

第58回P2Mクラブ

ビジネスを駆動する プロジェクトマネージャにとっての システムズエンジニアリング

2024年7月25日

日本プロジェクトマネジメント協会

会員

佐藤 健司



はじめに

プロジェクトは、その複雑さと境界の曖昧さにより成功の不確実性が増す中、新しいプラットフォームを活用した革新的なイノベーションの出現によって、業界の地図が大きく塗り替えられています。

また、多様な技術領域が交錯する中、これまでの画一的な取り組みではプロジェクトの成功を収めることがますます難しくなっています。

こうした状況下で革新的なビジネスモデルを実現し、企業の成長をリードするためには、既存の安全領域から飛び出し、日々挑戦と成長を目指す人材が欠かせません。

今回、システムズエンジニアリングの知識を駆使し、プロジェクトマネジメントの最前線で活躍するビジネスドライバーとなる人材の育成についての課題と解決策について探り、皆さまと議論できればと考えております。

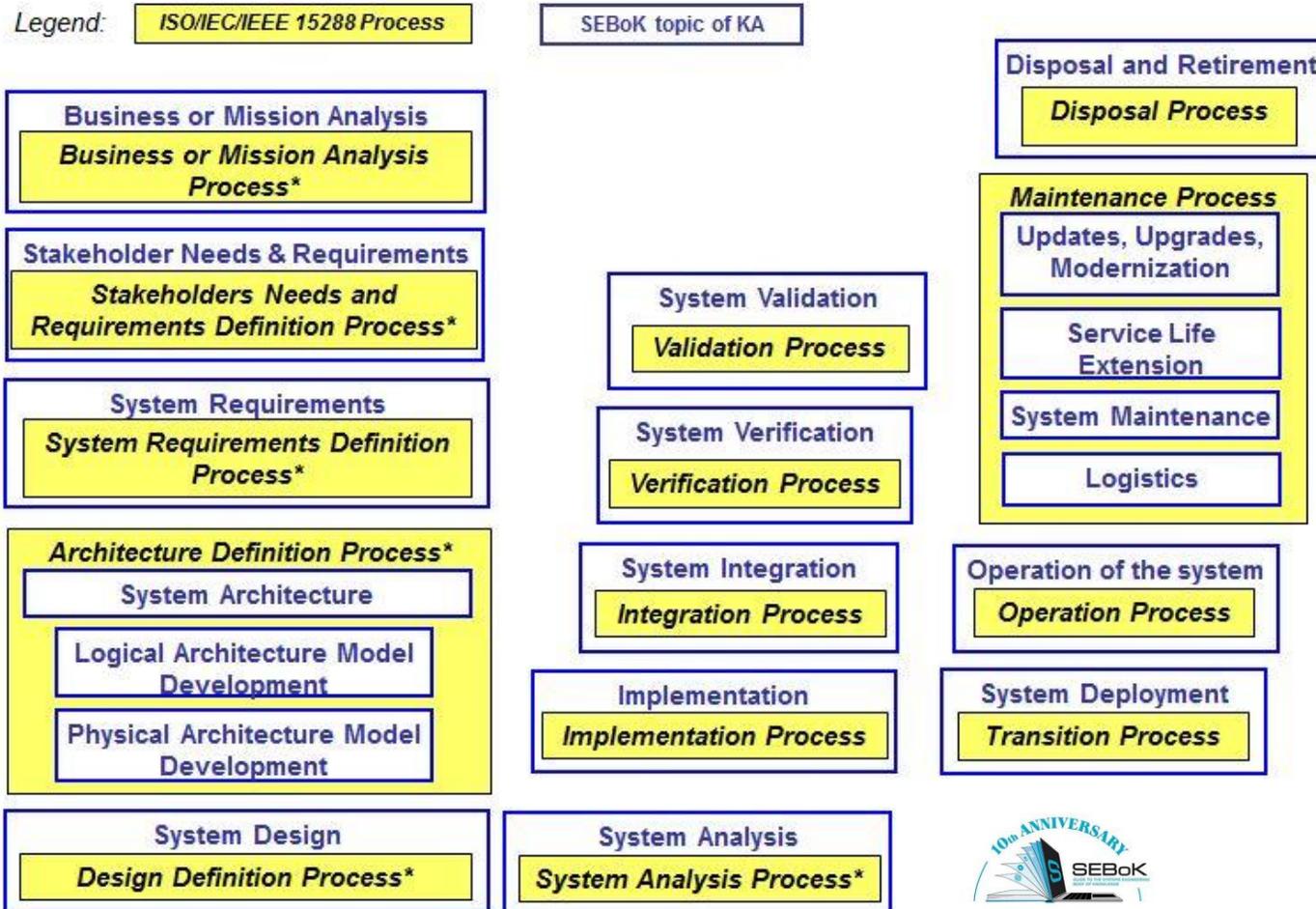
Agenda

第一部 ビジネスモデル

1. ビジネスモデルを生み出す
2. ビジネスモデルに成功をもたらす
3. ビジネスモデルを駆動する

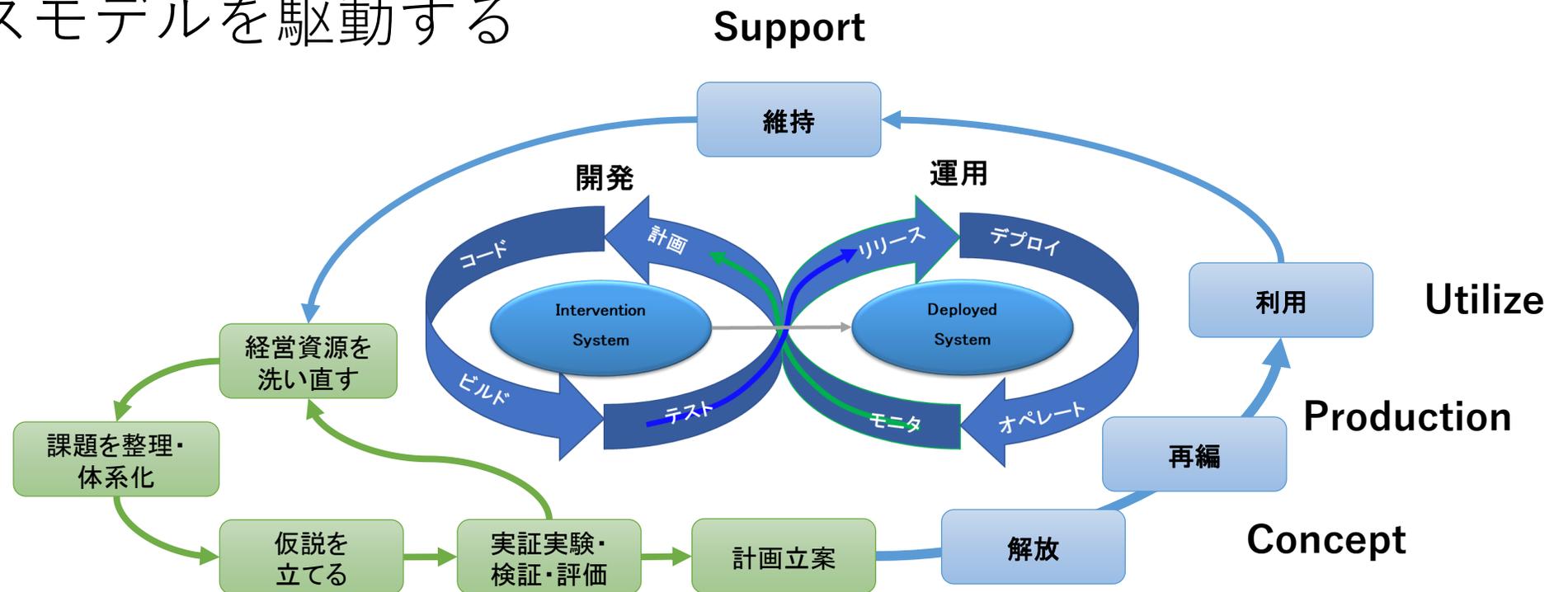
第二部 システムズエンジニアリング

1. その意味
2. 駆動する力
3. 規範的な力
4. ディシプリン
5. 適用の滞り
6. コンテキストの理解
7. その意思決定



第一部 ビジネスモデル

1. ビジネスモデルを生み出す
2. ビジネスモデルに成功をもたらす
3. ビジネスモデルを駆動する



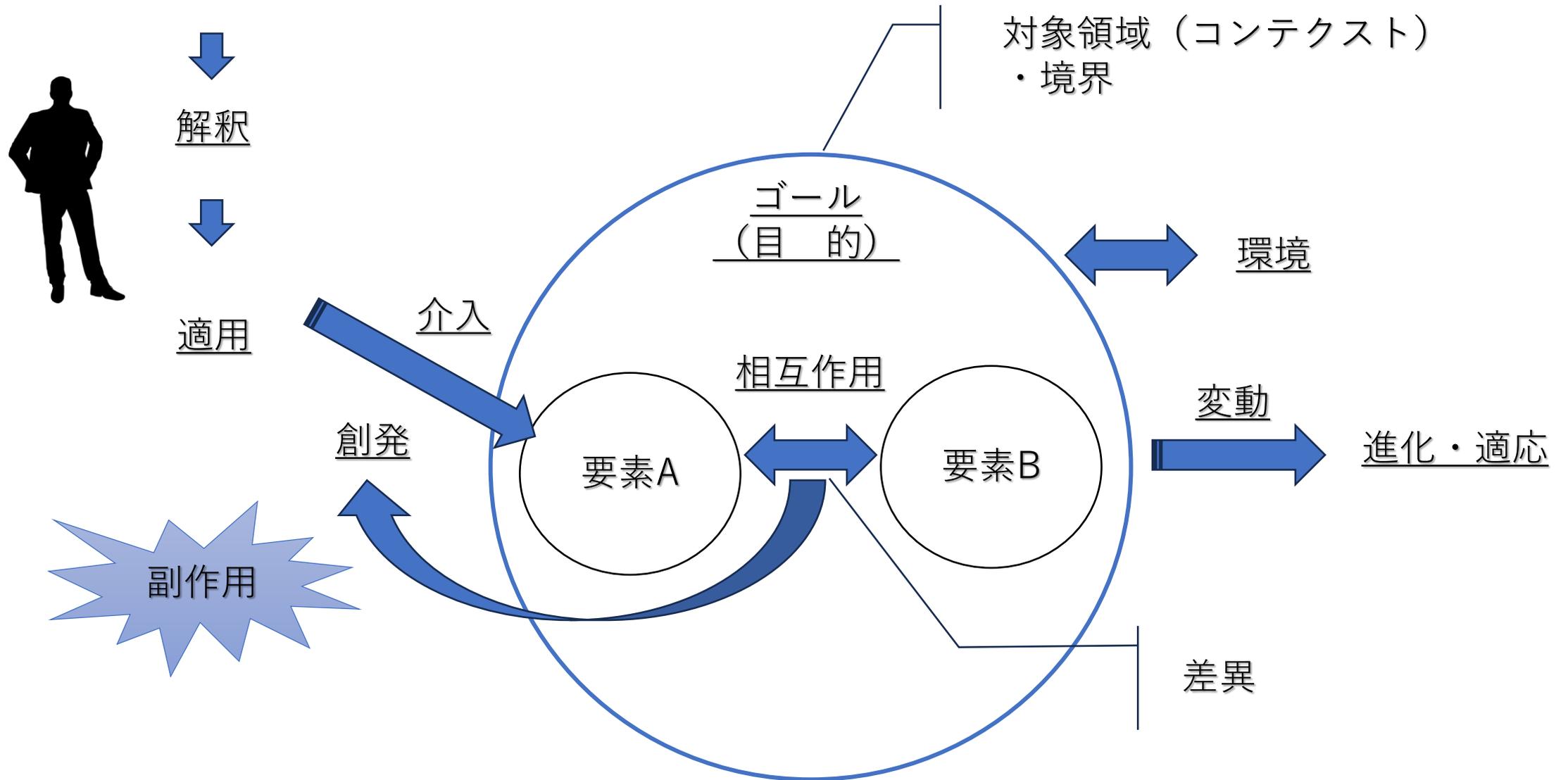


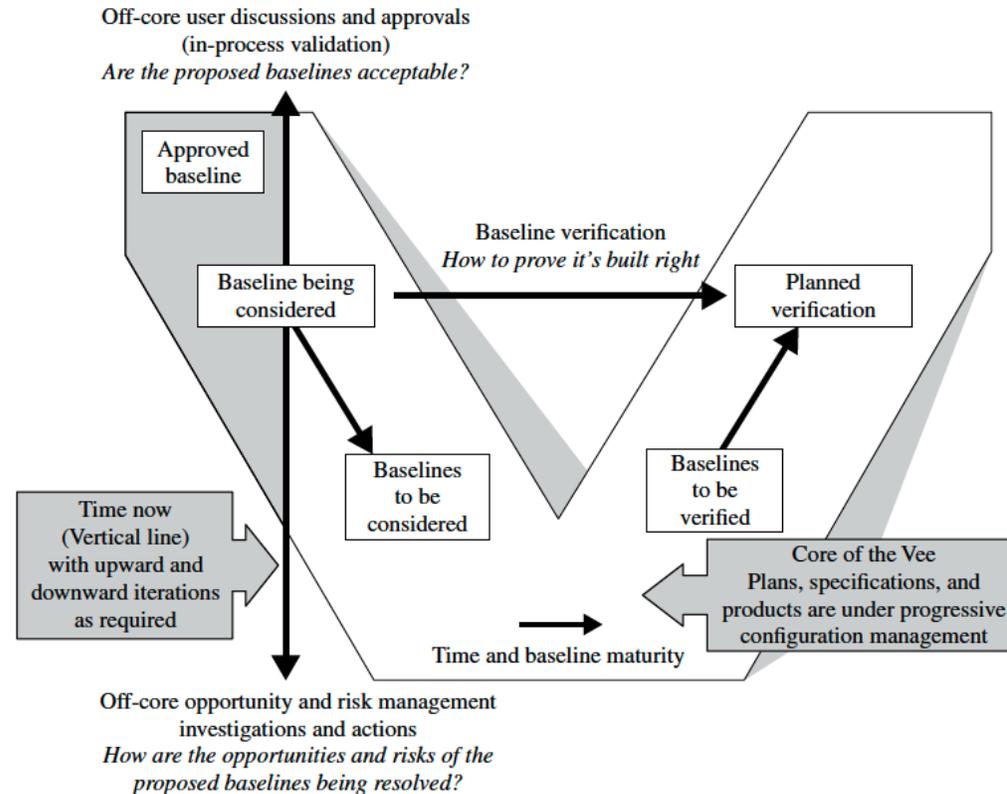
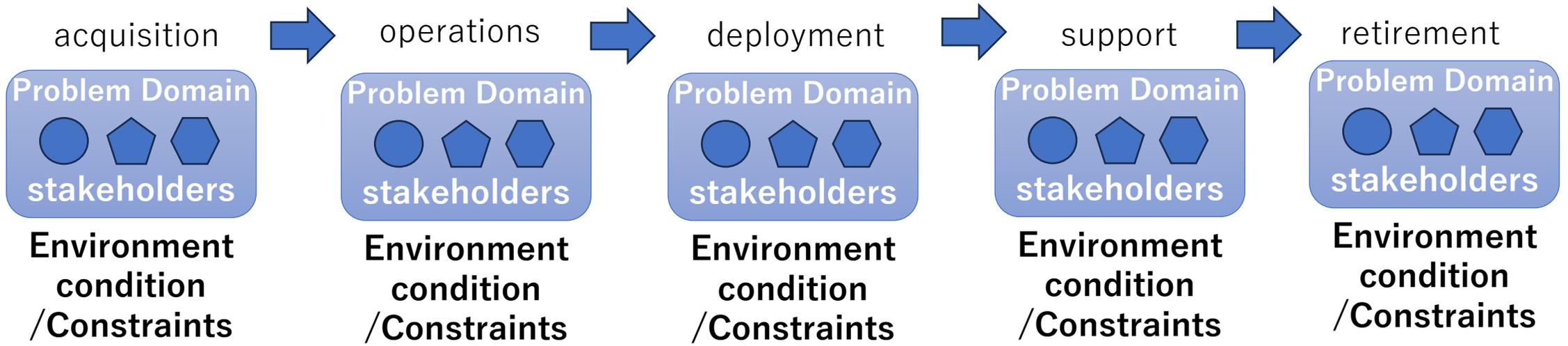
Reference

4.1 Business Mission Analysis Process

1. Defining the problem domain
2. Identifying major stakeholders
3. Identifying environmental condition and constraints that bound the solution domain
4. Developing preliminary life cycle concepts for acquisition, operations, deployment, support, and retirement
5. Developing the business requirements and validation criteria

システムエンジニアの仕事はプロジェクトを言語化すること



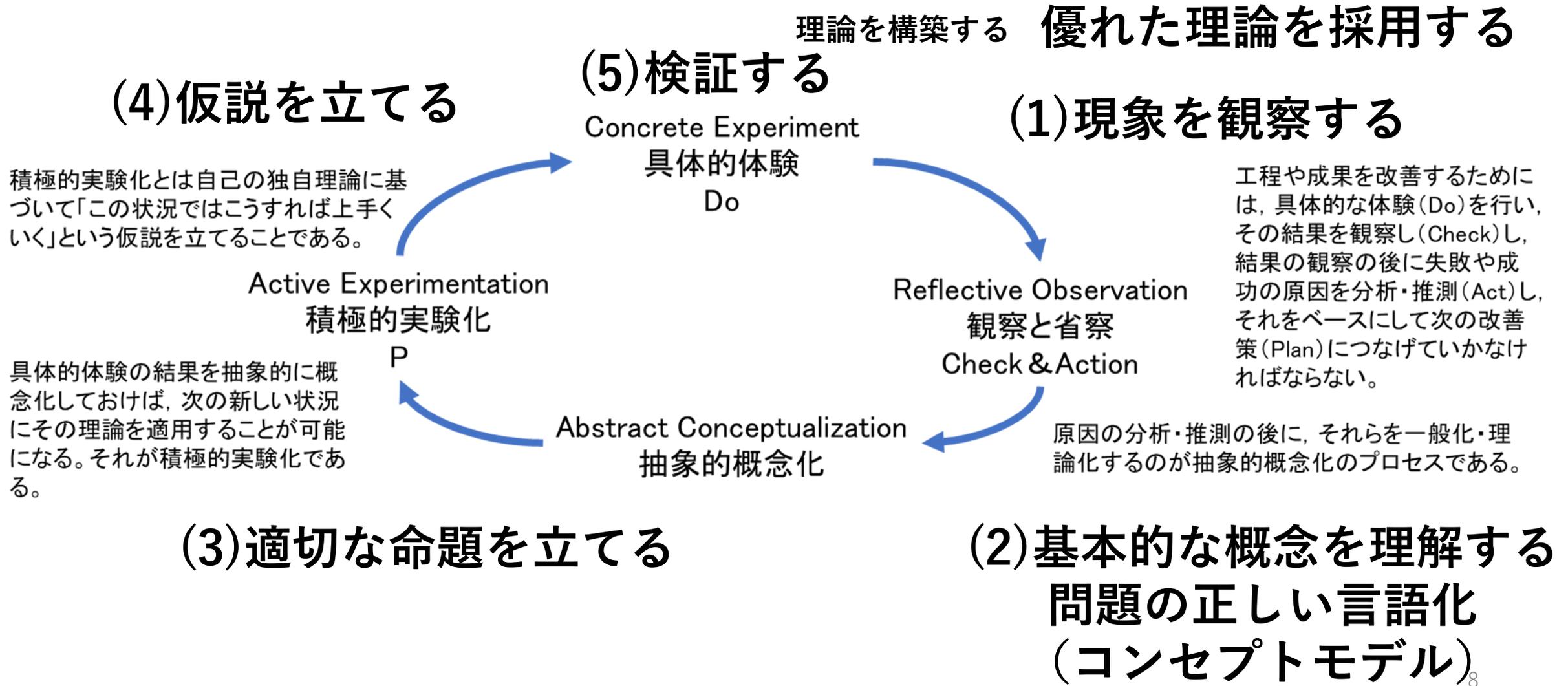


出典：SEBoK

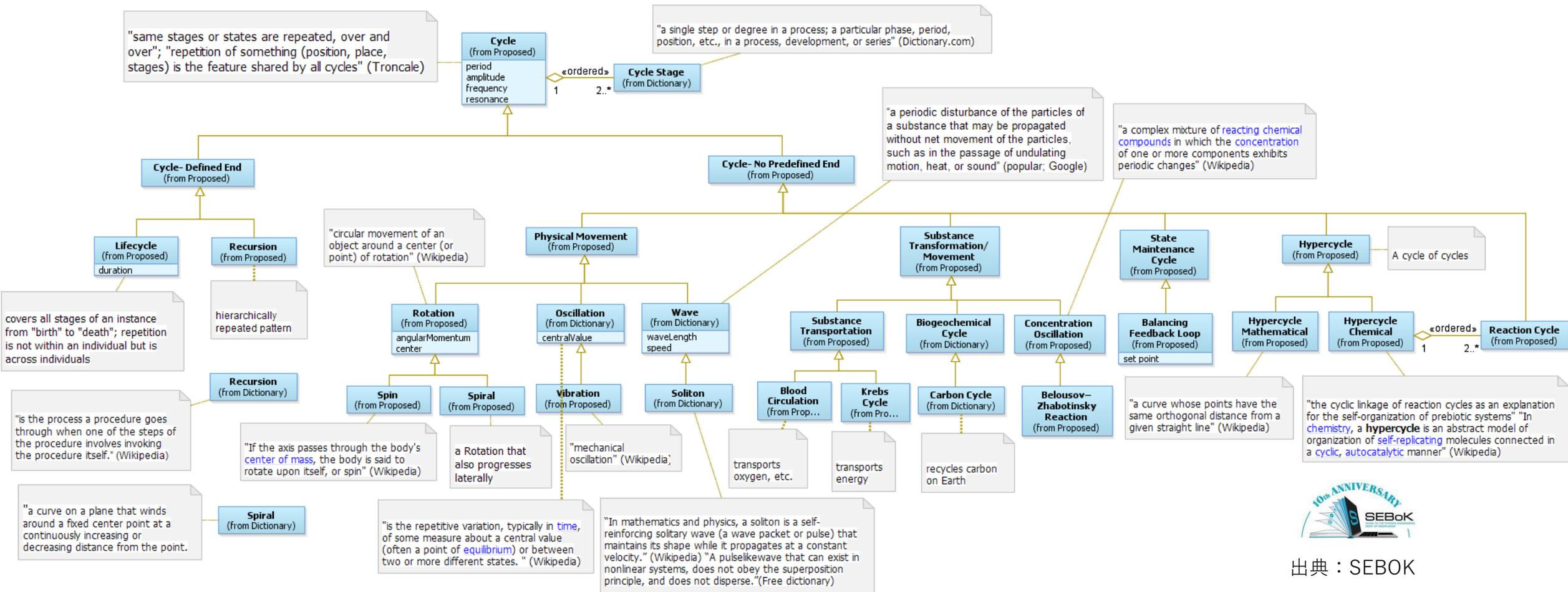


Reference

1. ビジネスモデルを生み出す(1/2)



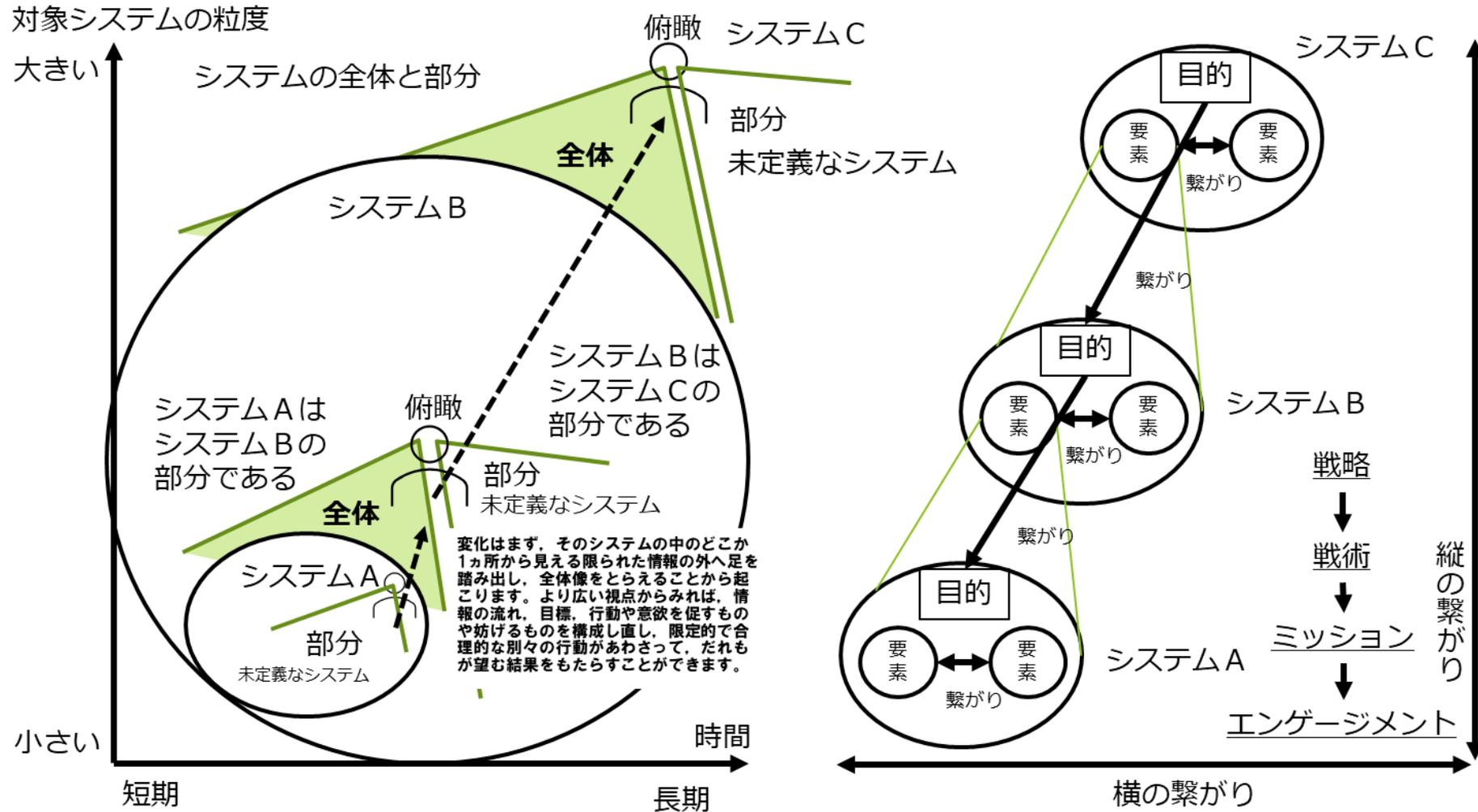
1. ビジネスモデルを生み出す(2/2)



出典：SEBOK

1. ビジネスモデルを生み出す

(1) 対象領域を決め、現象を観察する



動的平衡にみる「全体と部分」



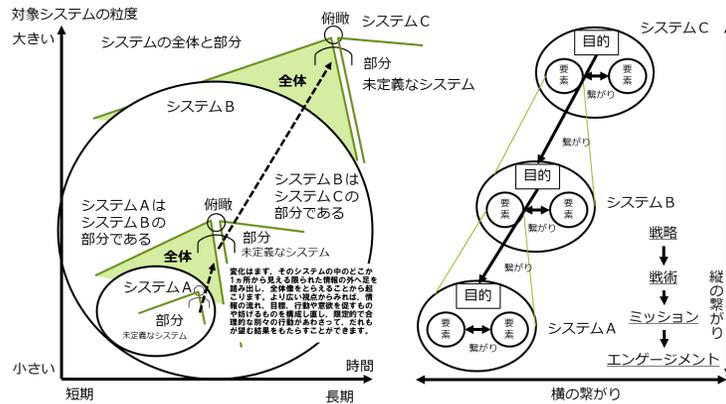
Reference

この世界のあらゆる要素は、互いに連関し、すべてが一对多の関係で繋がっている。

つまり世界にも、身体にも本来、部分はない。部分と呼び、部分として切り出せるものもない。世界のあらゆる因子は、互いに他を律し、あるいは相補している。そのやり取りには、ある瞬間だけ捉えてみると、供し手と受け手があるように見える。

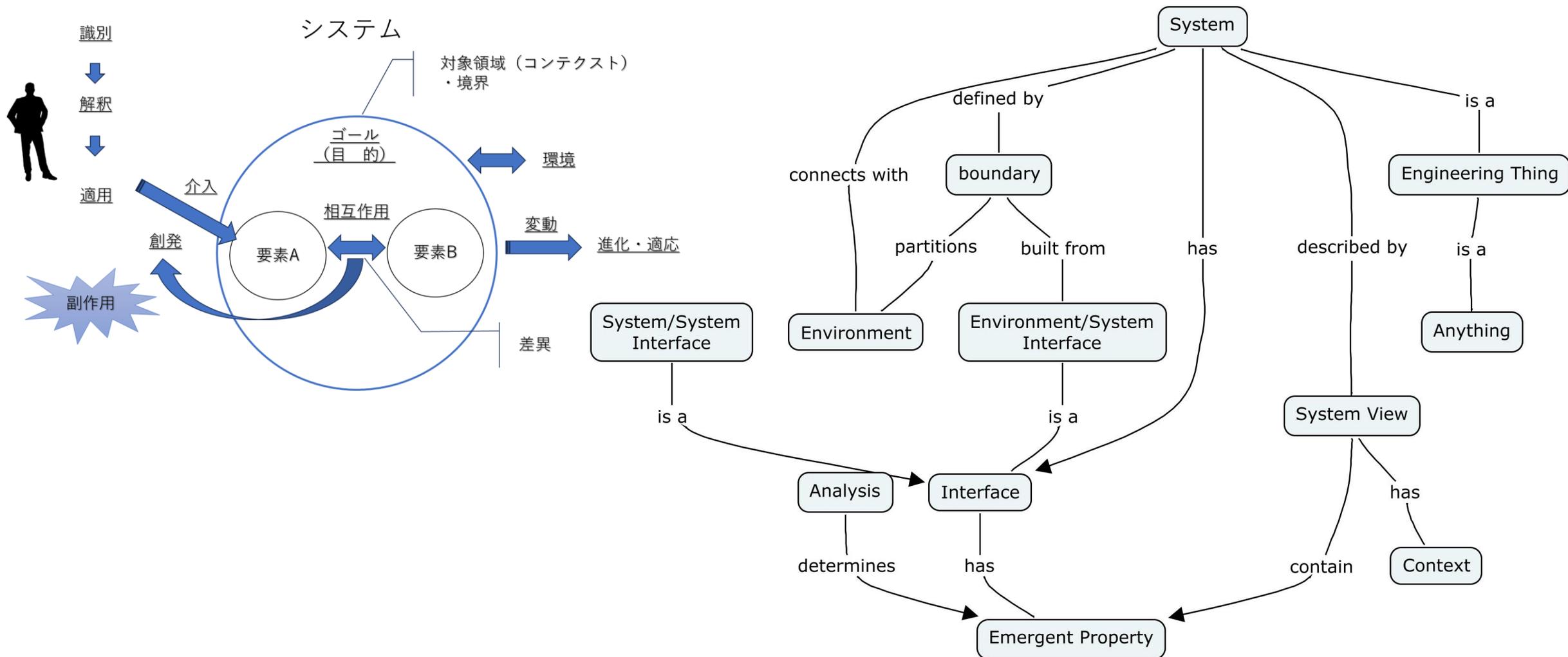
しかし、その微分を解き、その瞬間を見ると、原因と結果は逆転している。あるいは、別の平衡を求めて動いている。つまり、この世界には、ほんとうの意味で因果関係と呼ぶものもまた存在していない。世界は分けないうことにはわからない。しかし、世界は分けてもわからないのである。

(出典；福岡伸一著「動的平衡3」)



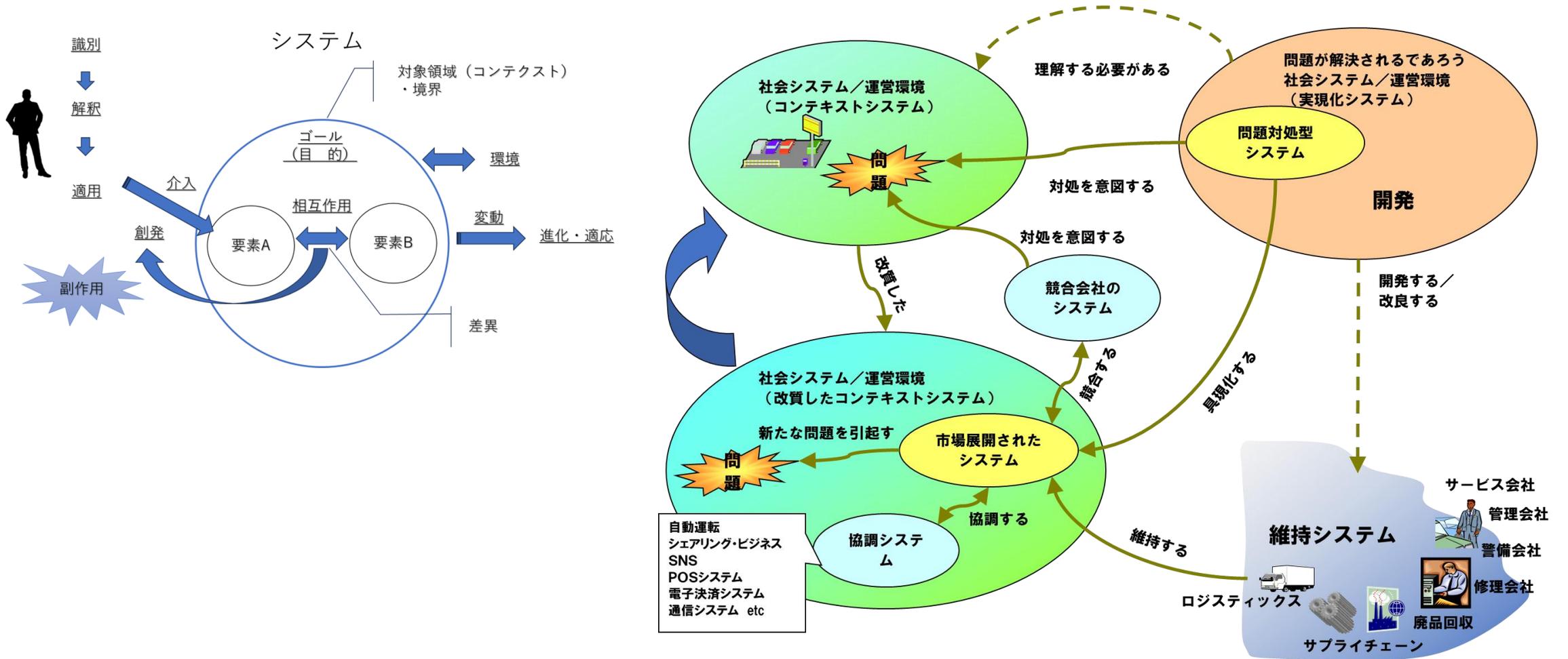
1. ビジネスモデルを生み出す

(2) 対象領域の基本的な概念を理解する(1/2)

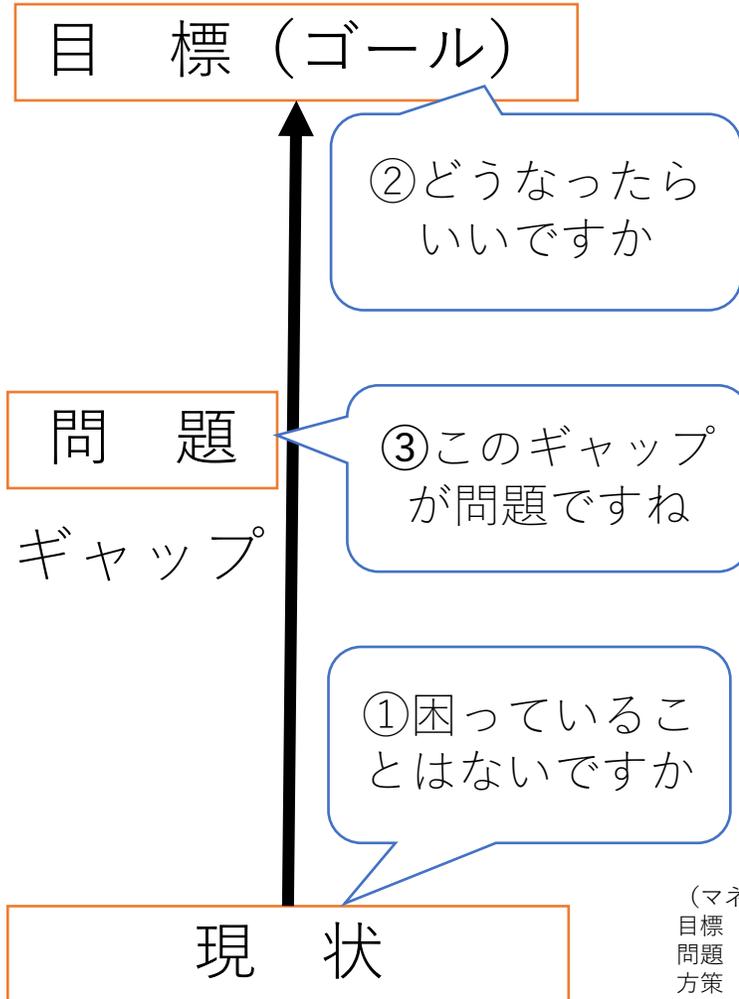


1. ビジネスモデルを生み出す

(2) 対象領域の基本的な概念を理解する(2/2)



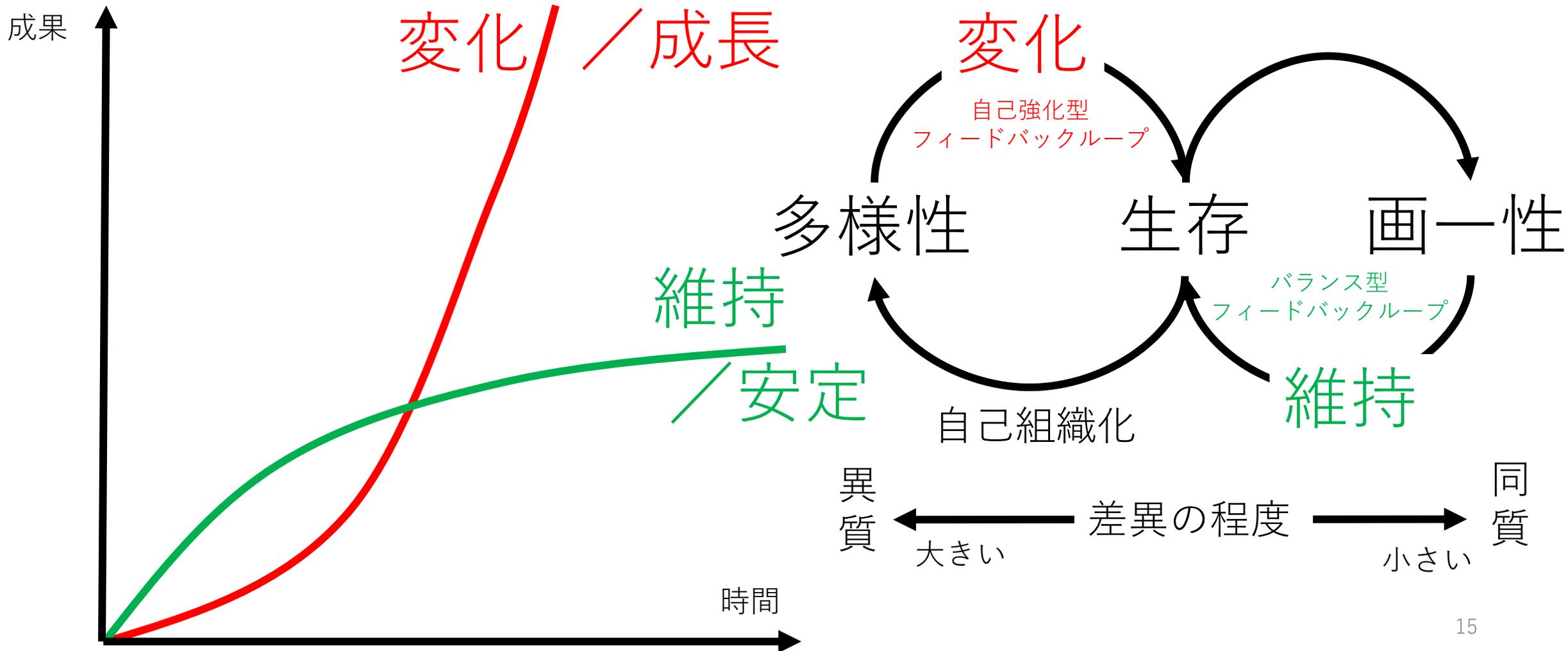
1. ビジネスモデルを生み出す (3) 適切な命題を立てる(1/3)



1. 問題とは、目標と現状とのギャップである。
2. 目標「ゴール」なきところに「良い問い」は立てられない。
3. 誤った問いからは正しい答えは得られない。
4. 問いの立て方とは、問題を正確に定義すること。問題を正確に定義できたら、半分解決したも同じ。
5. 目標と現状とのギャップは学ぶための素材にすぎない。

(マネジメントシステムのパフォーマンス改善－継続的改善の手順及び技法の指針 (JIS Q 9024 : 2003))
目標 (objectives) 方針又は重点課題の達成に向けた取り組みにおいて、追求し、目指す到達点。
問題 (problem) 設定してある目標と現実との、対策して克服する必要のあるギャップ。
方策 (means) 目標を達成するために、選ばれる手段。

1. ビジネスモデルを生み出す
(3) 適切な命題を立てる(2/3)



維持／安定



Reference

「**制度の繰り返しでおきる行動パターンで安定してかつ一定の評価を得ているもの**」：

制度は、**社会内で繰り返し行われる行動パターンが安定**しており、それが社会的に認知され、評価されることによって成り立ちます。安定した行動パターンとは、一定の規則や慣行に従った行動が継続的に行われることを意味し、それが人々にとって予測可能であり、信頼できるものとなります。例えば、民主主義制度は選挙という行動パターンが繰り返し行われ、その結果に基づいて政府が形成されるという安定したプロセスがあります。このプロセスが社会に広く受け入れられ、評価されていることで、制度として確立されています。

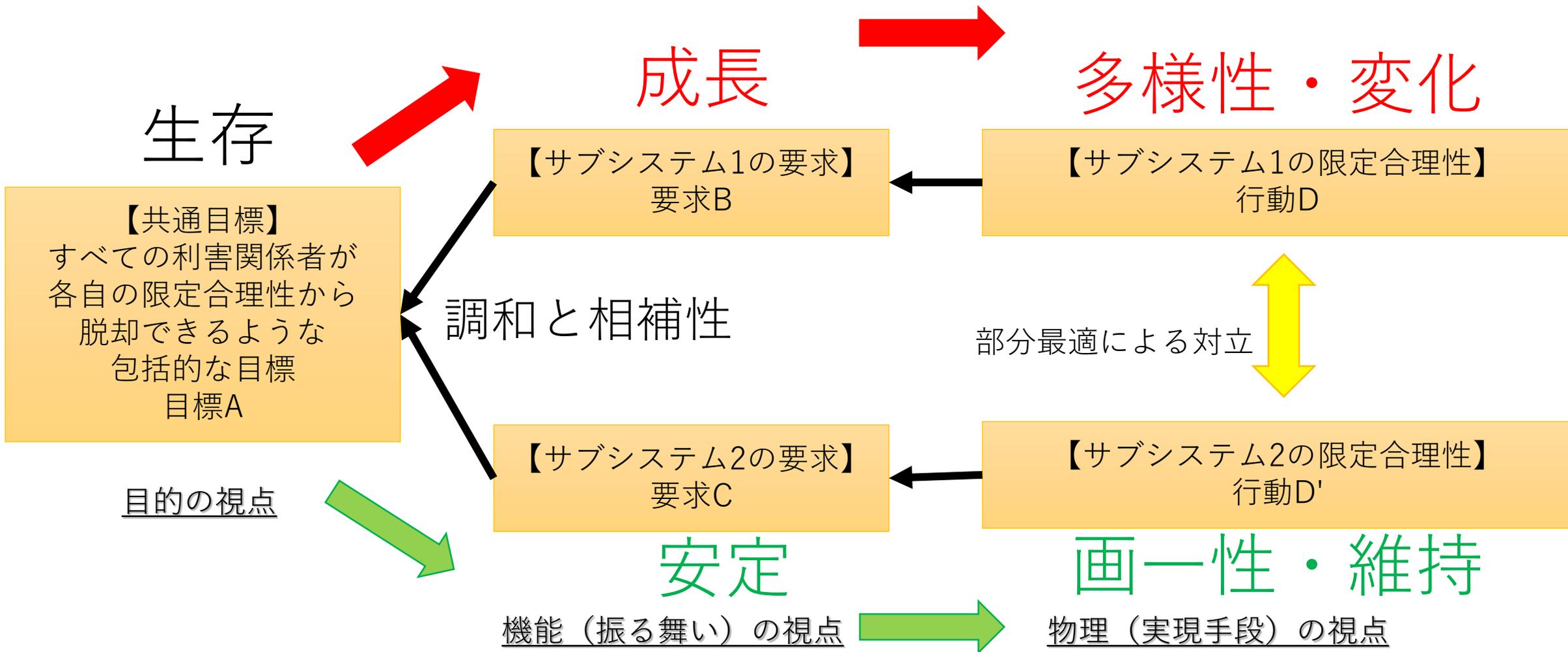
「**人間の行動をかたちづくり、制限し、一定の方向に導くルールで長く続くもの**」：

制度は、人々の行動を形成し、制限し、特定の方向に導く役割を果たします。これは、法律、規則、規範、慣習などを通じて実現されます。これらの**ルールは、個々の行動を予測可能で秩序立ったものにし、社会全体の調和を維持する**ために重要です。また、長く続くという点は、これらのルールが時間をかけて確立され、世代を超えて受け継がれることで、社会の安定と持続性を支えることを意味します。例えば、法治国家における法律の体系は、個人の行動を規制し、社会秩序を維持するための長期的なルールとして機能します。

「**ある歴史的な時点における必要に対応して形成された持続性のある行動パターン**」：

制度は、特定の歴史的な背景や社会的な必要に応じて形成されます。つまり、ある時点で社会が直面した問題や課題に対応するために、特定の行動パターンが生まれ、それが制度として確立されていきます。このような**制度は、その当初の必要性を超えて、持続的に機能し続けることが期待されます**。例えば、社会保障制度は、産業革命期における労働者の生活を保障するために生まれた制度ですが、現代においてもその基本的な構造は維持され、社会の安全網として機能し続けています。

1. ビジネスモデルを生み出す (3) 適切な命題を立てる(3/3)





Reference

バランスをとる

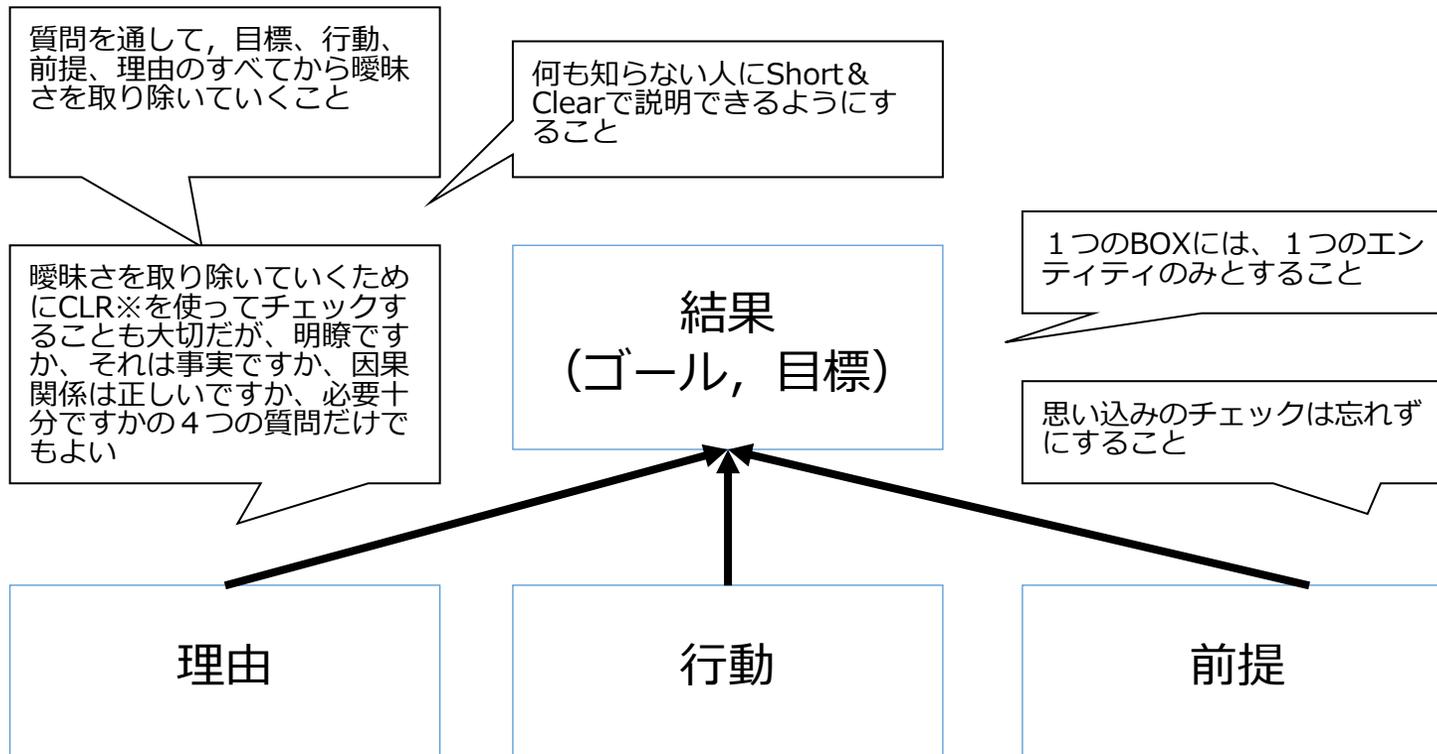
複雑なシステムでは、最適化はしばしば逆効果になる。

一部が最適化されると全体が最適化されなくなるか、最適化された全体が硬直化し、状況の変化に対応できなくなるかのどちらかである。

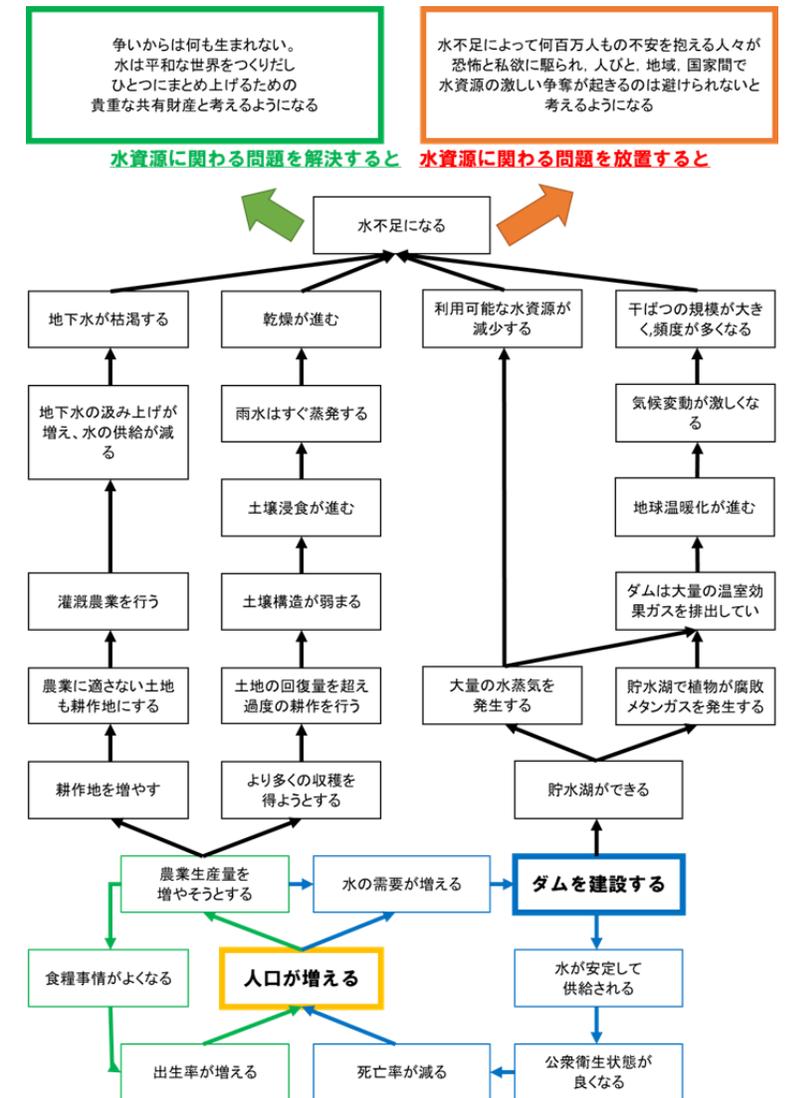
複雑なシステムエンジニアは、最適化する代わりに、プロジェクト内の競合する緊張関係のバランスを追求すべきである。

システムエンジニアは、統合的思考を活用して、改善されたソリューションを生み出し、二元的などちらか一方のトレードオフを避けることができる。ゴールは、現在の状況が数多く変化しても、ニーズを満たし続けるシステムである。

1. ビジネスモデルを生み出す (4) 仮説を立てる(1/2)



論理的検証方法CLR (Category of Legitimate Reservation) は, 因果関係のロジックにおいて論理的矛盾がないか検証するための方法



ウィルフリド・セラーズ

「理由の論理的空間」 (the Space of Reasons)



Reference

セラーズは、認識や行動に関する議論において、単に物理的な原因と結果の関係では説明できない、人間の意識的な行為や信念の正当化の側面を強調しています。彼の哲学における「理由の論理的空間」は、これらの行為や信念が相互に関連し、正当化され、必要に応じて修正される体系的な構造を指しています。

この概念を理解するためには、セラーズの二つの異なる空間（「因果の空間」と「理由の空間」）の区別を念頭に置くと良いでしょう。因果の空間は物理的な世界における原因と結果の関係を扱い、理由の空間は人間の思考や行動がどのように正当化され、相互に関連付けられるかを扱います。セラーズは、我々が「意味を理解する」とは、この理由の空間における規則や規範に従うことだと考えました。

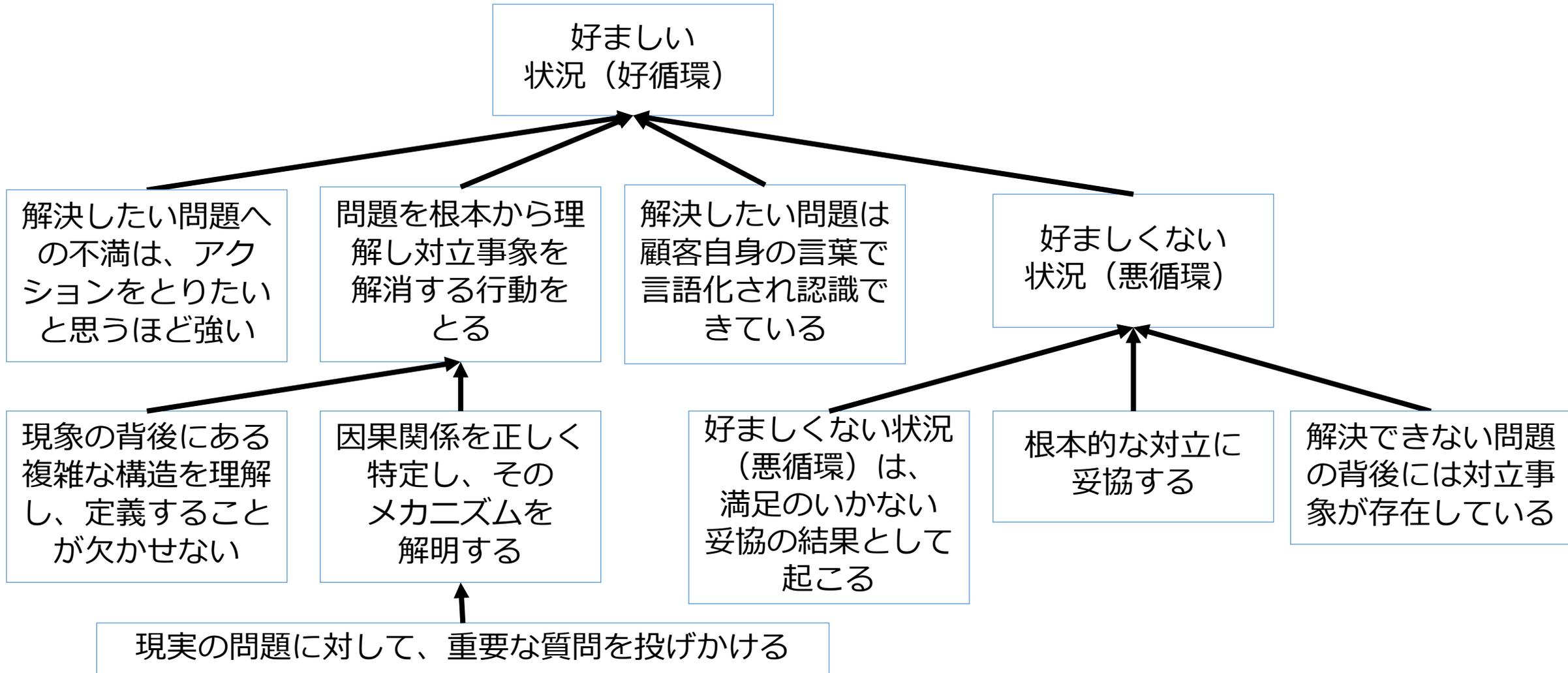
理由と正当化

「理由が存在しているところには常に何らかの種類の正当化が存在している」は、理由が存在する状況では必ずその理由を支える何かしらの正当化が伴っていることを意味します。例えば、誰かが「なぜそうしたのか」と問われるとき、その人は行動や信念を正当化するための理由を提供することが求められます。これは、単に事実を述べるだけではなく、その行動や信念がなぜ適切であるか、あるいはなぜ信じるに足るかを示すことです。

修正の可能性

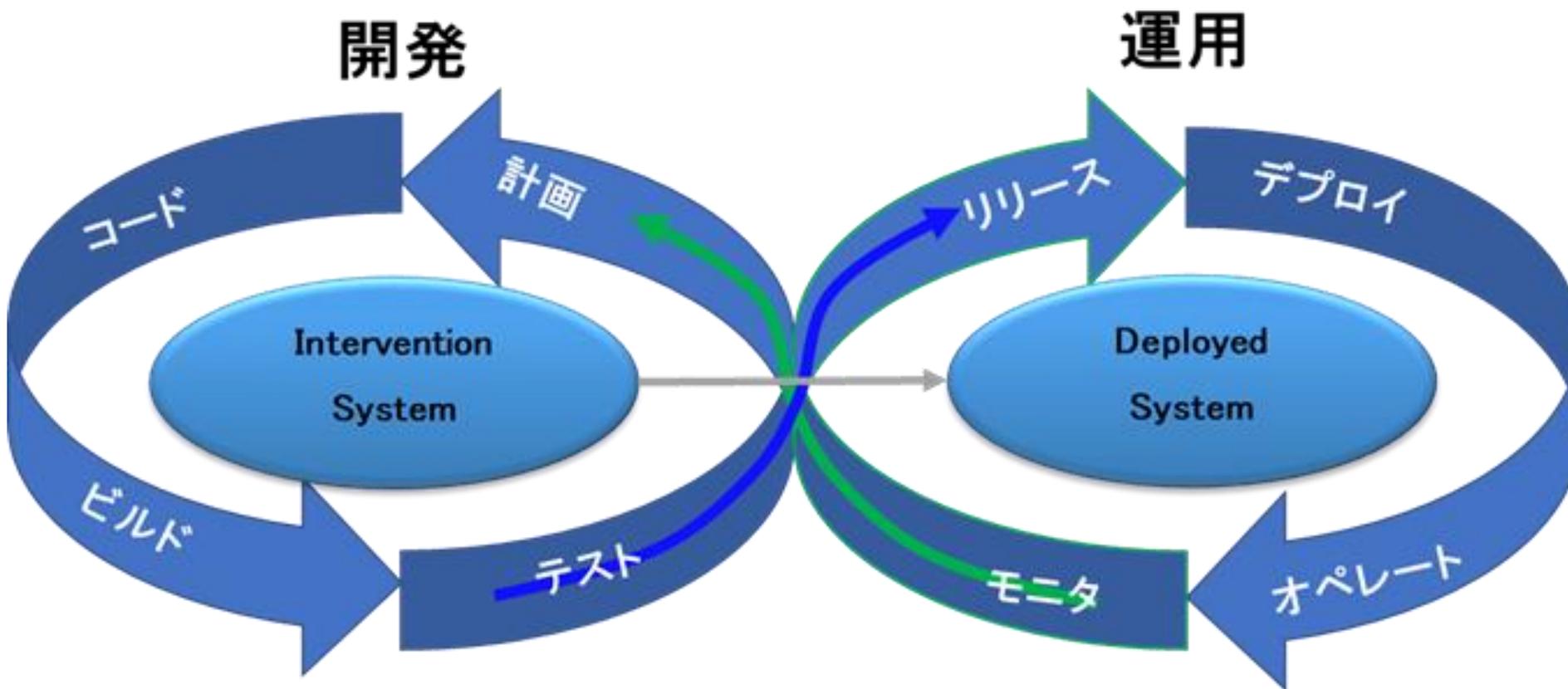
「何かが間違っていた場合の修正の可能性のための余地およびそれらの必要性が存在している」という部分は、理由や正当化には誤りの可能性が常に伴っていることを示しています。つまり、提供された理由や正当化が間違っている場合、それを修正する余地や必要性があるということです。これは、理由の論理的空間においては常に反省と修正が可能であり、また必要であることを意味します。

1. ビジネスモデルを生み出す (4) 仮説を立てる(2/2)



1. ビジネスモデルを生み出す
(5) 検証する

minimum valuable product(mvp)での検証



ダンカン・ワッツ 「複雑なシステムとモデル」

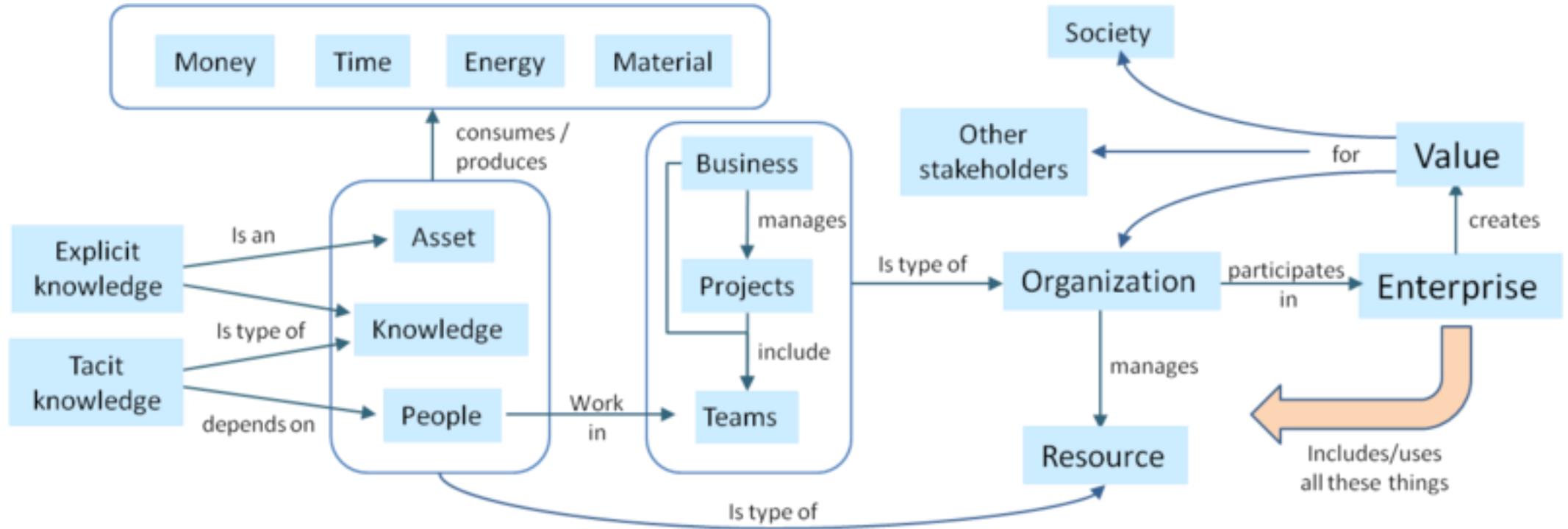


Reference

多数の独立した要素が非直線的な形作用し合うときに複雑性は生まれると一般に受け止められている。複雑なシステムでは、システムの一部におけるわずかな乱れが増幅され、他の部分に大きな影響を及ぼす。

複雑なシステムでは、小さな要因のひとつひとつが予測できない形で増幅されるのであれば、モデルが予測できることは限られている。このため、複雑なシステムのモデルは、単純なものになる傾向がある。単純なモデルの方が好結果を出せるからではなく、どのみち大きく誤る可能性が残っているために、改良してもほとんど意味がないからである。

2. ビジネスモデルに成功をもたらす (1) 経営資源を理解する



Notes:

1. All entities shown are decomposable, except people. For example, a business can have sub-businesses, a project can have subprojects, a resource can have sub-resources, an enterprise can have sub-enterprises.
2. All entities have other names. For example, a program can be a project comprising several subprojects (often called merely projects). Business can be an agency, team can be group, value can be utility, etc.
3. There is no attempt to be prescriptive in the names chosen for this diagram. The main goal of this is to show how this chapter uses these terms and how they are related to each other in a conceptual manner.



出典：SEBOK

2. ビジネスモデルに成功をもたらす (2) 経営資源を見つけ出す

問題解決とは、経営資源の然るべき組み合わせを決めるだけでなく、
そもそも経営資源を見つけ出すこと



ユストゥス・フォン・リービッヒの最小量の法則「樽の例え」は、植物の成長や生態系の健康が最も不足している要素によって制限されるという基本原則を示しています。

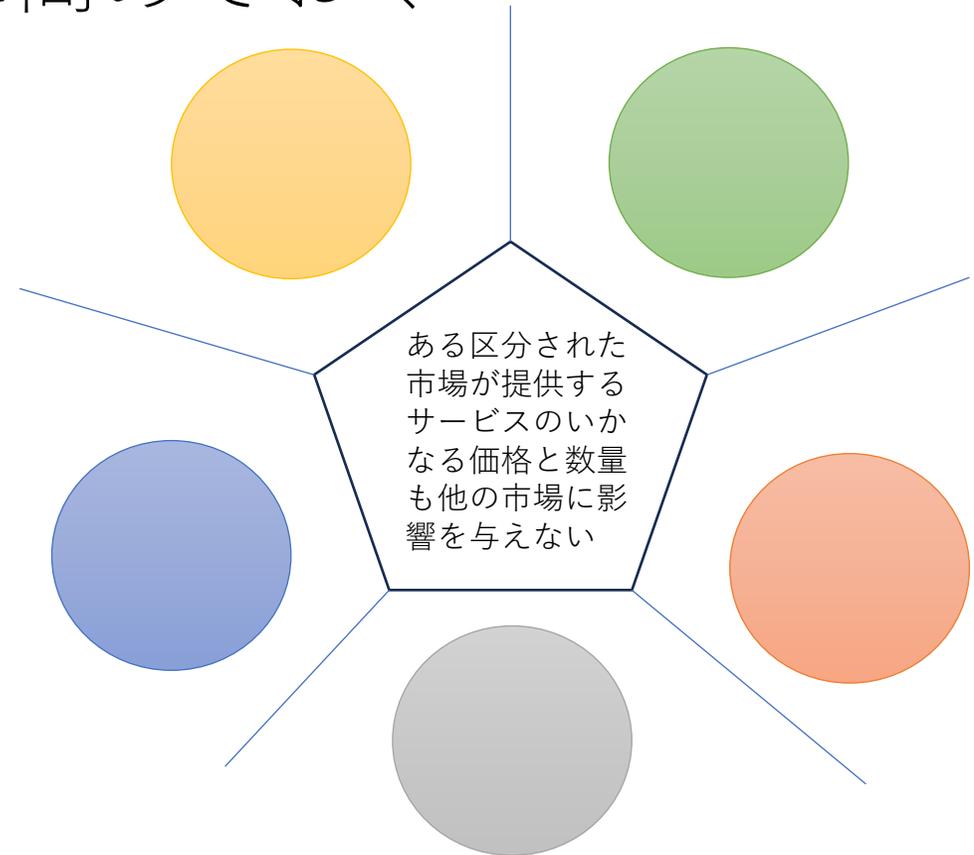
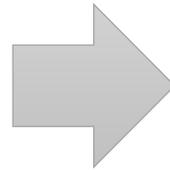
この法則は、農業、環境管理、生態学などの分野で広く応用されており、資源の効果的な管理と最適化において重要な役割を果たしています。

2. ビジネスモデルに成功をもたらす (3) 経営資源の自由度を常に高めておく



リソースセグメンテーション

市場に合わせてリソースをセグメント化してしまうと
市場の変化に柔軟に対応できない
(将来市場がどうなるかわからない)



マーケットセグメンテーション

市場をセグメント化し、自由度を手に入れる

2. ビジネスモデルに成功をもたらす
(4) 経営資源の質を高め連携する(1/2)

パフォーマンス
(成果)

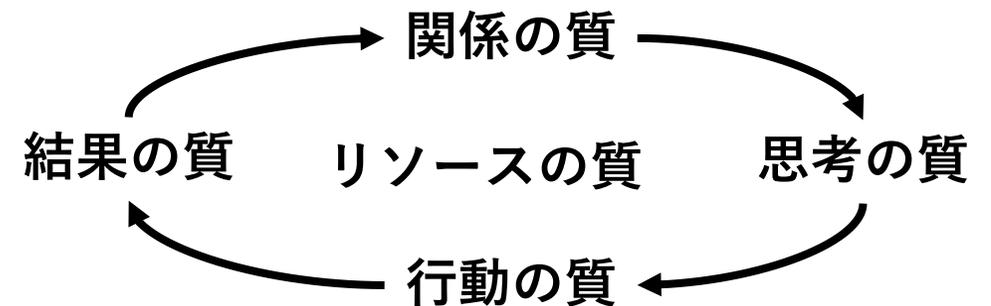
リソースの質

ドメイン (業界・業務)
知識

連携



ビジネス・エコシステム



2. ビジネスモデルに成功をもたらす (4) 経営資源の質を高め連携する(2/2)

システムズエンジニアリングのコンピテンシー分野

コアとなるシステムズエンジニアリングの原則



- ・ システム思考
- ・ ライフサイクル
- ・ ケイパビリティ エンジニアリング
- ・ エンジニアリング全般
- ・ クリティカルシンキング
- ・ システムのモデリングと分析

専門家としてのコンピテンシー



- ・ コミュニケーション
- ・ 倫理とプロフェッショナリズム
- ・ 技術的リーダーシップ
- ・ ネゴシエーション
- ・ チームダイナミクス
- ・ ファシリテーション
- ・ エモーショナルインテリジェンス
- ・ コーチングとメンタリング

技術的なコンピテンシー



- ・ 要求の定義
- ・ システムアーキテクティング
- ・ デザイン・フォー・エックス
- ・ 統合
- ・ インターフェース
- ・ 検証
- ・ 妥当性確認
- ・ 遷移
- ・ 運用とサポート

システムズエンジニアリングのマネジメントコンピテンシー



- ・ 計画
- ・ 監視と統制
- ・ 意思決定マネジメント
- ・ コンカレントエンジニアリング
- ・ ビジネスと企業の統合
- ・ 取得と供給
- ・ 情報マネジメント
- ・ 構成マネジメント
- ・ リスクと機会のマネジメント

統合コンピテンシー



- ・ プロジェクトマネジメント
- ・ 財務
- ・ ロジスティクス
- ・ 品質

出典：incose-se-vision-2035-v1.0_ja

システム思考



Reference

ピーター・センゲ (Peter Senge) は、1990年の著書『The Fifth Discipline (学習する組織)』でシステム思考について詳述しています。以下にセンゲのシステム思考に関する見解を説明します。

1. 全体を見るための規律：システム思考は、個々の部分や要素だけでなく、全体としてのシステムを理解するための規律です。これは、物事を分割して考えるのではなく、全体としての一貫性やパターンを見つけることを重視します。
2. 相互関係を見るためのフレームワーク：システム思考は、単なる個々の要素や事象ではなく、それらの間の相互関係や相互作用に焦点を当てます。これにより、複雑なシステム内での動的な関係性を理解しやすくなります。
3. 発見と診断のプロセス：システム思考は、問題を発見し、根本原因を診断するためのプロセスです。これにより、表面的な症状だけでなく、根本的な問題にアプローチすることが可能になります。
4. 微妙な相互接続性を感知する感性：システム思考は、生命システムの独自性を与える微妙な相互接続性を感知するための感性を養います。これは、システム内の微細な変化や相互作用を捉える能力を高め、システム全体の健全性を維持するために重要です。

2. ビジネスモデルに成功をもたらす (5) 経営資源を特定のセグメントに割り当て強化する

現在から将来にわたって利益を上げ続けることができる

競争優位を発揮できる市場セグメントを
数多く生み出す

もっと変化を！

理由

消費者はひとり一人のニーズに
合ったサービスとこれまで以上の
変化を要求するようになる
(サービスは早く陳腐化する)

行動

特定のセグメントに焦点を当て
そのセグメントに
経営資源を投入し
最適な製品やサービスを
提供する

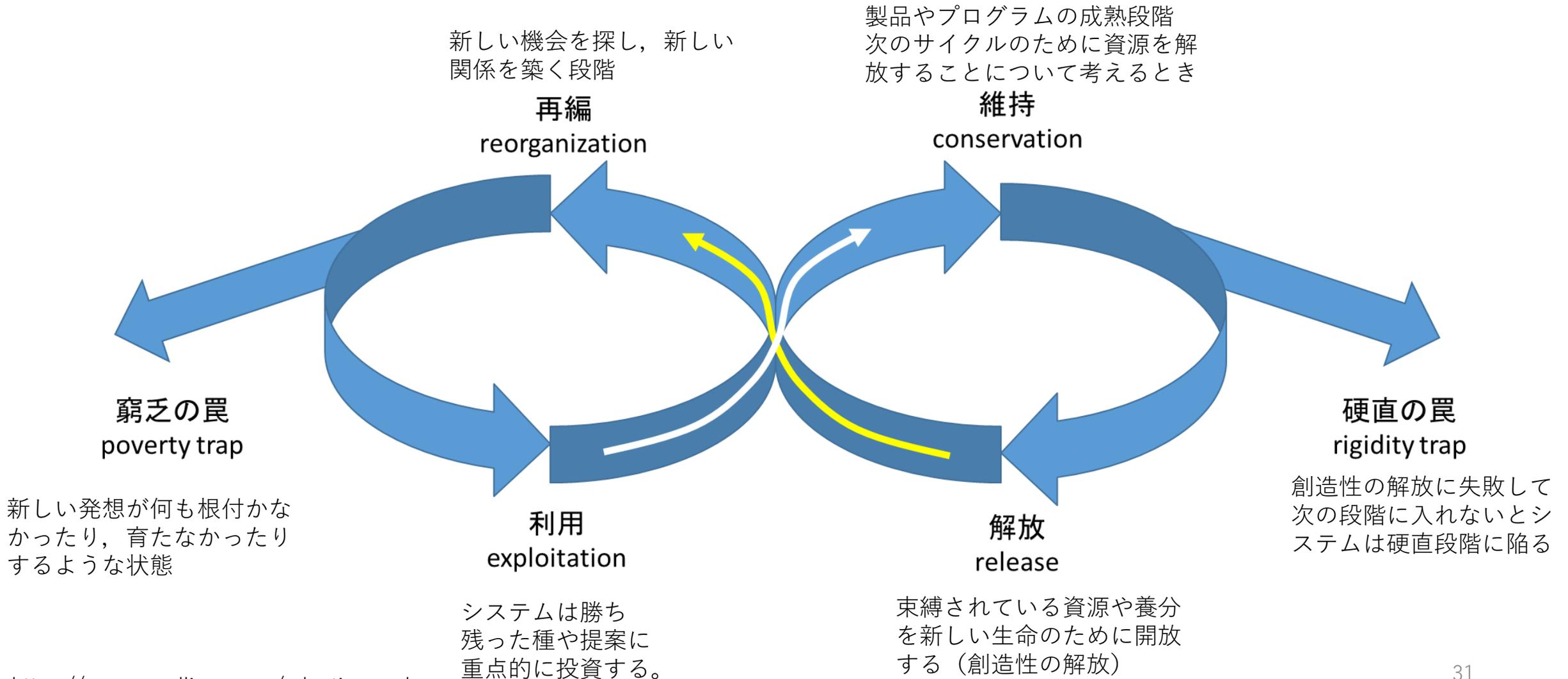
前提

市場はセグメント化されている

市場のニーズは多様である

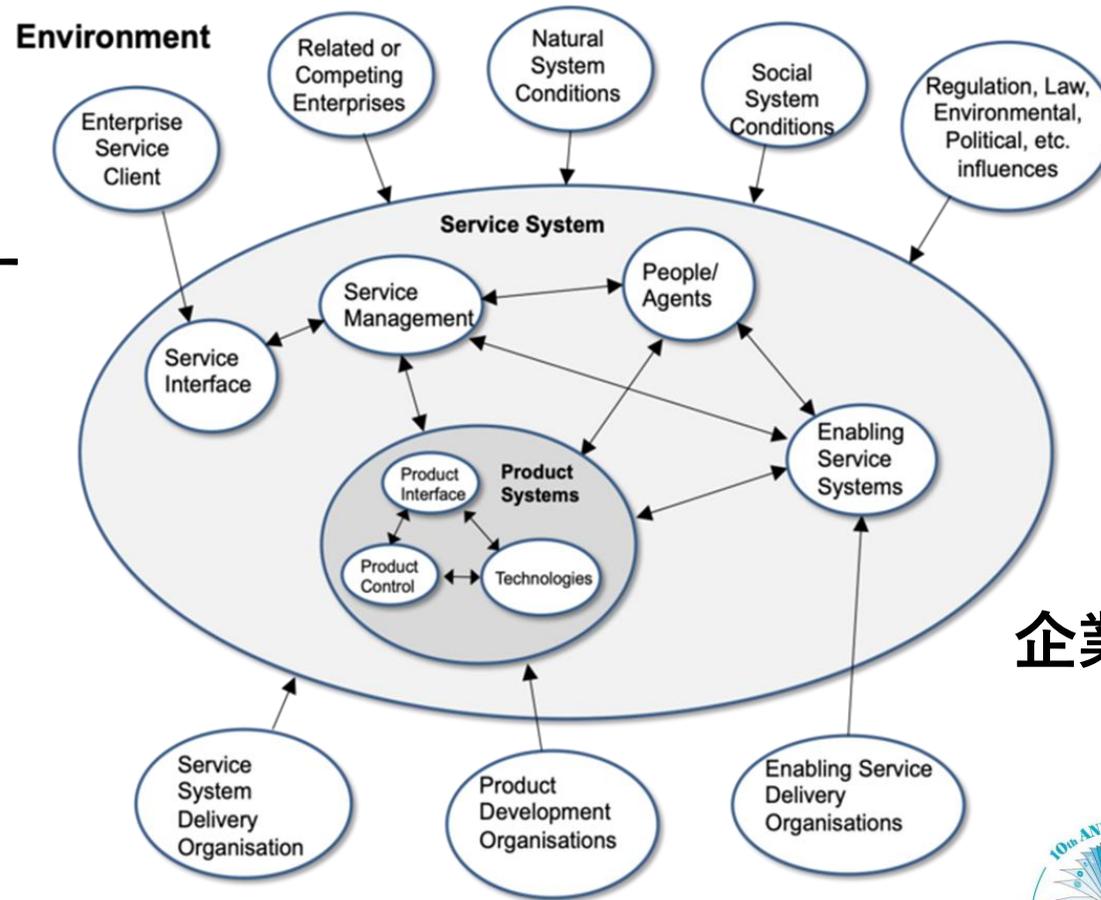
【これまでの行動】 市場があたかも画一的であるかのように捉え、単一価格で製品・サービスを提供することで市場を単一のセグメントと化している

3. ビジネスモデルを駆動する (1) 既存の領域（コンフォートゾーン）から飛び出す



3. ビジネスモデルを駆動する (2) ビジネスエコシステムを構築する(1/4)

もっと変化を起こす



ビジネスエコシステム

ビジネスエコシステムは、生物・植物が生存のために生態系を形成するように、業種や業界といった垣根を越え、共に成長するために企業同士を結びつける試みです。1990年代に「共に成長する企業群」を意味する用語として登場したビジネスエコシステムは、現在では消費者や社会をも内包した、多彩な意味合いを持っています。

変化を実現するために
企業はどんな役割を担うのか
決めなければならない



出典：SEBOK



Reference

システムズエンジニアリングと社会的資本

システムズエンジニアリングは、直接的に社会的資本と同義ではありませんが、社会的資本を構築・強化するための重要な手段といえます。これらの分野は、信頼、ネットワーク、規範の形成に寄与し、組織や社会全体の協力体制を強化することで、社会的資本の増進に貢献します。

協力とコラボレーション:

システムズエンジニアリングでは、多くの専門分野の協力が不可欠です。異なる分野の専門家が協力し合うことで、相互の信頼が醸成され、効果的なコラボレーションが促進されます。

知識の共有:

システムズエンジニアリングのプロセスでは、情報の共有と共同作業が強調されます。これにより、組織内外の知識ネットワークが強化され、全体の知識資本が向上します。

システムの思考:

システムズエンジニアリングのアプローチは、全体の最適化を目指すものであり、これにより組織全体のプロセスや相互関係の理解が深まり、長期的な信頼と協力が促進されます。



Reference

フランシス・フクヤマ 「社会的美德と繁栄の創造」から

社会資本、特に信頼が経済的繁栄や社会の調和にとって重要な役割を果たすことを詳述しています。彼は、信頼と共感が自然に生じるのではなく、社会や経済の発展過程において育まれるものであると主張しています。

フクヤマは、社会資本が社会や経済の発展にどう影響するかを分析し、信頼のレベルが異なる社会や文化の違いを説明しようとしています。高い信頼がある社会では、協力やネットワークが容易に形成され、経済活動が円滑に進む傾向があると述べています。一方、信頼が低い社会では、個々の取引がより多くのコストを伴い、社会全体の発展が阻害される可能性が高いとしています。

3. ビジネスモデルを駆動する

(2) ビジネスエコシステムを構築する(2/4)

ビジネスエコシステム構築の視点



1. 差異から出発する
2. 差異と異なる存在
3. 差異がものごとを際立たせる
4. 同一性の罠を避ける

3. ビジネスモデルを駆動する (2) ビジネスエコシステムを構築する(3/4)

ビジネスエコシステムのプラットフォームとして必要な条件



- 透明性
- 公平性
- セキュリティ
- コミュニケーションの促進
- 協力のインセンティブの提供
- 全体の信頼性と効率性を高める

ヒューゴ・メルシエとダニエル・スパーバー 「Enigma of Reason（理性の謎）」



Reference

「Enigma of Reason」でメルシエとスパーバーが主張しているのは、理性と言語が他人との相互作用、特に説得のために進化してきたということです。理性は、個々の意思決定を改善するためのものではなく、他人を納得させ、協力を得るためのツールとして発展したと考えられます。言語は、複雑で論理的な物語を伝える能力を持ち、それによって社会的な協力を促進し、生存競争において決定的な優位性をもたらしたとされます。

筋道だった物語の伝達:

言語は、筋道だった物語を伝える能力を持っています。これは、他人に自分の考えを理解させ、納得させるために非常に重要です。論理的で一貫した物語を通じて、自分の意見や行動の正当性を説明し、他人の協力を得ることができます。

情報の共有と協力:

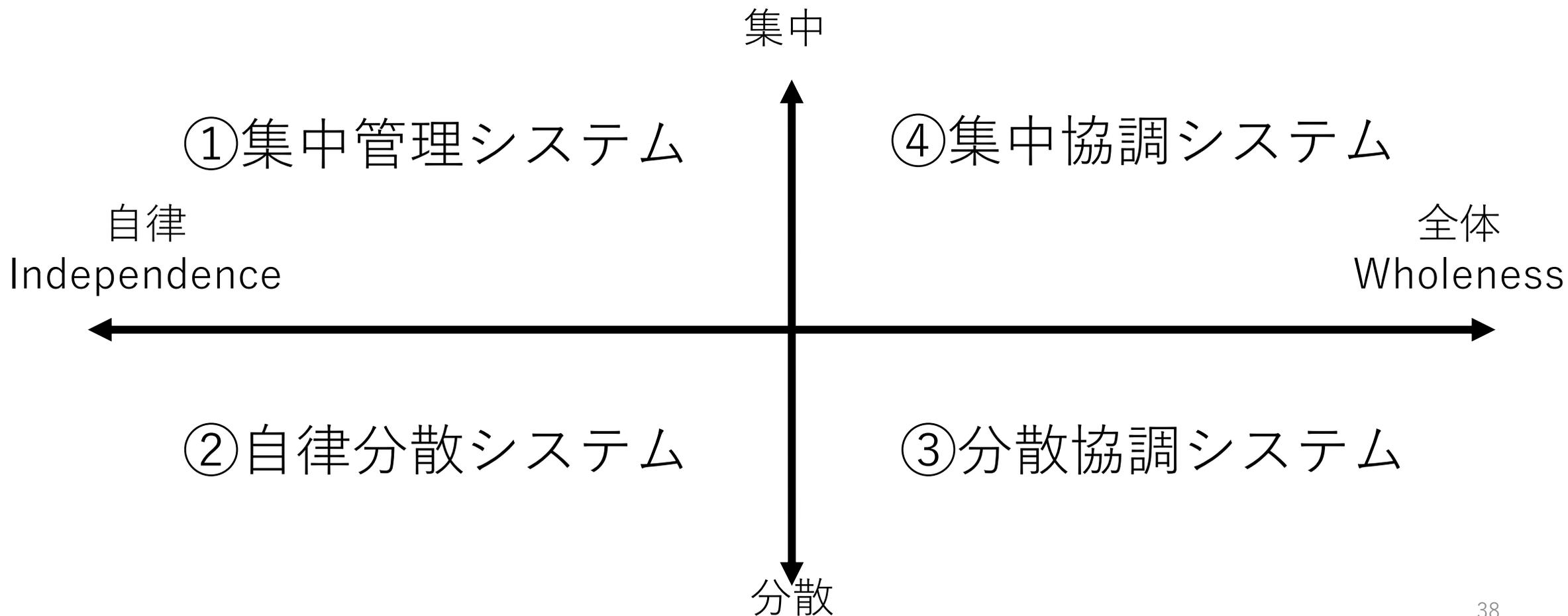
言語を使うことで、ホモサピエンスは情報を迅速かつ正確に共有することができます。これは、狩猟や採集、危険の回避などの生存に直結する活動において、非常に有利です。集団内での情報共有が効率的であれば、より効果的な協力が可能となり、集団全体の生存率が向上します。

社会的結束の強化:

言語はまた、社会的結束を強化する役割も果たします。共通の言語を持つことで、集団内のメンバー間での信頼と連帯感が醸成されます。これにより、集団の協力体制が強化され、生存競争において優位に立つことができます。

3. ビジネスモデルを駆動する
(2) ビジネスエコシステムを構築する(4/4)

ビジネスエコシステムの仕組み (概念化)



チャールズ・ペロウ「ノーマル・アクシデント理論」



Reference

「複雑性と密結合という二つの条件が、さまざまな産業に災害が起こるときの要素だ」と主張するのは、これらの条件がシステムの予測可能性を低下させ、問題が発生した際にその影響を迅速かつ広範に拡大させるためです。これにより、特に複雑で密結合したシステムでは、重大な災害が不可避免的に発生しやすくなるという理解が得られます。

複雑性 (Complexity)

多くの要素: システムが多くの異なるコンポーネントで構成されている。多様な相互作用: コンポーネント間の相互作用が多岐にわたり、予測が困難である。

不確実性:

システムの挙動が非線形であり、小さな変化が大きな影響を与える可能性がある。複雑性が高いシステムでは、システム全体の挙動を完全に予測・制御することが難しくなります。このため、どのような小さな問題や故障でも、システム全体に大きな影響を与える可能性があり、これが災害の一因となります。

密結合 (Tight Coupling)

迅速なフィードバック: ある部分で発生した問題が、他の部分に即座に影響を及ぼす。低い冗長性: 代替手段やバックアップが少ないため、1つの障害がシステム全体に広がる。

順序性: プロセスや手順が特定の順序で行われる必要があり、一部の中断が全体の機能停止につながる。密結合のシステムでは、一度問題が発生すると、それが連鎖的に他の部分に波及し、システム全体が迅速に影響を受けるため、災害が発生しやすくなります。

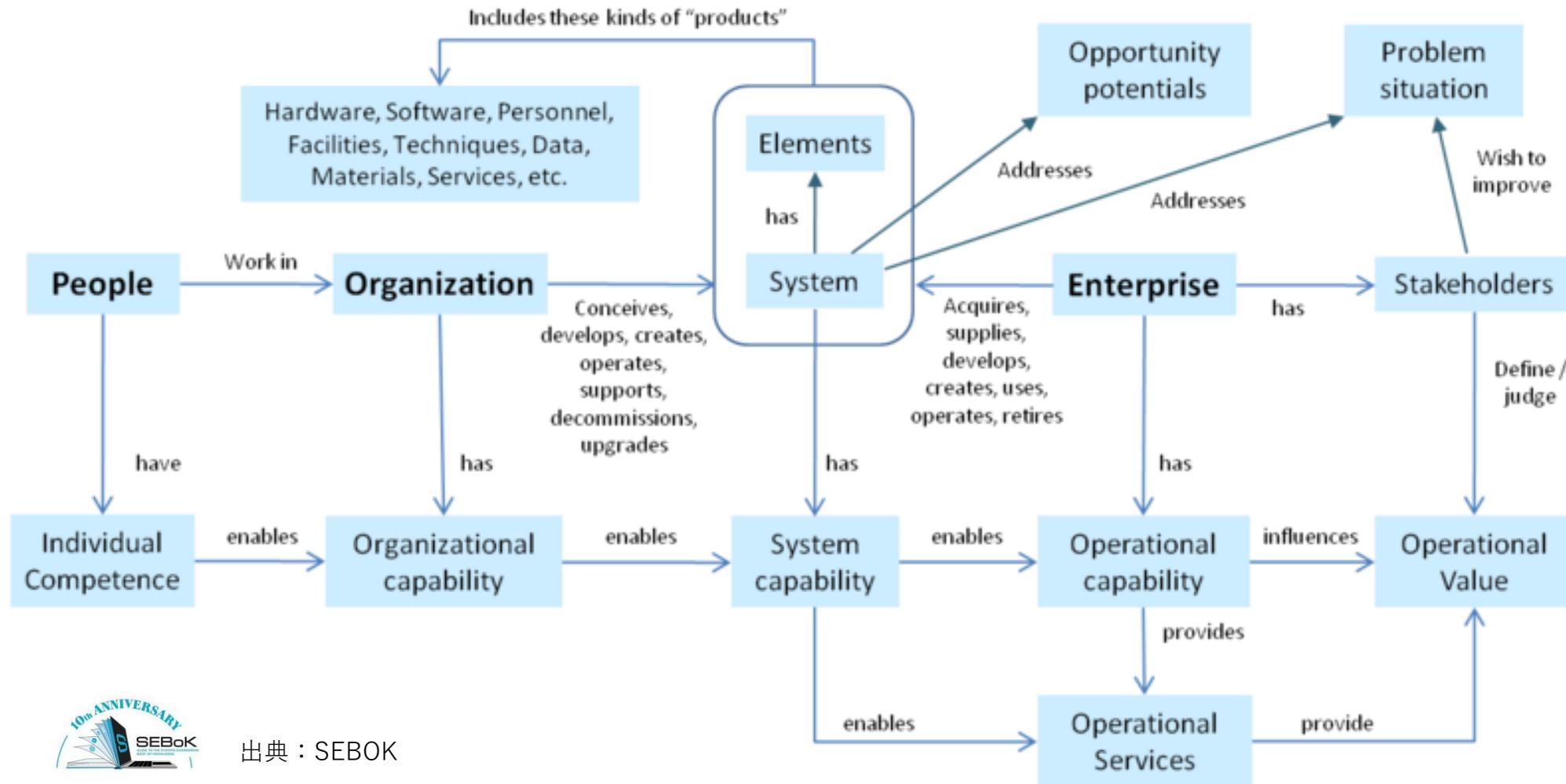
複雑性と密結合の相互作用

複雑性と密結合の二つの条件が相互に作用することで、次のような状況が生まれます:

不可避な事故: 複雑なシステムでは、全ての相互作用を予測し、管理することが事実上不可能であるため、事故が不可避免的に発生する。

事故の拡大: 密結合によって、1つの小さな事故が迅速に他の部分に影響を与え、全体的なシステムの崩壊を引き起こす可能性がある。

3. ビジネスモデルを駆動する (3) 資源の能力をマネジメントする



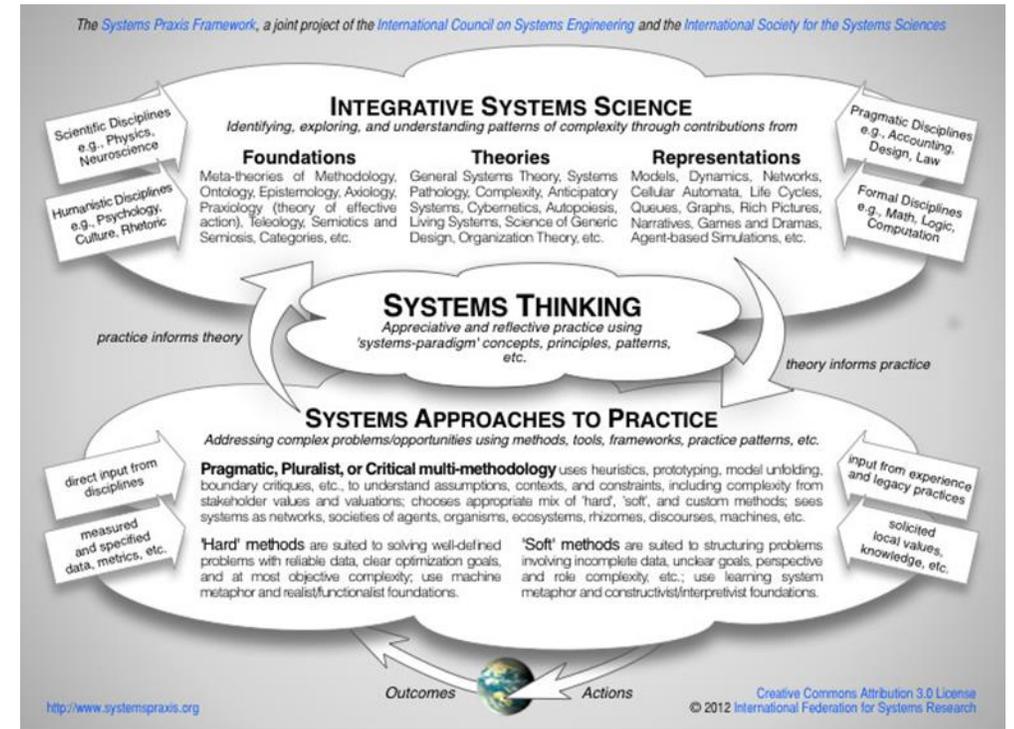
出典：SEBOK

Intended Blank

第二部

システムズエンジニアリング.... SO WHY? SO WHAT?

1. その意味
2. 駆動する力
3. 規範的な力
4. ディシプリン
5. 適用の滞り
6. コンテキストの理解
7. その意思決定



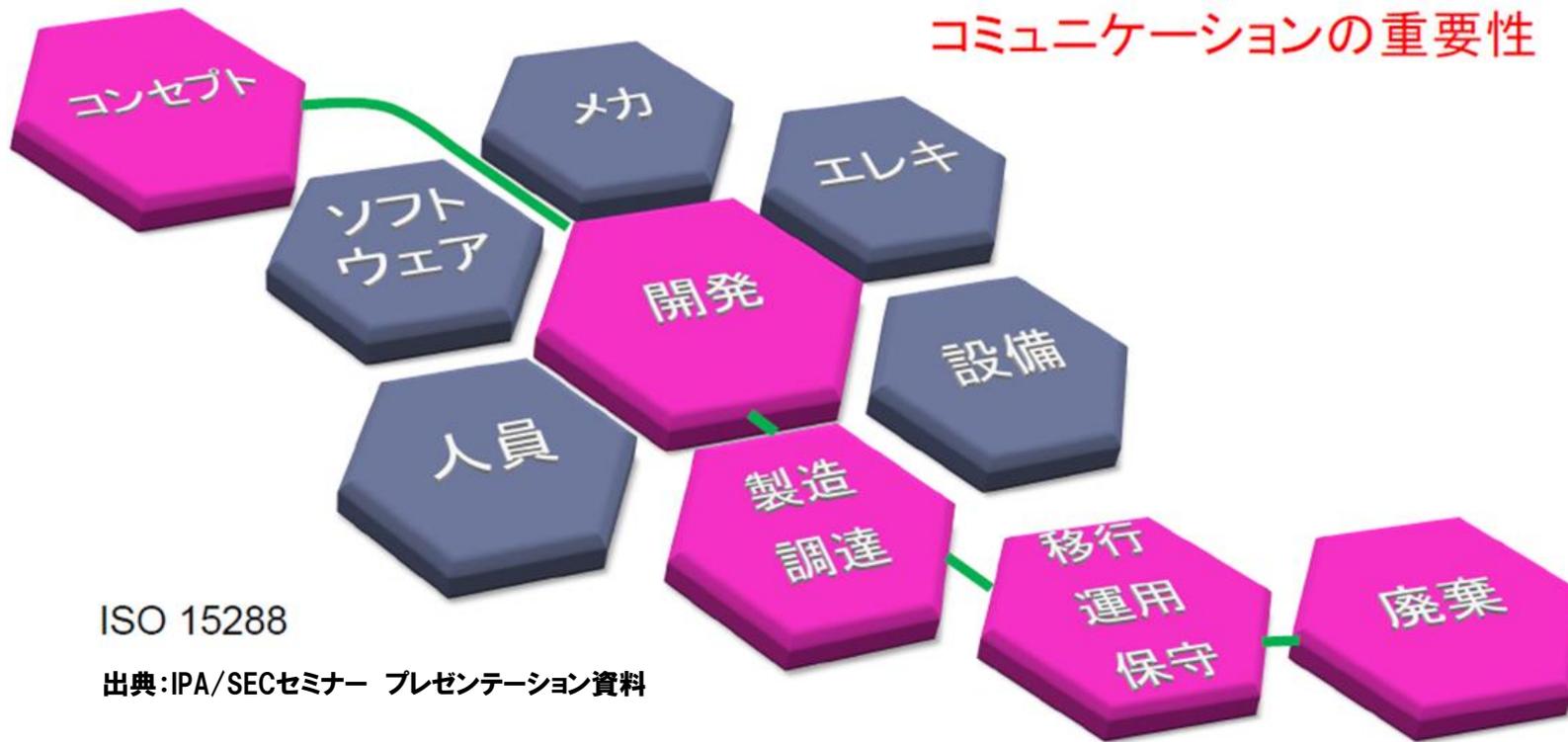
出典：SEBOK



1.システムズエンジニアリングとその意味(1/4)

システムを**成功裏に実現**するための**複数の分野**にまたがる
アプローチおよび手段

↓
コミュニケーションの重要性



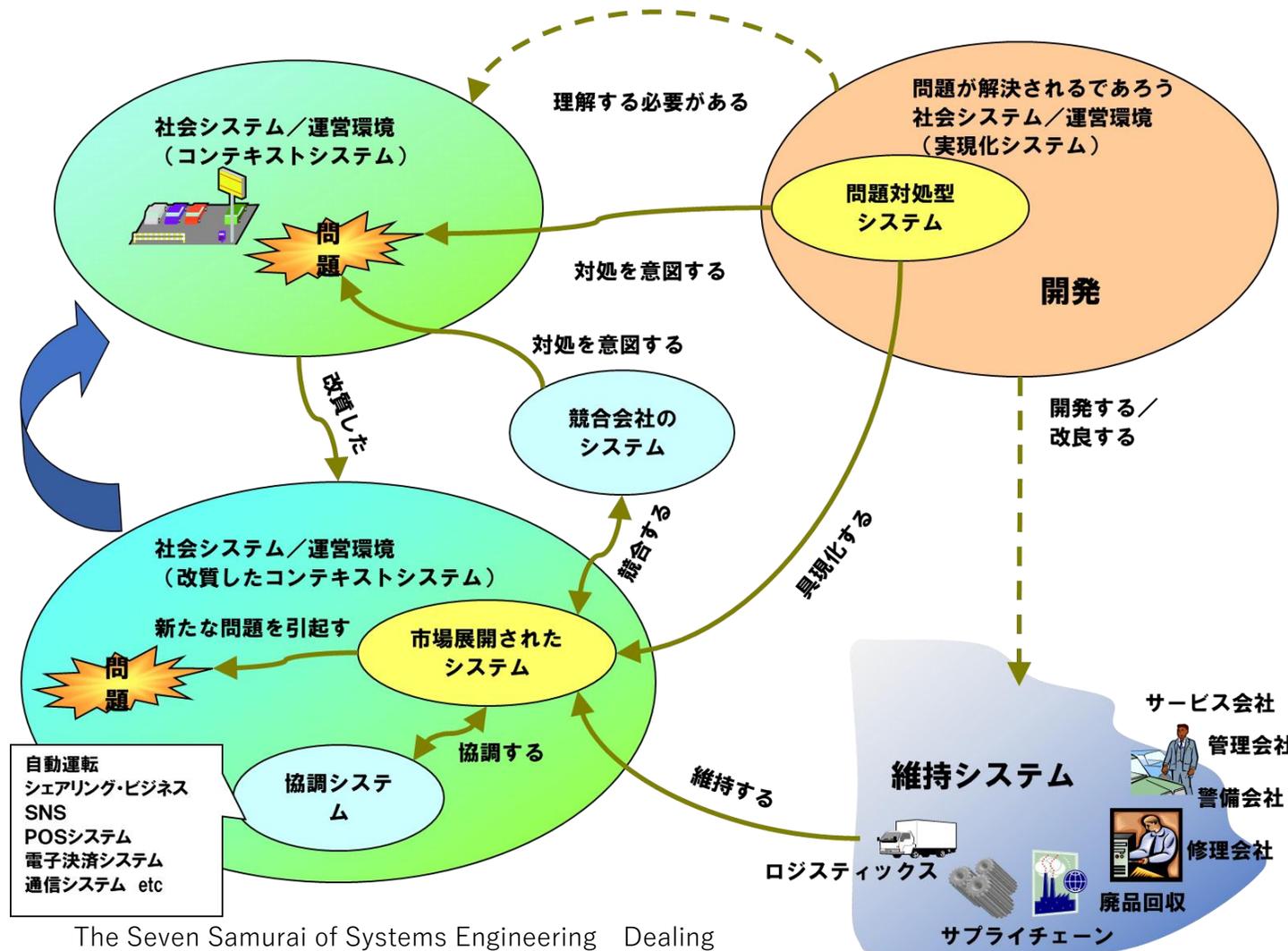
ISO 15288

出典:IPA/SECセミナー プレゼンテーション資料

1. **目的としての意味:** システムズエンジニアリングを教えるとき、その適用の誤りを訂正するとき、目的としての意味が強調されます。
2. **手段としての意味:** システムズエンジニアリングの基本概念や原則を明確にし、プロセスの重要性や価値を示すとき、手段としての意味が強調されます。
3. その「意味」を理解し、それを基に行動することで、システムズエンジニアリングの効果的かつ効率的な実践が可能になります。

「資料1_システムズエンジニアリング」にて概説

1. システムズエンジニアリングとその意味(2/4)



コンテキストの重要性:

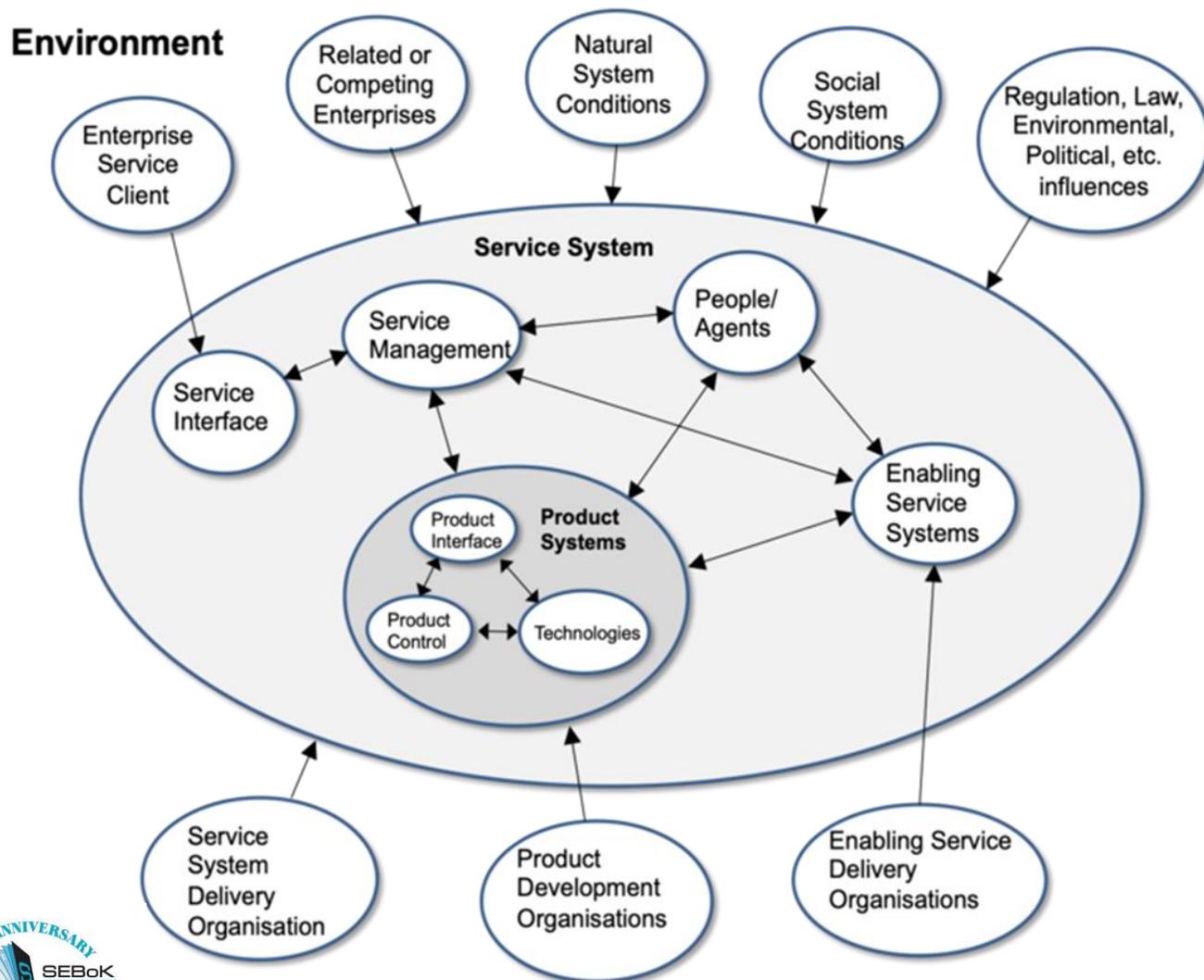
システムズエンジニアリングは、特定のコンテキストや目標に基づいて展開されるプロセスです。

そのため、その意味や概念をコンテキストから切り離して説明することは、その本質を正しく理解する上で不十分です。

コンテキストを無視すると、システムズエンジニアリングが解決すべき具体的な問題や要件が欠落し、その働きを正確に把握することが困難になります。

「資料3_システム思考」にて概説

1. システムズエンジニアリングとその意味(3/4)



多様性と柔軟性:

システムズエンジニアリングは、さまざまな業態や業界で異なる形で展開されます。

そのため、業態や業界に依存せず、一般的な概念や意味だけでシステムズエンジニアリングを説明しようとすると、その多様性や柔軟性を十分に反映して示すことができません。

「資料3_システム思考」にて概説



1. システムズエンジニアリングとその意味(4/4)

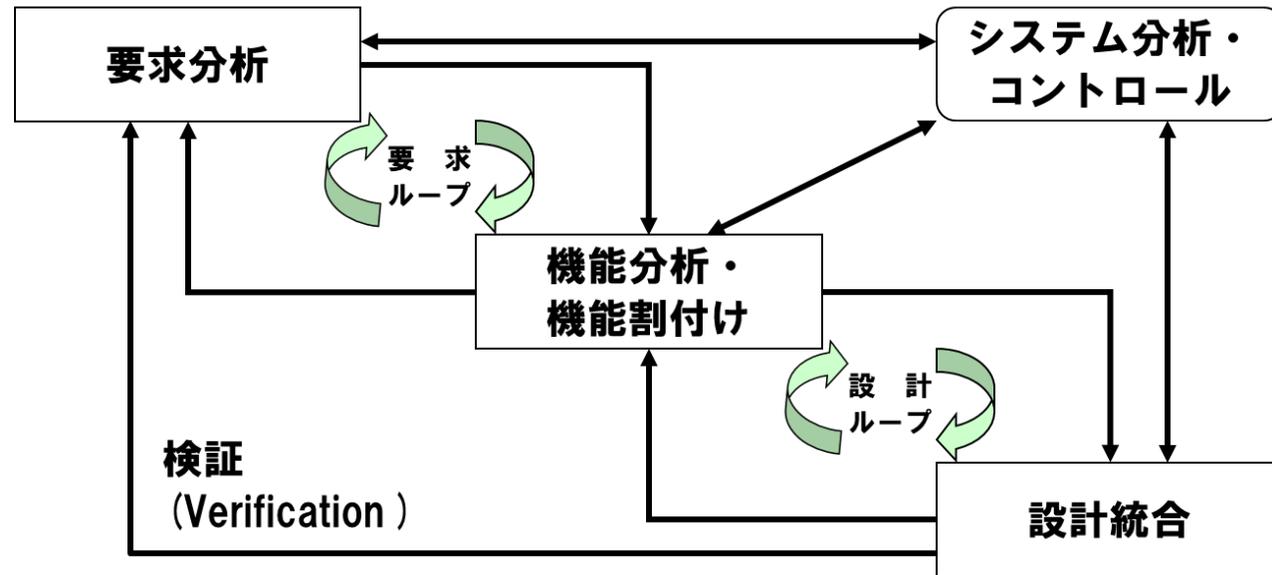


Fig SE Management Process (概要)

実践と理論の結びつき:

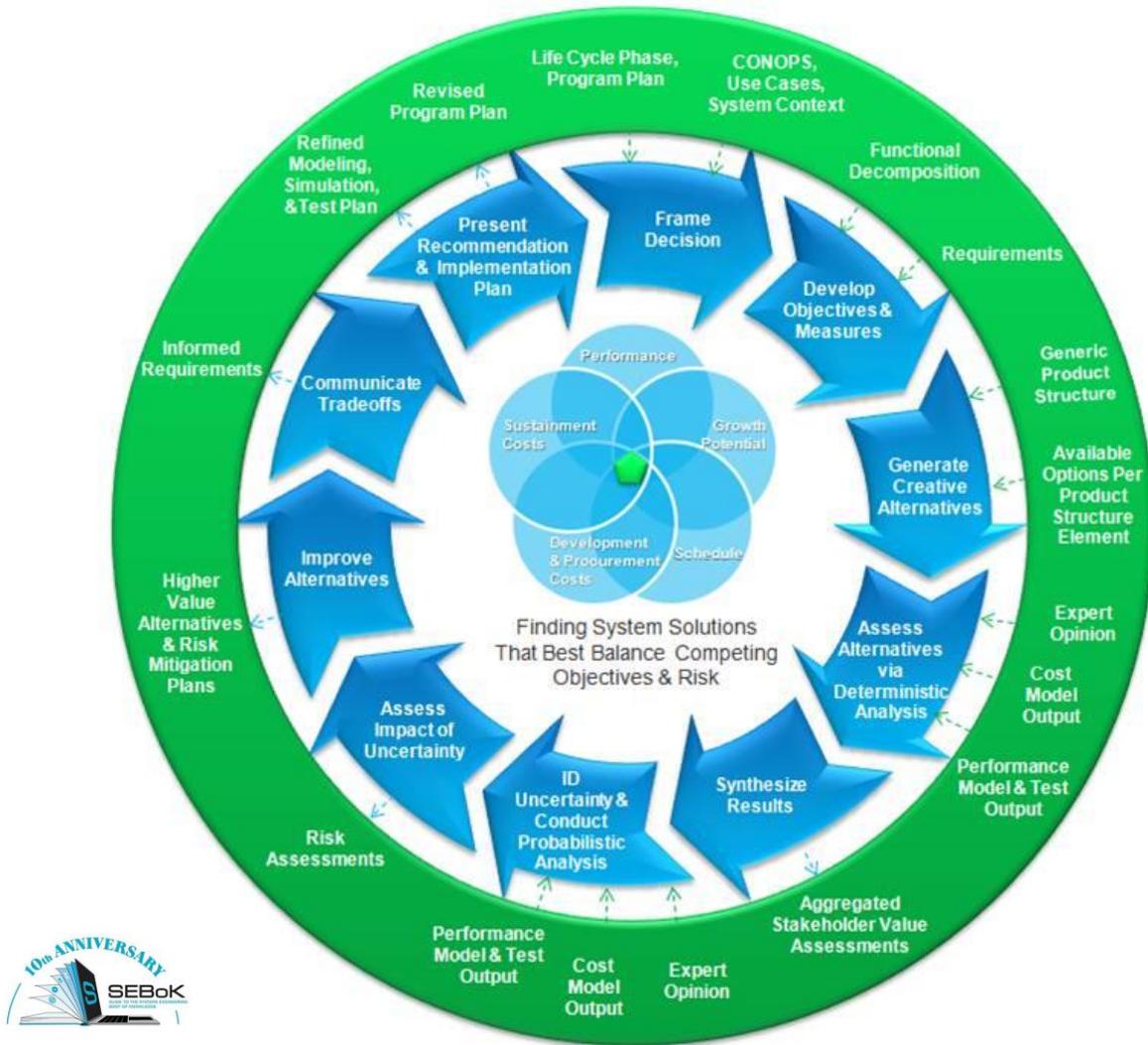
システムズエンジニアリングのプロセスは、その実践と密接に結びついています。

そのため、理論的な説明だけでなく、実践的な側面も考慮する必要があります。

コンテキストから切り離れた一般的な概念だけでは、実践的な側面が欠落し、システムズエンジニアリングの本質が理解できません。

「資料5_SEプロセス」にて概説

2.システムズエンジニアリングを駆動する力(1/3)



システムズエンジニアリングのプロセスは、組織やエンジニアの意志、経営者からのトップダウンの意思決定によって、駆動されることがあります。

実際には単なる意志の力だけでなく、試行錯誤や失敗から生まれる学びや洞察によっても駆動されます。

Figure . Decision Management Process (INCOSE DAWG 2013).出典 ; SEBOK

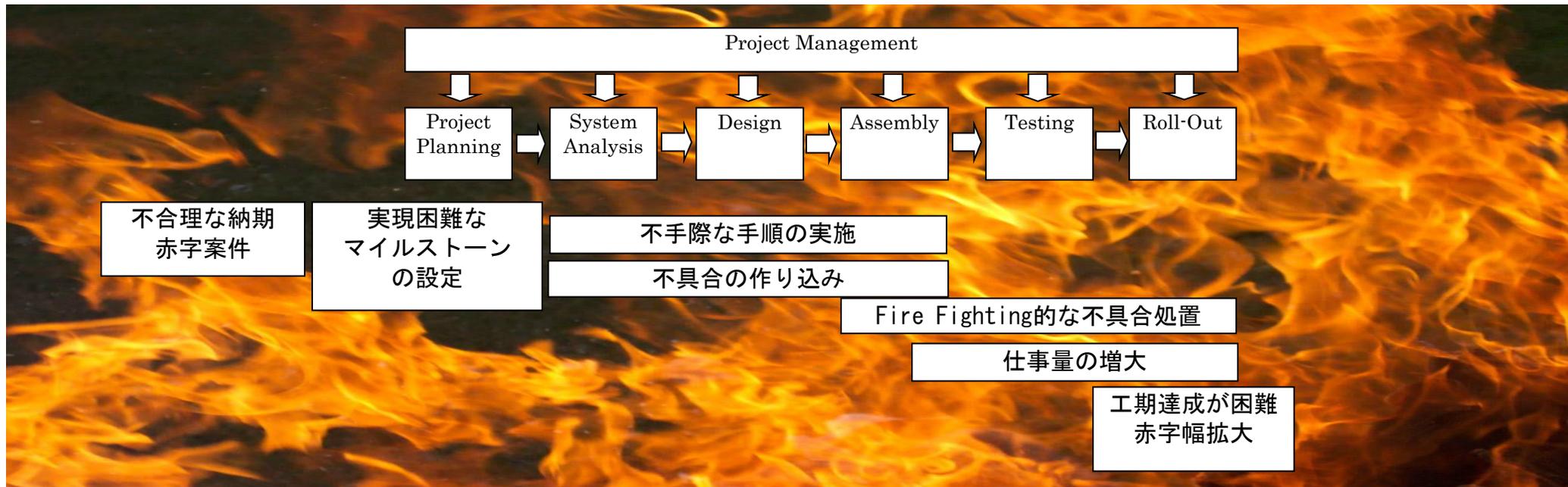
2. システムズエンジニアリングを駆動する力(2/3)

意志の力とプロセスの関係:

意志や意欲は、システムズエンジニアリングにおいて重要な要素です。

試みと失敗からの学び:

システムズエンジニアリングでは、試みや失敗からの学びが重要です。失敗した場面から得られた洞察や教訓が、プロセスの改善や進化に繋がります。このような試みと失敗から生まれる学びが、意志の力と同様にプロセスを駆動します。

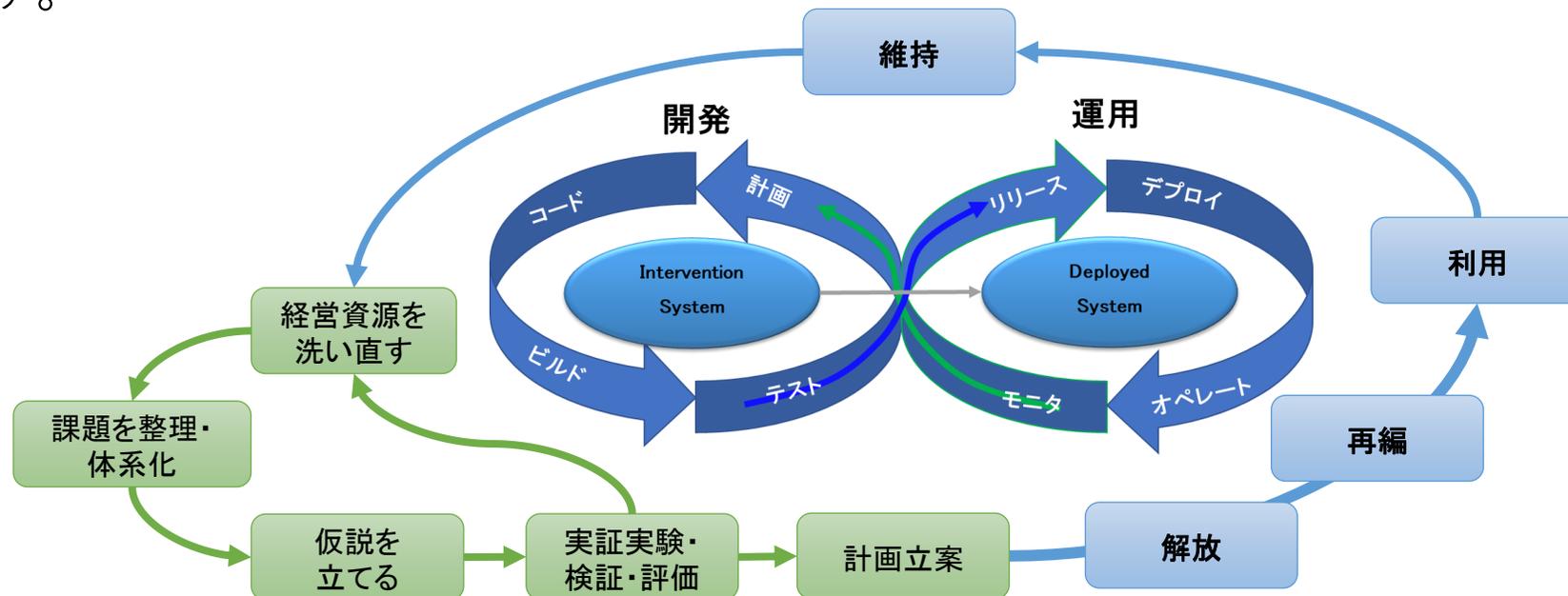


2.システムズエンジニアリングを駆動する力(3/3)

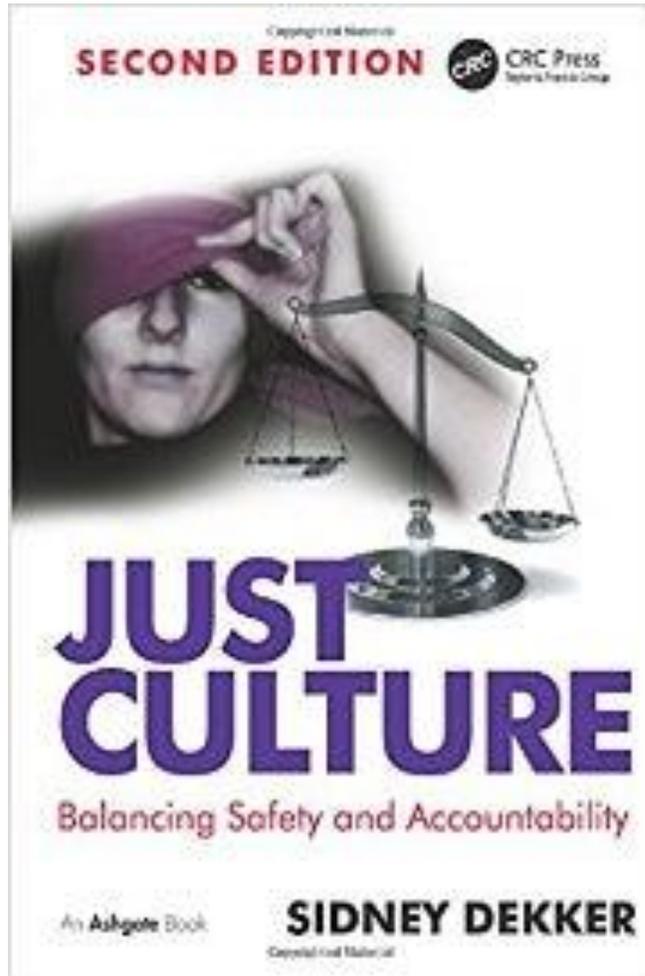
Proof of Concept (POC) の重要性:

Proof of Concept (POC) : 新しいアイデア、理論、技術、または手法が実現可能であり、実際に期待される成果を生み出せるかどうかを示すための実験的なプロジェクトや試作のことを指します。

Proof of Concept (POC) などの活動は、システムズエンジニアリングを実践しようとする意思が顕著になる場面です。POCを通じて、アイデアやコンセプトを実証し、実際の問題や課題に対する解決策を模索します。このような活動は、意志の力を具体的な行動に結び付ける重要な段階となります。



3. システムズエンジニアリングの規範的な力



規範的な力の働き:

システムズエンジニアリングでは、規範や基準が設定され、それに従って行動することが重要です。これによって、プロジェクトや組織が効果的に機能し、目標に向かって進むことができます。しかし、順調に進行している場合は、この規範的な力が目立たないことがあります。

混乱やミスの際の対応:

システムズエンジニアリングにおいては、混乱やミスが発生したときには、規範や基準に従うことが重要です。これによって、問題が解決され、プロジェクトの進行がスムーズになります。

規範的な力は、こうした混乱やミスの際にピンポイントで働くことがあります。

自然な従順:

一方で、順調に進行している場合は、規範的な力が目立たないかもしれません。このような場合、プロジェクトや組織のメンバーは、自然に規範や基準に従って行動することがあります。

規範的な力が必要とされるのは、問題や混乱が発生したときであり、その時に働くことで効果的な対応が可能になります。

4. システムズエンジニアリングのディシプリン



「ディシプリン（規律や方法論）」という語は、システムズエンジニアリングにおいて基本的な原則やルールを教えるときや、違反をたどすときに使用されます。

問題なくシステムズエンジニアリングの適用が行われている場合でも、実際にはディシプリンが遵守されていない場合もあります。

ディシプリンはプロジェクトの成功に不可欠であり、適切な適用が重要です。

1 SYSTEMS ENGINEERING HANDBOOK SCOPE

1.1 PURPOSE

This handbook defines the discipline and practice of systems engineering (SE) for students and practicing professionals alike and provides an authoritative reference to understand the SE discipline in terms of content and practice.

(出典；INCOSE発行 SYSTEMS ENGINEERING HANDBOOK FOURCE EDITION)

5.システムズエンジニアリング適用の滞り

システムズエンジニアリングの適用が滞ったり、問題が発生したりするような状況では、システムズエンジニアリングの意味やディシプリンが特に重要となります。

これらは、利害関係者に迷いや、混乱している場合に、方向を示し、プロジェクトが目標に向かって進む手助けをします。

一方、順調に進行している場合は、意味やディシプリンが自然に遵守されているとみなし、それほど強調する必要がないかもしれません。

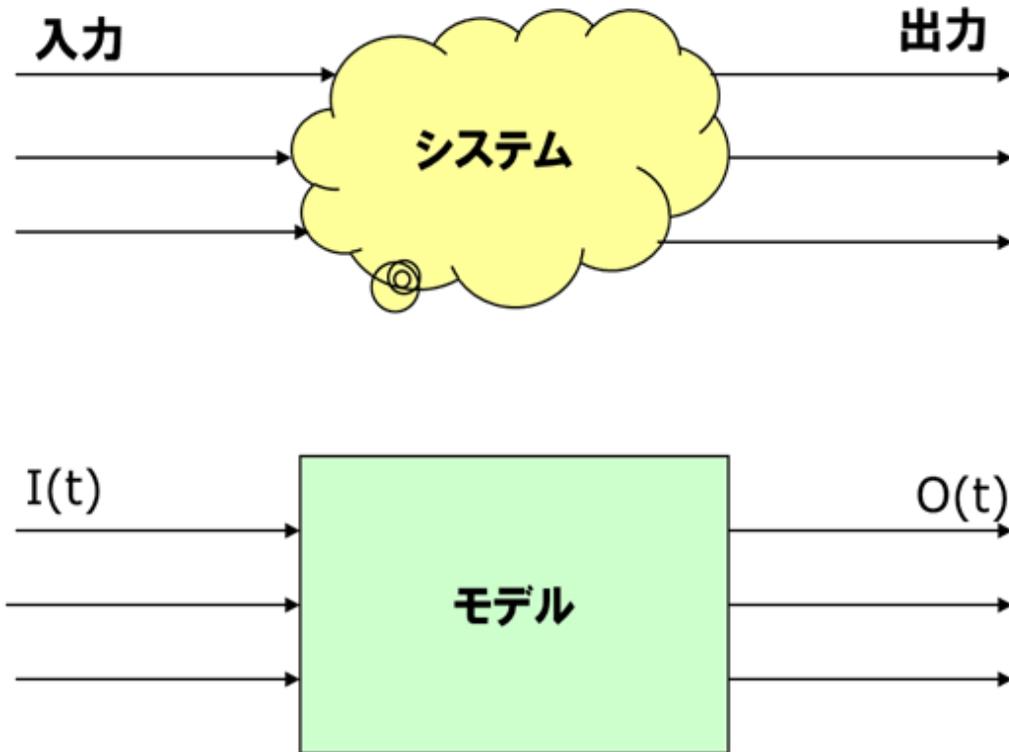


最大の障害は、人の心にある不安

- ◆ この方針で良いか？
- ◆ 他社に遅れをとっているのでは？
- ◆ 本当にできるか？

6. コンテキストの理解(1/5)

モデル

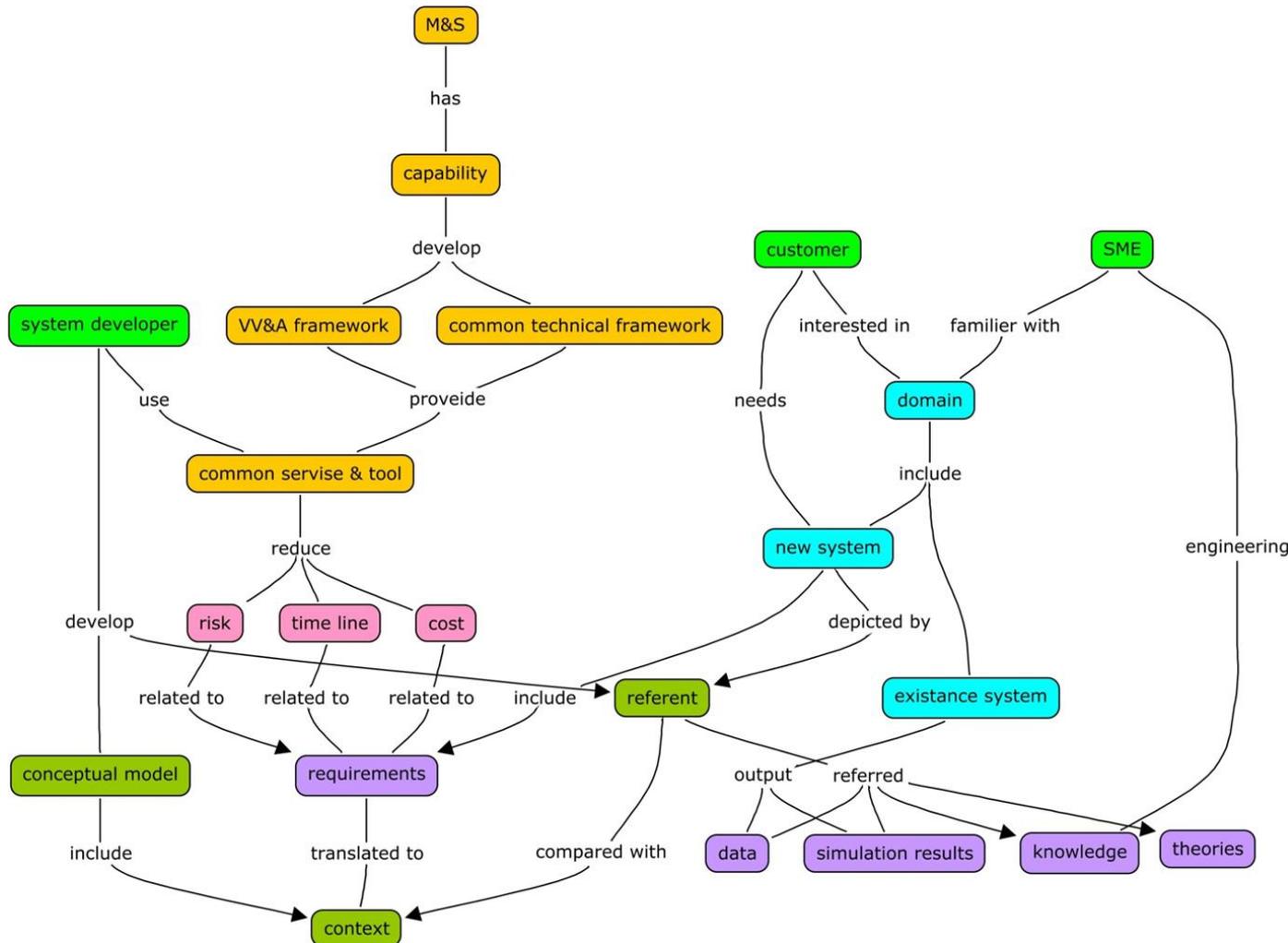


モデルや代理物がない場合、事象を再現するためには現地や現物を使用する必要があります。しかし、この場合、様々な可能性を試すことが難しくなります。なぜなら、現地や現物を使用する場合、実際の状況や条件の変更が容易ではなく、結果を予測することも難しいからです。

一方、モデルや代理物を使用することで、事象を再現する際に様々な可能性を試すことができます。モデルを用いることで、事象やシステムの振る舞いをシミュレートし、異なる条件やパラメータを変えて結果を予測することが可能です。これにより、現実の状況や条件を再現する際には難しかった様々なシナリオや仮説を検討することができます。

6.コンテキストの理解(2/5)

オントロジー



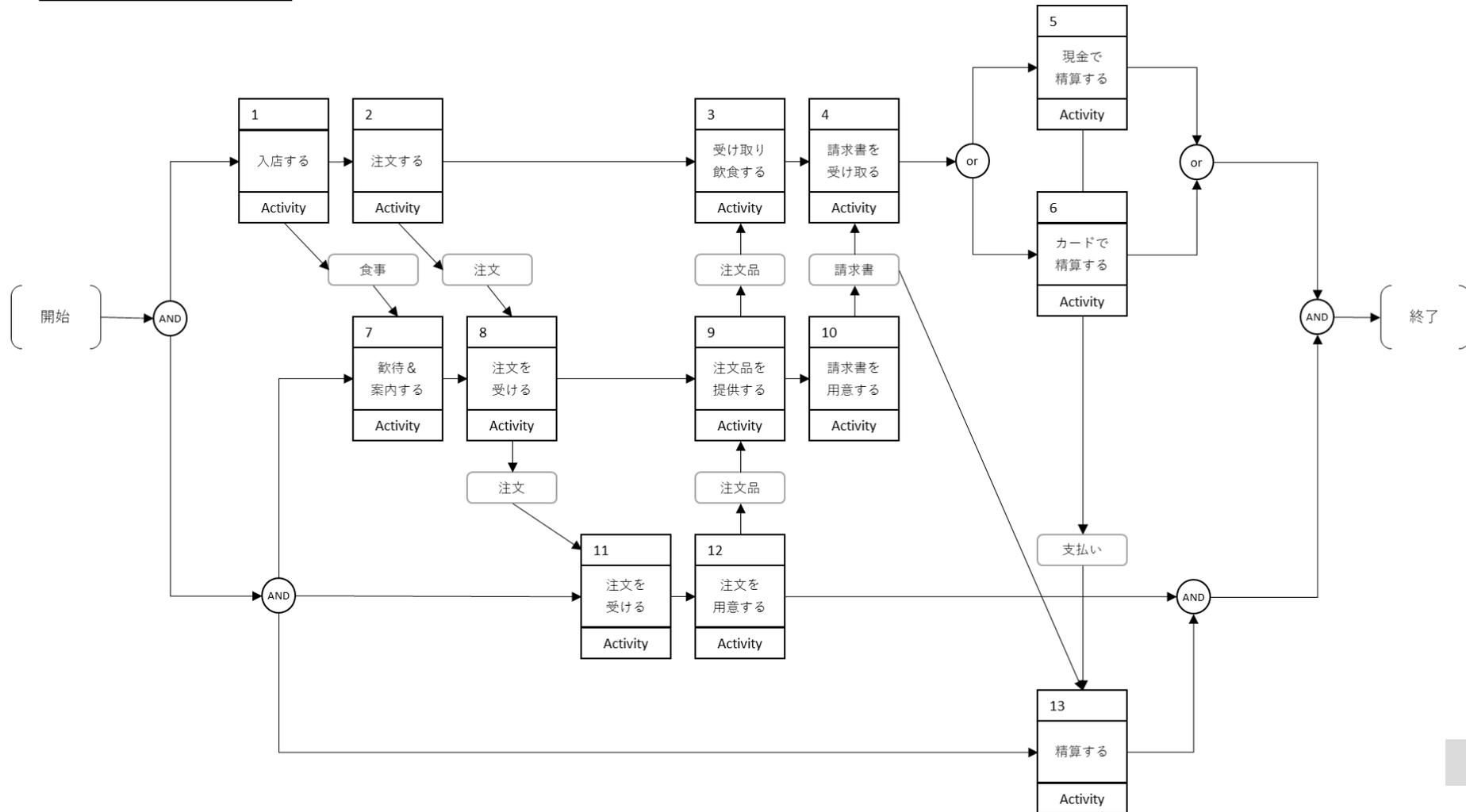
システムズエンジニアリングのコンテキスト理解は非常に複雑であり、組織やプロジェクトの複雑な相互作用、要素やプロセスを考慮に入れる必要があります。

システムズエンジニアリングのコンテキスト理解は、しばしば抽象的であり、概念的な要素やプロセスに焦点を当てます。

「資料Appendix C ツール」にて概説

6.コンテキストの理解(3/5)

ダイアグラム



システムズエンジニアリングにおけるダイアグラムやモデルは、直接に利害関係者の反応を促す有効な手段です。

しかし、その効果的な利用には、それを解釈し、意味を理解する過程が不可欠です。

ダイアグラムやモデルの視覚的・直感的な性質が迅速な反応を促す点は重要ですが、理論的には解釈や意味理解のプロセスも考慮する必要があります。

「資料Appendix C ツール」にて概説

6.コンテキストの理解(4/5)

システムを単にシステムを構成する要素に分解して説明することは、しばしばその全体像を捉えることには不十分であると考えられます。以下にその理由を示します。

相互作用と相関関係の重要性:

システムは、その要素間の相互作用や相関関係によって機能します。単に要素を分解して個々の特性や機能を説明するだけでは、システム全体がどのように機能し、目的を達成するののかについての情報が欠落しています。システムの意味は、その要素間の相互作用や相関関係によってもたらされます。

組織化された構造と機能:

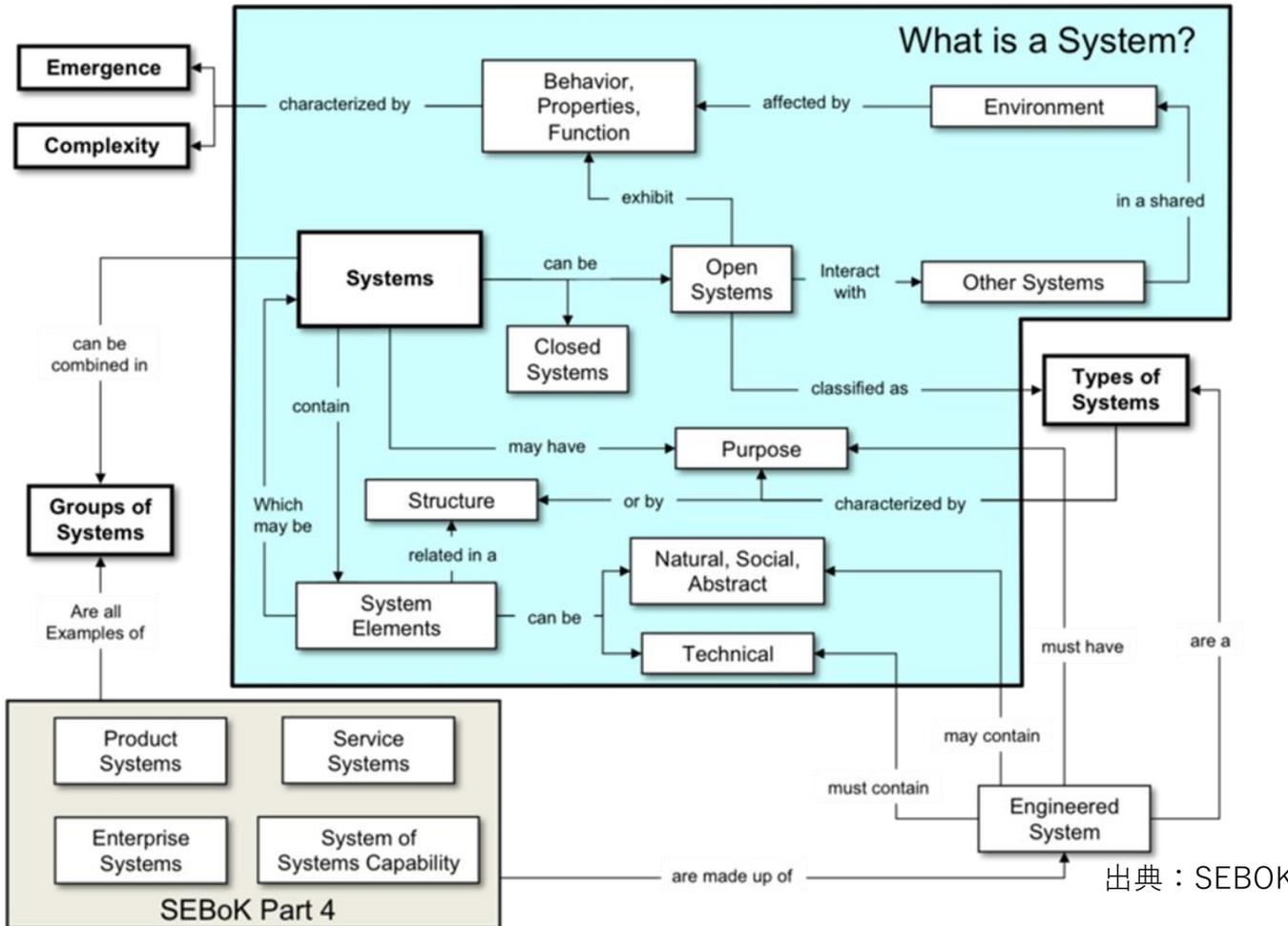
システムは、単なる要素の集合ではなく、組織化された構造と機能を持っています。システムを構成する要素は、特定のパターンや構造に従って組み合わされ、特定の機能を果たすように配置されます。そのため、システムの意味を説明するには、要素だけでなく、それらの組織化された構造や機能も考慮する必要があります。

エマージェントな特性:

システムは、要素の単純な組み合わせ以上の特性を示すことがあります。これをエマージェントな特性と呼びます。要素単独では現れないシステム全体の特性や振る舞いは、要素の組み合わせや相互作用から生じるものです。そのため、システムの意味を理解するには、要素だけでなく、それらが組み合わさって生じるエマージェントな特性も考慮する必要があります。

6.コンテキストの理解(5/5)

相互作用とエマージェンス



出典：SEBOK

今まで見えなかったつながりを見つけ出し、そのつながりの中に存在する価値を新たに定義し、実現することが価値創造です。しかし、ものごとの相互作用から生まれるエマージェント・プロパティと呼ばれる、予期せぬ振る舞いをコントロールできなければ、持続的に価値を生み出すことは困難となり、バブル（泡）と消え去ってしまいます。

エマージェンスは、相対的にミクロの一見ランダムに見える活動の上にマクロ的な現象や秩序が出現するという意味です。出現したマクロ的な秩序はその基底をなすミクロの活動に関する情報だけでは完全に予想・記述できないものであるとされます。部分の性質の単純な総和にとどまらない性質が、全体として現れることです。局所的な複数の相互作用が複雑に組織化することで、個別の要素の振る舞いからは予測できないようなシステムが構成されます。

7.システムズエンジニアリングにおける意思決定(1/2)

意思決定はアクティビティ:

意思決定は、ある状況や選択肢の中から最適な選択を行うためのプロセスです。これは単なる目標設定ではなく、積極的な行動や判断を伴うアクティビティです。したがって、意思決定自体がアクティビティであり、目標を達成するための重要な段階となります。

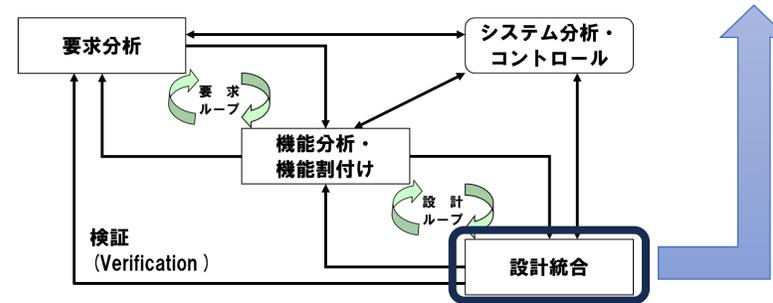
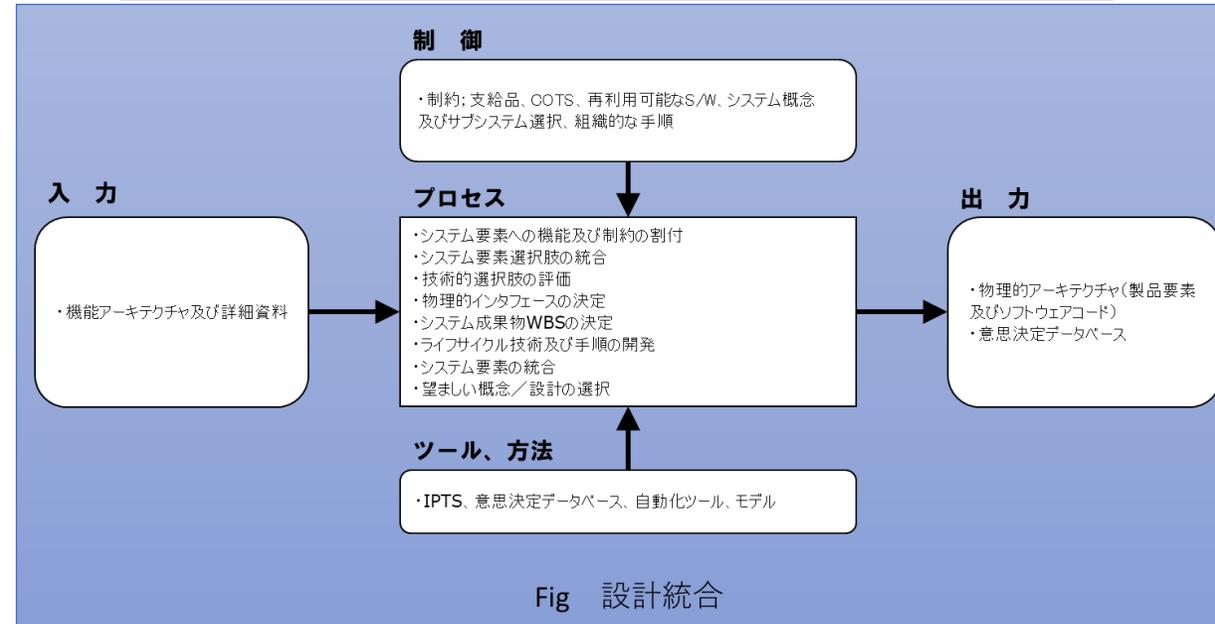
意思決定は他のアクティビティに影響を与える:

意思決定は、設計、製造、検証などの他のアクティビティに直接影響を与えることがあります。意思決定の結果によって、プロジェクトの方向性や計画が変わることがあります。したがって、意思決定は他のアクティビティを指導し、支配する役割を果たします。

意思決定は連続的なプロセス:

意思決定は一度だけではなく、連続的なプロセスです。状況や情報が変化するたびに、新たな意思決定が必要とされます。このような連続的なプロセスを通じて、プロジェクトや組織が進化し、目標に向かって進むことが可能になります。

「資料5_SEプロセス概要」にて概説



7. システムズエンジニアリングにおける意思決定(2/2)

設計レビュー

システムズエンジニアリングが開発業務において果たす役割は、単に技術的な設計や開発を行うだけでなく、利害関係者に対して適切な反応を引き起こし、プロジェクトの進行を円滑にするための行動を促すことにあります。

設計レビューはそのための重要な手段であり、利害関係者からのフィードバックを通じてプロジェクトを最適化し、成功に導くことができます。

この視点は、システムズエンジニアリングの実践において非常に妥当であり、現実的なプロジェクト運営においても重要な役割を果たします。

次に設計レビューの重要な役割を示します。

1. 利害関係者とのコミュニケーション
2. 設計レビューとフィードバック
3. 行動を促すための設計検証
4. プロジェクトの透明性と信頼性の向上
5. 否定的なフィードバックの重要性

品質が重要な部分であるとき、独立した確認が必要なのは、人々が信用できないからではなく、彼らが人間であるからだ。

(Watts Humphrey,
Managing the Software
Process)



最後に

今回は、ビジネスモデルとシステムズエンジニアリングの関係を中心に話題を共有しました。

今後、参加者のニーズや業界の動向に合わせて、継続的にSEの知識の共有と議論の場を提供する懇談会を開催してきたいと考えておりますが如何でしょうか。