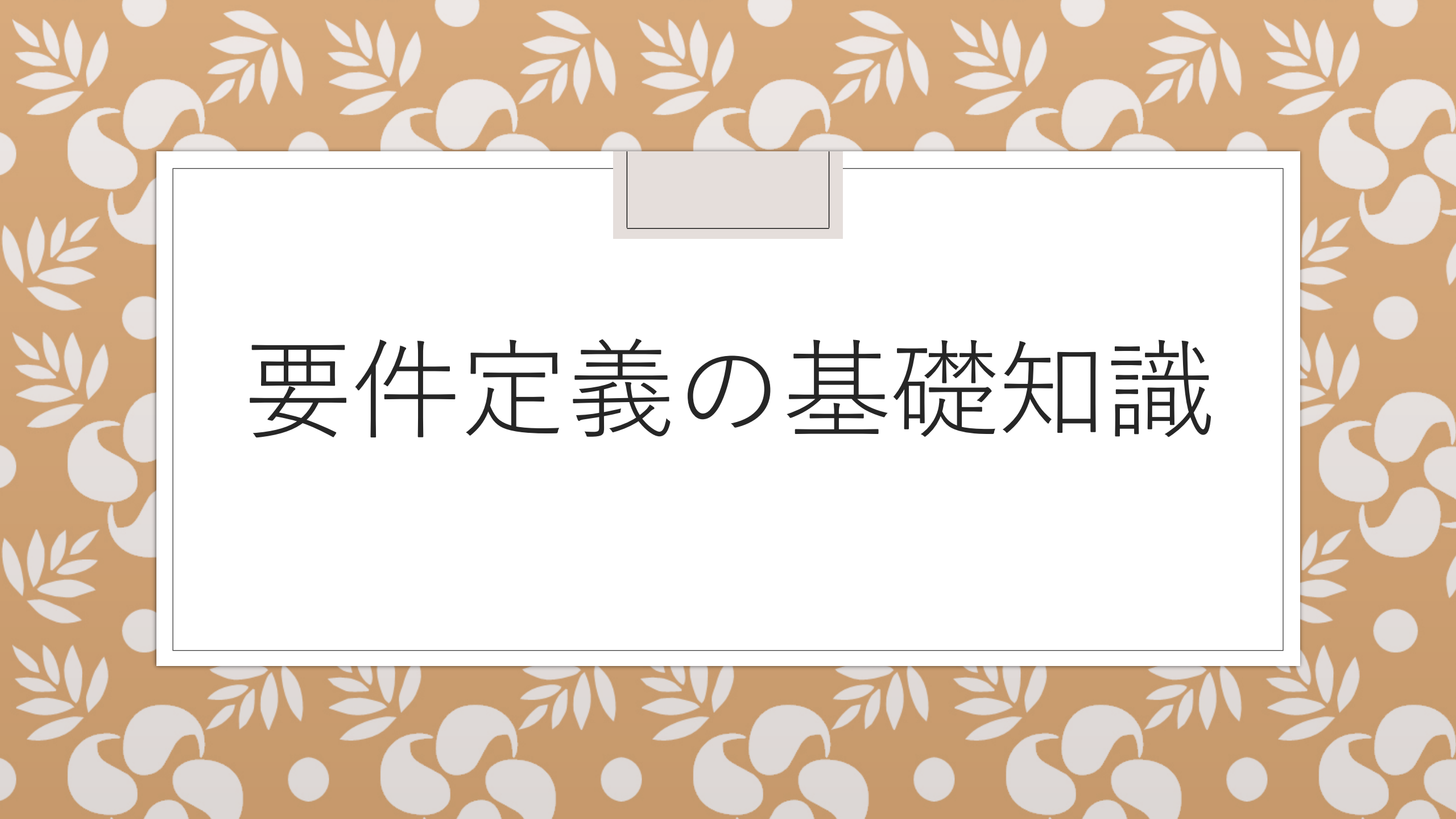




システム設計実践演習 第3回



要件定義の基礎知識

「要件」と「定義」とは

そもそも「要件」とは何でしょうか？

辞書で調べてみると、「求められる条件のこと」とあります。

「定義」とは「物事の意味・内容を他と区別できるように、**言葉で明確に限定すること**」であり、英語にあたるdefineには「**範囲を明らかにする**」という意味もあります。

「要件定義」とは

改めて、要件定義とは

「ユーザ企業 or 自社サービスの事業目標を達成するため、情報システムに求められるものを定め、その範囲を明らかにすること」

になります。

要件と要望、要求との違い

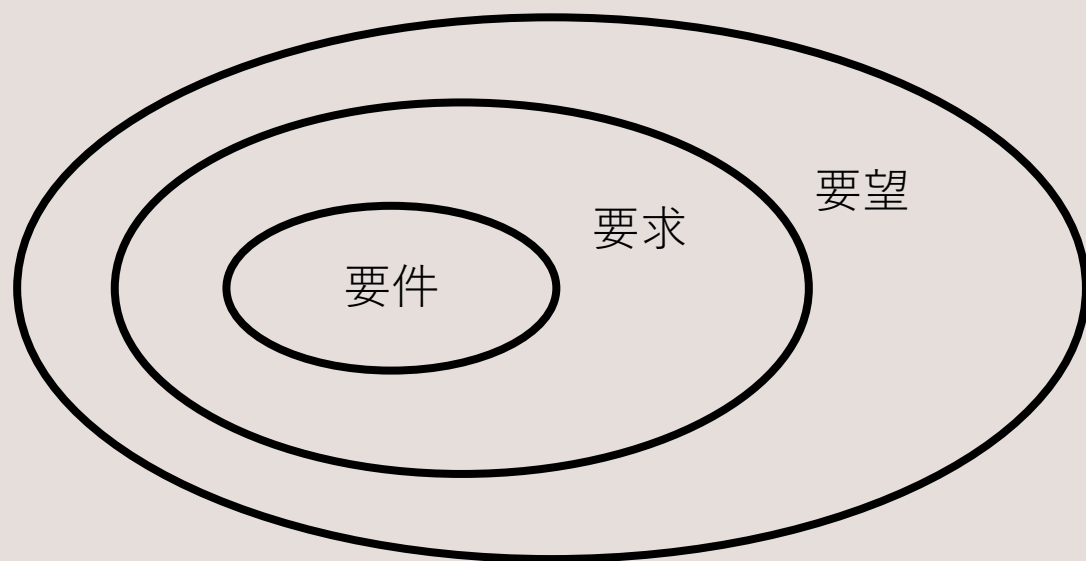
要件に似た言葉に、「要望」や「要求」があります。

「要望」は、こんなものがあるといいな・・・希望であり、個人の単なる夢やわがままが含まれます。組織で実現に向けて**取り組む価値があるかどうか吟味**しなければなりません。

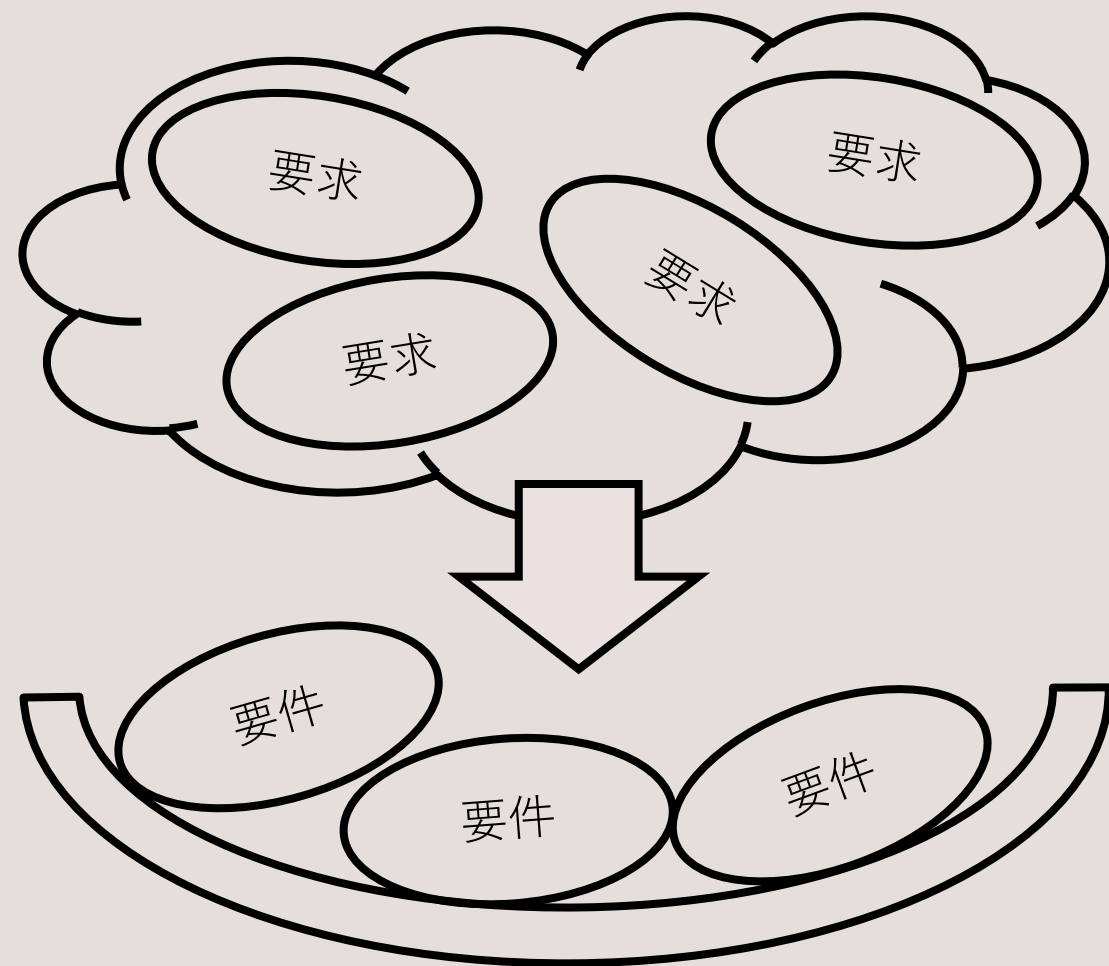
「要求」は、これが必要なんだという依頼であり、限られた予算の中ですべてを叶えることが出来ないので、**本当に必要なものを厳選**しなければなりません。

要望、要求が絞り込まれたものが「要件」

要望、要求、要件の関連性



要求から要件を絞り込む時には、
関係者がその判断に納得できる
合理的な理由が必要



隠れた要求を掘り起こして分析

ユーザは、漠然と「問題があるな」と感じていてもうまく表現できないことがあります。また、問題そのものに気づかないこともあります。或いはわざわざ言わなくてもわかるだろうという「暗黙の要求」もあります。

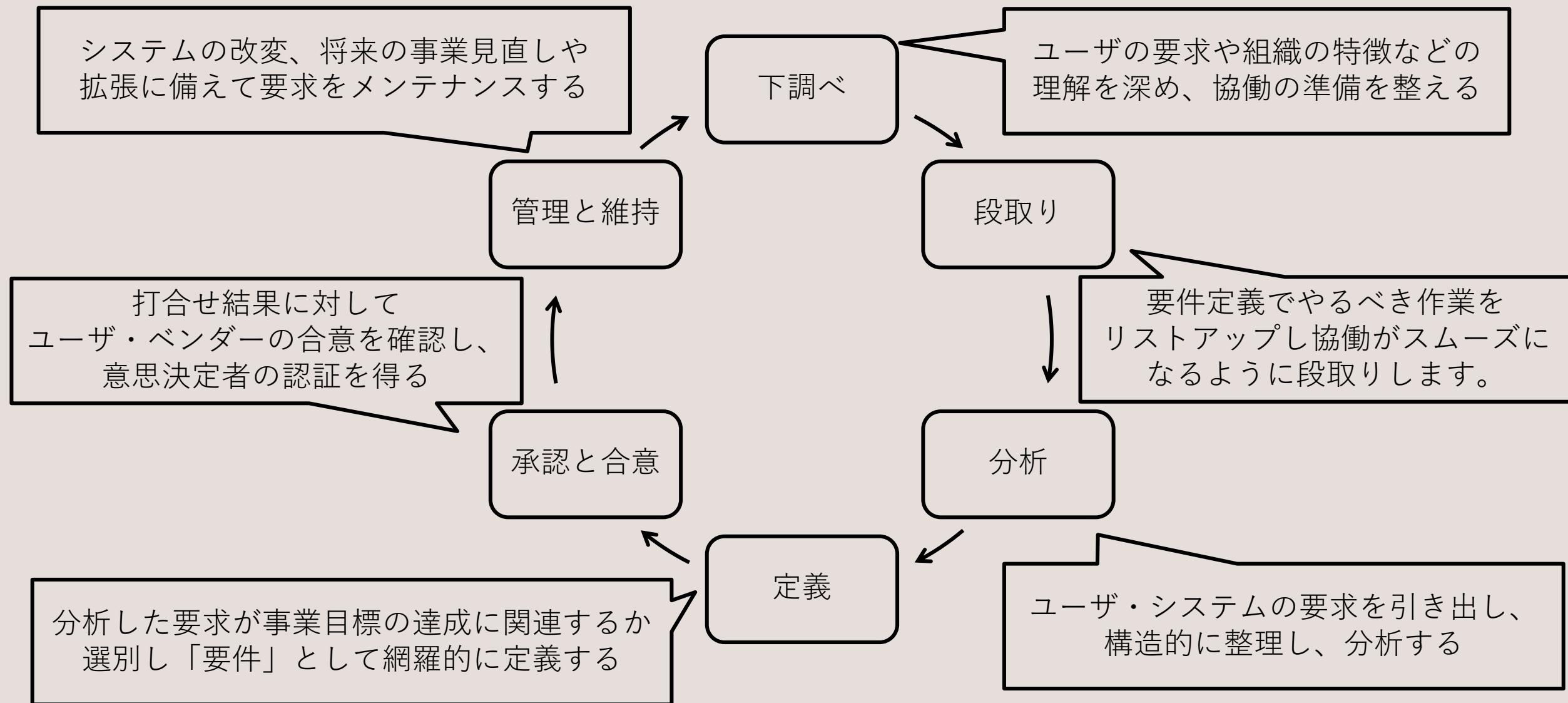
これらは、「要求」として表に出てこないので「要件」にもリストアップされません。

隠れた要求を掘り起こして分析

要件定義を進める場合、このような**隠れた要求もうまくインタビューして、掘り起こしていかなければなりません。**



要件定義作業の全体像



各工程の進め方

これらの工程は**順番通りに一度ずつ実施されるとは限りません。**

例えば、段取り、分析、定義ごとに合意を実行することもあるれば、業務要件定義、機能要件定義のそれぞれの後に承認を実行することもあります。

要件定義 完了時点での到達目標

要件定義が完了したら、次は「設計」に移ります。

設計フェーズは、要件定義を行った企業がそのまま請け負うこともあります。別の企業が設計を担当する場合があります。

ユーザ企業は、新たな「設計案件」の契約のために見積もりを行うのでそのための情報が必要です。

要件定義 完了時点での到達目標

要件定義の完了時点で、**システム規模の見積もり**ができ、**設計作業に必要な情報の引継ぎができること**を目指します。

システム規模が見積もれる具体性

設計工程の作業を進められる詳細度



下調べ・段取り 工程

下調べ・段取りの必要性

要件定義に着手する前の準備や計画の善し悪しは、要件定義の成否を左右します。

決してユーザと敵対するものではありませんが、相手の特徴を理解し、主導権をとる準備をすることで要件定義の作業がスムーズに進めることが可能です。

ユーザ企業の下調べを行う

要件定義をするにあたり、ユーザ組織の事業目標や、システム化の対象範囲、組織の特徴などを下調べし、ユーザとメンバーのベクトルを合わせます。

インタビューやディスカッションに参加してもらうため、組織の多くの関係者を巻き込むことが必要です。

報告や意思決定のスピード、稟議の回り方など、**組織の特徴をつかんでおく**と仕事が進めやすくなります。

文書化を行い、合意・周知を図る

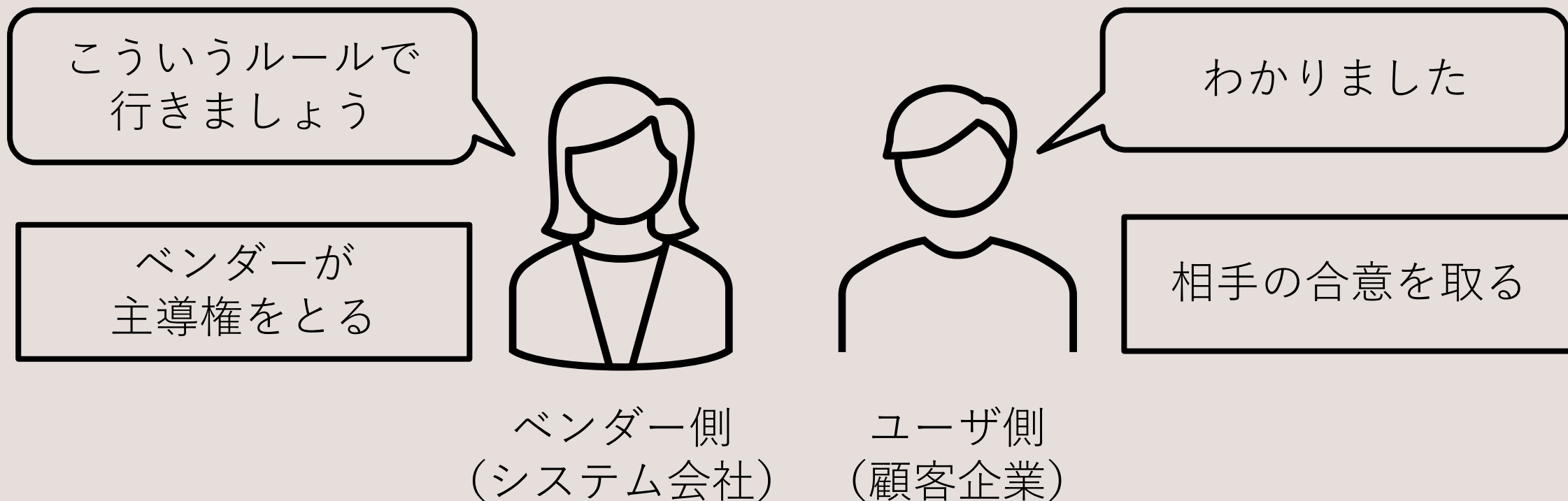
口頭のみやり取りだと、言った言わないでトラブルになります。

要件定義では、やるべき作業を洗い出し、いつ、誰が、どのように取り組むのかを計画し、スケジュールや役割分担などを文書化します。

この時、相手側に舵取りを任せると、相手の都合に振り回されかねません。要件定義の主体はあくまでユーザですが、システム会社側で段取りをして、「このように進めましょう」と提案しながら、リードするのがおすすめです。

文書化を行い、合意・周知を図る

主導権を取って進行すると、**作業の流れをコントロール**でき、自分たちのペースも守りやすくなります。



衝突解消のための基準を決めておく

要件定義で衝突が発生しがちなシーンがあり、事前に関係者が納得できるような解消方法を決めておくトラブルに発展しにくいです。

衝突が発生しがちな場面

- ・ 優先順位付け
- ・ 作業の開始・終了判定
- ・ 合意・承認

衝突解消のための基準を決めておく

それぞれシーン別に**判定のためのチェックリスト**を用意して
トラブルが発生してから判定方法を考えるのではなく、**計画時
点で予め準備**しておくといよいでしょう。

〇〇の場合は、
多数決で決める

□ □の場合は、
役員会に一任する

× ×の場合は、
コストの低いほうを
優先する

下調べの構成要素

事業目標の共有	プロジェクトの目標を ユーザと同じ視点で共有する
対象範囲の確認	プロジェクトを取り巻く環境や背景を理解し、その中でプロジェクトの対象範囲を確認する
ステークホルダー (利害関係者)の 認識	プロジェクトに直接関係する人だけでなく、間接的に関わり、影響を与えたり、受けたりする人、グループを認識する

事業目標の共有

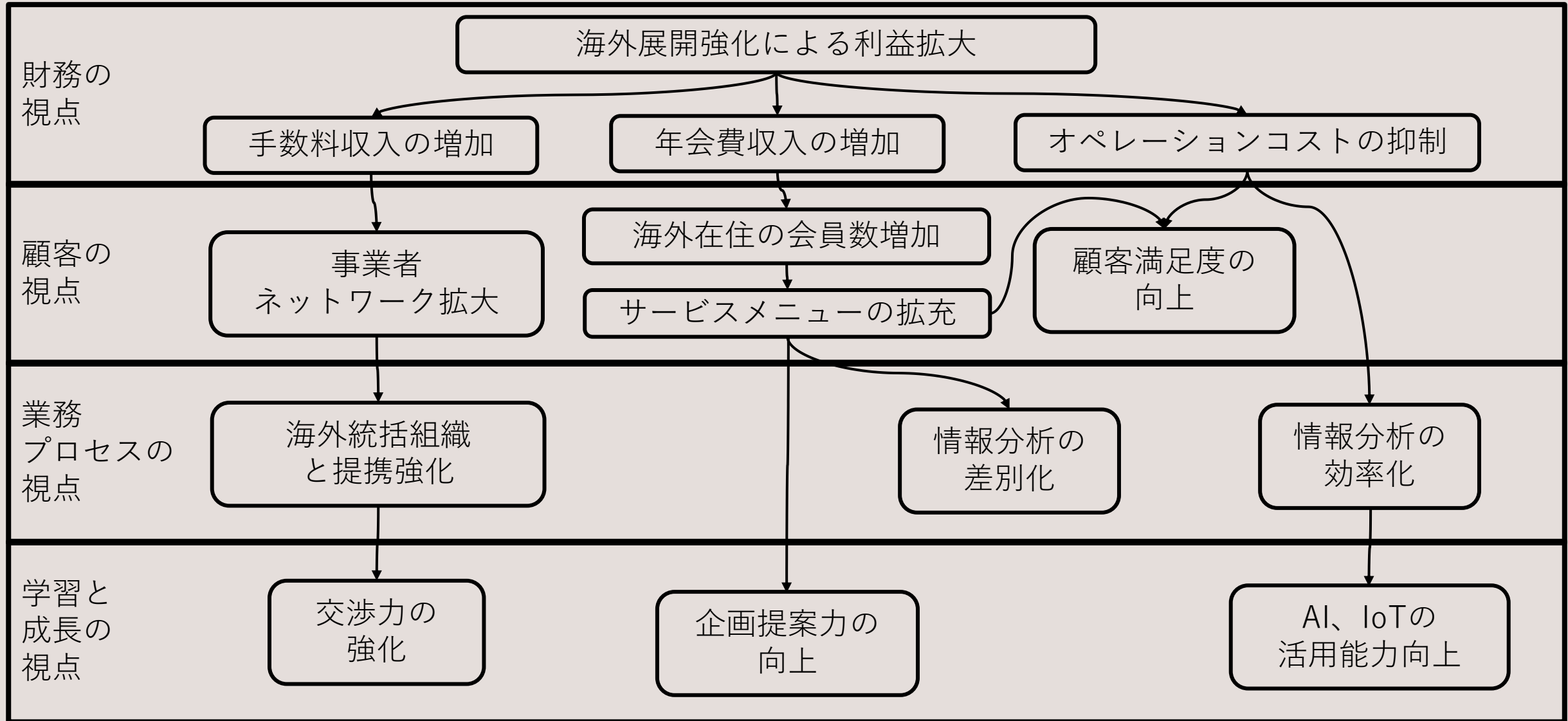
- ・ ユーザの事業を全体像で理解する

どのような事業の展開を期待しているのか、そのストーリーの中でこれから構築しようとしているシステムがどのように貢献するのか？
目標達成までの道筋をチーム内で共有する

- ・ 事業目標とシステム化の達成目標を因果関係で可視化する

バランススコアカード(BSC)の戦略マップを使い、4つの視点（財務・顧客・業務・学習と成長）で事業の戦略目標を図式化して因果関係を可視化する

【例】旅行代理店事業拡大の戦略マップ

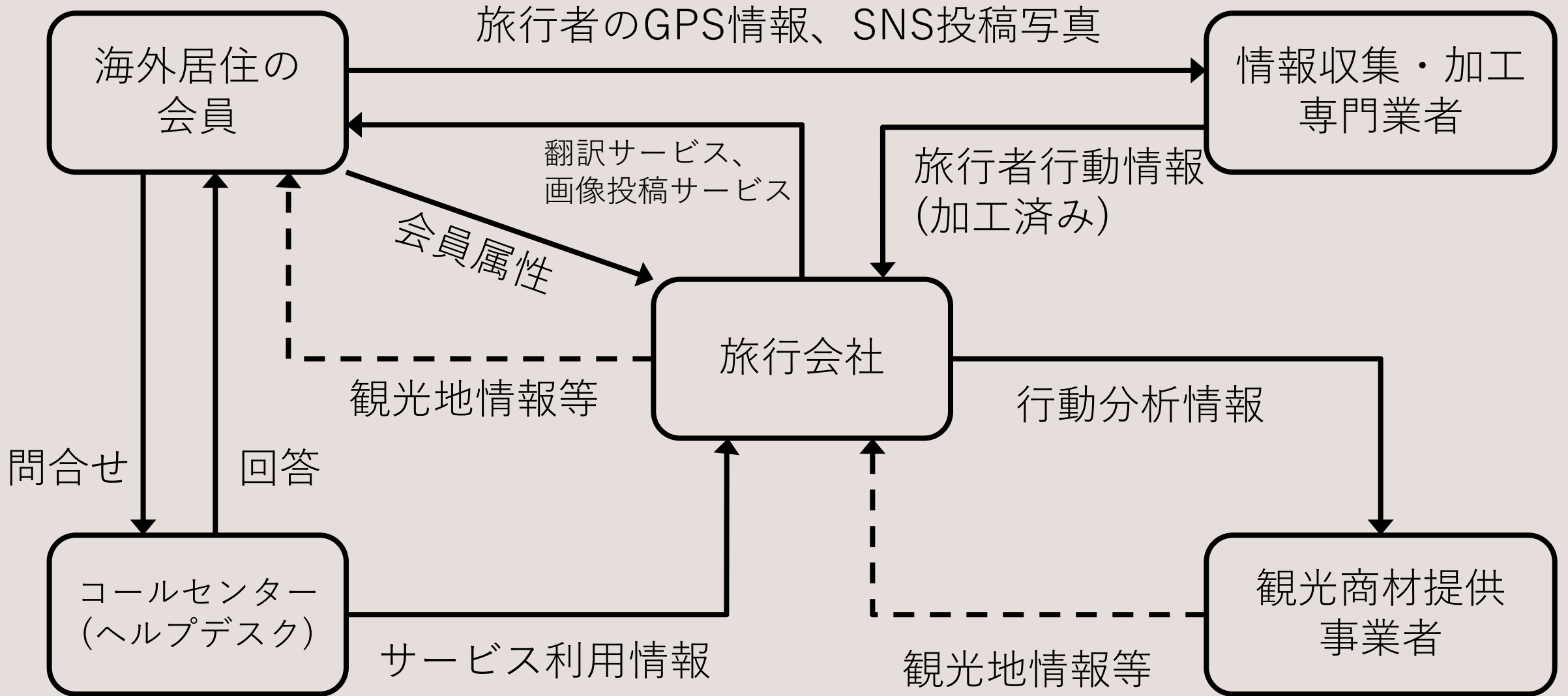


対象範囲の確認

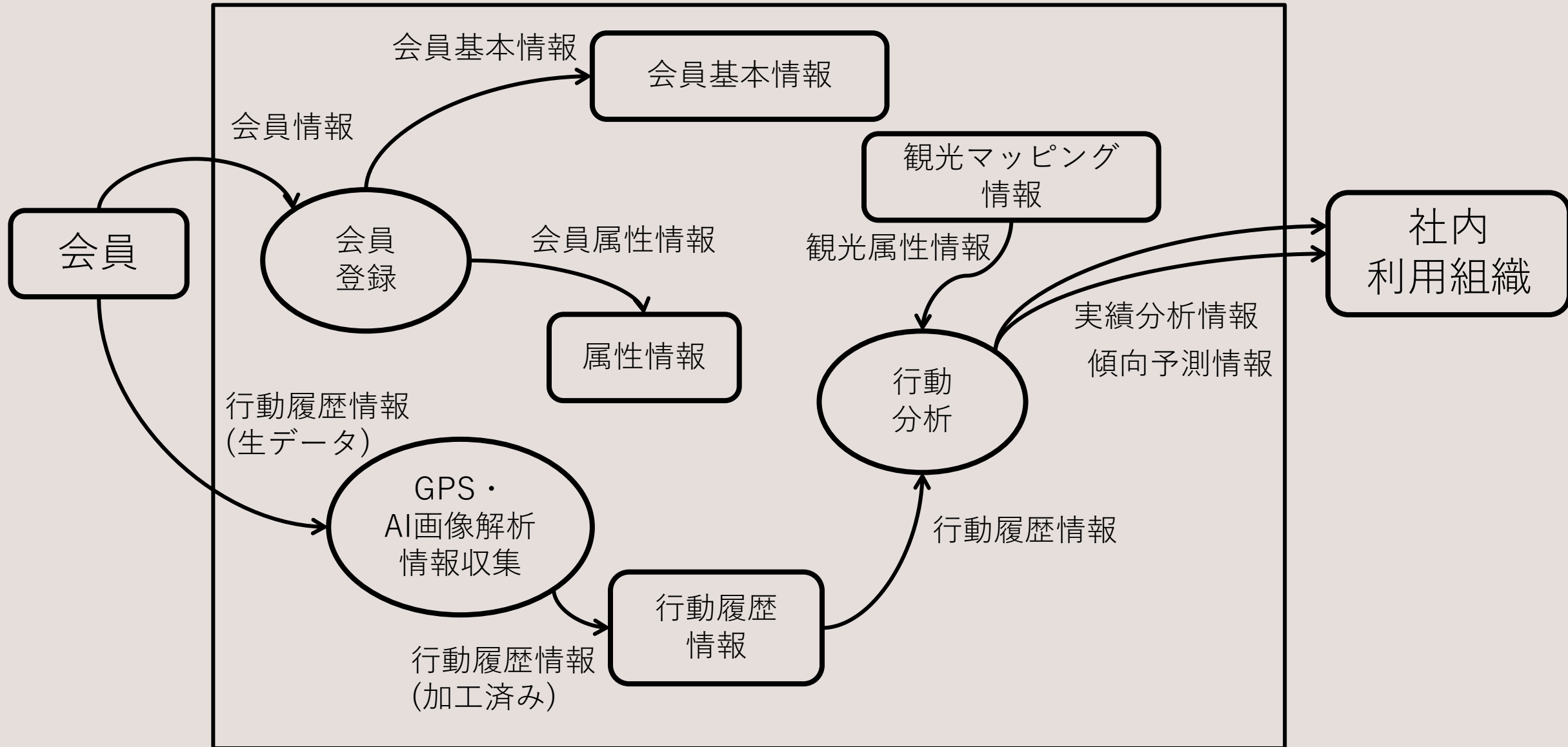
要件定義は、**対象のシステム範囲を明確にすることが目的の一つですが、その捉え方は、全体から捉えるトップダウンのアプローチ**が重要です。

- ・ エコシステムマップでシステムの相互作用をとらえる
事業活動全体を捉えるために、モノやお金の動き(物流・金流)を追いながら、そこで流れる情報(商流)を図式化する
- ・ コンテキスト図でデータの所在を追う
データの発生元、収集先、蓄積先を明らかにしてデータの流れを明らかにします。

【例】エコシステムマップ



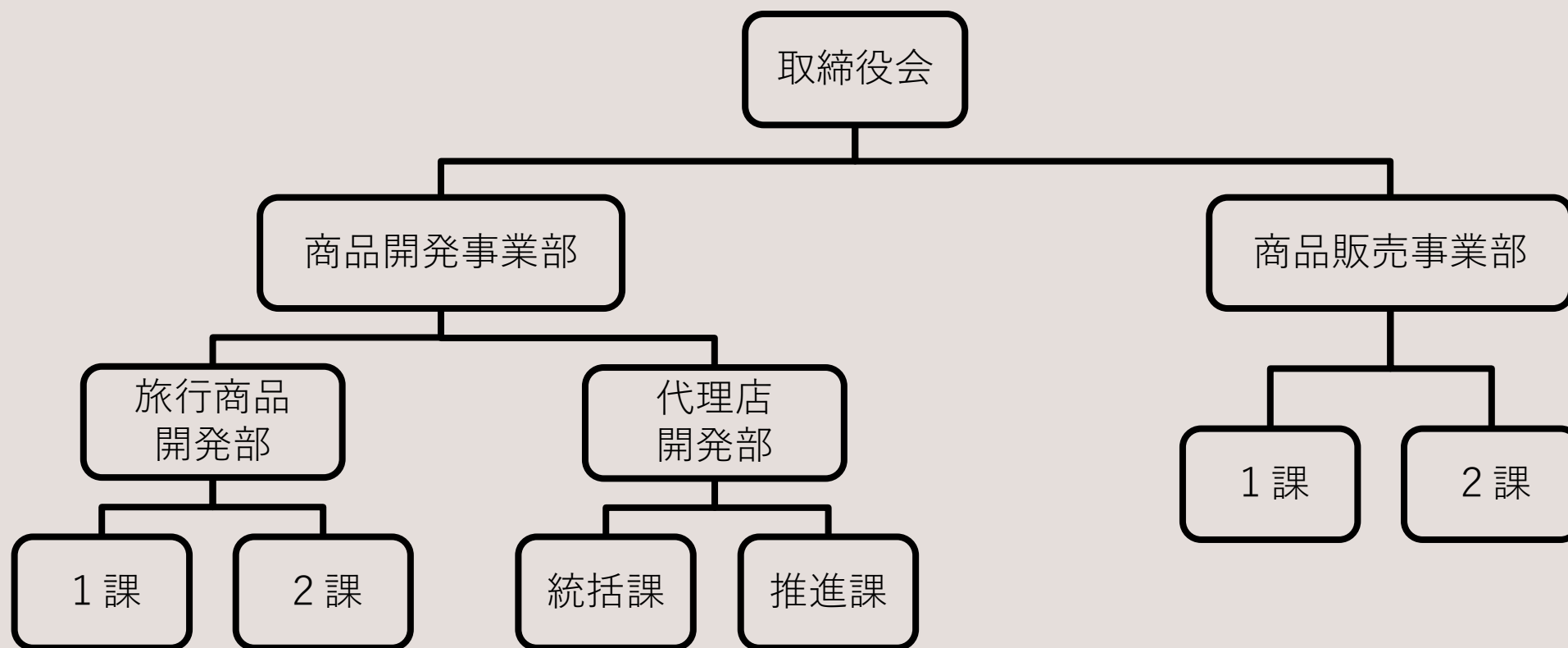
【例】コンテキスト図(DFD レベル0)



ステークホルダーの識別

- ・組織の活動や文化を理解する

組織に直接関係する人たちについては、組織図を入手し、組織の全体像や組織名、階層構造、役割を理解します。



作図の目的と効果

作図された図を前に置き、データの流れを指し示しながら、自分たちの認識や理解に不足がないかユーザに確認・共有することが、図を活用する目的でありメリットとなります。

段取りの構成要素

進め方の策定	作業内容を洗い出し、役割分担やスケジュールなどを策定する
ジャッジのための基準作り	要件定義の作業中に衝突が発生した場合の解消方法を予め策定する
計画書の作成	段取り時に決めた内容を計画書にまとめ、チーム内で共有する。ユーザに説明を行い不都合がないか確認・合意を残す

進め方の策定

- ・ToDoとコミュニケーション

要件定義を進めるに関係者の合意を得て周知をしてもらう必要があります。具体的には次の2点の取り決めを行います。

**やるべきこと ※何を、いつ、誰が行うか
コミュニケーションの取り方**

やるべきこと（何を、いつ、誰が行うか）

要件定義でやるべきことをできるだけ具体化してカテゴリー別にリストアップします。カテゴリーは「業務領域・システム領域」とか「インタビュー・集中討議・定例会議・分析作業・ドキュメント作成」など要件定義チームで決めた分類を記しておきます。次に、どの順番で実施すべきか、開始・終了日の予定を横棒で表す。

アクティビティ	1月	2月	3月	4月	5月	6月	7月	8月	9月
要件定義									
概要設計									
詳細設計									
プログラミング									
テスト									

コミュニケーションの取り方

要件定義では、インタビュー、レビュー会、報告会などさまざまなタイプの情報伝達が欠かせません。タイプ別に誰に集まってもらうか、どのようなやり方で実施するか、いつ、どこで実施するかを段取りします。

誰に	情報伝達のタイプ別に相手を特定
会議	目的、開催頻度、場所、参加メンバー
どのように	TV会議が必要か、時差の考慮必要か、直接会って会議するか、メールでのやり取りで十分か コミュニケーションの取る方法を決める

ジャッジのための基準づくり

利害の異なる大勢の関係者の間で、さまざまな決め事を行う場合、衝突がおこるのは、ある意味健全で当たり前のことです。

衝突に対して、**意見を調整して衝突を解消し、関係者が納得できる合意に導く**ことは、要件定義を行う担当者の重要な役割です。

衝突が発生しがちな場面

- ・ 優先順位付け
- ・ 作業の開始・終了判定
- ・ 合意・承認

優先順位付けの判断基準

要件定義の作業が進み、たくさんの要件が出た場合、構築予算や納期の都合上、取捨選択が必要になることがあります。

単純な多数決では関係者が納得できないこともあるので、「事業価値」や「投資対効果」などの判断軸で判定を行います。

また、MoSCoWという、汎用的な判断軸も良く使われます。

優先順位付けのランク	説明
M(Must)	必要不可欠（必須）
S(Should)	できるかぎりやるべき（推奨）
Co(Could)	費用、納期に余裕があるなら実施（可能）
W(Won't)	今回は不要（先送り）

開始と終了の判定基準

開始と終了の判定があいまいなまま作業を進めると、資料の不備や内容理解の不足により手戻りが発生してスケジュールが遅延する可能性を秘めています。

このような事態は、「何がどの程度準備できていれば作業開始可能」、「何がどの程度達成できていれば作業終了可能」といったルールやチェックリストを決めておくことで回避できます。

開始と終了の判定基準

これは、開発期間でも同様です。

過去のチームでバグが出ようがプログラムが書ければ作業完了という人いなかったですか？？？

作業の開始は、仕様を理解するところから？それともプログラム着手にはっていたら？

作業の終了は、プログラムコードがかけたら？単体テストが終わったら？それともプルリクエストを作成したら？

チーム内で開始と終了の基準、一致できていますか？

計画書の作成

進め方やコミュニケーションなど、さまざまな領域で段取りを策定したら計画書にまとめて、関係者の間で共有し、合意を得ます。計画策定のプロセスで、作ってきた中間産物を集め、表示と目次をつけて体裁を整えます。

今後、段取りの見直しや追加も考慮して、改訂履歴も含めておくとよいでしょう。