

**WP2/D2.1 :**

**ドイツ・欧州企業における  
システムズエンジニアリングの  
実践課題とベストプラクティス**

著者 :

Jens Heidrich

Binish Tanveer

Rolf van Lengen

Thomas Kleinberger

Liubov Gorodilova

Thomas Kuhn

Martin Becker

Thomas Bauer

Andreas Morgenstern

Fraunhofer IESE（実験ソフトウェアエンジニアリング研究所）はフラウンホーファー研究機構（Fraunhofer-Gesellschaft）の研究所の1つ。

同研究所は、革新的なソフトウェア開発技術、手法およびツールを工業的な実践に転換し、企業が各々のニーズに合ったソフトウェア能力を構築し、市場優位性を確立できるように支援している。

Fraunhofer IESE の代表：

工学博士 Peter Liggesmeyer 教授 (Executive Director)

Dieter Rombach 教授 (Business Development Director)

Fraunhofer-Platz 1

67663 Kaiserslautern

Germany

# 目次

<b>本書の概要</b>	<b>1</b>
<b>1 システムズエンジニアリングへ向かう傾向</b>	<b>4</b>
1.1 システム統合の傾向	4
1.2 システムズエンジニアリングへの動機	5
1.3 背景情報	7
1.3.1 インダストリー4.0	7
1.3.2 Cyber-Physical Systems (CPS)	8
1.3.3 Internet of Things (IoT)	8
1.3.4 クラウドコンピューティング	8
1.3.5 Internet of Services	9
1.3.6 Industrial Internet	9
1.3.7 ビッグ・データ	9
<b>2 システムズエンジニアリングについての関連調査</b>	<b>11</b>
2.1 モデルベース・システムズエンジニアリング (MBSE : Model-Based Systems Engineering) 手法	11
2.2 プログラム・マネージメントとシステムズエンジニアリングの統合の改善	12
2.3 システムズエンジニアリングの産業界での実践	13
2.4 システムズエンジニアリングの有効性	14
2.5 モデル駆動開発	15
<b>3 システムズエンジニアリングに関する本調査・分析結果</b>	<b>17</b>
3.1 背景	18
3.2 課題	19
3.3 解決への取り組み	23
3.4 見通しと能力・機能	26
3.5 潜在的なリスクと限界について	28
<b>4 本調査・分析で判明した重要項目</b>	<b>30</b>
4.1 重要な成果報告	30
4.2 推奨事項と活動分野	32
4.2.1 組織開発	32
4.2.2 技術開発	33
<b>5 抽出された産業界での実践および事例</b>	<b>35</b>
5.1 モデル駆動システム開発	35

5.1.1	アプローチ例	36
5.1.2	実施事例	37
5.1.3	教訓と推奨事項	38
5.2	システム要求開発	39
5.2.1	アプローチ例	39
5.2.2	実施事例	40
5.2.3	教訓と推奨事項	43
5.3	システムの検証と妥当性確認	44
5.3.1	アプローチ例	44
5.3.2	実施事例	45
5.3.3	教訓と推奨事項	46
5.4	統合されたツール・チェーン	46
5.4.1	アプローチ例	47
5.4.2	実施事例	48
5.4.3	教訓と推奨事項	50
5.5	システムの仮想開発	50
5.5.1	アプローチ例	51
5.5.2	実施事例	51
5.5.3	教訓と推奨事項	52
	<b>参考文献</b>	<b>54</b>

## 本書の概要

本書では、システムズエンジニアリングへと進む一般的な傾向について議論し、本分野における既存の調査について要約し、今回実施したドイツ語圏の先進的な企業でのシステムズエンジニアリング分野における課題とベストプラクティスに関する一連のインタビュー結果にスポットを当てる。さらに、いくつかのベストプラクティスを抽出して説明し、これらを実践している企業の例を要約する。本調査・分析は日本の独立行政法人 情報処理推進機構 (IPA)からの委託調査として、ドイツの Fraunhofer IESE によって実施されたものである。

### 第 1 部

以前は大部分がハードウェアで占められていた製品が、今やソフトウェアの果たす役割が増え続けている。ソフトウェアは、産業や社会のあらゆるセクターで、新しく革新的なサービスやビジネス・モデルを実現するイネーブラーと見なされている。システムズエンジニアリングは、異なる専門分野にまたがるアプローチで、あらゆる顧客のビジネス・ニーズと技術的ニーズの両方を対象としている。企業で適切なシステムズエンジニアリングの実践を確立できる能力は、優位性を保ち、予算内でタイムリーに高品質で革新的な製品を開発するうえで必要不可欠になってきている。

### 第 2 部

今回実施した調査・分析の目的は、ドイツ語圏におけるシステムズエンジニアリングの、特にベストプラクティス（作業プロセス、手法およびツール）に関する課題と解決法に注目して、その実践状況についてまとめることである。本調査・分析の範囲は、様々なドメインにおける企業でのシステムズエンジニアリングの実践に関するものであって、単一のドメインに特化するものではない。この目的と関連する既存の調査や研究をいくつかあるが、今回の範囲と合致するものはなかった。

### 第 3 部

全体で、34 企業の 42 名に対して参加要請を送付した。そのうち 22 名がインタビューを承諾した。最終的に、Airbus DS Electronics and Border Security、ETAS GmbH、Hella KGaA Hueck & Co.、Robert Bosch GmbH、ZF TRW Automotive Holdings Corp.などからの専門家を含め、18 企業 20 名とのインタビューを実施した。

**製品開発の傾向：**企業は主にシステム要求の複雑化（60%）と、これまでにないほど多様な製品タイプの要求（50%）に直面している。

**システムズエンジニアリングの重要度：**1（重要でない）から 10（存続のために不可欠）の範囲で、システムズエンジニアリングの重要度の平均値は 7.6 であるが、これは今後 5 年以内に 8.5 に上昇するであろう。

**システムズエンジニアリングの課題**：80%が、変化に対応できる組織改革が第1の課題であると答え、これに複雑な要求およびインターフェースの管理が続く。

**システムズエンジニアリング・プロセス**：大規模企業では基本的に ISO/IEC 15288 および 12207 のすべてのプロセス分野に取り組んでいるのに対し、中小企業では明らかに技術的プロセスと実装プロセスに重点を置いている。

**システムズエンジニアリングの実践**：すでに確立している実践のうち、多くの企業（50%近く、またはそれ以上）はモデル駆動開発、要求開発、テスト駆動開発、および検証と妥当性確認に関連する手法、技法およびアプローチを採用している。

**仕様言語およびツール**：80%を超える回答者が、主要な仕様言語として UML を挙げた。大規模企業では、さらにシステムモデリングに特化した言語として SysML を使用する傾向にある。言及されたシステムズエンジニアリング・ツールの 50%超は、システム全体またはシステムの一部としてのソフトウェアの様々な側面のモデリングに関連している。

**改善の可能性**：システムズエンジニアリングの最大の改善の可能性は、仮想開発の強化と、使用されるツール・チェーンの統合の向上にあると答え、それぞれについて 50%の回答者が言及している。

**システムズエンジニアリングの能力・機能**：組織／部署の大部分が、システムズエンジニアリング関連の能力・機能の向上を、内部および外部のトレーニングプログラムに依存している。さらに、システムズエンジニアリング関連のコンファレンスへの参加も挙げられた。

#### 第4部

本調査・分析での主要な結果に基づいて、システムズエンジニアリングに取り組む企業に対する推奨事項と実行分野が導き出された。

**組織開発**：企業は、システムズエンジニアリングの実践を導入するための適切な組織改革戦略を確立し、十分な一般的システムズエンジニアリング能力、特にソフトウェア開発能力を構築する必要がある。とりわけ、大規模企業の場合は、様々なシステムズエンジニアリング・プロジェクトを統括管理することを検討する必要がある。

**技術開発**：企業は、すべてのステークホルダーを含めたシステムズエンジニアリングのアプローチを開発および統合し、システム要求開発、モデル駆動システム開発、および検証と妥当性確認の分野で実践を確立する必要がある。より成熟した企業の場合は、仮想システムズ開発およびシステムズエンジニアリングのツール・チェーンの統合で実践を確立する必要がある。

## 第 5 部

前述した 5 つの技術開発の実践分野のいずれについても、すでに確立した技法、手法およびツールが存在する。これらは活動の相当大きな部分を扱っており、実用的な環境または技術において適用ならびに評価済みである。さらに現在、国家的あるいは国際的な研究・開発プロジェクトや構想で技法、手法およびツールが開発されている。

# 1 システムズエンジニアリングへ向かう傾向

## 1.1 システム統合の傾向

ほぼすべてのドメインにおいて、企業間で共通の目標によって駆動され、顧客固有のソリューションを提供する、いわゆるスマート・エコシステムと呼ばれる完全な統合システムへ進む傾向にある。これらのスマート・エコシステムでは、ビジネス・プロセスと技術的プロセスを管理する従来型の孤立したソリューションを分解して、1つに統合された全体的なソリューションに集約している。

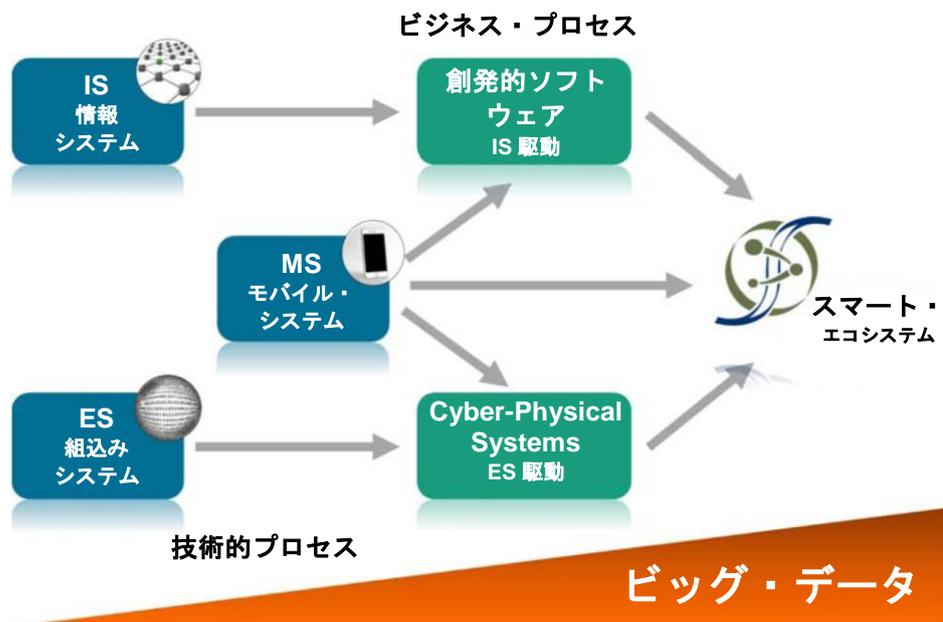


図 1: スマート・エコシステムへと進む傾向

これを達成するために、モノリシックな単一システムから、オープンで相互接続され、拡張可能（スケーラブル）でサービス指向型のソフトウェア・エコシステムへというパラダイムの変化が起ころうとしている。図 1 はこの傾向を図式化したものである。

情報システム（IS）は、システム・プロバイダー間、および支援されるビジネス・プロセス間における複数の情報システムの柔軟な組み合わせを可能にする創発的ソフトウェア・システムへと進化する。組込みシステム（ES）は、システムの接続に専用の通信インフラストラクチャと Internet of Things（IoT）を利用して、実世界の物理的オブジェクトのデジタル表現を可能にする Cyber-Physical Systems（CPS）へと進化する。現在は両方のドメインにおいて今、

モバイル・アプリケーションも急激にビジネス・プロセスに統合されつつある (Naab, Knodel, Kuhn, & Rost, 2016)。

スマート・エコシステムとは、企業間の境界を超え、ビジネス目標を達成するための重要な IS と、技術目標を達成するための重要な ES を柔軟に統合するシステムである。統合により 1 つのユニットとして動作し、単一のシステムでは達成できない、高度な共通の目標を達成する。

システム統合の高度化により、ビッグ・データの利用から得られるデータの価値と可能性は向上する。このシステム統合の傾向を示す好例は、自動車産業 (Car-2-X 通信など)、製造分野 (インダストリー 4.0)、エネルギー産業 (スマート・エネルギー)、医療技術 (スマート医療)、あるいは農業技術 (スマート農業) などに、あらゆるところで見つけられる。相互接続性の向上は、イノベーションの重要な要素であり、持続的成功の主要な誘因となる (図 2 参照)。



図 2: スマート・エコシステム：ドメイン間の傾向

## 1.2 システムズエンジニアリングへの動機

高度に統合されたシステムの開発の土台となるのは、モノリシックな単一システムから、オープンで相互接続され、スケーラブルでサービス指向型のソフトウェア・エコシステムへのパラダイム・シフトである。このビジョンを実現するには、システムの背後にある開発組織も変わらなければならない。以前は大部分がハードウェアで占められていた製品の中で、ソフトウェアの量は増え続けている。企業・組織間の協力は、ソフトウェア開発に成功するための重要な

要素である。物理的な世界はデジタル化およびスマート化し、Internet of Services、Internet of Things、Internet of Data が互いに結合しつつある。

ソフトウェアは、産業や社会のあらゆるセクターで、新しく革新的なサービスやビジネス・モデルを実現するイネーブラーと見なされ、ますます利用されている。他の市場参入者から差別化し、競争で優位に立つための要素は、今後ますます自社製品と他のシステムとの相互接続性から生まれるであろう。

International Council on Systems Engineering (INCOSE) によると、システムズエンジニアリングは異なる専門分野にわたるアプローチであり、ユーザーのニーズを満たす良質な製品を提供するという目標を持つ。また、すべての顧客のビジネス面と技術面の両方のニーズが考慮されるものであるという。

企業で適切なシステムズエンジニアリングの実践を確立できる能力は、優位性を保ち、予算内でタイムリーに高品質の革新的な製品を開発するために必要不可欠である。その能力によって企業は、以下に挙げる将来のシステムに典型的な性質に対応できる。

- (1) **複雑性**：これからのシステムはますます複雑になると予想される。企業には、そのような複雑さに対処する手段が必要である。これには、例えばテキスト記述に代わるモデルベース・エンジニアリングのアプローチ、適切なシステム要求開発、十分な柔軟性を実現するスケーラブルなアーキテクチャを持つこと、そして成熟したシステム開発プロセスが必要となる。
- (2) **多様性**：将来のシステムは、おそらく企業およびドメイン間にわたる多様なシステムとステークホルダーを包含および統合していくであろう。これには、例えば簡単な統合、インターフェースの標準化および QoS（サービスの質）を保証する相互運用可能なアーキテクチャが必要となる。
- (3) **不確実性**：時間の経過とともにプレーヤーとそれらとの関わり方が変化する可能性があるため、システムは不確実な環境に対応できなければならない。これに必要なのは、例えば一定の品質（システム性能、機能的安全性、セキュリティ、プライバシーなど）を実行時にもシミュレーション時にも保証できる適応性の高いシステムであり、開発時と実行時以降の結び付きをさらに緊密にする仮想開発のアプローチである。
- (4) **安全性とセキュリティ**：非常に重要な組込みシステムが、機密情報システムに統合される場合、最終的なシステムは機能的安全性とセキュリティの問題を同時に対応する必要がある。なぜなら、セキュリティ上の欠陥が安全性の問題となる可能性があるからだ。これには、例えばセキュリティと機能的安全性に同時に対応する統合モデルが必要となる。

(5) **ユーザー体験**：システムが急激に複雑化したとしても、ユーザーが使用し続けられるものでなければならない。製品の成功は、ユーザーが製品の使用から得る体験に大きく左右されるようになってきている。このユーザー体験を保証するためには、例えばシステムとユーザーとの対話に関して統合された戦略が必要となる。

(6) **自律性**：スマート・データの利用に基づき、将来のシステムはますます自律的もしくは半自律的に機能するようになる。これには、例えば、個々のシステムに高度な（人工）知能と適応性が必要となる。

(7) **データ駆動性**：将来のシステムの能力は、様々なソースから正しいデータに接続し、それらを適切に分析して、モデルを構築することに大きく依存するようになる。これには、例えば、一方では適切な品質のデータを識別して収集し、もう一方では個人のプライバシーを保証するための強力なデータ保護のメカニズムを導入できなければならない。

### 1.3 背景情報

以下に、システムズエンジニアリングへと進む傾向の基礎となっている概念に関する背景情報を以下に紹介する。

#### 1.3.1 インダストリー4.0

産業化の到来でテクノロジーは飛躍的に進歩し、それが4つの産業革命をもたらしてきた。すなわち、第1が機械化の分野における革命、第2が電気エネルギーの徹底した利用である。第3がデジタル化の普及で、物理的なあらゆる「モノ」がデジタルで表現されるようになった。そして、高度デジタル化、すなわちインターネットと「スマート」オブジェクト（機械と製品）分野の未来型テクノロジーの組み合わせが、次のパラダイム・シフトを促し、第4の革命であるインダストリー4.0（ドイツ経済エネルギー省、2016年）を工業生産にもたらし。

インダストリー4.0の戦略構想は、ドイツ政府によりハイテク戦略計画2020の中で提唱された。インダストリー4.0の概念を説明する際、ドイツ語圏外の国々では様々な用語が使用されている。例えば英語圏世界とEUレベルでは、IoTと、デジタル化へと進む傾向を第3の産業革命と読んでいる。ヨーロッパ、中国および米国では、スマート製造システムを構築するための生産のデジタル・ネットワークを指す言葉として、「スマート製品」、「スマート製造」または「スマート工場」といった用語が使われている（Kagermann, Helbig, Hellinger, & Wahlster, 2013）。

インダストリー4.0 のリファレンスアーキテクチャ・モデル (RAMI 4.0 : Reference Architecture Model for Industrie 4.0) (Hankel, M.; Bosch Rexroth, 2015) は、どんな標準やユースケースがインダストリー4.0に必要なか、そして関連性や詳細の議論を可能にするかという共通理解の目的を果たす統一的なアーキテクチャ・モデルである。RAMI 4.0では、インダストリー4.0の構成要素はそれぞれの構造と機能で定義される。したがって、付加価値ネットワーク全体を通じた企業間のネットワーキングと統合が可能になる。

製品のカスタマイズには、日常生活のニーズによって促される新しいビジネス・モデルが必要となる。この大量のデータの統合が、自己組織化、再組織化および自己最適化の能力を持つ、技術的なシステム・オブ・システムズを生み出す。こうしてカスタマイズされる製品には、開発時に設計される静的ソリューションから、実行時に自律的に適応し最適化する動的ソリューションへの移行が必要となる。

### 1.3.2 Cyber-Physical Systems (CPS)

Cyber-Physical Systems (CPS) は、組込みシステムを相互に、またウェブベース・サービスと結び付けることから進化したシステムである (acatech - National Academy of Science and Engineering, 2016)。つまり、CPSは物理的世界とIT世界の結合を意味し、適用業務プロセスにおいて人間とコンピューターとの対話を考慮しつつ、組込みシステム、アプリケーション・システムおよびインフラストラクチャの間の複雑な相互作用と統合から形成される。CPSは、インダストリー4.0の技術的な基礎である (Jazdi, 2014)。

### 1.3.3 Internet of Things (IoT)

ひと言で言えば、「IoTとは、インターネットを通じて、スマート・オブジェクト(モノ)相互間の相互作用、あるいはスマート・オブジェクトと物理リソースや仮想リソースとの間の相互作用に依存する新しいパラダイムである」(Cavalcante, et al., 2016)。IoTシステムの範囲は、一意に特定可能な「モノ」を備えた小さなシステムから、デジタル世界で物理的または仮想的な形(アイデンティティ、ステータス、場所など)をとって相互接続された無数の「モノ」から成るシステムまで多岐にわたる。こうした「モノ」は標準プロトコルを利用して相互接続され、知覚/操作能力や、場合によってはプログラミング能力まで備えて複雑なサービスを提供する。セキュリティを考慮すれば、こうした「モノ」によって提供されるサービスはいつでも、どこでも利用可能である (Minerva, Biru, & Rotondi, 2015)。

### 1.3.4 クラウドコンピューティング

米国国立標準技術研究所(NIST)は、クラウドコンピューティングを次のように定義している。「クラウドコンピューティングとは、管理負担あるいはサービス・プロバイダーの操作を最小限にして短期間で準備・公開できる構成可能

なコンピューティング・リソース（ネットワーク、サーバー、ストレージ、アプリケーション、サービスなど）の共有プールに対するネットワーク・アクセスを、遍在的（ユビキタス）かつ簡便にオンデマンドで実現できるモデルである」。紹介されているクラウド・モデルは、5つの基本的特性、3つのサービス・モデル、および4つの展開モデルで構成されている（Mell & Grance, 211）。

### 1.3.5 Internet of Services

IoT とクラウドコンピューティングが収束して、製造プロセスに Internet of Services と呼ばれる一連のサービスを提供する。これら一連のサービスは、クラウドベースのテクノロジーに基づいて公開および展開される。クラウドベース製造は、例えば、サービスの構成や、仮想エンタープライズへのソリューションとして使用される（Pisching, Junqueira, dos Santos Filho, & Miyagi, 2015）。顧客は、クラウドに接続されたウェブポータルを通じてサービスを要求することができる。これらのサービスの可用性を検証した後、バックグラウンド・システムはこれらのサービスを顧客に提供し、顧客のニーズを満たす。

### 1.3.6 Industrial Internet

Industrial Internet は、セーフティクリティカルな（安全を最重視する）産業用アプリケーションに重点を置いたIoTのサブ・パラダイムと見なされる（Bruner, 2013）。共通のネットワークにセンサーやソフトウェアを備えた複雑な物理機器の統合を指している。Industrial Internet は、要求の厳しい物理環境に対応する単一の仕組みを超えて、モジュール性、抽象、ソフトウェアといったウェブの主な特性を組み合わせ、大きい問題を細分化して解決し、その答えを再びつなぎ合わせる。Industrial Internet の基本を形成するテクノロジーの例として、「パーベイシブ・ネットワーク、オープンソース・マイクロコントローラ、大量のデータを分析し、人間の好みを理解し、多くの可変要素間で最適化する能力を持ったソフトウェア、そしてこの知能をどこでも安価に利用できるようにするために必要な演算力」（Wang, et al., 2015）が挙げられる。

### 1.3.7 ビッグ・データ

IoT への流れが急速になることで、データはますますアクセスしやすく、遍在的（ユビキタス）なものになり、これによってビッグ・データ環境が整備される。この現象によって、「データを有益で実行可能な情報に変換するための正しいアプローチとツールが必要になった」（Lee, Lapira, Bagheri, & Kao, 2013）。IDC と EMC が実施した「The Digital Universe of Opportunities」という調査によると、2020年の一年間で生成されるデジタルデータ量は44兆ギガバイトに達すると予測されている。デジタル化およびシステム統合の増加という、この最近の傾向により、「ビッグ・データ」と呼ばれる利用可能なデータの量が増加し（図1を参照）、新しいビジネス・モデルとイノベーションの主要なイネーブラーとなっている。

ビッグ・データは、データの量、データの提供と処理に必要な速度、データの多様化の3つの次元で特徴づけることができる。ビッグ・データの利用により、企業は、よりよい戦略決定を行う、プロセスを制御する、顧客をより深く理解する、コストを削減する、などが可能になる。しかし同時に、新しいビジネス・モデルを生み出すイネーブラーとしても見なすことができる（Heidrich, Trendowicz, & Ebert, 2016）。

## 2 システムズエンジニアリングについての関連調査

システムズエンジニアリング・プロセスならびに手法の利用とベストプラクティスに関する現行の国際的調査および出版物を分析した結果、以下に概要を示す3つの調査と2つの研究についてまとめた。

### 2.1 モデルベース・システムズエンジニアリング（MBSE : Model-Based Systems Engineering）手法

2007年にジェット推進研究所とカリフォルニア工科大学によって作成されたこの調査は、産業で使用されているいくつかの代表的なモデルベース・システムズエンジニアリング（MBSE）手法について概説している（Estefan, 2007）。調査の意図は、利用可能な様々な市販のMBSE手法およびその手法を支援するツールについて読者、主にINCOSE MBSE Focus Groupのメンバーを教育することであった。調査の対象となったのは、以下のMBSE手法である。

Telelogic Harmony-SE : Harmony-SE : Harmony®と呼ばれる、より大規模な統合システムとソフトウェア開発プロセスのサブセット。Harmony-SEでは、「サービス要求駆動」モデリング・アプローチが、Object Management Group™ Systems Modeling Language™（OMG SysML™）アーティファクト<sup>1</sup>と共に使用される。

INCOSE Object-Oriented Systems Engineering Method（OOSEM） : OOSEMは、OMG SysML™を使用してシステムの仕様、分析、設計および検証を支援するトップダウン、モデルベースのアプローチを統合する。

IBM Rational Unified Process for Systems Engineering（RUP® SE） for Model-Driven Systems Development（MDS） : RUP® SEは、Rational Unified Process®（RUP®）の派生品である。RUP®は、プロセスのフレームワークでもありプロセス製品の両方でもある手法で、政府機関や産業で広く使用されてきている。RUP® SEは特に、システムズエンジニアリング・プロジェクトのニーズに対応している。それが作られた目的は、ソフトウェア開発のためのRUP®の専門分野とベストプラクティスをシステムの仕様、分析、設計および開発についての課題に適用することであった。

Vitech Model-Based System Engineering（MBSE） Methodology : Vitech MBSE手法は、共通のシステム設計リポジトリを介してリンク付けられ、維持される4つの主要な同時進行SE活動を基にしている。これらの主要なSE活動はそ

---

<sup>1</sup> 注釈 : この調査が行われた2007年の時点で、Telelogicは独立企業であった。その間、Telelogicを買収により組み入れたRationalとHarmony Development Processは、さらにIBM Rational Harmony for Systems Engineering（IBM）として開発されている。

れぞれ、対応する「ドメイン」のコンテキスト内でリンク付けられる。この場合、SE 活動はプロセス・ドメインとして知られる特殊なドメインの要素と見なされる。モデルのアーティファクトの管理には MBSE System Definition Language (SDL) が使用される。

State Analysis (SA) : State Analysis (SA) は、モデルおよび状態ベースの制御アーキテクチャを活用する MBSE 手法である。この場合、状態は「進化するシステムの一時的な状態の表現」として定義され、モデルは状態が進化する方法を記述するものである。SA は、明示的なモデルの形式でシステムおよびソフトウェア要求を定義するためのプロセスであり、システム・エンジニアによって指定されるソフトウェア要求と、ソフトウェア・エンジニアによるその要求と実装との間でギャップを減らすのに役立つ。

## 2.2 プログラム・マネージメントとシステムズエンジニアリングの統合の改善

この調査では 2012 年に、3,000 人の INCOSE メンバー（システム・エンジニア）と 5,000 人の PMI メンバー（プログラム・マネージャー）に対し、組織内でプログラム・マネージメントとシステムズエンジニアリングがどのように統合されているかというメンバーの理解について尋ね、また標準の使用、統合、正式化、有効性レベルの間の相互作用、ならびにプログラム・マネージメントとシステムズエンジニアリングの間に存在する非生産的な緊張の度合いについて記述するように求めた (Conforto, Rossi, Rebentisch, Oehmen, & Pacenza, 2013)。

680 人のチーフ・システム・エンジニアとプログラム・マネージャーが回答した。彼らの属性は、主に営利組織（78%）であり、拠点は米国（58%）をはじめ、インド、イギリス、ドイツ、中国、南アフリカであった。中心事業は、専門／科学系（36%）、製造（13%）、行政機関（12%）、輸送（6%）、医療（4%）であった。

回答者の約 30%は、システムズエンジニアリングとプログラム・マネージメントの間でいくらか、もしくは大きな非生産的な緊張があると表明した。約 20%は、非生産的な緊張はまったくないと答えた。小規模な企業（年収 5 億ドル未満）と大規模企業（50 億ドル超）は特に、非生産的な緊張による悪影響を受けるリスクを負っている。統合された計画の欠如が、非生産的な緊張の主な要因である。完全に統合された企業では、非生産的な緊張がほぼまったくないか、あっても最低限のレベルである。

非生産的な緊張を減らすための主要な方策は、両方のドメインの標準を使用することによるシステムズエンジニアリングとプログラム・マネージメントとの統合の改善、統合の定義の正式化、統合されたエンジニアリング・プログラムの評価方法の開発、リスク管理、品質、ライフサイクル計画および外部サプライヤーについての効果的な責任分担である。

## 2.3 システムズエンジニアリングの産業界での実践

2013年に Fraunhofer IPT（生産技術研究所）、Heinz Nixdorf Institute および Unity Consulting & Innovation によって実施されたこの調査の目的は、システムズエンジニアリングの能力を明確に把握することと、実践およびトレーニング・上級教育活動における現在のシステムズエンジニアリングの使用レベルを把握することにあった（Gausemeier, et al., 2015）。潜在的な利点の完全活用を阻んでいる現在の障壁が強調され、それらを克服するための推奨事項が示されている。この調査は、ドイツ、オーストリア、スイス（ドイツ語圏）の製造業企業およびサービス・プロバイダーに所属する専門家に対して行われた 33 回のインタビューを基にしている。この調査の回答者は企業内の様々な役職に就いており、主に CEO、開発マネージャー、生産マネージャーおよびシステム・エンジニアが面談のインタビューとなった。主な結果を以下に記す。

システムズエンジニアリングという用語は実際によく知られており、ほとんどの企業が基本的な理解を持っている。しかし、深く理解しているのは真の専門家のみである。多くの場合、システムズエンジニアリングを議論する際、ソフトウェア開発のみが注目されており、焦点が狭すぎる。

基本的に、全回答者がシステムズエンジニアリングの適用に相当の可能性を見ている。特に中小企業では、システムズエンジニアリングに関する話題は、きわめて属人的である。しかし、これらの人々が特にシステムズエンジニアリングの考え方とアプローチを日々の仕事に持ち込むことに最も熱心である。同様に、大企業でも全社的なシステムズエンジニアリングの認知はまだ顕著ではなかった。

全セクターおよび企業規模で、インタビューを受けた人のシステムズエンジニアリングに関する専門知識の習熟度に関わらず、すべての項目が重要であると見なされた。平均して、システムズエンジニアリングに関する専門知識が少ない企業の方が多くの企業よりも自分たちのことを高く評価していた。ここで克服すべきは、ノウハウと系統的アプローチの不足、そして不十分なツール支援であった。

産業の観点からシステムズエンジニアリングは複雑な技術的システムを開発するために必要な前提条件であることが、この調査で証明されている。これは、知的能力が伸び、ネットワーク化が進む将来のシステムに関してのみでなく、現在の製品ならびに開発される製品システムに関しても言える。個々の専門分野に固有のアプローチだけではもはや対応不可能な、システムの多専門分野性が、複雑性を助長する大きい要因になっている。

ドイツ語圏では、システムズエンジニアリングの適用はセクターごとに大きく異なる。したがって、予期したとおり、航空宇宙業では長年にわたってしっかりと確立され、必要不可欠となっている。一方、自動車製造業ではシステムズ

エンジニアリングは重要な「イネーブラー」と見なされている。特にドイツの OEM 企業は、システムインテグレーターとしての地位を維持するための可能性として認識されている。一方、他の産業分野については、特に大部分が中小企業で構成される機械産業およびプラントエンジニアリングの分野では、システムズエンジニアリングはほとんど知られていなかった。

## 2.4 システムズエンジニアリングの有効性

2012 年に National Defense Industrial Association Systems Engineering Division (NDIA-SED) は、Institute of Electrical and Electronic Engineers Aerospace and Electronic Systems Society (IEEE-AESS) および Carnegie Mellon<sup>®</sup>の Software Engineering Institute (SEI) と、プロジェクトのパフォーマンスに関するシステムズエンジニアリング (SE) のベストプラクティスの利点に関して量的証拠を得るための共同調査を行った (Elm & Goldenson, 2012)。この調査の目的は、プロジェクトで使われる SE のベストプラクティスを識別し、これらのプロジェクトのパフォーマンス・データを収集し、SE ベストプラクティスの適用とプロジェクトのパフォーマンスとの間に関係を見出すことにあった。

調査の対象は、NDIA-SED、IEEE-AESS および INCOSE が連絡を取ったシステム開発者によって実施されるプロジェクトやプログラムで構成された。約 148 名の回答者が調査に協力した。回答の大部分は、米国国内で米国国防総省と契約する米国の防衛産業企業からであった。

概して、この調査ではプロジェクトへの SE ベストプラクティスの適用とプロジェクトのパフォーマンスとの間で、明確かつ有意な関係が示された。この調査の結果、プロジェクトのパフォーマンスと最も強い関係を持つ SE プロセス・グループが特定された。また、課題が多いプロジェクトの方が、課題が少ないプロジェクトよりもパフォーマンスが悪い傾向があることも示された。しかし、課題にあまり直面しないプロジェクトは、それでもシステムズエンジニアリングのベストプラクティスを実装することで恩恵を受ける傾向がある。さらに、システムズエンジニアリングのベストプラクティス適用の効果は、課題が多いプロジェクトの場合はさらに大きくなる。(図 3 参照)

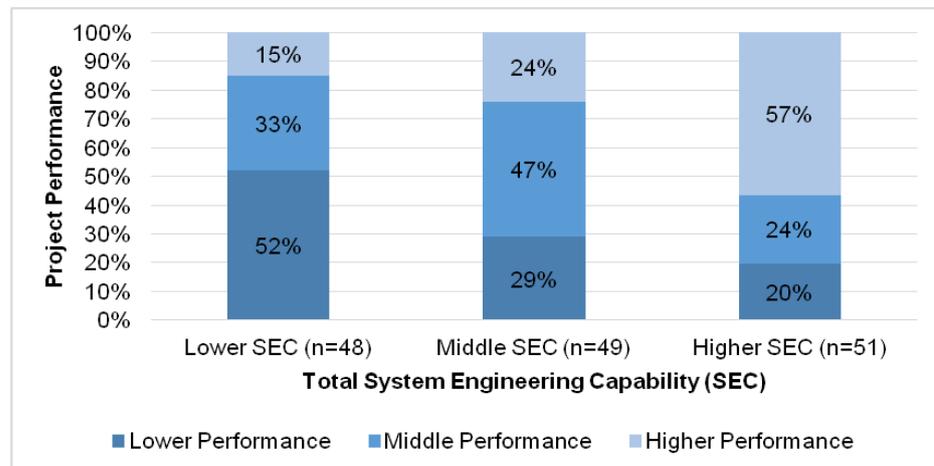


図 3 : Relationship between Systems Engineering and Performance  
 (Source: Carnegie Mellon University, Software Engineering Institute)

## 2.5 モデル駆動開発

この調査は、2013 年に Fraunhofer IESE により、ドイツ自動車業界の大規模な製造企業（匿名希望）のために実施された。調査の目的は、ソフトウェアおよびシステムの実際の開発においてモデル駆動開発のアプローチがどのように使われているかを調査することにあった。興味深かったのは、開発プロセスに実装されているツール群を、ツール・チェーンとして統合していることである。この調査には、自動車業界の企業の生産および研究を行うソフトウェア開発部門に所属する 36 名が参加した。

回答者の大部分が機能的挙動（Simulink および ASCET を使用）とソフトウェア構造（UML/SysML ならびに Simulink/ASCET を使用）のモデリングを行っていた。モデリングは主にシステムアーキテクチャおよびインターフェースに適用されている。一般に、要求はそれほど頻繁にはモデリングされていなかった。

モデルの適用に関しては、調査から以下の結果が得られた。要求抽出については、Simulink と Visio が構造モデル（SysML、UML）に部分的に使用されている。機能モデリングについては、実行可能モデルに Simulink、ASCET、Modelica が、構造モデルに SysML、UML が使用されている。アーキテクチャについては、構造モデル（SysML、UML）に Visio が使用されている。設計については、UML、Simulink が使用されている。テストについては、Simulink あるいはその他のツールが使用されている。

モデルの作成に関しては、調査から以下の結果が得られた。モデリングに投資する時間は十分価値がある。モデルは決定の妥当性を確認するために頻繁に使用される。モデルはアルゴリズムを理解するのに役立つ。モデルは早期に欠陥

を検出するのに役立つ。コードの自動生成は、常に役立つわけではない（生成されるコードの品質が不十分なことが多い）。

モデルの品質に関しては、回答者の多くが SysML/UML モデルは欠陥、矛盾がある、または最新のものではなかったと述べている。Simulink モデルの品質は通常 SysML/UML モデルよりも高い。

### 3 システムズエンジニアリングに関する本調査・分析結果

本調査・分析の目的は、ドイツ語圏におけるシステムズエンジニアリング、特にベストプラクティス（作業プロセス、手法およびツール）に関する課題と解決法に重点を置いて、その実践状況について調査・分析することである。本調査・分析の範囲は、様々なドメインにおけるシステムズエンジニアリングの実践に関するものであって、単一のドメインに特化したものではない。全体で、34 企業・組織の 42 名に対して参加要請を送付した。そのうち 22 名がインタビューを承諾し、最終的には 18 企業 20 名のインタビューを実施した。6 つのが中小企業、14 が大規模企業として分類された。以下に、本調査・分析での参加企業として公表することに同意した企業名を示す。

企業	ドメイン	種類
Airbus DS Electronics and Border Security	航空宇宙、電子機器	大規模企業
Art of Technology AG	製造、医療、航空宇宙	中小企業
AVL LIST GmbH	自動車	大規模企業
Binder Elektronik GmbH	産業エレクトロニクス、医療	中小企業
camLine GmbH	製造、医療、自動車、航空宇宙 および半導体向けソフトウェア サプライヤー	中小企業
CIBEK technology + trading GmbH	高齢者向けソリューション、自 動化テクノロジー	中小企業
ETAS GmbH	自動車	大規模企業
Hella KGaA Hueck & Co.	自動車、電子機器、照明	大規模企業
Robert Bosch GmbH	製造、自動車、消費者向け電子 製品	大規模企業
ZF TRW Automotive Holdings Corp.	自動車	大規模企業

表 1：研究の参加者のリスト

インタビューの平均時間は約 60 分であった。1 回のインタビューには 29 の質問（12 の主質問と各補足質問）が含まれた。質問は、回答者とその企業、組織／部署の背景、直面するシステムズエンジニアリング関連の課題、その課題に対処するための解決法と実践、およびシステムズエンジニアリングと企業の一般的な能力・機能が将来的に改善する見通しという 4 つの異なる部分に分類された。

### 3.1 背景

**ドメイン:** 企業のドメイン別内訳を図4に示す。大部分は自動車および製造ドメインで、それ以外は航空宇宙、輸送、医療と続く。ごく一部が電子機器および機械工学ドメインに属する。

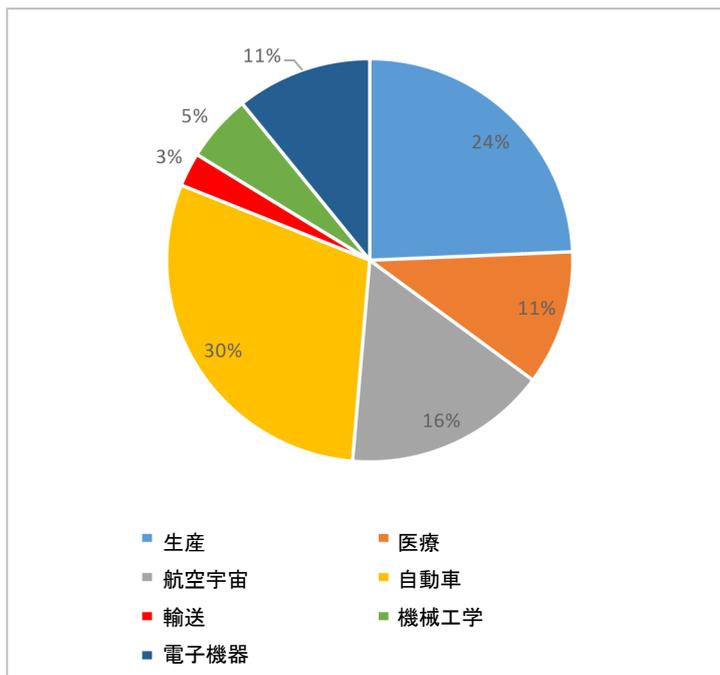


図4：ドメイン間の参加者の内訳

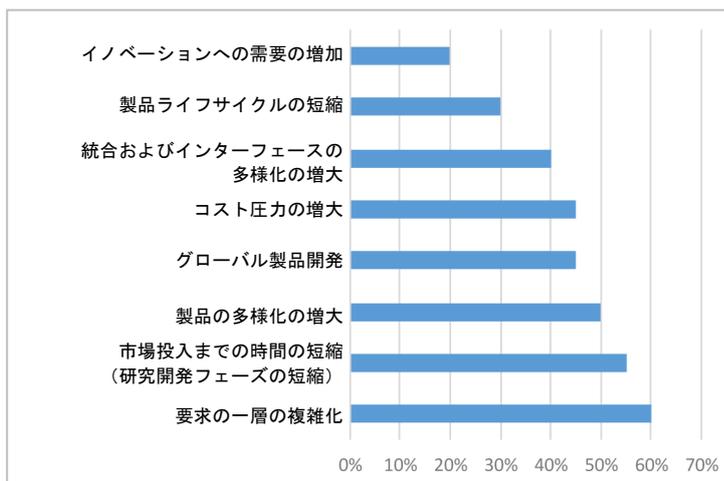


図5：最近の傾向

**製品開発における現在の傾向：**企業は現在、製品開発において多くの傾向に直面している。共通するのは、例えば、要求がますます複雑になっている（60%）、市場投入までの時間、あるいは研究開発フェーズが短縮されている（55%）、カスタマイズ製品を求める顧客の要望により製品が多様化している（回答者の50%）などである。回答者の45%が、コスト圧力の増大とグローバル製品開発の増加を最近の傾向として挙げている。特に一般的な傾向のリストを図5に示す。

**製品開発における将来の傾向：**図6に示すように、回答者の多くは、現在の傾向が今後5年間継続すると想定しているが、回答者によると、今後の主要な傾向は、複数の専門分野にわたる開発の増加、コスト圧力の増大、市場投入までの時間の短縮（挙げられた割合はそれぞれ20%）であろうと述べている。

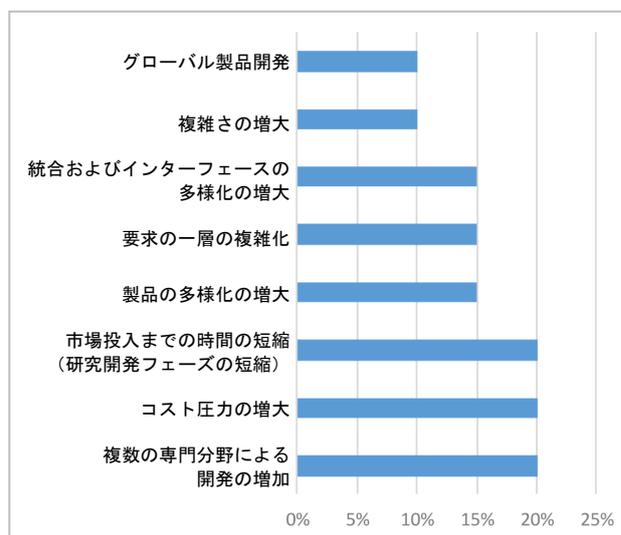


図6：今後5年間の傾向

### 3.2 課題

本項では、現在および今後5年間に企業が直面しなければならないシステムズエンジニアリングに関する（製品、システム開発プロセス、組織構造、必要な能力などに関する）実用面の課題について議論する。

**現在におけるシステムズエンジニアリングの重要度：**1（重要でない）から10（存続のために不可欠）の範囲で、すべての企業が、実装されたシステムズエンジニアリングのプロセス（プロジェクトプロセス、技術的プロセス、合意プロセスおよび組織的プロセス）は重要、非常に重要、または不可欠であると述べている（図7参照）。参加企業の回答で最も低い重要度は5（中程度の重要性）で、最も高い値は10（存続のために不可欠）である。平均すると7.6であり、重要であることを意味する。

参加企業の約 25%にとって、このプロセスは真に不可欠である（重要度が 9 か 10）。全参加企業の約 35%がシステムズエンジニアリングは中程度の重要性（重要度が 5 か 6）しかないと答えている。その他の参加企業はその中間の値を選択している。標準偏差は 1.5 で、これは回答の違いが僅差であることを意味している。

システムズエンジニアリングが何の役割も果たしていない、あるいはわずかな役割しか果たしていないと回答した企業はない。大規模企業と中小企業間で有意な違いは見られない。

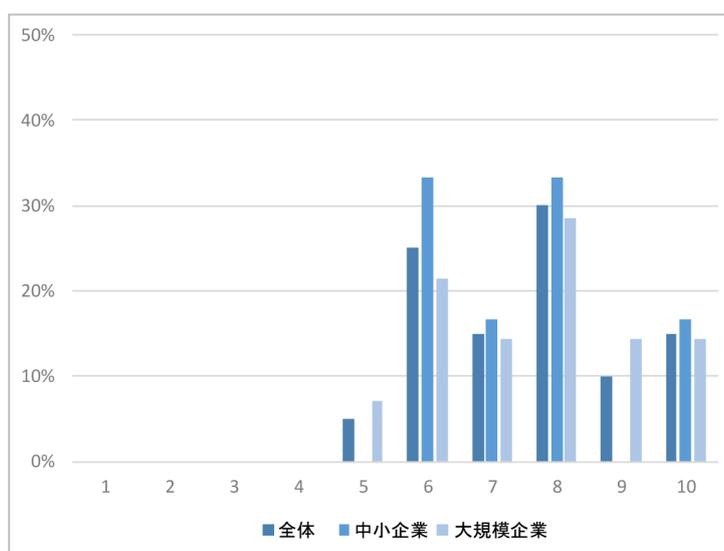


図 7：現在におけるシステムズエンジニアリングの重要度

**5年後のシステムズエンジニアリングの重要度**：ほぼすべての企業がシステムズエンジニアリングは今後ますます重要になると予測している。重要度は、今後 5 年間に平均で 7.6 から 8.7 へと大きく伸びる見込みである（図 10 から図 11 へ重要度が上向きに推移していることに注目）。この上昇は、すべての種類および規模の企業間、ならびにすべての業務ドメイン間で共通して見られる。それでもやはり、中小企業より大規模企業のほうが、5 年後の重要性を高く予測していることが、図 11 から分かる。システムズエンジニアリングは今後、中小企業より大規模企業で重要な役割を果たすだろうと解釈できるのである。

標準偏差は 1.5 から 1.1 へと低下しており、これは 5 年後に予測される重要度が、8.7 という平均の重要度にさらに集中している（同じように重要度が高くなると予測している企業が増えた）ことを意味する。

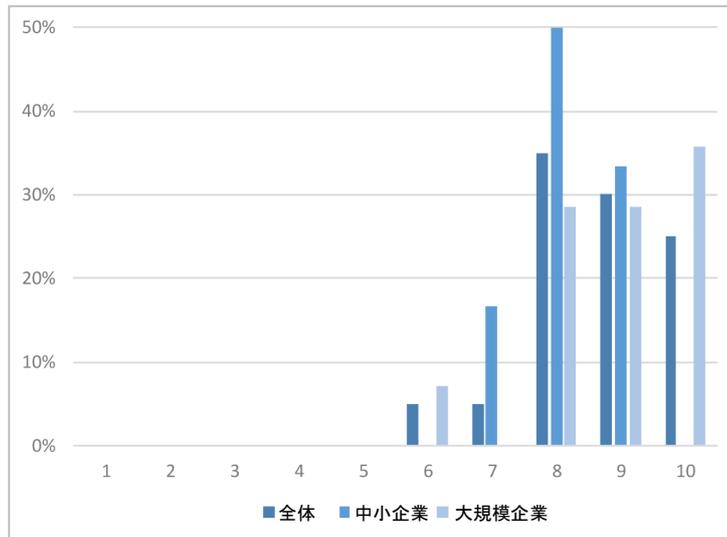


図 11：5年後のシステムズエンジニアリングの重要度

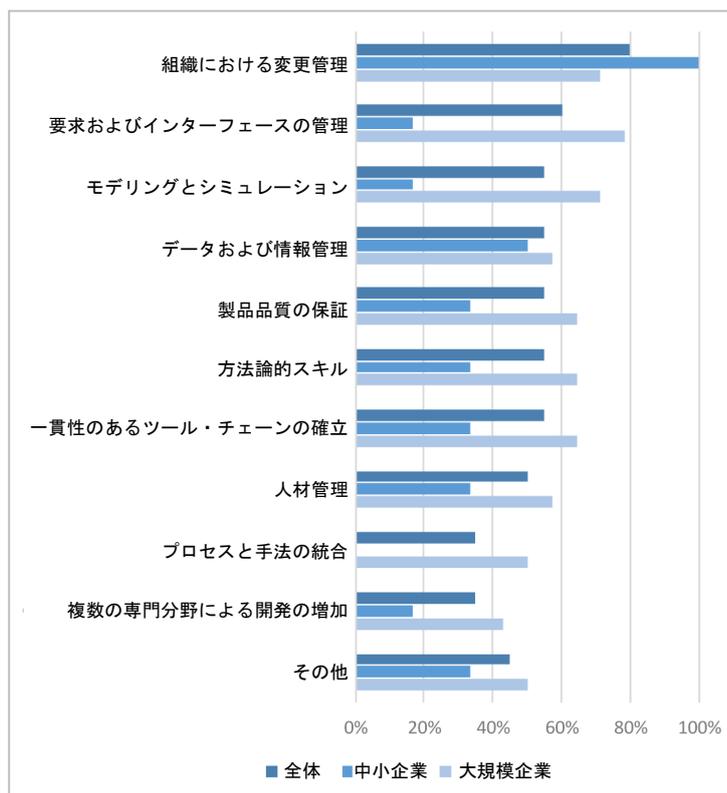


図 8：システムズエンジニアリングにおける現在の課題

**システムズエンジニアリングにおける現在の課題：**ほぼすべての企業が現在直面している最大の課題は、変化に対応できる企業への変革であり、それに要求およびインターフェースの管理が続く（図 8 参照）。その他の重要な課題としては、モデリングとシミュレーション、データおよび情報管理、製品品質の保

証、専門家の専門分野内ならびに専門分野間の方法論的スキルの確立／維持、一貫性のあるツール・チェーンの確立、人材管理などがあった。

中小企業が直面する主な課題としては、変化に対応できる企業への変革、専門家の専門分野内ならびに専門分野間の手法面のスキル、企業内および企業間での一貫性のあるツール・チェーンの確立、製品品質の確保（信頼性、安全性、セキュリティなど）が挙げられる。

大規模企業が直面する主な課題としては、複雑なシステムまたはシステムオブシステムズ (SoS) の要求およびインターフェースの管理、モデリングとシミュレーション、変化に対応できる企業への変革をもたらす新しいアプローチやテクノロジーの受容、企業内でも企業間でも一貫性のあるツール・チェーンの確立、製品品質の保証（信頼性、安全性、セキュリティなど）が挙げられる。

また、ほとんどの企業では、現行製品ロードマップ、プロセス組織、インフラストラクチャまたはツール・チェーンに応じて、上記以外の課題に取り組んでいる。図8では、では、そうした個別の課題を「その他」の項目にまとめている。

**システムズエンジニアリングにおける今後の課題：**ほとんどの企業で、今後5年間のシステムズエンジニアリング関連の課題は、大部分が現在直面している課題と変わらない。加えてシステムズエンジニアリングに関するその他の将来的な課題は、業務ドメインと、各企業の個別のシステム開発プロセスによって多岐にわたる。

技術的プロセスのレベルでは、これらの課題の範囲は、モデルベース開発からアジャイル開発またはラピッドプロトタイピングを経て、仮想プロトタイピングおよびシミュレーションによる検証と妥当性確認まで及ぶ。今後到来するシステムオブシステムズ (SoS) に備えた要求およびインターフェースの管理の向上という回答もあった。

プロジェクトプロセスのレベルでは、投入する製品タイプの拡大、製品品質の維持（特にセキュリティおよび安全性に関して）といった新しい課題が重要性を増しつつある。このレベルについて中小企業は、データおよび情報管理と変化に対応できる組織への変革の導入に特に関心を持っている。

組織プロセスのレベルでは、企業におけるデジタル化の転換プロセスへの対応を目的とした変化に対応できる組織への変革の向上、あるいはシステムズエンジニアリング・プロセスを実際実施するための厳格なリーダーシップといった新しい課題も重要であることが見受けられる。いくつかの企業は、システムズエンジニアリング・プロセスのノウハウを築いて保持するために、人材管理に重点を移すことを計画している。

ここでは、企業の種類または規模間に共通する傾向は見られない。しかし、個々の業務ドメインでは新機能に対する特定の傾向が存在する。例えば自動車ドメインでは、自動運転の登場によって、ネットワーク接続されたシステムの要求およびインターフェースの管理、製品／ソリューションのモデリングおよびシミュレーション、安全性要求をめぐる新しい課題が生じている。

### 3.3 解決への取り組み

システムズエンジニアリングに関連する主要な課題を議論した後、本項では業界で選択されている最も効果が期待される解決法について議論する。これは、ベストプラクティス、標準、手法、ツールなどの使用で構成される。

**システムズエンジニアリング・プロセス：**大規模企業は基本的に ISO/IEC 15288 および 12207 のすべてのプロセス分野に取り組んでいるのに対し、中小企業は明らかに技術的プロセスと実装プロセスに重点を置いている。企業が準拠している標準は、ISO 9001 などのごく一般的なアプローチを除けば、かなりドメイン限定である。大規模企業の 40%は明示的に ISO/IEC 15288 を挙げている。

使用しているプロセスモデルに関しては、45%を超える大規模企業および中小企業がアジャイルモデルに従っていると述べたのに対し、50%を超える大規模企業はウォーターフォールモデルまたは反復型ウォーターフォールモデルに従っている。さらに、80%を超える大規模企業が、標準プロセスの様々な派生を挙げている。

中小企業の大部分（83%）は、1つの共通の開発プロセスを定義し、派生はほとんどない。大規模企業の 86%では、複数の種類の開発プロセスが存在している。しかし、一般には、プロジェクトのニーズに合わせてカスタマイズされた標準開発プロセスが定義されている。

**ステークホルダーの関与：**大規模企業では、各ステークホルダーならびに専門分野を、定義されたプロセスに従って連携および調整している（85%）。中小企業の場合は、製品開発を取りまとめる手段として個人間のコミュニケーションが選択されている（20%）。

プロジェクトに対する共通の理解を得るためのワークショップや混合チームの設置は全企業において確立されている（中小企業で 80%、大規模企業で 57%、全体で 50%）。企業の境界を超えて共通のデータプールやツールを使用すること、全企業にとって課題となっている（中小企業で 20%、大規模企業で 50%）。

開発活動へのサプライヤーの統合は、主にサプライヤー契約によって実施されている（中小企業で 40%、大規模企業で 93%）。大規模企業では、外注管理を通じて緊密な関係が確立されている（43%）。

また、大規模企業の場合、開発活動に外部ナレッジを提供するために一時的な人材リースの概念が広く採用されている（64%）。開発活動で共通の理解を得るために小規模な企業で行われている手段としては、外部サプライヤーと共にワークショップを行うといったトレーニング活動がある（40%）。

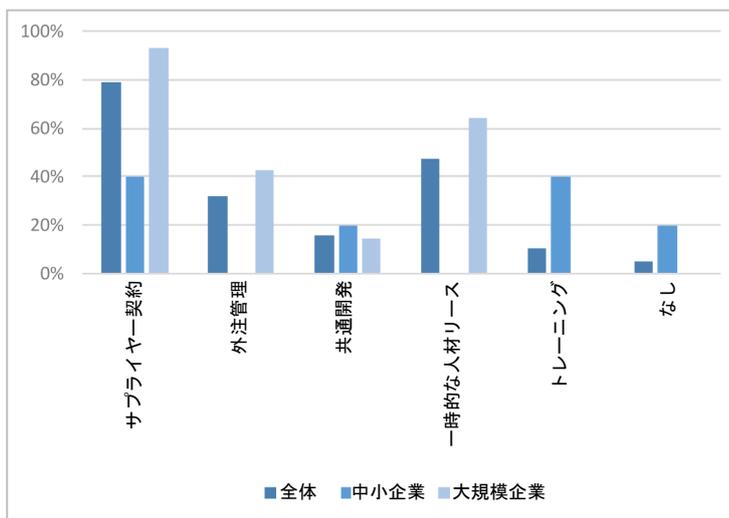


図 9：外部サプライヤーの統合

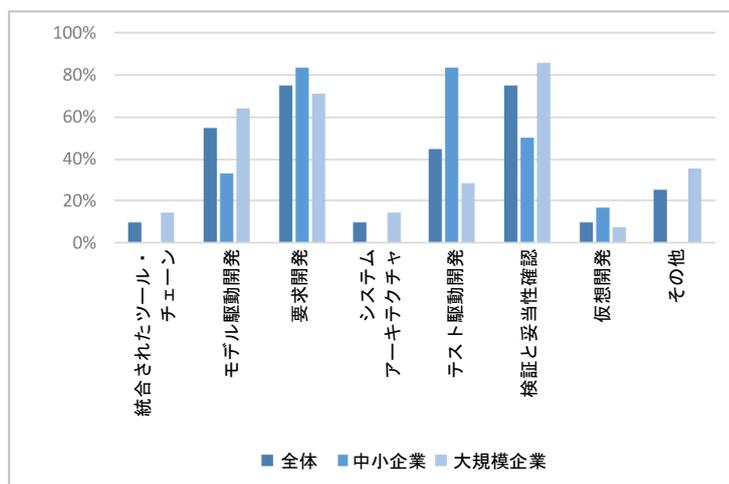


図 10：確立したシステムズエンジニアリングの実践の上位

**確立した上位 3 つの実践：** 図 10 から分かるように、すでに確立している実践のうちで企業が広く（50%近く、またはそれ以上）適用しているのは、モデル駆動開発、要求開発、テスト駆動開発、および検証と妥当性確認に関連する手法、技法、アプローチである。少なくとも 2 企業以上 1 つの企業によって言及されたさらなる将来の実践には、統合されたツール・チェーンと仮想開発、および全体的なシステム全体のアーキテクチャなどがある。その他の実践として

2 企業以上から挙げたのは、統合されたツール・チェーンと仮想開発、全体的なシステムアーキテクチャなどだった。

また、実践の上位に選ばれたものは大規模企業と中小企業の間で異なっている。特に違いが大きいのは、モデル駆動開発とシステムの検証および妥当性確認である。大規模企業ではそれぞれ60%と80%を超える企業が選択したのに対し、中小企業ではそれぞれ40%未満と約50%しか選択していない。80%を超える中小企業が、確立した最上位の実践としてテスト駆動開発を挙げている。大規模企業の場合、テスト駆動開発は約30%にとどまっている。とはいえ、大規模企業がテスト駆動開発を行っていないというわけではなく、単に上位3つの実践の1つとして選択しなかった点に留意してほしい。

最も強く影響を受けるプロセス分野については、技術的プロセスとソフトウェア実装プロセスの分野（ISO/IEC 15288 ならびに 12207 による定義）が、こうした実践によって影響を受けていると、回答者の大多数が同意している。

**システムモデリングの言語：**図 11 に示すように、大部分がモデリング言語として UML に言及している。大規模企業では、システムモデリングにより特化した言語として SysML (UML ベースの言語) を使用する傾向にある。さらに、ドメイン固有の言語も挙げられた。単独の回答には、DFD (データフロー図)、FMI (ファンクショナル・モックアップ・インターフェース)、OSLC (Open Services for Lifecycle Collaboration)、構造化分析、XML/XMI、IDef0、AUTOSAR などがあつた。

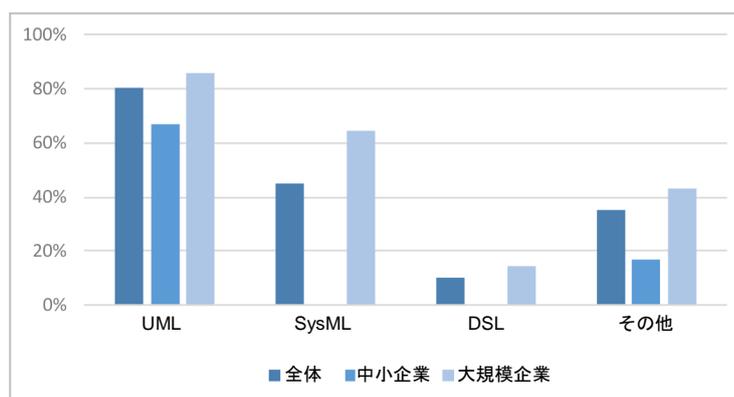


図 11：使用されている仕様言語

**システムズエンジニアリングのツール：**全体で、システムズエンジニアリング関連のツールの使用状況に関しては、本調査・分析の回答者から 90 件以上のコメントがあり（そのうち 80%は大規模企業からのもの）、その中で 40 以上の異なるツールまたはツールのコンポーネントが挙げられた。

言及されたツールの大部分は、システム全体またはシステムの一部としてのソフトウェアの様々な側面のモデリングに関連している。ドメインによっては、CAD ソフトウェアなど、きわめてドメイン特有のツールもあった。さらに、前述した実践を支援するものとして、ほとんどが要求固有のシミュレーションツールおよびテストツールを挙げた。

モデリングツールに関しては、回答者の約 50%が「Enterprise Architect」と「MATLAB」を使用していると述べている。要求ツールに関しては、30%が「DOORS」と「Microsoft Office」を使用している。シミュレーションツールに関しては、40%が MATLAB の「Simulink」エクステンションを使用している。テストツールに関しては、様々なツールが挙げられた。

言及されたツールまたはコンポーネントの 90%近くが特定の開発活動の用途固有であるのに対し、10%強が多目的ツールあるいは統合されたツールスイートであった。さらに、回答の 10%近くが自社開発したツールを挙げた。そのほとんどはシミュレーションの分野で使用されている（約 5%）。

**新しいテクノロジーとニーズ：**新しいテクノロジーうち、40%近い割合で最も目立ったのは、非形式的なテキストによる仕様に代わって、形式手法とモデルベースのシステム開発アプローチを採用することであった。また、近い将来の技術分野としては、よりよく統合されたツール・チェーン、シミュレーションなどの仮想開発、独自の特殊ツールの開発が全般的に必要であると指摘された。さらに、ビッグ・データ、IoT、サービス指向など、テクノロジーの選択に影響を与える一般的な製品／機能関連の傾向もいくつか挙げられた。

### 3.4 見通しと能力・機能

**改善の可能性：**図 12 に示すように、システムズエンジニアリングに関して最も大きな改善の可能性があるのは、仮想開発の拡大と、使用されるツール・チェーンの統合強化で、いずれも 50%を上回っている。これらの要求は中小企業でより強いことが見受けられる。

大規模企業の 40%近くについては、プログラム・マネージメント（プロジェクト・ポートフォリオ管理ともいう）の改善についても言及しておく必要がある。大規模企業ほど、より多くのプロジェクトの実行を同時に進める必要があるため、これは当然と考えられる。中小企業の 40%近くは、自動化をさらに進めることが、改善の可能性として重要であると考えている。

**システムズエンジニアリングにおける能力・機能の向上：**システムズエンジニアリングにおける企業自体の能力・機能の向上方法に関して、様々な回答が得られた。しかし、図 13 から分かるように、大部分は内部および外部のトレーニングプログラムを利用することに依存している。しかも、ほぼすべての企業がそのようなトレーニングプログラムを社内ですべて実施している。一方、同業者間または研究者との間での知識や経験を交換したり、傾向や解決法について議論

したりする目的で、システムズエンジニアリング関連のコンファレンスに参加する  
という答えが、全回答者については 50%を超え、大規模企業については 60%  
を超えた。

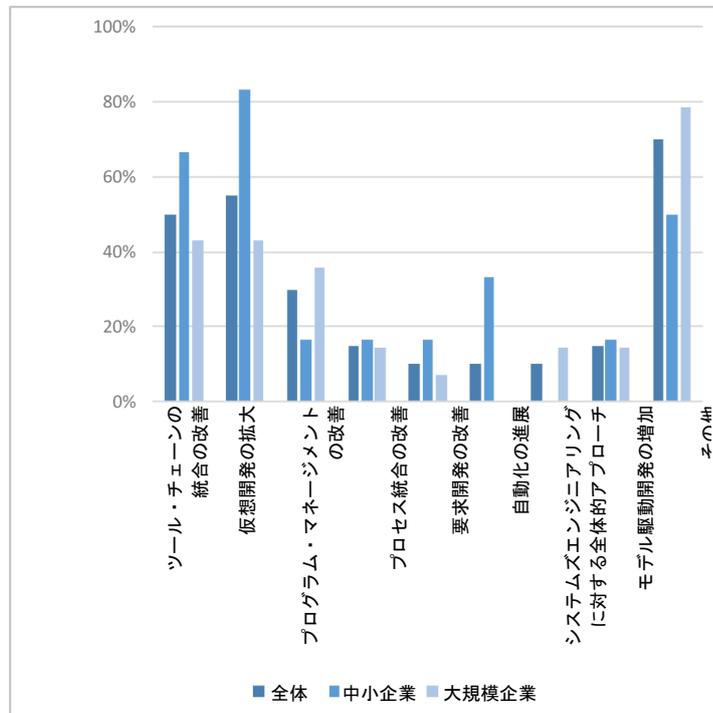


図 12 : システムズエンジニアリングに関する改善の可能性が見られる分野

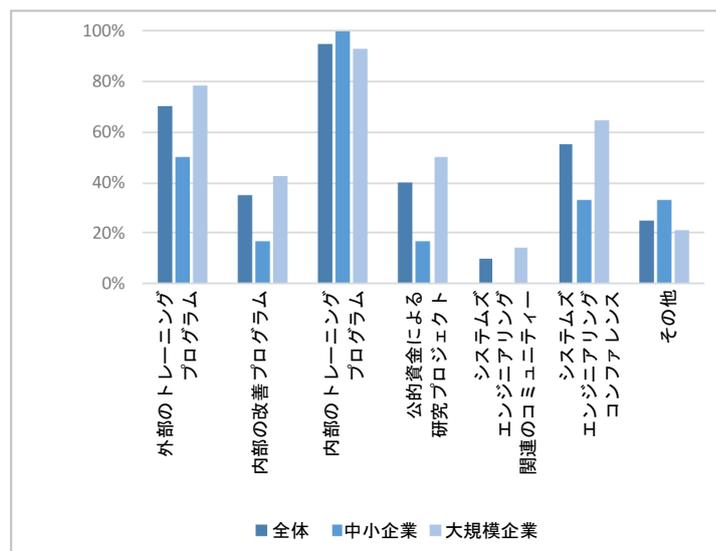


図 13 : システムズエンジニアリングにおける能力・機能の向上方法

### 3.5 潜在的なリスクと限界について

本項では、今回実施した調査・分析で採用した方法と実施者に関連する、潜在的なリスクと限界について要約する。

(1) サンプルの規模：実施したインタビューの合計数はかなり少ない。そのため、結果の一般化可能性が制限されるという面もあり、回答間の関係を分析する可能性が制限されるという面もある。しかし、少ないインタビュー数でも、1つ1つの回答により重点を置くことができ、それぞれのケースについてより深い洞察が得られるという利点がある。

(2) 自己申告データ：本調査・分析の回答者には、特定の知識に関して、また所属する企業・組織／部署の特定の背景に限った経験について報告することも求めた。自分たちの直接経験に基づいて明示的に回答することとし、自分の職務以外や専門分野外で起こっていることについて憶測では回答しないよう依頼した。

(3) 質問表：インタビューを行う際の指針として使用した質問表は、本調査・分析の目標から体系的に作られ、Fraunhofer IESE の研究員による内部ピアレビューを受けたものである。最初の2回のインタビュー中に、質問の意図に関する何らかのヒントがないと答えづらい質問がいくつかあることが分かった。また、インタビューは当初計画していた（最長60分）よりも長くかかった。そのため、29の質問のうち13の質問に対して回答例（大部分は予想される選択肢を含む）を追加した。回答例は、以前に分析した関連調査と、過去のシステムズエンジニアリングプロジェクトでFraunhofer IESE 研究者が得た経験に基づいて作成した。ただし、被インタビュー者には与えられた回答例から選択するだけでなく、それにとらわれずに考えてもらいたいことも明確に伝えた。その結果、多くの回答が既存のリストに該当したため、回答のコーディングが簡素化された。これにより回答間の相互比較が容易になり、インタビューの実施時間も大幅に短縮できた。

(4) 信頼と開放性：フラウンホーファー研究機構（Fraunhofer-Gesellschaft）は、客観性、中立性、独立性を保ったパートナーとしてドイツ国内でよく知られている。このため、調査・分析に参加することについても、課題および解決法について率直に語ることにしても（少なくとも、企業のコアな知的財産を侵害することがない限り）一定の開放性が保証される。またインタビューを要請する際には、結果の使用目的を明らかにし、面談記録を分析前に匿名化すること、被面談者は面談記録を確認する機会があり、分析の一環としてインタビュー記録の利用を明示的に承認する必要があることについて明確にした。

(5) 言語：すべてのインタビューはドイツ語で行い、その後の分析のために英語に翻訳した。翻訳後、インタビュー内容は承認のために参加者に送付された。これにより、判断できる限りで翻訳の正確さをチェックし、インタビュー記録

が被インタビュー者の意見を正しく反映していることを確認する機会を被インタビュー者に与えることができた。さらに被インタビュー者は、特定の回答について説明および訂正を加える機会も与えられた。

## 4 本調査・分析で判明した重要項目

### 4.1 重要項目

インタビューから分析に基づいて導き出された重要な成果報告は、以下のとおりである。

**(1) 製品開発の傾向：**企業を動かしている主な要因は、システム要求が複雑になっていること（60%）と、顧客の需要によって製品の種類がかつてないほど増えていること（50%）である。これが、市場投入までの時間の短縮（約 55%）と併せて、現在のシステムズエンジニアリングへの多大な圧力となっている。今後は、専門分野を超えた開発が要因に加わり（20%）、その結果としてさらにプロジェクトの複雑化・多様化がますます進むと見られる。コスト圧力が増大し、市場投入までの時間が短縮される傾向は、今後も続くと予測される。

**(2) ソフトウェアの重要性：**回答者の約 70%がハードウェア開発のみに由来する業種であると答えたにも関わらず、企業の 85%以上がその製品ではソフトウェアが主要な役割を果たしていると答えている。また、85%が開発予算の 30%以上（最大では 90%）をソフトウェア開発に費やしていると述べている。回答者の半数以上が、これは今後 5 年以内にさらに増加するという予測している。

**(3) システムズエンジニアリングの重要性：**1（重要でない）から 10（存続のために不可欠）までの範囲で、システムズエンジニアリングの重要度の平均値は 7.6 である。システムズエンジニアリングは今でもすでに非常に重要だが、重要度は今後 5 年以内に 8.7 にまで上昇する見込みである。ほとんどの回答者が、システムズエンジニアリングの重要性が高くなる理由は製品がさらに複雑化・多様化することと、顧客が求める製品品質が高くなっていることであると述べている。これは、システムプラットフォームとシステム統合に関する要求に特に当てはまる。

**(4) システムズエンジニアリングの課題：**回答者の 80%が、変化に対応できる組織への改革が第 1 の課題であると答え、複雑な要求およびインターフェースの管理（特にシステム・オブ・システムズ：SoS）がそれに続く。加えて、将来におけるシステムズエンジニアリングの課題としては、人材管理、組織およびシステムズエンジニアリングに向けたデータ管理、情報管理を含めた組織プロセスが挙げられている。

**(5) システムズエンジニアリング・プロセス：**大規模企業は基本的に ISO/IEC 15288 および 12207 のすべてのプロセス分野に取り組んでいるのに対し、中小企業は明らかに技術的プロセスと実装プロセスに重点を置いている。企業が

準拠している標準は、ISO 9001 などのごく一般的なアプローチを除けば、かなりドメイン特有である。大規模企業の 40%は明示的に ISO/IEC 15288 を挙げている。適用しているプロセスモデルに関しては、45%を超える大規模企業および中小企業がアジャイルモデルに従っていると述べたのに対し、50%を超える大規模企業はウォーターフォールモデルまたは反復型ウォーターフォールモデルに従っている。さらに、80%を超える大規模企業が、標準プロセスの様々な派生を挙げている。

**(6) 複数のステークホルダー：**システムズエンジニアリング・プロセスには、規模に関わらず全企業を通じて多様な専門分野と、それに対応するステークホルダーが関与している。しかし、ハードウェアエンジニアやソフトウェアエンジニアといった旧来の開発の専門分野は、企業内でいまだに「隔離された」専門分野として見なされている。「システム・エンジニア」という特定の役割は、大規模企業でしか定義されていない。大規模企業の 85%では、様々なステークホルダーと専門分野が、定義されたプロセスに従って連携および統合されている。中小企業の場合、この数字が 20%未満にとどまる。その代わりに、小規模な企業の場合は個人間のコミュニケーションが好まれている。企業の 60%~70%が、協力の目的で共同チームを作り、共同のワークショップや会議を実施している。

**(7) 外部サプライヤー：**約 60%の企業では、外部ソースから供給される製品部品は 25%未満にとどまっている。その一方、3 分の 1 の企業が最大 50%の部品の供給を外部サプライヤーに頼っている。外部供給される製品部品の平均的な割合は、全企業で見ると約 25%である。外部供給されるコンポーネントの知的財産に関する重要度の平均は、1（重要ではない）から 10（非常に重要である）までの範囲で 3.5 である。

**(8) システムズエンジニアリングの実践：**すでに確立している実践のうちで企業が広く（50%近く、またはそれ以上）適用しているのは、モデル駆動開発、要求開発、テスト駆動開発、および検証と妥当性確認に関連する手法、技法、アプローチである。その他の実践として 2 企業以上から挙げたのは、統合されたツール・チェーンと仮想エンジニアリング、全体的なシステムアーキテクチャなどである。大規模企業では、モデル駆動開発ならびにシステムの検証と妥当性確認に重点を置いているが（それぞれ 60%と 80%が抽出）、中小企業の約 80%は確立した実践の最上位としてテスト駆動開発を挙げている。

**(9) 影響を受けるプロセス：**技術的プロセスとソフトウェア実装開発のプロセス分野（ISO/IEC 15288 ならびに 12207 による定義）がシステムズエンジニアリングの実践において特に影響を受けていると、回答者の大多数が述べている。

**(10) 仕様言語およびツール：**80%を超える回答者が、主要な仕様言語として UML を挙げた。大規模企業では、システムモデリングへの特化した言語として SysML を使用する傾向にある。また、ドメイン特有の言語も広く挙げられた。

言及されたシステムズエンジニアリング・ツールの50%以上は、システム全体またはシステムの一部としてソフトウェアの様々な側面をモデリングすることに関連している。さらに、30%が要求を、40%がシミュレーションツールを関連ありとして挙げた。また、回答の10%近くが自社開発のツールを挙げており、これらのほとんどがシミュレーションの分野で使用されている（約5%）。その上、近い将来に対応予定の技術分野として、40%近くが非形式的なテキストに代わり、形式手法とモデルベースのシステム開発アプローチを仕様書に採用することを挙げた。

**(11) 改善の可能性：**システムズエンジニアリングの最大の改善の可能性は、仮想開発の増強と、使用されるツール・チェーンの統合の改善にあり、どちらについても回答者の50%が言及している。これらの要求は中小企業の場合にさらに大きいことが見受けられる。大規模企業の40%近くが、プログラム・マネージメント（プロジェクト・ポートフォリオ管理とも言う）の改善についても言及しておく必要がある。中小企業の40%近くは、自動化がさらに進むことが改善の可能性として重要であると考えている。

**(12) システムズエンジニアリングにおける能力・機能：**組織／部署の大部分が、システムズエンジニアリング関連の能力・機能の向上を図る上で、内部および外部のトレーニングプログラムに依存している（それぞれほぼ100%と60%超）。また、システムズエンジニアリングコンファレンスの参加も、全回答者の50%超、大規模企業の60%超によって挙げられた。

## 4.2 推奨事項と活動分野

本調査・分析で得られた12の重要項目から、システムズエンジニアリングに取り組む企業に対する推奨事項と活動分野が導き出された。これらの推奨事項および対策は、本調査・分析の結果によって導き出されたものだが、同じ目標を達成するには別の戦略も存在し、ある程度は主観的である点に留意してもらいたい。

ここでは推奨事項と活動分野を、組織開発に関連するものと、企業のシステム開発方法に技術的に関連するものに分けている。

### 4.2.1 組織開発

**(O1) 変化に対応できる組織への変革戦略：**企業の80%が、システムズエンジニアリングの主要な課題は企業における組織改革であると答えている（重要項目(4)参照）。したがって、システムズエンジニアリングの課題に取り組むためには、どのような組織構造およびプロセスが最適かをオープンに考えることが重要である。特に、合意形成し、改革の動機づけと伝達を的確に行って、その変更をどう実施するか慎重に計画するためには、あらゆるステークホルダーをそのプロセスに取り込むことが重要である（重要項目(6)参照）。

**(O2) システムズエンジニアリング能力**：様々なシステムズエンジニアリング項目に関する内部トレーニングプログラムの作成および外部トレーニングプログラムの購入は、ほとんどの企業で義務化されている（重要項目(12)参照）。また、最近の開発情報を得たり推奨／禁止事項に関する経験を共有したりするために、システムズエンジニアリング関連のコンファレンスに参加し、それぞれのコミュニティの積極的なメンバーになることを推奨する（重要項目(12)参照）。

**(O3) ソフトウェアエンジニアリング能力**：元々の業種がハードウェア開発寄りであるにも関わらず、企業の 85%以上が製品でソフトウェアが主要な役割を果たすと答え（重要項目(2)参照）、またこの数字は今後も伸びることが予想されるため、企業が適切にソフトウェア開発能力を構築する、あるいは維持することは重要である。その程度は、製品がソフトウェアに依存する度合いと、企業の主要な IP（知的財産）および USP（独自の売り）がどこに存在するかによって異なる。IP/USP がソフトウェアそのものに存在する、またはソフトウェアになりつつある場合、ソフトウェア開発の分野で自らのリソースを構築することは必要である。また、ソフトウェアが1つの目的を達成するための手段にすぎない場合は、外部のソフトウェアサプライヤーおよびパートナーを管理するための能力を構築することは、有効なことではある（重要項目(7)参照）。

**(O4) プロジェクト・ポートフォリオ管理**：大規模企業が、改善のための着目点として挙げているように（重要項目(11)参照）、プロジェクトのポートフォリオ全体の管理と、プロジェクト間の相互連関および依存関係に特に重点を置くべきである。

#### 4.2.2 技術開発

**(T1) システムズエンジニアリングの統合アプローチ**：システムズエンジニアリングの統合アプローチ：新製品を市場投入するまでの時間が短縮されるのと同時に、製品の複雑化・多様化が進んでいるため（重要項目(1)参照）、顧客に効率的かつ効果的に価値を提供することが重要である。システムズエンジニアリングは、そうした問題に対処するため、特にシステムプラットフォームおよびシステム統合の場合に重要性が高い（重要項目(3)参照）。これには、関与するすべての専門分野を巻き込んで十分に統合された、また整合されたアプローチが必要となる（成果報告(6)参照）。特に、技術的プロセスと実装プロセスにおいては、システムズエンジニアリングがどのような影響を持つかについて（重要項目(9)参照）、特効薬となるようなアプローチは存在しないが、個別の企業のニーズに最も適するように調整されたプロセスはどうあるべきかについて（重要項目(5)参照）、注意深く検討する必要がある。

**(T2) システム要求開発**：時間と共にシステム要求はますます複雑化し、製品の多様化も増大している。実際、近い将来において専門分野を広く超える開発が定着するにつれ、この傾向はさらに強まるであろう（重要項目(1)参照）。それ

に伴い、企業はシステムレベルでどうやって要求を引き出し、開発し、長期にわたり系統立って管理するかについて考えざるを得なくなる。また、どうやって下位レベル（特にソフトウェア）の要求に分割するかも考えなくてはならない（重要項目(2)参照）。

**(T3) モデル駆動システム開発**：本調査・分析では、システムのモデル駆動開発が企業にとって重要な実践と見なされていることが確認された。大規模企業ではすでに最低でも部分的に実装しているか（重要項目(8)参照）、またこれを改善の可能性として重要と見なしている（重要項目(11)参照）。UML、SysMLなどは特に目立っているが、モデリング言語の実際の使用状況は一様ではない。ツールサポートの面では、様々なツールも挙げられた（重要項目(10)参照）。そのため、企業はシステム仕様のどの側面をモデリングするか、適切な範囲でどんな言語とツールサポートを利用できるかを注意して評価する必要がある。ここでのツール選定には、開発プロセスのツール環境においてシームレスな統合を実現する上で適切かつツール提供のインターフェースを検討すべきである（重要項目(10)および下記 T6 参照）。

**(T4) システムの検証と妥当性確認**：企業は、システムの検証と妥当性確認、ならびに特にテスト駆動開発のために適切な技法および手法の確立を考慮する必要がある。多くの企業にとって重要であると見なされているからである（重要項目(8)参照）。また、開発プロセスによって、システムの検証と妥当性確認を、常にシステム要求と適切に関連付ける必要もある。

**(T5) 仮想システムズエンジニアリング**：製品がますます複雑化し（重要項目(1)を参照）、複数の専門分野による開発が進むにつれて（重要項目(6)参照）、物理的に様々なシステム部品を構成することは難しくなり、コスト負担も非常に重くなる。そのため、企業は確固としたモデルに基づく仮想システム開発について検討する必要がある。将来、これは開発速度を上げるという点で主要な改善の可能性があるとして見なされている（重要項目(11)参照）。一部の企業はすでにシステムの検証と妥当性確認用に独自のシミュレーションツールを導入または開発している（重要項目(10)参照）。

**(T6) 統合されたシステムズエンジニアリングのツール・チェーン**：前述したように、企業においてはシステムズエンジニアリングのために多様なツールが使用されている。さらに、企業は特定のタスクのため、また既存のツールの不足を克服するために独自のツールを開発している（重要項目(10)参照）。ツール・チェーン統合の改善は、主要な改善点である（重要項目(11)参照）。このため、特にシステムズエンジニアリングを始める際、企業はツールの相互運用性と、ツール・チェーンをできるだけ統合することに特に重点を置くべきである。

## 5 抽出された産業界での実践および事例

本項では、本調査・分析の結果に基づき、企業において選択したシステムズエンジニアリングのベストプラクティスをどのように実装するかについて、さらに詳細を示す。すでに確立した実践のうちで、企業が広く（50%、またはそれ以上）適用しているのは、以下の分野に関連する手法、技法およびアプローチである。

1. モデル駆動システム開発
2. システム要求開発
3. システムの検証と妥当性確認

以下は、システムズエンジニアリングで最も大きな改善が期待されるとして、企業の50%が挙げている分野である。

4. システムズエンジニアリング・ツール・チェーンの統合
5. システムの仮想開発

1.2.3 はインタビューした企業の半数以上で少なくとも部分的には適用されていることが証明されたのに対し、4.5 はまだ実際の適用および実装の初期段階であるが、システムズエンジニアリングにとって大きな可能性を持つと見なされている。したがって、以下の各項では、5つの実践分野のそれぞれについて概要を述べ、手法、技法ならびにアプローチの具体例のいくつかを挙げた上で、実際の環境における実装および適用に関する産業事例を示し、いくつかの推奨事項と教訓について要約する。

### 5.1 モデル駆動システム開発

近代的システムの開発は、ますます困難で多くの課題を抱える傾向にある。自動車、航空電子機器などのドメインでは、システムの適切な実行を保証するためにハードウェアとソフトウェアの間に高度な整合性が求められている。例えば、民間航空機には、地表接近、ナビゲーション、エンジンコマンドなどを制御するシステムが搭載されている。そのため、確実にシステムのあらゆる側面が適切に記述され、理解されていることが重要である。

*モデル駆動システム開発*は、モデルの洗練化に基づいたシステム開発法である。通常、このモデルの洗練化は異なる抽象レベルから始まるが、システムが直ちに実装可能になるか、あるいは理想としてはモデルから自動的にシステムが生成されるまで、詳細さの度合いを高めていく。通常、モデルベース要求開発から、モデルベース設計、モデル駆動実装（モデルからのコードの抽出によって部分的に自動化が可能）、これらのシステムの検証（フォルトツリーなどのモ

モデリング技法を使用)まで、実質的な開発専門分野のすべてを支援する、一連の統合されたモデリング技法とツールが提供されている。

**モデル駆動ソフトウェア開発**は、全体的なモデル駆動システム開発の一部で、モデル駆動要求開発、モデル駆動設計、モデルからのコード生成、モデル駆動テスト、モデル駆動によるソフトウェアの改修など、ソフトウェア開発活動全体を通じて様々な技法が取り込まれている。

モデル駆動システム開発は、現代のシステムの複雑さを管理する重要な技法である。これらのシステムの検証(フォルトツリーなどのモデリング技法を使用)など、実質的な開発専門分野をすべて支援する、一連の統合されたモデリング技法とツールが提供されている。

### 5.1.1 アプローチ例

組込みシステムの開発に関する SPES 2020 手法 (Pohl, Achatz, & Broy, 2012) は、モデル駆動システム開発のために確立された手法である。SPES (Software Platform Embedded Systems、ソフトウェアプラットフォーム組込みシステム) 2020 イニシアチブは、航空電子機器、自動車、医療、エネルギーなど、様々なドメインの産学連携による共同研究開発プロジェクトである。SPES 2020 モデリングフレームワークでは、モデル駆動システム開発をアーキテクチャ上の 4 つの観点に整理している。

**要求の観点**：システム要求の抽出、文書化および管理における要求開発プロセスの支援に照準を定めた観点。SPES では、要求の抽出は、ユーザー、ステークホルダー、およびそのシステムと何らかの形で相互作用する外部システムなどのシステム・コンテキストの識別から始まる。これらのエンティティは、要求の観点から、システムをブラックボックスとして表し、システムと環境との相互作用を文書化するシステム/コンテキストダイアグラムの形で文書化される。このダイアグラムは、従来の RE 技法 (シナリオのようなもの) を補完し、すべて要求の観点で文書化される機能と品質に関する要求、ビジネスドライバー (促進要因) および制約 (法的制約など) を抽出するのに役立つ。

**機能的観点**：どんなシステムも一連の機能を持ち、それぞれの機能がシステムのユーザーに特定のサービスを提供する。機能的観点の主な目的は、これらの機能を識別し、形式化することである。SPES 手法では、ユーザー機能を識別する際の開始点として、要求の観点から見たコンテキストダイアグラムおよびシナリオが使用される。そして、機能的観点で、機能的データフローダイアグラムなどによって入力/出力動作を定義することによって、これらのユーザー機能を形式化する。また、機能的観点は、機能間の依存関係が識別され、従属機能からユーザー機能への洗練化が行われるところでもある。

**論理的観点：**機能的モデルを確立した次の段階は、ソフトウェアまたはハードウェアによって（あるいはその混合として）実装する機能を特定することである。つまり、論理的モデルでは、（機能的観点で識別したとおりに）システムの機能を、通信および提携コンポーネントのネットワークに分解する方法が定義される。機能的観点が理想的には課題指向のみであるのに対し、論理的観点は設計の決定が行われる出発点であり、解決指向である。

**技術的観点：**この観点では、ハードウェアおよびソフトウェア要素が論理的コンポーネントを具象化する実装エンティティで表現される。詳細なソフトウェア設計のほか、この観点にはバス、センサー、アクチュエータのようなハードウェア要素のネットワークを示すハードウェア・ネットワークビューと、ハードウェア・エンティティに対する論理的ソフトウェア・コンポーネントの展開戦略を示す展開ビューが含まれる。

前述した各観点は、要求の観点から技術的観点へのウォーターフォールのようなプロセスを示しているように見えるが、実際には SPES 手法はもっと反復的である。通常、機能的モデルで形式化され、論理的／技術的レベルで具象化される要求を、かなり抽象的なレベルで抽出することから始まる。論理的レベルで識別されたコンポーネントはそれぞれ単独で、それ自体、より具体的な要求、機能、論理的および技術的解決法を有する、まだ検討過程にあるシステムとして見なすことができる。

## 5.1.2 実施事例

この手法は、様々なドメインの製造企業で適用に成功している（Pohl, Achatz, & Broy, 2012）。ここでは、宇宙からのガンマ線を観測する大型望遠鏡アレイの制御ソフトウェアの開発における実施事例について説明する（Achary & Actis, 2013）。アーキテクチャ・モデルは、SPES の観点を支援する洗練された UML プロファイルを使用して作成された（Kuhn & Antonino, 2014）。この研究は文献（Oya, et al., 2016）で詳細に説明されている。

チェレンコフ望遠鏡アレイ（CTA）は、チェレンコフ・ガンマ線望遠鏡を2つの巨大なアレイとして建設する計画である。幅広い天体物理学コミュニティのためのオープンな国際天文台として機能し、非熱的高エネルギー宇宙に対する深い洞察を得られることが期待されている。チェレンコフ望遠鏡システムは、地上に設置されている望遠鏡に搭載されているカメラで検出可能な、いわゆるチェレンコフ光シャワーを放射する粒子カスケードをガンマ線が発する効果を利用する。CTA の目的は、科学上の大きい推進力となる次の3つの主要テーマに大別される。すなわち宇宙線の起源と宇宙におけるその役割を理解すること、ブラックホール周囲の粒子加速の性質と種類を理解すること、そして標準モデルを超えて物質の究極の性質と物理法則を探ることである。

CTA におけるアレイ制御およびデータ取得 (ACTL) プロジェクトでは、CTA 計装機器からデータを取得し制御するソフトウェアを開発する。その目的は、ソフトウェア開発、調整プロセス、意思決定の間で緊密な統合を可能にする単一のアーキテクチャ・モデルを作成し保守することである。まず、このモデルは、例えば作業分解図 (WBS) の生成、作業量の見積り、リスクの評価、優先順位の設定と未計画作業の特定に必要な情報の提供など、プロジェクト管理組織を管理するための主要な情報を提供する。また、開発するソフトウェアのコンテキストを理解できるように、チームの開発者向けに仕様書や契約書を提供するので、開発者文書を自動生成するために使用される。さらに、このモデルは、開発者によって提供されるコードをテストするための検証と妥当性確認プロセスを推進するものでなければならない。最後に、ワークパッケージの内外で、要求、決定事項および適用された技術的な解決法を伝達できるようにする。

### 5.1.3 教訓と推奨事項

SPES 2020 (Pohl, Achatz, & Broy, 2012) に基づいて、様々なドメインでの実用性評価から推奨事項と教訓を引き出した。

- SPES 手法は、ドメインごとに統合度は異なるが、すべてのエンジニアリング・ライフサイクル段階において、追跡可能でシームレスな支援が実現された。自動車分野では最も高いレベルの統合が得られた。
- 完全性、一貫性、安全性または追跡可能性に対するユーザーの期待に対応するため、初期段階でのシステムプロパティの検討および妥当性確認が可能であった。
- この手法は、安全性、標準の遵守、保証可能性へのニーズに対応する。例えば、自動車ドメインでは、論理的アーキテクチャによって AUTOSAR アプリケーション・コンポーネントへの自動変換が可能になった。統合された設計および安全性モデリングでは、エンジニアが同じモデルでシームレスに作業できることが示された。

さらに、SPES 手法による支援が得られるよう統合された開発は、組込みシステムの開発に不可欠であると結論付けられた。

## 5.2 システム要求開発

複数の研究機関が要求開発（RE）という用語の定義を提供している。国際要求工学委員会（IREB：International Requirements Engineering Board）では、次のように定義している。「要求開発とは、要求分析および管理への体系的、方法的アプローチである」(Sophist)。IEEE では RE を次のように定義している。「要求開発とは、ソフトウェア主体のシステムの要求特性および制約の管理と、その環境で達成されるべき目標に関わるシステムズエンジニアリングの分野である。システムの課題分析段階から実装および保守段階まで、このような側面が考慮される。各種ドメインの課題は異なるため、行政向けソフトウェアからワークフロー・システム、グループウェア、組込みシステムおよび制御ソフトウェアまで、多岐にわたる」。

RE の目標は、適切な要求を開発し、開発中にリスクおよび品質に関してこれらを管理することである。RE は、ライフサイクル全体における橋渡しの役割を果たし、製品またはプロジェクトの成否を決める、システム内の専門分野であり、ソフトウェア開発である。その専門性と体系的なアプローチから、開発の専門分野のひとつであると言える（Ebert, 2014）。

### 5.2.1 アプローチ例

RE に使用されるプロセスは、適用するドメイン、関与する人々、および要求を開発する企業に応じて大きく異なる。以下の一般的な活動は、すべてのプロセスに共通するものである（Software Engineering Institute, Carnegie Mellon）。

- 要求の抽出：システムに関するユーザーのニーズと制約を発見、評価、文書化および理解するプロセス。
- 要求分析：ユーザーのニーズと制約を洗練化するプロセス。
- 要求の妥当性確認：システム要求が完全で正しく、一貫性があり、明確であることを保証するプロセス。
- 要求の仕様：ユーザーのニーズと制約を明確かつ正確に文書化するプロセス。
- 要求の変更管理：要求開発活動（つまり、抽出、分析、仕様および妥当性確認）をスケジュールリング、調整ならびに文書化するプロセス。

これらの一般的な活動には、それぞれ、プロセスの実行に適用できる多くの技法が存在する。『Requirements Engineering: Best Practice』（Fricker, Grau, & Zwingli, 2014）に、ベストプラクティスの概要に関する良い記述がある。

RE は、製品またはプロジェクトの非常に重要な部分である。要求分析が不十分なためにプロジェクトが失敗することは、多くの研究で示されている (The Standish Group, 2014)。最新の RE 技法も多く存在するが、いまだに適切な実装と満足が行く実行例が得られていないのは明らかである。頻繁に観察されている RE プロセスの欠陥に関する良い概要が、『Trends & Benchmark Reports on Software Development』(SwissQ & University St. Gallen, 2014) に示されている。典型的な欠陥は、コミュニケーションにおける誤解、絶え間ない要件の増加または変更、あるいは時間のプレッシャーである。

このような不備を克服するために、近年モデルベース要求開発という新しい手法が開発されている。モデルベース要求開発 (MBRE) は、自然言語のみでなく、形式てき、半形式てき、非形式的なモデリング言語を使用して、要求および関連するビジネスならびに開発情報の収集、整理、構造化を行うアプローチである (Teufl, Khalil, & Mou, 2013)。これらは主に、対象物を、現実世界と比較してずっと簡潔にかつ記述的に表したものである。MBRE のモデルでは、自然言語のみで文書化するよりも解釈の余地が少ない (たいていの場合はグラフィカルな) 言語で、要求および要求と他のアーティファクトとの関係を文書化することが可能になる。さらに、データについて抽象化および様々な観点を提供できるため、すべてのステークホルダー間でコミュニケーションと議論が支援される。UML または SysML を使用したモデリングは、MBRE では一般的な実践である。

## 5.2.2 実施事例

ドイツ国防省は、効率的で均一的な製品取得および利用プロセスに関する指令を出した。顧客製品管理 (CPM : Customer Product Management) (Germany Federal Ministry of Defense, 2012) と呼ばれるこのプロセスでは、ドイツ軍の RE、調達および稼働中支援に関する手順が規定されている。システムのライフサイクルにおける初期の概念段階から廃止に至るまで、システムのライフサイクル全体を非常に高いレベルで記述している。この指令では、システムのライフサイクルを 3 つの主要な段階に分けて定義している。

- 分析段階 : 分析段階で最初の目的は、能力ギャップを識別し、そのようなギャップを埋めるための対策に優先順位を付けることである。この段階で、RE 活動と、システムモデリング活動の大部分が実行される。
- 生産段階 : 生産段階の目的は、ユーザー/オペレーターに適切で使用可能な製品およびサービスをタイムリーに提供することである。要求モデルとシステムモデルはさらに開発され、構築契約締結のための仕様のベースを構築する。
- 稼働段階 : 稼働段階では、意図された目的に沿った製品およびサービスの使用について規定されている。製品およびサービスが廃棄に至るまで現実的な条件で合法的に、安全かつ経済的に使用されることを保証するため、

運用可能性、能力ならびに即応性を維持および復元するためのあらゆる対策が実施されなければならない。

Fraunhofer IESE は、ドイツ国防省を代表する Federal Office of Bundeswehr Equipment, Information Technology and In-Service Support (BAAINBw) と共同で、新しいモジュール式多目的戦闘艦向けに、システムズエンジニアリング・アプローチを使用して分析段階と生産段階の一部を実施した。このプロジェクトは規模が非常に大きく、予算と日程が厳しく固定されているため、実用的なシステムズエンジニアリングのアプローチが適用された (Webel, et al., 2015)。

**分析段階 1:** CPM プロセスにおける分析段階の最初の部分に位置する RE 活動の結果、重要度または致命度に応じて優先順位を付けた、機能的要求の階層的カタログ（優先要求カタログ）を作成した。このカタログは、定義済みのオペレーションと使用条件、ならびに関連する使用プロファイルに基づいている。

- オペレーションのニーズのみを文書化し、技術的解決法は未解決としておくために、要求は機能的な形で形式化された。
- 共通の性能パラメータ（最上位レベルの機能性やオペレーションなど）に従って、要求は階層的に機能グループに分類された。
- 不達成の場合にプロジェクトを中止する基準を設けるために、要求に優先順位が付けられた。
- 要求は比較検討された後、各要求の企業能力への貢献度が文書化され、さらに、重要な要求とそれほど重要でない要求が区別された。
- すべての要求は、正当性、完全性、一貫性、起源までの追跡可能性、検証可能性、明瞭性など、IEEE 830:1998 ソフトウェア要求仕様に規定された品質基準を満たすために、品質保証プロセスを通して吟味された。

最後に、NATO アーキテクチャ・フレームワーク 3.1 版 (NATO) を翻案した Architecture Data Model of the Armed Forces (軍隊に関するアーキテクチャ・データモデル) に準拠するオペレーションアーキテクチャが開発された。このアーキテクチャ・フレームワークは SysML に基づき、防衛プロジェクトのニーズに個別に適応したドメイン固有の拡張によってモデリング能力を拡張する。分析段階 1 では、オペレーションアーキテクチャには主に以下が含まれる。

- タスクや活動、オペレーション要素、情報交換に関する説明を備えた企業能力に関する高レベルの理解が示されている能力ビューモデル
- ユーザーの観点（シナリオ）から見た要求またはオペレーションを全般的に反映するオペレーション・ビューモデル

- システムの実際の、または提案された実装および記述と、システム・コンテキストが示されているシステム・ビューモデルについてのトップレベルの一覧

RE とシステムモデリングを統合したアプローチは、特にツール・チェーンの統合が保証されるように開発されたため、要求とシステムモデルが同期する。例えば、機能的要求は、アーキテクチャ・モデルのシステム・ビューで定義されている最初のシステム・コンポーネントにマッピングされたシステム機能に変換された。

また、スタイル・ガイダンスは、適切な言語スタイルで要求を指定するのに役立ち、要求定義の具体的手順およびツールに関する助言を与えてくれる。

**分析段階 2 :** CPM プロセスにおける分析段階の 2 番目の部分では、さらに機能的要求と非機能的要求が追加された。要求モデルは、技術、物流、製品ライフサイクル、プロジェクトおよびリスク管理、品質管理など、様々な種類の要求が最大 10,000 件を超えるまでに増加した。システム構造が開発され、すべての要求が一方はシステム構造に、もう一方は追跡可能性を保証するために分析段階の最初の部分で構築された機能的要求モデルにそれぞれ関連付けられた。

このアーキテクチャ・モデルを、Architecture Data Model of the Armed Forces (軍隊に関するアーキテクチャ・データモデル) に準拠するシステムアーキテクチャによって拡張した。システム・ビューは、要求モデルのシステム構造と同期する、より詳細な構造ビューによって拡張された。システム・コンポーネント間の重要なシステムの相互接続性とコンテキストが追加された。

この段階では、RE とシステムモデリングの統合されたアプローチがさらに発展し、技術的要求がシステムモデル内のシステム・コンポーネントへマッピングされ、分析段階中に同期が維持されるようになった。

**生産段階 :** CPM プロセスに従って生産段階に入ると、実現の順序を決める土台となるシステム仕様の開発を目的に、開発・製造パートナーと交渉しながら、要求モデルとシステムアーキテクチャについて、さらに明確な細部まで指定した。この段階は現在も進行中である。

要求の定義に関する指向と、モデル駆動設計に関する指向のどちらも、高度に統合されている。要求、オペレーションアーキテクチャ、システムアーキテクチャの全モデルにおいて、同一の機能的および非機能的要求を指定し、適切なモデル要素にマッピングしたことによって、システムのライフサイクルにおける全段階で追跡可能性を保証できるようになった。

これを効率的に実行するため、要求管理、システムモデリング、追跡可能性と影響分析、要求の優先順位付け、コミュニケーションとコラボレーションで構成してカスタマイズしたツール・チェーンが開発された。

RE とシステムモデリングの定義されたプロセスに従って、シームレスな相互作用を実現するため、統合されたプログラミングエンジンまたは追加のスクリプトにより、すべてのツールがカスタマイズおよび拡張された。さらに、プロジェクトの CPM プロセスに関与する 100 名を超えるステークホルダーが、正しいツールを使用して必要なデータを収集できるように、またこれらのツールを正しい方法で使用して処理を行い、プロセスの次の段階に進むことができるようにするため、いくつかの操作説明書が作成され、チュートリアルが何回か実施された。

### 5.2.3 教訓と推奨事項

このプロジェクトにおける顧客との共同研究で得た教訓に基づき、同様の産業プロジェクトにおける RE 手法と RE ツールの導入に関して、IESE の観点から以下の推奨事項を提案できる。

- 要求の抽出：適切な要求の抽出技法の選択は、プロジェクトに影響する要因、特に必要とされるステークホルダーが対応可能かどうか、またはスキルがあるかどうかに基づく。これらの要因には相当のリスクがあり、それへの対応が求められる。『RE-primer (RE の手引き)』(SOPHIST, 2016) には、どの影響要因にどの抽出技法が最適であるかを考える際に非常に良い手がかりとなる選択基準表が含まれている。
- 要求分析：ユーザーのニーズおよび制約を洗練化する分析プロセスは、総合プロジェクト計画で考慮しなければならない。多くの場合、ユーザーのニーズおよび制約を洗練化すると、追加作業が発生する。この場合、元のステークホルダーが再び関与し、提案された変更に同意する必要がある。このプロセスは、RE ツールを使用して、再考が必要な要求に関する特定の選択肢および観点を作成することによって非常に強力に支援することができる。前述のプロジェクトでは、使用された RE ツールは高度に設定可能で、スクリプトで拡張することにより、これらの特定の選択肢およびビューを作成できたことは非常に貴重であった。
- 要求の妥当性確認：要求の完全性、正当性、明瞭性および一貫性を保証するプロセスは、IEEE:830 標準で定義されているものなど、標準化されている品質基準に合わせる必要がある。これは、個別のプロジェクト／製品固有のパラメータを自動的に確認するため、属性ごとの制約の定義や、スクリプトまたはプログラムによる拡張が可能な RE ツールを使用することによって多大な支援が可能である。品質欠陥が発見された場合のために、常に要求の再考段階をプロジェクトのスケジュールに含めなければならない、

元のステークホルダーもこれらの問題を解決するために再び関与する必要がある。

- 要求の仕様：ユーザーのニーズおよび制約を必要なすべての属性と共に適切な方法で明確かつ正確に文書化することは、RE ツールによって支援可能である。このツールは、すべての要求開発プロセスの包括的な支援を提供するため、ツール・チェーンを構築するスクリプティングエンジンを使用して調整または拡張可能でなければならない。
- 要求の変更管理：要求開発プロセスに関するプロジェクト／製品固有のプロセス、およびこれらのプロセスを支援するためのすべてのツール（分析ツール、インポート／エクスポートツール、モデリングツール）の適用方法を定義することは、処理対象となるすべてのデータを制御し続けるために必要である。特にこれらのプロセスを支援するために設定されたコラボレーションツールは、これを実装し、定義されたプロセスにすべてのステークホルダーを従わせるのに役立つ。さらに、前述した大規模プロジェクトで得られた考察で特筆すべき点は、他のプロジェクト／製品アーティファクトへの影響を考慮に入れた上で要求に関する意図的な変更を確認してからリリースする変更管理委員会をREプロセスで実装することで、要求データベースに意図しない変更が入力されるのを回避するのに役立つ点である。

### 5.3 システムの検証と妥当性確認

組込みシステムドメインの多くで、技術機器内のソフトウェアが広く利用されるようになったことが、新しいイノベーションの主要な原動力になっている。ソフトウェア制御部品が複雑化することで、統合されたシステムの品質特性にソフトウェア欠陥が及ぼす影響はさらに増大する。指定された品質要求と顧客のニーズを満たすため、システムの検証および妥当性確認プロセスの有効性と効率は、ソフトウェアおよびソフトウェア集約的システムの開発において重要な役割を果たす。

要求、設計モデル、プログラムコード、統合された電子制御ユニットなど、開発活動の様々な出力アーティファクトの分析、検証と妥当性確認のために、革新的なモデルベース品質保証技法が開発されている。最近の傾向としては、仮想検証技法などの早期品質保証（Feth, Bauer, & Kuhn, 2015）と、数理モデルおよび適切なツール・チェーンを使用して高度に自動化された検証と妥当性確認に重点が置かれている。

#### 5.3.1 アプローチ例

例として、統合品質保証（InQA：Integrated Quality Assurance）のアプローチが挙げられる（Elberzhager, Rosbach, & Bauer, 2014）。これは、様々な形式検証、レビュー、テスト技法など、静的品質保証の技法と動的品質保証の技法を組み合わせた体系的な手法である。複数の技法を組み合わせることの利点は、

コンポーネントの範囲や欠陥分布など、適用された1つの品質保証技法を指針として利用し後続の技法を微調整することによって品質保証プロセスの向上を図ることができるという、相乗効果が得られる点である。

InQA アプローチは、定義、較正および適用の3つの主要な段階で構成される。定義段階では、テスト・エンジニアの経験、利用可能な労力、開発するソフトウェアの種類と成熟度など、品質保証技法を統合して適用する目的とコンテキストを定義する。第2段階の較正は、継続的に実施することで、スマート統合の有効で成熟した知識ベースを得ることができる。最初に、検討した品質保証技法を選択し、注文を定義し、適用のレベル（コンポーネントやシステムのレベルなど）を決定する。次に、データと、欠陥数、コンポーネントの複雑性、労力数など考慮すべき測定基準を決定する。第3段階の適用では、仮定を検証し、そのデータを利用して品質保証活動中にどのような相乗効果を引き出せるかを評価する。

### 5.3.2 実施事例

輸送ドメインの技術的システムに InQA アプローチを適用できるかどうか、その影響はどうかをヨーロッパの大規模なプロジェクト MBAT (Klås, Bauer, Dereani, Söderqvist, & Helle, 2015) で評価された。評価プロジェクトでは、技術的システムの早期品質保証のために InQA アプローチが調整・適合された。評価で用いられた技法としては、システム・シミュレーションによるシステム設計アーティファクトの検証と妥当性確認と、モデルベースの品質保証があった。後者は、コード構造またはシステム構成のモデルなどの特性と、データフローまたは動作モデルなどの機能テストケースの導出を検証するために具体的な数学モデルで機能する技法である。

MBAT プロジェクトでは、8 か国から 39 企業・組織の研究パートナー、ツール・ベンダーおよび産業ユースケース・プロバイダーが参加し、合同で自動車、航空電子機器、鉄道システムなど様々な輸送ドメインの安全関連のソフトウェア集約的システムに関する品質保証技法と、対応するツール・プラットフォームの調査ならびに開発を行った。Daimler や Volvo のような各種最先端企業から 13 件の業界ユースケースが実施され、評価された。ユースケースには、コンテキスト、設定、およびそのテクノロジーで対応すべき問題が記述され、対応する事例研究を実施することで定量的フィードバックを得る機会が与えられた。これらのユースケースには、システムおよびソフトウェアの品質管理プロセスの様々な段階が含まれ、同時に機能面の正当性、時間効率性、乗用車に関する規格 ISO 26262 (ISO, 2011) といった標準への準拠など様々な品質特性も含まれており、技法の適用範囲が評価された。例えば、Daimler (乗用車のライト制御サブシステム) と Volvo (ブレーキ・バイ・ワイヤ・サブシステム) による自動車のユースケースは、ソフトウェア設計アーティファクトとプログラムコードの検証と妥当性確認に関するものであった。

評価では、業界ユースケースの関連目標と評価面が決定され、洗練化された。主な目標は、検証と妥当性確認のコスト、欠陥コストおよびシステム品質で構成された。すべてのユースケースと、InQA など基礎となる品質管理アプローチの組み合わせが、これらの目標について数回繰り返して評価された。

### 5.3.3 教訓と推奨事項

検証と妥当性確認技法の適用コストは、13 件の事例研究で収集されたデータの検討から、平均 32%と、大幅に削減できる可能性が示された。後続の開発段階の残存欠陥によって発生するコストも、平均 27%削減できる可能性が示された。システム品質に関する目標は、テストの範囲やリリース後の欠陥といった、評価された副次的な目標が多岐にわたるため、コスト項目のように総括することはできなかった。ユースケースでは、この目標のすべての副次的基準が少なくとも 8%向上する可能性が示された。すべてのユースケースでコストおよび品質面で大きな向上が見られたので、InQA などの品質管理を組み合わせるアプローチの応用は今後きわめて有望である。

### 5.4 統合されたツール・チェーン

製造現場での実践では、一連の手法とツール関連の障害がシステムズエンジニアリング全般、特にモデルベース・システムズエンジニアリングを困難にする。

まず、ライフサイクル全体を通じてエンジニアリング・プラットフォームには、異種混合の様々なエンジニアリング手法、ツールおよびデータが関与する。第二に、例えば安全性が最重視されるシステムなど、開発プラットフォームと操作プラットフォームとの間のギャップを埋めることに対するニーズが高まっている。そうすることにより、人間参加型または仮想テスト、異種混合コ・シミュレーション、または大規模分散アプリケーションの監視と保守によって、大規模な開発組織の開発および意思決定プロセスを改善できる。第三に、今日の大規模企業における開発チームは分散型・多層型であり、複数の国とサプライヤーにまたがる。このような要因の共通の特徴は、現在の（まだ断片的な）ツール環境においてシームレスな相互運用が必要であるという点である。

#### 5.4.1 アプローチ例

自動車、航空宇宙、鉄道、医療分野にまたがる CRYSTAL (CRITICAL sYSTEM engineering AccELeration、重要システムズエンジニアリングの加速) プロジェクトでは、システムズエンジニアリングにおける手法およびツールの相互運用性を改善するために、幅広い業界の参加により様々なアプローチが研究された。このプロジェクトでは、各種適用ドメインにおける典型的なエンジニアリング手法やツール・チェーンの概要、相互運用性の課題の明確化、相互運用性の仕様 (IOS)、ツールのインターフェースおよびアダプタの改善、有効なシステムズエンジニアリング・ユースケースに沿った所見など、貴重な結果が得られた (CRYSTAL, 2016)。

CRYSTAL プロジェクトで前面に出された重要な考え方は、開発の専門分野、適用ドメインおよびツール・プロバイダー間で統合が重複する問題を克服することを目的として、ライフサイクルの相互運用性に対応すべく標準化された統合インターフェースに依拠することである。

そのような標準化された統合インターフェースでは、開発サイクルを通じて総体的に使用されるすべての成果物の共通の特徴として、簡易で汎用的な概念を定義する必要がある。ライフサイクルの相互運用性というコンテキストでは、開発の専門分野の境界を超えた成果物間のリンクおよび依存関係の意味に重点が置かれる。

新たに登場したオープン標準の OSLC (Open Services for Lifecycle Collaboration、ライフサイクルでの協業のためのオープン・サービス) は、相互運用性の問題に取り組む土台になるものとして認識されている (Open Services for Lifecycle Collaboration, 2016)。OSLC では、ライフサイクル活動の支援に重点を置いた一連の仕様が定義される。一方で、OSLC のオープン・イニシアチブは「緩く団結した」ウェブ・コミュニティからオープン標準化機関 OASIS の参加組織にまで成長した。多くの商用およびオープンソース製品でオープン標準が採用され、参加組織の数は増え続けている。

OSLC はすでに優れた相互運用性の土台となっているが、CRYSTAL プロジェクトではさらに他の相互運用性のニーズがいくつか明らかになり、次の拡張が提案されている。

- OSLC 構成および変更管理の仕様
- OSLC 追跡リソースセットの仕様。サーバーで完全なリソースセットを公開でき、セットに対する追加または削除を追跡したり、そのセットでリソースの変更を追跡したりできる。

これらの仕様により、OSLC はシステムズエンジニアリングにおける相互運用性の課題軽減を期待できるアプローチであり、ツール業界で積極的に採用されるようになってきている。

## 5.4.2 実施事例

航空宇宙および自動車ドメインにおける CRYSTAL プロジェクトの公開ユースケースでは、代表的な開発手法、関連する相互運用性の課題、考えられるツールの統合の概要がよく示されている。航空宇宙の事例では、地域間輸送用航空機の除氷装置の仕様、分析、設計およびシミュレーションに焦点を当てている。ここで取り上げられているプロセスの各段階を図 14 に示す。

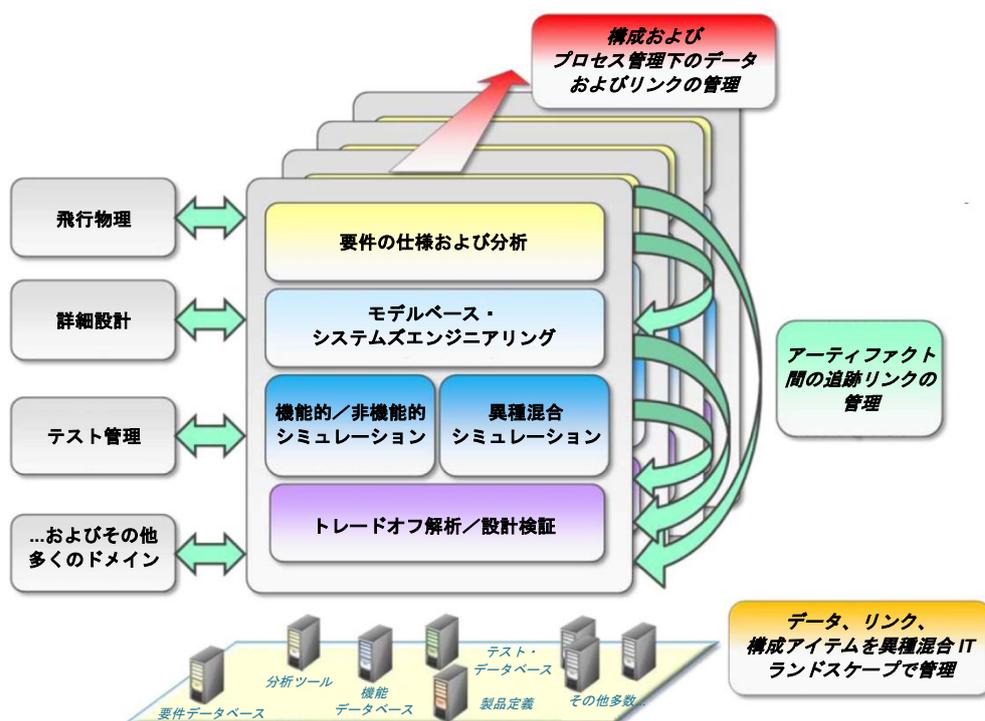


図 14 : CRYSTAL の航空宇宙業界の公開事例で取り上げられているプロセスの各段階(Source: Airbus Group)

自動車の事例では、車両レベルでのパワートレインの設計からマイクロプロセッサやソフトウェアの開発まで、車の開発における様々な段階が研究され、自動車メーカー各社の開発設定における知見が示されている。この事例では、システム分析、変動性または派生の管理、機能的安全性および追跡可能性など、Vモデル全体を通して発生する相互運用性の課題に焦点が当てられている。ここで取り上げられている各段階および分野を図 15 に示す。

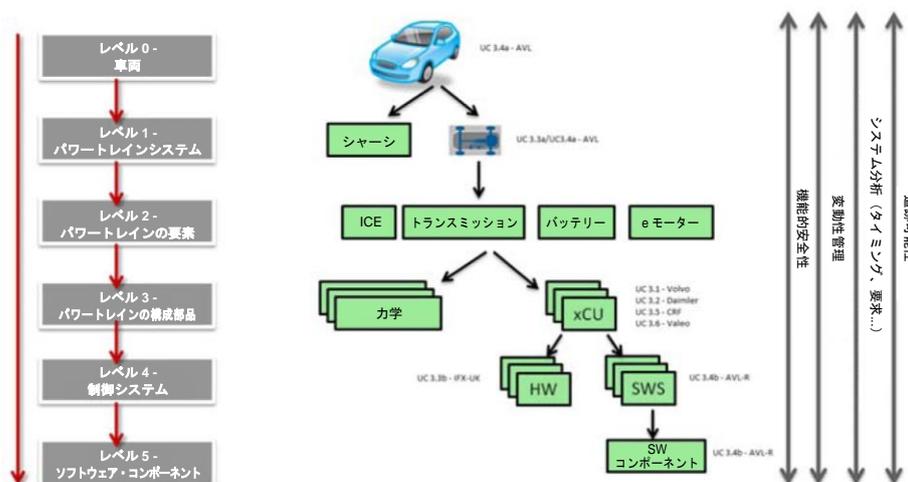


図 15 : CRYSTAL の自動車業界の公開事例で取り上げられている各段階および分野(Source: AVL)

各種のユースケースに含まれている実践領域およびツールは以下のもので構成され、通常、システムズエンジニアリング設定で使用されるツールの機能性の相当量を取り上げられている。

- 高度追跡可能性 : Reqtify、Rational SSE、RELM
- モデルベース・システムズエンジニアリング (MBSE) : RequisitePro、IBM DOORS NG、Rhapsody、Design Manager、PTC Integrity、Modeler
- 要求の品質分析 : Requirements Quality Suite
- テスト管理 : TVS assureSign
- 安全性分析 : Fault Tree+
- シミュレーション : Simulink、OpenModelica
- 変動性管理 : pure::variants
- プロセスの自動化 : Rational Method Composer、Rational Team Concert

ツール・チェーンの相互運用性に関しては、開発者は通常、以下のことを期待する。

- 均一な影響分析とカバレッジ分析、レポートおよび測定基準に対応するツール間でセマンティック・リンクが作成されていること。
- 要求が品質特性と照合され、改善の指針が示されること。
- 共通の成果物を使用して、システム設計と機能安全がシームレスに統合されていること。
- システムの各側面に重点を置いたシミュレーションモデルが、全体的なシステム・シミュレーションに組み込まれること。
- 変動モデルでは、様々な種類の成果物トの明示的な変化点が管理され、様々な製品の派生への可変性が解決されること。

- ワークフローの支援が提供され、プロセスへの準拠が保証されること。

### 5.4.3 教訓と推奨事項

このユースケースでの経験では、ツールの相互運用性は OSLC に基づいて大きく向上する可能性があり、最近実際に大きく進歩してきたことが示された。IBM、PTC、PureSystems、Siemens をはじめ多くのツール・プロバイダーが、他のツールと相互運用するための標準化されたインターフェースを提供するようになってきており、様々なツール・プロバイダーからのツールの利用による、注目すべきツール相互運用シナリオが示されている。

これらの利点のほか、ユースケースでは、ツールアダプタならびにデータの設定および保守に関する複雑さも明らかになった。その他の問題はリンク管理である。OSLC では今のところ、リンクをどこでどのように管理するかはほとんど指定しておらず、インターフェースの開発者に完全に依存している。OSLC ベースのインターフェースの実装に関しては、ソースコードの可用性が前提条件であるため、既存のツールに OSLC インターフェースを実装するのはかなり難しいことが明らかになった。

ツールの相互運用性について得られた主な教訓は、ツールの相互運用が依然として、具体的な設定でツールごとに詳細に調査すべき問題であるという点である。一般的な相互運用シナリオに関するメタ・モデルとツールのインターフェースの一般化は進行中であり、達成されるかどうか不明である。ツールの相互運用を改善するためにユースケース駆動のアプローチに従うのは、ツール・チェーンにおける欠点と、付加価値をもたらす改善点を、体系的かつ測定可能な方法で識別する上で効果的な実践である。また、これは、新しいツールの機能について開発者を教育するのにも役立つ。

ツールの相互運用以外では、過去のプロジェクトのデータにアクセスするために複雑なツールのインフラストラクチャを再インストールするという手間を経ずに重要なデータをアーカイブできるので、オープンデータ形式は役立つ。

## 5.5 システムの仮想開発

仮想開発では、実際の成果物がシミュレーションモデルで置き換えられる。成果物を置き換える例としては、機械部品を置き換える例 CAD モデル、命令セットシミュレータを実装して開発中のハードウェア・プラットフォームを置き換える仮想プラットフォーム、ソフトウェア実装を置き換える Simulink 動作モデルなどがある。

仮想開発の主な目的は、開発中の成果物の重要な特性と機能を保証することである。仮想プロトタイプなら、実際のプロトタイプの実装よりもずっと早い段階で入手できる。仮想プロトタイプを使用して機能を評価することで、現物による実装／試作した場合よりもはるかに早い段階で定量評価が可能になる。こ

れにより、欠陥や誤った仕様が早い時点で検出されるため、複雑なシステムを開発する際のリスクが減り、必要なやり直し作業も少なくなる。

複雑なシステムは様々な成果物で構成されるため、すべての側面が含まれるシミュレーションを開発者が作成することは難しい。しかしこれは、将来の設計における新たな影響要因を検出するために必要になる。新たな影響要因は、システムの様々な部品の相互作用で発生するが、その評価例の1つとして、車どうしの通信の性能評価がある。これは、使用される無線ネットワーク、運転手の行動、交通モデル、プロトコルの動作によって異なるからである。一般に、シミュレーションツールは特殊化されていて、適用範囲が狭い。これらは特定の効果のみのシミュレーションを行う。このため、新たな影響要因のシミュレーションは、開発中のシステムのすべての関連部品用のシミュレータを統合しない限り、簡単には評価できない。近年、このような状況を克服するために、シミュレータの結合が広く採用されるようになってきている。

### 5.5.1 アプローチ例

シミュレータ結合の開発は、シミュレーションモデルの演算および通信モデル（MOCC）が異なることが多いため困難である。これらは一貫して統合される必要がある（Kuhn, Forster, Braun, & Gotzhein, 2013）。

MOCC では、いつシミュレーションモデルが実行され、どのようにして通信するかが定義される。一般的な MOCC には 3 つあり、不連続時間モデル、不連続イベントモデル、連続時間モデルである。一般的な産業用途は、電気／電子（E/E）アーキテクチャの仮想評価である。E/E アーキテクチャはネットワークで接続されたハードウェア制御装置で構成される。様々なネットワークがゲートウェイを介して接続され、タスクが電子制御装置に配置される。実装される制御アルゴリズムによっては、タスクはスケジューリングおよび通信遅延に伴う影響を受けやすい、または通信およびスケジューリングの揺らぎによる影響を受けやすい可能性がある。このため、タスクを特定の制御装置の特定の処理装置に配置するかどうかの決定は、アルゴリズムの性能に大きく影響する可能性がある。さらに、ネットワーク通信および安全機構は、仮想 E/E アーキテクチャを使用して評価できる。

### 5.5.2 実施事例

次の適用例は、ある産業プロジェクト（匿名）から得られた結果である。このプロジェクトは、トラックに装備されるリモート制御のリフト用安全機構の開発に関するものであった。リフトはスマートフォンで制御しなければならないが、スマートフォンは安全でないデバイスとして見なされている。トラックのゲートウェイ・ハードウェアには安全ガード機構を実装できるが、スマートフォンはただのセンサーとなる恐れがあるからだ。実際、安全機構は、入力されたユーザーの意図が確信にもとづくことを保証するスマートフォンの UI 概念、安全通信層が実装されたスマートフォン側の通信プロトコル、ゲートウェイ

イ装置側の通信層、および受信される結果を評価する論理で構成される。この論理では、測定されるセンサー入力が、どのセンサーの欠陥も有効なユーザー入力を阻害せず、どの伝送エラーも発生せず、どんな矛盾する命令も受信されないという、ユーザーの意図を確実に表すことが保証されなければならない。

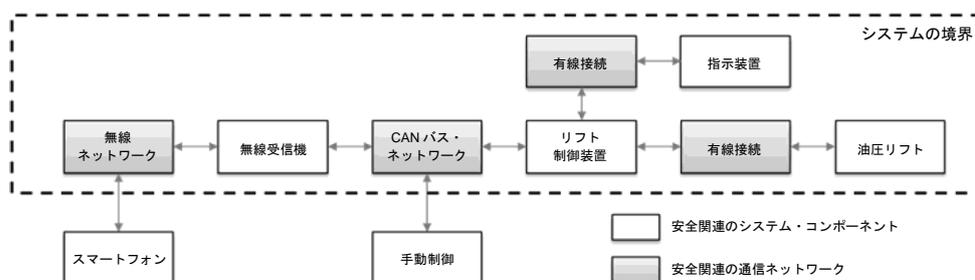


図 16：仮想評価の適用例

仮想評価では、すべての場合においてゲートウェイの論理が安全システムの動作を引き出すことが確認されなければならない。これは、正しい動作を引き出す多数のテストケース（シナリオ）を定義し、安全システムの動作が記述された規則を定義することによって実行される。シミュレーションは、例えば通信ネットワークのビット反転や、センサーの縮退故障などの故障モデルに応じて故障を発生させる。すべての場合において、システムはユーザーが意図した動作か、事前定義されたいずれかの安全動作のどれかを引き出す必要がある。油圧リフトの場合、これはすべての動きを停止することである。

### 5.5.3 教訓と推奨事項

仮想開発を使用することで、既存のアプローチの進化ではなく、新しいアイデアの実現である革新的な概念を創出することも可能になる。シミュレーションによって、開発者は経験を集め、様々なアプローチの性能を定量的に評価して比較できるようになる。シミュレーションにより、リスクなく、重要な面を早期に評価できる能力は、シミュレーションモデルの速度および精度の向上と併せ、仮想開発技法の重要性、ならびに適用可能性を高める。

近い将来、製品の複雑性は著しく増大し、システム開発はますます複数の専門分野にまたがって行われるようになるであろう。その結果、大規模なシステムの組み立てはより難しく、コスト負担が高くなる。

今日、実ハードウェアを用いたテストは産業界で一般的に実践されている。しかし、統合テストが実施されるのはプロジェクト段階のかなり後の段階である。そのため、欠陥の修正作業は、コストの増大、複雑化、および不必要なプロジェクトの遅延につながる。システムやアーキテクチャの複雑化が進むことを考慮し、開発プロセスのできるだけ早い時期に統合テストを開始するべきである。

効率的なアプローチとして、既存のシミュレータ結合によって作成される仮想ハードウェアを実装したテストベッドを検討することを奨励する。

## 参考文献

- acatech - National Academy of Science and Engineering. (2016, 9 23). Retrieved from <http://www.acatech.de>
- Achary, B., & Actis, M. (2013). Introducing the CTA concept. *Astroparticle Physics*, 43(3). Bruner, J. (2013). *Industrial Internet*. O'Reilly Media, Inc.
- Cavalcante, E., Pereira, J., Alves, M., Maia, P., Moura, R., Batista, T., . . . Pires, P. (2016). On the interplay of Internet of Things and Cloud Computing: A systematic mapping study. *Computer Communications*.
- Conforto, E., Rossi, M., Rebentisch, E., Oehmen, J., & Pacenza, M. (2013). *Survey Report: Improving Integration of Program Management and Systems Engineering*. Philadelphia: PMI and INCOSE.
- CRYSTAL. (2016). Retrieved from CRYSTAL - Critical System Engineering Acceleration: <http://www.crystal-artemis.eu>
- Ebert, C. (2014). *Requirements Engineering - Industry Practice*. Vector Consulting Services. Retrieved 09 08, 2016, from [http://vector.com/portal/medien/vector\\_consulting/publications/Ebert\\_RequirementsEngineering\\_Overview\\_EN.pdf](http://vector.com/portal/medien/vector_consulting/publications/Ebert_RequirementsEngineering_Overview_EN.pdf)
- Elberzhager, F., Rosbach, A., & Bauer, T. (2014). *An Integrated Analysis and Testing Methodology to Support Model-Based Quality Assurance*. *Software quality days (SWQD 2014)*. Vienna: Springer.
- Elm, J. P., & Goldenson, D. R. (2012). *The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Study*. Carnegie Mellon University, Software Engineering Institute, AESS, NDIA.
- Estefan, J. A. (2007). *Survey of Model-Based Systems Engineering (MBSE) Methodologies*. INCOSE MBSE Focus Group.
- Federal Ministry for Economic Affairs and Energy. (2016, 9 23). *Plattform Industrie 4.0*. Retrieved from <http://www.plattform-i40.de>
- Feth, P., Bauer, T., & Kuhn, T. (2015). *Virtual Validation of Cyber Physical Systems*. *Software Engineering and Management 2015 (SE 2015)* (pp. 201-206). Dresden: Springer.
- Fricke, S., Grau, R., & Zwingli, A. (2014). *Requirements Engineering: Best Practice*. In S. A. Fricker, C. Thümmel, & A. Gavras, *Requirements Engineering for Digital Health* (pp. 25-38). Heidelberg, New York, Dordrecht, London: Springer.
- Gausemeier, P.-I., Dumitrescu, R., Steffen, D., Czaja, A., Wiederkehr, O., & Tschirner, C. (2015). *Systems Engineering in industrial Practice*. Heinz Nixdorf Institute, Fraunhofer Institute for Production Technology, Unity AG.
- Germany Federal Ministry of Defense. (2012). *Customer Product Management (amended)*. German Federal Ministry of Defense.
- Hankel, M.; Bosch Rexroth. (2015). *Industrie 4.0: The Reference Architectural Model Industrie 4.0 (RAMI 4.0)*. Frankfurt am Main, Germany: ZVEI - German Electrical and Electronic Manufacturers' Association.
- Heidrich, J., Trendowicz, A., & Ebert, C. (2016). *Exploiting Big Data's Benefits*. *IEEE Software*, 33(4), 111-116.
- IBM. (n.d.). *IBM Rational Harmony for Systems Engineering: The Harmony Process*. IBM. Retrieved 09 22, 2016, from [http://www.ibm.com/support/knowledge-center/SSB2MU\\_8.1.5/com.btc.tcatg.user.doc/topics/atgreqcov\\_SecSysControllerHarmony.html](http://www.ibm.com/support/knowledge-center/SSB2MU_8.1.5/com.btc.tcatg.user.doc/topics/atgreqcov_SecSysControllerHarmony.html)

- ISO. (2011). ISO 26262: Road vehicles – Functional safety. International Organization for Standardization.
- Jazdi, N. (2014). Cyber physical systems in the context of Industry 4.0. IEEE International Conference on Automation, Quality and Testing, Robotics (pp. 1-4). IEEE.
- Kagermann, H., Helbig, J., Hellinger, A., & Wahlster, W. (2013). Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry. Final report of the Industrie 4.0 Working Group, Forschungsunion. Retrieved from [http://www.acatech.de/fileadmin/user\\_upload/Baumstruktur\\_nach\\_Website/Acatech/root/de/Material\\_fuer\\_Sonderseiten/Industrie\\_4.0/Final\\_report\\_Industrie\\_4.0\\_accessible.pdf](http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Material_fuer_Sonderseiten/Industrie_4.0/Final_report_Industrie_4.0_accessible.pdf)
- Kläs, M., Bauer, T., Dereani, A., Söderqvist, T., & Helle, P. (2015). A Large-Scale Technology Evaluation Study: Effects of Model-Based Analysis and Testing. 37th International Conference on Software Engineering (ICSE 2015) (pp. 119-128). Stockholm: IEEE Computer Society.
- Kuhn, T., & Antonino, P. (2014). Model Driven Development of Embedded Systems. Embedded Software Engineering Kongress (ESE).
- Kuhn, T., Forster, T., Braun, T., & Gotzheim, R. (2013). FERAL – Framework for Simulator Coupling on Requirements and Architecture Level. ACM-IEEE International Conference on Formal Methods and Models for System Design. Portland, USA.
- Lee, J., Lapira, E., Bagheri, B., & Kao, H. (2013). Recent advances and trends in predictive manufacturing systems in big data environment. Manufacturing Letters, 1(1), 38-41.
- Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing. Gaithersburg, Maryland, USA: National Institute of Standards and Technology.
- Minerva, R., Biru, A., & Rotondi, D. (2015). Towards a definition of the Internet of Things (IoT). IEEE Internet Initiative.
- Naab, M., Knodel, J., Kuhn, T., & Rost, D. (2016). Smart Ecosystems Reference Model. Fraunhofer IESE.
- NATO. (n.d.). NATO Architecture Framework RFCP Regarding NAF V3.1 Chapter 5: NATO Architecture Framework Metamodel (NMM) and Architecture Data Exchange Specification (ADES). Retrieved from <http://www.nhqc3s.nato.int/ARCHITECTURE>
- Open Services for Lifecycle Collaboration. (2016). Retrieved from <http://open-services.net>
- Oya, I., Fülling, M., Oliveira Antonino, P., Conforti, V., Hagge, L., Melkumyan, D., . . . the CTA consortium. (2016). The Software Architecture for the Cherenkov Telescope Array. SPIE - International Society for Optics and Photonics Conference and Exhibition 2016. 9913. Edinburgh, Scotland, UK: Software and Cyberinfrastructure for Astronomy III.
- Pisching, M., Junqueira, F., dos Santos Filho, D., & Miyagi, P. (2015). AN ARCHITECTURE FOR ORGANIZING AND LOCATING SERVICES TO THE INDUSTRY 4.0. ABCM International Congress of Mechanical Engineering. Rio de Janeiro, Brazil.
- Pohl, K., Achatz, R., & Broy, M. (2012). Model-Based Engineering of Embedded Systems- The SPES 2020 Methodology. Springer.
- Software Engineering Institute, Carnegie Mellon. (n.d.). A Framework for Software Product Line Practice - Requirements Engineering. Retrieved 09 08, 2016, from [http://www.sei.cmu.edu/productlines/frame\\_report/req\\_eng.htm](http://www.sei.cmu.edu/productlines/frame_report/req_eng.htm)
- SOPHIST. (2016). RE Primer. Retrieved 09 21, 2016, from <https://www.sophist.de/publikationen/wissen-for-free/>
- Sophist. (n.d.). FAQ Requirements Engineering. Retrieved 09 08, 2016, from <https://www.sophist.de/en/requirements/requirements-engineering/faq-requirements-engineering/>
- SwissQ, & University St. Gallen. (2014). Trends & Benchmarks Report in Software Development. Zürich: SwissQ Consulting AG. Retrieved 09 08, 2016, from [http://swissq.it/wp-content/uploads/2016/02/Agile\\_RE\\_Testing-Trends\\_und\\_Benchmarks2014.pdf](http://swissq.it/wp-content/uploads/2016/02/Agile_RE_Testing-Trends_und_Benchmarks2014.pdf)

- Teufl, S., Khalil, M., & Mou, D. (2013). Requirements for a Model-based Requirements: Systematic Literature Review and Survey. Munich: fortiss GmbH. Retrieved 09 08, 2016, from [http://download.fortiss.org/public/projects/af3/research/2013/MbRE\\_tool\\_requirements\\_for\\_embedded\\_systems.pdf](http://download.fortiss.org/public/projects/af3/research/2013/MbRE_tool_requirements_for_embedded_systems.pdf)
- The Standish Group. (2014). CHAOS Report. The Standish Group.
- Wang, H., Osen, O., Li, G., Li, W., Dai, H., & Zeng, W. (2015). Big data and industrial internet of things for the maritime industry in northwestern norway. IEEE Region 10 Conference TENCON (pp. 1-5). IEEE.
- Webel, C., Darting, S., Schmitt, M., Kleinberger, T., Braun, R., & Weber, J. (2015). Pragmatisches Systems Engineering in einem Großprojekt mit Einschränkungen. Tag des Systems Engineering 2015 (pp. 323-332). Munich: Carl Hanser Verlag GmbH & Co. KG.

# 文書情報

タイトル： WP1/D1.5：システム  
エンジニアリングの研究：  
課題とベストプラクティス

日付： 2016年9月23日  
ステータス： 最終版

Copyright 2016 Fraunhofer IESE. / IPA/SEC2016  
All rights reserved. 本出版物のいかなる部分も、発行者の書面による事前の許可なく、いかなる形態または手段（コピー、記録、またはそれ以外を含むがこれらに限定されない）によっても、複製、情報検索システムに格納、または伝送することを禁ずる。この出版物を非商用目的で配布する場合、書面による許可は不要。