

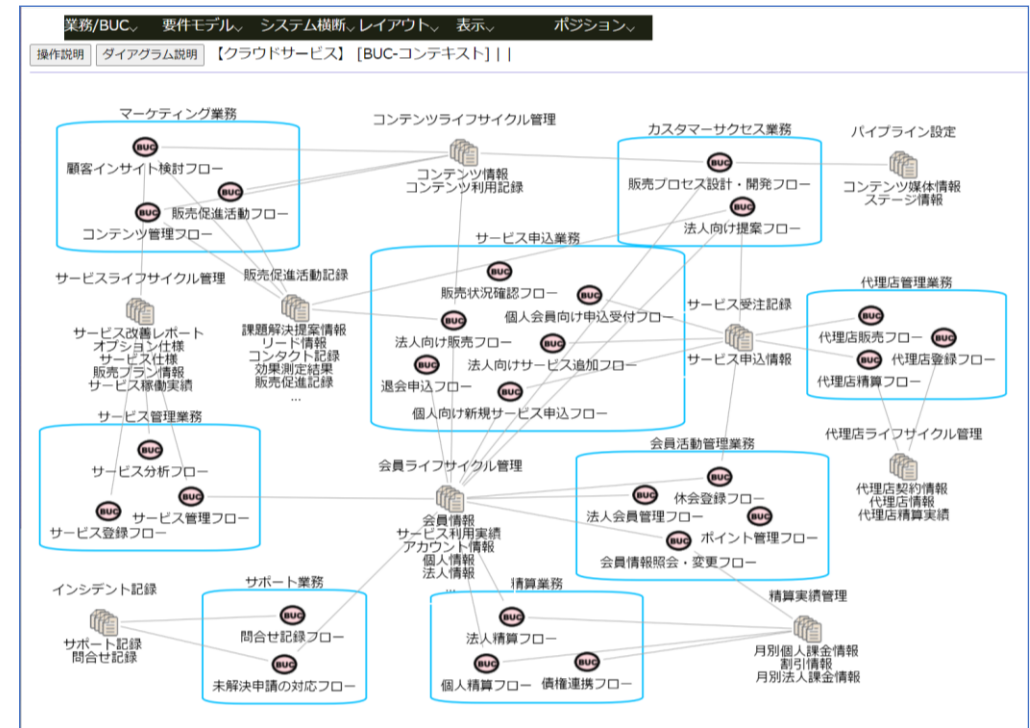


要件定義支援 サービス

バリューソースの要件定義の特色

- バリューソースの要件定義支援はRDRAという手法を使い、素早く精度の高い要件を定義します
- そのためにツールを使い、機械的に不整合を検知し、ビジュアルに要件を分析します
- 手法とツールを使うことで1 2回の打合せで要件を定義することが可能になります

1	業務	BUC	先	アクティビティ	次	UC	関連モ	関連オブジェクト	関連モデ	関連オブジェク	
2	サービス管理業務	サービス登録フロー		サービス仕様登録		↓				アクター	サービス企画
3											サービス企画
4											サービス企画
5											サービス企画
6											サービス企画
7											サービス企画
8											サービス企画
9											サービス企画
10											サービス企画
11											サービス企画
12											サービス企画
13											サービス企画
14											サービス企画
15											サービス企画
16											サービス企画
17											サービス企画
18											サービス企画
19											サービス企画
20											サービス企画
21											サービス企画
22											サービス企画
23											サービス企画
24	マーケティング業務	コンテンツ管理フロー		コンテンツ作成		↓				アクター	カスタマーサクセス
25											カスタマーサクセス
26											カスタマーサクセス
27											カスタマーサクセス



1. RDRA推進手順

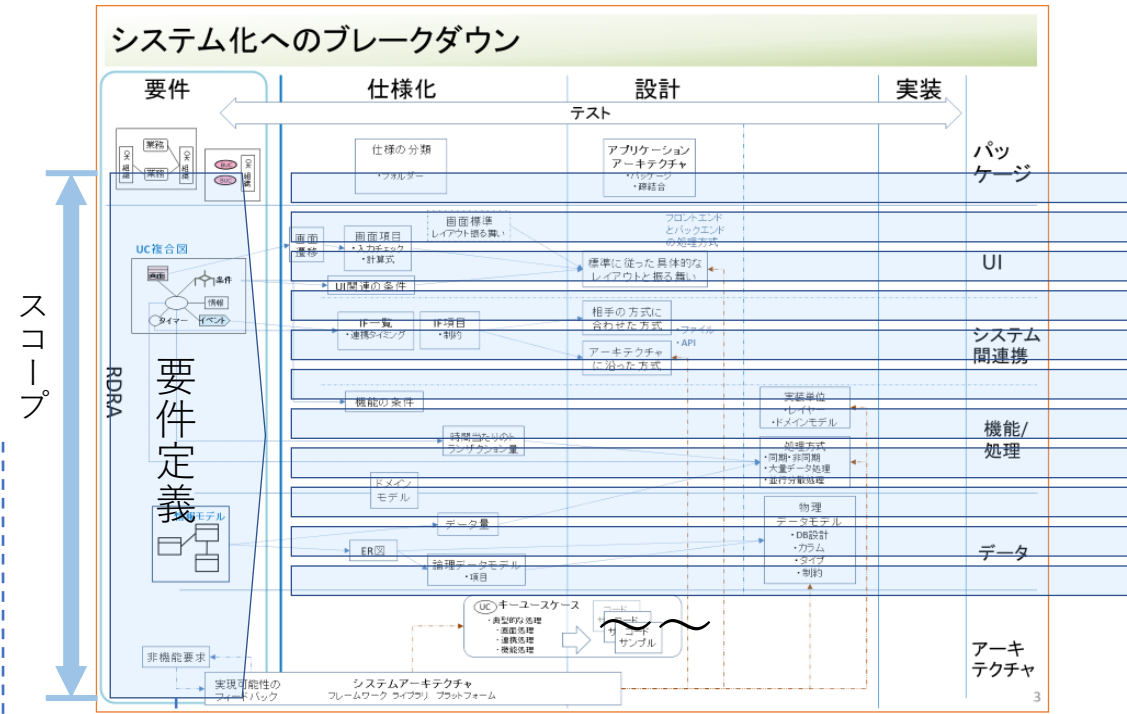
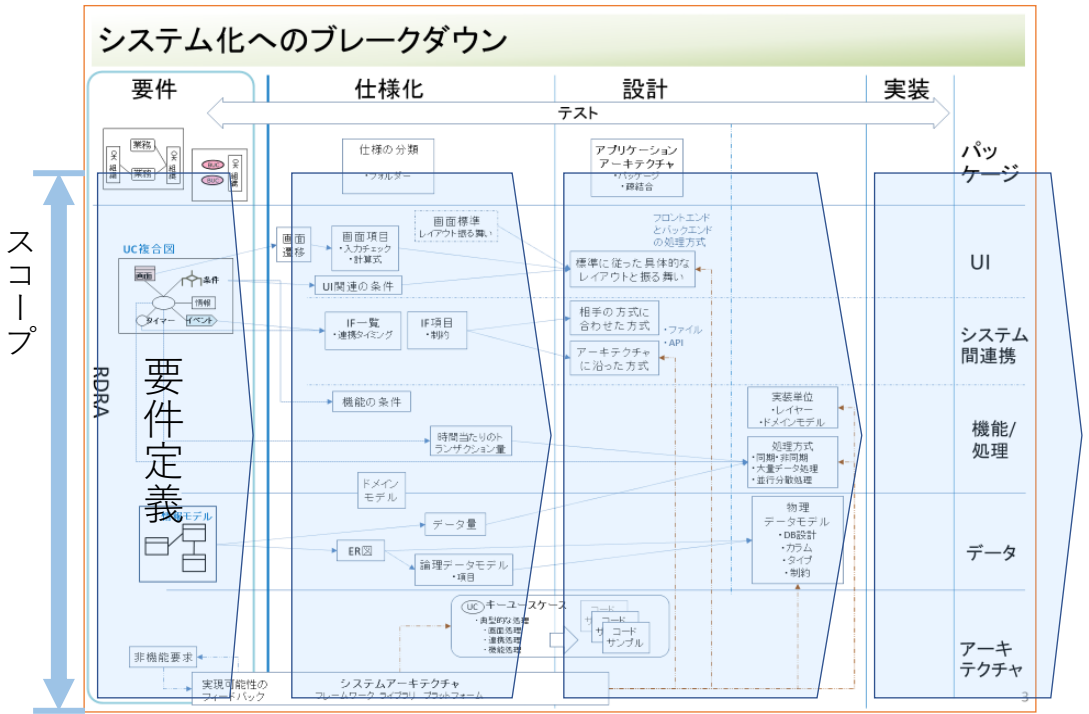
RDRA導入シナリオ

- RDRAモデリングのスタンス
 - RDRAモデルをドキュメント作成活動とは考えずに、プロジェクトメンバーのコミュニケーションを促進するための活動ととらえる
 - コミュニケーションを円滑にするために有効な形で組み立てる
 - 可視化の手順も議論が促進するように組み立てられている
- 可視化手順
 - フェーズ1：議論のベースを作る
 - 議論が空中戦にならないように、ユースケースと情報、状態を粗いレベルで洗い出す
 - 4ステップで枠組みを洗い出す
 - Step1：業務、BUCを洗い出す
 - Step2：スコープの把握
 - Step3：情報モデルを作成する
 - Step4：状態モデルを作成する
 - フェーズ2：要件の形を作る
 - ユースケースに情報、条件、イベントをつなげて要件を組み立てる
 - 各々のユースケースが画面、イベントなどの入出力を使って、情報を更新し状態を変化させていく。その関係性を使ってシステムが行うことを組み立てていく
 - 2ステップで要件を組み立てる
 - Step1：トップダウンで組み立てる
 - Step2：情報・状態から見直す
 - フェーズ3：ビジネスルールで仕様化可能にする
 - ビジネスルールを表すバリエーションと・条件を洗い出す
 - ユースケース個別に条件を洗い出す
 - 3ステップで洗い出す
 - Step1:トップダウンでバリエーションを整理
 - Step2:ボトムアップでバリエーションを整理
 - Step3:要素の洗い出し

RDRAのプロセス上での位置づけ

ウォーターフォール

繰り返し開発



満たすべき水準が分かるまで並行作業しない

本番リリースするために満たすべき水準（整合性を確保する）が分かる範囲をスコープとする

要件定義の状況を把握する

- 要件定義は混とんとした状況の中で、段階的に決めることが求められる
- 段階的に決めても新たに不明・問題が浮かび上がる
- 上記状況の様子と段階的な決定に向けた状況認識を以下の表で整理
- 下記の状況認識のもとフェーズ1の各ステップを定義した

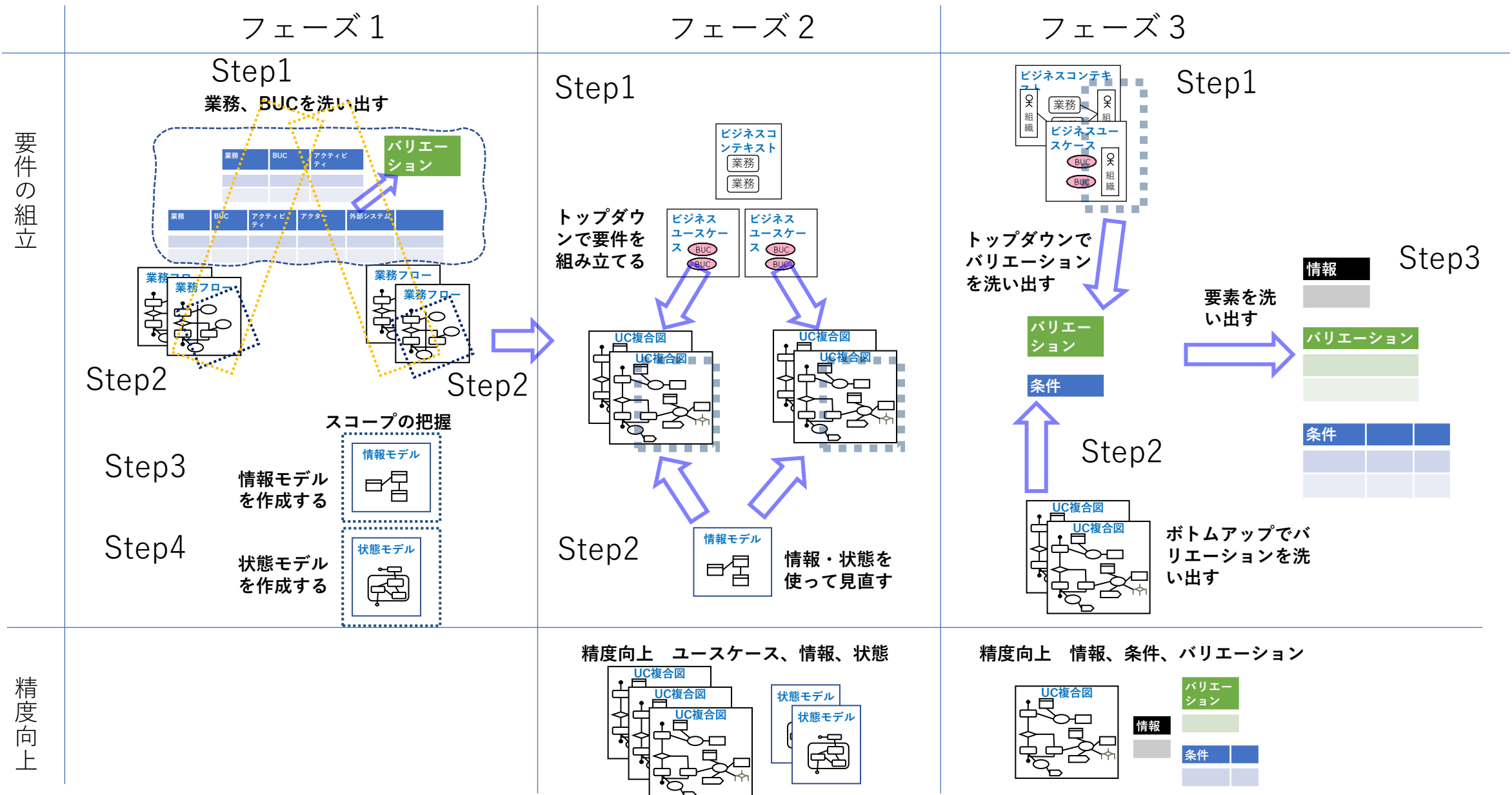
段階的な問題点	状況認識
混沌期	議論のベースができていないので、空中戦になり話し合いが記録されることもなく、無駄な議論が続く
拡散期	システムのスコープがぶれるので、議論が拡散しがちになる
システム化スコープの混沌期	業務のスコープが見えてきても、システムが何を行えばいいかが見えないので、システムスコープの議論がぶれる
システムの核が見えない	システム化の範囲が見えてきたが、システムの整合性をいかに合わすかが見えない
ビジネスを駆動するルールが見えない	システム化の対象とそこで扱う情報や状態を把握できたが、システムの駆動原理が見えていない
システムの一貫性ができてない	新しいシステムのもと整合性のある業務、システムを組み立てが必要になる 誰も答えのない中での組み立てが求められる
仕様化とのつながりが見えない	仕様化につながるビジネスルールが整理されていないので、要件として仕様化の方向性を決める必要がある
ゴールを決める段階になる	次工程につなげるための精度を求められる 最終的な整合性の落としどころを決め、次につなげる

フェーズ別組み立て

- 各フェーズを実施するときの状況と、その結果の状況を示すことで、各フェーズの意味を理解する

	Step	状況認識	作業結果の状況
議論のベースを作る	1：業務、BUCを洗い出す	最初の段階ではプロジェクトメンバー間の合意事項はなく、表現しようとしているシステムのスコープも定まっていない ⇒まずは議論可能なベースを作る	業務、BUCが可視化されることで、システム化対象の業務スコープが明らかになり、議論の対象を指し示せるようになる
	2：スコープの把握	業務の分類（業務、ビジネスユースケース、アクティビティ）が見えてきたところで、システム境界を検討する。 ⇒業務をベースにすることでシステム境界（ユースケース）の議論がブレズに進めることができる	ユースケースが可視化されることで、システム化スコープが見えてくる。 既存の入出力の位置づけも明確にできる
	3：情報モデルを作成する	ユースケースでシステム境界が見えているが、まだ具体的な処理イメージがわいていない状況である。 ⇒扱う情報を洗い出すことでシステムが行うことをイメージする	ユースケースと情報が見えてくることで、システムが行うべきことが見えてくる
	4：状態モデルを作成する	情報が見えているが、それらの情報をどのように活用するかが見えていない状況である。 ⇒情報がとりえる状態を明らかにすることで、システムが実現すべきことが見えてくる	仕事は状態を認識しながら進めるので、その状態が見えることで、システムと仕事の関係性が見えてくる
要件の形を作る	1：トップダウンで要件を組み立てる	ユースケースでシステム境界が見えるので、そこに入出力（画面、イベント）と操作する情報を結びつけ、場合分けがある場合に条件によって明示する。 ⇒この作業を繰り返すことでシステム化すべき要件が明らかにする	システム化スコープが明らかになり、一通りの要件が見えるようになる
	2：情報・状態を使って見直す	情報のライフサイクルから関わるユースケースを見直す ユースケースを状態遷移に対応付け抜けているユースケースをみなおす	ユースケースの精度が向上することでモデル全体の精度が向上する
ビジネスルールの明示	1：トップダウンでバリエーションを洗い出す	ユースケース、情報、状態は見えてきたが、ビジネスルールは明らかになっていない状態 ⇒ビジネスコンテキスト、ビジネスユースケースのビジネス要素を明らかにし、そこからバリエーションを導き出す	ビジネスルールを構成する軸となるバリエーションが見えてきた
	2：ボトムアップでバリエーションを洗い出す	ユースケースが明らかになることで、システムとの接点が明確になり、そこでの入出力と情報が見えてきたが、個々のユースケース独自の条件は見えていない状態 ⇒ユースケース毎の条件とバリエーションを洗い出す	ユースケースに関わる条件も見え、細かいビジネスルールが見えてきた
	3：要素の洗い出し	情報やバリエーション、条件が洗い出されているが、具体的な要素までは見えていない ⇒洗い出されたバリエーション、条件の要素を洗い出す。	バリエーション、条件の要素も見え、仕様化につながる粒度まで見えてきた

RDRA推進手順イメージ

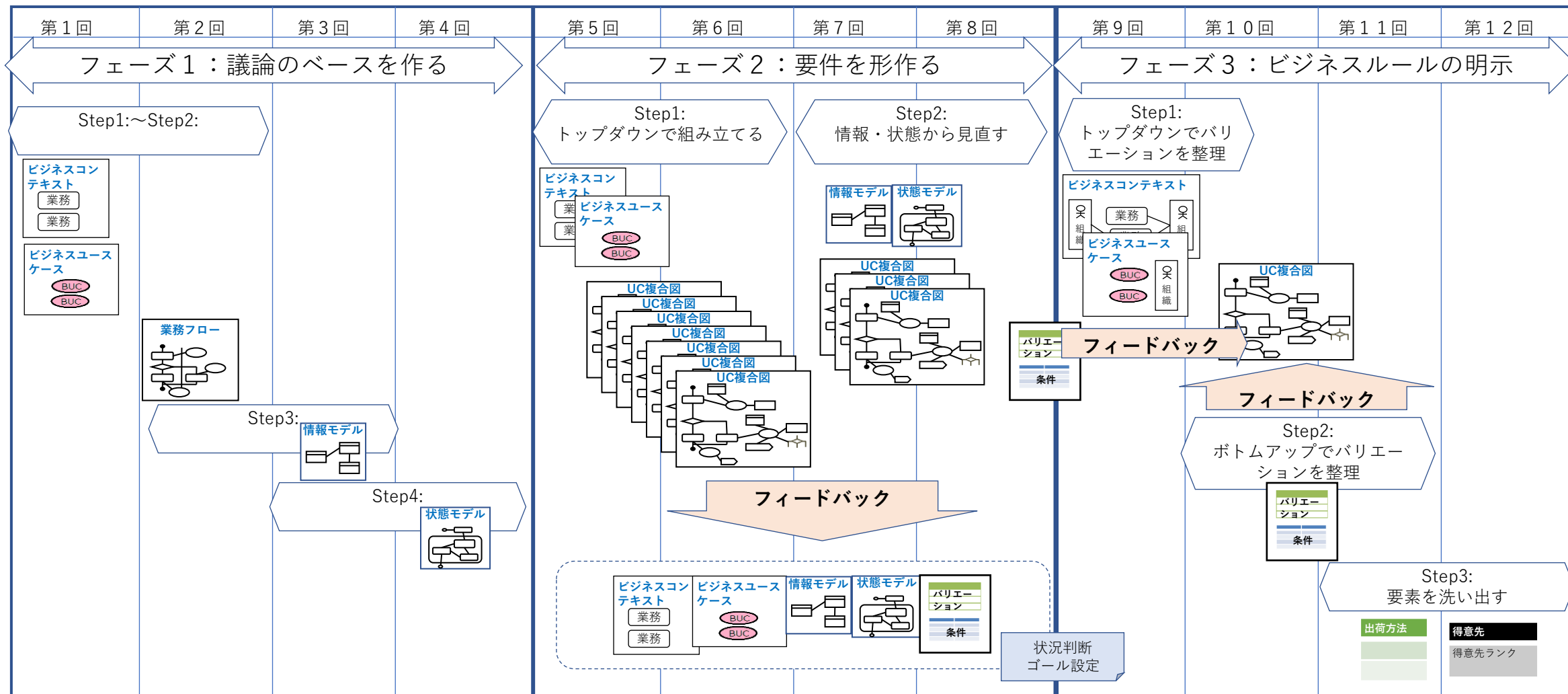




2. 計画の組み立て

標準スケジュール

- RDRA推進手順で示したフェーズとステップをマイルストーンとイテレーションにマッピング
- 標準として12回のステップを示している
- プロジェクトの規模、特徴をもとにStep毎の期間を調整する



状況判断とゴール設定（例）

- 今の状況を共有しゴールの認識を合わす
- 状況判断シートを参考に現在の状況を判断する
- ゴールは切りのいいところ（同一ダイアグラムは同じ水準まで整備する）で設定する

ダイアグラム	要素	状況判断	ゴール設定
ビジネスコンテキスト	業務	業務を満たすBUCが洗い出され、全ての業務が洗い出されている	今の状態で完了とする
	ビジネス要素	業務につながるビジネス要素が全て洗い出されている	今の状態で完了とする
ビジネスユースケース図	BUC	BUCを実現するアクティビティが洗い出され、全ての業務が洗い出されている	今の状態で完了とする
	ビジネス要素	業務につながるビジネス要素が全て洗い出されている	今の状態で完了とする
	上記の状態のBUC関数	上記の状況にあるBUC関数は全体の4/5	全BUC関数を上記の状態にする
UC複合図	業務フロー	主要なフローは洗い出した	主要なアクティビティは洗い出す
	UC	必要なUCは洗い出した	必要なUCは全て洗い出す
	情報	重要なものだけつなげた	レビューを行い完了とする
	入出力画面、イベント、IF情報	必要なものは洗い出したが、レビューはしていない	レビューを行い完了とする
	条件	条件はまだ紐づけていないものがある	必要と思われる条件を全てつなげる
	上記の状態のUC複合関数	上記の状況にあるUC複合関数は全体の5/7	全てのUC複合関数を上記状態にする
情報モデル	構造	必要な情報はすべて出されている	情報は全てUCにつながっていることを確認する
	属性（バリエーション）	出していない	今回はここまで行わない
状態モデル	状態と遷移	必要なものは全て洗い出されている	今の状態で完了とする
	遷移にUCマッピング	状態遷移へのUCマッピングは50%程度	全ての状態遷移にUCをマッピングする
バリエーション・条件	バリエーション	バリエーションそのものは洗い出された 要素は洗い出されていない	要素は洗い出さない
	条件	UCに結びつけた条件に対応する条件の表を作成したが、各条件の要素は洗い出していない	条件の要素は洗い出さない
精度向上	情報・状態・UCの関係確認	実施していない	今回はここまでの精度は求めない
	条件と情報、UCの関係確認	実施していない	今回はここまでの精度は求めない

3. 支援方法と費用

支援内容と費用

- 概要
 - 要件定義そのものはお客様に行っていただき、1 2回の打合せで要件をGoogle SpreadSheetで定義します
 - 毎回の打合せの進め方、定義内容などをリードし、確実に1 2回の打合せでシステムの枠組みが把握できるレベルで要件を定義します
- 内容
 - 形態：ビデオ会議による打合せ
 - 方法：1 2回の打合せによる要件定義支援、3ヶ月のメールフォロー
- 方法
 - 1 2回の打合せを週1回または2回で実施
 - Google SpreadSheetでの要件定義方法を支援
- 成果物
 - Google SpreadSheetによる要件定義
 - 成果物例： [クラウドサービスの定義例](#)
- 費用
 - 150万円/1回（1 2回の打合せ+ 3ヶ月メールフォロー）
 - 以後の追加作業は1回の打合せ毎の費用計算となり、10万円/1回の打合せ