

# システムズエンジニアリング概要

## ～ VUCA時代のシステムデザインアプローチ～

慶應義塾大学大学院システムデザイン・マネジメント研究科 准教授 白坂 成功

ますますシステムを構築することが難しくなっている時代において、システム構築の基本的な考え方であるシステムズエンジニアリングの概要について説明する。とくにシステムとはどういったものであり、システムズエンジニアリングはどのようなものであるかを概説する。

### 1 背景

VUCAという言葉をご存知だろうか？ VUCAとは、Volatility (変動性)、Uncertainty (不確実性)、Complexity (複雑性)、Ambiguity (曖昧性)から構成された言葉である。現在の世の中はVUCAワールドと呼ばれ、先の予測ができない、計画通りにならない時代であると言われている。このような時代だからこそ、どのように環境を捉え、どのように考えてシステムを構築するかが重要となる。そのとき役に立つのがシステムズエンジニアリングである。例えば、どのように環境を捉え、そのためにどのような要求を持ち、それをどのように考えて設計したのかが分かっているから、環境が変化したときに、何が影響を受けるのかを把握できる。また、つながるシステムの代表であるIoTシステムのように、当初の想定とは違うものがつながり、複雑さがどんどん増す中で、その複雑さをコントロールする手段があるからこそ、対象を俯瞰的に捉えることができる。本稿では、VUCAワールドと言われる時代のシステムを考えるためには欠くことができないシステムズエンジニアリングの概要について説明する。

### 2 システムズエンジニアリング概要

#### 2.1 システムとは

世の中にあるほとんどのシステム、サービスのいずれも、複数の要素から構成されているという点で、システ

ムである。では、そのシステムというものをもう一度考えておくことは重要である。

システムは、「ハードウェア、ソフトウェア、ファームウェア、人、情報、技術、設備、サービス及びほかの支援要素を含む、定義された目的を成し遂げるための、相互作用する要素を組み合わせたもの」(INCOSE Systems Engineering Handbook)であると定義されている。つまり、いわゆるハードウェアやソフトウェアだけがシステムなのではなく、その中に人などを含むことがもともと想定されている。いくら詳細に個別の要素を見ても決して理解できないのがシステムである。システムとしてものごとを理解するためには、ものごとの個別要素にとらわれるのではなく、全体を一つのものとして捉える必要がある。

このようなシステムには幾つかの性質が存在する。ここでは、前述した定義に含まれている性質から説明し、その後、定義には書かれてないが、よく言われるシステムの持っている性質について説明する。これにより、システムがどういったものであるかについて、より明確に分かるのではないかとと思われる。

#### 目的性：システムは、それ固有の目的を持つ

前述した定義には「定義された目的を成し遂げるための」という記述がある。つまり、システムはそれ固有の目的を持つことが前提となっている。実際には、「システム」というものを一般的に扱うと必ずしも目的を持たないものも存在する。例えば、太陽系というシステムは自然システムの代表例であるが、目的を有していないため、目的性がないと言える。しかしながら、ここで扱うシステ

ム(人が作り出す人工システム)は、何らかの目的を持っており、それを実現するために存在するものに限定する。実際に、人間が作り出すシステムは何らかの目的を有しており、ここで扱う範囲においては上記性質を持っていると考えて問題ないはずである。

**集合性：システムは、目的を果たすために必要な複数の要素の集合である**

前述した定義では「要素を組み合わせたもの」と書かれている。つまり、要素が単体として存在しているわけではなく、複数の要素から構成されていることを示している。

**相互関係性：システムを構成する要素間にはシステムの目的を果たすために必要な何らかの関係が存在する**

前述した定義では、「相互に作用する要素」という記述がある。上述した通り、複数の要素が集まればシステムというわけではなく、複数の要素が相互に関係することにより、結果として新たな特質が生まれたとき(これを「創発」と呼ぶ)に、それをシステムと呼ぶ。

**階層性：システムの構成要素は、それ自体がシステムであって良い**

例えば、パソコンを考えてみる。もちろん、パソコンはシステムである。このパソコンというシステムは、モニター、本体、キーボード、マウスなどから構成されている。では、本体という要素を考えると、その中にはハードディスクやCPU、メモリなど多くの要素から構成されている。つまり、システムの要素がシステムとなっていると言える。実は、この考え方は大変重要なものである。なぜなら、このシステムの階層性という考え方を理解すると、その考え方を繰り返し使うことで、全体から詳細に至るまで統一的な考え方で設計を進めていけるようになる。この階層性という考え方を「Building Block」と呼び、システムの構成要素のことを「サブシステム」と呼ぶ。図1では、システムはサブシステムから構成されており、このサブシステムもまたシステムであるということを示している。これがシステムの階層性である。

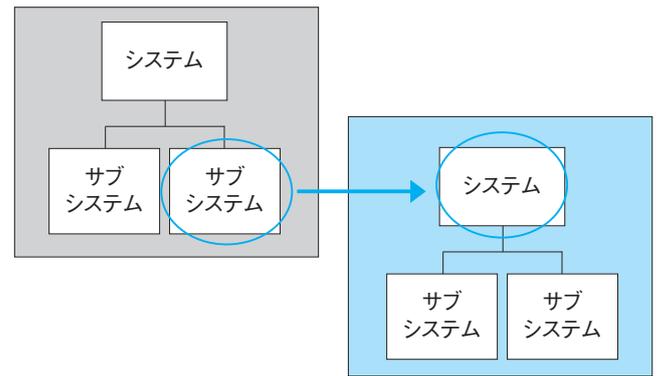


図1 システムの階層性を表すBuilding Block

**任意性：システムの範囲はあらかじめ普遍的に決められたものではなく、個別に定義され決められるものである**

システムというものを考えると、そこには明確な範囲というものがあるように感じるかもしれない。しかし実際には決してそうではない。どこまでがシステムの範囲であり、どこからがシステムの外部であるかは、その目的や立場から定義するものである。例えば、同じシステムを見るときでも、その利用者も含めてシステムとして捉える場合もあれば、利用者はあくまでもシステムの外部にいる人として捉える場合もある。前者の場合には、利用者にも積極的に役割を与え、訓練などを実施することでシステムの構成要素としての役割を担うことを期待することなどがある。

上述したようにシステムは幾つかの代表的な性質があり、これらの性質を知っておくことは、システムを理解し、扱うために重要なことである。

## 2.2 システムズエンジニアリングとは

システムズエンジニアリングとは、「システムの実現を成功させることができる複数の専門分野にまたがるアプローチ及び手段」と定義される。(INCOSE Systems Engineering Handbook) つまり、システムズエンジニアリングとは、複数の専門分野(例えば、電気工学、機械工学、ソフトウェア工学など)を統合し、束ねるためのアプローチである。単一の専門分野だけで課題を解決したり、価値を創造したりすることは難しく、複数の専門分野の統合が必要となることを考えると、システムズエンジニアリングは課題解決や価値創造のために必要な、基本的な考え方であると言える。

次にシステムズエンジニアリングを実施するために役に立つポイントを紹介する。

① 多視点から見る

システムズエンジニアリングの実施のためには、対象を多視点から見て考えることが重要である。対象によって最適な視点は異なるが、一般的に役立つ視点がある。それは、時間の視点、空間の視点、機能の視点、物理の視点である。時間の視点は、対象となるシステムの観点ではライフサイクルと呼ばれ、ユーザの観点ではカスタマージャーニーなどと呼ばれる。機能と物理は別の視点であることを認識して、分離して考えるだけで有益である。

② 俯瞰的に捉え、系統的に考える

多視点で見たものを、それぞれの視点ごとに俯瞰的に捉え、系統的に考えることが必要である。例えば、時間の視点で見て、俯瞰的に捉えることでライフサイクル全体を視野に入れ、系統的に考えることで、ライフサイクルをもれなく分割して系統的に考えることがこれに当たる。

③ 抽象度をコントロールする

ある視点から見て、俯瞰的に捉え、系統的に考えるときには、抽象度をコントロールすることが重要である。人は、あまり多くの数を一度に把握することが難しい。このため、いきなり全体を20や30に分割するのではなく、まずは5つに分割し、それぞれを更に5つに分割する。こうすれば、最終的には全体を25に分割することになるが、全体感を失わないように徐々に細部を考えていくことが可能となる。

以上のように、システムズエンジニアリングは、複数の専門分野の統合のためのアプローチであり、この実施時には3つのポイントを意識しておくが良い。

2.3 Vモデル

Vモデルは、システムズエンジニアリングの基本的な考え方を表す。図2に示したように、Vモデルの左側はシステムデザインと言われ、要求分析とアーキテクティングを実施することでシステムを構成するサブシステムへと分解することを表している。Vモデルの右側は、実

現されたシステムを構成する要素を統合(インテグレーション)してシステムを実現することを表している。そして、Vモデルの左側・右側の両方で、評価・解析を実施する。Vモデルの左側で実施する評価・解析としては、例えば、設計が正しいかどうかを確認するためのシミュレーションや、複数の設計候補案から一つを選定するために実施するトレードオフ分析などが該当する。Vモデルの右側で実施する評価・解析としては、試験が挙げられる。このとき、左側と右側のレベルが合わせてあり、Vモデルの左側での設計に対応した試験が、Vモデルの右側で実施される。また、右側で実施される試験のことを考えて、左側で設計を実施する。

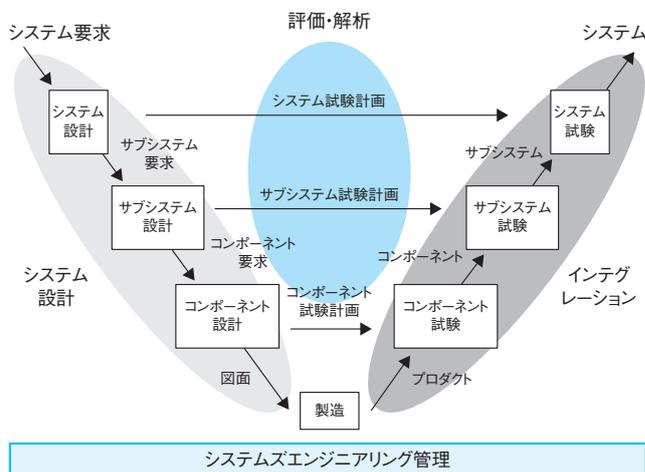


図2 Vモデルとシステムズエンジニアリングの構成

Vモデルは、システム開発全体プロセスを表すもの、と誤解されることがある。そのような場合もあるが、基本的にはシステム開発全体のライフサイクルとVモデルは独立に考えられるものである。つまり、システムのライフサイクルを通じて、Vモデルが何度も繰り返されることもあり、また、対象となるシステムの“部分”を表すときにも使われる。例えば、システムを構成する構成要素の一つが、これまでに使ったことのないような技術を活用したもので実現することを考えてみていただきたい。そのような場合、新しいことをいきなりやってみてうまくできない可能性を減らすために、試しに作ってみるということを考えるであろう。つまり、部分的に設計、製造、試験をして、うまくできることを確認する。その上で、

全体を別のVモデルに従って設計、製造、試験をする。このようにVモデルは、ライフサイクルを通じて複数となることもあり、またその範囲がシステム全体のときも、システムの一部であることもあり得る。

Vモデルは、色々なところで使われるようになった結果、「作業の順番を定義している」と捉えられているケースがあるが、実際には、順番を規定しているのではなく、あくまでも考え方を表現したものであるという点は注意が必要である。ここを間違えると、より進化したシステムズエンジニアリングを理解できなくなってしまう。あくまでも、Vモデルはシステムズエンジニアリングの概念を表したモデルであり、例えば、システムの要求は、サブシステムの要求の統合として実現されるということであり、Vの左側で行われる活動結果が、Vの右側で行われる活動で確認可能であるということである。

## 2.4 システムエンジニアリングプロセス

システムズエンジニアリングは、大きく4つのプロセスから構成される。上述したVモデルの説明と重複するところもあるが、ここでもう一度まとめて説明する。

### ① システム設計

まず、ステークホルダの要求を分析し、全体が整合し、抜け漏れがない形にすることで、要求を定義する。次に、その定義された要求を実現するために、アーキテクチャ設計を実施する。具体的には、複数の視点 (viewpoint) からシステムを見て、それぞれの視点から見えるもの (view) における要素と要素間の関係を定義し、異なる見えるもの (view) 間の関係性を定義することで、下位への要求を導出する。この活動はVモデルの左側として実施される。システム設計の実施時には、Vモデルの右側で実施されるインテグレーション及び試験のことを考慮して行う。

### ② インテグレーション

検証の終わったサブシステムを統合する活動である。

### ③ 評価・解析

エンジニアリング活動における解析及び検証 (verification)・妥当性確認 (validation) などの活動である。Vモデルの左

側では、システム設計をシミュレーションで評価したり、代替案をトレードオフ分析をすることなどがこれに当たる。また、Vモデルの右側では、できあがったサブシステムを試験したり、インテグレーションされたシステムを試験することがこれに当たる。

### ④ システムズエンジニアリング管理

QCDを満たすために、ライフサイクルを通じて各種活動の計画・実施・評価を行う活動がこれに当たる。この過程において、どのような範囲で何回Vモデルを実施するかを決めていくことも行う。例えば、初めて利用する技術であれば、評価のために、その範囲のVモデルをほかの部分よりも増やして実施することを計画し、実施するなどがこれに当たる。

## 3 最後に

本稿では、システムズエンジニアリングの基本的な考え方を紹介した。今後、VUCAが進んでくると、対象となるシステムを捉えることが容易でなくなるからこそ、システムを扱う基本的な考え方であるシステムズエンジニアリングの基本をしっかりと理解し、実施できるようになっていることが重要である。そのような基礎をおさえおいてはじめて、システムズエンジニアリングから進化したアプローチを活用できるようになる。そのためにも今一度、システムズエンジニアリングの基本を理解しておいていただければと考えている。