



# **Tech-Clarity Insight :** **システム エンジニアリング** **バイヤーズ ガイド**

**システム エンジニアリング**  
**ソリューションに関するエクス**  
**パートによるガイド**

## 目次

概要 .....	3
システム エンジニアリングのビジネス価値 .....	4
まずはプロセス定義から .....	6
要件の管理.....	9
システムの設計 .....	11
モジュール式システムの設計.....	12
製品ライン バリエーションのサポート .....	14
詳細設計への対応.....	15
システムの検証と確認.....	16
サービス要件の評価 .....	18
ベンダー属性の考慮 .....	19
企業固有のニーズの特定 .....	20
まとめ.....	21
推奨事項 .....	22
執筆者について .....	23



## 概要

今、世界では熾烈な競争が繰り広げられており、企業は競争力の面で自社の製品を差別化するための新たな手法を求めようになってきています。多くの企業が、製品にソフトウェアを組み込むことで、競争において自社の製品を一歩リードさせようとしています。組み込みソフトウェアにより、製品にインテリジェンスを取り入れ、よりパーソナライズされたエクスペリエンスを顧客に提供するチャンスがもたらされます。モノのインターネット (IoT) は、スマート コネクティッド デバイスにより製品の新しいサービスと差別化されたサービスを実現する可能性をもたらすことで、この傾向をさらに促進します。ただし、これを活用するには、高度なシステムエンジニアリング手法が必要です。

メカニカル コンポーネント、電子機器、ソフトウェアを組み合わせることでイノベーションのチャンスがもたらされますが、その一方で固有の課題も生まれ、現在の製品と製品開発プロセスがさらに複雑になります。つまり、今日の製品で成功を収めるには、高度なシステム エンジニアリング手法が不可欠です。高度なシステム エンジニアリング手法を導入するうえで鍵となるのが、それを支えるテクノロジーを持つことです。このバイヤーズ ガイドは、システム エンジニアリングを支える適切なソフトウェアを選択しようとするメーカーをお手伝いします。

---

**メカニカル コンポーネント、電子機器、ソフトウェアを組み合わせることでイノベーションのチャンスがもたらされますが、固有の課題も生まれます。**

---

このガイドは、システム エンジニアリング ソフトウェア ツールの機能、サービス要件、ベンダーの特性、企業固有の考慮事項という各トピックを扱った 4 つのセクションから構成されます (図 1)。各セクションには、システム エンジニアリングを支えるソフトウェア ツールを選択する際に検討が必要な主要な要件のチェックリストが設けられています。

高度なシステム エンジニアリング手法の基盤として、企業はシステム エンジニアリング プロセス全体に注目し、システム エンジニアリングのあらゆる側面に対応したソリューションを用意する必要があります。これによって企業は競争力を高めることができ、これが大きな成長と収益性の向上につながります。

---

**高度なシステム エンジニアリング手法の基盤として、企業はシステム エンジニアリング プロセス全体に注目し、システム エンジニアリングのあらゆる側面に対応したソリューションを用意する必要があります。**

---

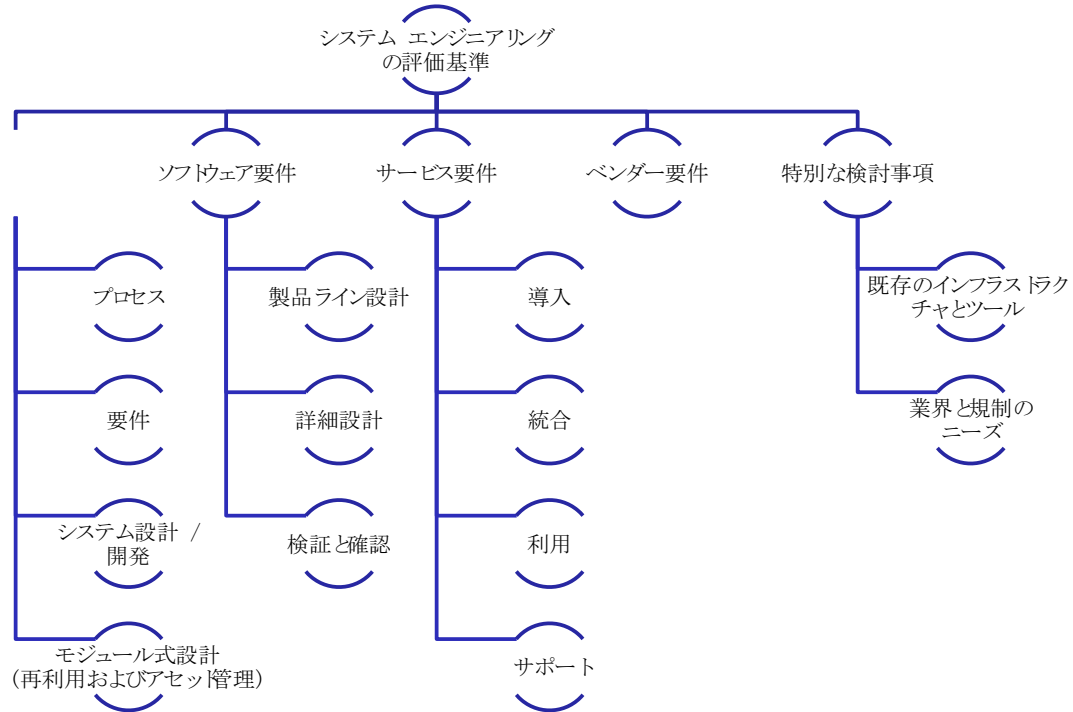


図 1: システム エンジニアリングの評価フレームワーク

このガイドは、すべてを網羅した要件リストを示すものではありません。システム エンジニアリングのための概要を示しています。企業は、このバイヤーズ ガイドで取り上げているシステム エンジニアリング機能に加えて、各エンジニアリング部門の個々のエンジニア向けツールの必要性も考慮する必要があります。

## システム エンジニアリングのビジネス価値

Tech-Clarity の過去の調査から、高度なシステム エンジニアリング プロセスが現代の製品を成功させるうえでいかに重要かがわかります。高度なシステム エンジニアリング手法が必要とされる理由の 1 つとして、製品に採用されている組み込みソフトウェアの増加が挙げられます。Tech-Clarity の『[Developing Software-Intensive Products: Addressing the Innovation Complexity Conundrum](#)』に、次のようなデータがあります。

- 調査対象ディスクリート メーカーの 75 % が、今後 5 年間で製品に組み込むソフトウェアを増加させる

この増加の理由は、成長と収益性の目標を達成するうえで、組み込みソフトウェアが戦略的に役立つとメーカーが考えていることです。調査回答企業によれば、ソフトウェアの使用を促進している製品戦略の上位は次のとおりです。

- 製品機能の改良 (74 %)
- よりスマートな製品の開発 (68 %)
- 製品イノベーションの強化 (62 %)
- 製品カスタマイズ機能の向上 (49 %)
- プラットフォーム設計の実現 (43 %)

これらはすべて、製品の差別化を図り、企業が競争力を高めるうえで重要な手法です。しかし、高度なシステム エンジニアリング手法がない場合、メーカーは組み込みソフトウェアの必要性につながるビジネス目標を下方修正することになっているのが現状です。

---

**高度なシステム エンジニアリング手法がない場合、メーカーは組み込みソフトウェアの必要性につながるビジネス目標を下方修正することになっているのが現状です。**

---

図 2 に、組み込みソフトウェアを含む製品を開発するときに発生し得るビジネスへのマイナスの影響を示します。

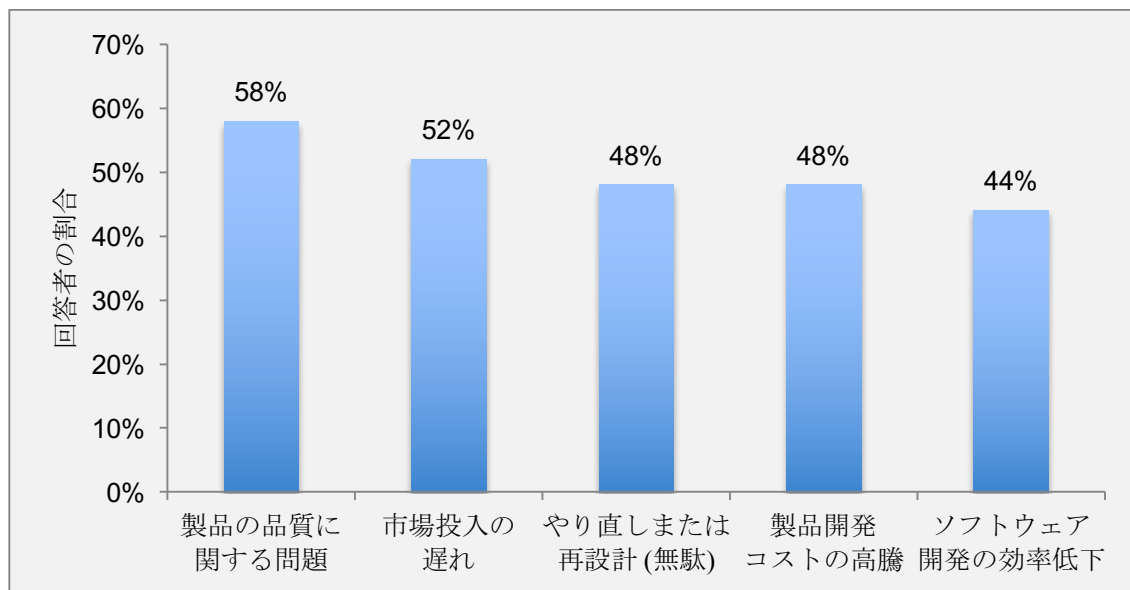


図 2: 組み込みソフトウェアを使用する製品の開発がビジネスに与えるマイナスの影響



高度なシステム エンジニアリング手法に従うことで、企業はシステム開発を管理するためのフレームワークを手にするようになります。システム開発は非常に複雑であるため、これらの手法がなければ状況が悪化しがちな領域は多数あります。これらの問題に対処しなければ、製品の品質が犠牲になります。品質の低下によってブランドの評判が傷つき、特に乗り換え可能な競合企業が存在する場合は、顧客を失いかねません。また、特に開発サイクルの後期で問題が見つかった場合、その修正によって遅延が生じます。遅延は市場投入期間の長期化を意味し、収益を得て市場シェアを獲得するために残された時間が少なくなります。遅延が発生した場合だけでなく、問題を修正するためのやり直しや再設計が必要になった場合も、コストがかさみます。問題の根本原因を特定し修正する中で、新しい設計および開発作業に費やすことができる時間が減り、コストの高騰と効率の低下を招きます。これらすべての問題によって収入が減り、コストが増加し、結果的に収益性が打撃を受けます。

しかし、幸いなことに、高度なシステム エンジニアリング手法の導入に集中的に取り組めば、メーカーはこうしたビジネスへのマイナスの影響を回避することができます。高度なシステム エンジニアリング手法の実現においては、テクノロジーが重要な役割を果たします。メーカーはテクノロジーを活用して、要件の適切な決定から製品ラインの管理まで、現代の製品の複雑性に対処できます。これは、多様な顧客セグメントにアピールするために必要です。

---

**しかし、幸いなことに、高度なシステム エンジニアリング手法の導入に  
集中的に  
取り組めば、メーカーはこうしたビジネスへのマイナスの影響を回避する  
ことが  
できます。**

---

## まずはプロセス定義から

高度なシステム エンジニアリング手法を導入するプロセスのスタート地点として最適なのは、プロセス定義です。Tech-Clarity の『[Developing Software-Intensive Products: Addressing the Innovation Complexity Conundrum](#)』によれば、システム エンジニアリングの最大の課題は、複雑なグローバル開発ネットワークの管理です (49 % のメーカーが回答)。プロセス定義は、こうした複雑性に対処するための 1 つの方法となります。

---

## プロセスの定義は、最も効果的な手法を特定し、それらを一貫して実践するための助けとなります。

---

プロセスを明確に定義することで、コミュニケーションと生産性が向上します。プロセスの定義は、最も効果的な手法を特定し、それらを一貫して実践するための助けとなります。一貫したベスト プラクティスの導入により、開発チームの効率性が向上し、市場投入までの期間が短縮されます。その結果、競争優位性が得られるとともに、開発投資を回収できるチャンスが広がり、収益性が向上します。

プロセスをサポートするうえで、テクノロジーは以下の役割を果たします。

- プロセスのオーサリングを合理化し簡略化する
- プロセスの採用を促進する
- プロセスの確実な実践を統括する
- プロセスのパフォーマンスの評価指標を取り込む
- 継続的な改善を可能にする
- プロセスとタスクのオーナーシップを管理する
- システム エンジニアリングの成果物と統合する

次に示すのは、これをサポートし、システム エンジニアリング プロセスをサポートすると考えられるソフトウェア要件のリストです。



要件	考慮事項
ベスト プラクティスを取り込むためのオーサリング ツール	ベスト プラクティスを文書化することで、効率性の向上につながる一貫性が得られます。オーサリング ツールにより、文書化プロセスを簡略化し、可能な限り自動化する必要があります。
業務プロセスとベスト プラクティスのグラフィカル モデリング	グラフィカル モデルにより、複雑なプロセスをより簡単に伝達できるビジュアル フレームワークを得られます。これによって理解が深まり、プロセスに従いやすくなります。OMG BPMN (Object Management Group Business Process Modeling Notation) などの、著作権使用料無料で、業界のソート リーダーシップに基づく実証済みの標準を提供する業界標準に準拠したツールを検討してください。
次の手順に関する通知と指示を自動的に行うことによって自動化されたベスト プラクティス	プロセスの自動化によりプロセスを導入して実施しやすくなり、その採用率が高まります。自動 To-Do リストや自動電子メール通知などの機能は、引き継ぎの合理化により、プロセスのボトルネックの克服と効率性の向上を支援します。
プロセスの修正	変更を容易に加えることができるため、教訓を応用できるようになり、継続的な改善につながります。これにより効率が上がり、プロセスを取り込むこともできるため、品質の向上とコストの削減が実現します。プロセスは、プロセスの作成者または実施担当者、あるいは UML/SysML モデラーなどのシステム エンジニアリング プロセスに最も近い担当者が修正できなければなりません。SysML は、システム エンジニアリングのためのモデリング言語です。そのベースとなっているのは、よりソフトウェア主体のモデリング言語である UML です。



要件 (続き)	考慮事項
サブプロセスの変更がそのサブプロセスを使用するほかのすべてのプロセスに自動的に反映される	これによってプロセスの一貫性が向上します。また、プロセス作成者にとっては、手作業による更新が減り、時間の節約になります。さらに、継続的な改善が後押しされ、教訓が一貫して生かされるようになります。
新しいプロセスやプロジェクト固有のプロセスの定義に利用可能な既存のプロセスのライブラリ	ライブラリを使用すると、プロセスを一から作成する代わりに、既存の実証済みプロセスを再利用できるため、時間の節約になります。また、以前のプロセスで学んだベスト プラクティスを新しいプロセスに適用することができます。
既存のベスト プラクティスに基づくプロジェクト固有のプロセス	時間を節約するには、既存のベスト プラクティスをコピーしてプロジェクト固有のプロセスを作成する必要があります。また、個々のプロジェクトのニーズに応じてプロセスが編集可能であることも必要です。これにより、最新のベスト プラクティスに基づいて新しいプロジェクト プランを策定できるだけでなく、プロジェクト固有のニーズに対応するために必要な柔軟性が実現します。
一般的なフォーマットにプロセス コンテンツをインポートおよびエクスポート	Microsoft Word、PDF、XML、Microsoft Project などの一般的なフォーマットを検討する必要があります。プロセスを定義している既存のコンテンツを利用することで、時間の節約になります。プロジェクト計画を任意のツールで使用できるため、採用率が高まります。
監査の自動化	これにより、プロセスが定義されたとおりに実践されます。

図 3: プロセス定義の機能的な要件

## 要件の管理

要件によって製品の土台が確立されます。開発作業のためのガイダンスとなります。理想的な要件管理ソリューションとは、高度なシステムビューから個々のコンポーネントまで、あらゆる要件を管理できるものです。システムエンジニアリングの課題の上位 2 つは、変更を管理すること (Tech-Clarity の調査に回答した企業の 56 %) と、要件から検証までのループを閉じること (同 48 %) です。要件管理ソリューションは要件のトレーサビリティによってこれに対処できるため、変更があった時点で要件間の依存関係を把握できます。トレーサビリティにより、定義から成果物、全エンジニアリング部門に及ぶテストまで、要件を追跡することが可能になります。

---

### 理想的な要件管理ソリューションとは、高度なシステムビューから個々のコンポーネントまで、あらゆる要件を管理できるものです。

---

次のリストには、要件管理ソリューションの主な基準の詳細を示しています。

要件	考慮事項
トップレベルとサブシステム両方の要件の定義	要件階層を管理することがシステムの複雑性の管理の手助けとなります。これにより、製品の品質をより簡単に維持できるようになると同時に、コストを低く抑え、コンプライアンスを効率化することができます。ほかの場所で作成された要件をインポートできる必要があります。
機械、電気、ソフトウェアなどを含むすべての分野にわたる要件のサポート	全部門の要件を一元化することで、エラーのリスクが減り、一貫した要件定義によりコミュニケーションが改善されます。
要件、設計、実装、テスト全体にわたるトレーサビリティと依存関係	変更の影響を適切に理解して、影響を受けるすべての構成部品を確実に更新するには、要件、設計、実装、テストにまたがるトレーサビリティが必要です。トレーサビリティは、規制コンプライアンスのためにも必要になります。
要件(続き)	考慮事項
要件の再利用	要件を再利用することで時間を節約できます。再利用する要件間の関係がどの程度精巧であるべきか検討してください。たとえば、要件を再利用する際、元の要件へのリンクが存在すべきでしょうか。要件を変更する際、その要件が再利用されているすべての場所で要件にフラグを付けるべきでしょうか。

要件ダイアグラム	SysML などのビジュアル モデルが要件のビジュアル フレームワークとなり、要件間の関係を理解しやすくなります。
要件の承認	FDA 21CFR などの規制への準拠が必要な企業は、必要な監査証跡、規格に適合した電子署名など、コンプライアンス要件を満たすための機能を備えたソリューションの導入を検討すべきです。
システム モデルのすべての要素に要件を割り当て	システム モデルに要件をマッピングすることで、要件、サブシステム、コンポーネント間の依存関係を視覚的に管理して、複雑性を管理できます。
システム モデルのフィルタ	1 つのモデル要素を選択すると、すべての関連要素がフィルタまたは特定されるようにすることで、トレーサビリティをサポートします。トレーサビリティにより変更を適用する際のエラーの可能性が減り、その結果、コスト超過の削減と品質の向上が可能になります。
変更による影響の分析	モデル要素または要件が変更されたときに、影響を受けるほかのモデル要素をツールで特定できなければなりません。これにより、変更オーダーの実行にかかる時間を短縮できるとともに、変更のあらゆる影響をより容易に理解できるようになります。
ほかの要件管理ツールとの統合	これにより、組織でうまく機能している既存のツールへの投資を利用するための柔軟性が得られます。

図 4: 要件を管理するための機能的な要件

## システムの設計

システム エンジニアリングが抱える多くの課題の根本原因は複雑性です。この複雑性を緩和する方法の 1 つに、システム モデルの使用があります。システム モデルとは、システムを視覚的に表現したものです。ビジュアル リファレンスは、システムの設計、解析、検証、確認に対応しています。わかりやすいシステムの共通のリファレンスを提供するため、製品開発チーム全体のコミュニケーションとコラボレーションが改善されます。これによって、より効率的で効果的なチームとなることができます。

**システム モデルはわかりやすいシステムの共通リファレンスを提供するため、**

## 製品開発チーム全体のコミュニケーションとコラボレーションが改善されます。

次に示すのは、システムモデリングをサポートするためのソフトウェア要件のリストです。

要件	考慮事項
業界標準の遵守	システムモデリング向けに検討すべき手法は SysML です。SysML は、システムエンジニアリングのためのモデリング言語です。
SysML 実行可能挙動モデル	SysML 実行可能挙動モデルは、早期検証のためにシステム挙動をシミュレートできます。こうして早期段階で挙動を把握することで、システム設計が顧客のニーズを満たすかどうかを簡単に判断できます。
パラメトリックダイアグラム	パラメトリックダイアグラムは、システムプロパティの拘束を示します。品質の向上につながる、パフォーマンスおよび定量解析に使用されます。
パラメトリックダイアグラムのシミュレーション / 評価ツールへの統合	パラメトリックダイアグラムを、Matlab や Modelica などのツールと統合することで、ツールどうしが連携し、早期段階での検証、最適化、トレードオフ解析を実行できます。これによって、より短い期間でより質の高い製品を生み出すことができます。
要件管理ツールとの統合	システムモデルに要件を統合することで、要件、サブシステム、コンポーネント間の依存関係を管理する視覚的な手法により、複雑性を管理できます。また、要件全体でシステムモデルからのトレーサビリティを維持することもできます。

図 5: システム設計の機能的な要件

## モジュール式システムの設計

Tech-Clarity の調査からわかるように、メーカーの 42 % が、設計の再利用がシステムエンジニアリングが抱える主な課題の 1 つであると報告しています。再利用により、設計、開発、テストの時間を短縮することで、市場投入までの期間を短縮することができます。さらに、複雑なシステムの場合、実証済み、検証済みのコンポーネントおよびサブシステムを再利用することで、エラーを削減することができます。困難な点は、システムが非常に複雑なため、再利用可能なコンポーネントを特定するのが難しいということです。ここで役立つのがモジュール式设计戦略です。

**複雑なシステムの場合、実証済み、検証済みのコンポーネントおよびサブシステムを再利用することで、エラーを削減することができます。**

モジュール式システムでは、理解しやすい小さなセクションにシステムを分割します。設計が進行する中で、これらの小さなセクションは検証が容易で、問題をより早い段階で特定できます。検証後、コンポーネントとサブシステムを再利用可能なアセットのライブラリにパブリッシュし、将来の類似したシステムで使用できます。

次に示すのは、モジュール式システム開発をサポートする機能のリストです。

要件	考慮事項
再利用可能なサブシステムとコンポーネントのライブラリ	既存のコンポーネントのリポジトリを使用すると、再利用が容易になり、時間を節約できます。既存の検証済みコンポーネントを使用すると、エラーのリスクが減り、結果的に高品質と低コストを実現できます。再利用のための業界のベスト プラクティスに関するガイドラインと推奨事項を示す業界仕様、OMG RAS に準拠したソリューションを検討してください。
ライブラリ コンポーネントがインタフェース、バリエーション、要件を定義	これによって、将来のシステムでコンポーネントを適切に再利用するのが容易になります。
再利用可能なコンポーネントまたはサブシステム（アセット）を特定するための検索ツール	インデックシング、分類、タグ付けなどの機能を使用すると、再利用可能なコンポーネントがよりすばやく簡単に見つかるため、設計時間とテスト時間の両方を短縮できます。
要件 (続き)	考慮事項
ライブラリ パブリッシング ツール	サブシステム、そのインタフェース、コンポーネント、モデルのユース ケースと要件、およびインタフェース ファイルを公開することで、再利用を促進できます。トップダウン モデリングまたはボトムアップ システム マイニング アプローチにも対応しています。
再利用可能なアセットのバージョン管理および履歴	バージョン履歴により、コンポーネントの新しいバージョンが使用可能になったときに、その利用者に通知できます。システム モデルでも、使用されているコンポーネントのバージョンを管理できます。

再利用を評価するためのダッシュボード	何がうまくいき、何がうまくいかなかったかを突き止める際にメトリックスが役立ちます。コンポーネントとサブシステムがどの程度の頻度で再利用されているかを追跡することで、再利用戦略の有効性を評価し、ライブラリを継続的に改善することができます。
--------------------	------------------------------------------------------------------------------------------------------------------------

図 6: モジュール式設計の機能的な要件

## 製品ラインバリエーションのサポート

システムの観点から見れば、製品ラインは類似した製品のグループです。製品ラインはさまざまなバリエーションから構成されており、それぞれに共通の基本機能と変更可能なオプションがあります。その目的は、再利用可能なコンポーネントを持つ共通のプラットフォームを定めることです。それらのコンポーネントを組み合わせ、固有の製品モデルを作り出すことができます。企業が製品に追加するソフトウェアを増やしている主な理由の 1 つがプラットフォーム設計の実現ですが、49 % が製品プラットフォーム バリエーションの設計が最大の課題であると回答しています。製品を顧客に固有のニーズに合わせてカスタマイズする機能により、製品がより望ましいもの、より競争力の高いものになります。ソフトウェアは、さまざまな製品機能を提供するためのコスト効率の高い手法となります。企業が、好みも規制も異なる可能性のある新しいグローバル市場に製品を展開しようとする中、この手法はさらに威力を発揮しています。ただし、1 つの製品に複数のコンフィギュレーションがある場合、単一システムの複雑性は飛躍的に増大します。

### 製品を顧客に固有のニーズに合わせてカスタマイズする機能により、製品がより

**望ましいもの、より競争力の高いものになります。**

次のソフトウェア要件のリストに従うことで、この複雑性への対処、コンフィギュレーション間での変更の管理、製品ラインの競争力の強化を容易に実現することができます。

要件	考慮事項
「ISO 26550:2013 Software and Systems Engineering - Reference Model for Product Line Management and Engineering」のサポート	この ISO 標準をサポートすることで、企業は業界のベスト プラクティスを利用することができます。

システム モデルに埋め込まれたコンポーネントおよびサブシステムの依存関係と拘束	これによって、選択されたオプションに基づいてさまざまな製品バリエーションの作成が自動化されるため、時間を節約し、エラーのリスクを減らすことができます。
製品モデルの自動生成	選択されたフィーチャーに基づくモデルの自動生成機能は、さまざまなオプションとバリエーションを比較するトレード研究解析をサポートするうえで役立ちます。
システム モデルを使用してさまざまな製品バリエーションを作成	さまざまな製品バリエーションを作成するための視覚的なガイダンスとしてシステム モデルを使用すると、複雑性がより管理しやすくなります。また、システム モデルを使用することで、変更が複数の製品バリエーションに及ぼす影響が視覚的に把握しやすくなります。
可変性モデリング ダイアグラム	このダイアグラムにより、さまざまなバリエーションをモデル化できます。可変性モデリング ダイアグラムの要素を SysML ダイアグラムで使用する機能により、時間を節約し、一貫性を確保することができます。
ライブラリの再利用のためにバリエーション オプションおよびバリエーションが公開可能	この機能は、追加の製品バリエーションの作成を助け、再利用を促進します。

図 7: 製品ライン バリエーションの機能的な要件

## 詳細設計への対応

46 % が、部門間のコミュニケーションがシステム エンジニアリングの主な課題であると回答しています。ソフトウェアは、チームがより容易にコラボレーションし連携できるようにするための複数の機能を提供します。複数の部門からなるチームがより効率的に連携するうえで役立つ機能を、次の表にリストしています。

要件	考慮事項
平行 / 同時モデリングおよび開発	単一のデータベースで同時に複数の開発者とエンジニアが作業することで、効率を高めることができます。データのロック機能により、変更の上書きを防ぐことができます。



コラボレーション ツールが用意されています	エンジニアリング部門ごとに知識がサイロ化されていると、コラボレーションが難しくなります。より優れたコラボレーションを促進するテクノロジーを利用することで、チーム内の連携が容易になります。具体的には、たとえば、自動化されたワークフローや、要件、テスト プラン、設計の詳細、アセット、変更の一元的な管理などがあります。
インクリメンタルなベースラインの管理	複数のチームが同時に作業を進める場合は、すべてのチームが常に最新の情報を使用することが重要ですが、ベースラインはそのための手助けとなります。
モデル比較	設計が進行し変更が加えられていく中で、グラフィックとデータの両面でモデルを比較すると、変更内容が明確になります。
プロジェクト管理ツール	プロジェクト管理ツールは、タスク、責任、アクセス権を割り当てるうえで役立ちます。複雑なプロジェクトに取り組んでいる場合、誰が何の担当者かをチームで把握する必要があります。サード パーティが関与する場合を含め、誰が何を変更したかを管理するために、アクセス制御が特に重要になります。
<b>要件 (続き)</b>	<b>考慮事項</b>
トレーサビリティ	高レベルから詳細設計までシステムを追跡できれば、変更の影響を理解したうえで変更を反映することがより容易になります。検証中に見つかった問題の根本原因の究明も容易になります。コンフィギュレーションの管理と開発も、エンドツーエンドのトレーサビリティが確立されていれば、より簡単です。

図 8: 詳細設計の機能的な要件

## システムの検証と確認

検証では、適切な製品が製造されることを確認します。確認では、製品が適切に製造されたことを確認します。複雑なシステムでは、最終的な製品が当初意図したとおりに動作することを確認するのは容易ではありません。実際、48 % が、要件と検証の間のループを閉じることを主要な課題の 1 つに挙げています。



できるだけ早い段階で問題を発見することが、製品が適切なものであることを確認するうえで重要となります。検証のプロセスは、開発サイクル全体を通じて継続的に実施する必要があります。これを行うことで、最終段階で問題が見つかるリスクを低減できます。問題の発見が遅れると、問題の修正にかかるコストが増大し、製品発売日に製品を発売できなくなるリスクが生じます。さらに、定期的にテストを実施することで、製品の品質が向上し、市場で成功を収める確率が高まります。

次に示すのは、早い段階で定期的に検証を行うための機能的な要件のリストです。これらの機能により、企業は、適切な製品を製造していることを、早い段階から開発ライフサイクル全体を通じて継続的に検証することができるようになります。

要件	考慮事項
テストのプランニング、定義、作成と要件の定義の統合	要件定義は、テスト プランとリンクされていると、よりテストしやすくなります。各要件とテストと関連付けることで、テスト漏れがなくなり、要件が満たされていることを検証がより容易になります。その結果、最終的な製品が顧客および市場の要件を満たす確率が高くなります。
要件 (続き)	考慮事項
テスト ケースと要件の間のトレーサビリティ	各要件をテストにリンクすることで、テストが関連付けられていない要件にフラグを付けることが可能になり、テスト漏れがなくなります。さらに、テストが失敗した場合、それが要件やシステムに与える影響を理解するのがより容易になります。
開発者が利用可能なテスト実行ソフトウェア	開発者がコードを作成しながらテストできれば、より早い段階で問題を特定できるため、品質の向上につながります。
システム コンフィギュレーション別にテストを実施	メインのプラットフォームだけでなく、製品バリエーションについてもテストが実施されるため、製品ライン エンジニアリングの助けとなります。
テスト結果を自動的に取り込むための中央リポジトリ	中央リポジトリを使用すると、関係者は、テストの網羅状況、テスト結果、不具合をリアルタイムで確認し、市場投入までの期間、品質、コストに関する目標に影響する問題について深く理解することができます。

図 9: 検証と確認のための機能的な要件



---

定期的にテストを実施することで、製品の品質が向上し、市場で成功を収める確率が高まります。

---

## サービス要件の評価

投資利益を得るには、まずソフトウェアを実装する必要がありますが、次に、特に重要なことです。ユーザーがそのソフトウェアを導入できる必要があります。ユーザーがソフトウェアを快適に使用できる必要があります。これは、良質なトレーニング リソースを用意し、ユーザーが疑問や問題を抱えているときにサポートを提供することで実現できます。最後に、システム開発にはさまざまなエンジニアリング分野が関与することを考慮し、システム エンジニアリングのためのツールをほかのソリューションと統合する必要があります。統合によって、異なるエンジニアリング分野間のワークフローがよりシームレスになり、各エンジニアリング部門内に存在する自然な知識のサイロの克服を手助けできます。

---

統合によって、異なるエンジニアリング分野間のワークフローがよりシームレスになり、各エンジニアリング部門内に存在する自然な知識のサイロの克服を手助けできます。

---

次に示すのは、ユーザーがソフトウェア投資を活用できるようにするためのサポート要件のリストです。

要件	考慮事項
資格のあるトレーナー	採用を確実にするには、ツールおよび標準に精通したトレーナーからユーザーが指導を受けられるようにする必要があります。
専任のサポート スタッフ	ヘルプが必要になったときに、手助けを提供するリソースをユーザーが利用できる必要があります。手助けは熟練のシステム エンジニアが行うのが理想的です。サポート対応時間は、ユーザーが必要とするタイミングに合わせるようにしてください。
ツールのインストール、管理、運用に関するマニュアル	電話とコンサルティングによるサポートに加えて、ヘルプを提供するためのリソースとしてマニュアルも用意する必要があります。



ほかのツールと統合するためのアプリケーション プログラミング インタフェース (API)	システム エンジニアリング ツールは多数の部門に及ぶため、既存のツールと統合する必要があります。
----------------------------------------------	--------------------------------------------------

図 10: サービス要件

## ベンダー属性の考慮

システム エンジニアリングなどの戦略的な活動に投資する際には、ソフトウェア ベンダーに、信頼のおけるパートナーとなる能力があることが重要です。

---

**システム エンジニアリングなどの戦略的な活動に投資する際には、ソフトウェア ベンダーに、信頼のおけるパートナーとなる能力があることが重要です。**

---

次に示すのは、ベンダーに関して考慮すべき基準のリストです。

要件	考慮事項
財務的な安定性	そのベンダーは利益を上げていますか。研究開発に投資していますか。
業界標準のサポート	システム開発に使用されるツールの数を考えると、それらのツール間で情報を共有できることが、開発プロセスの調整のために役立ちます。業界標準のサポートがそうしたツールの統合をより容易にします。
製品への投資	そのベンダーは、ポートフォリオを改善し拡張するために、ポートフォリオに継続的に投資していますか。
より広範なソリューションを提供	システム エンジニアリングは非常に範囲が広いいため、ツールをほかのエンジニアリング部門と統合する必要があります。そのベンダーは、PLM や ALM などの広範な製品開発ソリューションを提供していますか。そしてそれらはニーズに応じた拡張が可能であるか、別のベンダーのソリューションと統合するオプションが用意されていますか。
ほかのベンダーとのパートナーシップに前向き	システム エンジニアリングは複数のエンジニアリング分野に及ぶため、ほかの計算 / 補完ツールとの統合が必要になります。
柔軟なライセンス	ユーザーがツールを終了すると解放されるフローティング ライセンスなどのライセンス スキームにより、ユーザーが必要なときにツールにアクセスできる柔軟性が提供されます。
経歴	システム エンジニアリングの問題の解決に関して、そのベンダーはどのような経験を持っていますか。そのベンダーには、あなたの業界の専門知識がありますか。

図 11: ベンダー要件

## 企業固有のニーズの特定

システム エンジニアリングは複雑で、数多くの職種が関与します。場合によっては、とてもうまく機能している既存のポイント ソリューションがすでに社内にあるかもしれません。そのような場合は、ツールを統合する必要があるほかのソリューションについて考慮する必要があります。さらに、企業または業界に固有の規則または法規制に準拠しなければならない場合もあります。こうした企業固有のニーズを考慮することが重要です。たとえば、御社の環境について考えてみてください。どのプログラミング言語が必要ですか。どの CAD ツールが必要ですか。

次に示すリストを企業固有の考慮事項について検討する際に参考にしてください。

要件	考慮事項
既存のエンタープライズ ツール	PLM や ALM などとの統合が必要ですか。
既存の設計ツール	CAD、EDA、シミュレーション、統合開発環境に何を使用していますか。
既存のソフトウェア インフラストラクチャ	コンフィギュレーション管理、データ管理、要件管理、エンジニアリング パフォーマンスおよび解析ツール、ほかのモデリング ツールなど、どの既存のツールと統合する必要があるか検討します。
プロセスのサポート	プロセスが既成のままでサポートできない場合は、プロセスをサポートするためのソリューションのカスタマイズをどの程度簡単に行えるでしょうか。
ローカライズ	グローバル チームのためにサポートする必要がある言語があるでしょうか。
ソフトウェア言語	C、C++、C#、Ada、Java、VB などのサポートが必要になるでしょうか。
拡張性	どの程度のサイズのプロジェクトをサポートする必要がありますか。さまざまなサイズのプロジェクトに合わせてスケールを調整できますか。
規制コンプライアンス	準拠すべき規制：ISO 26262、DO-178B、DO-178C、DO-254、SPICE、IEC 61508、IEC 62305、AUTOSAR など。

図 12: 企業固有のニーズに関する要件

## まとめ

高度なシステム エンジニアリング手法は、組み込みソフトウェアとモノのインターネットを介して手に入るイノベーションを活用するための鍵となります。スマート コネクティッド デバイ

スを開発するチャンスを手にした企業は、自社の製品と競合の製品の差別化を図ることができ、新規顧客の獲得や収益の拡大が可能になります。ただし、機械、電気、ソフトウェアの各コンポーネントからなるシステムを統合する作業は複雑です。この複雑性は、異なるコンフィギュレーションの顧客が抱えるさまざまなニーズを企業が満たそうとするとときに、飛躍的に増大します。複雑性とは、事態がうまくいかなくなるリスクがあることを意味します。これらの問題はビジネスに大きく影響し、製品の収益性を損ねるおそれがあります。高度なシステム エンジニアリング手法と、それをサポートする適切なソフトウェア ツールを同時に導入することで、この複雑性に対処し、収益性の高い製品を容易に市場に投入できるようになります。現代のスマートコネクティッド デバイスは比較的シンプルではありますが、進化して安全性に影響を与えるような重要なサービスを提供するようになっていく中で、これらのデバイスの複雑性が増し、同程度の高度なシステム エンジニアリング手法が必要になります。成長を視野に入れている企業は、現在のニーズと将来のニーズの両方を考慮しなければなりません。

---

## 高度なシステム エンジニアリング手法は、組み込みソフトウェアとモノ の インターネットを介して手に入るイノベーションを活用するための鍵となります。

---

しかし、システム エンジニアリングにはあまりに数多くの側面があるため、各企業に適したソリューションを特定するのが非常に難しい場合があります。このガイドで示しているようなツールやプロセスの評価基準の大まかなリストを使用すると、機能面または技術面の詳細なレビューを行う前に、ソリューションおよびパートナーとの相性を見極める簡単な "リトマス試験" を行い、ソリューションの候補を絞り込むことができます。最終的に重要なのは、ソリューションを選択する際には、機能、サービス、ベンダー、特殊な要件がすべて考慮されることです。

## 推奨事項

業界での経験と本レポートの調査結果に基づき、Tech-Clarity では次の事項を推奨します。

- 企業のニーズ、既存のアプリケーション、業界、独自の製品とプロセスの要件に基づいてシステム エンジニアリング要件を特定し、検討する
- ソリューションを詳細に評価する前に、このガイドで示しているような大まかな要件を使用して、ソリューションと企業の相性を調べる
- ビジネスおよびプロセスの長期的な成長ニーズと、複数の製品ライン、部門、エンジニアリング分野にわたる潜在的な拡張性を考慮する
- システム エンジニアリング ソリューションに投資する際は、プロセスから要件、設計、検証まで、システム エンジニアリングの全段階について検討する
- 信頼のおけるパートナーとなる能力を持つベンダーを選択する



## 執筆者について

ミッシェル・ブーシェ氏は、リサーチ会社 Tech-Clarity のエンジニアリング ソフトウェア リサーチ担当副社長です。これまで 20 年以上、エンジニアリング、マーケティング、マネジメントの各分野に携わり、アナリストとして活躍してきました。製品設計、シミュレーション、システムエンジニアリング、メカトロニクス、組み込みシステム、PCB 設計、製品性能の改善、プロセス向上、マス カスタマイゼーションなど、その経験は多岐にわたります。バブソン大学で MBA を取得し、極めて優秀な成績で卒業。ウースター工科大学で優秀な成績を収め、機械工学の学士を取得しています。

ブーシェ氏は、Pratt & Whitney 社および KONA 社 (現在の Synventive Molding Solutions) で機械エンジニアとしてキャリアをスタートさせました。その後の 10 年間は、MCAD および PLM の大手ソリューション プロバイダである PTC に在籍。PTC では、テクニカル サポート、マネジメント、製品マーケティングの業務を通じてエンド ユーザーのニーズへの理解を深めました。射出成形シミュレーションの市場リーダー Moldflow Corporation (後に Autodesk が買収) ではテクニカル マーケティングを担当。製品の位置づけおよび販売戦略の策定に尽力しました。その後、Aberdeen Group に入社して、製品イノベーション、製品開発、エンジニアリング プロセスに携わり、製品イノベーションおよびエンジニアリング手法を手がけました。

ブーシェ氏は経験豊富な調査員であり、執筆者です。これまでにベンチマーク調査を行った製品開発関係者は 7,000 人以上に及び、製品開発に関するレポートを 90 本以上発表しています。現在は主に、製品、市場、設計環境、バリュー チェーンを取り巻く今日の複雑な状況を打開し、収益を向上しようと努力している企業を支援しています。