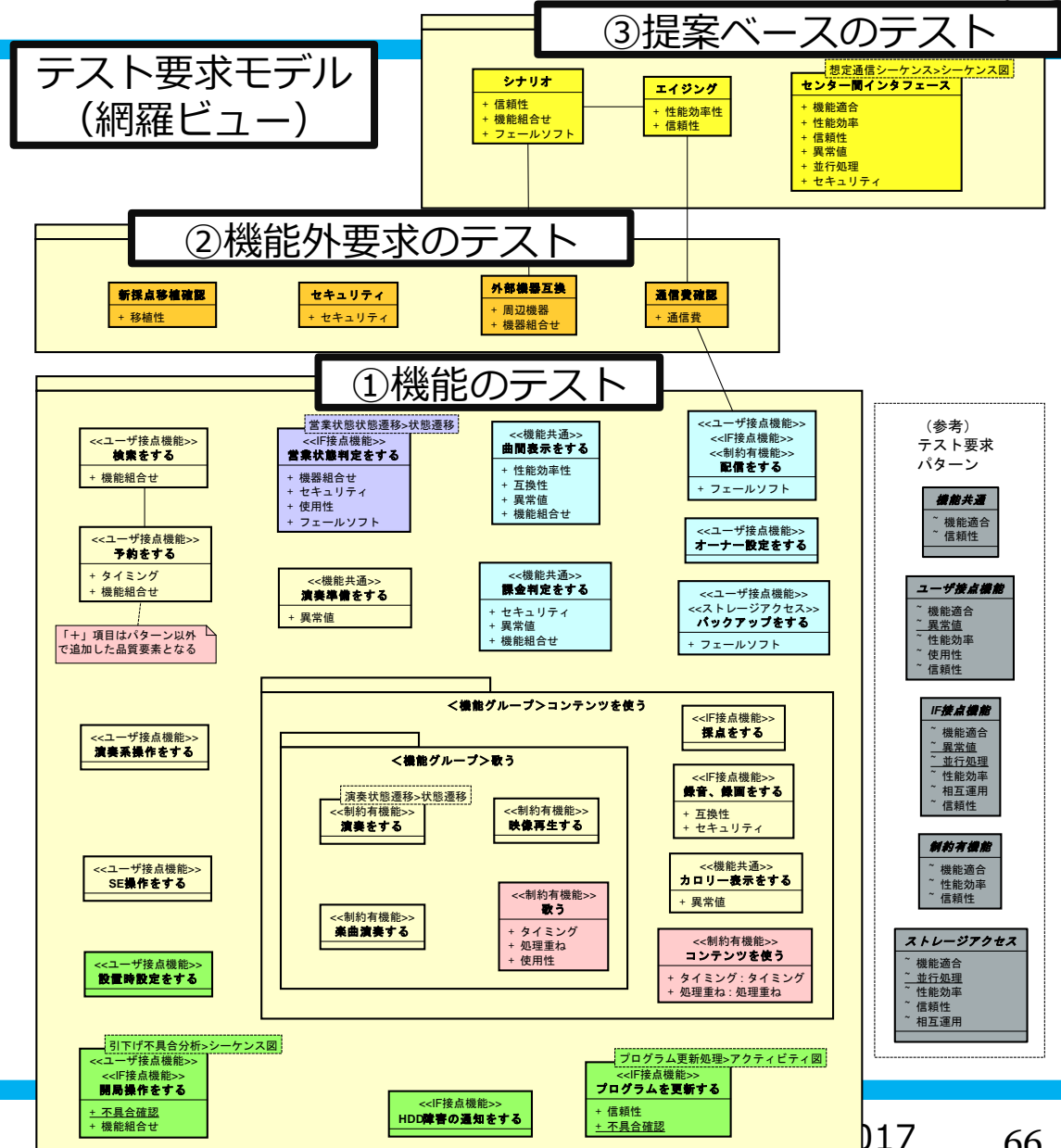
■ テストベースの網羅

① 機能のテスト：
DFD参照モデルベース

② 機能外要求のテスト：
テストベース記載の機能外以外の内容

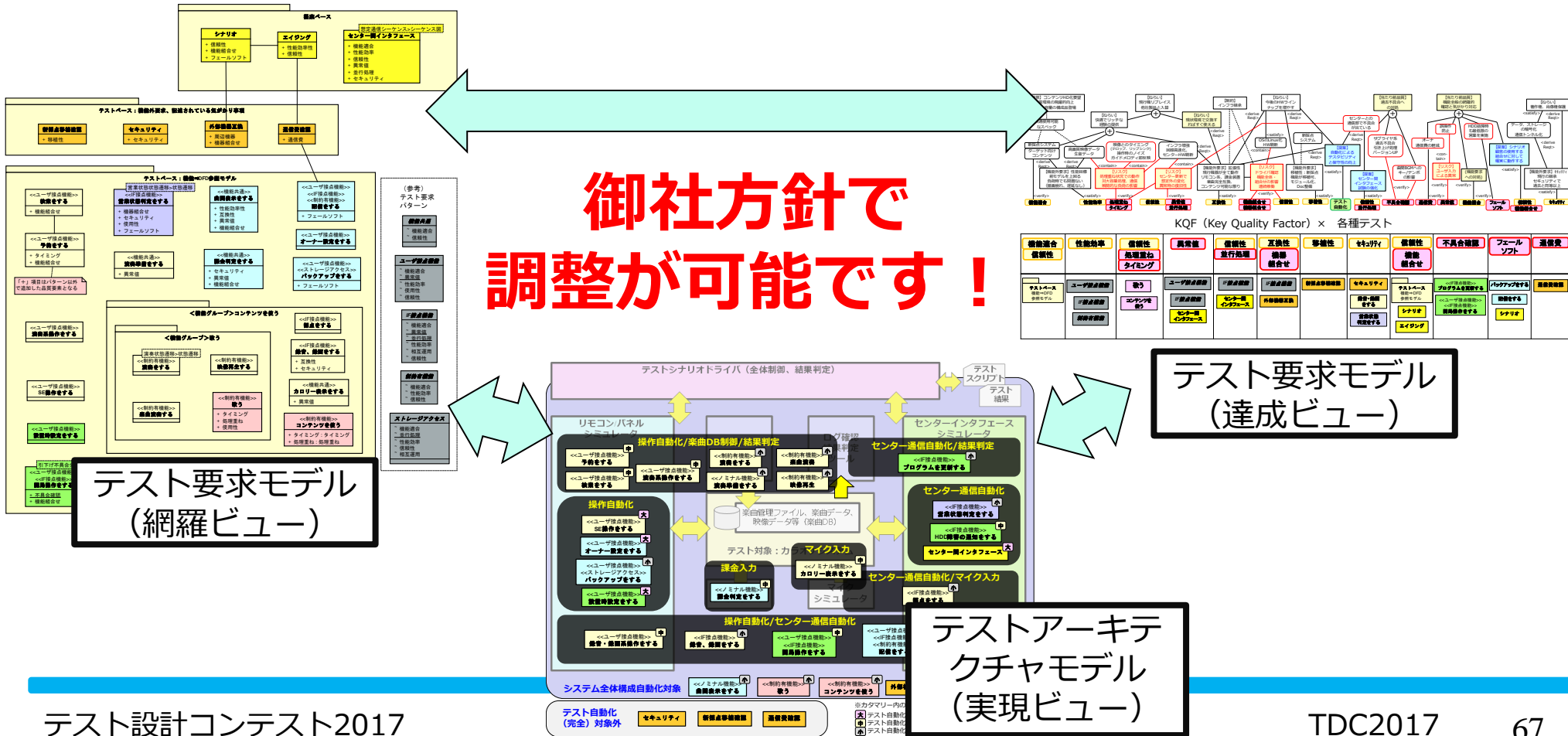
■ 提案ベースのテスト

③ 提案ベースのテスト：
ねらいを達成するために必要と考えたテストの提案



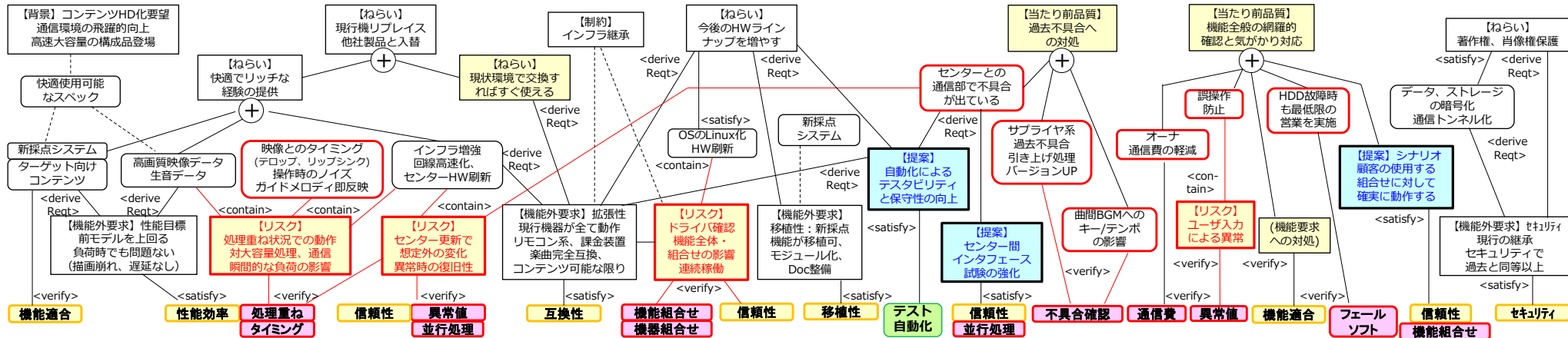
御社にあわせたテストのカスタマイズ

御社にあわせたテストのカスタマイズのための「プラン」について提案します。



カスタマイズ：紹介した内容⇒推奨プラン

提案を含めて、先ほどもで紹介した内容は、弊社における「推奨プラン」となります。



KQF (Key Quality Factor) × 各種テスト

機能適合	性能効率	信頼性	異常値	信頼性	互換性	移植性	セキュリティ	信頼性	不具合確認	フェールソフト	通信費
信頼性		処理重ね タイミング		並行処理	機器 組合せ			機能 組合せ			
テストベース 機能⇒DFD 参照モデル	ユーザ接点機能 IF接点機能 制約有機能	歌う コンテンツを 使う	ユーザ接点機能 IF接点機能 センター間 インタフェース	IF接点機能 センター間 インタフェース	IF接点機能 外部機器互換	新採点移植確認	セキュリティ 録音・録画 をする	テストベース 機能⇒DFD 参照モデル シナリオ エイジング	<<IF接点機能>> プログラムを更新する <<ユーザ接点機能>> <<IF接点機能>> 開局操作をする	バックアップをする 配信をする シナリオ	通信費確認

**推奨プラン
テスト一式**

カスタマイズ：提案するプラン

今回提案するプランを紹介します。

御社にあわせたテストの提案

プラン・選定内容	プラン内容の紹介
■メインプラン 推奨プラン	メインのプランとして選択してください。 紹介したテストの内容の具体的設計・実施までを行うことが可能です。
選択プラン CPMバリュープラン	推奨プランから予算や状況にあわせ絞り込みが出来ます。 (紹介した内容とは異なります)トレーサビリティが高く、兎に角お安く出来ます。※品質はお客様次第です。
■+サイドプラン 提案テスト 追加プラン	ご希望で追加可能なテストの提案となります。 選択プランに追加する形式になります。センター間インタフェース、シナリオ、エイジングという推奨テストの実施をご希望により追加が可能です。
テスト自動化プラン コンサルプラン	テスタビリティ向上と今後の保守性向上へ、テスト自動化の仕組みを構築します。段階的もしくは一部のみ構築も可能です。 御社の過去のテスト資産をもとにテストを構築するための支援を行うプランとなります。

御社に最適の テストを提供 ～STUDIO IBURI～

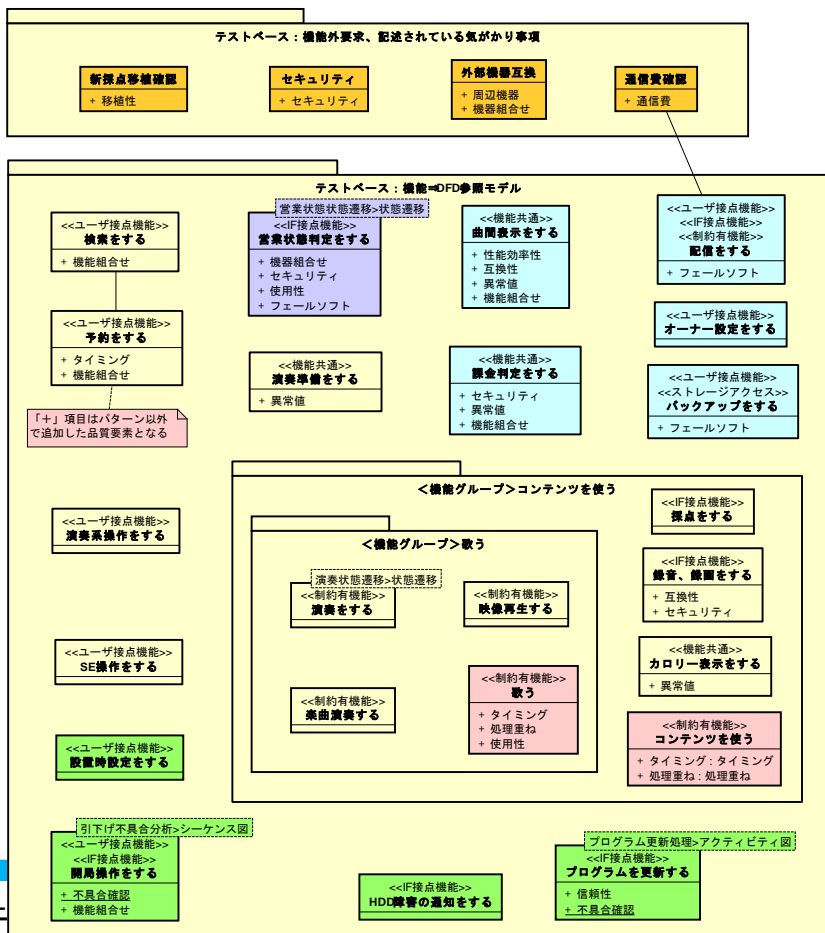


VSTeP	VSTeP -Qualab http://qualab.jp/vstep/
ゆもつよメソッド	ソフトウェアテストPRESS vol10「今こそ聞きたい テストの上流設計」
参加要項	テスト設計コンテスト'17 OPENクラス 参加要項・審査基準 http://aster.or.jp/business/contest/rulebook.html
PFD	PFD(Process Flow Diagram) の書き方 第3版 2009/9/21 http://kohablog.cocolog-nifty.com/blog/pfd.html
ISO/IEC25010 SQuaRE	ISO/IEC 25000 series : System and software product Quality Requirements and Evaluation(SQuaRE) 25010 : Quality Model
テスト要求パターン 関連論文	K. Yoshioka et al. "Stepwise Test Design Method", 6th World Congress for Software Quality (6WCSQ), September 2014
要求図	システムエンジニアリングでSysML を使いこなす https://www.ogis-ri.co.jp/otc/hiroba/technical/SysEngSysML/SysEngSysML2.html
テストカタマリー 記法紹介	どしろうと製作所：テストカタマリーの紹介 http://blog.amateur-factory.jp/?eid=1444276

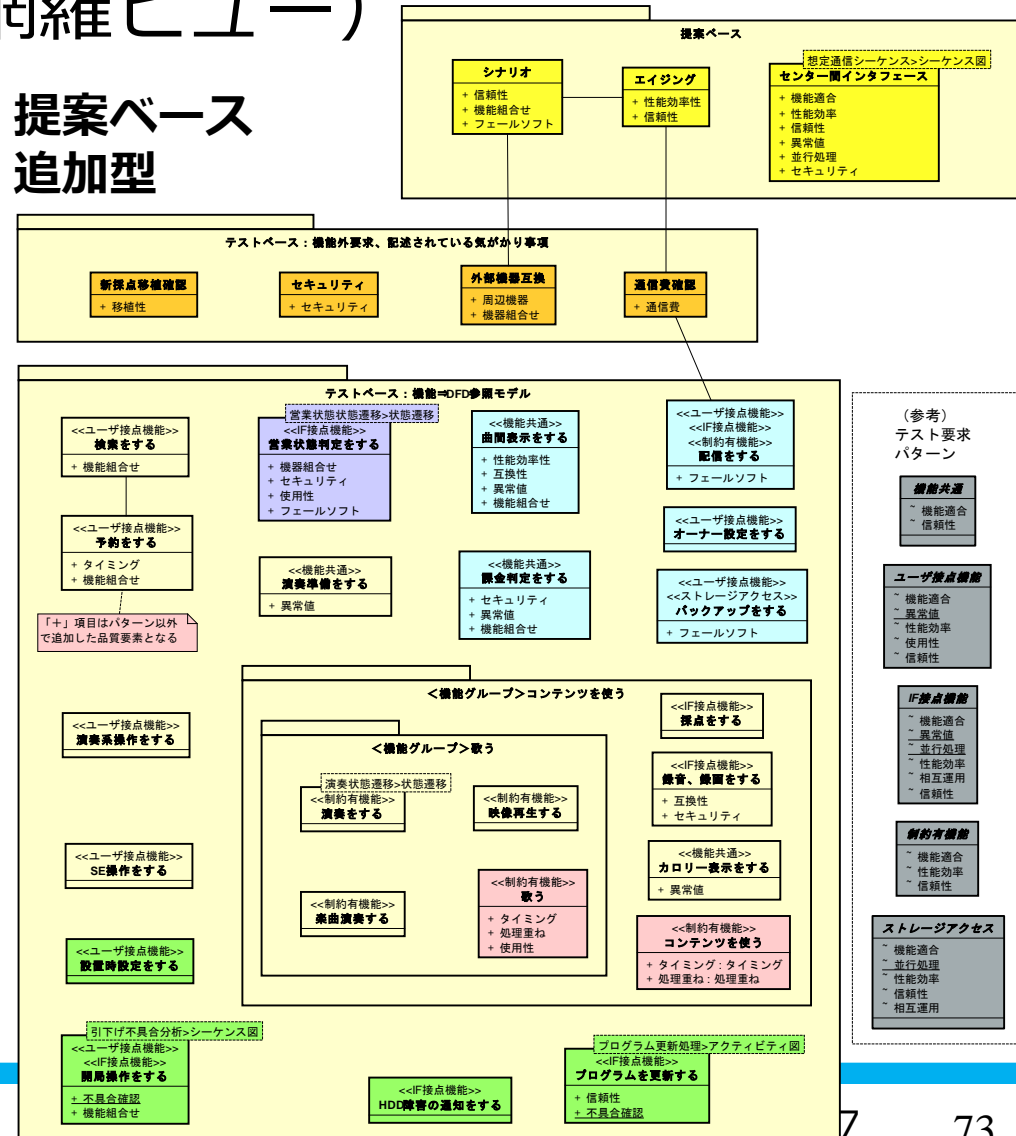
参考：プランによるモデルの変化

テストカタマリー全体図（網羅ビュー）

テストベース完全準拠型



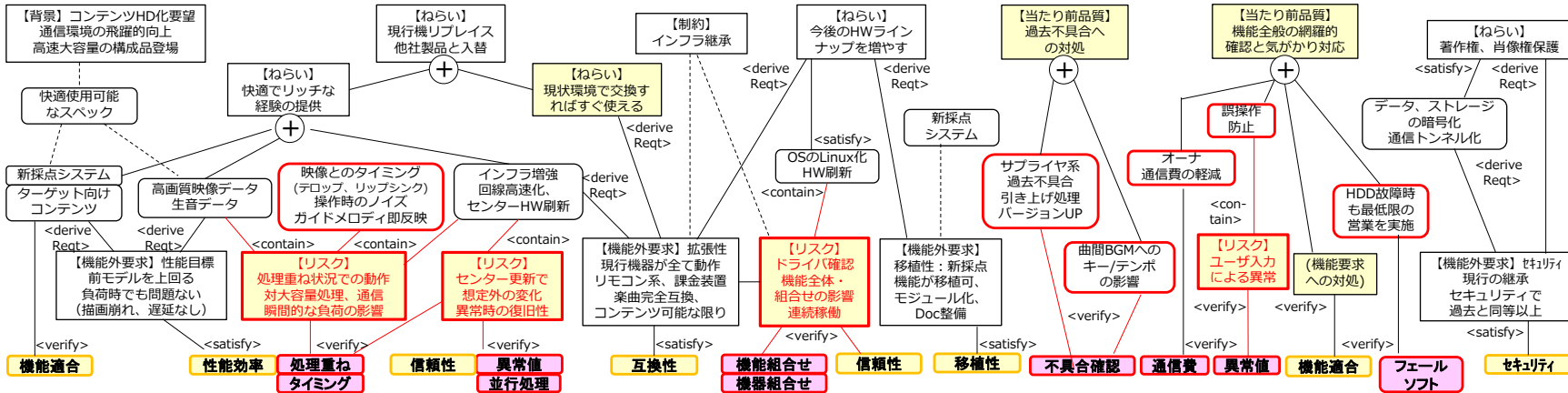
提案ベース追加型



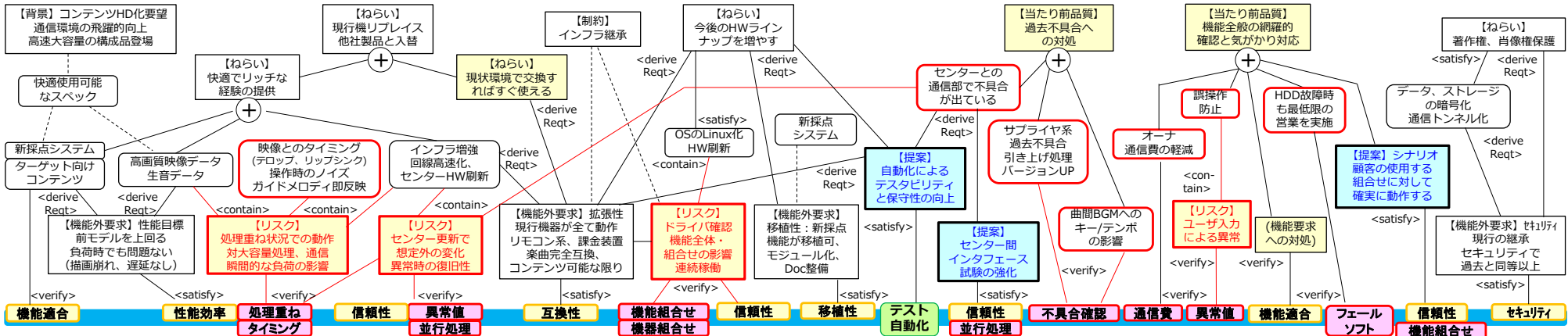
参考：プランによるモデルの変化

テスト達成マップ（達成ビュー）

テストベース完全準拠型



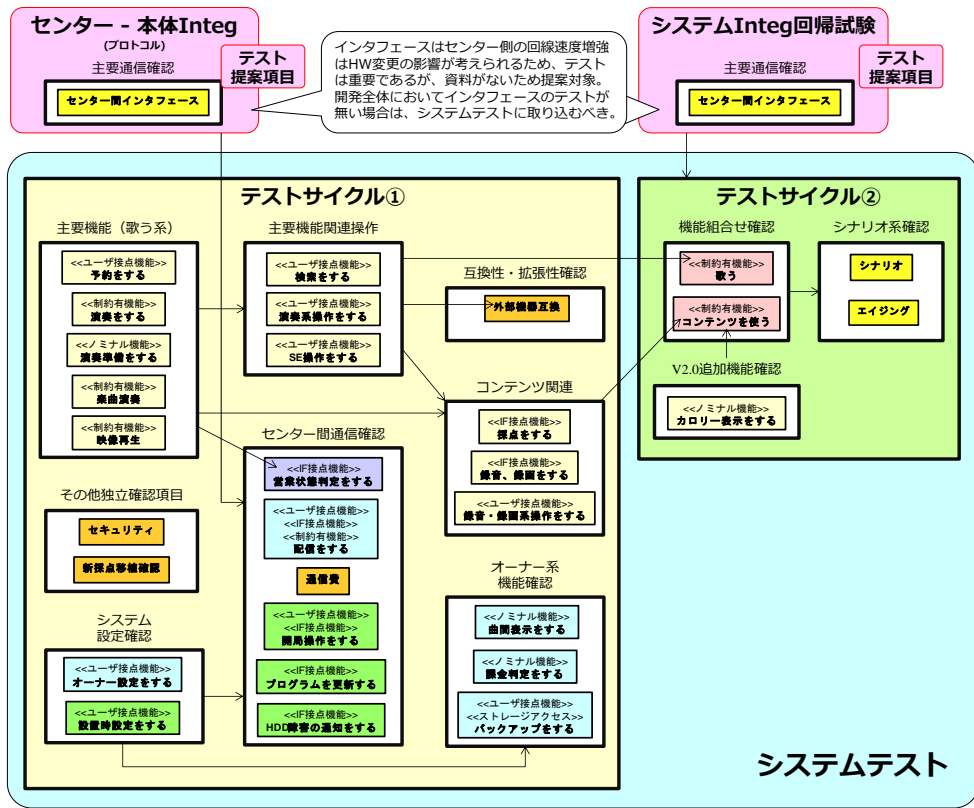
提案ベース追加型



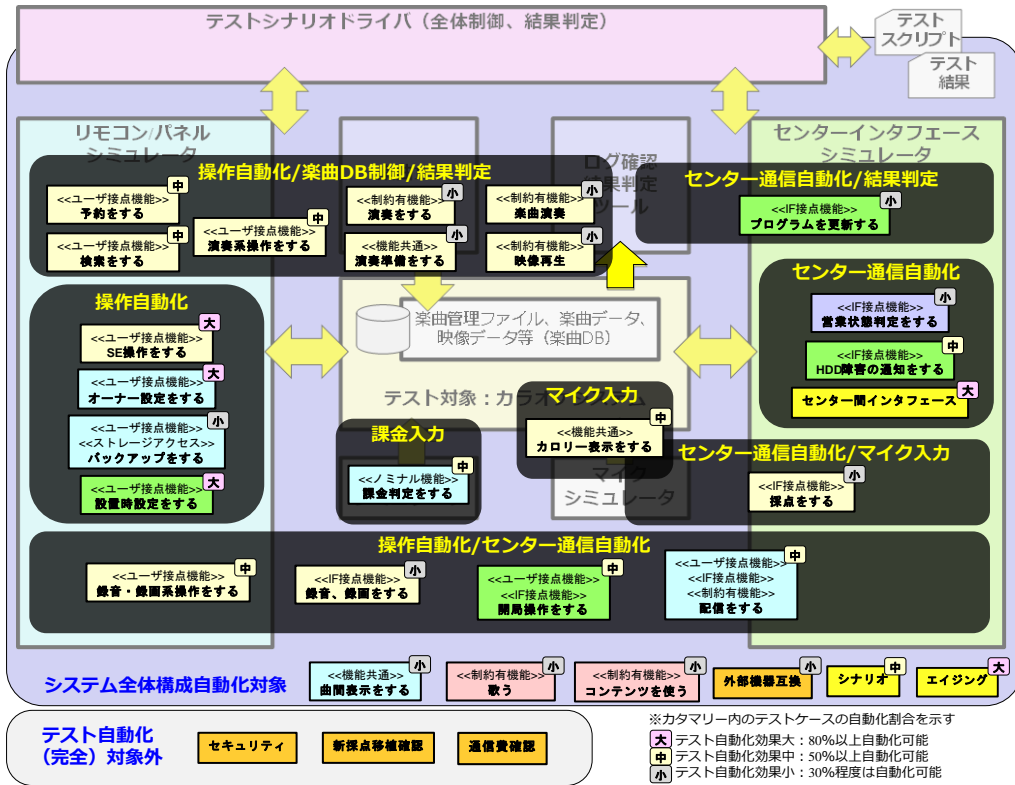
参考：プランによるモデルの変化

テストアーキテクチャ（達成ビュー）

順序型実現ビュー



テスト自動化実現ビュー



参考：記法とプロセスの説明は以下でまとめています



テストカタマリーの紹介：まとめ（各種記事へリンクがあります）

<http://blog.amateur-factory.jp/?eid=1444276>

テストカタマリーを活用したテスト設計プロセス案：まとめ

<http://blog.amateur-factory.jp/?eid=1444278>

※2017/02中に全体記事を更新する予定です。

