

# 成功事例に学ぶシステムズエンジニアリング ～IoT時代のシステム開発アプローチ～



## はじめに

情報通信技術は継続的に発展し、従来とは大きく様相の異なる段階に到達した。人工知能、クラウド、ビッグデータなどの技術の成熟と相まってIoT化が急速に進展している。第4次産業革命と言われるほどに産業構造、社会構造が変化し、競争の前提が大きく変わろうとしている。グローバル化によって潜在的な市場は拡大し、新技術を応用して新たな製品／サービスを実現できる可能性も高まっている。

IoTの進展により、製品／サービスが多様化しネットワークにつながるようになったことを背景に、これまでにない組み合わせでモノ・コトをつなげて、高い価値をもたらすことを目指す取り組みが始まっている。その対象となる分野・範囲の広がり等により、製品／サービスの企画・開発を行おうとする際、特定分野に特化した従来型のアプローチそのままでは目的の達成が難しくなっている。

多くの製品や技術を組み合わせ、多様なステークホルダと連携しなければならない一方で、機敏な行動も求められる。Volatility(変動性)、Uncertainty(不確実性)、Complexity(複雑性)、Ambiguity(曖昧性)といった概念(VUCA)で象徴される環境に適応しなければならない。挑戦的な取り組みにはリスクがつきものであり、変化を前提としたアプローチやマネジメントも必要になる。

このように企業を取り巻く事業環境が急速に変化する中、本書は、産業競争力を高める上で、システムズエンジニアリングの適用が重要かつ有効であるとの観察に基づいて作成された。今後、多くの企業でシステムズエンジニアリングのアプローチが利用されることを企図し、国内企業の事例を用いて、システムズエンジニアリングの有効性を示し、理解を促進することを目的としている。本書は、特定の産業に限定するものではなく、様々な産業におけるシステムズエンジニアリングの応用に向けて、基礎的な情報を提供する。

システムズエンジニアリングは、機械工学、電気工学、ソフトウェア工学など複数の専門分野を統合するアプローチであり、課題解決や価値実現に適用できる考え方として蓄積、整理されてきた。新たな製品／サービスの実現に向けて、対象や状況を多面的、俯瞰的に捉え、系統的に考えることが、従来以上に

重要になっている。様々な領域においてイノベーションを引き起こすために、様々な分野の技術者や管理者、そして経営者に系統的なものの見方、考え方が求められている。

本書は、システムズエンジニアリングについて詳細な手順書を目指したものではなく、システムズエンジニアリングの概要を理解し、自らの業務の中でシステムズエンジニアリングのアプローチを利用するための第一歩を踏み出す手助けを提供する。本書は、システムズエンジニアリングに関する体系化された標準プロセスなどの知見と、現実の事例をリンクさせて解説したことに特徴があるが、体系化された知見に偏って詳細を説明すると難しくなってしまうので、体系化された知見については入り口の情報を提供するとともに、事例に則した解説を中心に構成した。

本書は、システムやサービスの企画・開発に取り組もうとするマネジメント層、リーダー、担当者向けに作成した。例えば、

- ◆ 多様な利害関係者や専門家を含んだプロジェクトを実施しようとしている人
- ◆ 従来、単品の製品を開発し、一定の成功は収めてきたが、その製品を含めた付加価値の高いサービスを実現しようとする人
- ◆ 要件が決まればきっちり作る自信はあるが、自らの技術、製品を取り巻く環境を一段高い視点から分析しなければならなくなった人

といった方々を想定している。本書は、モノづくりの過程で、アプローチを変えざるを得なくなった方々をガイドする情報を提供しようとしている。

本書に掲載した事例では、絡み合った事情の中で最適解を目指して苦労を重ね、解決にたどり着く過程を紹介している。その過程に見出せるシステムズエンジニアリングのアプローチに説明を加えている。事例は特定の業界における特定のシステムを描いているが、他の領域においても適用可能な一般性の高いアプローチが多数登場するので、様々な問題解決に直面する実務者にお読み頂きたい。

2017年に発行した「経営者のためのシステムズエンジニアリング導入の薦め」および「開発者のためのシステムズエンジニアリング導入の薦め」は、システムズエンジニアリングの重要性や有用性を簡潔に訴えるものであり、導入意欲をかき立てるためのものであった。本書は、導入意欲を感じた人が、それに取り組もうとする際に、その手始めに何をすれば良いかを分かり易く解説す

る入門書である。国際規格 ISO/IEC/IEEE 15288:2015 を理論的拠り所とし、重要な用語や概念はこの国際規格に基づいて説明している。単に現場の興味深い事例として紹介するのみならず、その優れた実践内容をシステムズエンジニアリングの理論体系の中に位置づけながら解説した。

より多くの現場の技術者や管理者の皆様にも、システムズエンジニアリングの重要な考え方やアプローチをご理解頂き、業務の中でご活用頂ければ幸いです。

---

#### 本書の内容に関して

- ・ 本書の一部あるいは全部について、著者、発行人の許諾を得ずに無断で転載、複写複製、電子データ化することは禁じられています。
- ・ 本書に記載した情報に関する正誤や追加情報がある場合は、IPA/SEC のウェブサイトに掲載します。下記の URL をご参照ください。

独立行政法人情報処理推進機構（IPA）技術本部

ソフトウェア高信頼化センター（SEC）

<https://www.ipa.go.jp/sec>

---

#### 商標

- ・ 本書に記載する会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。
- ・ 本書の文中においては、これらの表記において商標登録表示、その他の商標表示を省略しています。

## 目次

はじめに	1
<b>第1章 システムズエンジニアリング概説</b>	<b>7</b>
1.1 環境の変化	8
1.2 システム開発の課題	11
1.3 システムズエンジニアリングの重要性和4つのポイント	14
1.4 システムズエンジニアリングの基本的な概念	19
1.5 本書の構成	24
<b>第2章 事例でみるシステムズエンジニアリングによる問題解決</b>	<b>26</b>
2.1 事例の読み方	27
2.1.1 事例の記載項目	27
2.1.2 各事例で実践されたプロセス	28
2.1.3 ポイント×プロセス対応表	28
2.1.4 事例一覧	29
2.2 地域活性化イベント向け情報共有基盤の開発	30
2.2.1 本事例の内容	30
2.2.2 本事例の背景と課題	31
2.2.3 アプローチ	33
2.2.4 対策の詳細	37
2.2.5 対策の効果のまとめ	44
2.2.6 教訓	45
2.2.7 ポイント×プロセス対応表	46
2.3 電子お薬手帳システムに適用したセキュリティ設計	48
2.3.1 本事例の内容	48
2.3.2 本事例の背景と課題	49
2.3.3 アプローチ	50
2.3.4 対策の詳細	52
2.3.5 対策の効果のまとめ	59
2.3.6 教訓	59
2.3.7 ポイント×プロセス対応表	60
2.4 多様な要求を満足させる自動車エンジンの開発 ～システムズエンジニアリング	

視点からのSKYACTIV-G～	62
2.4.1 本事例の内容	62
2.4.2 本事例の背景と課題	63
2.4.3 アプローチ	64
2.4.4 対策の詳細	67
2.4.5 対策の効果のまとめ	73
2.4.6 教訓	74
2.4.7 ポイント×プロセス対応表	75
2.5 首都圏の高密度輸送を支えるデジタルATC開発	77
2.5.1 本事例の内容	77
2.5.2 本事例の背景と課題	79
2.5.3 アプローチ	80
2.5.4 対策の詳細	83
2.5.5 対策の効果のまとめ	87
2.5.6 教訓	88
2.5.7 ポイント×プロセス対応表	89
2.6 Webスキャンシステムの企画開発	91
2.6.1 本事例の内容	91
2.6.2 本事例の背景と課題	93
2.6.3 アプローチ	93
2.6.4 対策の詳細	97
2.6.5 対策の効果のまとめ	101
2.6.6 教訓	101
2.6.7 ポイント×プロセス対応表	102
第3章 システムズエンジニアリングを始めるにあたってのヒント	104
3.1 システムズエンジニアリングを始めるにあたって考えるべきこと	105
3.1.1 どこから始めるのか	105
3.1.2 どのように進めるのか	106
3.2 システムズエンジニアリング実践ケース	109
3.2.1 実践ケースの概要	109
3.2.2 社内標準にシステムズエンジニアリングを取り入れた場合	110

---

<b>3.2.3 現場適用に先立つパイロット・プロジェクトを実施した場合</b> .....	113
<b>3.3 まとめ</b> .....	118
コラム:八百屋をシステムと見なすと .....	119
<b>第4章 システムズエンジニアリングのライフサイクルプロセスの紹介</b> .....	124
<b>4.1 システムライフサイクルプロセスの全容</b> .....	125
<b>4.2 ISO/IEC/IEEE 15288:2015 システムライフサイクルプロセスの概要</b> ...	127
<b>4.3 システムライフサイクルプロセスの中で効果的とされるプロセス</b> .....	129
<b>4.4 ISO/IEC/IEEE 15288:2015における上流プロセスの紹介</b> .....	130
<b>4.5 上流プロセスで使われている用語</b> .....	142
参照文献 .....	149
おわりに .....	150

# 第1章 システムズエンジニアリング 概説

本章では、IoT時代の製品／サービスの開発に関わる課題を紹介し、これらの課題を解決するためには新しいアプローチが必要であることを説明する。そして、そのような場面において、システムズエンジニアリングが提供してくれる主要なアプローチが役立つことを紹介し、合わせて重要な用語や概念を解説する。



## 1.1 環境の変化

情報技術が発展、普及し、第4次産業革命と言われるほどに産業構造、社会構造が変化しようとしている [1]。この変化によって、新たな産業の創出や既存産業の効率化と高度化、様々な分野における利便性の向上が進展し、社会課題の解決 [2] や質の高い生活の享受が可能になる。一方で、急激な変化によって既存産業や雇用に大きな影響が生じる可能性もあり、デジタル化の進展による既存ビジネスへの影響や日本の立ち遅れも懸念されている [3]。さらにサイバーセキュリティや機能安全の領域で従来にない問題が発生する可能性もある。ビジネスを取り巻く環境は、以下のような観点で大きく変化している。

### (1) 迫られるビジネスモデルの変革

自動運転を先導する Google、世界最大の小売業となった Amazon.com など、米国の大手 IT 企業による積極的な研究開発投資と新サービスの提供が世界を変えようとしている。Uber や Airbnb のように技術を利用した新企業の機敏な活動によって、既成の業界秩序が崩れ、ビジネスのルールが変化し続けている。既存企業は変化に対応するため、ビジネスモデルの変革を求められている。

従来、日本は多くの優れた製品を開発し、世界中に提供してきた。その技術力は依然として多くの分野で優位性があるが、いくら高度な技術をもって製品を開発しても、社会が求める適切な価値を実現できなければ利益を上げることは困難になってきた。

### (2) モノからコトへ～重視されるシステムの視点～

IoT (Internet of Things) 時代の製品/サービスは、単品ではなく、様々な製品/サービスとつながることで価値を発揮する。個別の製品/サービスの持つ機能的価値 (モノ) を大量販売する時代から、製品/サービスによってもたらされる一連の体験 (コト) を提供する時代になってきた [4]。

デジタル化の進展によって、企業による顧客価値実現の手段が変わりつつある。「モノの所有」から「機能の利用」へ、そして「体験の提供」によって顧客価値の実現が図られる傾向が高まってきた [5]。モノの販売によってビジネスが完結するのではなく、モノの利用を通じて顧客ニーズの充実を図るサービスやソリューションの提供力が重視されるようになってきた。

このような状況では、ソリューションを構成する製品／サービスが相互に依存し、統合されて価値を高め合うシステムとしての視点が重要になってくる。さらにモノを売った後に、そのモノの利用によって生じる連鎖的、継続的な消費が大きくなる場合も多く、企画の段階から運用や保全を考慮したシステムの開発も重要になってきた。顧客起点でバリューチェーン全体を見据えることにより、顧客価値を高めるアプローチが求められている。

### (3) 協創の重要性

「一連の体験」をシステムとして構成するのは、必ずしも自社製品だけではなく、体験の価値を高めるためには、他社との協力も見逃せない。日本情報システムユーザー協会 [3] の調査では、9割近くの回答者が「他社との連携が必要」と回答した。相乗効果に加えて、迅速なサービス提供のためにも強みを合わせる必要があり、マルチベンダー化やサードパーティの巻き込みなど、パートナー戦略の重要性が増している。国内にとどまらず、グローバルな提携も積極的に検討する必要が生じている。多様なパートナーとの協創に向けては、標準化、規格化など、協力するためのオープンな基盤作りも重要になった。

オープンイノベーション、M&A、ジョイントベンチャー、アライアンスやコラボレーションなど、様々な形で外部経営資源を活用して、自らの能力を補完、強化し、俊敏な経営を目指す動きが高まっている。その方針に合致した組織見直しも進められている [5]。

### (4) 不確実性の増大

従来にない挑戦的な活動には不確実性がつきものであり、リスクへの対応が求められる。新たな製品／サービスを実現するには、一定のリスクをとりつつ情報収集や関係構築を図るなど、多面的なアプローチも求められる。数多くのものがつながり、複雑さが増大すれば、システム全体の挙動を事前に把握することは難しくなり、やってみなければわからないことが多々発生する。このような場合には、仮説検証を含む反復型アプローチが必要になる。将来に向かってビジョンを掲げることは必要だが、綿密な計画を立ててもその通りには進まないことを前提にせざるを得ない。

さらに、ビジネス環境の変化が早く、予想が立てられない。企画時に考えたシステムが、外部環境の変化により、環境と適合しなくなり、変更を余儀なく

されることも多い。このような変化に対応するためには、可能な限り変化に対応可能な設計をし、変化を捉え、変化に機敏に対応することが求められる。

## 1.2 システム開発の課題

前述した環境変化の下、種々の課題が顕在化し始めている。主なものとして、以下の事項が挙げられる。

- (1) 考慮すべき範囲、条件の拡大
- (2) 必要な技術、専門知識の広範さ
- (3) 製品／サービスを実現するシステム自体の複雑さ

以下に、その内容について述べる。

### (1) 考慮すべき範囲、条件の拡大

事業として考慮しなければならない範囲や、開発対象となる製品／サービスの範囲が拡大している。製品／サービスをシステムとして実現する専門家だけでなく、さらに広範なステークホルダの存在を意識する必要がある。第2章

【地域活性化】の事例では、出版社、自治体、商店街、交通事業者、プロ野球球団など、多様なステークホルダとの関係構築が必要になった。

顧客や利用者のニーズや行動を深く理解する必要性が増し、業界や国境を越えた協業が必要になることもある。スポンサー、投資家との関係も重要である。新たな基準や規制の作成、あるいはそれへの対応が求められることもある。

提供される製品／サービスの開発、提供、運用、そして廃棄に至るまでのそれぞれの時期に応じた検討、設計、対策を企画段階に考慮して開発を進めることが必要になる。特に長年の運用の間にどんなことが発生するのか、多面的に検討する必要がある。第2章【デジタルATC】の事例では、広範囲な設備を日常の運行に支障をきたさないように更新したため、長期的な計画や、新旧両システムの並行運用が必要になった。

市場に提供された製品／サービスが、設計時には想定しない形で広く利用されることがある。歩きスマホは社会問題化した。この事態を事前に予測できたであろうか。ビットコインは、2017年の1年間で20倍高騰し、12月以降は急落して、先行きは読めない。新たな製品やサービスは、急速に変化する社会の中で利用されることを想定しなければならない。

広く提供される製品／サービスには、安全性や情報セキュリティに関する配慮も重要である。特に様々なものがネットワークに接続され、遠隔で情報を入力でき、便利な機能が利用できるということは、様々な脅威に備える必要が出

てくる。人体への影響、悪意による制御、情報漏えいなどの防止に向けた対応が求められる。スマートスピーカーの普及で何が起きるのか、楽しみであると同時に不安も残る。安全性など、部分での対応は意味をなさず、拡大するシステム全体での対応を考慮しなければならない。

## (2) 必要な技術、専門知識の広範さ

機械、電気、電子、通信、コンピュータソフトウェアといった技術を組み合わせ、対象領域によっては、化学、材料、建設、土木、医療、金融など、様々な専門知識が必要になる。従来交流したことのない専門家同士が協力して問題解決にあたる必要がある。技術面だけでなく、様々な業務知識を持った有識者との連携が求められるだろう。それぞれ異なる歴史や背景、管理やコミュニケーションの慣行を持った専門家が協力して、統合されたシステムとして製品／サービスを実現することが重要である。

第2章【電子お薬手帳】の事例では、医療情報システムの専門家と情報セキュリティ技術者の連携が必要であった。第2章【自動車エンジン】の事例では、同じ会社の内燃機関の技術者達の間で専門分野ごとに高度化と分業化が進んでおり、全体最適のために多数のチームが組織の壁を越えて連携する必要があった。第2章【Web スキャン】の事例では、スキャナー単体の性能アップに努めてきた開発者達が、サービスの視点に立って海外のASP業者との協業に乗り出した。

## (3) 製品／サービスを実現するシステム自体の複雑さ

多種多様なセンサーや機器が大量にネットワークに接続され、そこから発生する膨大なデータを効率的に処理、分析し、さらに人工知能を利用して学習、判断を迅速化して、付加価値を生み出すことが可能になってきた。生産設備や自動車、さらにはロボットなど高度な機械にIoTが組み込まれ、製造業のみならず農業やサービス業をも大胆に変革していくことが予想できる。作成される製品／サービス自体が従来とは異なった種類の複雑さを有することになる。

第2章【Web スキャン】の事例では、従来パソコンに単体で接続して使用していたスキャナーを、クラウドから利用可能にするために、新たなAPIを用意する必要が生じ、セキュリティ対策も高度になった。第2章【自動車エンジ

ン】の事例では、電子制御によって多数の要因を繊細に制御できるようになったために自由度が増し、その最適化問題が飛躍的に複雑になった。

## 1.3 システムズエンジニアリングの重要性と 4つのポイント

システムズエンジニアリングは、システムを成功させるための複数の専門分野にまたがるアプローチと手段である。ここで言う「システム」は、コンピュータシステムにとどまらず、機械、電気機器、人間系（操作者）、環境など広い意味を持っている。

システムズエンジニアリングは、航空・宇宙等の領域で長年にわたって培われてきた企画・開発のアプローチを汎用的に体系化したものである。ソフトウェアシステムやハードウェアシステムだけでなく、新規事業の開発、社会システムの設計など、様々な領域に適用可能である。

システムズエンジニアリングの効能としては、以下がある。

- ◆ 本来の目的を明確化し、より高い視点や幅広い知見を取りまとめてアプローチする方法なので、新たなビジネスチャンスの把握やリスクの明確化、既存の製品／サービスの刷新が可能となる。
- ◆ 関係者が共通的に理解できるような概念整理を行い、繰り返しブラッシュアップすることにより、これまでにない製品／サービスを生み出すことができる。
- ◆ 製品／サービスの構成を、意図を持って明示的に設計することにより、環境変化に対応可能で、安全性などの立証が可能な製品／サービスを実現することができる。

これから求められる製品／サービスの開発においては、前節で指摘したように、考慮すべき範囲や条件が広がり、用いる技術や専門知識が増大して、最終的に複雑なものを作らざるを得ない。そのような課題に対処していくには、システムズエンジニアリングが提供してくれる効能が大きな支えとなる。

ここでは、システムズエンジニアリングの特徴的なポイントを4つに絞って説明する。この4つのポイントは、既刊の「システムズエンジニアリング導入の薦め」において説明しており、ここでは、本書で紹介する事例と結び付けて説明する。

## (1) 目的指向と全体俯瞰

システムの企画・開発においては、限られた経営資源（ヒト、モノ、カネ）でゴールに到達することが必要である。また、システムをライフサイクルという時間的視点や、システムを取り巻く環境などの空間的視点から俯瞰することが必要である。

そのために、システムズエンジニアリングでは、

- ① 目的指向の考え方によるアプローチ
  - ② 全体俯瞰の考え方によるアプローチ
- のふたつのアプローチを行う。



- ① 目的指向の考え方によるアプローチ

解決策を考える前に本来の目的を明確に定義し、常に目的を意識しながら考えるアプローチを指す。

第2章【地域活性化】では、外部からの集客という目的のために、多数の手段を講じ、その中の施策として聖地巡りスタンプラリーが企画された。同じく【自動車エンジン】の事例も、顧客に「走る歓び」と「優れた環境・安全性能」を提供するという目的を明確にするところから始まっている。

- ② 全体俯瞰の考え方によるアプローチ

視点と視野を変えながら全体を俯瞰して対象を捉えるアプローチを指す。視点としては、時間的視点、空間的視点、意味的視点がある。

まず、時間的視点での俯瞰とは、例えば製品開発の初期から利用終了後の廃棄までのライフサイクル全体を考えることである。その中には、置き換え対象となる製品の世代交代までを円滑に行うことを含んでいる。第2章【デジタルATC】の事例では、社会インフラとして日々利用されている旅客サービスを止めずに、新システムへの移行を実現するための設計を行っている。

次に、空間的視点での俯瞰とは、システムを取り巻く環境における関係性を考えることである。ライフサイクルのステージごとに、その関係性が変わることも考慮しておかなければならない。【デジタルATC】の事例では、近接する



複数の車両に搭載する装置、地上の制御装置、信号を伝達する信号配線と線路など、広範囲に分散するシステム要素を含めた検討、設計を行っている。

さらに、意味的視点での俯瞰とは、ユーザーはどのような目的でその製品／サービスを必要とするのか、どのように使うのか、どの機能を必要とするのか、何を用いて機能を実現するのかなどといった観点で検討することを意味する。第2章【電子お薬手帳】では、患者とその家族、薬剤師、医師が、調剤情報にどう関わるのか、詳細に分析している。

実際に製品／サービスを企画・開発する際には、まず「対象とするもの」の課題を認識し、目的を明確にする。次に、時間的視点、空間的視点、意味的視点に沿って、ライフサイクルに着目した考察、課題に関連するコンテキスト（文脈、背景情報、状況）の考察、およびステップを踏んだ考察を付加していく。最終的に、最初の課題の解決策をビジネスに貢献するものとして構築していく。

## (2) 多様な専門分野を統合

IoT時代の製品／サービスは、大規模化、複雑化が進んでいるため、前項でシステムの全体俯瞰を進めることが重要であると述べた。この全体俯瞰を行うと、多様な専門分野の存在が明らかになり、それらを統合することが必要になってくる。

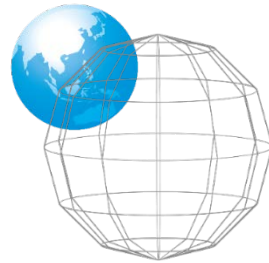
そのような状況では、システム全体として求められる特性や特徴を実現するため、特定の専門分野を深めるだけでは達成できない高い付加価値を目指して、多様な専門分野に関わる知見を統合した全体設計を行う。つまり、単に複数の専門知識を束ねるのではなく、それぞれの考え、意見を取りまとめることにより新たなシステムを作る。

第2章【電子お薬手帳】の事例では、医療情報システムの専門家と、情報セキュリティ技術の専門家が従来にない協力体制を敷いて、システム開発にあたった。また、【地域活性化】の事例では、出版社、自治体、商店街、交通事業者、プロ野球球団などが、それぞれの特性を活かして、集客に貢献した。



### (3) 抽象化・モデル化

IoT時代の製品／サービスの企画・開発には、多様な専門分野の関係者が関与し、これらの人々による共通理解、本質理解のために、概念やシステムを抽象化・モデル化するアプローチが必要になる。抽象化・モデル化とは、対象（概念やシステム）を目的にあった視点から捉え、その仕組みを適切なレベルに単純化して捉えること、およびその単純化した構造で表現することである。対象を俯瞰的に捉えるためには、抽象度を上げることが必要となる。



その実施に際しては、抽象化の視点を柔軟に設定し、目的のために必要となる複数の適切な視点から対象を構造化することが重要である。それによって構造的に整理されたシステムに関する様々な情報のネットワーク（相互関係）が得られる。さらにその関係を分かり易い形でモデルとして可視化することが重要である。

効能として、異なる関係者間の視点から見た対象を統合することが可能となり、関係者間の共通理解を促進することができる。さらに、精度が高く自由度の大きいトレーサビリティの実現、それによる各種エンジニアリング活動との有機的なつながりの実現なども期待できる。加えて、モデル化により各種検討を機械可読とし自動化につなげることで開発効率を改善できる。

第2章【自動車エンジン】の事例では、内燃機関で生じる損失の原因を詳細に分析し、そのモデルを構築して内燃機関の構成要素の関係を明確化した。それによって各部に課される目標を明確化するとともに、依存関係のモデルを利用して、検証、チューニングを行い、高度な最適化を実現した。

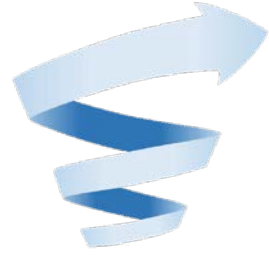
## (4) 反復による発見と進化

システム開発の初期段階では、目的とする製品仕様を明確に把握して、確定することが難しい事例が出てきている。従来存在しない新たな製品／サービスに対して要求事項を決めることは誰にとっても難しい。このような場合に、避けられない「初期の不確定要素」への戦略的な対応を図ることが必要である。

システムズエンジニアリングでは、設計と評価を対応させ、さらに目的達成を考慮しながら、適切に再評価とフィードバックを反復して、新たな解決方法を発見し、段階的に明確化・進化させることを行う。これにより、想定していた環境が変化した場合への柔軟で迅速な対応、「初期の不確定要素」への戦略的な対応ができる。

反復は、全体プロセスを繰り返すことだけではなく、途中段階の部分プロセスにおいて、システム全体としての目的達成に寄与できるかを都度評価し、必要ならば反復的に設計、評価を繰り返していくことを示している。具体的には、開発プロセス全体を通して、想定する使用者と協働して、仮想実験やプロトタイピングを駆使し、評価、確認を繰り返し実行する方法がある。

第2章【地域活性化】の事例では、まず聖地巡りスタンプラリーを成功させた後、参加者の再訪を促進するために必要な情報を収集すべく、2回目にはスマホアプリによるスタンプラリーを実現して、参加者の行動分析に必要な情報収集を可能にした。



## 1.4 システムズエンジニアリングの基本的な概念

システムズエンジニアリングが提供する効能を享受するには、基本的な用語や概念構造を理解する必要がある。ここでは、前節で取り上げたシステムズエンジニアリングの主要なアプローチの背景にある考え方や、第2章で説明する事例の要点を理解するために必要な用語と概念を解説する。当節に出てくる用語は、後述の4.5節に国際規格の用語定義を掲載している。

「システムズエンジニアリング」は、様々な分野にまたがる対処法 (interdisciplinary approach) であり、以下のような特徴がある。

- ◆ ステークホルダのニーズ、期待、および制約から解決策 (ソリューション) を導く
- ◆ その解決策のライフサイクル全体を取り扱う
- ◆ これらに必要な技術面および管理面の活動全体を統制する

システムズエンジニアリングは、特定の技術領域に依存せず、抽象的であると同時に汎用的に適用可能な知識体系としてまとめられており、問題解決に取り組むマインドセット (思考様式、価値観) とも言える。

ここでの記述は簡潔さと分かり易さを優先しており、厳密な定義は ISO/IEC/IEEE 15288:2015 などを確認して頂きたい。代表的な文書としては以下のものがある。

- ◆ ISO/IEC/IEEE 15288:2015 “Systems and software engineering - System life cycle processes” [6]
- ◆ INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th edition [7]

### (1) システムと階層構造

「システム」とは、「相互作用する要素群であり、目的を達成するように編成された組み合わせ」である。例えば、ハードウェア、ソフトウェア、データ、人間、プロセス、手続き、設備、材料などが、システムを構成する要素になることがある。また、ひとつのシステムが、製品やサービスになることがある。

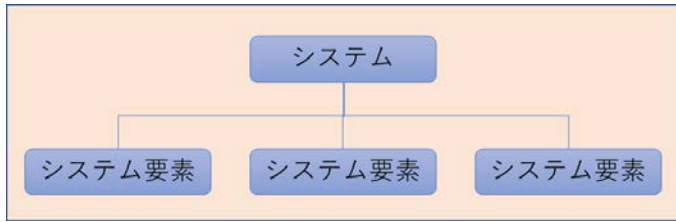


図 1.4.1 システムの構造

図 1.4.1 のようにシステムは複数のシステム要素（サブシステムと言われることもある）で構成され、システム要素間には相互作用がある。システム全体としての目標を達成するために、配下のシステム要素には、それぞれ要求事項が割り当てられる。システム要素は、さらに複数のシステム要素で構成されることがある。したがって、システムは多段階の階層構造で表現される。図 1.4.2 は、システム要素 C が見方を変えれば、システム X と見なすことができることを表している。

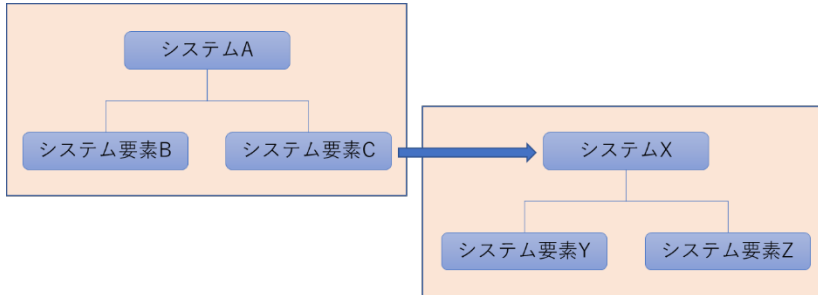


図 1.4.2 システムの階層構造

このような階層構造を利用して複雑なシステムを表現し、検討の範囲を明示化し、限定することができる。さらに複数の人やグループ、組織の間で役割分担や責任範囲を明確化することも可能になる。検討の視点、観点を変え、分解と統合を繰り返すことによって、新たな気づきやアイデアにつなげることができる。

## (2) 対象システムとコンテキスト

一般に検討対象、すなわち構築し運用しようとしているシステムの境界 (Boundary) 内を「対象システム (SoI: System of Interest)」と言う。対象システムを意識することで、問題やリスクが明確になることが多い。

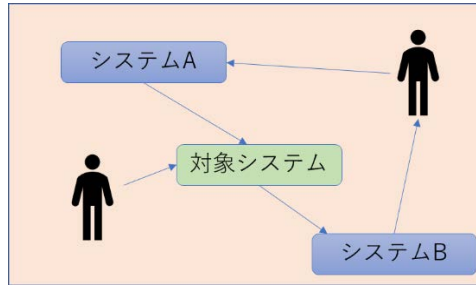


図 1.4.3 対象システムとコンテキスト

対象システムを取り囲む環境を「コンテキスト」と言う。コンテキストは、対象システム以外のシステムやステークホルダなどによって構成される。図 1.4.3 は、対象システムが、システム A、システム B、それに何人かのステークホルダとやりとりをしていることを表している。このようなコンテキストを考えることによって、対象システムに対する要求事項や制約を明らかにすることができる。自らが作ろうとしている対象システムだけに注目しても、要求事項は固まらない。対象システムの外側を検討する必要がある。

第 2 章【地域活性化】の図 2.2.1、図 2.2.2 および図 2.2.3 では、集客を増やすための取り組みを SoI とし、SoI と環境との関わり合いを示している。

## (3) 妥当性確認と検証

ビジネス上の目的を達成するシステムが得られるかを確認する活動には、「妥当性確認」と「検証」がある。

**妥当性確認**は、

客観的証拠を提示することによって、**特定の意図された用途または適用に関する要求事項**が満たされていることを確認する。

**検証**は、

客観的証拠を提示することによって、**規定要求事項**が満たされていることを確認する。

(JIS Q 9000 の用語定義 [8])

対象システムのコンテキストを検討して、対象システムに求められる機能や制約、対象システムが目的を達成できる理由を明らかにし、それらがアーキテクチャや設計に反映され、実装に引き継がれていく必要がある。構築する対象システムが、意図した環境で意図した目的を達成することを上流で確認し、その後の実装など、ライフサイクル全般にわたって達成されていることを確実にする活動が妥当性確認である。検証は、要求定義書や設計書の記述の通りに実装されていることを確認する活動であり、妥当性確認の前提になる。

確認すべき妥当性を定義することは、ビジネスの成功を定義することに等しく、エンジニアリングにとって根源的に重要なテーマである。システムズエンジニアリングは、何をすべきか、経営方針を定めるような方法ではないが、一定の目的が示された後に、その目的を達成するために、エンジニアリングの観点から適切な目標の再定義を行う。コンテキストの分析によって、ビジネスとエンジニアリングの間に橋を架け、妥当性確認によってその関係を維持することになる。

#### (4) アーキテクチャ

対象システムの内部構造は「アーキテクチャ」と言われることがある。アーキテクチャとは、システムの基本的な考え方や特性であり、システム要素およびそれらの関係として表現され、システムの設計や発展を支配する原則をなしている。目的に合致したアーキテクチャを作成し、将来の環境変化にも適応し易い構造にできれば、製品／サービスとしてシステムの価値を高められる。

対象システムがその目的を達成するようシステムを設計する上で、適切なアーキテクチャを定義することは本質的に重要である。対象システムをどのようなシステム要素群に分割し、機能や制約をどのシステム要素に割り当てるのか、その後の設計の具体化や実装、構築や試験、そして運用や保守にも大きな影響を及ぼす。

#### (5) ステークホルダ

現実のシステムには、多様な人々が関与することが多い。顧客、供給者、システム構築に参加している多様な専門家など、対象システムに影響を与え、また出来上がったシステムから影響を受ける人々を「ステークホルダ」と言う。それぞれに立場と関心事、知識や文化が異なり、コミュニケーションをとるこ

とさえ困難な場合もある。対象システムに何が求められ、どうすればステークホルダの期待に応えられるのか、適切なステークホルダを特定し、しかるべき協力が得られるように巻き込んでいくことは、目的達成に欠くことができない。

このような人々の間のコミュニケーションでは、適切に絞り込んだ情報で論点を明らかにする必要がある、適切な抽象化によって、分かり易いモデルを示して、理解を求めたり、情報を引き出ししたりすることが重要である。

## (6) ライフサイクル

システムには、それが着想されてから、構築、運用され、廃棄されるまでの進化発展の段階がある。この期間を「ライフサイクル」と言う。ライフサイクルも対象システムを検討する上で、重要な視点を提供する。立ち上げ段階から、要求定義と設計、統合、移行、運用など、システムが目的を果たすために必要な一連の作業の関連性や依存関係を明確に意識して、活動の計画と管理を行うことが重要である。またシステムの移行、運用、保守、廃棄といった段階における要求事項や制約を、上流の要求定義や設計の段階で考慮することも重要である。ライフサイクルという捉え方によって、このような検討の視点が提供される。



## 1.5 本書の構成

本書は、事例を通じてシステムズエンジニアリングの考え方が問題解決のどのような場面でどのように活かされるのか、どう使うことでどのようなベネフィットがあるのかを説明している。

第2章で取り上げる事例では、第1章で取り上げた今日的な事業上の課題、または製品／サービスを実現するための課題に向き合い、その解決に向けて、結果としてシステムズエンジニアリングのアプローチが有効に機能したと解釈できる事例を紹介する。そして、システムズエンジニアリングの特徴的な4つのポイントや、第4章で紹介する ISO/IEC/IEEE 15288:2015 で言及されているプロセス、アクティビティに対応付けている。なお、ここで取り上げる事例は、システムズエンジニアリングを体系的、意識的に適用しようとして取り組んだものではなく、それぞれの課題解決の過程で、システムズエンジニアリングのアプローチを効果的に使いこなしたと見なせる事例である。

第3章は、システムズエンジニアリングを意図的に取り上げようとした事例を引用している。それぞれの状況において、システムズエンジニアリングを学習しながら利用しようとした事例である。エンジニアリング活動としての成否を評価するところまで進行していないが、その準備や試行の過程を紹介している。問題解決にシステムズエンジニアリングを利用しようとした場合、何を考え、どうすれば前進できるのか、どのような障害が立ちはだかり、それをどう克服するのか、ということを考える上でのヒントを提供している。

第4章には、ISO/IEC/IEEE 15288:2015 の中から第2章の事例と関連の深い部分を掲載している。この規格は、システムズエンジニアリングの重要な概念や必然的に実施することになる活動を記述している。これらはシステムズエンジニアリングの専門家達が何年もかけて議論し、取りまとめ、洗練を重ねた結果である。システムズエンジニアリングを使いこなすには、先達の知恵を借りるべきであり、多くの事例に触れることが重要である。第2章の事例と国際規格として整理された知識を対比しながら、システムズエンジニアリングに対する理解を深めて頂きたい。システムズエンジニアリングにおいても他の学問領域と同様に、正規の教育は専門用語を多用する。本書はシステムズエンジニアリングの専門家の解説を理解するための基礎として、重要な用語定義や概念、基本的なアクティビティやタスクを紹介している。

第2章の事例に登場するアプローチの有効性に着目し、それをご自身の業務に活かせる方はぜひそうして頂きたい。しかし、特定の状況に依存した事例がそのまま使えるケースは少ない。事例の中の実践を、国際規格の用語や概念に照らし合わせて一旦抽象化し、事例を消化してそのエッセンスを汲み取って頂きたい。また、事例と国際規格を照らし合わせることを通じて、国際規格が伝えようとするものの真意を読み取って欲しい。加えて、読者が事例や国際規格の中に有効なアプローチを見出したとして、それを組織的に活用して成果に結び付けるには、周囲の人達の理解を得る必要もある。第3章に示した事項をヒントとして粘り強く取り組んで頂きたい。

本書は、事例を通じて、システムズエンジニアリングを利用した問題解決へのアプローチを学び、それを整理された体系的な知識の枠組みの中に位置づけて、応用可能な知恵にする過程を支援することを狙っている。それによってシステムズエンジニアリングの思考様式や価値観を伝えたいと考えている。本書を皮切りに、システムズエンジニアリングの専門家のコミュニティに参加したり、事例を学んだり、体系的な学習を始めたりする一方で、まずは断片的であっても学んだことを実践して、問題解決の能力を高めて頂きたい。

## 第2章 事例でみるシステムズ エンジニアリングによる問題解決

本章では、ビジネスやプロジェクトの課題に対し、システムズエンジニアリングのアプローチに合致する様々な活動を実践することにより、システム開発に成功した事例を紹介する。

## 2.1 事例の読み方

本章では、システムズエンジニアリングのアプローチを用いて製品／サービスを開発した事例を紹介し、それぞれの事例において具体的にどのように実践したのか、どの部分にシステムズエンジニアリングのアプローチを活用したかという例を記載している。

### 2.1.1 事例の記載項目

各事例は以下の構成で記載されている。

#### 1) 事例の内容

事例の概要と、その事例における、システムズエンジニアリングの観点での特徴となる部分を説明している。

#### 2) 事例の背景と課題

事例が対象とするビジネスや、プロジェクトの背景および課題を説明している。

#### 3) アプローチ

事例が抱える課題に対して、どのようなアプローチで進めたか、どのような対策を立てたのかについて説明している。

#### 4) 対策の詳細

アプローチに示した対策のそれぞれの詳細について、どの部分がシステムズエンジニアリングの観点に関係しているのかを説明している。

この点を捉え易くするために、第1章で説明した「システムズエンジニアリングの4つのポイント」、および第4章で説明する「システムズエンジニアリングプロセス」と対応するものには、説明文中に【】の形で記載している。

#### 5) 対策の効果のまとめ

対策において、システムズエンジニアリングのアプローチを実践したことによる効果を説明している。

#### 6) 教訓

システムズエンジニアリングのアプローチを実践することにより、それまでの開発の進め方からどう変化したか、システムズエンジニアリングのアプローチを実践していなかったらどうなっていたか等、事例で得られた気づき・教訓について説明している。

## 7) ポイント×プロセス対応表

事例で実践された「システムズエンジニアリングの4つのポイント」と「システムズエンジニアリングプロセス」との対応関係を、表にまとめている。

## 2.1.2 各事例で実践されたプロセス

各事例では、実践されたことがシステムズエンジニアリングプロセスのどこに該当するかを【】で記載している。また各プロセスの解説を本書の第4章に記載している。それぞれの実践内容とプロセスの解説とを相互参照し、対応付けながら事例内容を理解することで、読者自身が担当しているプロジェクトへ適用する際の参考にして頂きたい。

## 2.1.3 ポイント×プロセス対応表

それぞれの事例には、ポイント×プロセス対応表を記載している。実践したことが第1章で説明したシステムズエンジニアリングの4つのポイント、および第4章に掲載したシステムズエンジニアリングプロセスのどこに位置し、どのようなポイントが活かされているのかを把握することができる。

	ポイント				
	目的指向と全体俯瞰	多様な専門分野を統合	抽象化・モデル化	反復による発見と進化	その他
(6.4.1) 【ビジネスあるいは ミッションの分析】			4つのポイント		
(6.4.2) 【利害関係者ニーズと 要求事項の定義】					
(6.4.3) 【システム要求事項の 定義】					
プロセス (6.4.4) 【アーキテクチャの 定義】	代表的なプロセス				
(6.4.6) 【システム解析】					
(6.4.9) 【検証】					
(6.4.11) 【妥当性確認】					

図 2.1.1 ポイント×プロセス対応表

## 2.1.4 事例一覧

第2章では、システムズエンジニアリングのアプローチに合致した活動により問題解決をした5事例を紹介する。各案件の概要と、システムズエンジニアリングの4つのポイントのうち事例で適用したポイントを下記に示す。

表 2.1.1 事例一覧

章節	案件名	概要	適用したポイント
2.2	<b>地域活性化</b> イベント向け情報共有基盤の開発	地域活性化を目的として、多様な関係者を巻き込みながら都度ステークホルダのニーズと要求を明確化し、全体を俯瞰しながら段階的に対象システム(SoI)を拡張・改変して価値を高め、継続的な地域活性化活動につなげた。目的指向と全体俯瞰を徹底して超上流プロセスを繰り返し、効果的にシステムを実現した事例。	<ul style="list-style-type: none"> <li>目的指向と全体俯瞰</li> <li>多様な専門分野を統合</li> <li>抽象化・モデル化</li> <li>反復による発見と進化</li> </ul>
2.3	<b>電子お薬手帳</b> システムに適用したセキュリティ設計	医療サービス向上の施策として、「お薬手帳」の電子化を実現した。目的指向と全体俯瞰を通してセキュリティの課題を浮き彫りにして、医療とITという複数の分野にまたがる複雑な問題に対し、抽象化・モデル化を活用した系統的なアプローチでソリューションを創出した事例。	<ul style="list-style-type: none"> <li>目的指向と全体俯瞰</li> <li>多様な専門分野を統合</li> <li>抽象化・モデル化</li> </ul>
2.4	多様な要求を満足させる <b>自動車エンジン</b> の開発	内燃機関開発にあたり、センサー、部品、アクチュエータ、ソフトウェア等の個別部品を個別最適に設計するのではなく、これらが統合されるシステムとして捉え、全体最適の観点から機能目標を定義して設計を進めた。個々の部品の物理設計に先行して機能開発することで、設計から検証まで含め、効率的に開発を実現した事例。	<ul style="list-style-type: none"> <li>目的指向と全体俯瞰</li> <li>多様な専門分野を統合</li> <li>抽象化・モデル化</li> <li>反復による発見と進化</li> </ul>
2.5	首都圏の高密度輸送を支える <b>デジタルATC</b> の開発	開発当初から全工程の課題を分析し、2世代先まで見通した段階的な開発を行った。早い段階から移行や運用までも視野に入れて課題を見据え、要件や設計に反映した結果、試験時間帯の制約などの課題を克服し、開発を実現した事例。	<ul style="list-style-type: none"> <li>目的指向と全体俯瞰</li> <li>反復による発見と進化</li> </ul>
2.6	<b>Webスキャン</b> システムの企画開発	成熟しつつある製品であるスキャナーについて、従来と異なる一段上の視点からビジネスシーンを俯瞰し、ビジネス分析およびステークホルダ要求分析を行った。これにより新しい使い方の提案を導き出し、新たなサービス提供に結び付けた事例。	<ul style="list-style-type: none"> <li>目的指向と全体俯瞰</li> <li>多様な専門分野を統合</li> <li>抽象化・モデル化</li> <li>反復による発見と進化</li> </ul>

## 2.2 地域活性化イベント向け情報共有基盤の開発

### 2.2.1 本事例の内容

#### (1) 概要

本事例は、アニメ作品の「聖地巡り」を軸としたアニメコラボ集客イベントによって、経済効果と若者の集客の両面で「地域活性化」を図ることを企画して、実際にイベントを推進した事例である。

自治体、商店街、交通事業者、プロ野球球団、出版社、など、多様な利害関係者の協力を得て、第1回のイベントは多数の集客を実現し成功した。中でも「聖地巡り」は「紙のスタンプラリー」として実行し、多数の参加者を得ることができた。

第1回のイベントで「聖地巡り」の集客力を実証できたので、スタンプラリーの顧客情報を取得する仕組みを確立し、顧客像を把握して再訪を促すような継続的な活動につなげることが次の課題であった。

この課題を解決すべく、データ共有に関する要件を十分に吟味して解決策を探った結果、ステークホルダ間の公平な関係でのデータ共有を実現するための技術的解決策として、ブロックチェーンを利用した仕組みを発案することができた。これにより第2回のイベントとしてスマホアプリによるスタンプラリーを実現した結果、顧客情報を取得し、参加者の行動の詳細分析による付加価値の高い情報取得が可能になり、地域活性化に向けた活動として継続していく道筋ができた。

#### (2) 本事例の特徴

地域活性化という目的に対して、アニメの聖地巡りという対策からのアプローチを実現するため、多くの関係者を巻き込み、それぞれにメリットがある関係を構築した。昨今の新サービスの実現には適切なパートナーとの協創が必要であり、異なる目的を有するパートナーとWin-Winの関係を構築することが鍵になる。その際に目先の目標（例：イベントの集客数）に拘泥することなく、高い視野から状況を分析し、ステークホルダと共有できる目的の設定が重要になることを示している【目的指向と全体俯瞰】【多様な専門分野を統合】。

また、段階的に対策を発展させ、それぞれが抱える懸念を解消しつつ、ステークホルダとの関係を深めていくことで、協創成果の価値を高めていくことができるということを示している【抽象化・モデル化】【反復による発見と進化】。なお、反復の過程で本来のビジネスの目的に立ち返って検討していることは大きな特徴である。



2. 事例でみるシステムズエンジニアリングによる問題解決

## 2.2.2 本事例の背景と課題

### (1) ビジネスの背景

本件は、2015年および2017年に、千葉市にて地域活性化イベントを行った事例である。

千葉市では、ここ数年地域としての集客力が減少しており、2016年度には、千葉駅周辺のふたつの大型商業施設が撤退している。特に、商業施設のひとつは若者向けであったこともあり、千葉駅周辺からの若者離れが進むことへの危惧があり、若者を中心とした集客を増やして地域のビジネスを活性化するための試みを模索していた。



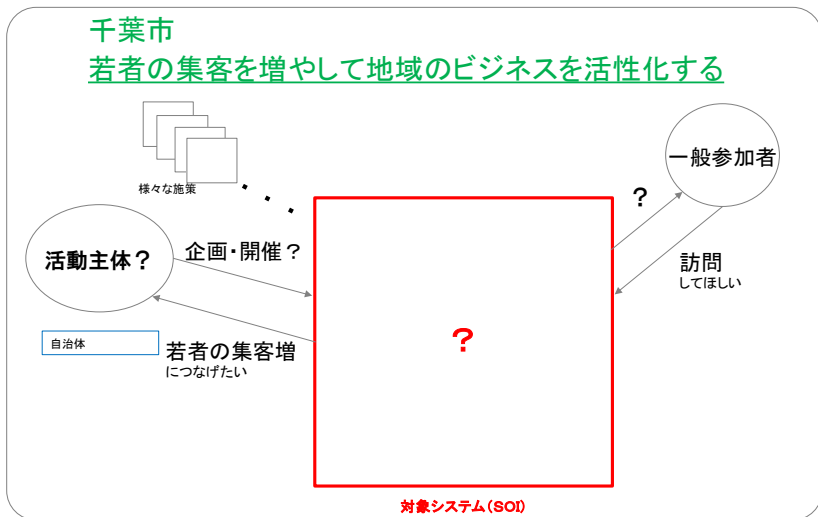


図 2.2.1 課題認識時点のコンテキスト(SoI は定まっていない)

この検討の中で以下の着想が生まれた。

ここ数年、地域活性化の成功事例として注目を集めているのが、その地域を舞台としたアニメ作品とコラボし、作品のファンに回遊してもらう「聖地巡り」という手法である。

この手法は、若者だけでなく、クールジャパンとして、インバウンドの集客にも成果を上げていることから、今回の活動に有効ではないかと考え、取り組みを開始した。

## (2) ビジネスの課題

まず、このプロジェクトの立ち上げにあたって、前提条件として解決すべきであった重要な課題について説明する。アニメ作品の「聖地巡り」という取り組みは、強力ではあるが、実現に向けていくつかの満たすべきビジネス的な前提条件がある。

まず、その地域を舞台とした、人気のあるアニメ作品が存在し、それを利用できる必要がある。また、スタンプラリーのようなイベントの枠組みを構築し、そのイベントの費用を誰かが負担する必要がある。〈→対策1〉

また、この前提を満たし、実現に向けて動き出す段階では以下のふたつの課題があった。

まず、関係者すべてのメリットの享受、Win-Winの実現である。普段全く接点のない関係者がひとつの共同のイベントによって、それぞれメリットを享受できるように実践しなければならない。各関係者にメリットをもたらす集客ができてはじめて成功と言えるのである。〈→対策2〉

ふたつめは、単なる一過性のイベントとしての集客ではなく、継続的な地域活性化の活動として継続できるやり方を確立することである。一回限りのイベントが盛り上がり、イベントが終わったらもとに戻ってしまう、そのような活動では地域活性化の活動として成功とは言えず、継続的な集客につながる活動ができる基盤作りが課題であった。〈→対策3, 4, 5, 6〉

## 2.2.3 アプローチ

### (1) 前提条件を満たすための対策

アニメ作品については、幸いにも、ライトノベル（学生など若者向けの小説ジャンル）に、千葉市を舞台とした地域密着型の作品があった。すでに500万部を越える売れ行きを示していたこと、2015年には、TVアニメ化が検討されていたことから、そのアニメを題材の候補として、権利元の出版社に、地域活性化コラボでの活用を打診した。

出版社の編集部では、3つの観点から前向きに考え、協力してくれることになった。ひとつは、TVアニメ化にあたり、プロモーションとして地域イベントは有効であるというマーケティングの観点。もうひとつは、作品が千葉を舞台としており、原作者の先生も千葉出身で地元愛が深いことから、先生も喜ばれるだろうという観点。そして、ファンにとっても、舞台である千葉でイベントが行われることは納得感があるだろうというファン目線での観点である。このようにしてコンテンツは決定した。

次にスタンプラリーの枠組み、イベントの費用の負担については、2015年時点では、政府がクールジャパン戦略の下で地域創生を重視していたため、地域商店街のための助成金があり、これを利用できる可能性がないか、千葉市に相談した。千葉市からは、いくつかの商店街を紹介してもらい、また、他に、連

携できそうな事業者として、交通事業者と、プロ野球球団を紹介してもらった。これらに働きかけを行うことにより、商店街、交通事業者、プロ野球球団が、同じアニメコンテンツを用いて地域コラボを行うという地域活性化の枠組みが決まり、前提条件が整った<→対策1>。

## (2) 2015 年のイベントにおける対策

「アニメを用いた集客による地域活性化」という目標のもとに、関係するステークホルダのニーズと要求を明確化し、さらにそのニーズを満足するシステム要求を明確化させた<→対策2>。ここで関係するステークホルダとは、自治体、アニメのコンテンツホルダ（出版社）、地域事業者としての交通事業者・商店街・プロ野球球団、ITベンダー、そして参加者として想定しているアニメファンなど多様であり、特定の主催者や出資者に依存しない協力関係での実施であった。

さらにスタンプラリーでの訪問実績等の情報を関係者間で情報共有するための手段として ICT の仕組みの検討に入り、大きな問題にぶつかった。本イベントは、特定の主催者や出資者によって推進されるものではないため、ICT の仕組みを導入するための費用を誰が負担するのか、データを誰が持つかといった問題が解決できず、想定したソリューション（スマホアプリ）の利用を実現することはできなかった。

そこで、ICT の仕組みの利用はあきらめ、参加者には紙のスタンプラリーで周遊してもらうというアナログな仕掛けで第1回イベントを実施することとした。

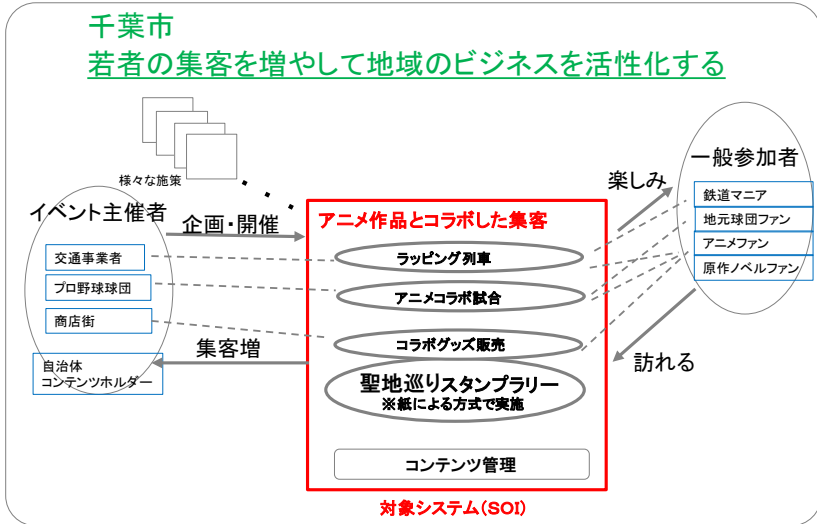


図 2.2.2 第 1 回イベント実施時点のコンテキスト

この結果、集客イベントとしては成功し、このアニメ作品の「聖地巡り」の集客効果は実証で確かめられた。しかし、参加者の属性や、動線（どこをどういう順番で訪れたかを記録したデータ）などを取得できておらず、「分析による新たな集客施策の導出」や「参加者へのプロモーション活動」などに活かして、再訪問を喚起するような活動につなげることはできなかった。このままでは一過性のイベントに過ぎなかった。第 1 回のイベントではこの一過性で終わらせないための課題が残った。（対策 3 以降として 2017 年に継続）

### (3) 2017 年のイベントにおける対策

2017 年のイベントにおいて、「継続的な集客につながる活動ができる基盤作り」という課題の解決に向け、懸案の費用負担およびデータの管理に関する現実的な対策を見出すため、ビジネスの課題に立ち戻って再検討するとともに、新たなデジタル技術を用いた情報共有基盤の可能性を検討した。

改めてデータ共有に関する要件を十分に吟味したことで、公平な協力関係でのデータ共有ができる技術的解決策が必要であることが明確になった。

そして、その解決策を探索した結果、分散台帳技術を用いたブロックチェーンの仕組みを採用することで、データの管理責任を共通に負うことができ、データの囲い込みが起きない解決策を実現できることが分かった。その案を採用

し、ブロックチェーンの実証実験に興味を持つ地元企業にも新たにスポンサーとして加わって頂いて、スタンプラリーをサポートする ICT システム（スマホを使って周遊できる仕組み）の構築を開始することができた<→対策 1, 対策 2, 対策 3 の見直し/反復、対策 4>。

構築にあたっては、各関係者の役割を整理、分析し、データ層、ビジネス層、集客層に層別化し、一般化して開発を行った。その結果、参加者の属性や動線といったデータを取得することができ、今後の地域活性化に向けた分析などに活用できることに加え、今後、付加価値を追加していくことができる拡張性に配慮した構造とした<→対策 5>。

これにより、第 2 回イベントは、紙の代わりにスマホを使用して、情報が取得できる形で実施することができた。

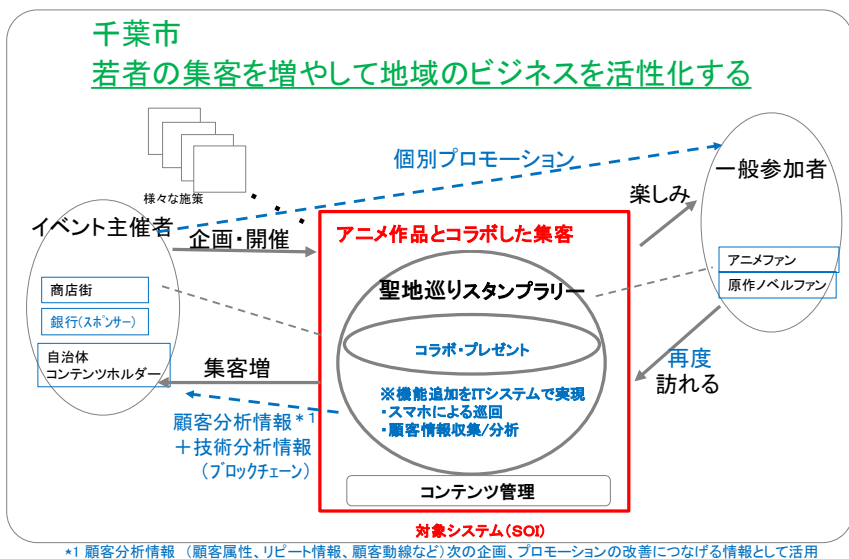


図 2.2.3 第 2 回イベント時点のコンテキスト

今後は、取得データと構築した ICT システムの拡張性を活かして、さらに付加価値を追加して第 3 回の実施を計画している<→対策 6>。(例：関連グッズ売上に つなげるための仮想通貨連携等)

また、この ICT システムを用いて他地域での同様な取り組みも広げている。

## (4) アプローチの効果

2015年の集客の成功に引き続き、2017年も地域活性化イベントとして集客、および顧客情報の取得に成功し、今後、データを活かした計画を継続していくことで、地域活性化の成果を上げていくための活動を続けていくことにつながった。

第1回は、紙ベースで実施していたスタンプラリーだったが、第2回の結果のデータを分析した結果を活かし、さらに効果を見込める施策を第3回の企画に反映している。そこでは、仮想通貨を用いたスタンプラリーを行うことを計画している。継続した地域活性化活動につなげられる仕組みを作り上げることができたと言える。

また、同じ仕組みで、新宿でもイベントを行うことができた。新宿でも、地域の様々な事業者が、共通イベントでデータをとっていかうという狙いは同じである。

### 2.2.4 対策の詳細

前項で挙げた各対策を、システムズエンジニアリングとしての特徴を挙げながら詳しく説明する。

#### <対策1> 地域連携の多様なニーズおよびビジネスミッションの分析

経済効果と「若者の集客」の両面での「地域活性化」を最上位の目的として、今回の活動の「アニメを用いた地域活性化」という目標のもとに、地域の課題は何であるかを明確にした。そして、その課題を解決するために、潮流としてのクールジャパンの地域創生予算の活用、具体的なコンテンツとしてのアニメの選定・協力体制確保の調整、それらを支援する政府の政策、そしてイベントを支えるITの仕組みなど、地域連携の場合の多様なニーズを抽出した。必要なステークホルダーを巻き込み、ビジネスミッションの解析を進めた。

図2.2.4にそれら、すべてが絡み合っていることを示す。

この対策の結果、活動の前提としてのアニメコンテンツが使用可能となり、予算的裏付けも明確になった。さらに、本プロジェクトの関係者の範囲も明確となった。

なお、第2回のイベントにあっても、対策1に相当する対策を繰り返し実施している。

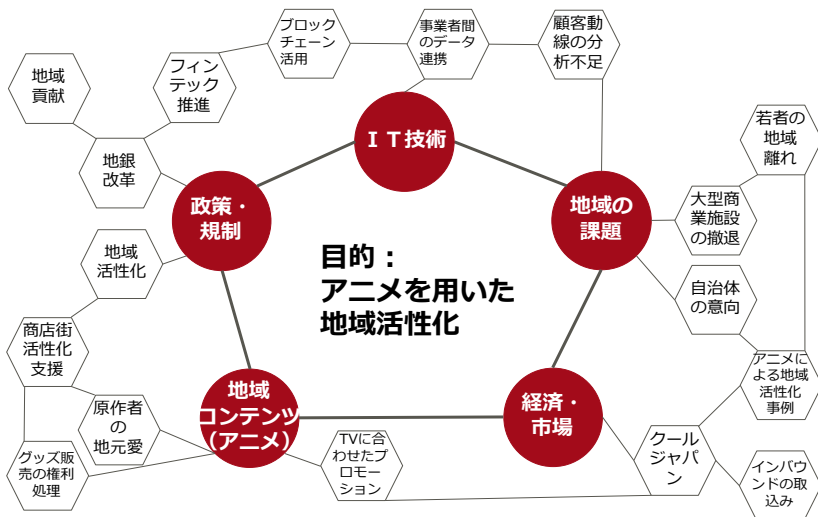


図 2.24 地域連携の相互関係の分析(2017年のイベント)

以下に、この対策における4つのポイントとの対応および、システムズエンジニアリングのプロセスとの紐付けについて示す。

【システムズエンジニアリングのポイント：目的指向と全体俯瞰】

【関連プロセス：ビジネスあるいはミッションの分析】

## <対策2> 関係するステークホルダのニーズと要求の明確化

このプロジェクトに関係するプレイヤーは、自治体、アニメのコンテンツホルダ（出版社）、地域事業者としての交通事業者、商店街、プロ野球球団、およびITベンダーがあり、それに2017年にはスポンサーとなった銀行が加わっている。対策1で協力は取り付けたが、重要なのは、それぞれが、自社のメリットを追求し、互いに得意領域を提供するという、Win-Winの関係を成り立たせることである。地域活性化イベントの場合、ともすればボランティア的なも

のやIR活動となり、持続性や広がり難が出てしまう。そこで、ビジネスモデルを踏まえて、各社のメリットを組み合わせることで、統合的なニーズを抽出し、その要求を明確化した。その際、それぞれの業界での類似事例を調査することで、実施内容を具体的に検討し、取り組みの効果を想定する根拠とした。

多様な参加プレイヤーすべてに対しニーズと要求の明確化を実施していくことで、それぞれのメリットを組み合わせ、Win-Winが形作れたからこそ、仕掛けが成り立ったと考える。

なお、第2回のイベントにあたって、対策2に相当する対策を繰り返し実施している。

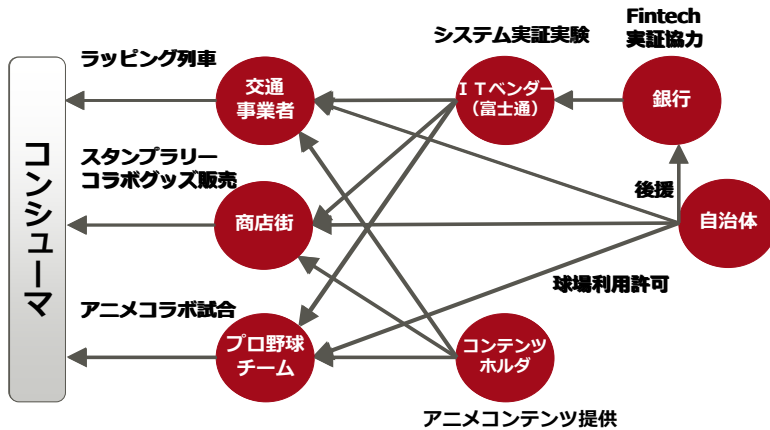


図 2.2.5 ステークホルダ要求の抽出(2015年・2017年の全体に対応)

一例を示す。地域の関係者全体にメリットを享受してもらうため、スタンプラリーで各地を周遊する方式を採用した。ただ、それだけで参加者が地域にメリットある周遊をしてくれるか、確実性が伴わない。その点の確度をより高めるためには、まず、周遊コースの設計に、地域関係者とアニメファンの双方へのメリットが仕込まれていることが重要である。

アニメファンに来て欲しければ「訪問したくなる」魅力が重要になる。そこで、「アニメの原作に登場する場所およびエピソード」と「周遊先のスポット」との関係丁寧に考慮したコース設計を行うことが必要である。



また、その前提として。周遊先のスポットとして地域にメリットのあるポイントが設定されていなければならない。もちろん、訪問して欲しい「場所」を直接スポットとするだけでなく、主要スポット間の移動経路に配置するなど、きめ細かい配慮を行っている。

この辺の妥当な効果を狙うためにも、双方のニーズを理解してきめ細かい設計につなげるが必要だった。

この対策における4つのポイントとの対応および、システムズエンジニアリングのプロセスとの紐付けについて示す。

【システムズエンジニアリングのポイント：多様な専門分野の統合】

【関連プロセス：利害関係者ニーズと要求事項の定義】

## <対策 3>システム要求の明確化 および <対策 4>アーキテクチャの明確化

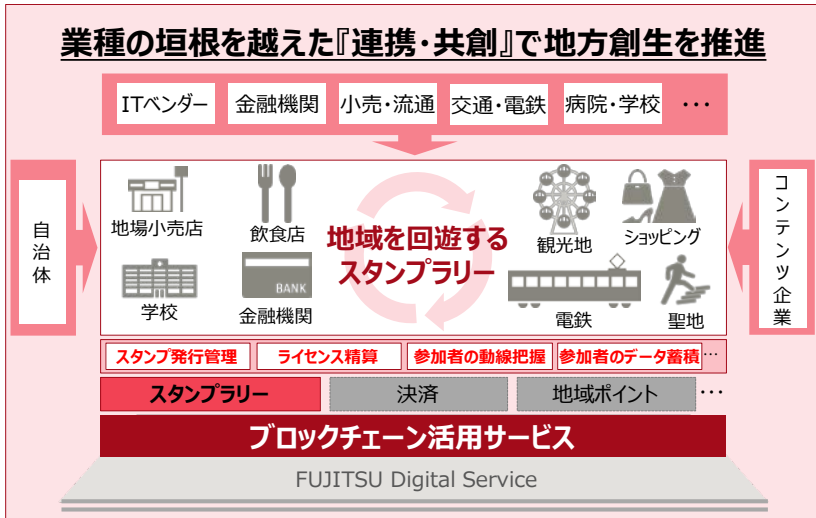
データ共有の仕組みとして、データの管理責任を共通に負うことができ、データの囲い込みが起きない技術が必要であることを導き出した。そして、2015年のイベントの反省を踏まえ、地域の複数の事業者でデータを共有する仕掛けとして、分散台帳技術を用いたブロックチェーンという仕組みを用いるアーキテクチャを採用することにした。

これは、仮想通貨などに用いられている技術であるが、複数の参加事業者が、それぞれ全く同じデータを持つことができる。地域連携の場合、参加事業者が新規に参加したり、脱退したりすることが想定され、特定の1社が継続的に責任を持ち、データを保管し続けることは難しいため、この分散台帳技術の適用は有効な仕組みだと考えられた。

そこで、この仕組みを用いて2017年の第2回イベントのインフラ構築に着手した。

前回と異なり、商店街活性化助成金の制度がなくなっていたが、政府が取り組む地銀改革の流れの中で、地方銀行も、地域活性化や、ブロックチェーンのようなフィンテック技術に関心があるのではないかと考え、打診したところ、スポンサーになって頂くことができた。

そして、IT 企業として参画した富士通も、ブロックチェーンを利用したスマホスタンプラリーを提供し、参加者の属性（年齢、性別、居住地など）や、動線を把握し、地域の事業者提供できる仕組みに結び付けることができた。



2. 事例でみるシステムズエンジニアリングによる問題解決

図 2.26 ブロックチェーンによる情報共有基盤の概念図

この対策における4つのポイントとの対応および、システムズエンジニアリングのプロセスとの紐付けについて示す。

【システムズエンジニアリングのポイント：目的指向と全体俯瞰】

【関連プロセス：システム要求事項の定義、アーキテクチャの定義】

## <対策 5> システム機能の整理と層別化、一般化

第2回スタンプラリーに向けたICTソリューションを組み上げるにあたって行った設計上の対策として、システムの機能を集客層、ビジネス層、データ層に層別化して整理した過程を以下に記載する。

イベント開催側のステークホルダは、大きく3つのレイヤーに分類できた。ひとつは、集客のためにコンテンツを提供した出版社（権利元）。次に、そのコンテンツを用いてビジネスを行った交通事業者や、プロ野球球団、商店街で

ある。最後に、データの取得・分析の仕組みを提供した富士通や、協力のために参加した銀行である。

各関係者の役割を整理・分析し、上記に対応してイベントを集客層、ビジネス層、データ層に層別化し、システム機能として一般化した。このように抽象化・モデル化することで、千葉市で繰り返し実施する際、および他のエリアで検討をする際に、事業者の選定や役割分担を考えるためのフレームワークとして活用できるものにできたと言える。

こうして組み上げた仕組みで実施した2017年のスタンプラリーも、集客面で地域活性化イベントとして成功し、さらに種々のデータを取得して分析できるようになった。それらの分析結果を活かして、来年度には第3回を行うことを検討している。

#### ■システム機能をレイヤーにて整理

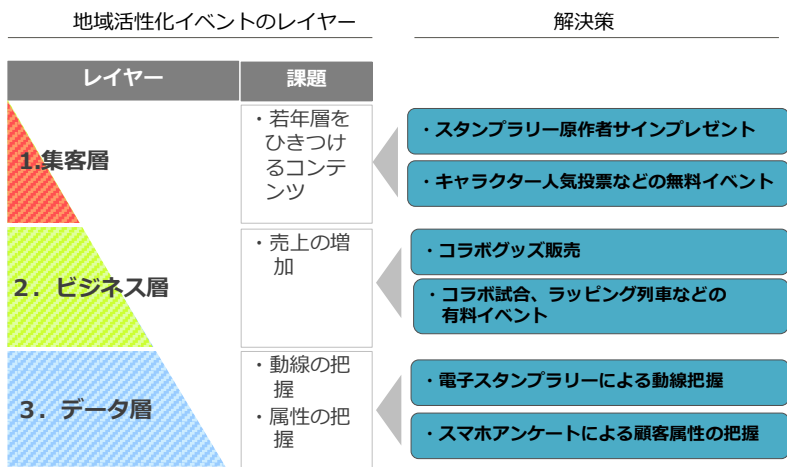


図 2.27 SoI の機能の層別整理

この対策における4つのポイントとの対応および、システムズエンジニアリングのプロセスとの紐付けについて示す。

【システムズエンジニアリングのポイント：抽象化・モデル化】

【関連プロセス：アーキテクチャの定義】

## <対策 6> 一般化して構築したインフラを用いて継続的活動の実現

一般化して構築したインフラを用いて、付加価値を付与しながら継続的に地域活性化イベントを繰り返し行っていく道筋ができた。また、同様の仕組みで、他地域でもイベントを行うことができた。ここでは、汎用的なインフラを生み出したアプローチを最初の工程から振り返った視点で特徴を記載する。

地域の様々な事業者が、共通イベントでデータをとっていこうという狙いは他地域における活動でも同じである。

これに向けて、いきなり ICT システムを作り込むことは、ワークフロー等に不安定要素が大きく、投資にも不安が大きい。また、アニメコラボによる地域活性化の可能性を検証し、コラボ実施における課題や、複数事業者連携ならではの課題を洗い出すためには、適切に課題設定して実証試験として行えば、紙のスタンプラリーでも役割は果たせる。今回の第 1 回、第 2 回の段階的機能追加のアプローチは、必要な機能を見極めるための適切なイタレーションの取り組みを行ったと考えられる。

2. 事例でみるシステムズエンジニアリングによる問題解決

### ■ 顧客の動線分析と地域におけるデータ共有の仕組みを進化

#### 2015年千葉アニメコラボ

- ・千葉市の商店街
- ・千葉ロッテマリーンズ様
- ・千葉都市モノレール様

#### 紙によるスタンプラリーを提供



著/渡 航 イラスト/ほんかん® 小学館「カガカ文庫」刊

#### 2017年千葉アニメコラボ

- ・千葉市の商店街
- ・千葉都市モノレール様
- ・千葉銀行様

#### 電子通貨の仕組みを利用したスタンプラリーを提供



参加者

システム

図 2.28 データ共有の仕組みを進化

この対策における4つのポイントとの対応および、システムズエンジニアリングのプロセスとの紐付けについて示す。

【システムズエンジニアリングのポイント：反復による発見と進化】

【関連プロセス：ビジネスあるいはミッションの分析、利害関係者ニーズと要求事項の定義、システム要求事項の定義、アーキテクチャの定義】

千葉市からは、今回は、より商店街の売上につながるよう、コラボグッズの発売を強化して欲しいと要望されている。

そこで、今回は、仮想通貨にプログラムを埋め込み、ある商品が売れるたびに、一定の割合の仮想通貨が、権利元に自動的に戻るような仕組みの試行を検討している。

イベント終了後に、その仮想通貨に応じた法定通貨を、商店街が売上から戻す仕組みである。

このように、権利関係の透明性などを追加し、全体の仕組みのさらなる進化につなげたいと考えている。

## 2.2.5 対策の効果のまとめ

各対策において、以下に示すシステムズエンジニアリングの効果が認められる。

- ◆ 「アニメを用いた地域活性化」という目的指向と、具体的コンテンツとしてのアニメの選定、政府の支援の活用、インフラとしてのIT技術活用の方向性など、ビジネスミッションを全体俯瞰して解析することで、方向を決めることができた。
- ◆ 多様な専門分野を統合し、関係するステークホルダのニーズと要求を明確化し、さらにそのニーズを満足するシステム要求を明確化させることができた。
- ◆ その結果、データ共有の仕組みとして、データの管理責任を共通に負うことができ、データの囲い込みが起きない技術が必要であることを導き出せた。また、検討の上、分散台帳技術を用いたブロックチェーンの仕組みが採用できた。

- ◆ 結果として地域活性化イベントとして成功し、次年度開催（第3回）も検討されている。

## 2.2.6 教訓

本事例を通して、以下の知見・教訓が得られた。

### (1) 出来上がったシステムの特徴

多様な関係者を巻き込んだ大きなイベントを実現することができた。その背景として多様な関係者が協力して進める仕組みを作り上げ、しかも継続的な展開が可能な仕組みにまで深めていくことができたということが大きな特徴である。

第1回の段階では、ステークホルダのニーズに基づくシステム要求を明確化した結果、その時点の解決策（一社による情報共有基盤）では、関係者の Win-Win が成り立たず、合意が形成できないことが分かったため、IT化に関わらず、一部を紙運用にしてプロジェクトをスタートさせた。その活動を通して、地域の事業連携特有の条件（事業者の入会、退会が頻繁に起こること、特定の社にデータの管理責任を負わせることが難しいことなど）を導き出した。そして、データ共有の仕組みとして、データの管理責任を共通に負うことができ、データの囲い込みが起きない技術が必要であるというシステムへの要求を明確化し、解決に取り組んだ。第2回ではそれらの要求に応えるための仕組みとして、分散台帳技術を用いたブロックチェーンの仕組みを採用して、継続的な取り組みまで深めることに成功した。

### (2) 結果を導いたアプローチの特徴

全体俯瞰、目的指向を明確にした合意形成の推進が本アプローチの特徴である。

目的指向を明確にし、そのためのアニメの採用、クールジャパン、政府の政策の利用など事業領域を超え全体を俯瞰し、関係者を割り出した。

また、自治体、出版社、交通事業者、商店街、プロ野球球団、ITベンダー、銀行というすべてのステークホルダの要求を把握し、例えば第1回には運用の一部を紙にしていなくても、まずプロジェクトを進めるといった判断をした。引き続き本来の目的達成に向けた対応の検討を続け、ブロックチェーンという新しい技術を環境に適用することで、全体の目的達成に向けた工夫を実践した。さらに、実装にあたって一般化を施し、固定のイベントではなく、継続的に価値を付加していける基盤とした。（その結果、次回は購買喚起のための仮想通貨連携を仕込んで試行する予定）

### (3) 今までのやり方では何ができなかったのか

前記の特徴的なアプローチがなかったら、全体の大きな構想が描けず、そもそもプロジェクトがスタートしなかった。

細部について言うと、共有技術の方式次第では、参加者が限定され、小さな規模で終わる可能性があった。そして新たな実現案（ブロックチェーン）を見つけることができず、第2回の再スタートができなかった。

また、結果の階層化による一般化がなければ、第3回以降への継続的な展開がなく、同じことの繰り返しで終了した。他地域への展開もなく、地域的にも単発で終了した。

## 2.2.7 ポイント×プロセス対応表

活動全体を「システムズエンジニアリングのプロセス」と「システムズエンジニアリングの4つのポイント」に対応付けた表を示す。

表 2.2.1 ポイント×プロセス対応表

地域活性化イベント向け情報共有基盤の開発		ポイント		その他
	目的指向と全体俯瞰	多様な専門分野を統合	抽象化・モデル化	反復による発見と進化
(6.4.1)【ビジネスあるいはミッションの分析】	<p>＜対策＞:「アニメを用いた地域活性化」という目的のもとに、地域の課題、具体的コンテンツとしてのアニメの選定、政府の支援の活用、インフラとしてのIT技術活用の方向性など、地域連携の場合の多様なニーズを抽出し、必要なステークホルダーを巻き込み、ビジネスミッションを解剖した。</p>	<p>＜対策＞: ニーズと要求の相幹として、自治体、アニメのコンテンツホルダー(出版社)、地域事業者としての交通事業者、商店街、プロ野球チーム、ITベンダーなどの多様なプレイヤーすべてが、自社(自身)のメリットを追求でき、かつ得意領域を提供できるというWin-Winの関係を構築する必要があることを明確化した。</p>		
(6.4.2)【利害関係者ニーズと要求事項の定義】				
(6.4.3)【システム要求事項の定義】	<p>＜対策＞: 事業者の入会、退会が頻繁に起こること、特定の1社にデータの管理責任を負わせることが難しいことから、データを共有の仕組みとして、データの管理責任を共通に負うことができ、データの固い込みが起きない技術が必要であることなどをシステム要求として明確化した。</p>			<p>＜対策＞: 一般化して構築したITシステムにさらに機能を追加すること、新たな価値を付加して継続する通筋を確立した。また、他地域でも再利用して同様のイベントを実現した。</p>
(6.4.4)【アーキテクチャの定義】	<p>＜対策＞: 代替案を比較し、実現方法として、ブロックチェーンの仕組みを採用した。</p>		<p>＜対策＞: 各関係者の役割を整理し、分析。データ層、ビジネス層、集客層に層別化し、一般化した。</p>	
(6.4.6)【システム解析】				
(6.4.9)【検証】				
(6.4.11)【妥当性確認】				

プロセス

2. 事例でみるシステムズエンジニアリングによる問題解決



## 2.3 電子お薬手帳システムに適用したセキュリティ設計

### 2.3.1 本事例の内容

#### (1) 概要

本事例では、従来は紙ベースであったお薬手帳を電子化し、薬歴情報等を個人のスマートフォンなどから活用できるシステムを開発した。このシステムにより、例えば遠隔地にいる家族の薬歴情報をいつでも参照できるといった患者側のメリットや、あるいは医療機関や薬局においても、他の機関等が取り扱った薬歴情報も含めて把握することにより、より精緻な診断やアドバイスに役立てるなどのメリットを提供することができるようになる。

一方で、薬歴情報など機密性の高いプライバシー情報を扱うことから、個人の端末が医療情報システムとつながる際にどこまで/どのようにセキュリティを確保するかが新たな課題として浮上した。

本事例では、新たな解決方法の実践により、課題であるセキュリティ要求の導出に成功し、システム構築を実現した。実践されたことを振り返ると、システムズエンジニアリングのアプローチと合致している点を見出すことができる。

#### (2) 本事例の特徴

新たに個人の端末が加わることに伴い、電子お薬手帳システム全体を見据えたセキュリティ要求の導出が必要となった。しかし医療情報システム関係者とセキュリティ技術者との間では、相手の専門分野の知見に乏しく、情報システムのセキュリティ要求の導出に困難が生じていた。

そこでこの事例では、

- ◆ セキュリティ要求を「事象」という形でモデル化することにより、医療情報システム関係者とセキュリティ技術者との間の意思疎通を図る。【抽象化・モデル化】
- ◆ セキュリティ要求を全体俯瞰し、「事象文言パターン」を機械的に組み合わせることによって、網羅性のある事象一式を導出する。【目的指向と全体俯瞰】

というアプローチをとることにより、セキュリティ要求を確定し、システムの構

築を実現した。【多様な専門分野を統合】



## 2.3.2 本事例の背景と課題

### (1) プロジェクトの背景

従来は紙ベースであったお薬手帳を電子化し、薬歴情報等を一元管理することにより、患者、薬局、医療機関がつながり、様々なメリットを享受できるようになる。患者にとっては、所有するスマートフォン等から、遠隔地にいる家族の分を含め、薬歴情報をいつでも確認することができるようになる。また薬局（薬剤師）や医療機関（医師）は、患者の服薬状況を、他の機関等が取り扱った薬歴情報も含めて把握することが可能となるため、的確な診断やアドバイスの提供、ひいては患者満足度の向上を図ることができる。

一方でシステムがつながる範囲が拡大することに伴い、情報の管理方法について、システム全体を俯瞰した様々な考慮が必要になった。

### (2) プロジェクトの課題

電子お薬手帳システムでは、電子お薬手帳サービス事業者が保有するデータベースに薬歴情報が集められ、インターネットを介して患者や薬剤師、医師がこれを参照する形となる。特に薬歴情報はプライバシー性が高いため、システム全体を見据えたセキュリティの確保が求められる。医療情報システムのセキュリティ確保に「医療情報システムの安全管理に関するガイドライン」が広く活用されているが、患者が所有するスマートフォン等を用いるシステムについて

ては言及されていなかったため、ガイドラインの代わりとなる適切なセキュリティ要求の導出が必要であった。

## 2.3.3 アプローチ

### (1) 対策の全体像

前述の通り、適切なセキュリティ要求の導出が大きな課題であった。しかしながら実際には、医療情報システム関係者には医療分野での知識はあっても個人の端末を加えた情報システムのセキュリティに関しては知見が乏しく、またセキュリティ技術者にはセキュリティ分野での知識はあっても医療分野の知識がなかった。網羅性のあるセキュリティ要求の導出が必要であったが、それに加え、分野を超えた共同作業を可能にするようなセキュリティ要求の導出方法の確立が必要となった。

そこで本開発では、「脅威となり得る事象」という形式を用いることで、医療情報システム関係者とセキュリティ技術者とが共同でセキュリティ要件を導出することを可能とし、さらに事象の文言を機械的に組み合わせることでセキュリティ要件の網羅性を確保した。

### (2) 個別対策の概要

システム化の対象とその周辺との関係について、紙ベースのシステムと電子化されたシステムそれぞれの整理を行った。これによって、機密性の高い情報が取り扱われる場所が、紙ベースのシステムでは薬局内に限られていたのに対し、電子お薬手帳システムでは、患者が所有するスマートフォンやネットワーク上に広がることを確認した。それぞれの存在場所に応じたセキュリティの確保が必要であることが明確になった<→対策1>。

次にセキュリティを確保するために、セキュリティ要求の導出が必要となった。セキュリティ要求とは、システムで生じ得る望ましくない事象（以下、「脅威」と言う）への対抗内容を定めた文である。セキュリティ要求の一式を導出し、それらの要求をすべて満たすシステムを開発することでシステムのセキュリティを確保することが可能となる。しかし医療分野とセキュリティ分野の知識を併せ持った人材がいないため、電子お薬手帳システム全体、特に既存

のガイドラインでは想定されていない範囲の脅威を洗い出すことが困難な状況であった。

医療情報システム関係者とセキュリティ技術者が共同でセキュリティ要求の検討を進めていく中で、例えば医療情報システム関係者に対して「望ましくない事象を洗い出してください」と言っても「無理です」と言われてしまうことがあった。ところが「もしこんな事象が起きたら問題ですか?」と尋ねると「はい、それは問題です」とか「いや別に問題ないです」といった具合に脅威であるか否かを容易に判断してもらえることができることが分かった。

そこでセキュリティ技術者が当該システムにおける脅威となり得る事象一式を洗い出し、個々の事象に対して医療情報システム関係者が脅威であるか否かを区分けすることにした。こうして「事象」という形で整理しなおすことにより、セキュリティ要求の導出を可能にした<→対策2>。

一方の事象一式の網羅性の課題については、複数の要素を機械的に組み合わせることで事象の文言を生成することで対応した<→対策3>。

### (3) アプローチの効果

セキュリティ分野の知識を持たない医療情報システム関係者と、医療分野の知識を持たないセキュリティ技術者による共同作業によって、システム設計に活かせるセキュリティ要求一式を導出することができた。また、機械的手順に基づいて導出したセキュリティ要求であるから一定の網羅性の説明が可能であり、要求漏れの不安も解消できた。

一旦セキュリティ要求一式が導出できると、これらの要求に基づいてシステムの設計や実装の善し悪しをセキュリティ技術者も判断できるようになり、医療分野の知識がなくても設計や実装にセキュリティ観点の分析、レビュー、アドバイスができるようになった。

このようにして得られたセキュリティ要求をもとに、従来の紙ベースのお薬手帳では得られなかった様々なメリットを持ち、かつセキュアな電子お薬手帳システムの開発を実現した。

## 2.3.4 対策の詳細

### <対策 1> ミッション解析

従来の紙ベースのお薬手帳システムと、その周辺との関係を図 2.3.1 に示す。

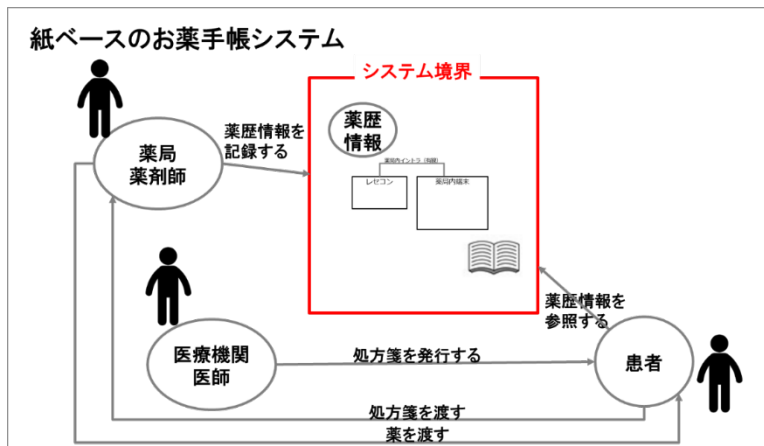


図 2.3.1 紙ベースのお薬手帳システムのコンテキスト

紙ベースのお薬手帳の場合は、医師が発行した処方箋を薬局に提示し、調剤をする際に薬局内のレセプトコンピュータ（レセコン）のシステムにデータ入力される。そのデータをお薬手帳に転記することで、患者は自分の薬歴の記録を参照することができるようになる。なお医師はこの薬歴データには直接は関与しない。

電子お薬手帳システムではシステムの対象領域が紙ベースのお薬手帳から電子的なものに変わることになる。患者はお薬手帳の冊子を持参していない場合でも、所有するスマートフォン等からいつでも薬歴情報を参照できるようになる。

開発を進める際、従来は個別の薬局内で保持されていた薬歴データを一元的に集約することで、紙ベースのお薬手帳では得られなかった様々な利便性を生むことができることが分かった。患者にとっては、手元にお薬手帳がない、遠隔地にいる家族の薬歴情報なども参照することが可能になる。薬局側では、他の薬局で調剤したものも含めた継続的な薬歴情報を把握することで、的確なア

ドバイス等が可能になる。また医師はこれまでお薬手帳の薬歴情報には直接関与していなかったが、これを参照することにより診断の精緻化に役立てることが可能になる。

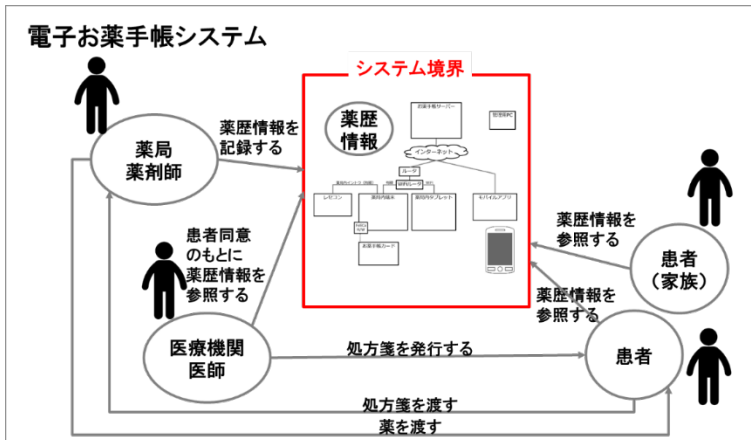


図 2.3.2 電子お薬手帳システムのコンテキスト

このようなメリットがあることから、システムアーキテクチャは一元管理された薬歴情報を想定したものになった。

一方、薬歴情報を扱う範囲が広がることに伴い、システム全体を俯瞰した様々な考慮が必要になった。薬歴情報を一元管理し、インターネットを介して共有する仕組みの場合、システムの各構成要素にどのように情報が流れ、管理するのかを整理する必要がある。図 2.3.3 は、情報が流れる（存在し得る）場所を示したものである。

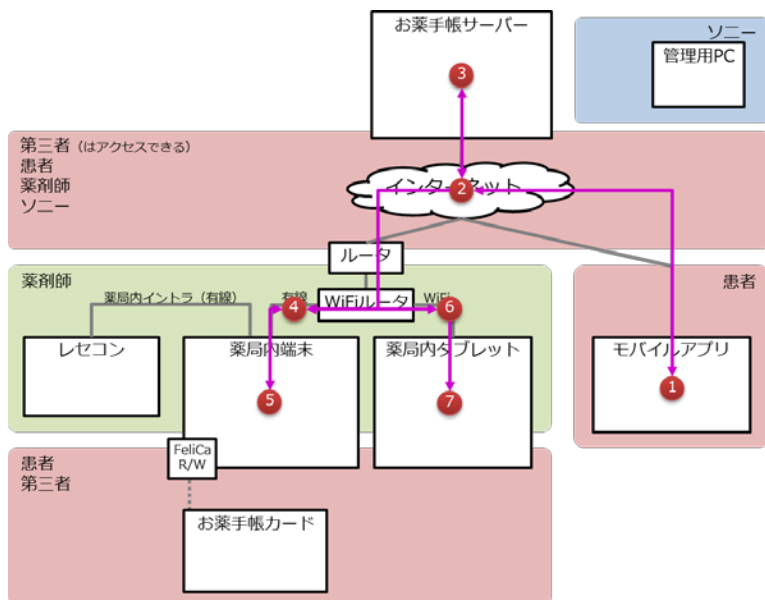


図 2.3.3 情報が存在し得る場所を洗い出したデータフロー図

## <対策 2> 分野を超えた協力によるセキュリティ要求導出

プライバシー性の高い情報を扱う範囲が広がるため、情報が存在し得る場所それぞれに応じたセキュリティの確保が必要となった。しかし前述のように、医療情報システム関係者には医療分野での知識はあっても個人の端末を加えた情報システムのセキュリティに関しては知見が乏しく、またセキュリティ技術者にはセキュリティ分野での知識はあっても医療分野の知識が乏しかった。そのためセキュリティ要求を導出できる両者の協力方法が課題であった。

そこで以下の(2-1)～(2-4)に示す方法により「事象」という形を用いて、医療情報システム関係者とセキュリティ技術者が共同でセキュリティ要求抽出作業を行った。

【システムズエンジニアリングのポイント：多様な専門分野を統合】

【関連プロセス：利害関係者ニーズと要求事項の定義】

## (2-1) 事象文言の構造

事象文言生成法が実効性を持つためには、事象文言一式の分量を区分け作業できる規模に抑え込むこと、個々の事象文言が脅威（望ましくない事象）であるかを判断できる意味（情報）を持たせることの両立が必要であった。

セキュリティ技術者の経験から、脅威となる事象文言の表現には次のような傾向があることが分かっていた。ひとつ目の傾向は、攻撃から守るべき対象（以下、「資産」と言う）には「情報」と「機能」の2種類があること。ふたつ目の傾向は、脅威文言を単純化するとその骨格には情報を「読まれた」、情報を「書かれた」、機能を「使われた」というように、資産に対する「操作」が表現されていること。3つ目の傾向は、ある資産に対するある操作が実施されたとき、それを実施した人物によってその操作は不正利用（脅威）であるか正当利用であるか判断が異なること。以上の3つの傾向から「人物」「資産」「操作」の3つの要素の組み合わせで、図 2.3.4 のように脅威の候補となる事象文言を機械的に生成することにした。

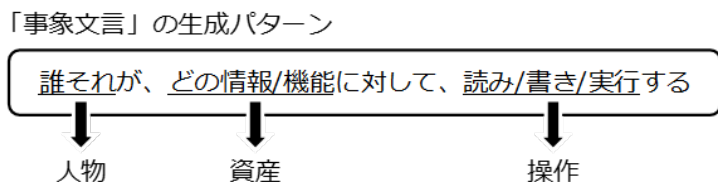


図 2.3.4 「事象文言の生成パターン」

## (2-2) 人物リストの作成

対象システムにアクセスし得る人物リストを作成する。対象システムの設計文書に記載されている人物リストに「第三者」を加えるだけである。なおここで挙げた人物は被害者になる他、攻撃者になることもあることに注意すること。



表 2.3.1 お薬手帳システムにおける人物リスト

人物	人物の説明
患者	当システムを利用している患者
薬剤師	当システムを導入した薬局の薬剤師
システム運用者	当システムの運用にあたる特別な権限を有する特定少数
第三者	上記のいずれでもない人物

### (2-3) 資産リストの作成

対象システムの資産リスト（攻撃から守るべき対象のリスト）を作成する。資産の種類には「情報」と「機能」があるので、設計文書を以下の手順でレビューすることで資産を抽出し、リストアップすることができる。

- ① 「情報」または「機能」に類する単語を見つける
- ② その単語が「勝手に使われたら困りそうか？」を考える
- ③ YES であれば、その単語を資産リストに追加する

### (2-4) 資産リストのグループ化

資産リストの項目数は大きくなる傾向がある。脅威文言リストの生成で組み合わせ爆発のリスクがあるので、資産をグループ化して項目数を小さくする。例えば患者の「氏名」「住所」「電話番号」といった資産をひとつの「患者の個人情報」としてまとめ上げる。

患者の個人情報

└患者の氏名

└患者の住所

└患者の電話番号

グループにまとめる資産の選び方に少々課題がある。グループ化すべき資産の選び方を端的に言うと、後工程で出来上がる「脅威文言の脅威か否かの判断

結果が同じになる資産をグループ化する」ことになる。しかしこれは鶏卵問題であるため、まずは仮にグループ化して後で見直し調整することになる。

以上のように、「人物」「資産」「操作」の項目数を小さく抑える工夫も重ねることで、事象文言一式の規模を抑え込むことと、事象文言に脅威判断ができる意味（情報）を持たせることの両立も実現した。

【システムズエンジニアリングのポイント：抽象化・モデル化】

## <対策 3> 網羅性のある事象文言の生成

以下の(3-1)～(3-5)に示す方法によりセキュリティ要件の網羅性の課題に対応し、最終的なセキュリティ要件の確定を行った。

【システムズエンジニアリングのポイント：目的指向と全体俯瞰】

### (3-1) 脅威文言の候補リストの生成

表 2.3.2 の〔攻撃者〕〔被害者〕には人物リストから選んだ単語を、〔情報〕〔機能〕には資産リストから選んだ単語を割り当てることで、脅威文言を機械的に生成する。機械的な組み合わせにより漏れなくリスト化する。脅威文言は着目している資産の種別に応じて選ぶ。

表 2.3.2 脅威文言の生成パターン

種別	脅威文言の生成パターン
情報	〔攻撃者〕が〔情報〕を READ する 〔攻撃者〕が〔情報〕を WRITE する 〔被害者〕が〔情報〕を READ することが妨害される 〔被害者〕が〔情報〕を WRITE することが妨害される
機能	〔攻撃者〕が〔機能〕を EXECUTE する 〔被害者〕が〔機能〕を EXECUTE することが妨害される

この段階では医療情報システム関係者による脅威か否かの判断がされていない段階であるので、脅威文言の「候補」リストであることに注意すること。

### (3-2) 医療情報システム関係者による脅威判定

脅威文言の「候補」リストからひとつずつ脅威文言を取り出し、医療情報システム関係者に脅威であるか否かを判定してもらう。脅威でない脅威文言をリストからすべて削除し、その結果残ったリストが脅威文言リストとなる。

### (3-3) 脅威文言の調整

医療情報システム関係者による脅威判定の中で、少なからず医療情報システム関係者から脅威文言を調整する提案がある。例えば「薬剤師が〇〇情報を READ する」に脅威文言について、「患者からお薬手帳カードが提示されたときなら問題ないですが、提示がないのに薬剤師が〇〇情報を READ してしまうと問題なのです。」と医療情報システム関係者が教えてくれる等である。これを受けて、「患者がお薬手帳カードを提示していないとき、薬剤師が〇〇情報を READ する」といった脅威文言の調整をする。

### (3-4) セキュリティ要求の定義

以上の手順により脅威文言リストができた。脅威文言は対象システムにおける望ましくない事象を記述したものであるため、これに対してシステムがどのようなべきかを要求形式で記述したものがセキュリティ要求である。

セキュリティ要求の最も単純な作り方は、脅威文言に対して「できない」と語尾を変えることである。例えば脅威文言「第三者が〇〇情報を READ する」に対して「第三者は〇〇情報を READ できない（こと）」といった表現でセキュリティ要求を記述する。

### (3-5) 具体的成果物の説明と例示

表 2.3.3 は作成したセキュリティ要求一式の例である。個々のセキュリティ要求を満たすようにシステムを設計、実装することでシステムのセキュリティを確保する。

表 2.3.3 セキュリティ要件例

資産	セキュリティ要件
○○情報 └ ○○X情報 └ ○○Y情報 └ ○○Z情報	[第三者]は[○○情報]を[READ]できない [第三者]は[○○情報]を[WRITE]できない [患者]は[他人の○○情報]を[READ]できない [患者]は[他人の○○情報]を[WRITE]できない [薬剤師]は[お薬手帳カードが提供されないとき][○○情報]を[READ]できない [薬剤師]は[○○情報]を[WRITE]できない
△△情報	[第三者]は[△△情報]を[READ]できない [第三者]は[△△情報]を[WRITE]できない [患者]は[他人の△△情報]を[READ]できない [患者]は[他人の△△情報]を[WRITE]できない [薬剤師]は[非自局患者の△△情報]を[READ]できない [薬剤師]は[非自局患者の△△情報]を[WRITE]できない
□□情報 └ □□X情報 └ □□Y情報 └ □□Z情報	[第三者]は[□□情報]を[READ]できない [第三者]は[□□情報]を[WRITE]できない [患者]は[自分の□□情報]を[WRITE]できない [患者]は[他人の□□情報]を[READ]できない [患者]は[他人の□□情報]を[WRITE]できない [薬剤師]は[他局発行の自局患者の□□情報]を[WRITE]できない [薬剤師]は[他局発行の非自局患者の□□情報]を[READ]できない [薬剤師]は[他局発行の非自局患者の□□情報]を[WRITE]できない

## 2.3.5 対策の効果のまとめ

システムズエンジニアリングのアプローチに合致した方法で上記の各対策を実践することにより、異なる専門分野の知見をもとに、適切なセキュリティ要件を導出することができた。また要求の抽象化とグループ分けの工夫により要求項目数を小さくし、網羅性を担保した上での工数削減を実現した。

## 2.3.6 教訓

### (1) 出来上がったシステムの特徴

薬歴情報を一元管理し、インターネットを介して情報共有する電子お薬手帳システムにより、利用者は従来の紙ベースのお薬手帳では実現できなかった様々なメリットを享受できるようになった。

その反面、情報を取り扱う範囲が拡大するため、システムの実現においては、従来のガイドラインでは整備されていない部分までもカバーしたセキュリティの確保が求められた。本システムはセキュリティ要求の漏れのない電子お薬手帳システムとなっている。

## (2) 結果を導いたアプローチの特徴

システムの対象範囲が広がることにより、異なる専門分野の技術の統合が必要となった。具体的にはセキュリティ分野の知識を持たない医療情報システム関係者と、医療分野の知識を持たないセキュリティ技術者との共同作業が必要となった。本システム開発では、セキュリティ要求を「事象文言」という形に抽象化・モデル化するという工夫により、共同作業を可能にした。

またセキュリティ要求を全体俯瞰し、「事象文言パターン」を機械的に組み合わせるという方法を用いることで網羅性を担保し、要求漏れの不安も解消できた。

## (3) 今までのやり方では何ができなかったのか

今までのやり方と同様に、医療情報システム関係者のみで開発を進めた場合には、セキュリティ要求の抽出漏れ、それに起因する設計や実装の不備により、セキュリティ面に脆弱な部分の残るシステムになっていた可能性がある。

そのような状態を防止するために、セキュリティ技術者が参画して共同作業を行うことになった場合においても、抽象化・モデル化や全体俯瞰の視点によるセキュリティ要求導出の新たな工夫を実施していなければ、依然としてセキュリティ要求漏れの不安が残るシステムになっていたであろう。

異なる専門分野を統合する際には、双方が理解できるようにするための工夫が不可欠である。

## 2.3.7 ポイント×プロセス対応表

活動全体を「システムズエンジニアリングのプロセス」と「システムズエンジニアリングの4つのポイント」に対応付けた表を示す。

表 2.34 ポイント×プロセス対応表

電子お薬手帳システムに適用したセキュリティ設計 ポイント		多様な専門分野を統合	抽象化・モデル化	反復による発見と進化	その他
(6.4.1)【ビジネスあり】 【不正アクセス防止 のミッションの 分析】	目的指向と全体俯瞰 ・悪意情報等の不正アクセスを防止 する、セキュアな電子お薬手帳シ ステム。				
(6.4.2)【利害関係 者ニーズと要求事 項の定義】	・網羅的にセキュリティ要求の一式 を導出。網羅性を担保するため、複 数の要素を機械的に組み合わせる ことで事象文言を生成。	・医療分野、情報セキュリティ分野 の有識者による共同作業により、セ キュリティ要求を導出。 ・まずセキュリティ技術者が脅威に なり得る事象を洗い出し、個々の事 象に対して医療情報システム関係 者が判断するやり方(脅威モデリン グ)で実施。	・脅威にのみなり得る事象を単純化す ることにより、セキュリティ要件に対 する共通理解を促進。		
プ ロ セ ス	(6.4.3)【システム 要求事項の定義】				
	(6.4.4)【アークテク チャの定義】				
	(6.4.6)【システム 解析】				
	(6.4.9)【検証】				
	(6.4.11)【妥当性確 認】				

2. 事例でみるシステムズエンジニアリングに  
よる問題解決

## 2.4 多様な要求を満足させる自動車エンジンの開発 ～システムズエンジニアリング視点からの SKYACTIV-G～

### 2.4.1 本事例の内容

#### (1) 概要

本事例は、自動車業界において画期的と言われるマツダの開発革新の中でも、中核的な内燃機関の燃焼効率改善を取り上げる。

この開発においては、部品の設計に先行して、目的指向・全体俯瞰で抽象化した機能レベルでの定義、分解を行い、細分化された機能を実現するハードウェアやソフトウェアを全体最適で設計した。この結果、統合（インテグレーション）、検証のトラブルは減少し、開発効率が向上し、市場での品質も劇的に改善された。

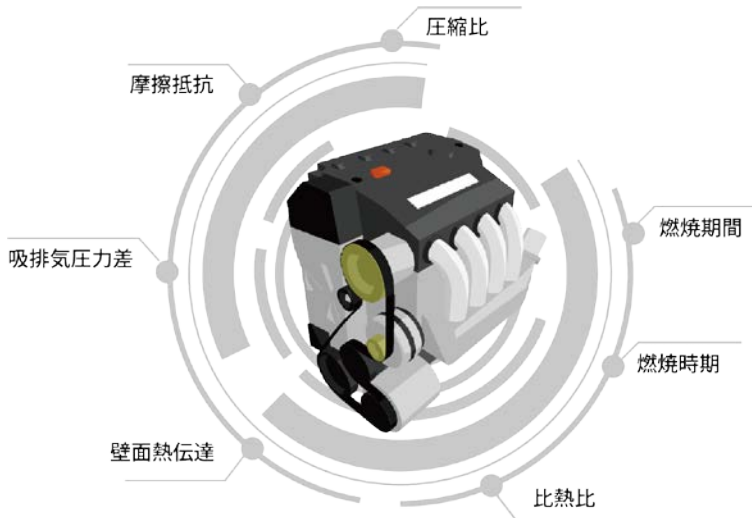
機能的には、量産ガソリンエンジンとして世界で初めて高圧縮比(14.0)を実現し、エンジンの効率を大幅に高め、従来比で燃費・トルクをともに15%向上させることができた。

また排気量ごとの開発について、従来は個々に最適化を実施していたため全体的な開発効率が高くなかったので、本質的な特性を揃えることにより、同一の設計を1.3L/1.5L/2.0L/2.5Lの4つの排気量に展開できるようにした。

#### (2) 本事例の特徴

すでに限界とみられていた内燃機関の効率改善に取り組み、個別最適の延長ではなく、本質的な課題の把握から取り組み、大幅な効率改善を実現した。

【目的指向と全体俯瞰】 高度な機能を実現するために複雑化、巨大化したシステムを開発しなければならない状況で、多数の制御因子の組み合わせを最適化する場合に効果的なアプローチを示している。【多様な専門分野を統合】【抽象化・モデル化】 さらに、ベテランに依存していた開発から、徹底したモデルベース開発によって、従来の開発方法では実現困難な出力・燃費性能を達成した。他の領域においても常識を越えたブレークスルーが生み出される可能性を示唆している。



全体最適による高効率エンジンの開発

2. 事例でみるシステムズエンジニアリングによる問題解決

## 2.4.2 本事例の背景と課題

### (1) ビジネスの背景

すべての顧客に「走る喜び」と「優れた環境・安全性能」を提供するという目的のため、「2020年までにマツダ車の全社平均燃費を50%向上」（2008年比）、そのために「段階的に電動化技術を取り入れるビルディングブロック戦略を推進する」という技術開発の長期ビジョンを設定した。

技術の選択にあたっては、将来電動化が確実に進むものの、内燃機関をベースとした動力源と予測されることから(図2.4.1)、内燃機関を磨くこととした。

また、多くの車種展開をしている中、優れた技術をすべての車種に展開できなければならない。そのために、一括して企画し、コモンアーキテクチャのもとに開発する必要がある。従来は車種ごと、機能ごとに最適構造を設計し、他の車種や車格に展開する開発を行っていたが、各車種の開発タイミングで個々に設計するのではなく、今後5～10年先の商品を計画して、全商品を最初に一



括して企画した。つまりエンジン、変速機、ボディ、シャシー等でひとつの開発／生産コンセプトを全車種で共通化し、ラインアップ全体でスケールメリットを追求した。そのため 1.3L から 2.5L まで、すべてのサイズのガソリンエンジンで同じ体質の開発が必要となった。

## 自動車年間販売台数予測

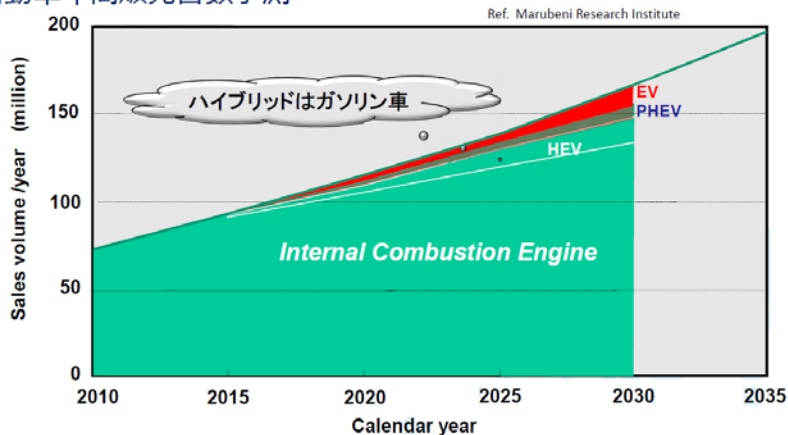


図 2.4.1 内燃機関の利用動向予測

## (2) ビジネスの課題

内燃機関の効率改善とは、内燃機関が仕事をする上での損失を減らすことである。

内燃機関は、燃料（ガソリンや軽油）と空気が入って、排ガスとパワーが出てくるものであり、損失は排気の損失／冷却による損失／機械抵抗、ポンプ損失といった損失に分解することができるが、それぞれ改善の余地が残っている。また、エンジン開発は数百人が細かな担当に別れ、縦割りで部分最適に邁進しているため全体最適が図りづらい状況でもある。

これらの損失に影響している因子を最適にすることが設計になる（課題となる）。そのためまずこれらの損失を制御できる機能の特性（制御因子）を明確にする必要があった。

## 2.4.3 アプローチ

### (1) 対策の全体像

今回取り組んだ対策の概要を示す。

内燃機関を構成するセンサー、部品、アクチュエータ、ソフトウェア等の個別部品を個別最適に設計するのではなく、これらが統合されるシステムと捉えた。そして、内燃機関全体に課された要求から機能目標を定義した。トレードオフスタディを行いながら内燃機関の機能を分解し、分解した機能を実現する個別部品を設計するにあたり、全体最適の観点から、機能目標を達成するべく設計を進めた。

従来であれば、それぞれの部品の性能や品質を高めるように設計し、そのように最適化された部品を統合して、内燃機関を組み立てていった。SKYACTIV-Gでは、目的を達成するために、上位の機能からどうあるべきかを分析し、そのような機能を実現するために、各構成要素がどうあるべきかを導出して目標を割り当てていった。

内燃機関は多くの機構・部品から構成されている。これまでの研究・開発成果を集約し改善を重ねてきたシステムであるが、エネルギー効率は3割程度であり、7割は損失として失われていた。

従来のそれぞれ分割された個別最適の部品の寄せ集めの改善アプローチでは、費用や期間のわりに、効果が限られていた。そこで、そもそも損失とは何か、その損失を低減させるためにはどうすれば良いかを改めて整理することから始めた。

## (2) 個別対策の概要

### <対策1> コンテキストでの要求分析

制御ソフトウェアに関するコンテキストである内燃機関の効率改善という目的を、損失という視点で分解する。排気損失/冷却損失/機械抵抗損失/ポンプ損失など、損失の構成を分析した。

### <対策2> 機能の特性を定義

各損失をさらに細分化し、制御できる機能の特性（制御因子）を定義し、各機能の目標を設定した。例えば「排気損失」を低減させるためには、大きく圧縮し大きく膨張させ、圧縮比を大きくすることで理論的熱効率を向上させられる。

### <対策3> 反復による発見と進化

機能間のトレードスタディを行い、機能の分割を繰り返した。

例えば、圧縮比を高めると、ノッキングと呼ばれ異常振動が現れる。これを防ぐには高温の排気ガスの逆流を防ぎ、混合気生成、火炎形成を理想化する必要がある、排気システム、燃料噴射制御、ピストンが影響する。

＜対策4＞機能を実現するための物理設計

センサー、部品、アクチュエータ、ソフトウェア等の専門家が全体最適となるように設計した。

上位の目的・機能から綿密に紐付けされた要求条件に従い、全体として無駄や間違いを排除し、設計品質を向上させることを目指した。

### (3) アプローチの効果

上記の対策の結果、気にしていた「ノッキング」は発生せず、内部の機能が増えることでソフトウェアの開発規模は大きくなったが、規模あたりでの不具合は減少した。また、開発のシステム設計上流、実装、システム検証と分けると、実装での不具合が5分の1となり、下流の検証での問題も減少した。

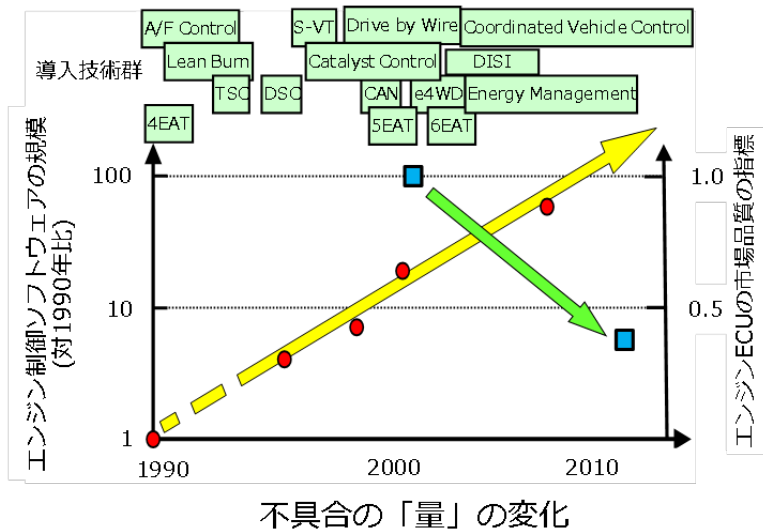


図 2.4.2 不具合の減少

図 2.4.2 に示すように、従来の開発に比べて、ソフトウェアの規模が巨大化したにもかかわらず、市場での品質問題は大幅に低減した。図 2.4.3 のように、開発工程内での不具合は、上流に大きくシフトし、下流における検証での問題が削減された。

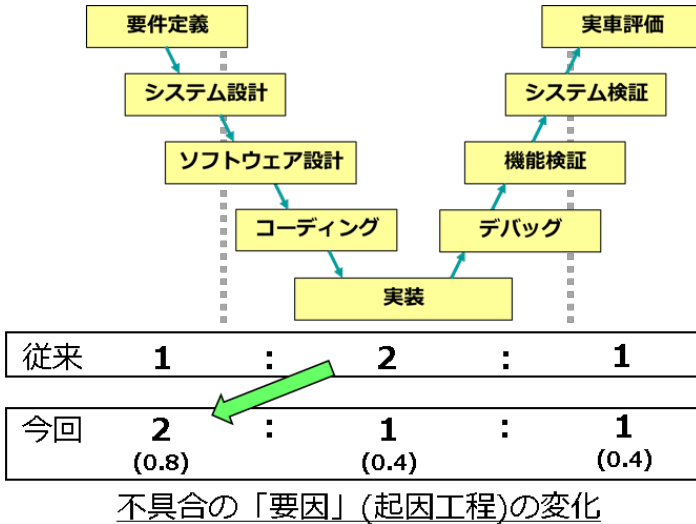


図 2.4.3 不具合の早期発見

## 2.4.4 対策の詳細

### <対策 1> コンテキストでの要求分解

内燃機関は単純化すると図 2.4.4 にあるような構造となっている。すべての損失が 0 であればエネルギーを 100% 有効な仕事に変換することができるが、実際には各部において各種の損失が発生する。内燃機関のエネルギー効率はそれらの損失の和が減殺されたものとなって表れ、従来は 3 割程度であった。(すなわち損失が 7 割)

内燃機関の効率改善とは、この損失を減らすことに他ならない。すなわち排気損失/冷却損失/機械抵抗損失/ポンプ損失の低減が内燃機関の効率改善のための要求となる。内燃機関全体の熱効率の改善に向けては、排気損失/冷却損失/機械抵抗損失/ポンプ損失それぞれについて、改善の余地があった。

各損失の構成は、制御ソフトウェアの設計を検討する際のコンテキストと考えられる。要求分析に続いて、システム設計を行う際には、各損失の特性や関係をモデル化し、目的を明確に理解しながら進めた。

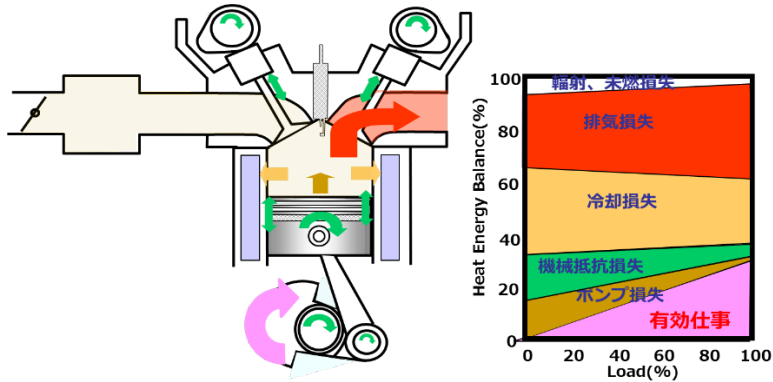


図 2.4.4 内燃機関の損失の構成

例えば、図 2.4.5 において、排気損失の1次制御因子は、圧縮比、燃焼期間、燃焼時期、燃焼時期、比熱比、壁面熱伝導に支配される。そして、燃焼期間は、火炎伝播速度、火炎伝播距離、火炎伝播空間に左右される。このような制御因子の関係をひとつひとつ分析し、モデルを形成した。

【システムズエンジニアリングのポイント：目的指向と全体俯瞰、抽象化・モデル化】

【関連プロセス：システム要求事項の定義】

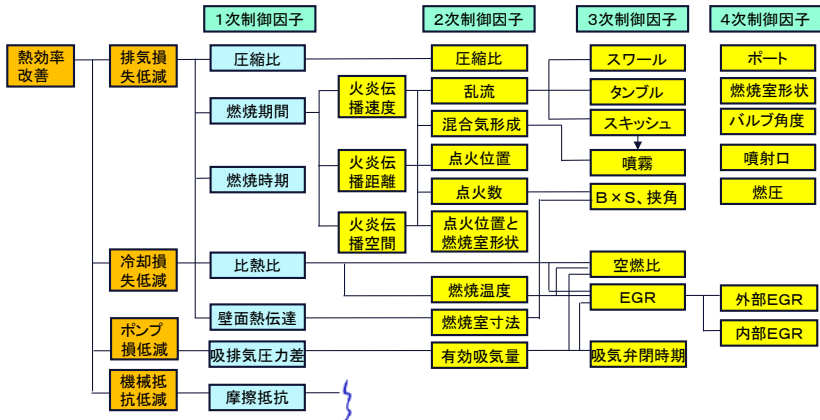


図 2.4.5 損失に影響する因子

## <対策 2> 機能の特性を定義

各損失をさらに細分化し、制御できる機能の特性（制御因子）を定義し、各機能の目標を設定した。各制御因子を最適化することが設計になる。どの制御

因子がどの損失に影響するのか、その関係を明確化し、各分野を担当する技術者間で共有した。様々な領域の技術者が課題を理解し易いように、制御因子間の関係を適切に抽象化、モデル化した。

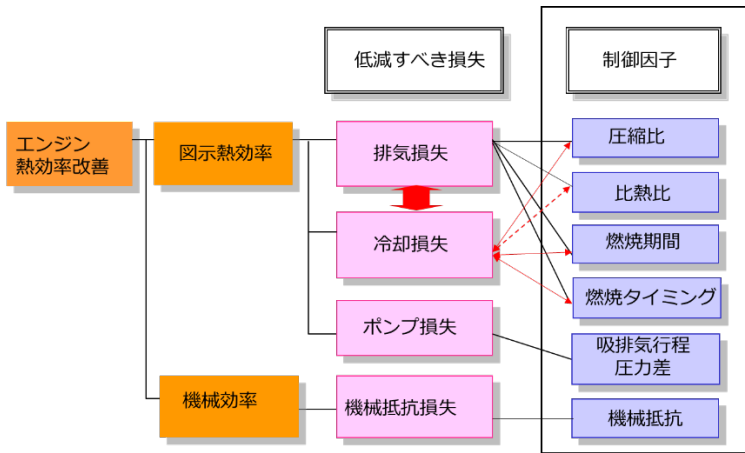


図 2.4.6 機能レベルの制御因子

図 2.4.6 で示す制御因子は、ストローク、点火タイミング、バブルタイミングといった物理的特性値ではなく、抽象度が高い機能レベルの特性値になっている。物理構成の特性値設計の前に、各損失を制御できる機能の特性を制御因子として定義した。図 2.4.6 に示す圧縮比、比熱比、燃焼期間、燃焼タイミングなどが、機能レベルの制御因子である。このようにして、設計における目的指向を徹底した。

【システムズエンジニアリングのポイント：抽象化・モデル化、目的指向と全体俯瞰】

【関連プロセス：アーキテクチャの定義、システム解析】

### <対策 3> 反復による発見と進化

機能間には依存関係があり、ある機能を最適化しようとすると、別の部分に不具合が生じることがある。したがって、トレードスタディを行い、機能の分割を繰り返す必要があった。

例えば「排気損失」を低減させるためには、圧縮比を高め、大きく圧縮し、大きく膨張させることで理論的熱効率を向上させることができる。しかし、圧縮比を高めようとすると、様々な問題が発生する。ピストンの適正位置で点火

する必要があるが、排気ガスの逆流が起こるとシリンダ内の温度が高くなり、点火前に着火してしまう。これはノッキング（異常燃焼）と呼ばれ、異常振動等の不具合となって現れる。したがって、排気ガスの逆流を防ぎ、高温の残留ガスの排気を促し、温度を下げる必要がある。ここまでの関係を整理すると図 2.4.7 のようになる。

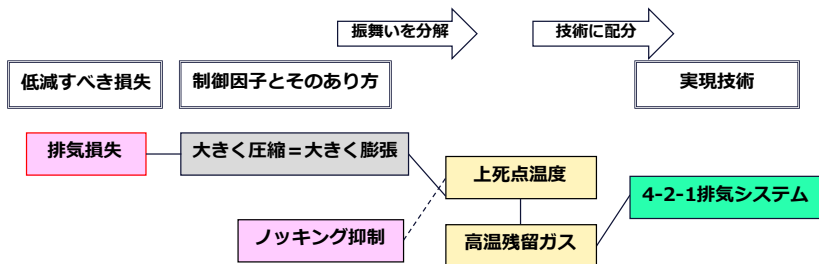


図 2.4.7 高圧縮とノッキングの関係

このような検討の結果、排気システムには、気筒間の排気経路が長いものを採用した。これによって、残留ガスの影響を抑えて、高圧縮比を可能にした。

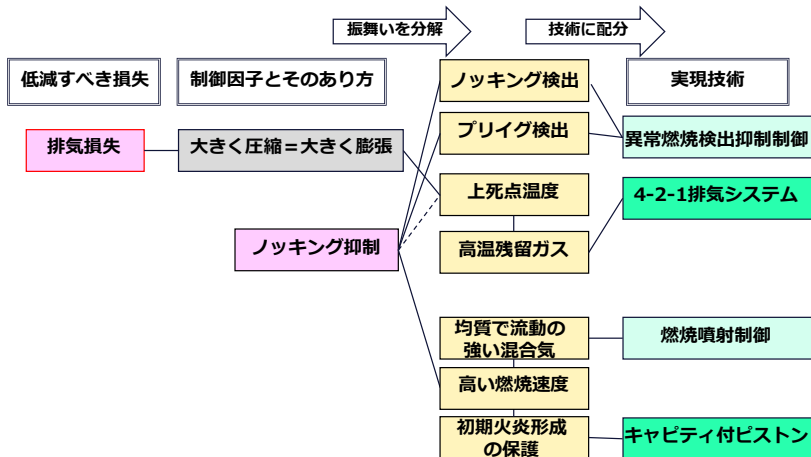


図 2.4.8 ノッキング抑制の対策

さらに、図 2.4.8 のように、その他の要因で発生するノッキングの抑制も検討した。ノッキングや着火を検出する異常燃焼検出抑制制御と、高い燃焼速度を得るために均質で流動の強い混合気を生成する燃料噴射制御と初期の火炎形成を保護するキャビティ付ピストンを採用することにした。

一方、排気経路の長い4-2-1排気システムを採用したことによる問題もあった。触媒までの距離が長いと、触媒に到達する排気ガスの温度が低下し、始動直後は触媒が活性化できなかった。点火時期を遅らせ、排気ガス温度を上昇させることで、触媒の早期活性化を促す方法があるが、点火時期を遅らせ過ぎると、燃焼が不安定になり、触媒の後方配置には限界があった。

SKYACTIV-Gでは、マルチホールインジェクターにより燃料噴射を最適化し、プラグ周りに成層混合気生成されるようにした。これにより、点火時期を大幅に遅らせても燃焼が不安定にならず、4-2-1排気システムでの触媒後方配置を実現できた。

このように、各機能の依存関係を理解し、実験やシミュレーションによる解析を行い、各領域の技術者が協力しながら各部の調整を繰り返して設計を進めた。

【システムズエンジニアリングのポイント：反復による発見と進化、抽象化・モデル化、多様な専門分野を統合】

【関連プロセス：システム解析、アーキテクチャの定義】

## <対策4>機能を実現するための物理設計

機能を実現する技術を明確にしたところで、機能をシステム上に物理的に配置した。その部分的な概要を図2.4.9に示す。車両内のレイアウトも踏まえ、空間的な制約も考慮して配置した。センサー、部品、アクチュエータ、ソフトウェア等の専門家が全体最適となるように設計した。従来の各担当にまかされていた個別最適と異なり、最上位の目的・機能から綿密に紐付けされているため、全体として無駄や間違いを排除し易くなり、設計品質を向上させることができた。

制御システムのソフトウェアは、これまでも追加、修正を重ねてきたため、中身が複雑になっており、手を入れるのが難しくなっていた。

SKYACTIV-Gでは、機能の制御因子を明確化し、その特性値を設計し、そのために制御システムのソフトウェアで実現することを明確にしていた。その論理的な構造に従って、ソフトウェアの構造化を行った。機能構造に対応した構造とすることで、検証、統合（インテグレーション）を容易にした。図2.4.10にアプリケーション層の論理的構造を示す。



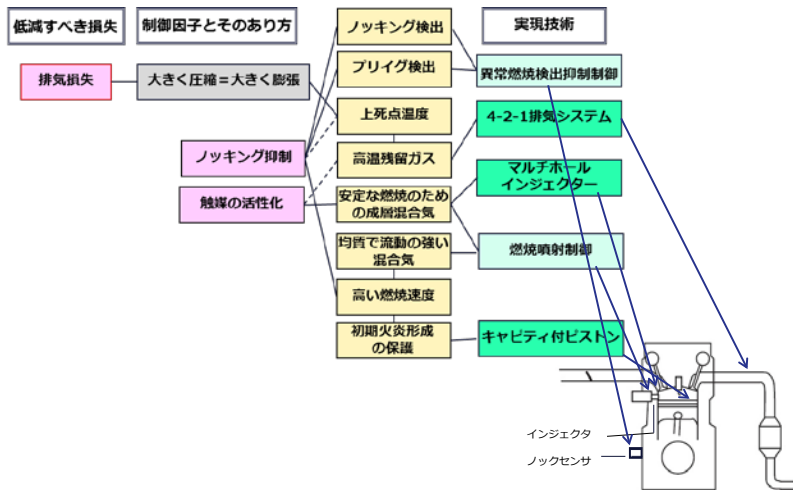


図 2.4.9 機能を実現する技術の物理的配置

図 2.4.10 において、車両の挙動を制御する層（灰色部分）と、エンジン挙動を制御する層（白色部分）とに階層化した。その間の情報伝達は、赤矢印と青矢印で示す「トルク」に集約した。エンジン制御内（白色部分）では、吸排気/燃料/点火に機能を分解し、それらを協調させる構造とした。

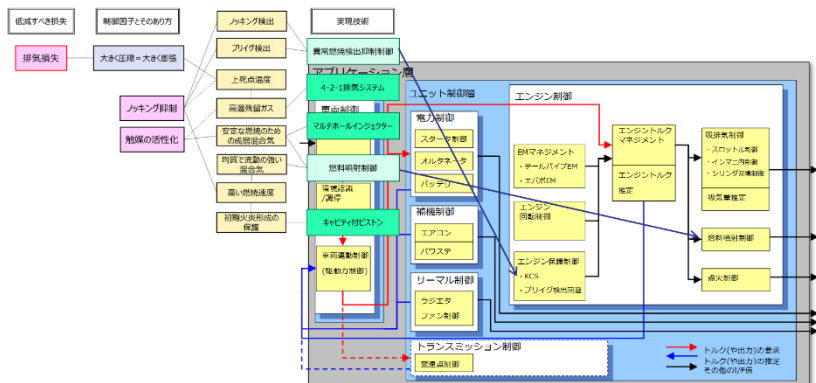


図 2.4.10 制御システムのソフトウェアの構造

一方、開発効率を高めるため、排気量に依らず同一な設計を採用することにしてきた。これについても、理想を実現するため、反復して設計したものを組み上げていった。まずは、エンジンの理想の振る舞いを決め、その振る舞いに

なるように3D-CAE技術を利用して、ハードウェアを設計し、制御ソフトウェアを開発した。

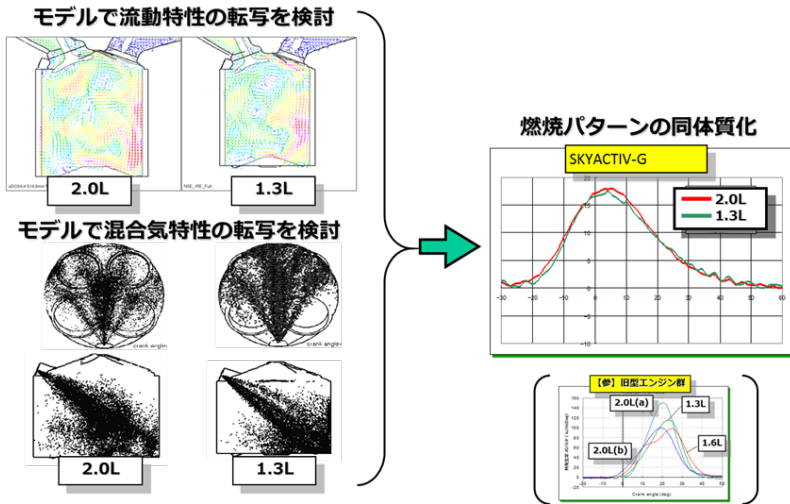


図 2.4.11 エンジンサイズに依らない燃焼パターンの同体質化

図 2.4.11 に示す通り、シリンダ内の流動特性や混合気特性をモデルで検討し、排気量に依らず、一括で理想の燃焼になるように開発し、狙った燃費性能、出力性能を実現した。実際の燃焼実験とシミュレーターによる分析を組み合わせ、多数の設定で繰り返し検証できたことが成功の鍵となった。

【システムズエンジニアリングのポイント：多様な専門分野を統合、目的指向と全体俯瞰、抽象化・モデル化、反復による発見と進化】

【関連プロセス：検証、妥当性確認、システム解析】

## 2.4.5 対策の効果のまとめ

要求～機能～物理構造が段階的に分解され、統合（インテグレーション）の結果を段階的に検証できた。図 2.4.12 に示す通り、実際のエンジンや制御ユニット (ECU) ができる前に、シミュレーターを用いて、モデルやソフトウェアを検証していた。まずはソフトウェアを検証し、制御ユニットができるとソフトウェアとハードウェアを統合して、電気信号レベルでの検証を行い、HILS を用いて、機能安全も含めた制御ユニットの機能確認を行った (図 2.4.12 の①)。制御ユニットの検証ができていますので、実制御対象 (エンジン) とつないだシス

テム評価時のトラブルも減少した（図 2.4.12 の②）。結果的に、試作 1 号機一発着火という当事者としては快挙を実現できた。

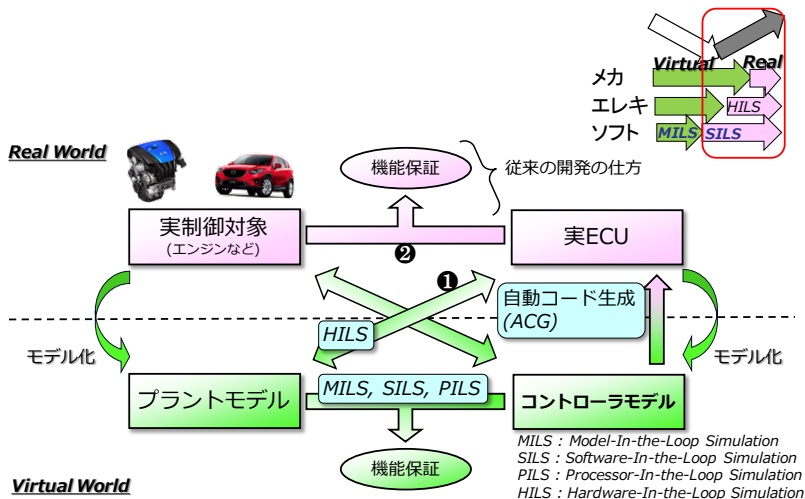


図 2.4.12 検証の効率化

## 2.4.6 教訓

### (1) 出来上がったシステムの特徴

個々の部品の物理設計に先行して機能開発することで、設計から検証を効率的に実施できた。従来であれば、システムとしての性能向上に向けて、部品ごとに個別に最適化を目指していたが、部品間の依存関係、影響に目を向けることになった。目指す機能に影響する要因の分析、モデル化に大きく注力し、機能の特性値を決めていった。結果として燃費・トルクを大幅に向上することができた。

### (2) 結果を導いたアプローチの特徴

システム全体の最適化を目指して、目的を明確化し、関係者間で共有を図った。個別部分の最適化ではなく、システムの目的に照らして、各部分の機能、特性を決めた。部品の物理設計の前に、一段抽象度の高い機能レベルで特性を明確化することにより、依存関係のある機能や部品の担当間での連携も容易になった。さらに、要求—機能—物理と段階的に詳細化し、その構造に従って統合を進めたので、検証も容易かつ確実に実施できた。

### (3) 今までのやり方では何ができなかったのか

SKYACTIV-Gの開発は、システムズエンジニアリングがなければ実現できなかった。上記のような目的の明確化や全体最適、構造的な設計と検証などに加え、内燃機関の効率を支配する損失に対する詳細な分析と、損失に影響する制御因子を解明することが重要であった。そして、それらの制御因子の振る舞いを検討した上で、実現技術、機能配置を決定し、ソフトウェアの構造も固めていった。

機能の実現においては、MILS、HILSなどシミュレーション技術を活用した検証、チューニングを多用しており、3D-CAE技術などコンピュータパワーとソフトウェアの蓄積がなければ、目標性能を実現することはできなかった。プロトタイプや実機を用いた実験や分析には、費用も時間もかかり、評価できるパターンの数が限られてしまう。コンピュータ上でのシミュレーション、解析を行うことにより、従来では不可能なレベルの最適化を達成できた。内燃機関の各部の振る舞いを詳細に分析し、多くの数理モデルを長年にわたり構築し、蓄積していたことにより、高度なシミュレーションやCAEを活用できた。内燃機関というシステムの振る舞いを構成する各要素とそれらの相互関係を階層的、構造的に分析し、理解するという考え方が基礎になっていた。

## 2.4.7 ポイント×プロセス対応表

活動全体を「システムズエンジニアリングのプロセス」と「システムズエンジニアリングの4つのポイント」に対応付けた表を示す。

表 2.4.1 ポイント×プロセス対応表

	ポイント				その他
	目的指向と全体俯瞰	多様な専門分野を統合	抽象化・モデル化	反復による発見と進化	
(6.4.1)【ビジネスあるいはミッションの分析】	<対策1>コンテキストでの要求分析(損失の構成を分析)		<対策1>コンテキストでの要求分析(損失の特性や関係をモデル化)		
(6.4.2)【利害関係者ニーズと要求事項の定義】			<対策2>機能の特性を定義(機能レベルの制御因子を定義し、機能目標を設定)		
(6.4.3)【システム要求事項の定義】				<対策3>反復による発見と進化(トレードオフ解析で機能を進化)	
(6.4.4)【アーキテクチャの定義】		<対策4>機能を実現するための物理設計(多様な専門分野で全体最適)			
(6.4.6)【システム解析】	目的を指向した評価	電気系、機械系等すべてを機能モデルで結合	動く仕様書となる解析モデルとMILS	<対策3>負の影響を発見	
(6.4.9)【検証】	排気量に寄らない理想の燃焼を開発	HILS、実機ベンチで評価ポイントを決めて検証	HILSで制御ユニットの機能検証		
(6.4.11)【妥当性確認】					

多様な要求を満足させる自動車エンジンの開発

## 2.5 首都圏の高密度輸送を支えるデジタルATC 開発

### 2.5.1 本事例の内容

#### (1) 概要

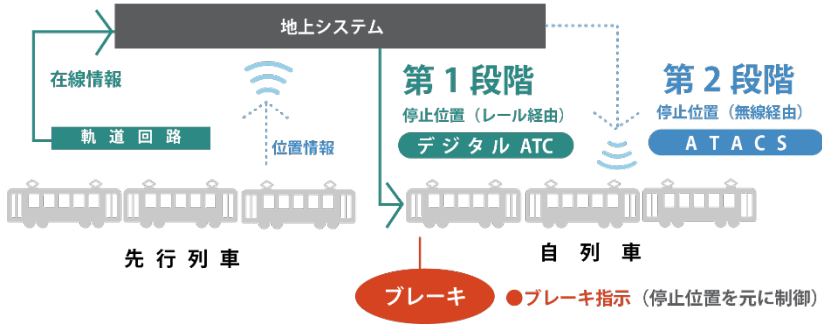
本事例は、数十年にわたり改新を続けていく鉄道システムにあって、2世代後の方式も見据えつつ、次世代ATC（Automatic Train Control）の方式設計に取り組んだ開発アプローチを取り上げる。その際、一度に方式を全面的に更新するのではなく、アナログATC、デジタルATC、ATACSと段階的に機能を更新することとした。加えて、毎日行われている運行に支障をきたさず、旧システムから新システムへ迅速な切り換え（この場合はアナログATCからデジタルATC）を行えるように対処した事例である。

#### (2) 本事例の特徴

首都圏線区の混雑率緩和のため列車運転間隔短縮を図ることとし、将来を見据えた方式としてATCの制御方式を、信号区間内の許容速度を指示する方式から、先行列車の位置と自列車の位置からその間隔を計算し、速度制御を行う方式に変更することを定めた【目的指向と全体俯瞰】。先行列車位置検出は、山手・京浜東北線で無線列車制御システムを採用するには技術的課題が多いため、第1段階（デジタルATC）として従来通りレール経由の信号によるものとし、第2段階（ATACS）で無線による方式を採用することとした。列車の自位置認識方式は、当初から、その両方を想定したものとした【反復による発見と進化】。

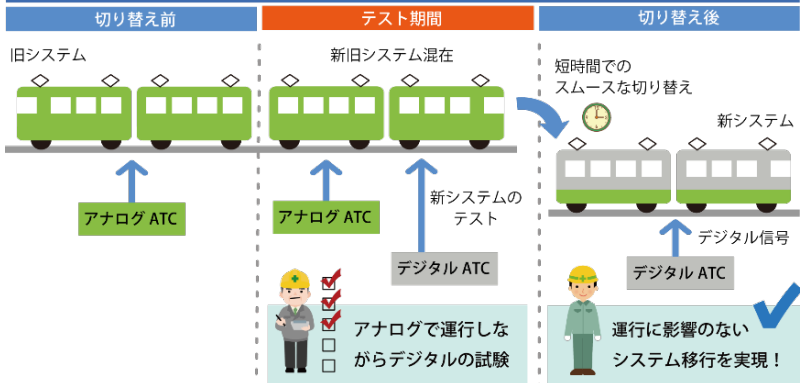
また、旧システム（アナログATC）から新システム第1段階（デジタルATC）への移行の際、運行を維持しながら迅速に新システムへの切り換えを行うため、開発当初からそこに焦点を当てて方式の設計を行い、旧システムで運転中に新システムの電文の送信、チェックが可能な方式を採用した。システムのライフサイクルを俯瞰した好事例である【目的指向と全体俯瞰】。

## 2世代以降の方式を見据えた段階的な機能の更新



2. 事例でみるシステムエンジニアリングによる問題解決

## 開発当初から全工程の課題を見据えた設計



## 2.5.2 本事例の背景と課題

### (1) ビジネスの背景

山手・京浜東北線は首都圏の最混雑線区（混雑率 236%；1998 年）である。混雑率の緩和は当時の大きな社会的要求であった。混雑緩和の対策として①編成両数増、②乗車定員の増、③運転本数の増（運転時隔短縮）などの対策があるが、①と②についてはすでに実施してきており山手・東北線でこれ以上の改善はできない状況であり、③の対策しか残されていない。ところが、当時の自動列車制御設備であるアナログ ATC 装置での列車本数は 1 時間 24 本が限界であり、これ以上の増発は不可能となっていた。

当時、山手・京浜東北線のアナログ ATC 装置は 1981 年に使用開始してから 18 年を経過しており、老朽取り替え時期を迎えることとなり、取り替えにあたって、より高密度運転が可能であるシステムに変更することとなった。

それに関し、1987 年に JR 東日本になってから、次世代信号システムの在り方についての検討を行ってきた。そして、先行列車位置を無線で後方列車に伝えて、後方列車は先行列車の位置と自列車の位置からその間隔を計算し、必要なブレーキ制御を行う「無線列車制御システム」が将来のシステムのあるべき姿であるといった結論が得られていたため、その方式への移行が期待されていた。

### (2) ビジネスの課題

アナログ ATC の速度制御は、列車に対する許容速度の条件を、信号区間ごとに階段状に（例えば 90km/h-65-45-25-0 のように）指定し、車上でその速度まで列車を減速させるための階段状のブレーキ制御を行っていた。このため、無駄なブレーキ・緩解動作が多く、下位速度の区間に入ると突然ブレーキが動作し乗り心地が悪い、車両のブレーキ性能が向上しても制限速度は信号の位置によって固定されているため、列車間隔を詰めることができない、という課題があった。

また、速度を指定する地上機器の多くが 1 箇所集中し、そこから各軌道回路ケーブルを敷設し信号を伝達する構成となっているため、重厚で高コストな設備となっていた。一方、前述の「無線列車制御システム」を採用するには、無線での伝達技術を確立する必要があるが、山手・京浜東北線という首都圏



の重要な線区に初めて導入するには、セキュリティの確保と妨害対策など技術的課題が多いという問題もあった。

さらに、新方式への切り換えにあたっては、深夜から早朝までの夜間の短い保守・試験のための時間に加えて、昼間でも試験を行える環境を作り、試験のための時間を確保する必要があった。そのため、アナログATCで営業運転を行いながら、新システムの試験を行えるようにするため両システムを共存させることが求められた。

加えて、全線区を3回に分けて切り換えを行うことから、アナログATCとデジタルATCの境界が存在することとなり、この境界において自動的にシステム切り換えを行って運転の継続を確保していく必要もあった。

## 2.5.3 アプローチ

### (1) 対策の全体像

将来を見据えた方式としてATCの制御方式を、信号区間内の許容速度を指示する方式(図2.5.1)から、先行列車の位置と自列車の位置からその間隔を計算し、速度制御を行う方式に変更することを定めた【目的指向と全体俯瞰】。先行列車位置検出は、山手・京浜東北線で無線列車制御システムを採用するには技術的課題が多いため、第1段階(デジタルATC)として従来通りレール経由の信号によるもの(図2.5.2)とし、第2段階(ATACS)で無線による方式(図2.5.3)を採用することとした。列車の自位置認識方式は、当初から、その両方を想定したものとした【反復による発見と進化】。

また、旧システム(アナログATC)から新システム(デジタルATC)への移行の際、旧システムに影響しない新システムの検証方法が必要であり、運行を維持しながら新システムへ迅速に切り換えを行わなければならなかった。そこで、開発当初から、全工程の課題を見据え、システムの切換移行についても設計時より焦点を当てて方式の設計を行い、旧システムで運転中に新システムの電文の送信、チェックが可能な方式を採用した【目的指向と全体俯瞰】。

### (2) 個別対策の概要

<対策1>：将来を見据えた方式としてATCの制御方式を、信号区間内の許容速度を指示する方式から、先行列車の位置情報を地上からデジタル情報とし

て送り、自列車の位置からその間隔を計算し、速度制御を行う方式に変更する新方式を開発した【目的指向と全体俯瞰】。

<対策2>：速度制御を行う機能は当初から開発したが、先行列車位置の後方列車への伝達は、最終の無線による伝送を目指しつつ、従来通りレールを通して伝送するシステムとした。一方、ブレーキ制御の論理を車上側に持っており、地上装置の簡素化を図るとともに乗り心地を考慮したブレーキ制御を行う最新技術を使ったデジタルATCの開発を行った【反復による発見と進化】【目的指向と全体俯瞰】。

<対策3>：デジタルATCの周波数選択をアナログATCとは異なるものとし、軌道回路（レールに信号を流すための回路）の長さも異なるのでアナログATC信号のみを通すバイパスフィルターを開発採用し、アナログATCとデジタルATCを共存させることを可能とした。さらに、このデジタル信号に試験用か実制御用かのフラグ（識別データ）を付けることにより、昼間の試験が可能となったとともに、アナログATCからデジタルATCへの切り換えのときはこのフラグを書き換えるだけにして、短時間の切り換えも可能とした。

<対策4>：工事切り換えを何回かに分けることにより、新旧ATCの突き合わせ境界ができ、この境界での運転を継続させるため、デジタルATC信号に切り換え区間か否かの情報を持たせるとともに、車両の制御装置にデジタルATC信号を受けたときにどのような動作をするかの論理を構築した。これにより段階構築を可能とすることができた。

### (3) コンテキスト

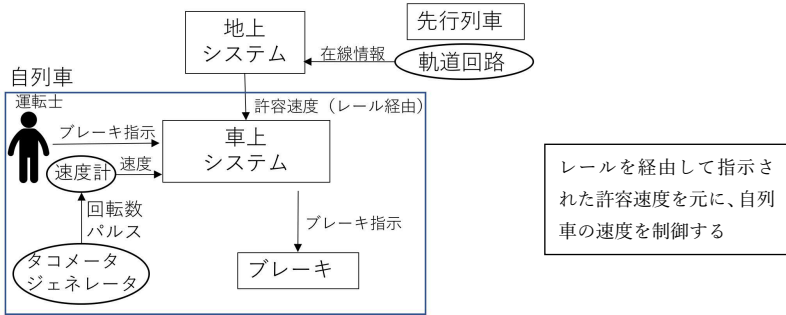


図 2.5.1 旧システム(アナログ ATC)のコンテキスト

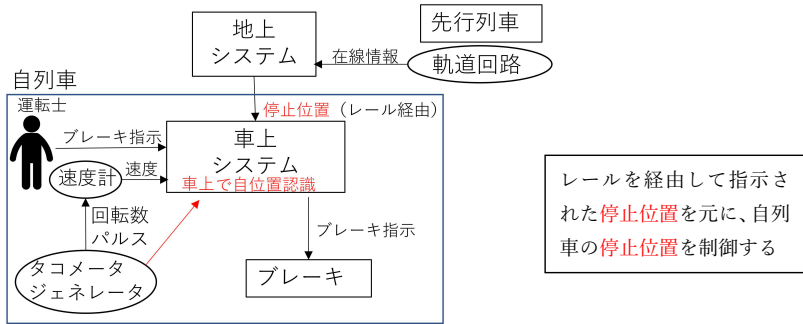


図 2.5.2 新システム第1段階(デジタル ATC)のコンテキスト

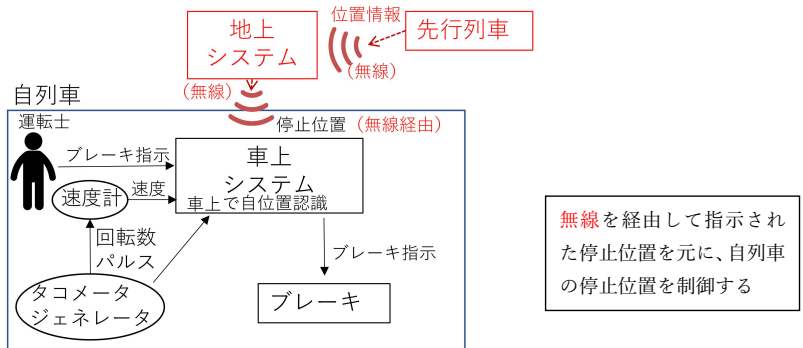


図 2.5.3 新システム第2段階(ATACS)のコンテキスト

## (4) プロジェクトの効果

従来の地上主体形のアナログ ATC に比べて、低コストでかつ機能拡張性のある車上主体形のデジタル ATC システムの開発を行い、京浜東北線（南浦和～鶴見）に 2003 年 12 月、山手線に 2006 年 7 年、京浜東北線・根岸線に 2009 年 8 月にそれぞれ導入使用開始を行った。

このデジタル ATC により一時間あたりの運転本数を 24 本から 26 本とでき、最混雑区間での乗車率も 236%から 220%以下にすることができた。また、コストもアナログ ATC 同種取替に比べて 20%程度下げることができた。

乗り心地についてもスムーズなブレーキ制御はお客様からの評判も上々であった。

夜間での試験の一部を昼間の運行時間帯に行うことができたことにより、試験期間も大幅に短縮でき、新旧 ATC の境界での運転も工事期間中問題の発生は一回もなかった。

さらに、アナログ ATC はリレー主体の装置であったものをコンピュータ主体のシステムとすることにより、システムの故障率も  $6 \times 10^{-5}$  [h] から  $4 \times 10^{-6}$  [h] とすることができ信頼度の向上が図れ、安定輸送に寄与することができた。

## 2.5.4 対策の詳細

### <対策 1> 許容速度指示方式から列車間隔制御方式への変更

将来を見据えた方式として ATC の制御方式を、信号区間内の許容速度を指示する方式から、先行列車の位置と自列車の位置からその間隔を計算し、速度制御を行う方式に変更することを定めた。列車制御装置として本来あるべき姿を考えた場合、列車が必要とする情報は「停止すべき位置」、「自列車位置」と「自列車速度」である。すなわち、地上からは停止すべき点の位置情報のみをデジタル情報として送り、車上は自列車位置を認識し停止点と自列車の距離を常に計算し、かつ線路の曲線、勾配などの条件を加味して必要な時機に適切なブレーキをかけるのが、デジタル ATC の基本原理である。このように車上自律システムであるところが、これまでのアナログ ATC と大きく異なる点である。

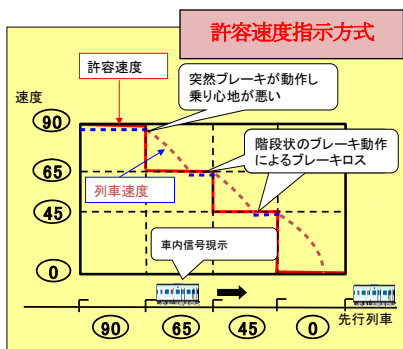


図 2.5.4 アナログ ATC 制御方式

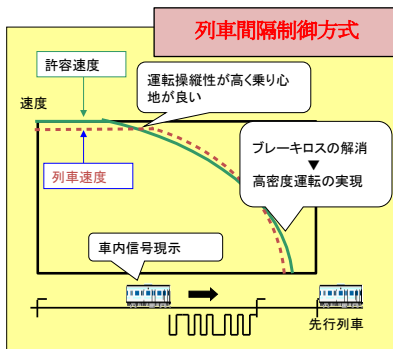


図 2.5.5 デジタル ATC 制御方式

停止位置を送る方式としたことで、停止位置までのブレーキを緩めることなく連続的にブレーキ制御を行い、ブレーキ距離を短くすることができ、結果、列車間隔を短縮できた。具体的には、アナログ ATC での運転時隔の 2 分 30 秒時隔を、減速度  $3 \text{ km/h/s}$  ( $0.83 \text{ m/s}^2$ )、停車時分 50 秒の条件のもとで理論的には 2 分に短縮可能なものとなった。実際の運転ダイヤでは、2 分 20 秒の運転時隔を採用している。

## <対策 2> デジタル ATC 化

先行列車の位置と自列車の位置からその間隔を計算し、速度制御を行う機能は当初から開発したが、先行列車位置の後方列車への伝達は、第一段階では、従来通り ATC 信号を、レールを通して伝送する方式にすることとし、第二段階として無線による伝達を行うこととした。一方、ブレーキ制御は<対策 1>に述べたように、その論理を車上側に持っていく、最新技術を使ったデジタル ATC の開発を行った。【反復による発見と進化】これまでのアナログ ATC は、速度信号を集中機器室からケーブルを通してレールに伝送する方式であったため地上に設備が集中しており、工事費が高くなっていた。これに対して、停止位置情報をデジタル信号で送り、速度制御を車上で自律的に行う方式としたことによって地上装置を分散構成のスリム化でき、車上装置のコストは上がったものの全体として低コストなシステムとすることができた。

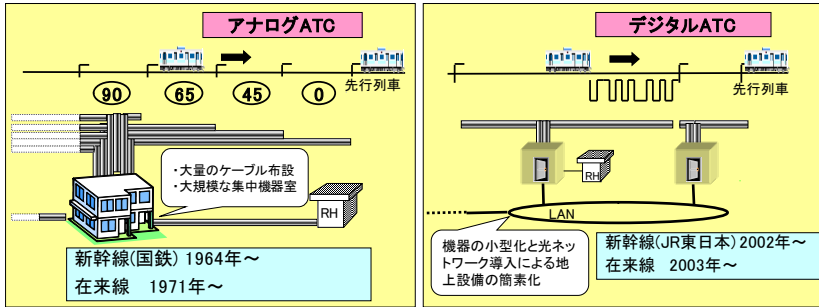


図 2.5.6 アナログ ATC における信号伝達 図 2.5.7 デジタル ATC における信号伝達

加えて、列車がブレーキパターンに接近すると、まず緩和ブレーキが動作し、乗り心地の悪化を防止する。さらにパターンに接近すると、フィードバック制御でブレーキ出力を調整しながら、パターンに沿って減速していき、停止直前で再度緩和ブレーキが動作し、大きなショックなく停止する。このように停止位置を目指してブレーキ制御を行う、車上自律制御を採用することにより、乗り心地も改善できるばかりでなく、開通状況を乗務員に示すため運転操縦性の改善も同時に図ることとなった。

### <対策 3>アナログ ATC とデジタル ATC の共存

首都圏の山手・京浜東北線は輸送量が国内でも最大で、早朝から深夜までの運転を行っており、ATC の切り換えにあたってこの輸送を阻害することは避けなければならない。そのためには、3 時間程度という深夜の保守作業時間の中で、新旧の ATC の切り換えを行うことが必須となる。

そこでまず、デジタル ATC の周波数選択をアナログ ATC とは異なるものとし、軌道回路（レールに信号を流すための回路）の長さも異なるのでアナログ ATC 信号のみを通すバイパスフィルターを開発採用し、アナログ ATC とデジタル ATC を共存させることを可能とした。

また、このデジタル信号に試験用か実制御用かのフラグ（識別データ）を付けることにより、アナログ ATC 制御のままデジタル ATC 信号の送信試験ができ、現行システムを生かしたまま、昼間での試験を可能とした。加えて、アナログ ATC（旧システム）からデジタル ATC（新システム）への切り換えのとき

は、デジタル ATC 信号の情報を「試験」から「制御」に変更するだけで、短時間の間にシステムの切り換えを実施することができるようにした。

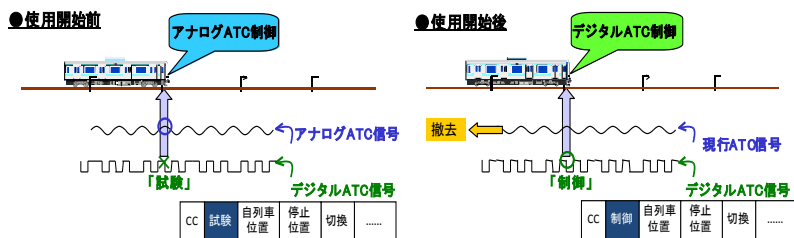


図 2.5.8 アナログ ATC からデジタル ATC への切り換えの概念

## <対策 4> 工事区間の境界箇所での ATC 方式の切り換え

ATC の更新においては、全線区を一度に更新するのではなく、以下の理由によりいくつかの段階に分けて更新を行わざるを得ない。

- ① 切換作業量が膨大となり、全線区を一度に更新するための人員が確保できない。
- ② コストがかさむ。
- ③ 切換時間が限られており、その時間内に施工するには、全線区は困難である。

したがって、全線の切り換えが完了するまでは、工事区間のシステムの境界箇所においてアナログ ATC とデジタル ATC の切換処理が必須となる。

これまで ATC の切り換えは基本的に乗務員が手動で行っていたが、走行中自動切換可能とし、ヒューマンエラーの防止を図る設計を検討した。これにもデジタル ATC 信号に切り換えを行うか否かの情報を付加し、乗務員の切換操作に変えることを考えた。

具体的には、アナログ ATC からデジタル ATC への切り換えを行う箇所では、アナログ ATC にデジタル ATC を重畳して送信しておいた。これにより、アナログ ATC 信号出制御しているときに、「デジタル ATC への切り換え」の情報を持ったデジタル ATC 信号を受信すると、デジタル ATC モードに自動的に切り換わるように車上システムにロジックを組み込んだ。逆に、デジタル ATC からアナログ ATC への切り換えを行う箇所では、デジタル ATC 電文に「アナログ ATC

への切り換え」の情報を付加し、この信号を受けると自動的にアナログ ATC モードに切り換えるよう車上システムにロジックを組み込んだ。

このように、多モードのデータを重複して流し、車上 ATC システムが判断、選択し、処理する自律分散機能を実現した。

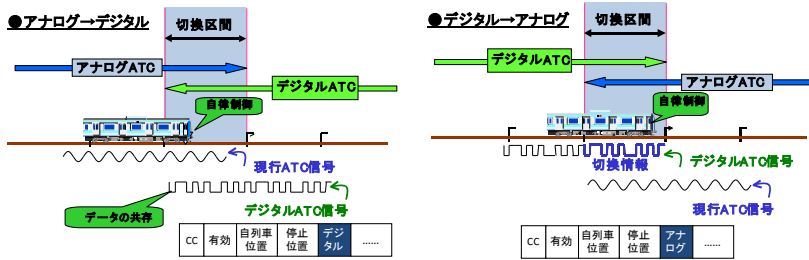


図 2.5.9 工事区間の境界箇所での ATC 方式の切り換えの概念

## 2.5.5 対策の効果のまとめ

山手・京浜東北線のラッシュ時の混雑緩和に対して、社会的要請が強まっていたので、運転本数を増加できる新しい列車制御システムを 2003～2009 年に導入した。当初の解決すべき課題は、混雑緩和、乗り心地改善とコストダウンであったが、旧システムから新システムへのスムーズな移行も重要な課題のひとつであった。つまり、現行システムに影響しない新システムの試験方法や、運行を維持しながらの新システムへの迅速な切り換えを行うことが必要であった。

開発当初から全工程の課題を見据え（時間軸の俯瞰）、システムの切り換え・移行についても設計時より焦点を当てて方式の設計を行い、旧システムで運転中に新システムの電文の送信、チェックが可能な方式を採用した。システムのライフサイクルを俯瞰して適切な対策を施した事例である。

その結果、テスト、および切り換えを短時間でできるシステムを実現し、運行サービスに影響することなくシステム移行を実践することが実現できた。また、ATACS における位置認識は、デジタル ATC で検証済みの方式を踏襲している。

これまでの、列車制御システムの開発変遷を以下に参考として示す。



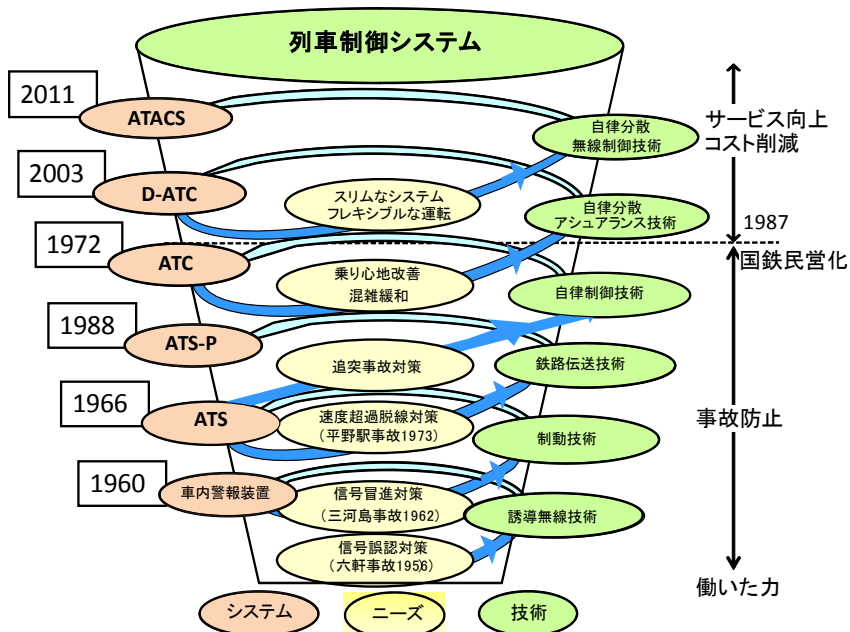


図 2.5.10 列車制御システム開発の変遷

## 2.5.6 教訓

### (1) 出来上がったシステムの特徴

列車の速度を制御する方式として、列車外部から速度信号を指示する方式から、その列車の停止位置を伝える方式に変更し、車上システムで速度を自律的に制御する方式を採用した。このことにより、列車運転間隔と乗り心地の改善を実現した。これを実現するために無線による列車停止位置伝送を俯瞰しつつ、レールを通したデジタル信号として列車停止位置を伝送する方式の開発、および列車が自らの位置を認識するための車輪の回転数から移動距離を算出するロジックの開発などの技術革新を行った。

また、方式の切り換えを短時間でを行うために、ATC 信号に識別符号を付けて、この符号を変えるだけで切り換えを実施するようにならただけでなく、営業時間帯でのテスト等円滑な移行も可能とした。

このように、システム設計を行う中で、時間的な俯瞰をしつつ、課題の解決手法を編み出してきた。

## (2) 結果を導いたアプローチの特徴

無線による伝送という将来の理想的な方式を見据えながら、技術的実現性を勘案し、停止位置を知らせる方式として、当面はレールを通して停止位置を知らせる方式を踏襲し、将来無線による方式へと段階的に進めることとした。ただ、その際デジタル伝送方式、車輪の回転数から移動距離を算出する技術などの先進技術の一部は先行して取り入れた。

また、切り換えにあたって、新旧のシステム制御が同時に存在できる方式を指向し、スムーズな切り換えを目指した。このため、基本システムを構築したのち、切り換え前後を想定し解決すべき課題について対策を検討し、システム設計にひとつひとつ反映していった。つまり、試験のフェーズ、切り換えのフェーズ、工事期間のフェーズ、運用段階でのフェーズごとに解決すべき課題を洗い出して、事業者（運転側、開発側、保守側）とメーカーとがプロジェクトを組んで検討していった、システム設計に組み込んでいった。

## (3) 今までのやり方では何ができなかったのか

これまで通りの進め方というのは、同種取替、つまりアナログ ATC を再び新設することとなる。この場合、中央装置はコンピュータを使用したものにするので併設できるが、ATC の信号をレールに送るケーブルは共用できないため新設するか、試験のたびに現行 ATC から新 ATC に切り換える必要が生ずる。このため、工事が膨大となりコストも高くなることとなる。

## 2.5.7 ポイント×プロセス対応表

活動全体を「システムズエンジニアリングのプロセス」と「システムズエンジニアリングの4つのポイント」に対応付けた表を示す。

表 2.5.1 ポイント×プロセス対応表

首都圏の高密度輸送を支えるデジタルATC開発		ポイント		その他	
	目的指向と全体俯瞰	多様な専門分野を統合	抽象化・モデル化	反復による発見と進化	
プロセス	(6.4.1)【ビジネスあるいはミッションの分析】				
	(6.4.2)【利害関係者と要求事項の定義】				
	(6.4.3)【システム要求事項の定義】				
	(6.4.4)【アーキテクチャの定義】	<p>＜対策1＞ATCの制御方式を、信号区間内の許容速度を指示する方式から、先行列車の位置と自列車の位置からその間隔を計算し、速度制御を行う方式に変更することを定めた。</p> <p>＜対策3＞アナログATCとデジタルATCを共存させることを可能とした。さらに、このデジタル信号に試験用が実制御用かのフラグ（識別データ）をつけることにより、屋間での試験が可能となったとともに、アナログATCからデジタルATCへの切換の時はこのフラグを書き換えるだけにして短時間の切換も可能とした。</p> <p>＜対策4＞新旧ATCの共存せ境界での運転を継続させるため、デジタルATC信号に切換え区間か否かの情報を持たせるとともに、車両の制御装置にデジタルATC信号を受けた時にどのような動作をするかの論理を構築した。これにより段階構築を可能とすることができた。</p>			
	(6.4.6)【システム解析】				
(6.4.9)【検証】					
(6.4.11)【妥当性確認】					

プロセス

＜対策2＞速度制御を行う機能は当初から開発し第1段階のデジタルATCから実装したが、先行列車位置の後方列車への伝達は、第2段階のATACSでは無線による伝達を目指しつつ、デジタルATCとしては従来通りレールを通して伝送するシステムとした。

## 2.6 Web スキャンシステムの企画開発

### 2.6.1 本事例の内容

#### (1) 概要

成熟しつつある製品においては、売上拡大の施策として「よりコンパクトに」、「より高速に」、またはそれらを含めた「品質の向上」といった目標を設定し、新製品の機能を追求することが、マーケットシェア拡大につながると考える傾向にある。

しかしながら、ユーザー側からは、機能向上は望ましいが、ほとんど使用しない機能や細かい設定が増えて逆に分かり難くなっているとの意見もある。

新しい機能やトータルでの品質の向上自体は、当然必要ではあるが、それだけにとられない、発想の転換が必要である。

この事例においては、スキャナーという技術的には成熟した商品において、近視眼的になりがちな既存の企画・開発部門に一石を投じるために、全体俯瞰により取り巻くビジネス環境の中での製品の立ち位置を見直し、新しい使い方を提案するに至っている。この一連の取り組みにより“モノ”から“コト”へのビジネス転換を実現したといえることができる。

#### (2) 本事例の特徴

他部門から新しいリーダーを迎えて部門横断のタスクフォースを立ち上げ、検討に取り組む環境を整えた上で、従来とは異なったアプローチを実践した。このタスクフォースの中ではリーダーを含め、システムズエンジニアリングの体系だった知識はなかったものの物事を進めるにあたっては、ビジネス自体を見直すというリーダーの強い意志があった。これに従ってビジネス環境を俯瞰することで新たなステークホルダを特定し、要求仕様の獲得と実現というプロセスを繰り返し実施したことがポイントである。

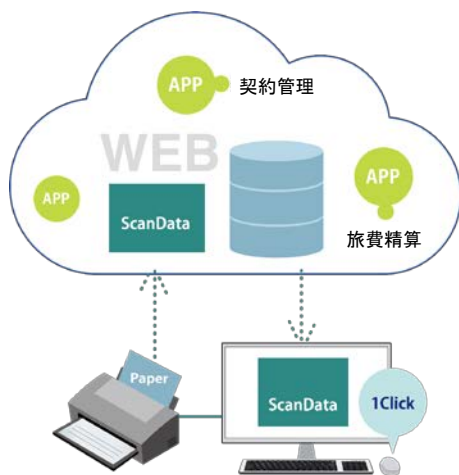
まず「スキャナー」のビジネスシーンでの使い方を見直す中で、想定する業務において Application Service Provider(以降 ASP 事業者) が提供するクラウド上のサービス利用が増加していることに着目した。

そこで、今まではターゲットとしてこなかった新しいステークホルダである ASP 事業者に対して、サービスの一機能としてどうしたらスキャナーを使って

もらえるのか（要求）を獲得するために、市場規模が大きくかつASP事業の先進である米国でのヒアリングを敢行することにした。ASP事業者からの要求に対して、プロトタイプを繰り返し、より詳細な要求を獲得して最終的には、今までありそうでなかったブラウザからワンクリックでスキャナーを起動してデータを取り込む機能を開発するに至った。

さらに、ASP事業者の先には、これらサービスを契約した不特定の企業と従業員が存在するため、コスト削減の観点から、取り込んだファイルの仕分けやデータ利活用を自動化する機能も盛り込むことで新しい使い方を提案し、これらを実現する機能をSDK（Software Development Kit）という形で無償提供し、他社との差別化を図った。

ビジネス面では、ASP事業者の連携ソリューションとして認知され、ASP事業者からの推奨/PRを得ることで販路が拡大している。



## “モノ” から“コト” への ビジネス転換の実現

ビジネスシーンにおける  
製品の立ち位置を見直し、  
新しい使い方の提案

## 2.6.2 本事例の背景と課題

### (1) ビジネスの背景

スキャナー市場は、先進国で販売台数が伸び悩み、前年比2%減（業界全体）、さらに今後は出荷台数が横ばい、あるいは微減する見通しとなっており、市場自体に拡大傾向を望むことが困難な状況となっている。

加えて、新規参入メーカーと既存メーカーの間での競争が激化し、各社ともに販促強化策を打ち出している。

他方、ペーパーレス化・電子化が加速しており、新しいビジネススタイルの台頭、法改正もこれに拍車をかけている。

- ◆ 紙を使用しないビジネスモデルの台頭
- ◆ 領収書等の証憑電子化の認可

このような状況下で、早急に新しい施策を要求されていた。

### (2) ビジネスの課題

潜在的なニーズを探索し、既存のビジネスモデル（「プロダクトアウト型」）を見直す必要があり、新たなスキャナーの使い方をハードウェアメーカー側から提案し、マーケットを拡大させる必要があった。

また、ハードウェアの機能自体が成熟しつつある中でいかにして他社との差別化を図っていくかということが重要課題であり、クラウドやRPA（Robotic Process Automation）といった流れにどのように対応していくのかを明らかにしていくことも今後のビジネスを左右すると考えられていた。

## 2.6.3 アプローチ

### (1) 対策の全体像

これまでは単純に紙ドキュメントを電子化する機械としての位置づけのもと、ハードウェアを主体としたビジネスを展開していた。

具体的には、スキャナーを使って紙媒体を電子化してPC内のHDDにファイルを保存、保存されたファイルを指定して特定のフォルダにアップロードするという単純なモデルで、当然のことながら使用用途には関与せずユーザーが必要

に応じてスキャナーを使用していた。この時点でのコンテキストを図 2.6.1 に示す。

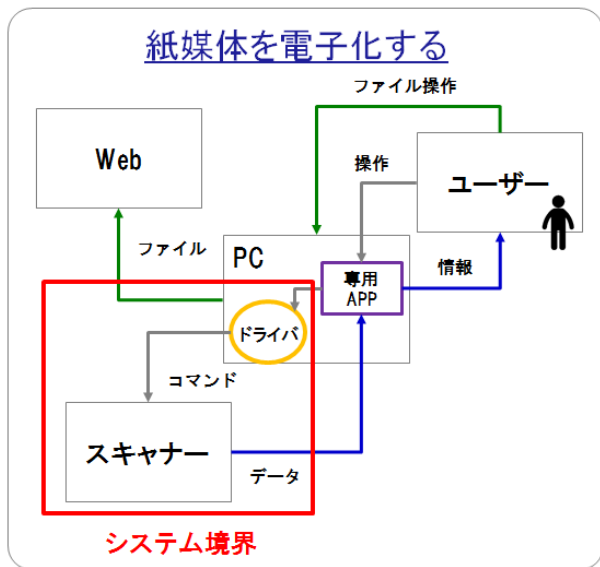


図 2.6.1 これまでのコンテキスト

これに対して今回の取り組みの結果、Web アプリケーションと連携する機能を新規開発し、クラウド上の Web アプリケーションからワンクリックでスキャナーの起動、読み込み、読み込んだファイルを ASP 事業者が提供するクラウド上のストレージに保存するという一連の流れを自動化した。これにより、Web アプリケーションサービスの一部としてのビジネスモデルを確立することができた。

## (2) 各個別対策の概要

システムズエンジニアリング視点では、下記 1~5 の対策が挙げられ、これらの対策が有効に機能した結果、新しいビジネスモデルを確立した。

<対策 1>：市場の閉塞感を打破するために、今までとは全く違った視点でビジネスを見直すことで新しいステークホルダを見出した。また、正しいユー

ザーニーズを獲得するために最大マーケットである米国において、新しいステークホルダである ASP 事業者から直接ヒアリングを実施した。

＜対策 2＞：より詳細なニーズを発掘するためにプロトタイプ開発を採用し、ASP 事業者と合意形成を図りながらプロジェクトを進めた。

＜対策 3＞：ASP 事業者の先にいるエンドユーザーニーズを実現するために業務を作業単位に分解して効果を把握した。その上で実装方法として無償で SDK を提供するスタイルをとり、エンドユーザーのシステム管理部門で簡単に実装可能な仕組みとした。

＜対策 4＞：Web で活用するためには、セキュリティが最優先と捉え、プロトタイプ段階からセキュリティ対策を実装した。

＜対策 5＞：妥当性を維持するために、節目においては随時レビューを実施し、決定事項に対しては常に理由を明記することでトレーサビリティの確保に努めた。



上記対策を進める中で、新たに定義されたコンテキストを図 2.6.2 に示す。

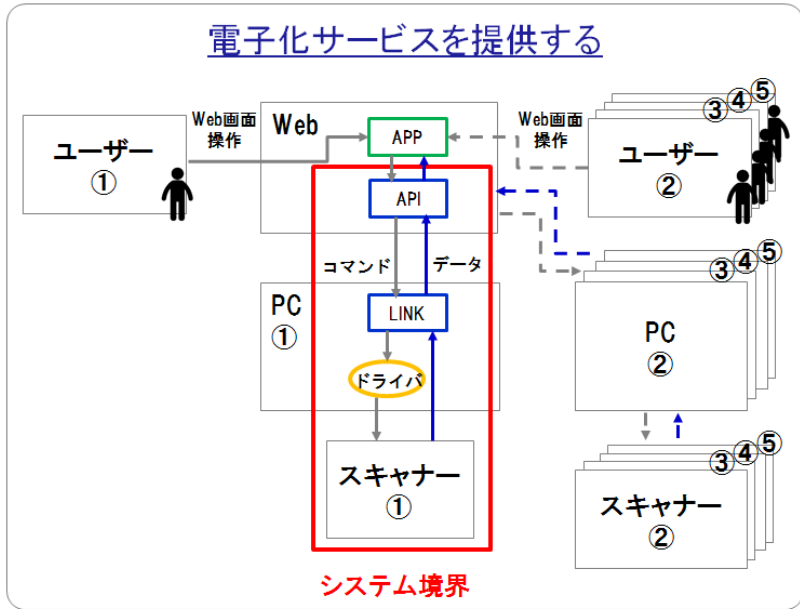


図 2.6.2 新しいコンテキスト

### (3) アプローチの効果

これまでとは違った視点、具体的にはビジネスシーンでの使われ方の視点で俯瞰することにより新しいステークホルダを見出すことができた。

新しいステークホルダとは、具体的には、ASP 事業者および ASP ユーザーであるが、彼らの要求を実現することでこれまでと違った新しいビジネスモデルを確立することができ、実現するための新しい仕組みを提案することができた。

ステークホルダの要求を整理していく中で「使い勝手」や「重要ポイント」、「懸念事項」などを標準とそれ以外に分類し、標準部分を ASP 事業者共通の要求として実装することで複数の ASP 事業者へのサービス提供を進めている。

## 2.6.4 対策の詳細

### <対策 1>ビジネスの俯瞰

視点を一段上げて、スキャナーが使われるビジネスシーンを想定して検討を重ねるうちに、ビジネスシーンでの対象と想定する業務にASP事業者が提供するクラウド上のサービスを活用しているケースが多くみられることが分かってきた。このように従来の発想にとらわれない視点でビジネスシーンを観察することで、結果としてASP事業者という新しいステークホルダを見出すことができた。これらASP事業者から意見聴取するために、市場規模が大きくかつASP事業の先進である米国での調査を敢行することでビジネスの骨格が固まった。

【システムズエンジニアリングのポイント：目的指向と全体俯瞰】

【関連プロセス：ビジネスあるいはミッションの分析】

### <対策 2>プロトタイピングによる反復

ASP事業者から聴取した内容をベースにプロトタイプを作成し、プレゼンテーション・評価により詳細なニーズを発掘し、さらにプロトタイプモデルに反映するというルーチンを繰り返して合意形成を図った結果、齟齬を解消し信頼を獲得するに至った。

また、米国のみの特化したニーズについてはフィルタリングすることでシンプルな仕組みとした。

【システムズエンジニアリングのポイント：反復による発見と進化】

【関連プロセス：利害関係者ニーズと要求事項の定義】

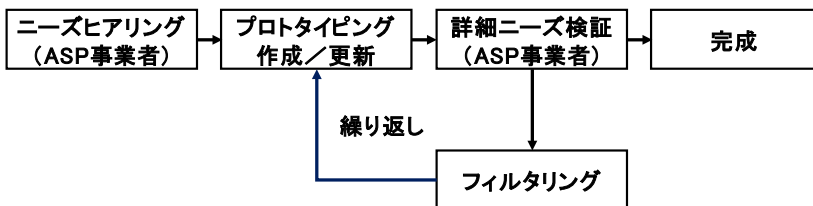


図 2.6.3 プロトタイピングによる反復

## <対策 3> ユーザーニーズを踏まえた機能提供

ニーズを定義する場合に、ASP 事業者の背後には不特定多数の企業・ユーザーが存在することも考慮に入れておかなければならなかった。

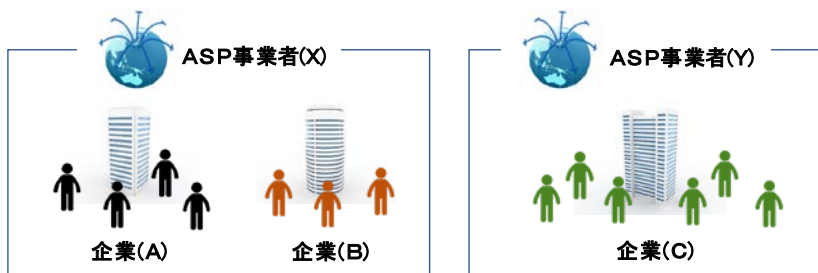


図 2.6.4 ステークホルダ

このようなユーザーのニーズの中で最大の関心事項が「コストの削減」であると理解し、ユーザー業務を作業レベルに分解して「スキャナーを保有していないケース/Web スキャンシステム<sup>1</sup>を使用するケース」の形で整理した。

作業自体を減らすことができるか、または作業負荷を軽減することができるかを明らかにした上で業務をシンプルにするための機能として実装した。

図 2.6.5 および図 2.6.6 に示すようにエンドユーザーの要求を満足し、作業工数を削減することで「運用コストの削減」を実現した。

また、提供方法としては SDK というスタイルで無償配布し、SI ベンダー等の力を借りることなく、エンドユーザーのシステム管理部門において Web アプリケーションに機能をアドオンできる容易性を備えることで、「導入コスト」の削減もあわせて実現した。

【システムズエンジニアリングのポイント：目的指向と全体俯瞰、抽象化・モデル化】

【関連プロセス：システム要求事項の定義】

<sup>1</sup> スキャナー本体とスキャンツール群を合わせた総称

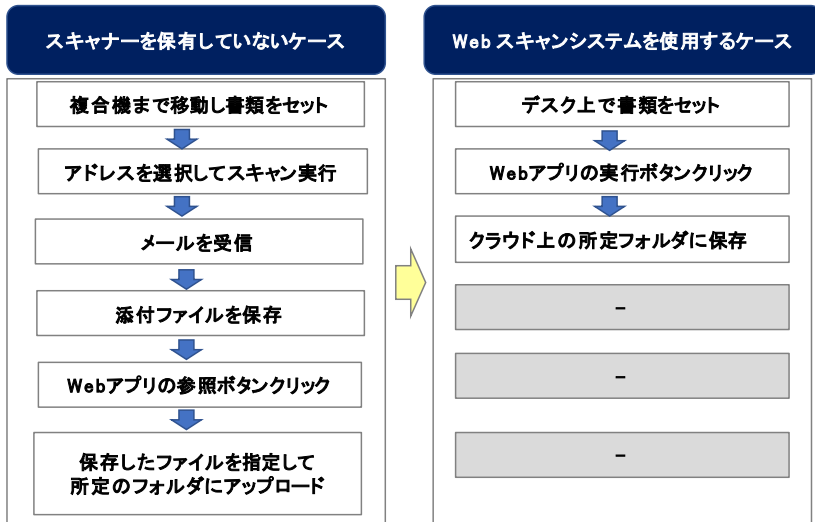
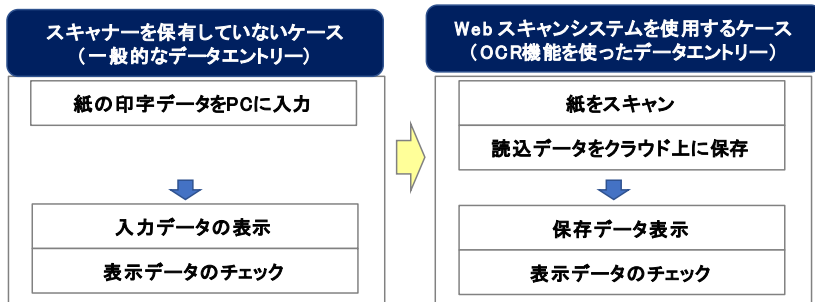


図 2.6.5 スキャンの作業フロー例



OCR: Optical Character Reader

図 2.6.6 データエントリーの作業フロー例

## <対策 4>セキュリティの作り込み

アーキテクチャを検討する中で、ASP 事業者にとって特に重要な関心事がセキュリティであることが判明した。これまでのハード主体のビジネスモデルでは、セキュリティを意識する必要はなかったため、関係者の知見を集めて事前にリスク評価を実施した上でプロトタイプ of 製作に着手した。本来セキュリティ

いは要求段階から検討して作り込むものであるため、このプロセスに従ったものと言える。

リスク評価の結果、

- ◆ データのアップロード時の認証
- ◆ Web から実行コマンドが発行されたときの安全性（悪意のある Web サイト対策）

が最重要と考えられた。

アップロード時の対策としては、一般的に普及している「OAuth 認証」を実装することで対応し、悪意のある Web サイトから保護するための対策としては、「CORS」を実装してアクセス制御をかけた。リスク評価結果を踏まえて、初回プロトタイプからセキュリティ対策を実装したことで、ASP 事業者の不安を一掃することができた。

この際、新たなリーダー（決定権者）のもとに既存の開発部門に加えて、Web 技術者やセキュリティ技術者などの知見を集め合意形成活動を進めた。

【システムズエンジニアリングのポイント：目的指向と全体俯瞰、多様な専門分野を統合】

【関連プロセス：システム要求事項の定義】

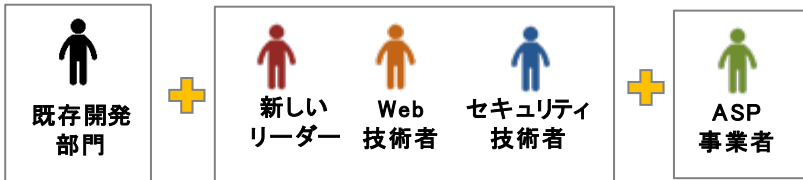


図 2.6.7 多様な専門分野

## <対策 5>トレーサビリティの確保

ビジネス要求を実現するということを念頭に随時レビューを実施し、方向性やリスクの確認を行うとともに、決定事項には常に理由を明記した。

理由を明記することでビジネス要求側からシステム要求側へのトレース、逆にシステム要求側からビジネス要求側へのトレーサビリティを確保した。

【システムズエンジニアリングのポイント：目的指向と全体俯瞰】

【関連プロセス：妥当性確認】

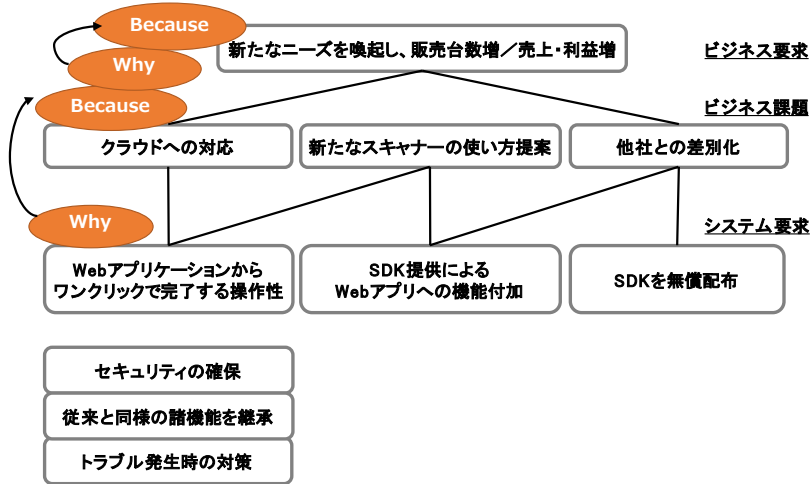


図 2.6.8 双方向でのトレース

## 2.6.5 対策の効果のまとめ

これまでとは、異なるステークホルダ（ASP 事業者）をターゲットとすることでビジネスの視点が変わり、サービスの視点で見ることによって、ビジネスの幅を広げることができた。

また新しいスキャナーの使用方法、すなわちサービスの一機能として利活用するというモデルの提案が可能となり、コアコンピタンスとして位置づけて、他社との差別化を図っている。サービス指向に舵をきったことで、新しいサービス形態に対応するための新しい要求を汲み取れる土壌ができつつあると思われる。

## 2.6.6 教訓

### (1) 出来上がったシステムの特徴

スキャナーを Web アプリケーションに連携させることを命題に、既存のスキャナーと Web 上からスキャンするためのツール群を対象システムと捉えた。こうして開発を進めた結果、Web アプリケーションからワンクリックでスキャナーの起動からクラウド上へファイルを保存するまで、一連の作業を自動化でき

るシステムを市場投入することができた。スキャナーと Web アプリケーションはそれぞれに単独で動かすことができる独立したシステムであるが、このふたつのシステムを連携させることでサービス機能を提供し、スキャナーに新たな価値を付加することができた。

## (2) 結果を導いたアプローチの特徴

これまでとは異なり一段上からビジネスシーンを俯瞰し、ビジネス分析からステークホルダ要求を獲得するという一連のプロセスを踏むことの有効性を認識できた。

一段上からの俯瞰と言葉で言ってしまうと、至極当たり前のようではあるが、できているようで、実はできていないことが多い。外側の世界との関わりを俯瞰し、潜在的ニーズを獲得することが最も重要である。

このビジネスの俯瞰を発端として、新たなステークホルダを発見し、プロトタイプによる要求の獲得とそれら要求の実現を繰り返すことで、ビジネスパートナーとして良好な関係を構築することができている。

## (3) 今までのやり方では何ができなかったのか

取り巻くビジネスを俯瞰して新しいビジネスモデルを導き出すことができず、今までの延長線上で、製品開発を重視し、製品の機能向上（スキャン品質・速度）による差別化を追求してマーケットシェア向上を目指していたと想像される。

ハードウェアを生産・販売するビジネスモデルからは脱却できず、“モノ”から“コト”へとと言う時代の流れに乗り遅れていた可能性があると思われる。

## 2.6.7 ポイント×プロセス対応表

活動全体を「システムズエンジニアリングのプロセス」と「システムズエンジニアリングの4つのポイント」に対応付けた表を示す。

表 2.6.1 ポイント×プロセス対応表

Webシステム開発の企画開発		ポイント			反復による発見と進化	その他
	目的指向と全体俯瞰	多様な専門分野を統合	抽象化・モデル化			
(6.4.1)【ビジネスあるいはミッションの分析】	市場規模が大きい且つASP事業の先進である米国での調査を敢行することでビジネスの骨格が固まった。	他部門から決定権限者を招聘し、従来の発想にとらわれない視点でビジネスシーンを観察することでASP事業者という新しいステークホルダーを見出すことができた。				
(6.4.2)【利害関係者ニーズと要求事項の定義】	ASP業者の最大の関心は、セキュリティであることが判明。ASPユーザーの最大の関心は、コスト削減であることが判明。				ASP事業者から聴取した内容をベースにプロトタイプを作成し、プレゼンテーション・評価により詳細なニーズを発掘した。	
(6.4.3)【システム要求事項の定義】	ASP事業者の最大の関心は、セキュリティであることが判明した為、事前にリスク評価を実施した上でプロトタイプの製作に着手した。	新たなマネジメントのもと既存の開発部門加えて、Web技術者やセキュリティ技術者などの知見を集め合意形成活動を進めた。	ユーザー業務を作業レベルに分解してBefore/Afterの形で整理した。作業自体を減らす又は負荷軽減することができぬかを明らかにした。			
プロセス	(6.4.4)【アーキテクチャの定義】					
	(6.4.6)【システム解析】					
	(6.4.9)【検証】					
	(6.4.11)【妥当性確認】	ビジネス要求を裏支えるということを含頭に随時レビューを実施。				
その他	決定事項には常に理由を明記した。理由を明記することでビジネス要求側からシステム要求側へのトレース、逆にシステム要求側からビジネス要求側へのトレースを確保した。				プロトタイプ開発で合意形成を固めた結果、両者の相語を解消し信頼を獲得するに至った。米国のみに特化したニーズについてはフィルタリングすることでシンプルな仕組みとした。	

よる問題解決

2. 事例でみるシステムズエンジニアリングに



## 第3章 システムズエンジニアリング を始めるにあたってのヒント

システムズエンジニアリングが自社の業務に役立ちそうだなと思ったら、どうすれば良いだろうか。本書を読んでやる気になった人、システムズエンジニアリングのセミナー等を聞いて感銘を受けた人、上司からシステムズエンジニアリングに取り組みと指示を受けた人、そういう人達は、次に何をすれば良いのか。

本章では、具体的にシステムズエンジニアリングの考え方を問題解決に役立てようと思った人のために、自らの業務の中でシステムズエンジニアリングのアプローチを利用するにあたり、第一歩を踏み出すための手助けを、二つの実践ケースに基づき、考察を含めてヒントとして提供する。

## 3.1 システムズエンジニアリングを始めるにあたって考えるべきこと

活用のヒントとして、「どこから始めるのか」、「どのように進めるのか」がポイントとなる。それぞれにヒントとして考えらえる項目を紹介する。

### 3.1.1 どこから始めるのか

現実の業務の中でシステムズエンジニアリングを活用する場合、ほとんどの企業においては、すべてのプロセスを置き換えて実施することは現実的でなく、その第一歩として、例えば、以下のケースが想定される。

(1) システムズエンジニアリングのポイントを「目的指向と全体俯瞰」と捉え、既存の開発プロセスと並行して、ビジネスあるいはミッションの分析、または、利害関係者ニーズと要求事項の定義など、上流の活動を付加する。

(2) システム要求の明確化に資するために、「抽象化・モデル化」が重要と考え、モデリング手法の習得・導入を行い、既存プロセスを強化する。

(3) システム解析や、システム検証を行うために、上流工程で、検討・設計を行ったり、妥当性確認のレビューを行ったりする。

これ以外にも、いろいろなケースがありうるが、まずは例示として、これら三つのケースにつき、第2章で使用したポイント×プロセス対応表における位置づけを示す。

	ポイント				
	目的指向と全体俯瞰	多様な専門分野を統合	抽象化・モデル化	反復による発見と進化	その他
(6.4.1) 【ビジネスあるいは ミッションの分析】	(1)設計の上流と して、ビジネスと 開発対象を整理		(2)モデリング手 法の習得・導入	4つのポイント	
(6.4.2) 【利害関係者ニー ズと要求事項の定 義】					
(6.4.3) 【システム要求事 項の定義】					
プ ロ セ ス (6.4.4) 【アーキテクチャの 定義】	代表的なプロセス				
(6.4.6) 【システム解析】					
(6.4.9) 【検証】		一般的な既存開発プロセスの適用範囲			
(6.4.11) 【妥当性確認】	(3)上流工程で妥 当性確認レビュー				

図 3.1.1 システムズエンジニアリング活用の位置づけ

### 3.1.2 どのように進めるのか

新たな仕事の進め方を適用するにあたっては、過去いろいろな方法が提案されており [9]、それをもとに、どのように進めるのかのヒントとして、そのステップを以下のように整理した。

#### (1) 事業ニーズと導入動機の整合の整理

会社や組織の問題意識（事業ニーズ）と、システムズエンジニアリング導入はどうつながるのかを整理する必要がある。事業ニーズには、例えば、「新事業に挑戦する」場合、「ベテランの経験の明示化」の場合などが考えられる。

「新事業に挑戦する」場合は、現場（プロジェクトメンバー）にとってのシステムズエンジニアリング導入のメリットが腹落ちしているのか、仕事の質の向上を期待しているのか、あるいは現場の仕事の効率化を期待しているのか、などの目的と動機に関する確認が必要である。

また、「ベテランの経験の明示化」の場合は、暗黙知として扱われてきた知見の形式知化であるが、世代交代の継承のためなのか、あるいは、新環境にお

いて、従来明示的に重視してこなかった品質特性（例えばセキュリティなど）について、体系的な設計を行うための基礎固めが目的なのか、どのような目的であるかの明確化が必要となる。

事業ニーズをもとに、経営者（組織）がやれ、と言って、後から現場が現場なりの意義を見出す場合と、現場がやりたいことを、経営者（組織）に納得させる場合がありえる。いずれの場合も、新しい技術や方法（この場合、システムズエンジニアリング）に取り組もうという関係者の動機、目的は微妙に食い違っているのが普通であり、違う目的であっても、最終的にみんながやろうという気になるように進めることが大事である。

## (2) 現状プロセスの評価と導入対象の特定

従来の仕事の仕方、開発標準などと新しいやり方はどこがどう異なるのか調べる。

ギャップ分析、アセスメントなどとも言われる。従来のやり方と違う部分、欠けていた部分について、すべて新しいやり方に従うべきか？ギャップのうち、どこかに重点を絞ったり、優先順位を付けたりすることで、進展することもある。ギャップを調べる過程で、新しいやり方（システムズエンジニアリング）に対する理解が促進される。

## (3) 教育および支援の獲得

システムズエンジニアリングをどうやって勉強するのかの検討が必要となる。専門家、経験者のコンサルティングを受けられるのか、一部の先導者がじっくり勉強し、残りの大勢は軽く勉強して先導者に従うのか。少数の先導者（1人のことが多い）は、強い思いで行動（勉強）するが、チームを巻き込むのは大変である。他社事例、特に通し事例などが有効と考えられる。また、業界団体、専門家団体などの場で情報を得ることも期待される。また、システムズエンジニアリングを実践するプロセスは、ISO/IEC/IEEE 15288 System life cycle processes として長年にわたり ISO/IEC と IEEE において議論され改訂されてきている。第4章に紹介するので参照されたい。

#### (4) 限定適用とその効果の確認

とりあえず部分的にやってみましょう、ということになって、部分的な目標を限定してやってみる。現実にはやり方が稚拙だったり、結果がお粗末だったりすることもあるが、前向きな行動の中で、理解が深まったり、新しいやり方の特性が理解されてくる。

ここで気を付けたいのは、システムズエンジニアリングはあくまでも本来の目的に対してシステム全体の実現を進めることに寄与するアプローチであって、例えば、ライフサイクルを考慮せずに開発工程だけを対象にした一過性のバグ削減、コスト削減などの個別指標の改善手段として適用することに必ずしも適したものではない。したがって、限定適用と部分的な目標の設定においても十分に全体を考慮することが重要である。

#### (5) 理解の深まりと目的の再設定

限定適用の結果、どこにどう効くか、自分達なりの解釈が出来上がる。その解釈に沿って、適用する作業の範囲、関与する人などを増やそうという検討が可能になる。

#### (6) 本格適用に向けた課題の特定と対策

本格的なプロジェクトに適用するために、教育訓練を拡大するとか、作業要領的な文書を充実するとか、新しいやり方に沿ってプロジェクト計画や開発体制を手直しする。効果測定の手段を仕込む。また、実際の商談として適用されることもあるので、商談推進、企画、技術、保守などの各部門の協力体制を組んでおくことも大事である。

## 3.2 システムズエンジニアリング実践ケース

本節では、本書の読者が遭遇するであろう、より身近なケースを取材して、3.1に示したステップに従って考察を加えヒントを整理し、システムズエンジニアリングの実践ケースとして紹介する。

### 3.2.1 実践ケースの概要

最初のケースでは、システムズエンジニアリングの推進者が、システムズエンジニアリングの4つのポイントの中でも特に「目的指向と全体俯瞰」を高く評価し、システムズエンジニアリングに取り組んでみようと思って始めたものである。問題解決を日々の業務として高い熟練度で取り組んでいる組織で、「既存のやり方に従いつつ問題を解決したい」という現場からのニーズに対応しつつ、システムズエンジニアリングを導入したいと考えており、3.1.1(1)のケースに相当する。

各会社では通常、開発手順が社内標準として規定されており、各開発者もそれに従って、開発・製品化を行っていることであろう。したがって、現状の社内標準の意図を守りながら、既存の開発プロセスと並行して、ビジネスあるいはミッションの分析、または、利害関係者ニーズと要求事項の定義を付加し、問題解決を図る場合の課題が抽出されている。

次のケースは、システムズエンジニアリングの推進者が、システム要求の明確化に資するために「抽象化・モデル化」が重要と考え、モデリング手法の習得・導入を意図した場合である。このケースでは、実際の現場での適用に先立ち、パイロット的にプロジェクトを実施し、社内外関係者に対してその有効性を示しつつ、自らのスキルアップにつなげており、3.1.1(2)の場合に相当する。

このとき工夫したのが、パイロット的プロジェクトの対象の選定であった。モデル化の十分なスキルを取得するには、そのプロジェクト対象の詳細なドメイン知識が得られることが重要である。しかし、簡単な例をパイロットとして選定すると、その対象の内包する課題が矮小であり、スキルアップにつながらない。一方、複雑な例では、そのドメイン知識が十分に得られないことから、システムズエンジニアリングとして消化不良になることがあり、十分な成果が

得られない。この場合、「農園モニタリングシステム」という適度な複雑さを持ったシステムを対象にしたことおよび、その例の専門家があり、そのドメイン知識を獲得可能な環境のふたつを同時に満たせる対象の選定が可能となった。

## 3.2.2 社内標準にシステムズエンジニアリングを取り入れた場合

### (1) 事業ニーズと導入動機の整合の整理

本ケースは、問題解決を日々の業務として高い熟練度で取り組んでいる組織で、新たな環境のシステムに取り組むことをきっかけにシステムズエンジニアリングの導入を図る場合である。[10]

各会社では通常、開発手順が社内標準として規定されており、各開発者もそれに従って、開発・製品化を行っていることから、現状の開発手順を守りながら、システムズエンジニアリングを参考にし、新たな環境における問題解決を図りたいという希望は、当然のことと思われる。

本ケースでは、システムズエンジニアリングの推進者が、システムズエンジニアリングの4つのポイントの中でも特に「目的指向と全体俯瞰」を高く評価し、それに組み込んでみようと思って始めた。そして、「既存のやり方に従いつつ問題を解決したい」という現場からのニーズに対応しつつ、システムズエンジニアリングを導入した。現状の社内標準の意図を守りながら、既存の開発プロセスと並行して、ビジネスあるいはミッションの分析、または、利害関係者ニーズと要求事項の定義を付加し、問題解決を図る場合の課題が抽出されている。

### (2) 現状プロセスの評価と導入対象の特定

対象者としては、従来、単品の製品を開発し、一定の成功は収めてきたが、その製品を含めた付加価値の高いサービスを実現しようとする人、および要件が決まればきっちり作る自信はあるが、自らの技術、製品を取り巻く環境を一段高い視点から分析しなければならなくなった人などが相当する。一方、現状プロセスは、一般的に図 3.1.1 に示したように、そのような「全体俯瞰」を行

うことが明確に規定されていないことが多い。そのため、その差異を埋めることが、導入の対象である。

### (3) 教育および支援の獲得

組織の開発標準とシステムズエンジニアリングのプロセスの差異を分析するため、その双方の知識が必要となる。想定されるケースは、社内の開発プロセスを熟知した担当者と、社外を含めて、システムズエンジニアリングのプロセスに知見を持った担当者が共同で業務にあたることである。システムズエンジニアリングを実践するプロセスの知識としては、前述したように、ISO/IEC/IEEE 15288 System life cycle processes も参考になる。

### (4) 限定適用とその効果の確認

その組織標準とシステムズエンジニアリングのプロセスの差異を分析して、その組織の弱みをもとにプロセス改善を行う。対象として、いくつかのケースが考えられるが、この事例では、以下の2種のケースを選択した。

① 基本的考え方としての「目的指向と全体俯瞰」の意識を高める

対象： システムのビジネス企画、開発に関わるすべての人

内容： 上記の考え方を備え、システムを企画、開発するグループ内に浸透させる。

期待効果： 今後のシステム開発には必要な考え方が浸透する。

なお、これは必要条件であり、これにより何か成就するわけではない。

② システムライフサイクルプロセスを参考にした改善を行う。

対象： 実績ある開発標準を持っている組織、プロジェクトなど

内容： 「システムライフサイクルプロセス」の7プロセス（第4章で示す7プロセス）を参照し、自部門の開発標準と照合して意味を考える。そして自部門の有識者の暗黙知を、この7プロセスを参考に整理、評価して、新たな開発標準の運用に供するようドキュメント化する。

期待効果： 効果的なシステムズエンジニアリングの活動着手のポイントが見えてくる。



システムライフサイクルプロセスと自部門の開発標準の照合においては、プロセスの単位で大きな比較を行い、差異を明らかにする。この際、細部の差異は追及しない。そして細部まで見ないと分からない一致性や差異は網羅できなくてもやむを得ないものとする。自部門の開発が標準化されていない、開発標準が形骸化して実態は標準化されていない、などの状況においては、ソフトウェアエンジニアリングの初歩から取り組むことを推奨する。

差異として見出されることが多いと想定されるのが以下の2点である。いずれも試行として取り組むことが望ましい。

- ◆ 「ビジネスあるいはミッションの分析」のプロセスに対応する内容が自社の開発標準にない

このプロセスは、2015年に発行されたISO/IEC/IEEE 15288のシステムライフサイクルプロセスの改版において追加されたことから、未対応の組織が多いと思われる。

- ◆ 上流設計の段階における妥当性確認をその開発標準の中で明文化されていない

専門性の高い決まった対象を開発している限りにおいては検証と専門性の高いメンバーの暗黙知でカバーできることが多く、あえて明文化されていないものと思われる。

## (5) 理解の深まりと目的の再設定

3.2.2 (4) を実施した結果、適用する作業の範囲、関与する人などを増やそうという検討が可能になる。例えば、上流工程における設計書の妥当性確認の実施を行うことが考えられる。

対象： ソフトウェア開発プロジェクトで主に設計、開発、検証に関わる人

内容： 設計書の妥当性確認レビューを行う。

期待効果： 本来の目的と設計が乖離することを防止できる。（問題の早期発見）

## (6) 本格適用に向けた課題の特定と対策

(4)②で照合して明らかになった差異を、自部門が新たに直面している環境を踏まえて再考し、改善すべきことを洗い出し、自ら課題として抽出し、取り組んでいくことが望ましい。

### 3.2.3 現場適用に先立つパイロット・プロジェクトを実施した場合

#### (1) 事業ニーズと導入動機の整合の整理

本ケースは金融、流通、公共などの顧客を対象とした情報システム開発を主に行ってきたシステム開発企業（SIer：A社）での例をもとにしたものである。

A社では、今まで行ってきたシステム開発は、ひとつの業界の一顧客を対象として、ノウハウを多く持つIT領域のシステム開発を行ってきた。しかし、今後のシステム開発ではこの前提条件が崩れ、複数業界・複数企業に関係し、ノウハウを持たない複数の専門領域にまたがるシステムを開発することが想定された。特に、IoTを組み入れたシステムの開発に取り組む際には、従来の情報システム開発の手法をそのまま適用したのではうまくいかないという事業ニーズが出てくることが予想された。このことが、新たなシステム開発に対応可能なシステムズエンジニアリングの研究開始と適用の動機となった。

#### (2) 現状プロセスの評価と導入対象の特定

前述したように、複数の専門領域にまたがるシステム開発を有効に進めるには、ステークホルダの間のコミュニケーションを促進し、変更の影響分析をし易くすることができる要求・仕様のモデル化による可視化が有効と考え、試行の活動を通してモデル化を研究対象とすることとした。

モデル化においては、抽象化の視点を柔軟に設定し、多視点から対象を構造化し、システムに関する様々なネットワークを通じて、システムを明らかにする。また、モデルを利用することによって異なる分野の人たちの間での概念共有、情報共有による共通理解の促進を図ることが重要である。そのためモデルには、

- ① ステークホルダが共通に理解できること
- ② 要求・仕様の多様な側面を捉えられること
- ③ 要求・仕様の要素間をトレースできること

が必要である。

このモデル化のためには、いわゆるポンチ絵による情報共有から、専用の言語・手法を使った情報共有まで、種々の方策があり、その手法の是非によっ

て、システムズエンジニアリングの効果が変わってくることが考えられる。今回の試行においては、標準化団体である OMG がシステムモデリング言語の標準として公開しているモデル言語である SysML で、モデル記述を試行することとした。

なお、先に述べたように、モデル化を試行する上で適度な複雑さと、そのドメイン知識の獲得可能環境のふたつを同時に満たせる、対象とするシステムを決める必要がある。適切な導入対象がないか探したところ、A 社のメンバーが、「農園モニタリングシステム」を題材として B 大学と共同研究をしていることが分かった。「農園モニタリングシステム」とは、カメラで農作物を自動撮影することにより、農作物の生育状況を観察できるようにするシステムである。このシステムは、すでに B 大学 C 教授が開発して試行しているので、ドメイン知識は C 教授から取得できる。さらに、研究メンバーには、栽培ドメインになじみがないメンバーも含まれていたことに加え、ソフトウェア系とハードウェア系のメンバーが混在していた。そのため、モデルの要件「①ステークホルダが共通に理解できること」が満たされるのかを評価する上でも、適しているメンバー構成であり、この題材を試行することとした。

### (3) 教育および支援の獲得

SysML によるモデル化の知識の入手においては、主に書籍「システムズモデリング言語 SysML」 [11]を参考にした。

SysML はモデル言語なので、これに加えて、どのようなアクティビティでどのようなモデルを作成するのかを規定したプロセスが必要となる。上記の書籍は、OOSEM (Object Oriented Systems Engineering Method) というシステムズエンジニアリングに準拠したモデリングのプロセスを提示している。SysML を採用した別のプロセスを解説した他の書籍もあったが、OOSEM はシステムズエンジニアリングを提唱している INCOSE が推奨しているため、採用することにした。

その他、システムズエンジニアリング、モデリングを対象とした社外のいくつかの研究グループに参加し、最新情報の取得、意見交換を通して見識の取得に努めた。

#### (4) 限定適用とその効果の確認

モデル化を試行するにあたり、まず、農園および栽培ドメインの要求に関する知識を獲得する必要があるため、Web上の情報から農園・栽培ドメインの基本知識とともに、現在IT・IoTが当ドメインにおいてどのように活用されているか、あるいは活用されようとしているか調査した。調査を進めつつ、農園ビジネス要求、栽培ドメインにおける現状運用の理解と問題点の抽出を進めた。そして、問題を解決するためのシステムに求められるミッションを定義した上で、今後の栽培ドメインのあるべき姿を、コンテキストを含めて表し、その中で対象とする「栽培支援システム」の位置づけを明確化した。そして、ミッションを果たすためにコンテキストとシステムがどのような場面でのどのようにやりとりするのかをシナリオとして整理することにより、システム要求を抽出した。

#### (5) 理解の深まりと目的の再設定

その上で、C教授にヒアリングすることで、実際の農園における要求として「作業者が働いている様子を把握したい」、「作業を間違え易いところにアラートを出したい」などを追加し、要求モデルをもとに研究メンバーの中で検討することにより要求モデルを洗練した。最終的にはIoTを活用したシステム要求として、「栽培作業を把握する」、「栽培作業の誤りを通知する」などを抽出できた。また、これらのシステム要求を満たすためのシステムの振る舞いを考えると、作業者が「栽培作業を実施する」際にシステムが「栽培作業を把握する」、「栽培作業の誤りを通知する」ことが明確になるため、圃場にいる作業者とシステムの間やりとりが発生することが明確になった。これらの点を踏まえてそもそものビジネス要求に立ち戻って確認することで、さらに「耕地面積を増やす」という要求にも気がついた。このビジネス要求により、今後作業者が作業する圃場が拡大する可能性がある。そのような際にも、作業者とシステムとのやりとりができる状態にする必要があることから、「耕地面積の拡大にも対応できるようにする」という非機能要求を抽出した。

## (6) 本格適用に向けた課題の特定と対策

本格適用に先立って、まず、モデリングの方法についての課題を抽出するため、今回の試行が、3.2.3(2)で述べたモデルの要件①、②（③を除く）を満たしているかを評価した（○：良い点、×：考慮が必要な点）。

- ① モデルの要件「ステークホルダが共通に理解できること」は満たされるか
- ◆ 今回の研究メンバーのように、SysMLのセマンティックやモデル化の知識を持つ技術者（機械系、組み込み系など）であれば、要求を共通に理解する上で効果がある（○）。
  - ◆ 一方で、モデル化に関する前提知識を持たない業務のステークホルダ（今回の場合には農園のオーナー）が理解するには難しい。そのようなステークホルダ向けには別途ポンチ絵などを示す必要がある（図3.2.1参照）。

### 試行結果—IoT適用のアイデア（ご参考）

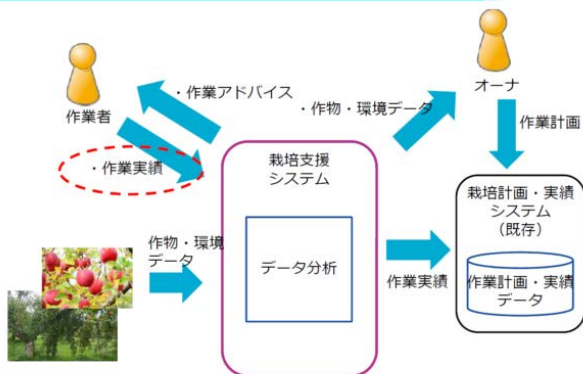


図 3.2.1 情報共有に使用したポンチ絵の例

- ② モデルの要件「要求・仕様の多様な側面を捉えられること」は満たされるか
- ◆ SysMLでは要求が様々な視点でモデルによって可視化されているため、上位レイヤーの要求に照らして、下位レイヤーの要求が妥当であるか確認しながら要求・仕様を定義できる。また、下位レイヤーからのフィードバックが容易である（○）。

- ◆ モデルの種類が多いため、モデル間の関係性を示す工夫が必要である。Excel などの表形式でまとめた方が良い成果物もある（×）。

本ケースは、システム要求の明確化の部分に特化しているため、本格適用の段階にはまだ至っていないが、それに向かって考慮すべき点を以下に列挙する。

- ◆ モデル間の関係性を示すには、影響分析を行うために有効であるトレーサビリティの確保が必要である。どのモデルあるいはモデル要素間を関連づければより効果的か、どのような仕組みにするとメンテナンスし易いかをさらに検討する必要がある。
- ◆ 今回は、SysML を用いてモデル化をするにあたり、学習教材が上記で述べた 1 冊の書籍しかなかったため、事例がひとつしかなく、また、理解しづらいところがあったため、学習に時間がかかった。より多くの人々が、平易に理解できるようにガイド類の整備が必要である。
- ◆ 今回は、ひとつおりの要求モデルを作成するところまでを試行したに過ぎないため、モデルの評価、アーキテクチャモデリング、フィードバックによる反復などを検討・試行する必要がある。
- ◆ 標準のモデル表記法としては、同じ OMG が規定している UML が有名であり、システムズエンジニアリングを行う際のモデル言語として評価が必要である。UML を採用した場合には、要求と制約（パラメトリック）を表すモデル図がないため、別途検討が必要である。また、モデリングプロセスは必要であり、その習得が必要となる点は、SysML と同様である。ただし、SysML に比べて UML を使うことができる技術者は多いため、導入し易くなる可能性がある。
- ◆ 加えて、専門家以外にも理解が進むよう、ポンチ絵や、Excel の併用方法も検討の余地がある。

### 3.3 まとめ

本章では、システムズエンジニアリングを問題解決に役立てようと思った人が第一歩を踏み出すための手助けとして、まず、先人がとったであろう行動の考え方（どこから始めるのか、どのように進めるのか）を紹介した。次にふたつの実践ケースを紹介し、その考察からヒントを整理した。以下に留意頂きたい点を付け加えておく。

まず、第一歩を踏み出すにあたって、どこから始めるのかは、対象とする課題の内容や、実行する体制によって異なってくるため、ケースバイケースで考える必要がある。図3.1.1を参考にしながら、従来の問題解決の進め方で足りない点を検討することをお勧めする。

また、どのように進めるのかについて、ここで紹介した考え方は、システムズエンジニアリングに特有ではなく、他の手法を業務に導入する場合にも一般的に言えることであり、以下の点にも留意が必要である。

例えば、本書を読んで、システムズエンジニアリングを有効な問題解決手段と認識し、目前の問題に対する結果が出た時点で達成感を得て、その後の本格適用に興味を失うことがある。一方、上司からシステムズエンジニアリングに取り組みと指示を受けた場合などに、試行を進めること自体が目的となり、問題解決の結果を出すことから離れ、形骸化する場合もある。いずれの場合も、システムズエンジニアリングの考え方をビジネスに有効に取り入れるという観点からは好ましくない。

本章での始め方、進め方がヒントになれば幸いである。

## コラム：八百屋をシステムと見なすと

八百屋をシステムと見なして、システムズエンジニアリングの考え方に基づいて、ミッションや要求事項を分析し、設計や実装を行ったらどうなるか、試行的に検討してみた。システムズエンジニアリングのアプローチを直感的に理解して頂くために、身近な例をシステムライフサイクルの上流プロセスに沿って説明してみる。なお、プロセス名および番号は ISO/IEC/IEEE15288:2015 に合わせている。(特にここで対象とするプロセスは 4.1 節 表 4.1.1 の下線部参照。ただし、6.4.5 は当コラムでの概説のみ。)



### 6.4.1 ビジネスあるいはミッションの分析プロセス

(Business or mission analysis process)

八百屋というシステムを考えるにあたり、八百屋の商売とはそもそも何なのか？を問うことからスタートする。「野菜を売っている」には違いないが、八百屋というシステムのコンテキストを考えてみると、最も重要なステークホルダについて、

- ◆ 買い物客
- ◆ 野菜が買いたい人

と捉えるのは、いささか単純過ぎる。例えば、ある住宅街の八百屋を想定すると、

- ◆ 今晚のおかずは何にしようか？どの材料を買おうか？と悩んでいる人と捉えることができる。しかも、このような八百屋では、買い物客が週に何度も訪れる。したがってこの想定の場合、八百屋というシステムの真の目的は



- ◆ 「今晚のおかずをどうするか困っている人」に対して、1週間を通じて有効なおかずの材料（野菜）を提供すること

と定義できる。

このゴールイメージから、八百屋のライフサイクル全般へと考えを広げ、各ライフサイクルステージですべきことを考え、時間的に俯瞰した八百屋というシステムのゴールを設定する。

こうした検討を進めながら

◆ 八百屋というシステムが『妥当である』若しくは『十分である』とは？という問いの答えを決めていくことになる。

- ◆ 「今晚のおかずをどうするか困っている人」を助けることができた程度（例えば、何人、何回/週、など）を観測・測定する

◆ 同じ人が繰り返し買いに来てくれたかどうかを観測・測定するなど、システムの有効性を知るための定性・定量的な指標（MOE: Measures of Effectiveness）を検討し、その候補やアイデアを抽出する。この時点では、システムの外から見て検討することが重要である。

#### 6.4.2 利害関係者ニーズと要求事項の定義プロセス

##### (Stakeholder needs and requirements definition process)

先のゴール設定の過程と結果から、ライフサイクルを通じて、八百屋というシステムが、どのようなコンテキストにおいて、何をなすべきか徐々に明らかになってくる。ここから各ライフサイクルステージを想像し、コンテキストとシステム（八百屋）の相互作用を明確化し、様々なステークホルダがシステムに何を求めているのかを洗い出す。八百屋というシステムでは、前述の通り、

◆ 今晚のおかずは何にしようか？どの材料を買おうか？と悩んでいる人の中でも、1週間に何度も買い物に来てくれるステークホルダを重視する必要がある。

このステークホルダを想像すると

- ◆ できるだけ晩御飯のメニューで悩みたくない
- ◆ 美味しいものが食べたい

というニーズを抽出できる。さらに、これらステークホルダニーズを分析、解釈すると、

- ◆ 日替わりのおかずのヒントが欲しい

◆ 旬の食材を使い切り分量だけ手に入れたい

など重要なステーキホルダが、

◆ 八百屋というシステムに期待する事柄（＝ステーキホルダの要求）を定義することができる。なお、ステーキホルダには

◆ 八百屋のオーナー

◆ 仕入先

なども含まれる。一般的にユーザーと呼ばれるステーキホルダばかり意識しがちだが、俯瞰的にステーキホルダを識別することが重要である。

並行して、システムの有効性を知るための定性・定量的な指標を具体化する。コンテキストやステーキホルダの観点から、八百屋というシステムがどうなれば『妥当である』若しくは『十分である』と言えるか考えて、

◆ システムが役割を十分に果たした状況で観察・測定されるべき事柄を定義する。例えば、

◆ 「今晚のおかずをどうするか困っている人」の集団を追跡し、「困っている度合い」および「解消度合い」を買い物の前後に主観的評価で取得する

といった指標を定義し、システムの有効性を統計的に評価することができる。

- ① ゴール設定
- ② ライフサイクルステージ設定
- ③ コンテキストとの相互作用の明確化
- ④ ステーキホルダニーズと要求事項の定義
- ⑤ システムの有効性を知るための定性・定量的な指標の定義

といった一連の活動において、現状の観察、関係者インタビュー、データ分析、専門家との議論など、適切な活動を踏んで、より『妥当』な定義が可能になる。ただし、これは多くのシステムにおいて難しい。

### 6.4.3 システム要求事項の定義プロセス

#### (System requirement definition process)

上記①～⑤の活動に基づいて、八百屋というシステムに求められることを、さらに具体的に検討していく。ここまでは、コンテキストやステーキホルダ側から見ていたが、ここで初めて、システム側が何を提供すべきかという観点

で検討する。これまでの分析結果に基づき、システムに求めるべき事柄（システム要求）を定義する。例えば

- [A] 曜日ごとに異なったおかず 1 品が作れるように、商品（野菜および果物）とそれを用いた 2 種類のレシピを提示する
- [B] 入店するとすぐに、その日の旬の食材が何であるか視覚的に分かるようにする

などがシステム要求になる。すべてのシステム要求の根拠は、上記①～⑤の結果に紐付けられることが望ましい。紐付きが明確にならない場合は、明確にならない理由を分析して記録しておくが良い。

#### 6.4.4 アーキテクチャの定義プロセス (Architecture definition process)

上記のシステム要求をもとに、八百屋というシステムのアーキテクチャを定義（設計）していく。システムのアーキテクチャを定義するとは、システムの「振る舞い（機能）」と「実現手段（物理）」が定義され、それぞれに割り付けられていることであり、結果としてサブシステムが明確に定義され、サブシステム間のインタフェースが明らかになっていることを指す。

例えば、システム要求[A]に対するシステムの振る舞いには、

- (ア) 曜日ごとに商品（野菜および果物）を提示
- (イ) 商品を用いたレシピ（ひとつ目）を伝達
- (ウ) 商品を用いたレシピ（ふたつ目）を伝達

がある。これらの実現手段の割り付けは、何通りも検討できる。すべての振る舞いを「店員（口頭）」というサブシステムに割り付けても良いし、「商品ポップ」「オンライン・レシピ」「店員（口頭）」にそれぞれを割り付けても良い。これまでの要求分析の結果を踏まえ、様々なアーキテクチャを設計し、比較検討してアーキテクチャ定義の意思決定を行う。「店員（口頭）」に頼ったアーキテクチャは、人材が確保できれば安定的に運用可能である。「オンライン・レシピ」を用いるアーキテクチャは、初期投資が膨らむが将来の発展性が期待できる。このようにアーキテクチャによってシステムの特性や特徴が大きく異なってくる。

#### 6.4.5 設計定義プロセス (Design definition process)

システムのアーキテクチャ定義に基づいて、システムの具体的な設計を進める。アーキテクチャ定義によって、各サブシステムが何の目的でどのように振る舞い、どのように相互作用するのか分かる。この情報を用いて各サブシステムを実際に開発、調達、製造するために、各部の設計を具体的に詰めていく。

#### 6.4.6 システム解析プロセス (System analysis process)

6.4.1 から 6.4.5 における諸アクティビティの成果を評価する記述に対応する解析を、6.4.9 と 6.4.11 における検証および妥当性確認とともに実施することになる。

#### 6.4.9 検証プロセス (Verification process)

##### ／6.4.11 妥当性確認プロセス (Validation process)

このようにして作られたシステムが、正しく働くか、また働いた結果そもそもの目的を果たせるのか、ふたつの観点で確認する必要がある。

まず、「システムが正しく働くか」を確認するとは、システムに求めるべき事柄 (システム要求) が満たされることを確かめる。八百屋というシステムでは、「商品ポップ」「オンライン・レシピ」「店員 (口頭)」の3つのサブシステムが、振る舞い (ア) (イ) (ウ) を間違いなく実施して、システム要求 [A] が満たされていることを確かめる。

さらに、システムは正しく働いた上で、「そもそもの目的を果たせるのか」を確かめる。すなわち、コンテキストやステークホルダの観点から考えて、八百屋というシステムが『妥当である』若しくは『十分である』と言えるか確かめる。八百屋というシステムが、

- ◆ 今晚のおかずは何にしようか？ どの材料を買おうか？ と悩んでいる人の
- ◆ できるだけ晩御飯のメニューで悩みたくない
- ◆ 美味しいものが食べたい

というニーズを十分に解消あるいは達成できたかどうか、確かめることになる。

## 第4章 システムズエンジニアリング のライフサイクルプロセスの紹介

システムズエンジニアリングを実践するプロセスは、ISO/IEC/IEEE 15288 System life cycle processes として長年にわたり ISO/IEC と IEEE において議論され改訂されてきている。現行の国際規格は、ISO/IEC/IEEE 15288:2015 (以降、発行年を追記し、名称省略) [6]である。2018年3月の時点では、現行の国際規格の邦訳冊子も JIS も提供されていない。この状況で、本章は、一般財団法人日本規格協会の許諾を得て、ISO/IEC/IEEE 15288:2015 のうち、システムズエンジニアリング導入効果を確認するにシステムライフサイクルの上流を従来以上に考慮することが有効とされているので、上流プロセスの抄訳として、提供するものである。

なお、現行の「JIS X0170:2013 システムライフサイクルプロセス」 [12] (以降、JIS X0170:2013 と省略)は、ISO/IEC 15288 IEEE Std 15288-2008 を日本語化したものである。4.1において、現行 JIS と現行国際規格の大凡の比較を表で示す。4.4以降の抄訳において、JIS X0170:2013 と ISO/IEC/IEEE 15288:2015 の英文が同一文面の箇所は、JIS を引用し、網掛けで示した。網掛け以外の部分は、ISO/IEC/IEEE 15288:2015 で新規あるいは修正されたもので、仮訳である。

## 4.1 システムライフサイクルプロセスの全容

ISO/IEC/IEEE 15288:2015 は、本書で紹介するプロセスのみならず、「ISO/IEC/IEEE 15288:2015 の図4 システムライフサイクルプロセス」 [6]にすべてのプロセスを紹介している。いくつかのプロセスをくくったものもプロセスと称している。表 4.1.1 において、下線で示すプロセスを 4.4 で紹介する。

表 4.1.1

JIS X 0170:2013 と ISO/IEC/IEEE 15288:2015 システムライフサイクルプロセスの比較

JIS X0170:2013	ISO/IEC/IEEE 15288:2015	主変更
6.1 合意プロセス	6.1 合意プロセス	
6.1.1 取得プロセス	6.1.1 取得プロセス	
6.1.2 供給プロセス	6.1.2 供給プロセス	
6.2 組織のプロジェクトイネー ブリングプロセス	6.2 組織のプロジェクトイネー ブリングプロセス	
6.2.1 ライフサイクルモデル管 理プロセス	6.2.1 ライフサイクルモデル 管理プロセス	
6.2.2 インフラストラクチャ 管理プロセス	6.2.2 インフラストラクチャ 管理プロセス	
6.2.3 プロジェクトポートフ ォリオ管理プロセス	6.2.3 ポートフォリオ管理プ ロセス	タイトル
6.2.4 人的資源管理プロセス	6.2.4 人的資源管理プロセス	
6.2.5 品質管理プロセス	6.2.5 品質管理プロセス	
	6.2.6 知識管理プロセス	新規
6.3 プロジェクトプロセス	6.3 テクニカル管理プロセス	タイトル
6.3.1 プロジェクト計画プロ セス	6.3.1 プロジェクト計画プロ セス	
6.3.2 プロジェクトアッセメ ント及び制御プロセス	6.3.2 プロジェクトアッセメ ント及び制御プロセス	
6.3.3 意思決定管理プロセス	6.3.3 意思決定管理プロセス	

6.3.4 リスク管理プロセス	6.3.4 リスク管理プロセス	
6.3.5 構成管理プロセス	6.3.5 構成管理プロセス	
6.3.6 情報管理プロセス	6.3.6 情報管理プロセス	
6.3.7 測定プロセス	6.3.7 測定プロセス	
	6.3.8 品質保証プロセス	新規
6.4 テクニカルプロセス	6.4 テクニカルプロセス	
	<u>6.4.1 ビジネスあるいはミッションの分析プロセス</u>	新規
6.4.1 利害関係者要求事項定義プロセス	<u>6.4.2 利害関係者ニーズと要求事項の定義プロセス</u>	タイトル、番号
6.4.2 要求事項分析プロセス	<u>6.4.3 システム要求事項の定義プロセス</u>	同上
6.4.3 方式設計プロセス	<u>6.4.4 アーキテクチャの定義プロセス</u>	同上
	6.4.5 設計定義プロセス	新規
	<u>6.4.6 システム解析プロセス</u>	新規
6.4.4 実装プロセス	6.4.7 実装プロセス	番号
6.4.5 結合プロセス	6.4.8 結合プロセス	同上
6.4.6 検証プロセス	<u>6.4.9 検証プロセス</u>	同上
6.4.7 移行プロセス	6.4.10 移行プロセス	同上
6.4.8 妥当性確認プロセス	<u>6.4.11 妥当性確認プロセス</u>	同上
6.4.9 運用プロセス	6.4.12 運用プロセス	同上
6.4.10 保守プロセス	6.4.13 保守プロセス	同上
6.4.11 廃棄プロセス	6.4.14 廃棄プロセス	同上

## 4.2 ISO/IEC/IEEE 15288:2015 システムライフサイクルプロセスの概要

ISO/IEC/IEEE 15288:2015 はその「1. 概要」で以下の項目を説明している

- 1 概要
- 1.1 適用範囲(Scope)
- 1.2 目的(Purpose)
- 1.3 適用分野(Field of application)
- 1.4 制限(Limitation)

1.1 と 1.2 について以下に紹介する

### 1.1 適用範囲(Scope)

この規格は、人が作り出すシステムのライフサイクルを記述するための共通の枠組みを確立する。この規格は、ひとそろいのプロセスおよび関連する用語を定義<sup>1</sup>する。これらのプロセスは、システムの階層構造のどのレベルにも適用可能である。これらのプロセスから選定された集合は、システムのライフサイクルの段階を管理、実行するために、ライフサイクルを通じて、適用することができる。これは、顧客満足を達成するという最終目標を持つ、対象となるすべての当事者<sup>2</sup>の参加を通して達成される。

この規格は、組織内またはプロジェクト内で使用するライフサイクルプロセスの定義、制御および改善を支援するプロセスも規定する。組織およびプロジェクトは、システムを取得および供給する場合にこれらのライフサイクルプロセスを使用することができる。

この規格は、人が作ったシステム、すなわち、次に示す一つ以上のもので構成されるシステムに関するものである。それは、ハードウェア、ソフトウェア、データ、人、プロセス(例えば、利用者にサービスを提供するプロセス)、手順(例えば、操作指示書)、設備、原材料および自然に発生する実体を指す。

<sup>1</sup> ISO/IEC/IEEE 15288:2015 では、「エンジニアリングの観点から」が「定義」の前に追記されている。

<sup>2</sup> 「当事者」という用語の英文は、ISO/IEC/IEEE 15288:2015 では stakeholder で、ISO/IEC 15288:2008 では interested parties。ここは JIS からの引用なので「当事者」を採用している。



システム要素がソフトウェアである場合は、そのシステム要素を実装するのに、JIS X 0160<sup>3</sup>で規定するソフトウェアライフサイクルプロセスを使用してもよい。この規格と JIS X 0160 とは、単一のプロジェクトまたは単一の組織で併用できるように調和が取られている。システム要素がハードウェアである場合は、該当する JIS または 国際規格を参照。

## 1.2 目的

この規格の目的は、システムのライフサイクルにおける、取得者、供給者および他の利害関係者の間で円滑に情報伝達を行う場合に必要な定義されたプロセスの集合を提供することである。

この規格は、組織が取得者の役割を行う場合でも、供給者の役割を行う場合でも適用される。単一の組織が、自主的に計画し開発する状況または複数の当事者で計画し開発する状況でも、使用することができる。当事者は、同じ組織または異なる組織であってもよく、この状況は、非公式の合意から公式の契約にまで及ぶ。

この規格の中のプロセスは、例えば、方法、手順、手法、道具および訓練された要員といった、業務環境を確立するための基礎として使用することができる。附属書 A は、これらのシステムライフサイクルプロセスの修整の仕方に関する基準を示す。

<sup>3</sup> 「JIS X 0160」は、ISO/IEC/IEEE 12207 に基づく。ISO/IEC/IEEE 12207 の最新版は、ISO/IEC/IEEE 12207:2017 [13]であるが、最新版 JIS 規格 は、ISO/IEC 12207:2008 [14] に基づく JIS X 0160:2012 [15]であり、日本で広く引用されている「共通フレーム 2013」 [16]は、JIS X 0160:2012 に基づいて作成されている。

## 4.3 システムライフサイクルプロセスの中で効果的とされるプロセス

実際のシステムズエンジニアリングを駆使したプロジェクトにおいて、4.1節で示す全プロセスの中でどのプロセスが他より効果的かについて、カーネギーメロン大学ソフトウェア工学研究所(SEI)により調査研究がなされた。調査研究をリードしたSEIのJoseph P. Elmは、その報告をPMI、INCOSE等の会議で発表しており、インターネットを通して入手可能である。詳しくは当該資料[17]を参照頂きたい。

結論として、「ライフサイクルの早い段階にシステムズエンジニアリングを導入することが最も効果的である」とし、以下のプロセスを指摘している。

- ◆ ユーザー要求の妥当性確認と運用概念
- ◆ システム要求とアーキテクチャ

SEIにおけるプロセス定義とISO/IEC/IEEE 15288:2015のプロセス定義は同一ではないので、上述プロセスは、結論として述べているようにシステムライフサイクルの早い段階で関係するプロセスを示していると読み取って頂きたい。この早い段階に関係するプロセスは、図4.3.1において黄色で示すプロセスである。

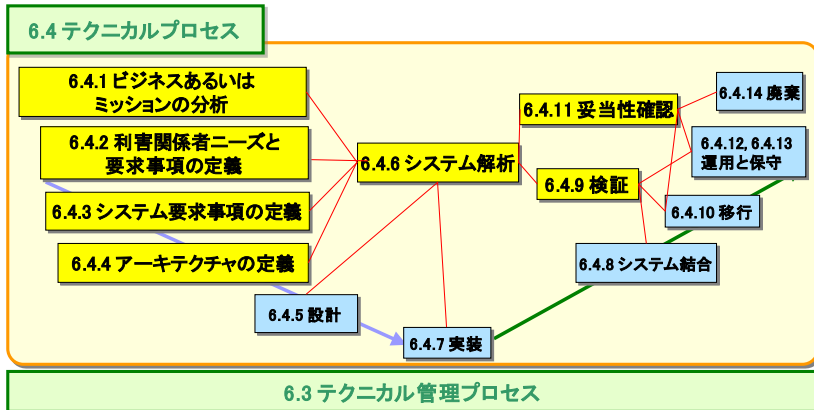


図4.3.1 システムライフサイクルプロセスの中で効果的とされるプロセス

## 4.4 ISO/IEC/IEEE 15288:2015 における 上流プロセスの紹介

プロセスは、以下の項目で定義される。ここでは、抄訳で上流プロセスを紹介する。現行 JIS から引用する部分は網掛けで示す。網掛けで示す JIS からの引用部分が少ないことは、4.1 節の表 4.1.1 で示す「主変更」以上に、現行国際規格の内容が修正・追加されていることを意味し、よって、詳しくは現行国際規格の本文を参照されたい。

- ◆ プロセス名
- ◆ 目的：当該プロセスの目的
- ◆ 成果：プロセスを実施したときに認識される状態
- ◆ アクティビティとタスク：プロセスを実行するというものの具体的な活動項目

### 6.4.1 ビジネスあるいはミッションの分析プロセス (Business or mission analysis process)

目的：

ビジネスあるいはミッションの問題あるいは機会を定義し、解決策の範囲を特徴付け、問題に対応するとともに機会を活かす解決案を決める。

成果：

- a) 問題あるいは機会が定義されている。
- b) 解空間は特徴付けられている。
- c) 予備的なシステム運用概念やライフサイクルステージにおける他の概念は特定されている。
- d) 候補となる代替案は特定され、解析されている。
- e) 望ましい代替案が選択されている。
- f) ビジネスあるいはミッション分析に必要なイネープリングシステムあるいはサービスが利用可能である。
- g) ビジネスあるいはミッション解析のトレーサビリティは確立されている。

**アクティビティとタスク：**

- a) ビジネスあるいはミッション分析の準備をする。
  - 1) 要望されている組織目標と目的に関連する組織戦略における特定された問題と機会をレビューする。
  - 2) ビジネスあるいはミッション分析戦略を定義する。
  - 3) 必要なイネープリングシステムあるいはサービスを識別する。
  - 4) イネープリングシステムあるいはサービスを使用可能にする。
- b) 問題あるいは機会の範囲を定義する。
  - 1) 関連するトレードオフ分析の文脈から問題と機会を分析する。
  - 2) ミッション、ビジネスあるいは問題あるいは機会を定義する。
- c) ソリューションの範囲を特定する。
  - 1) 予備的なシステム運用概念やライフサイクルにおける他の概念を定義する。
  - 2) 可能解の中で候補となる代替案を特定する。
- d) 代替案を評価する。
  - 1) 各代替案を評価する。
  - 2) 望ましい代替案を選択する。
- e) ビジネスあるいはミッション分析を管理する。
  - 1) ビジネスあるいはミッション分析のトレーサビリティを維持する。
  - 2) ベースラインとして選択された主要成果記録を整備する。

## 6.4.2 利害関係者ニーズと要求事項の定義プロセス (Stakeholder needs and requirements definition process)

**目的：**

定義された環境におけるユーザーおよび他の利害関係者に必要とされる能力を提供するシステムの要求事項を定義する。

**成果：**

- a) システムの利害関係者は識別されている。

- b) サービスで要求される特性およびサービスで要求される利用の文脈(内容, 状況および背景)が明示されている。
- c) システムソリューションにおける制約が定義されている
- d) 利害関係者ニーズは定義されている。
- e) 利害関係者ニーズは優先付けられ定義された利害関係者要求に変換されている。
- f) 重要なパフォーマンス測定値は定義されている。
- g) 利害関係者ニーズと期待が適切に反映されているという合意が得られている。
- h) 利害関係者ニーズと要求のためのイネープリングシステムとサービスが入手可能である。
- i) 利害関係者要求事項から, 利害関係者および利害関係者ニーズへの追跡可能性が樹立されている。

#### アクティビティとタスク :

- a) 利害関係者ニーズと要求事項の定義の準備をする。
  - 1) ライフサイクルを通して関心を持つ利害関係者を識別する。
  - 2) 利害関係者ニーズと要求事項を明確化する戦略を定義する。
  - 3) 必要なイネープリングシステムとサービスを識別する。
  - 4) イネープリングシステムとサービスを利用できるようにする。
- b) 利害関係者ニーズを定義する。
  - 1) 使用者のシステム運用概念と予備的なライフサイクル概念の範囲で使用する内容を定義する。
  - 2) 利害関係者ニーズを識別する。
  - 3) ニーズに優先順位を付ける。
  - 4) 利害関係者ニーズと論拠を定義する。
- c) 運用概念および他のライフサイクルの考え方を定義する。
  - 1) 予想され運用および他のライフサイクル概念に対応する要求される能力を識別するための代表的なシナリオを定義する。
  - 2) ユーザーとシステムの相互作用を識別する。
- d) 利害関係者ニーズを利害関係者要求に変換する。
  - 1) システム解における制約事項を識別する。

- 2) 重要な品質特性、例えば、保証、安全、セキュリティ、環境、あるいは健康、に関連する利害関係者要求と機能を識別する。
  - 3) ライフサイクル概念、シナリオ、相互作用、制約、重要な品質要求と整合性ある利害関係者要求を定義する。
- e) 利害関係者要求を分析する。
- 1) 利害関係者要求一式を分析する。
  - 2) 技術上の達成を評価できる重要なパフォーマンス測定値を定義する。
  - 3) 要求を分析した結果を要求者にフィードバックし、ニーズと期待値を適切に反映していることの妥当性確認する。
  - 4) 利害関係者課題を解決する。
- f) 利害関係者ニーズと要求事項の定義を管理する。
- 1) 利害関係者要求の細部にわたって合意を得る。
  - 2) 利害関係者ニーズと要求のトレーサビリティを維持する。
  - 3) ベースラインとして選択された主要成果記録を準備する。

### 6.4.3 システム要求事項の定義プロセス (System requirement definition process)

#### 目的：

利害関係者、ユーザー指向で要望される能力をユーザーの運用ニーズに合致する解決策の技術観点に変換する。

#### 成果：

- a) システム解として、インタフェース、機能と境界を含むシステム記述がされている。
- b) システム要求事項(機能、パフォーマンス、プロセス、非機能およびインタフェース)と設計の制約は定義されている。
- c) 重要なパフォーマンス測定値は定義されている。
- d) システム要求事項は分析されている。
- e) イネープリングシステムおよびサービスは利用可能である。
- f) システム要求事項の利害関係者要求へのトレーサビリティは開発されている。

### アクティビティとタスク：

- a) システム要求事項の定義のための準備をする。
  - 1) 提供される振る舞いおよび特性の観点からシステムの機能境界を定義する。
  - 2) システム要求事項を定義する戦略を定義する。
  - 3) システム要求事項を定義することを支援するために必要なイネープリングシステムとサービスを識別し入手の計画をする。
  - 4) 使用するイネープリングシステムあるいはサービスを入手するか利用可能にする。
- b) システム要求事項の定義
  - 1) システムが実行を要求されている各機能を定義する。
  - 2) 必要な実装制約条件を定義する。
  - 3) システムのリスク、システムの臨界性、あるいは重要な品質特性に関連する要求事項および機能を明示する。
  - 4) システム要求事項および論拠を定義する。
- c) システム要求事項を分析する。
  - 1) システム要求事項一式を分析する。
  - 2) 技術上の達成を評価するため測定値を定義する。
  - 3) システム要求事項を分析し関係する利害関係者にフィードバックしレビューを受ける。
  - 4) システム要求事項の課題事項を解決する。
- d) システム要求事項を管理する。
  - 1) システム要求事項に関する細部の合意を得る。
  - 2) システム要求事項のトレーサビリティを維持する。
  - 3) ベースラインとして選択された主要な成果記録を準備する。

## 6.4.4 アーキテクチャの定義プロセス (Architecture definition process)

### 目的：

システムアーキテクチャの代替案を生成し、利害関係者の関心事をカバーす

る一つあるいはそれ以上の代替案を選択し、これを整合性ある観点で表現する。

#### 成果：

- a) 利害関係者関心事はアーキテクチャにより対応されている。
- b) アーキテクチャビューポイントは展開されている。
- c) システムのコンテキスト、境界および外部とのインタフェースは明確にされている。
- d) システムのアーキテクチャビューおよびモデルが展開されている。
- e) システムのアーキテクチャを決定する概念、特性、特質、振る舞い、機能あるいは制約条件はアーキテクチャ要素に配置されている。
- f) システム要素とそれらのインタフェースは特定されている。
- g) アーキテクチャ候補は評価されている。
- h) ライフサイクルを通してプロセスのアーキテクチャ上のベースは達成されている。
- i) 要求と設計特性をアーキテクチャが結び付けられている。
- j) アーキテクチャの定義のために必要なイネープリングシステムあるいはサービスは利用可能である。
- k) アーキテクチャ要素から利害関係者およびシステム要求へのトレーサビリティは展開されている。

#### アクティビティとタスク：

- a) アーキテクチャの明確化のための準備をする。
  - 1) 適切な情報をレビューし、アーキテクチャの主要な要素を識別する。
  - 2) 利害関係者の関心事を識別する。
  - 3) アーキテクチャ定義のロードマップ、アプローチおよび戦略を定義する。
  - 4) 利害関係者の関心および主要な要求事項を基礎とする評価基準を定義する
  - 5) アーキテクチャの定義プロセスを支援するために必要なイネープリングシステムを特定し計画する。



- 6) イネープリングシステムあるいはサービスへのアクセスをするか取得して使用する
- b) アーキテクチャの構成要素を展開する。
  - 1) 利害関係者の関心事に基づく構成要素とモデルの種類を選択、適用あるいは展開する。
  - 2) モデルと観点を展開するために、アーキテクチャフレームワークを確立するか明確化する。
  - 3) フレームワーク、構成要素およびモデルの種別を選択するための根拠を獲得する。
  - 4) 支援するモデリング手法あるいはツールを選択するか開発する
- c) アーキテクチャ候補のモデルと観点を展開する。
  - 1) 外部要素とのインタフェースと相互作用の観点でシステムコンテキストおよび境界を明確化する。
  - 2) アーキテクチャ要素と主要な利害関係者の関心事と重要なシステム要求に対応する要素間の関係とを識別する。
  - 3) システムのアーキテクチャ決定からアーキテクチャ要素の決定に重要な概念、特性、特質、振る舞い、機能あるいは制約条件の割り当てを行う。
  - 4) システムの候補となるアーキテクチャモデルを選択、適用、あるいは開発する。
  - 5) 特定のビューポイントに従ってモデルからビューを作成し、アーキテクチャが利害関係者の関心に対処し、利害関係者およびシステム要求事項を満足していることを表現する。
  - 6) アーキテクチャモデルとビューを相互に調和する。
- d) アーキテクチャと設計を結び付ける。
  - 1) アーキテクチャ要素とこれらの間の関係性に関連するシステム要素を識別する。
  - 2) システム要素間および外部要素とのインタフェースおよび相互関係を明確化する。
  - 3) 要求事項からアーキテクチャ要素およびシステム要素に分割し、整合させ、そして割り当てる。
  - 4) システム構成要素とアーキテクチャの実体を設計特質にマッピング

- グする。
- 5) システム設計と進化のための幻想を定義する
- e) アーキテクチャ候補を評価する。
- 1) アーキテクチャ候補のそれぞれにつき制約と要求に対して評価する。
  - 2) アーキテクチャ候補のそれぞれにつき評価基準に基づき利害関係者の関心事で評価する。
  - 3) 望ましいアーキテクチャを選択し、主要な決定とその理由を補足する。
  - 4) 選択されたアーキテクチャのアーキテクチャ上のベースラインを確立する。
- f) 選択したアーキテクチャを管理する。
- 1) アーキテクチャのガバナンスアプローチを決め、ガバナンスに関連する役割と責任、説明責任と権限を定義する。
  - 2) 利害関係者によるアーキテクチャの受け入れを明確に獲得する。
  - 3) アーキテクチャの実体とそのアーキテクチャ上の特性の一致と完全性を維持する。
  - 4) アーキテクチャモデルとビューの進化を編成し、評価し、および制御する。
  - 5) アーキテクチャ定義と進化の作戦を維持する。
  - 6) アーキテクチャのトレーサビリティを維持する。
  - 7) ベースラインとして選択された主要な情報項目を提供する。

## 6.4.6 システム解析プロセス(System analysis process)

### 目的：

ライフサイクルを通して意思決定を支援する技術上の理解のためのデータと情報の厳密な基礎を提供する。

### 成果：

- a) 必要とされるシステム解析は識別されている。
- b) システム解析の仮定と結果の妥当性は確認されている。

- c) 決定のためにシステム解析結果は準備されている。
- d) システム解析のためのイネーブリングシステムとサービスは利用可能である。
- e) システム解析結果のトレーサビリティは確立されている。

#### アクティビティとタスク：

- a) システム解析の準備をする。
  - 1) システム解析を必要とする問題あるいは質問を識別する。
  - 2) システム解析の利害関係者を識別する。
  - 3) システム解析のスコープ、目的、および忠実さのレベルを決める。
  - 4) システム解析手法を選択する。
  - 5) システム解析戦略を決める。
  - 6) システム解析を支援するイネーブリングシステムあるいはサービスを識別し計画する。
  - 7) 使用するイネーブリングシステムあるいはサービスを利用可能にする。
  - 8) 解析のためのデータおよび入力情報を集める。
- b) システム解析を実施する。
  - 1) 仮定を識別し妥当性確認をする。
  - 2) 要求されるシステム解析を実施するために選択された手法を適用する。
  - 3) 品質と妥当性のために解析結果をレビューする。
  - 4) 結論と推奨を確立する。
  - 5) システム解析の結果を記録する。
- c) システム解析を管理する。
  - 1) システム解析結果のトレーサビリティを維持する。
  - 2) ベースラインとして選択された成果情報項目を準備する。

### 6.4.9 検証プロセス(Verification process)

#### 目的：

検証プロセスの目的は、システムあるいはシステム構成要素が特定された要求

事項と特質を満足していることの客観的な証拠を提供することである。

#### 成果：

- a) 検証の拘束条件で要求、アーキテクチャ、設計に影響を与えるものが特定されている。
- b) 検証に必要とされるイネープリングシステムあるいはサービスは利用可能である。
- c) システムあるいはシステム要素は検証されている。
- d) 是正処置のための情報を提供するデータが報告されている。
- e) 実現された製品がシステム要求事項および方式設計を満足することを示す客観的な証拠が提供されている。
- f) 検証の結果と不具合は特定されている。
- g) 検証されたシステム要素のトレーサビリティが取れている。

#### アクティビティとタスク：

- a) 検証の準備をする。
  - 1) 検証の範囲と対応する検証の活動を識別する。
  - 2) 検証を制限するような制約項目を識別する。
  - 3) 各検証活動に適切な検証手法あるいは技法を選択する。
  - 4) 検証の戦略を定義する。
  - 5) システム要求事項、アーキテクチャあるいは設計に組み込まれている検証の戦略からシステムの制約を識別する。
  - 6) 検証に必要なイネープリングシステムあるいはサービスのうち、必要なものを識別し計画する。
  - 7) 検証に必要なイネープリングシステムあるいはサービスのうち必要なものを入手あるいは利用可能にする。
- b) 検証を実施する
  - 1) 検証の一つあるいはそれ以上のまとめた活動に対応する検証の手順を定義する。
  - 2) 検証の手順を実施する。
- c) 検証の結果を管理する。
  - 1) 検証の結果および遭遇したどんな不具合も記録する。

- 2) 運用上の事故および問題を記録しその解決を追跡する。
- 3) システムあるいはシステム要素が指定された要求に合致していることの同意を利害関係者から得る。
- 4) 検証がされている要素のトレーサビリティを維持する。
- 5) ベースラインとして選択された主要な成果項目を準備する。

### 6.4.11 妥当性確認プロセス(Validation process)

#### 目的：

妥当性確認プロセスは、システムが使用される時に、ビジネスあるいはミッションの目標と利害関係者要求事項を遵守し、意図された運用環境で、意図された利用を達成していることを示す、客観的な証拠を提供することを目的とする。

注)2008 年版では、システムの代わりにサービスという用語を用いているが、2015 年版では「ビジネスあるいはミッションの分析プロセス」が新しく追加されたことにより、下線の記述に置き換えられている。

#### 成果：

- a) 利害関係者要求のための妥当性確認規範は決められている。
- b) 利害関係者に要求されているサービスの入手は確認されている。
- c) 妥当性確認の制約で要求、アーキテクチャあるいは設計に影響するものは識別されている。
- d) システムあるいはシステム要素の妥当性確認はされている。
- e) 妥当性確認に必要とされるイネープリングシステムあるいはサービスは利用可能になっている。
- f) 妥当性確認の結果および不具合は識別されている。
- g) 実現されたシステムあるいはシステム要素が利害関係者ニーズを満足しているとの客観的証拠が準備されている。
- h) 妥当性確認がされている要素のトレーサビリティが確立している。

#### アクティビティとタスク：

- a) 妥当性確認の準備をする
  - 1) 妥当性確認の範囲と対応する妥当性確認の活動を識別する。

- 2) 妥当性確認活動を制限する制約項目を識別する。
  - 3) 各妥当性確認に適切な妥当性確認手法あるいは技法を選択する。
  - 4) 妥当性確認の戦略を定義する。
  - 5) 利害関係者要求に組み込まれている妥当性確認の戦略からシステムの制約事項を識別する。
  - 6) 妥当性確認に必要なイネープリングシステムあるいはサービスのうち、必要なものを識別し計画する。
  - 7) 妥当性確認に必要なイネープリングシステムあるいはサービスのうち必要なものを入手あるいは利用可能にする。
- b) **妥当性確認を実施する**
- 1) 妥当性確認の一つあるいはまとめた活動に対応する妥当性確認の手順を定義する。
  - 2) 決められた環境での妥当性確認の実施をする。
  - 3) 利害関係者に求められているシステムのサービスが可能であることを確認するように妥当性確認の結果をレビューする。
- c) 妥当性確認の結果を管理する。
- 1) 妥当性確認の結果および遭遇したどんな不具合も記録する。
  - 2) 運用上の事故および問題を記録しその解決を追跡する。
  - 3) システムあるいはシステム要素が利害関係者ニーズに合致していることの同意を利害関係者から得る。
  - 4) 妥当性確認がされている要素のトレーサビリティを維持する。
  - 5) ベースラインとして選択された主要な成果項目を準備する。

## 4.5 上流プロセスで使われている用語

前節で紹介したプロセスを理解するうえで助けになる用語を ISO/IEC/IEEE 15288:2015 の「4 Terms, definition, and abbreviated terms」から紹介する。各用語は、参照の便益を考慮し、以下のように説明する。

### 日本語訳(英語)、最初に使用されるプロセス番号、説明

プロセス番号は、4.4 節のプロセスの項番で示す。共通に使用される用語は本書の 1.4 節、4.4 節で説明しているため、その節の番号を記載する。ただし、本書が取り上げた対象 7 プロセスに関連した説明で使用されない用語については番号を記載しない。

なお、ほとんどの用語が、6.4.1 ビジネスあるいはミッション分析プロセス、あるいは、6.4.2 利害関係者ニーズと要求事項の定義プロセスに初出している。このことはまさしく新しい概念はライフサイクルプロセスの早期段階で認識されていることを示す。また、すでに現行 JIS にて定義されている用語はプロセスと同様に網掛けで示す。

### アーキテクチャ(architecture)、1.4 節、注) 説明文は大幅に変更

その要素、関係、そしてその設計と進化の原則を具体化するその環境におけるシステムの基本的な概念や性質

### アーキテクチャビュー(architecture view)、6.4.4、注) 新規

システムのアーキテクチャを、特定のシステムに対する関心の観点から、表現した作業成果物

### アーキテクチャビューポイント(architecture viewpoint)、6.4.4、注) 新規

特定のシステムの関心事項を特定するためのアーキテクチャビューの構築、解釈、および使用のための合意を確立する作業成果物

### アーキテクチャフレームワーク(architecture framework)、6.4.4、注) 新規

特定のアプリケーションドメインおよび/または利害関係者のコミュニティ内に確立されたアーキテクチャの記述に関する合意、原則および実践

**アクティビティ (activity)、6.4.1**

プロセスの構成要素で、関連の強いタスクの集合

**イネープリングシステム(enabling system)、6.4.1**

ライフサイクル段階の期間中は対象システムを支援するが、運用の間中はその機能に直接的に寄与するとは限らないシステム

**インシデント(incident)、6.4.9、注)新規**

プロジェクト、製品、サービス、またはシステムのライフサイクルのいずれかの時点における異常な、または予期しないイベント、イベントの集合、状態、または状況

**環境(environment)、6.4.2、注)新規**

システムに対するすべての影響の設定と状況を決定するコンテキスト

**監査(audit) 注)説明文は大幅に変更**

仕様、規格、契約上の合意、あるいは他の基準と整合していることを評価するための成果物あるいは成果物一式の独立した検査

**関心(concern)、6.4.1、注)新規**

利害関係者の1人以上に関連するシステムへの関心

注)4.4節の6.4.1には出てこないが、15288:2015の6.4.1の詳細記述の中で用いられている。

**供給者(supplier)、6.4.2**

製品またはサービスの供給に関して取得者と合意を結ぶ組織または個人

**検証(verification)、6.4.2**

客観的証拠を提示することによって、規定要求事項が満たされていることを確認すること



**合意 (agreement)、6.4.2**

業務上の関係を実行するための取り決めの相互確認

**構成項目 (configuration item)、6.4.1、注)新規**

構成管理プロセスで単一のエンティティとして扱われるもので、構成管理のために指定されるハードウェア、ソフトウェア、またはその両方の項目または集約

注)4.4節の6.4.1には出てこないが、15288:2015の6.4.1の詳細説明には用いられている。

**顧客 (customer)、6.4.1**

製品またはサービスを受け取る組織または人

**サービス (service)、6.4.1、注)新規**

活動、作業、あるいは任務の実行

**資源 (resource)、6.4.2**

プロセスの実行中に利用または消費される資産

**システム (system)、1.4節**

一つ以上の明記された目的を達成するために組織された相互に作用する要素の組み合わせ

**システム運用概念 (operational concept)、6.4.1、注)新規**

システムまたは関連するシステム一式の運用または一連の運用に関し、組織の前提または意図を文章または図式により記述したもの

**使用者のシステム運用概念 (concept of operations)、6.4.2、注)新規**

運用または一連の運用に関し、組織の前提または意図の広範な概要を、文章および/または図式により記述したもの

**システムズエンジニアリング (systems engineering)、1.4節、注)新規**

利害関係者のニーズ、期待、制約の一式をソリューションに変換し、そのソリューションをその生涯を通じてサポートするために必要とされる、技術的および管理的な取り組みを統括する学際的アプローチ

#### システム要素(systems element)、6.4.2

システムを構成する要素の集合の一部分

#### 情報項目(information item)、6.4.1、注)新規

人間が使用するために生成、保管、および提供される個別に識別可能な情報そのもの

#### 取得(acquisition)、6.4.2

システム製品またはシステムサービスを入手するプロセス

#### 取得者(acquirer)、6.4.2

供給者から、製品またはサービスを取得または調達する利害関係者

#### ステージ(stage)、6.4.1

実体のライフサイクル内の期間で、実体記述または実体の状態に関連するもの

#### 製品(product)、すべてのプロセス

プロセスの結果

#### セキュリティ(security)、6.4.1、注)2008年版と同一用語だが説明は2015年版で全面修正

意図的な破壊あるいは強制された破損に対する保護。機密性、完全性、可用性、および説明責任の4つの属性に加えて5つめの使用容易性の複合で、これらすべてはそれぞれの保証に対して関連する課題を持つ

#### 設計する(design、動詞)、6.4.2、注)新規

アーキテクチャ、システム構成要素、インタフェース、およびシステムあるいはシステム構成要素を定義する

**設計 (design、名詞)、6.4.2、注) 新規**

「設計する」プロセスの成果

**設計特性 (design characteristic)、6.4.4、注) 新規**

製品またはサービスの測定可能な記述に関連する設計属性または識別機能

**設備 (facility)**

活動の実施を容易にする物理的手段または装置。例えば、建物、器具、道具など

**操作者 (operator)、6.4.2、注) 同一用語だが、定義が限定的になっている**

システムの運用を実施する個人あるいは組織

**組織 (organization)、6.4.1、注) 2008年版では“人又は人々”が、2015年版では“人々”のみに修正**

責任、権限および関係の取り合わせを持つ人々の集団および施設

**妥当性確認 (validation)、6.4.1**

客観的証拠を提示することによって、特定の意図された用途または適用に関する要求事項が満たされていることを確認すること

注) 本文ではプロセス名に用いられているが、6.4.1の詳細説明には用いられている。

**対象システム (system-of-interest)、1.4節**

この規格の文脈においてライフサイクルが検討対象となるシステム

**タスク (task)、6.4.1**

プロセスの一つ以上の成果の達成に貢献するように意図された要求事項、勧告または許される行動

**トレードオフ (Trade-off)、6.4.1**

利害関係者にとっての実質利益に基づいて、様々な要求事項と代替解決策とか

ら選定するという意思決定行為

#### 廃止(retirement)、6.4.1

運用組織または保守組織が実施中の支援を中止すること、新システムに一部分若しくは全体を置き換えるか、または改定したシステムを導入すること

#### パーティ(party)

契約に関わる組織

#### 品質管理(quality management)、6.4.2、注)新規

要求に関連する製品、プロセス、またはシステムの固有の特性

#### 品質特性(quality characteristic)、6.4.6、注)新規

品質に関連し組織を統率および制御する要求に関連する製品、プロセス、またはシステムの固有の特性

#### 品質保証(quality assurance)

品質管理の一部であり、とりわけ品質要求が充足されることへの信頼を提供する活動

#### プロジェクト(project)、6.4.1

定められた資源および要求事項に従って、製品またはサービスを作り出すために実施される、定義された開始条件および終了条件がある取り組み

#### プロセス(process)、4.4節

インプットをアウトプットに変換する、相互に関連するまたは相互に作用する一連のアクティビティ

#### プロセスの目的(process purpose)、4.4節

プロセスを実行する高水準の目標およびプロセスの効果的实施によって見込まれる成果

**ベースライン(baseline)、注)用語は同じだが、説明文が大幅に変更**

構成アイテムのライフサイクル中の特定の時刻に、メディアに関係なく正式に指定され、修正された、正式に承認された構成アイテムの版

**問題(problem)、6.4.1、注)新規**

調査および是正処置を必要とする困難さ、不確かさ、あるいは、さもなくば実現される望ましくないイベント、イベントの集合、条件あるいは状況

**ライフサイクル(life cycle)、1.4 節**

システム、製品、サービス、プロジェクトまたは人が作った他の実体の構想から廃止までの漸進的な変化

**ライフサイクルモデル(life cycle model)**

段階に編成されることもあるライフサイクルに関するプロセスおよびアクティビティの枠組みで、情報伝達および理解のための共通に参照できる役割を持つもの

**利害関係者(stakeholder)、6.4.2**

システムに、権利、持ち分、請求権若しくは関心を持っている個人若しくは組織、または、ニーズおよび期待に合致する特性をシステムが持つことに、権利、持ち分、請求権若しくは関心を持っている個人若しくは組織

**リスク(risk)、6.4.1、注)新規**

目標が不確かなことによる影響

**ユーザー(user)、6.4.1**

システムを利用する間、システムからの恩恵を受ける個人またはグループ

**要求(requirement)、6.4.1、注)新規**

ニーズとその関連する制約条件を翻訳または表現する記述

## 参考文献

- [1] “日本再興戦略 2016－第 4 次産業革命に向けて,” 日本経済再生本部, 2016 年 6 月.
- [2] “平成 29 年度版情報通信白書,” 総務省, 2017 年 7 月.
- [3] “デジタル化の進展に対する意識調査,” 日本情報システムユーザー協会, 2017 年 3 月.
- [4] “平成 27 年度地域経済産業活性化対策調査 (地域の魅力的な空間と機能づくりに関する調査) 報告書,” 経済産業省地域経済産業グループ, 2015 年 9 月.
- [5] “2017 年版ものづくり白書,” 経済産業省, 2017 年 6 月.
- [6] “ISO/IEC/IEEE 15288:2015 Systems and software engineering - System life cycle processes,” 2015.
- [7] “Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, 4th Edition,” INCOSE, 2015.
- [8] “JIS Q 9000:2015 品質マネジメントシステムー基本及び用語,” 日本工業規格, 2015.
- [9] B. McFeeley, “IDEAL: A User’s Guide for Software Process Improvement,” CMU/SEI-96-HB-001, 1996.
- [10] “システムズエンジニアリング導入実施の一事例 報告書,” IPA/SEC, <https://www.ipa.go.jp/sec/reports/20180301.html>, 2018.
- [11] サンフォード フリーデンタール他, システムズモデリング言語 SysML, 東京電機大学出版局, 2012.
- [12] “JIS X 170:2013 (ISO/IEC 15288:2008) システムライフサイクルプロセス” .
- [13] “ISO/IEC/IEEE 12207:2017 “Systems and software engineering — Software life cycle processes”,” 2017.
- [14] “ISO/IEC 12207:2008 “Systems and software engineering — Software life cycle processes”,” 2008.
- [15] “JIS X 160:2012 (ISO/IEC 12207:2008) ソフトウェアライフサイクルプロセス,” 2012.
- [16] 共通フレーム 2013～経営者、業務部門とともに取組む「使える」システムの実現～, IPA/SEC, 2013.
- [17] J. P. Elm, “Leveraging Systems Engineering to Improve Project Performance,” CMU/SEI, 2011.

## おわりに

近年、第4次産業革命をもたらすと言われているIoTが進展し、独立したシステムが互いに関係し合って価値を提供するSoS (System of Systems) のような複雑なシステムも増加している。また、システムに対する要求の多様化、高度化、複雑化、予測できないリスク、関連する分野の調和や統合の欠如など、システム開発を取り巻く課題は多く、従来以上に解決が困難になってきた。

IPA/SECでは、このようなシステム開発における課題の効果的かつ包括的な解決方法として、欧米で宇宙・航空分野を中心に実績があるシステムズエンジニアリングが、他分野においても有効であると考え、システムズエンジニアリング推進WGでの議論、検討を踏まえて、産業界に普及・展開する活動を推進している。

本WGでは、多様な分野の委員が有する豊富な開発経験を持ち寄り、実用的な視点からの検討を重ねてきた。2016年度成果として、「経営者のためのシステムズエンジニアリング導入の薦め」および「開発者のためのシステムズエンジニアリング導入の薦め」を発行するとともに、2017年度半ばにはその改訂版も発行した。このたび、2017年度成果として、「成功事例に学ぶシステムズエンジニアリング～IoT時代のシステム開発アプローチ～」の発行にこぎ着けた。この一連の発行物が皆様のシステムズエンジニアリング導入の一助になることを期待している。

最後に本書の発行にあたり、事例の提供を含む多大なるご支援を頂いた関係者の皆様、事例を提供して頂いたキヤノン電子(株)、ソニー(株)に感謝の意を表する。

本書は、独立行政法人情報処理推進機構(IPA) 技術本部 ソフトウェア高信頼化センター(SEC) システムズエンジニアリング推進 WG において作成しました。

**編著者** (敬称略、2018年3月15日現在)

主査	白坂 成功	慶應義塾大学大学院	
委員	足立 正和	株式会社デンソー	
	石橋 金徳	慶應義塾大学大学院	
	大西 純	株式会社三菱東京UFJ銀行	
	梶本 一夫	パナソニック株式会社	
	川野 卓	東日本旅客鉄道株式会社	
	艸薙 匠	株式会社東芝	
	末富 隆雅	マツダ株式会社	
	長堀 泉	株式会社富士通総研	
	福島 祐子	日本ユニシス株式会社	
	松並 勝	DNV GL ビジネス・アシュアランス・ジャパン株式会社	
	松本 雅行	松本信号システムコンサルタント	
	事務局	中尾 昌善	IPA/SEC
		端山 毅	IPA/SEC
岡本 拓也		IPA/SEC	
齋藤 毅		IPA/SEC	
齊藤 善治		IPA/SEC	
新谷 勝利		IPA/SEC	
西原 栄太郎		IPA/SEC	
室 修治		IPA/SEC	



成功事例に学ぶシステムズエンジニアリング  
～IoT時代のシステム開発アプローチ～

---

平成30年6月4日 第1版第1刷発行

監修者 独立行政法人情報処理推進機構 (IPA)  
技術本部 ソフトウェア高信頼化センター (SEC)

発行人 片岡 晃

発行所 独立行政法人情報処理推進機構 (IPA)  
〒113-6591  
東京都文京区本駒込 2-28-8  
文京グリーンコートセンターオフィス  
URL <https://www.ipa.go.jp/sec/>

©独立行政法人情報処理推進機構 2018

---

ISBN 978-4-905318-60-6 Printed in Japan