

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



1

「ビジネスを駆動するプロジェクトマネージャーにとってのシステムズエンジニアリング」というテーマについて情報を共有させていただきます。

表紙の左に示しているのはポストイットです。

皆さんと懇談しながら話題を共有できたら楽しいだろうなという話題を貼り出し、それをもとに今回の資料を作成しています。

ことのはじまりは、システムズエンジニアリングをどのような場面で適用するのか？そして、その適用の意義は何か？というテーマでプレゼン資料を作成しました。200ページを超えるバックグラウンド資料作りを含め、約4ヶ月間をかけたプレゼンを妻に見せたところ「つまらない」と一蹴され、はてさて、どうしようと悩みました。

そこで、資料の表紙の左半分に示したように、心に残る言葉や皆さんが興味を持ちそうなものを集め、壁にポストイットを貼り、どのような流れにするかを考えながらプレゼン資料を作成しました。

正直、PMAJの方々にシステムズエンジニアリングを紹介するのは難しいと感じていました。同じような業界知識を持つ人たちと共有するのは比較的簡単だと思いますが、PMAJの方々はそれぞれ知識領域も違いますし、グローバルで大きなプロジェクトを対象としている方が多いです。そのような方々に向けて、どうやって伝えるかが一番の悩みでした。

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。



2

今回、皆さんと共有したい大きなフレームワークは、新しいビジネスモデル、特にAmazonウェブサービスなどのプラットフォームを中心としたビジネスモデルについての話題です。

これらの大きなプラットフォームが登場したことで、新しい参加者が現れ、これまでにない、プロセス・イノベーションによって業界の質が大きく変わってきています。

従来の垂直統合的なビジネスのやり方は、時代の変化に追従することが難しくなり、これまでにない新しいビジネスに踏み出すには多くの挑戦が必要となってきました。

こうしたなか、プロジェクトマネージャーにとっては、与えられたプロジェクトを成功に導くだけでなく、自分でビジネスを作り出し、それをリードするビジネスドライバーとしての役割が求められてきているのではないのでしょうか。

システムズエンジニアリングは、ビジネスケースや予算など多くの制約に沿って技術的解決策を生み出すテクニカル・ソリューションの一つです。

システムズエンジニアリングは、様々な人たちと知識を共有し、それをまとめ上げるためには、欠かせない技術領域となっています。

今回、私の経験を通じて得たこの知識体系について紹介していきます。

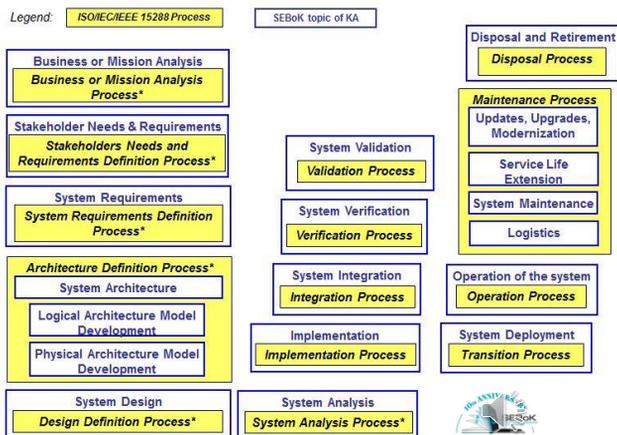
Agenda

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



第一部では「ビジネスモデル」をテーマに進めていきます。

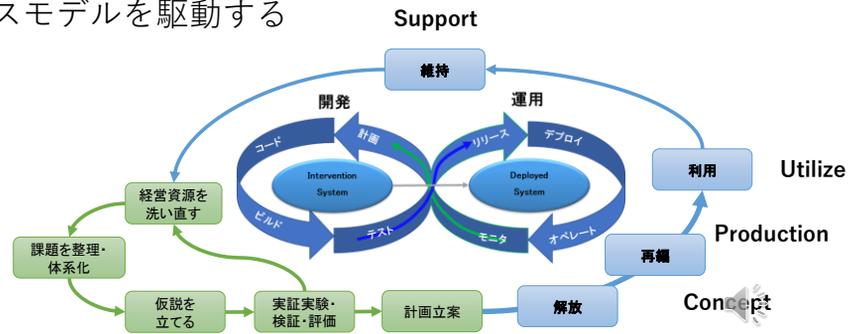
第二部として「システムズエンジニアリング」の具体的な活用方法についてご紹介する予定ですが、今日は時間の都合で第一部のみとなります。

右図は、ISO/IEC/IEEE 15288 :2015 Systems and software engineering—System life cycle processesについてSEBoK (Systems Engineering Body of Knowledge)に掲載されている図でプロセス群を示しているものです。

「日本産業規格JIS X 0170 : 2020 システムライフサイクルプロセス」としても規格化されています。

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



4

第一部では、ビジネスモデルについて説明します。

まず最初にビジネスモデルを生み出さなければ何も始まりません。

ビジネスモデルを生み出すとは、それによって、世の中に変革を起こすことです。

まず最初に世の中の課題を整理・体系化します。これにより、根本的な問題が明らかになります。その後、世の中の問題を解決するための仮説を立て、実証実験を行い検証・評価します。そして、経営資源を再評価するという試みを繰り返しながら、計画を立案していきます。

ビジネスモデルの成功には、試行錯誤としての実証実験（Proof of Concept）をいかに早く繰り返すかが重要です。最近では「DevOps（デブオプス：開発（Development）と運用（Operations）」という言葉もあります。開発段階から運用プロセスの開発も同時並行で行い、プロセスを早く回し、早く失敗し、多くの学びを得て、それを活かしながら市場に展開する方法です。

評価されたビジネスモデルを発展させるために、既存のリソースを再編し、維持・活用していきます。

新しい世界に向けて経営資源を再評価し、この流れを早く、継続的に、循環させることが、私たちのチャレンジとなります。

4.1 Business Mission Analysis Process



Reference

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

出典 INCOSE : SYSTEMS ENGINEERING HANDBOOK 4TH EDITION

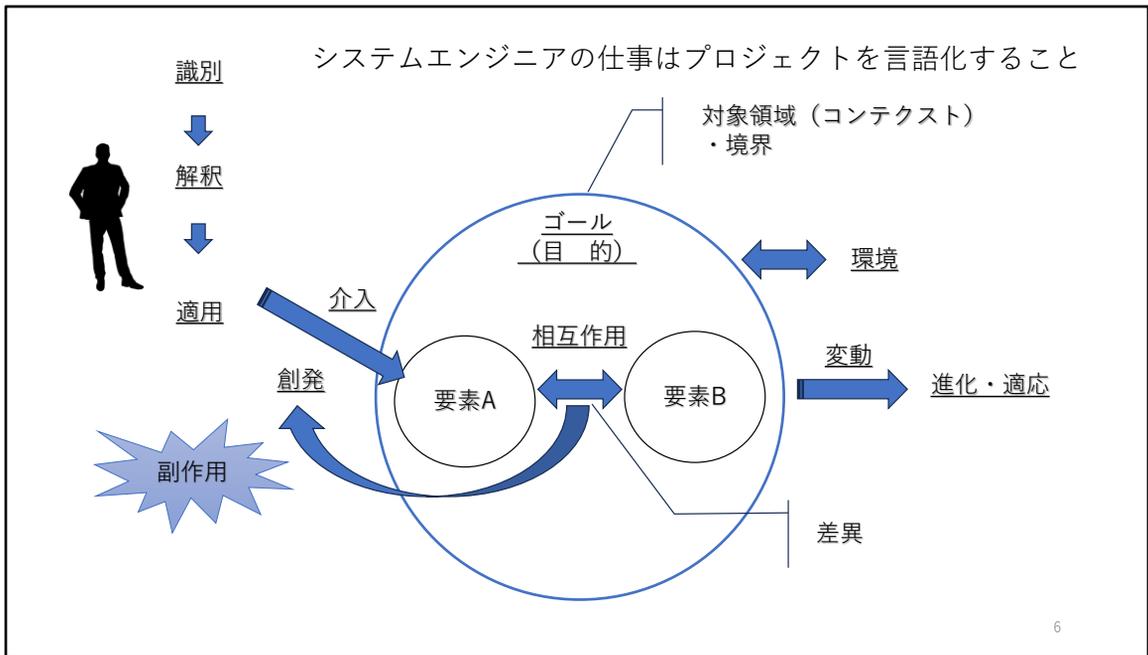
5

ここに示す文脈は、INCOSE : SYSTEMS ENGINEERING HANDBOOK 4TH EDITION のほんの一部です。多くの業界の知識が汎化され、強力な手順となっていることを理解していただければと思います。

最初に問題の領域を定義します。次いでその領域に存在するステークホルダーを識別し、問題解決領域における環境や制約事項を把握します。

これらは、一回限りのフレームではなく、ライフサイクル（取得、オペレーション、市場への展開、維持、そして廃棄まで）を考慮に入れ、繰り返す必要があります。市場投入までを考えるのではなく、トータルライフサイクルを見失わないようにすることが重要です。最終的には、ビジネスモデルが要求として妥当であるか、市場における妥当性評価基準を決めていきます。妥当性確認はそれが本当に有効に機能するかを確認するものです。実際に解決策を使ってみないと分からない部分も多いですが、しっかりとした妥当性評価基準を定義することが重要です。

このハンドブックは、防衛、宇宙、医薬品の開発、車の開発など、さまざまな分野において普遍的なプロセスを標準化したものです。広く適用できるこの抽象度の高いプロセス全体の流れを身につけることができれば、単なるビジネスミッションだけでなく、日常の行動や問題に対する取り組み方にも適用できます。



システムエンジニアの仕事は、プロジェクトを言語化（見える化）することです。

まず最初に、ビジネスのゴールを明確にし、そのゴールを達成するための対象領域とコンテキストをしっかりと把握する必要があります。この境界領域を最初に定義しなければ、関連するエンティティや要素を抽出することができません。そのため、境界をしっかりと定義することが重要です。

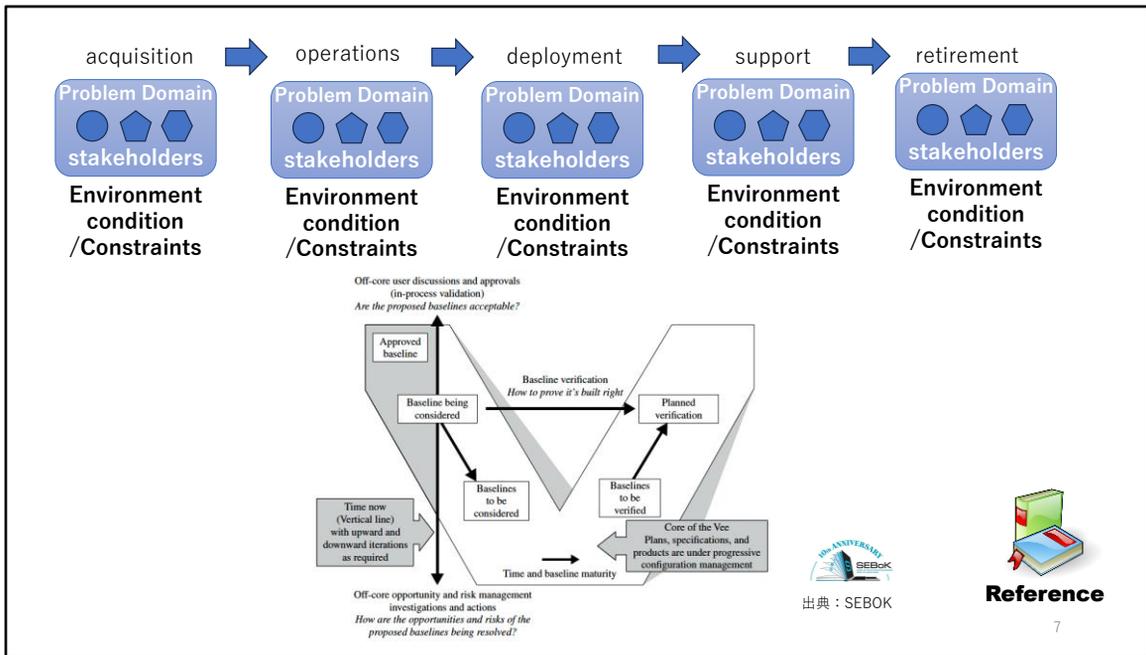
システムには、ゴールとしての目的があり、複数の要素で構成されています。これらの要素は有機的に結びついており、その結びつきがシステムの特徴となっています。

世の中をシステムのフレームを通して見て、対象とするシステムを理解し、定義するとともに、対象とするコンテキストに適用していくことで、世の中を正しくモデリングすることができます。

このシンプルな構造をプロジェクトとして如何に展開し、膨らませ、介入していくかがシステムエンジニアの仕事です。そして、その過程でシステム自体はが変動し、進化していくのです。

システムは自然界のエコシステムと同じように固定されたものではなく、創発特性や他の要素との相互作用を通じて変動し、進化、適応していきます。

この観点からも、システムをコントロールするのではなく、介入していくと考えることが適切です。



システムのライフサイクルに適応するためには、取得、オペレーション、市場への展開、維持、そして廃棄までという流れに従って、それぞれの段階における、前提、制約、そして要求を構築・確立していく必要があります。

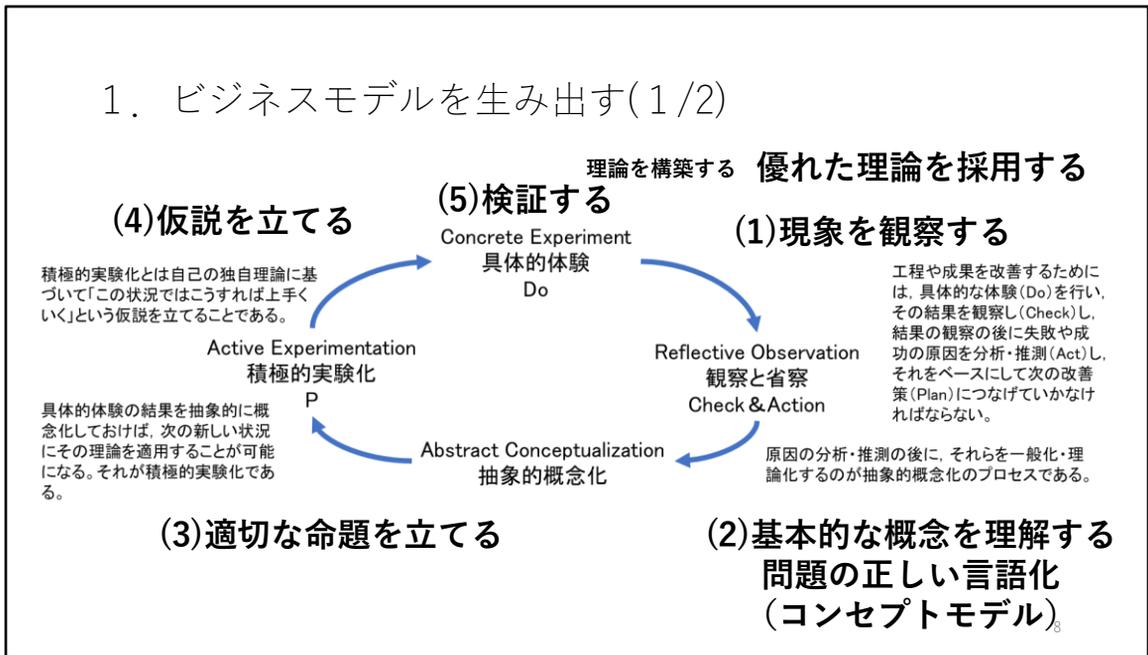
この過程、要求に対する検証と妥当性確認方法を検討し、システムを開発します。

ここまで説明してきた考え方に加えて、V（ヴィー）モデルの概念についても触れてみます。

Vモデルは必ずしもウォーターフォールモデルと同じではなく、要求、設計、製造のプロセス（左から中央）に対して検証・妥当性確認プロセス（左側）が対応していることを示しています。

さらに発展したWモデルもありますが、これは別の機会とします。

1. ビジネスモデルを生み出す(1/2)



次に、ビジネスモデルの生成についてです。

まず、現象を観察し、基本的な概念の理解を基に正しいコンセプトモデルを作り上げます。その後、適切な命題を立てることが重要です。うまく命題を立てた後、それに基づいて仮説を立て、検証していきます。こうして理論を構築していくのです。

優れた理論があれば、その理論に基づいて現象を観察します。理論が先かデータが先かという議論もありますが、理論がなければデータをどう集め、活用するかが難しくなります。無計画にデータを集めても意味がないため、現象の観察には優れた理論が必要なのです。

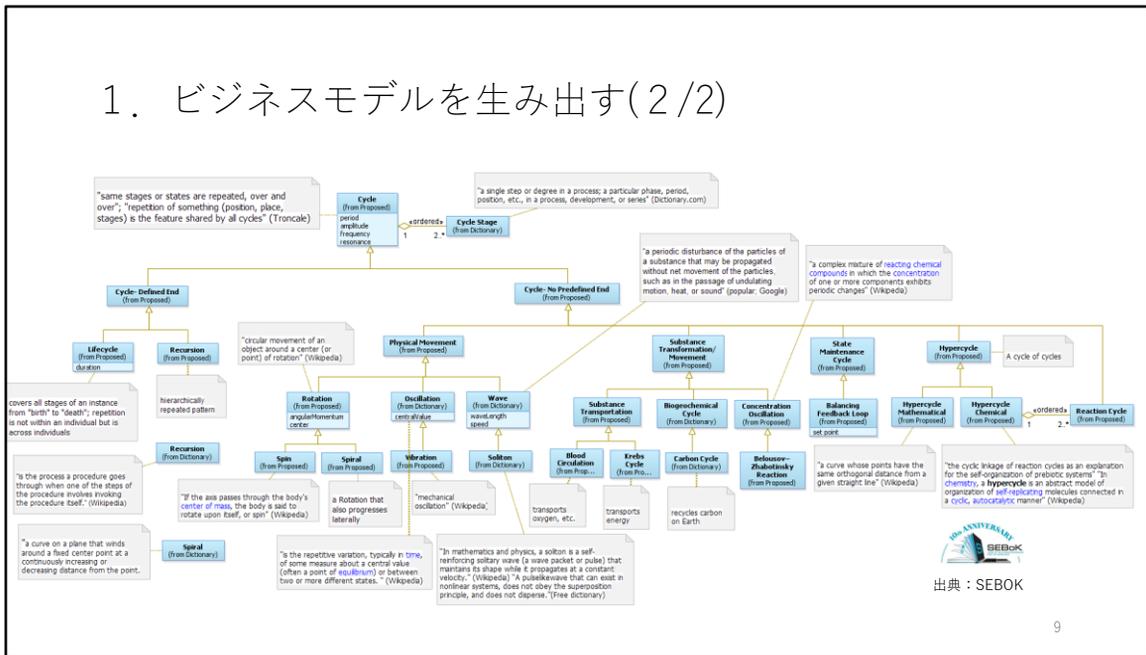
システムズエンジニアとして、どの理論が優れているかを理解することが重要です。

SYSTEMSTHINKERのサイトでは、システム原型 (Systems Archetypes) の概念が紹介されています。これらの原型を用いて現象を観察し、問題を識別し、適切な命題を立て、仮説を検証し、解決していくことは有効であると考えます。

SYSTEMS THINKER <https://thesystemsthinker.com/>



1. ビジネスモデルを生み出す(2/2)



これはSEBOKに記載されているモデルです。ここでは、サークルがどのようにカテゴリー分けされているかが示されています。

例えば、日本人が多くのキノコの種類を知っているのは、キノコに対する文化的な背景があるからです。おなじようにエンジニアとしてシステムに関する言葉を知っていることは、その人の強みになり、言葉を膨らませていくことは重要なことです。

サークルのような循環モデルは、二次元のモデルとして説明できます。

組織活動を例にとるとPDCA活動といったものです。

組織の進化や成長を立体的に捉えるとスパイラルとなり三次元的な視点となります。新しいプロジェクトが開始され、学びと改善を繰り返しながら次第に高度な成果を生み出す過程を表しているとも言えます。

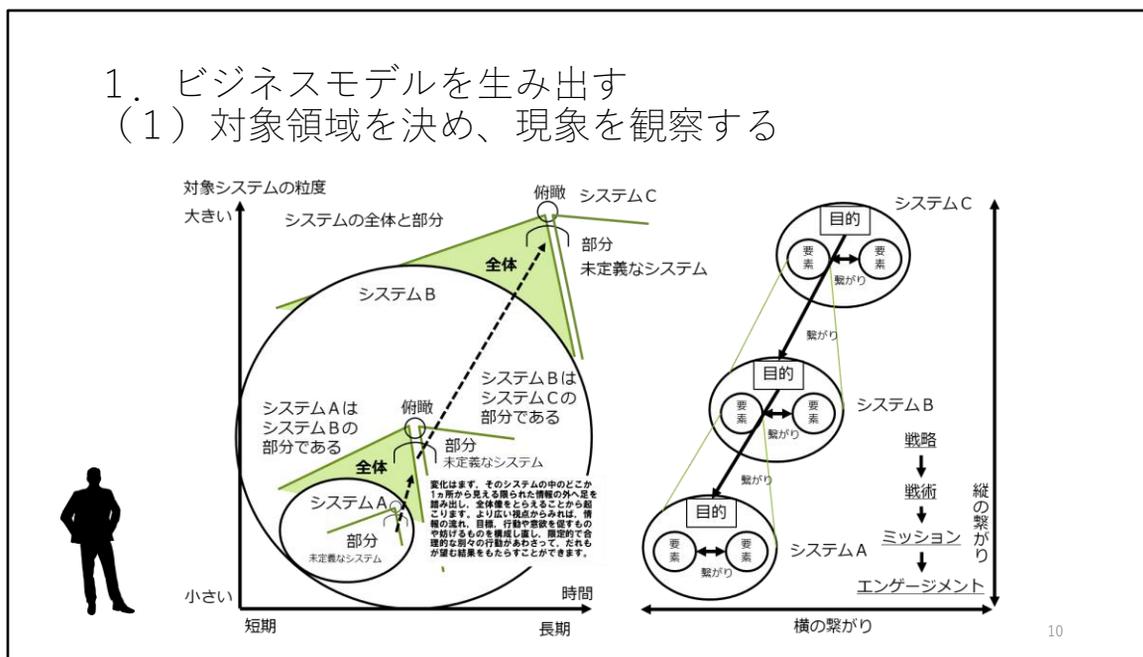
組織を中央集権的に一つの大きな組織にまとめるか、機能組織ごとに細かく分割するかを行ったり来たりしている様子は、線上を右往左往している様子として捉えると一次元的な視点となります。

このように、システムズエンジニアは、ものごとを多様なパースペクティブ（視点、視野、視座、ディメンションなど）で考えることが重要です。

サークルといった用語を知識体系として持ち、それを多様なパースペクティブを通して考えることが、システムシンカーとしての基本的な考え方です。

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



システムズエンジニアにとって重要なのは、木を見て森を見るように、全体を俯瞰しながら細部を見る能力です。これは、短期的な視点と長期的な視点の両方を持ち、全体像を大きな粒度で捉えた上で、徐々に粒度を細かくしていくというプロセスを意味します。

例えば、企業の戦略レベルでは、構成する要素（マーケットセグメント）の粒度は大きくし、10年を超える長期的な時間軸で捉えます。

担当者のレベルである、エンゲージメントレベルでは、構成する要素の粒度は小さくし、短期的な時間軸で捉えます。

動的平衡にみる「全体と部分」



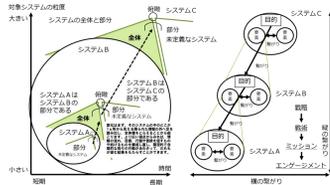
Reference

この世界のあらゆる要素は、互いに関連し、すべてが一対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」)



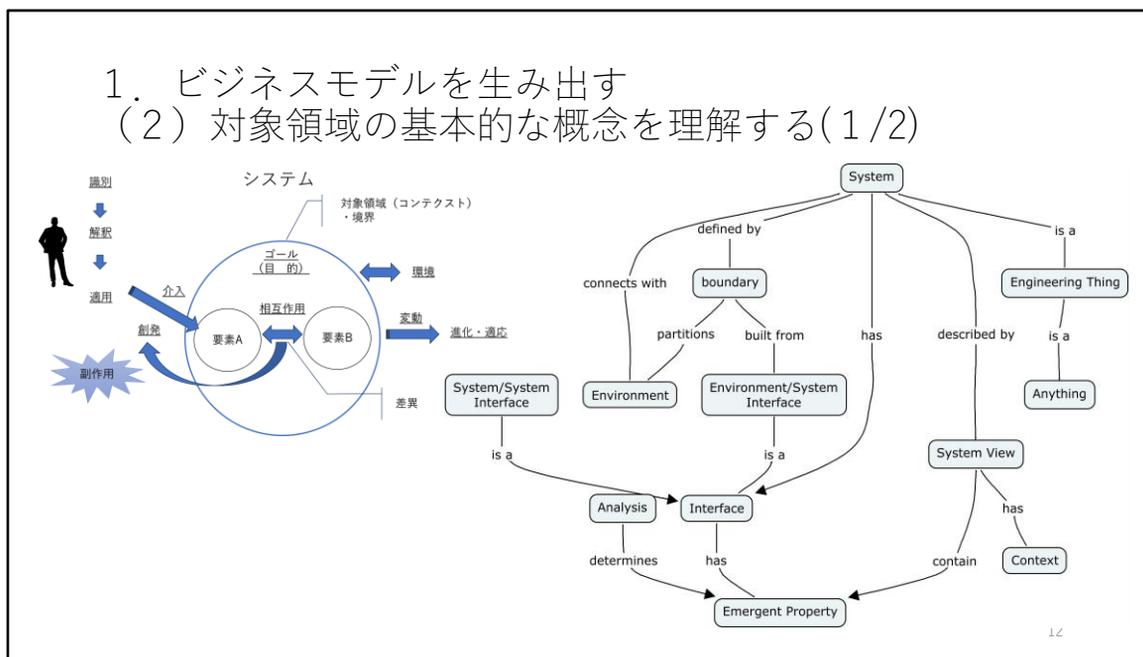
実際、ものごとというのは、全体と部分に分かれているわけではありません。全体は部分から成り立っていると言われますが、必ずしもそうではありません。これは、人間が管理しやすくするために分けているにすぎず、実際、世の中は全体と部分に分けることができないのです。

「動的平衡」という考え方もありますが、要素還元的にものごとを細かくしていけばコントロールできるという考え方は間違っています。全体と部分は切り離せないという認識が必要なのです。いかなる部分のシステムでも全体に統合された位置づけをしており、一部分だけの変更は不可能に近いのです。

また、「全体は部分の総和以上のものである。」という文脈も大切なものです。システムは、複数の要素と要素間の関係で構成され、全体としてある機能を果たす有機的なものです。ただし、やみくもに奥深く掘り下げ、要素還元的に、ただ単純にものごとを細かく打ち砕いたとしても、その膨大な因果関係を露呈するだけです。そして、その複雑さに圧倒されるだけで終わってしまうでしょう。

また、システムの要素に与えられている自由度が高ければ高いほど、そのつながりの中で、システムはますます複雑となっていきます。システムの全体は、部分の総和ではないのです。部分自体は極めて単純な要素であっても、互いに作用し合っています。その相互作用によって、創発的な特徴を生み出します。

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)



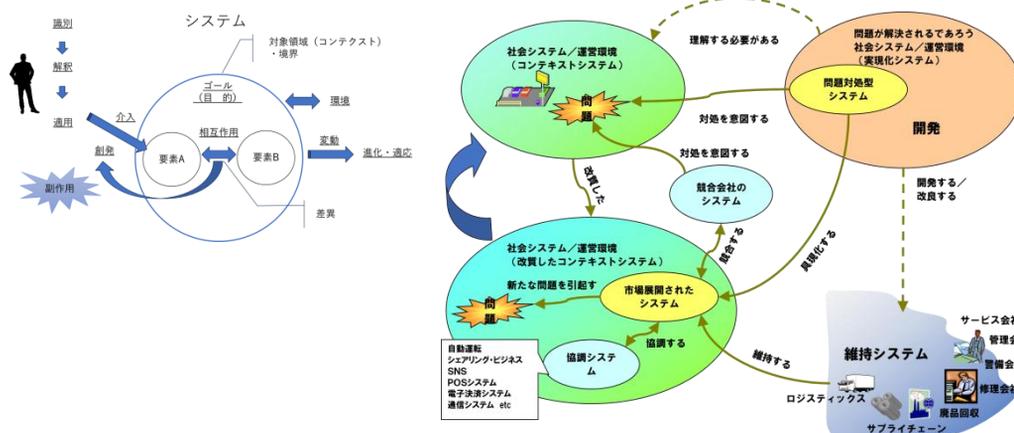
対象領域の基本的な概念を理解するためにオントロジーの活用は重要です。オントロジーは、存在や現実の構造を明確にするための哲学的な概念であり、情報科学や人工知能分野で特に重要です。具体的には、オントロジーは特定の領域の知識を体系的に整理し、その領域内の概念とその関係性を明確にするための枠組みとすることができます。

先ほど紹介した左図をオントロジーで表現すると、右図のようになります。システムにはインターフェイスがあり、インターフェイスはエマージェントプロパティ（創発特性）を持っていることを示しています。

このように、オントロジーは、利害関係者間でデータの共有や再利用を容易にし、異なるシステム間での相互運用性を向上させる役割を果たします。

自分の世界を表現するために、いくつものオントロジーを持っていると、ビジネスモデルを構築しやすくなります。フレームワークに囚われてしまうリスクはありますが、人間はモデルやパターンを通して世界を見ているので、まずしっかりとしたモデルやパターンを持つことが大切です。

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)



出典 : The Seven Samurai of Systems Engineering Dealing with the Complexity of 7 Interrelated Systems James N Martin

13

モデルを実際のコンテキストにどう落とし込むかについて、モデルの重要性をお話ししました。その一例として「7人の侍モデル」があります。これは、ジェームズ・マーチンのモデルで、7つの異なるシステムが相互作用しながら新しい世界を築く循環モデルとなっています。

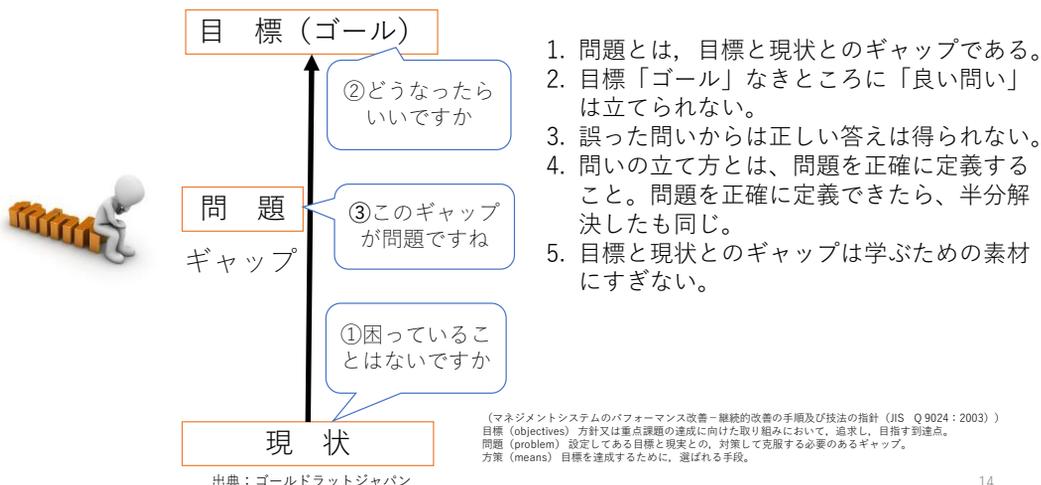
コンテキストにある情報や概念を理解するためには、それと対立するもの（競合システム）や反対の視点も同時に考慮する必要があります。

理解とは、固定されたものではなく、常に動的で、変化し続けるものです。理解は異なる視点や枠組み（7つの異なるシステム）が出会い、相互作用する中で生まれ、その過程で、新しい視点や意味が生じます。

理解とは、単に受動的なプロセスではなく、システムズエンジニアの現在の状況や問題意識に応じて、意味を再構築する能動的なプロセスであることを示しています。適用を通じて、システムズエンジニアは、過去の意味を現在の文脈（コンテキスト）に関連付け、新たな理解を形成します。これは、理解が動的で、実践的な性質を強調しています。

そうした意味で「7人の侍モデル」は優れたモデルといえるのではないのでしょうか。

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)



正しい答えは正しい問いからしか導き出せません。そのため、問題を適切に定義することが重要です。

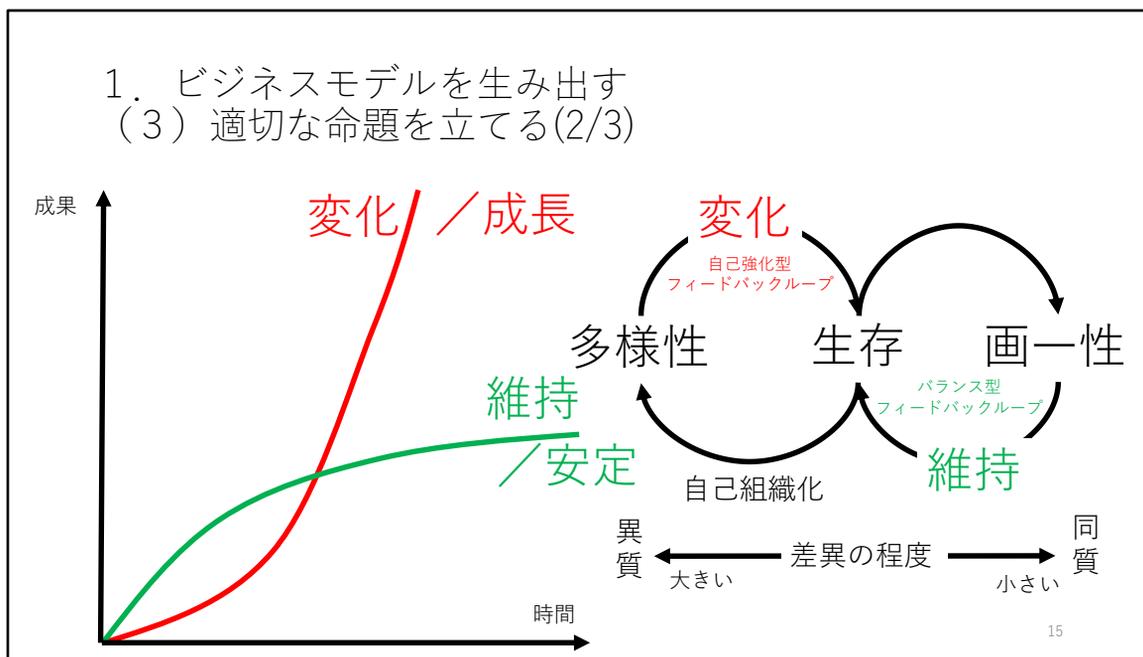
適切に問題を定義するには、目標の設定が大切です。現状と目標の間のギャップが問題であり、これは日本産業規格でもJISとして規定されています。

現状で困っていること、どうなれば良いかを明確にし、そのギャップを問題として捉えます。

目に見える単なる現象を問題として捉えるのではなく、「目標 (Goal) がなければ、目標と現状とのギャップ (Gap) を問題として認識することはできない」と強く意識することが大切です。

絶えず、目標と現状のギャップを問題として捉える習慣を身につけることが、適切な問題提起と解決の基礎となります。

1. ビジネスモデルを生み出す
 (3) 適切な命題を立てる(2/3)



生物は、生き残るために画一的な方法を取り入れています。これは安定性を維持するための有効な手段ですが、安定しすぎると進化が止まってしまいます。そこで、生物の特性である自己組織化が多様性が重要な特性となります。自己組織化が生み出すこの多様性が変化を生み出すのです。

ビジネスでも企業が生き残るためには、画一性と多様性の双方を取り入れる必要があります。

ビジネスモデルを考える際には、自然界のルールに基づき、画一性と多様性が共存する世界を作ることが大切です。多様性と画一性は対立するものではなく、互いに補完し合うものです。多様性ばかりを重視し、画一性を認めないとすれば、逆に単一的な考えに囚われ、同質性の罠に陥ることになります。

そのため、異質と同質は単なる差異の程度の違いに過ぎないことを理解することが重要です（ストローの両端を切っても左と右は存在します。これは、ものごとの本質が分割不可能であり、相互依存していることを示しており、左派が右派を排除して、右派は必ず存在するという喩えですが、これと同じように差異もその大小に過ぎないことは、ものごとの本質です）。

多様性に基づく変化と画一性に基づく安定は、これらが両立する場合、互いに補間し合い、システム全体の持続可能性、柔軟性、回復力を高めます。

維持／安定



Reference

「制度の繰り返し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。

出典：フランシス・フクヤマ著「政治の衰退」

16

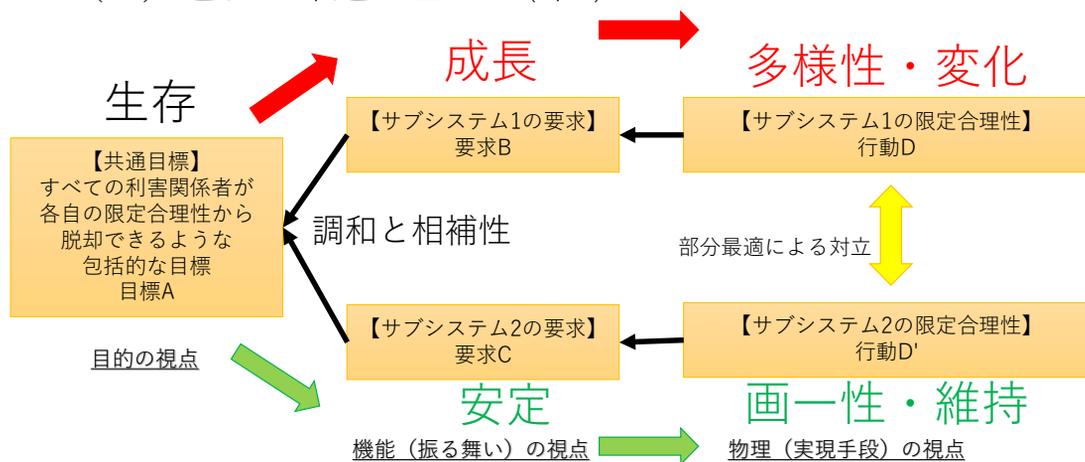
「政治の衰退」という出典から、維持と安定について紹介します。

私たちエンジニアは、理系やハードサイエンスといった分野だけではなく、リベラルアーツといった、ソフトサイエンスの考え方も持っていないとバランスを失ってしまいます。

リベラルアーツは、人文科学、社会科学、自然科学を含む広範な学問分野を指し、批判的思考、創造性、コミュニケーション能力を育むことを目的としています。現代では専門知識にとどまらず、幅広い視野と問題解決能力を持つ人材育成を目指しています。

システムが解決するのは、社会的な問題であり、ソリューションを提供するには、社会をしっかりと認識することが必要です。物作りだけに固執せず、ソフト的な考え方も持つべきです。

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)



システムズエンジニアリングとは、単なる技術やプロセスを知るだけでなく、リベラルアーツ的な視点も合わせて持つことが重要です。

命題を立てる際に、成長や多様性、安定性と画一性という二項対立軸に出くわすことが多々あります。よって、ビジネスモデルを構築する際には、調和と相補性を意識し、部分最適を避けることが重要です。多様性と画一性が対立するのではなく、調和の中で成り立つように展開することが求められます。

システムの中にいるそれぞれの主体者の限定合理性は、その主体者に影響を与える情報、行動や意識の推進要因、阻害要因、目標、ストレスなどの前提、制約、要求によって決まります。そして、解決できない問題の背後には、対立事象が存在しています。対立の負の循環は、矛盾を求めている状態（トートロジー）なのです。

調和という考え方は抽象的ですが、世の中や自然界を成立させるための基盤です。ビジネスもエコシステムのように調和を意識して構築しなければなりません。対立ではなく、バランスを取りながら相補的なものをうまく調整していくことが求められます。調和とは、心地よい、矛盾のない状態を形成する質であり、相補性として一見矛盾するような要素が調和することで、新たな展開や進歩が生まれます。

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。

出典 INCOSE : A Complexity Primer for Systems Engineers

18

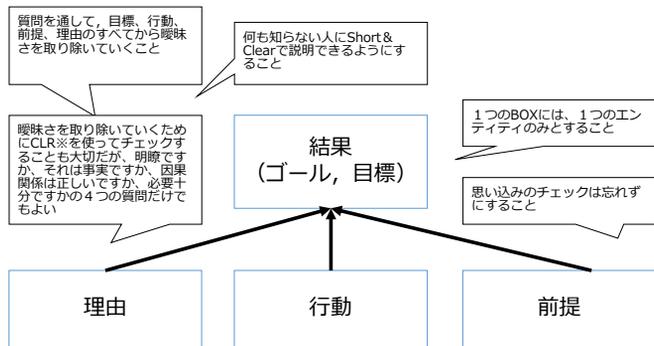
INCOSE発行“A Complexity Primer for Systems Engineers”には、複雑系に関するさまざまな事例が載っています。

これを参考にして、二元的な考え方ではなく、バランスの取り方についての知見を得ることができます。このような知見を活用して、世の中を考えていくことが重要です。

この文書の中で“ Achieve Balance ”というアプローチは、複雑なシステムにおいて最適化よりもバランスを取ることの重要性を強調しています。

最適化はしばしばシステムの柔軟性を失わせ、変化に対応できなくなるリスクがあります。システムエンジニアは、競合の中でバランスを取り、システム思考を活用することで、創造的かつ柔軟な解決策を見つけることができます。これにより、長期的に持続可能であり、変化に強いシステムの設計が可能になります。

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



うまく、原因と結果の関係を構築することができ根本的な原因をひとつに絞り込むことができれば、外見上は複雑に見えても実はシンプルなシステムということができます。なぜなら、基本的には、シンプルなルールがとてつもない複雑さを作り出しているからです。

右図は、水問題に関する事例です。あるとき、水問題に興味を持ち、関連する本を多数読んだことがありました。以前はダムがエコな発電システムだと思っていましたが、実は多くの環境破壊要素を含んでいることを知り、驚きました。世界ではエジプトやスーダンをはじめ、水問題を巡る紛争が多発しており、こうした事例研究を通じてでも、因果関係を用いて新しい視点を持つことができます。このように、行動、前提、理由を用いて、行動がどのような結果をもたらすのかを考えることで、私たちのもたらす様々な事象を新しい視点で捉えることができます。これを日々鍛えることで、システムズエンジニアとしての能力が向上します。

これにより、エンジニアとしての視点を養うことができます。仮説を立てることにチャレンジし、ビジネスだけでなくプライベートでもその考え方を活用していただくことをお勧めします。

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ことが良いでしょう。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。

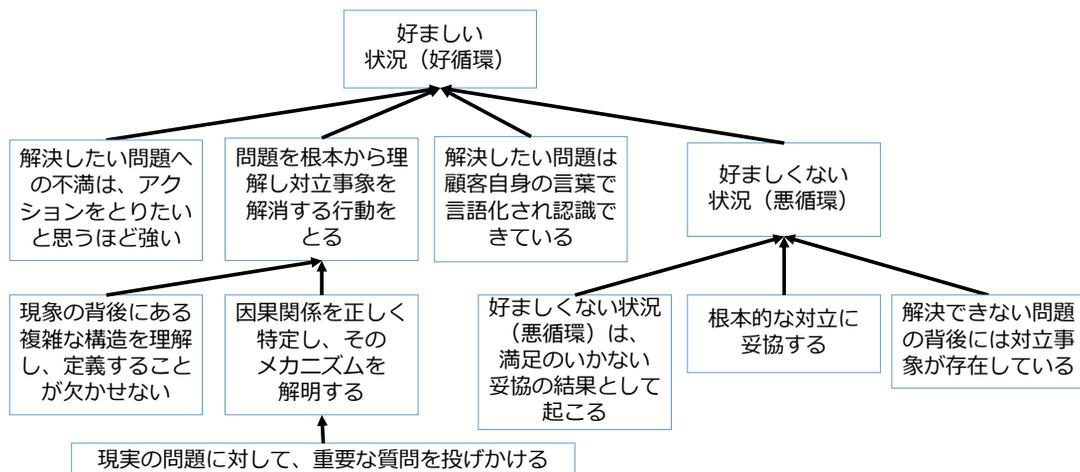
理由と正当化

「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)



21

世の中の好ましくない状況を好ましい状況に変えるという視点が重要です。そのためには、問題の背後にある対立要因を理解する必要があります。

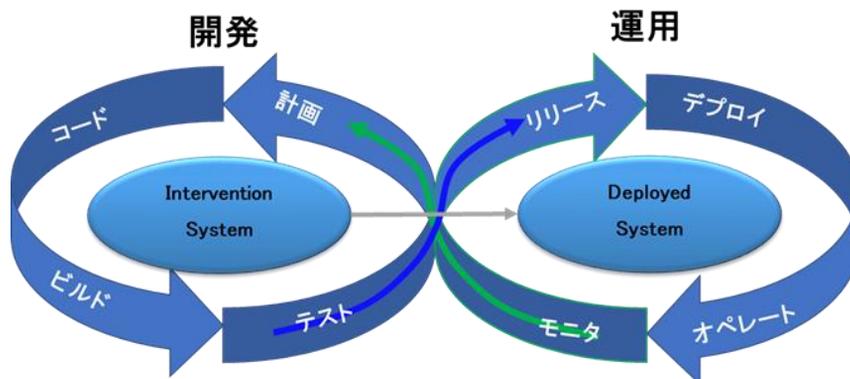
根本的な対立要因に妥協してしまうと、悪循環から抜け出せません。好ましくない状況には必ず根本的な原因があり、その妥協の結果が現状を招いていることを認識することが大切です。

現状と好ましい状況のギャップが解決すべき問題となります。この問題を根本的に理解し、対立要因を解消するためには、複雑な背景を正しく把握する必要があります。そのためには、因果関係を特定し、そのメカニズムを解明する手法を用いる必要があります。また、重要な質問を投げかけることも必要です。

不満や行動への意欲が強くなければ、問題解決に取り組むことはできません。そのため、熱意が非常に重要です。また、自分の言葉で問題を表現しなければ解決できません。自分の考えや目標を明確な言葉で表現することが大切です。システムズエンジニアは、プロジェクトを言語化して表現する能力が欠かせません。問題を言語化する能力がシステムエンジニアにとって重要なのです。

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



22

Jeff Patton著「ユーザーストーリーマッピング」にあるMVPの定義として「MVPは、仮定を証明または反証するために作られる。あるいは実行できる最小のものでもある。」とあります。

MVPは、ビジネスモデルを検証するためにも使用されます。新しいアイデアや仮定（例えば、「ユーザーはこのソリューションを必要としているか?」、「このソリューションは問題を解決するか?」）を最小限のコストと時間でテストすることを目的としています。これにより、仮定が正しいかどうかを迅速に確認でき、誤った仮定に基づいた無駄な開発を避けることができます。

「実行できる最小のもの」とは、製品やサービスの最小限のバージョンを指し、それが実際に使用可能であり、価値を提供するものであることを意味します。

このMVPは、マーケットが実際に試すことができるものであり、企業がフィードバックを得るためのベースラインとなります。これにより、プロジェクトチームは初期段階でマーケットの反応を確認し、必要な改善や方向転換を行うことができます。

ダンカン・ワッツ「複雑なシステムとモデル」



Reference

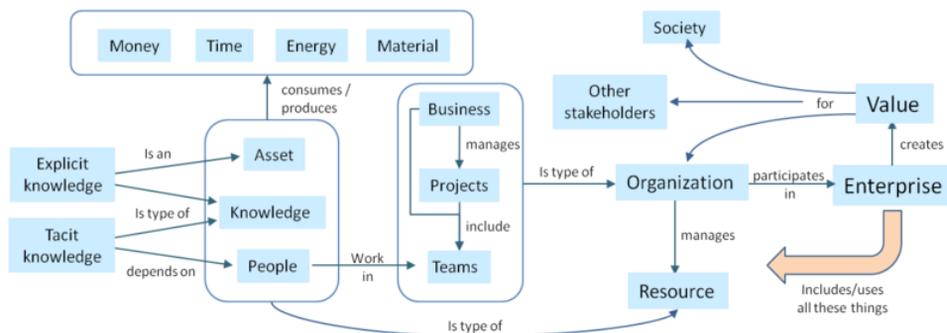
多数の独立した要素が非直線的な形作用し合うときに複雑性は生まれると一般に受け止められている。複雑なシステムでは、システムの一部におけるわずかな乱れが増幅され、他の部分に大きな影響を及ぼす。

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。

23

ダンカン・ワッツ (Duncan J. Watts) は、ネットワーク理論に関して非常に有益な知見を提供しています。複雑なシステムとモデルを考えると、ネットワークは切り離せません。モノとモノの繋がりが複雑さを増すため、ネットワークは複雑になりやすいことを理解していただければと思います。

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

ビジネスモデルを成功に導くためには経営資源をしっかりと理解する必要があります。知識には明確な知識と明確でない知識があり、これらの知識は資産として人々に依存しています。また、外部リソースとしてお金、時間、エネルギー、物資なども存在します。これらのリソースがチームとしてプロジェクトやビジネスを構成し、成り立たせます。この全体のフレームワークを通じて、経営資源とは何かを考えることが重要です。

経営資源として、お金や時間、人材などのシンプルな構造を理解するだけでなく、全体的な構造をしっかりと把握して考えることが大切です。最初のうちはSEBoKを参考にしながら、ビジネスの経営資源とは何かを捉えると良いでしょう。しかし、教科書がなくてもそれを考えることができるようになれば、優れたエンジニアと言えるでしょう。

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。

25

経営試験を見つけ出すことは非常に重要なスキルです。適切な資金が見つからなければ、ビジネスモデルを試すこともできません。

ここで示す「樽のたとえ」は、多くの要素の中で最も少ない要素が生態系を成立させるための最小要素となり、それが制約になることを示しています。

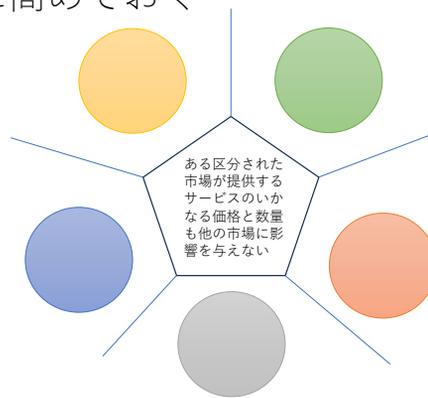
制約理論では、一番弱い鎖が全体のスループットの制約となり、その鎖の強さが全体の強さを決定すると言われています。このたとえにおいても、樽の最も短い部分が樽全体の貯水量を決定します。この制約を見つけ出し、強化することが重要です。

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

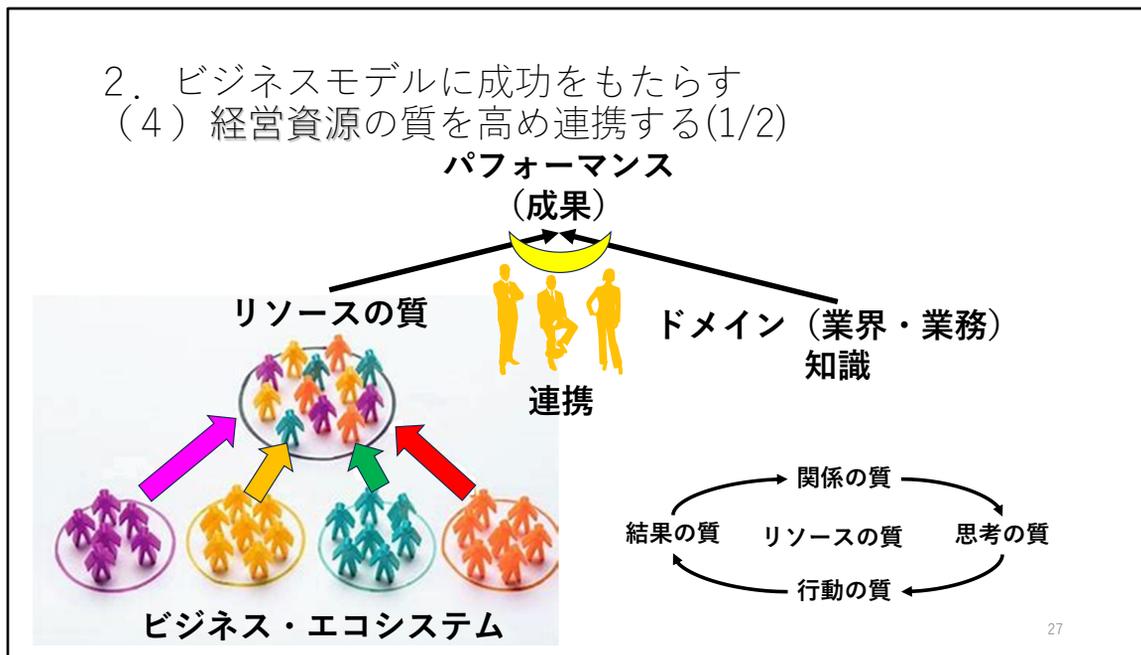
市場をセグメント化し、自由度を手に入れる

26

経営資源の重要な自由度を常に高めることが大切です。リソースのセグメンテーションに関しては、市場に合わせて組織を固定化してしまうと、市場が変化したときに対応できなくなります。リソースをセグメント化せず、市場をしっかりとセグメントし、そのセグメントに対してリソースを自由に組み合わせていくことが重要です。将来の市場変化に対応するために、リソースには柔軟性を持たせるべきです。このようにして、セグメントされたマーケットに対して適切なビジネスモデルを構築していきます。

実際、自分の組織だけで自由度の高いマーケットをコントロールすることはできません。そのため、さまざまな企業と協力し、マーケットのセグメンテーションを活用して利益を上げることが重要です。一つのカゴに卵をすべて入れるのではなく、複数のセグメントを持ち、それぞれの状況に応じて柔軟に対応することが理想です。

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)



27

システムズエンジニアとしては、ビジネスエコシステムのリソースとドメインを組み合わせ、パフォーマンスを向上させることが求められます。

ダニエル・キムのシステム原型の成功の循環モデルとして、関係の質、思考の質、行動の質、結果の質の順に改善していくことが重要です。良い関係の質から始めないと、良い結果の質は得られません。つまり、ビジネスにおけるリソースの質を高めるためには、関係の質を改善することが重要です。

システムズエンジニアは、様々なステークホルダー（顧客、ユーザ、技術者など）と密接に協力し、システムの要求と期待を調整します。これは、システムが実際のニーズに適合することを保証するために不可欠です。

また、性能、信頼性、セキュリティ、プライバシー、コストなど多様でしばしば競合する利害関係者のニーズと期待を満たすソリューションのバランスをとります。

激動する環境下においても、ビジネスエコシステムでのイノベーションの本質は、統合と適応です。統合能力は、新旧のコンピテンシー、内外の資産をまとめ、以前はバラバラであった要素を結びつけることによって、エコシステムにおける企業の役割を持続的に拡張し発展させることができます。

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
- ・ ネットワーキング
- ・ 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・エクセス
- ・ 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

28

経営の質を高めるためには、コンピテンシーモデルが役立ちます。

「SEビジョン2035」で紹介されているように、システムズエンジニアとしての基本的なスキルとして、システム思考、ライフサイクルの理解、クリティカルシンキング、システムのモデリングと分析が含まれます。さらに、専門家としての技術的なスキルや、プロジェクトマネジメントなどの統合的なスキルも重要です。自分の足りない部分を見つけ、それに対して必要なコンピテンシーを達成することが求められます。

エマージェントプロパティとは何か、それがあなたのビジネスモデルにどう影響するのか、といった質問に答えられるように日々訓練し、自分のスキルを高めていくことが重要です。

システム思考



Reference

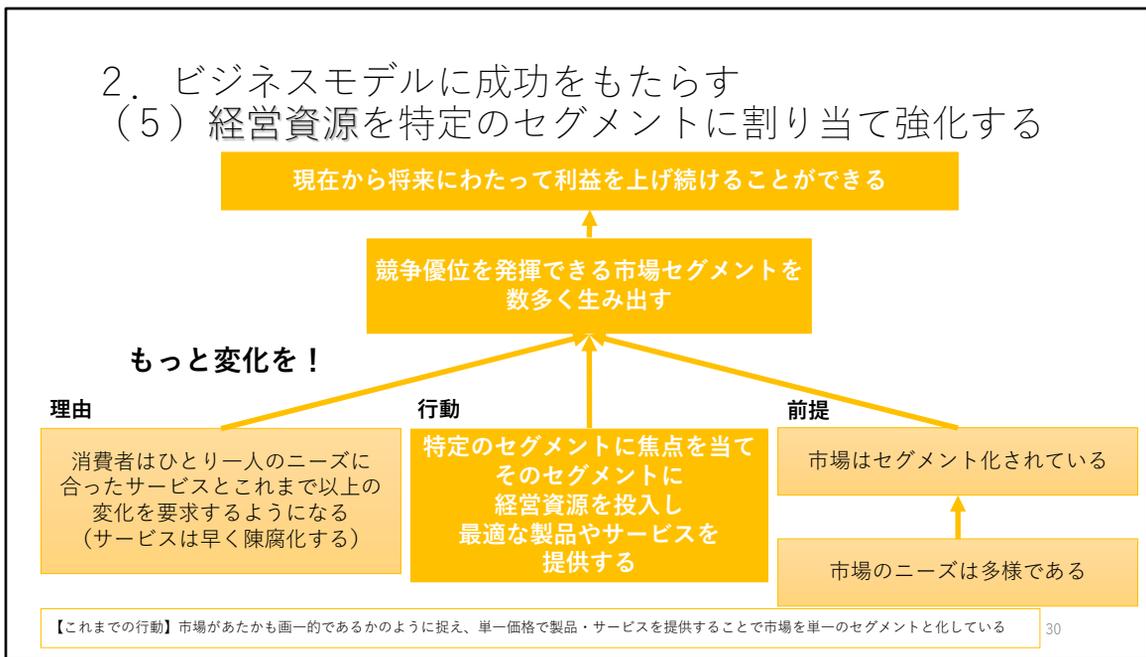
ピーター・センゲ (Peter Senge) は、1990年の著書『The Fifth Discipline (学習する組織)』でシステム思考について詳述しています。以下にセンゲのシステム思考に関する見解を説明します。

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。

29

ピーター・センゲの『学習する組織』は、システムダイナミクスに関する豊富な記載があり、非常に有益な本です。システム思考に関する様々な見解が示されており、時間がある時にぜひ読んでみてください。

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



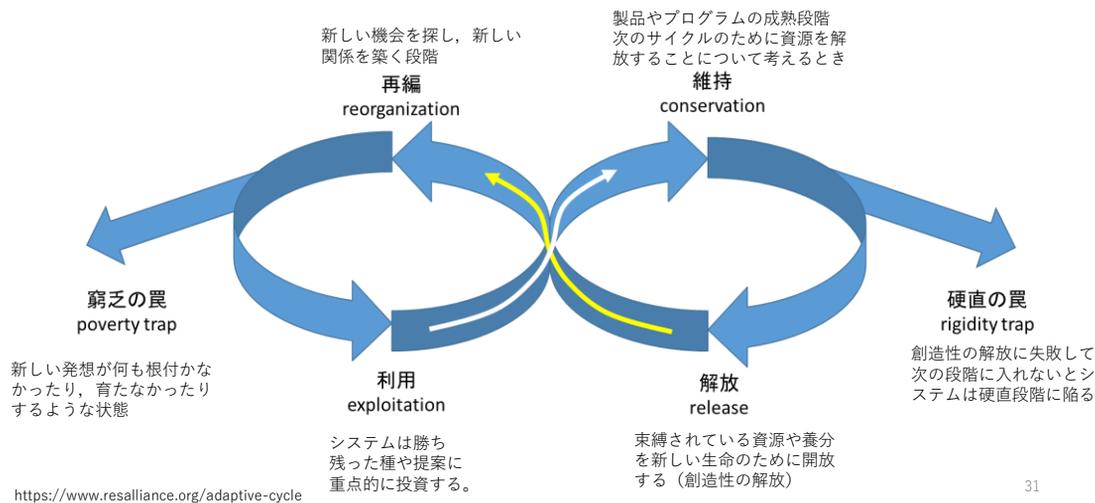
マーケットをセグメントしてそこに資源を配分するという考えを紹介しました。前提として、システムのニーズは多様であり、市場がセグメント化されていることが重要です。多様な市場ニーズを持つにもかかわらず、セグメントを一つにまとめてビジネスを構築してしまうのは矛盾があります。

市場のニーズが多様であるならば、セグメントも多様であることを前提に捉える必要があります。その上で、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ことが求められます。

消費者一人一人のニーズに合ったサービスを提供するためには、迅速に変化に対応することが必要です。市場が急速に変化し、新しいサービスが次々と登場する現状では、競争に打ち勝つために市場セグメントごとに戦略を立て、利益を追求することが重要です。

大規模な市場を抱え込むことは大きなリスクを伴います。そのため、ビジネスを推進するプロジェクトマネージャーやシステムズエンジニアが各セグメントで力を発揮し、それにより企業が成長していくのです。このような企業が今後も生き残り、発展していくと考えています。

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



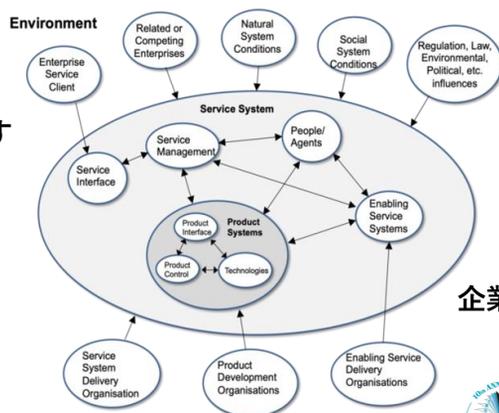
ビジネスモデルを駆動する際には、窮乏の罠や硬直の罠に陥らないように既存の領域から飛び出し、レジリエンス（resilience）と言われる、世の中の変化に対して強い耐性を持つことが求められます。

コンフォートゾーンとは、自分の支配力が及ぶと感じている範囲、あるいは、少なくとも十分な影響力を及ぼすことができる範囲のことです。また、人が原因と結果に関して、十分な知識を有していると領域、ある行動に対してどのような結果が予想されるのか十分な知識を有している領域です。

世界中の多種多様な状況での長期的な研究に基づく動的生態系モデルは、すべての生態系が四つの共通段階（急速な成長、保全、解放、再編成）を持つ循環サイクルを経ると主張しています。

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(1/4)

もっと変化を起こす



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

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。

ビジネスエコシステムを構築するには、自分だけではなく、生態系全体で考える必要があります。自分の企業が生態系の中でどのような役割を果たすかを定めることが大切です。そのため、現在の文脈でどのようなエコシステムを作ることによって強みを発揮できるかを考えることが重要です。周囲の企業と連携し、新たなエコシステムを構築することが1つの戦略です。

製品、テクノロジー、市場はイノベーションの関係性によって、互いに結び付けられています。これらの関係性を利用することは、企業の生産性を高め、組織を破壊から守り、革新のための能力を高め、進化し適応するために不可欠です。

どのような企業、組織、製品、テクノロジーも単独では行動できず、どの製品も単独で設計されることは不可能になってきています。

エコシステムのダイナミクスは、ビジネス戦略の持続可能性に関する、われわれの見方を変容させます。集合的な努力が企業のパフォーマンスにとって不可欠であるため、持続可能性は、個々の企業の能力ではなく、エコシステム全体の健全性に係る機能となります。

システムズエンジニアリングと社会的資本



Reference

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。

システム的な思考:

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。

33

システムズエンジニアリングと社会資本についてお話しします。

システムズエンジニアリング自体は社会的資本そのものではありませんが、社会的資本を構築する上で重要な役割を果たしています。エコシステムを作るには協力とコラボレーション、知識の共有、システム的な思考が必要であり、これによって社会的資本を高めていくことが重要です。

フランシス・フクヤマ 「社会的美德と繁栄の創造」から



Reference

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。

34

フランシス・フクヤマの「信頼：社会的徳と繁栄の創造」は、経済発展と社会の繁栄における信頼の重要性を論じています。これは、ビジネスについても当てはまります。信頼はビジネスの効果的な運用と持続可能な成長に不可欠な要素です。

システムズエンジニアリングは、複雑なシステムを設計、構築、運用するための統合的なアプローチを重視します。この過程では、異なる専門分野や部門が協力し、連携する必要があります。信頼は、こうした協力関係の基盤であり、情報の透明性、コミュニケーションの円滑さ、リスク管理の向上を促進します。フクヤマは、社会的な信頼が高いほど、企業や個人が共同でプロジェクトを遂行する際の取引コストが低下し、イノベーションや効率性が向上すると指摘しています。これは、システムズエンジニアリングにおいても同様で、信頼が高まることでプロジェクトの成功率が上がり、リソースの最適化が可能となります。さらに、信頼はシステムの持続可能性にも寄与します。高い信頼は、チームメンバーやステークホルダー間の長期的な関係を築く助けとなり、システムのライフサイクル全体を通じて安定性と革新をもたらします。したがって、フクヤマの信頼に関する洞察は、システムズエンジニアリングの実践においても重要な指針となります。

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

35

ビジネスエコシステムを構築する際には、違いを尊重することが大切です。同質なものが集まっても新しいものは生まれません。差異（違い）を見極め、その違いを活かすことで、力関係が生まれます。そして、同一性は限りなく珍しい差異に過ぎないということを意識しましょう。

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

36

ビジネスエコシステムを構築するためには、プラットフォームが必要です。例えば、Uberのようなプラットフォームは、共有と協力を促進するために透明性、公平性、セキュリティ、コミュニティの促進、参加者の協力インセンティブの提供などが重要です。適切なプラットフォームを選び、そのプラットフォームを中心にエコシステムを構築することが重要です。

信用は市場の発展と持続性にとっての基盤です。プラットフォームのガバナンス構造は、明確に定義されるべきです。共同組織をうまく機能させるためには、ガバナンス構造と目標を慎重に検討し、すべてのプラットフォーム参加者が完全に均衡して参加できることを保証しなければなりません。

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）」



「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。

37

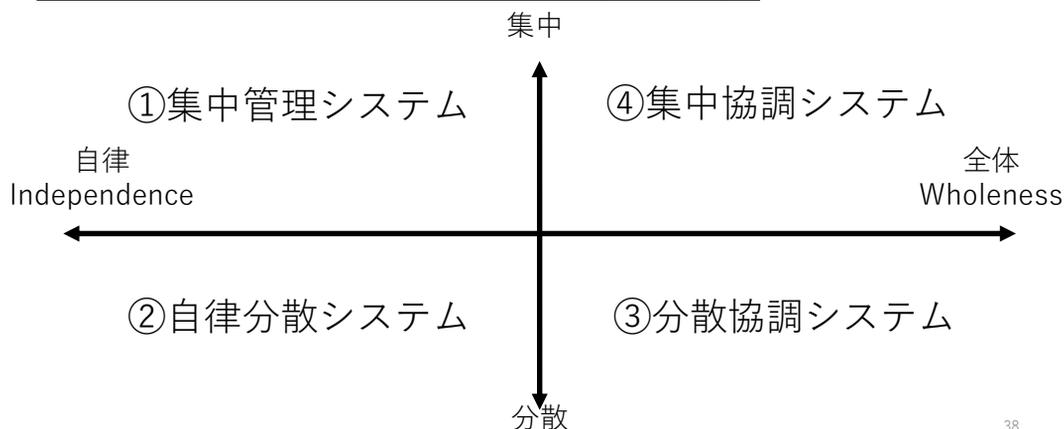
メルシエとスパーバーの「Enigma of Reason」に基づくと、理性はコミュニケーションと協力を促進するために進化しました。

ビジネスエコシステムにおけるプラットフォームも同様に、情報の共有、効果的なコミュニケーション、協力の促進を通じてエコシステム全体の成功に貢献します。

理性の進化的役割を理解することで、プラットフォームがどのようにしてビジネスエコシステム内で価値を創造し、維持するかをより深く理解することができます。

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)



38

ビジネスエコシステムを概念化するために、次の4つの軸を考えています。「集中と分散、自律と全体」です。以前は、中央管理システムが主流でしたが、現在は自律分散システムや分散協調システムなどがあります。

分散協調システムは、複数のシステムが協調して動作するシステムです。

また、プラットフォームメーカーが中心となる集中協調システムもあります。これらのシステムのどれが最適かはコンテキストによります。

ビジネスエコシステムを考える際には、これらの概念を把握し、具体的なシステムを選択と構築を行うことが重要です。この概念について議論する際には、集中管理システム、自律分散システム、分散協調システムなどについて特徴を理解し、自分たちのビジネスモデルに最も適したシステムを選び出すことが良い議論につながると思います。

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。

39

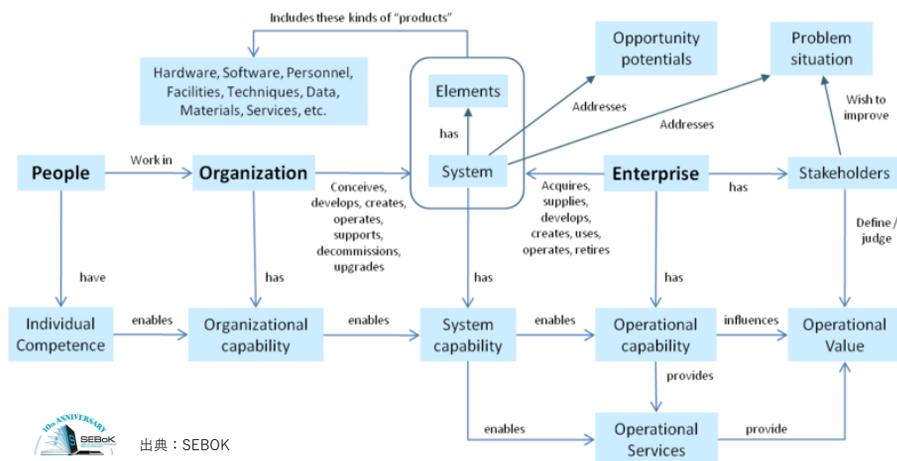
ノーマルアクシデント理論は、複雑なシステムと密接に結びついた事故について考える理論です。

ノーマル・アクシデント理論では、複雑で相互依存性の高いシステムでは高い確率で事故が避けられないとされているため、リスク分散の考え方は、システムの設計や運用において重要な手法となります。

つまり、システムの一部にエラーが発生した場合に全体に致命的な影響を与えないように、システム全体の構造や運用を分散型にすることが推奨されます。

例えば、一つのマーケットセグメントに全てのリソースを集中させるのではなく、複数のセグメントにリソースを分散させることで、個々の失敗が全体に与える影響を軽減することができます。これにより、ノーマル・アクシデントが発生した場合でも、その影響を局所化し、ビジネスシステム全体の健全性を保つことが可能となります。

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



40

モデルを駆動する資源の能力をマネジメントすることについてSEBoKには多くのモデルが存在しており、それを使って資源の能力をどのようにマネジメントするかを考えることができます。

人が組織を形成し、その組織がシステムにアクセスする構図、またエンタープライズ（企業）とそのステークホルダーの関係など、こういったビジネスモデルを作るための枠組みが提供されています。

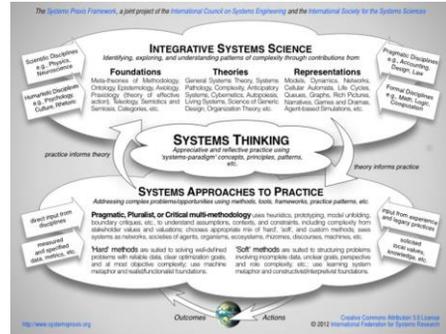
これらのモデルを活用することで、ビジネスのさまざまな要素を可視化し、モデル化することが可能になります。

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



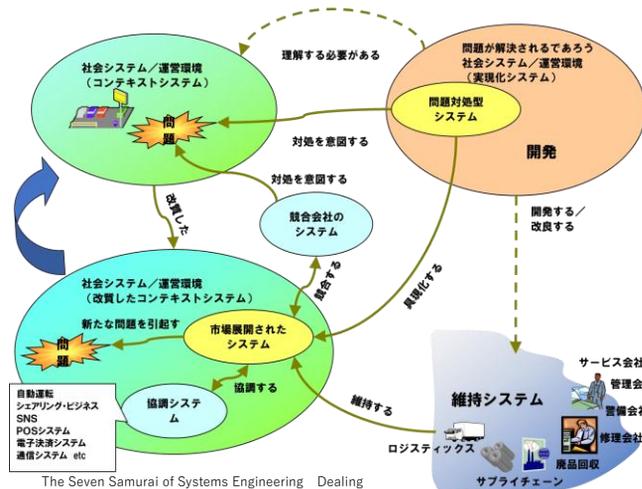
ISO 15288

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。

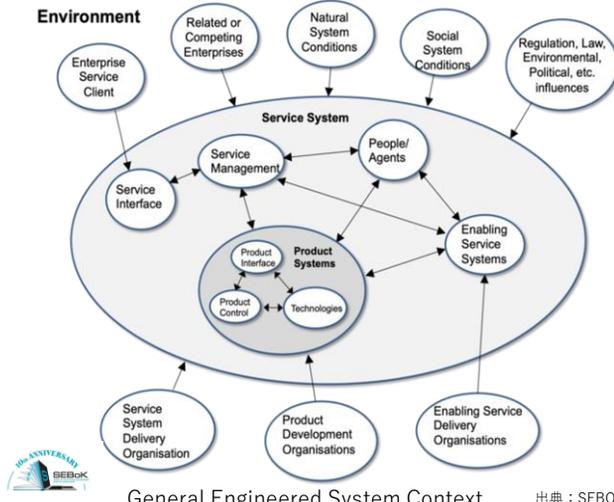
そのため、その意味や概念をコンテキストから切り離して説明することは、その本質を正しく理解する上で不十分です。

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

The Seven Samurai of Systems Engineering Dealing with the Complexity of 7 Interrelated Systems James N Martin

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



General Engineered System Context 出典：SEBOK

1.システムズエンジニアリングとその意味(4/4)

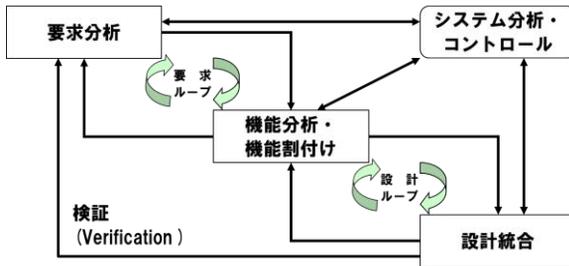


Fig SE Management Process (概要)

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し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)

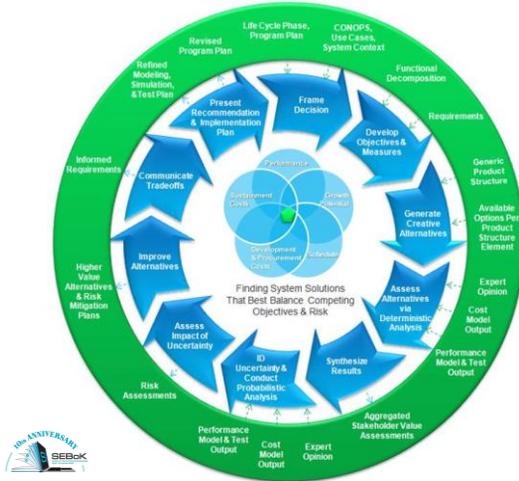


Figure . Decision Management Process (INCOSE DAWG 2013).出典 ; SEBOK

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、駆動されると考えら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。

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。



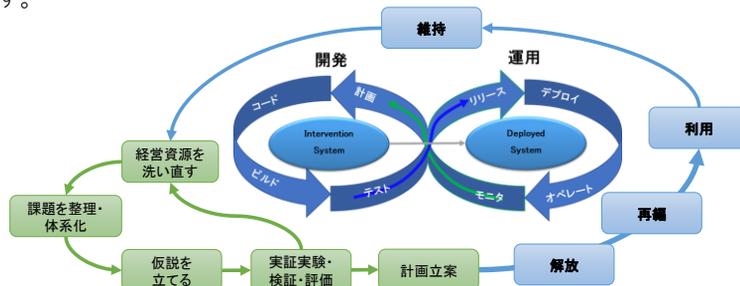
48

2.システムズエンジニアリングを駆動する力(3/3)

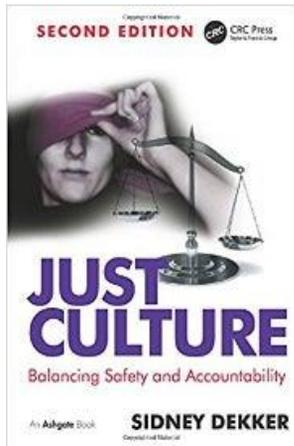
Proof of Concept (POC) の重要性:

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時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。

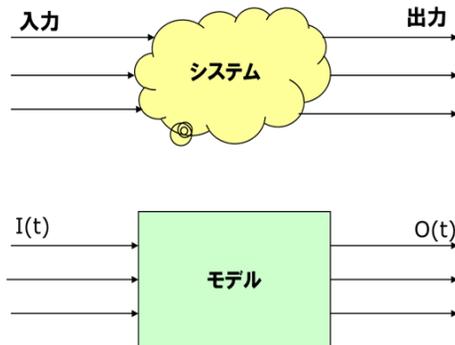


最大の障害は、人の心にある不安

- ◆ 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

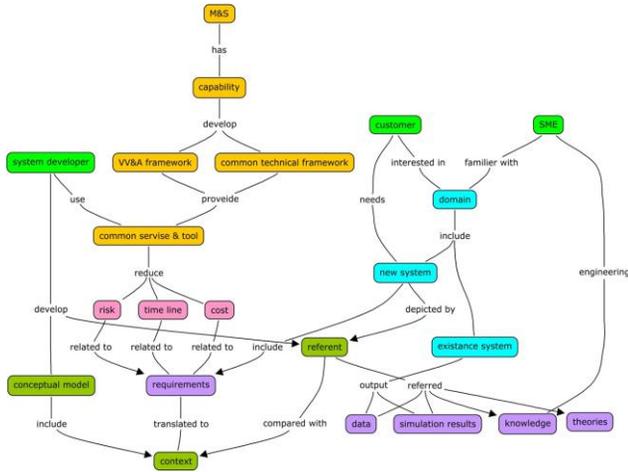


モデルや代理物がない場合、事象を再現するためには現地や現物を使用する必要があります。しかし、この場合、様々な可能性を試すことが難しくなります。なぜなら、現地や現物を使用する場合、実際の状況や条件の変更が容易ではなく、結果を予測することも難しいからです。

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



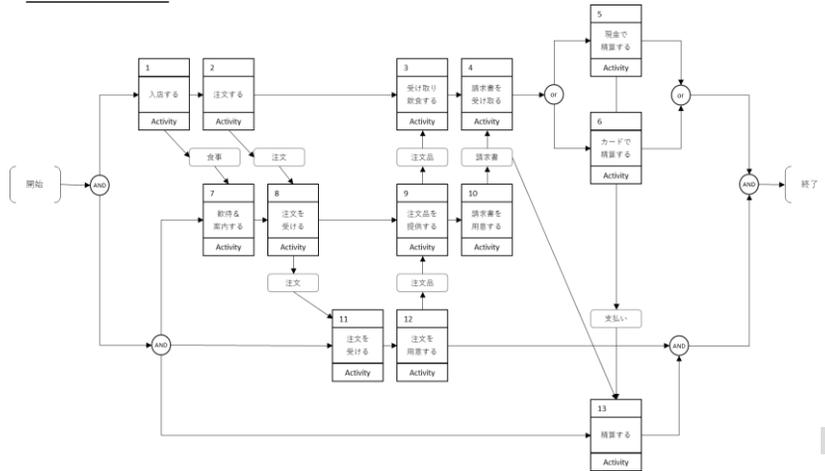
システムズエンジニアリングのコンテキスト理解は非常に複雑であり、組織やプロジェクトの複雑な相互作用、要素やプロセスを考慮に入れる必要があります。

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:

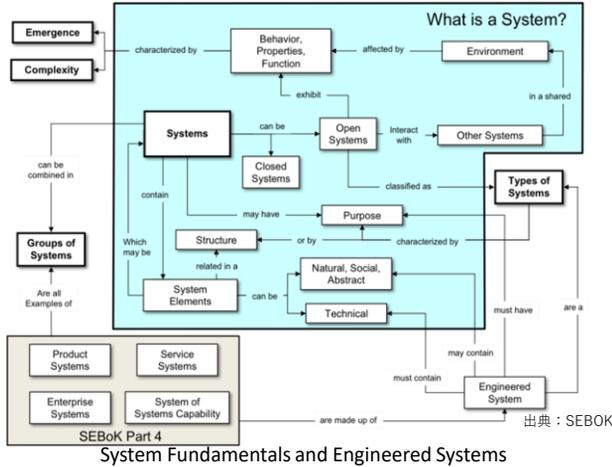
システムは、単なる要素の集合ではなく、組織化された構造と機能を持っています。システムを構成する要素は、特定のパターンや構造に従って組み合わせられ、特定の機能を果たすように配置されます。そのため、システムの意味を説明するには、要素だけでなく、それらの組織化された構造や機能も考慮する必要があります。

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



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、相対的にマイ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マイ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

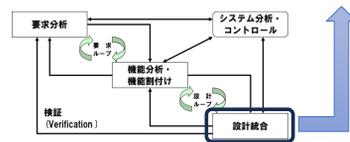
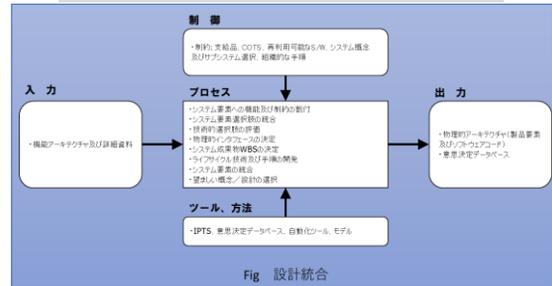


Fig SE Management Process (概要)

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)



59

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いきたいと考えておりますが如何でしょうか。