

JASAレポート: ソフトウェア駆動型社会への変革と 日本の競争戦略

～組込みソフトウェア業界の提言～
人月からの脱却とSDxによる新価値創造へ



一般社団法人 組込みシステム技術協会(JASA)
JASA改革プロジェクト

2025年12月

本レポートの目的と読み方について

本レポートは、一般社団法人組込みシステム技術協会(JASA)が作成する従来の技術レポートとは、その性質と目的を大きく異にしています。特定の技術要素を深く掘り下げた詳細な技術文書ではなく、急速に変化する技術環境と市場動向を踏まえ、組込みソフトウェア業界が今後進むべき方向性とビジネスモデルの変革について提言するものです。

ソフトウェア駆動型社会(SDx)への移行と生成AIの台頭という歴史的な転換点において、私たちは前例のない変化に直面しています。このような状況では、完璧な分析と確実な答えを待つていては、変化の速度に取り残されてしまいます。本レポートは、現時点における私たちの最良の理解と仮説を提示するものであり、すべての記述が絶対的な正解であることを保証するものではありません。

むしろ、本レポートは「完璧な地図を待つより、不完全でも羅針盤を持って一步を踏み出す」という考え方に基づいています。SDxのソフトウェア開発と同様に、時間をかけて厳密で完全なものを作り上げるよりも、現段階での気づきや見通しを仮説として提示し、業界全体での議論と実践を促すことを優先しています。いわば、アジャイル的なレポートとして位置づけられます。

したがって、本レポートは一度限りの完成版ではなく、継続的に更新されていく生きた文書です。会員企業からのフィードバック、技術環境の変化、実践から得られた知見を取り込みながら、定期的に内容を見直し、より実効性の高いものへと進化させていきます。読者の皆様には、記載内容を批判的に検討し、ご自身の組織での実践を通じて検証し、その結果をJASAや業界全体と共有いただくことを期待しています。

本レポートで提示する戦略や提言は、政府機関や業界有識者へのヒアリング、国内外の文献調査、JASA改革プロジェクトメンバーでの議論を基にしていますが、学術論文のような厳密な検証プロセスを経たものではありません。むしろ、実務家の視点から、業界が直面する課題に対する実践的な解決の方向性を示すことに重点を置いています。

技術革新の波の中で、確実な答えはありません。しかし、確実な道筋がないからこそ、仮説を立て、実行し、学習し、改善するという姿勢が重要です。本レポートが、その第一歩を踏み出すきっかけとなり、業界全体での建設的な対話と変革への行動を促す触媒となることを願っています。

本レポートの位置づけ

日本の製造業がソフトウェア駆動型社会への変革を遂げるためには、国全体の政策方針から個別産業分野の具体的な実践まで、段階的かつ体系的なアプローチが求められます。本レポートは、この大きな変革の流れの中で、三つの重要な提言の系譜を継承するものです。

第一段階として、経済産業省「PIVOT」戦略が、日本の製造業におけるソフトウェアの戦略的重要性を提示しました。製品価値の源泉がハードウェアからソフトウェアへと移行し、グローバル競争を勝ち抜くためにはビジネスモデルの根本的な転換が必要であるという基本認識が示されました。

第二段階として、IPA「ソフトウェアモダナイゼーション委員会」が2025年3月に発表したレポートが、2030年を転換点と位置づけ、AIを駆使したソフトウェア開発による労働集約型からの脱却という具体的なビジョンを提示しました。組織・人材・技術・制度とインフラという四つの観点から、Software-Defined Society実現に向けた包括的な取り組みの方向性が示されています。

第三段階として、本JASAレポートは、これらの政策・戦略を製造業向け組込みソフトウェアという特定領域に特化させ、より実践的な提言を行います。自動車、ロボティクス、医療機器、産業機械など、日本が世界的競争力を持つ領域において、リアルタイム性・安全性・信頼性という従来の要求に加え、SDxやAIという新たな技術的課題への対応策を具体的に示します。

このように、国の政策レベルでの大方針、産業横断的な技術戦略、そして個別産業分野における実践的提言という三段階のアプローチを通じて、日本の製造業がソフトウェア駆動型社会への変革を着実に進めるための道筋が整備されつつあります。本レポートは、この系譜の最終段階として、製造業向け組込みソフトウェアの領域において、具体的かつ実行可能な変革の指針を提供するものです。

エグゼクティブサマリ

背景と問題意識

日本の製造業は、製品価値の源泉がハードウェア単体からソフトウェアの継続的なアップデートによって創出されるサービスへと移行する「SDx（ソフトウェアデファインド）」の時代を迎えていました。同時に、生成AIの爆発的な普及は、組込み開発のあり方を根本から変えつつあります。

しかし、日本のソフトウェア開発現場は労働時間で価値を測る「人月ビジネス」に固執しており、SDxが求める数週間から数日単位での迅速なアップデートとの間に大きなギャップが存在します。本レポートは、SDxと生成AIの波を乗りこなし、持続的な成長を実現するための戦略的羅針盤として策定されました。

SDxと生成AIがもたらす技術的変革

SDx時代の組込みソフトウェア開発では、従来のクローズドシステムからOTAを通じて継続的に更新されるオープンなソフトウェアプラットフォームへの転換が求められます。この実現には、サービス指向アーキテクチャ(SOA)、CI/CDパイプラインといったモダンなアーキテクチャへの移行が不可欠です。これがソフトウェア再利用の基盤となり、加えて、生成AIの活用により、開発プロセスの効率化が劇的に進み、変更から展開までの時間が数週間から数時間へと短縮されます。

組込みAIは、従来型のルールベース、AI支援型、エンド・ツー・エンドという三つのパラダイムへと進化しています。この変革期において、日本企業は莫大な資本を必要とする基盤モデル開発で米中と競争するのではなく、グローバルなAI基盤を活用し、それを安全に社会実装する「翻訳層」としての役割に注力すべきです。

具体的には、ドメイン特化型のAI技術と深い現場知識を組み合わせ、汎用的なAI基盤技術を人命や社会インフラを支えるリアルな現場に実装・統合するラストワンマイルを担うことです。これは、製造業で長年培われた現場理解、品質へのこだわり、ミッションクリティカルな分野での高い信頼性という日本の強みを最大限に活かす戦略です。

スマートフォン時代とは異なり、産業機器や自動車、医療機器といった高信頼性製品の世界では、十年から二十年以上使い続けられることが前提であり、人命に関わるレベルの安全性とリアルタイム性が求められます。この要求水準の高さと、現場に根ざした深いドメイン知識の必要性こそが、米中の巨大企業が簡単には参入できない領域であり、日本企業が価値を発揮できる戦場なのです。

日本企業が直面する三つの根本的課題

第一に、SDx時代に対応した組込みソフトウェア開発手法の確立です。数日から数週間単位での迅速なアップデートを実現するには、システム全体をスコープとするソフトウェアプラットフォームを構築する必要があります。これによりモダンなソフトウェア開発が可能となり、ソフトウェアや知見の再利用が引き起こされ、従来の人海戦術に依存した開発プロセスから脱却し、知識集約型に進化させます。

第二に、生成AI時代における開発パラダイムの適切な活用です。三つの開発パラダイムを使い分け、日本企業の強みを活かした専門特化戦略を実行する必要があります。

第三に、ビジネスモデルの労働集約型から知識集約型への変換です。人月ビジネスは、SDx時代の継続的な価値開発と爆発的な生産性という要件と根本的に非互換であり、ベンダーは労働力提供から価値創出パートナーへの転換が求められています。

ベンダーの新たな道筋：二層構造による共存共栄モデル

本レポートは、組込みソフトウェア開発ベンダーに対して二つの戦略的アプローチを提示します。一つ目はSDxプラットフォームの開発者としての道であり、業界協力によって日本のモノづくり産業全体が利用できる基盤を開発します。このプラットフォームは、日本が長年培ってきた安全性と信頼性という強みを体現し、継続的なアップデートやクラウド連携といった動的な環境においても高い信頼性を維持できる仕組みで競争優位性を確立します。

二つ目は高度なナレッジ提供者としての道であり、生成AIと蓄積された専門知識を組み合わせたサービス提供を通じて顧客の課題解決に深くコミットします。この変革は、生成AI活用による効率化、専門性の向上およびナレッジのデジタル化とサービス化、プラットフォーム化による収益安定化という3つのステップで段階的に進めることができます。

重要なのは、この二つの道が相互排他的ではなく、二層構造による共存共栄モデルとして機能するという点です。プラットフォームの開発、そして利用には高度なナレッジが必要ですし、開発効率向上には生成AIの活用は不可欠です。また、プラットフォームを持つことによって、AIが必要とするデータを獲得し、それを開発に反映する、というDevOpsサイクルが実現され、それが優れたプロダクトとサービスの基盤となります。

JASAの提供価値

JASAは、課題の明確化と提言、技術的解決策の提示、ビジネスモデル変革の支援、人材育成とコミュニティ形成、そして共通プラットフォーム開発の促進という五つの観点から、日本の産業界がSDx時代に対応するための多様な価値を提供します。特に、合同プロジェクトの推進、参照アーキテクチャの策定、ベストプラクティスの文書化、成功事例の共有、技術セミナーやワークショップの開催を通じて、会員企業の変革を具体的に支援してまいります。

結論

SDxと生成AIがもたらす変革は、組織文化、ビジネスモデル、人材の根本的な変革を伴うものです。この変革は一朝一夕には実現しませんが、各ステップを着実に進め、失敗から学び、継続的に改善していく姿勢が必要です。日本企業は、製造業で培われた現場理解、品質へのこだわり、安全性・信頼性という強みを活かしながら、新たな競争優位性を確立できます。JASAは、この変革を推進する中核的な存在として、会員企業を支援し、日本の製造業の持続的な成長に貢献してまいります。長期的な視点に立てば、SDxと生成AIが社会に実装されることで、従来の市場をはるかに超える爆発的な市場拡大が広がっています。この大きな機会を捉えるための戦略的な準備と実行が、今まさに求められています。

目次

エグゼクティブサマリ

序章：ソフトウェア駆動型社会への変革

第1章：SDx時代の組込みソフトウェア開発

- 1-1 SDxによる「クローズドからオープンへ」の流れ
- 1-2 ソフトウェアアーキテクチャの大転換
- 1-3 生成AIによる開発プロセスの効率化
- 1-4 迅速に開発を回すための要素技術
- 1-5 オープンシステム・ディペンダビリティ

第2章：組込みAI開発の新たな潮流と日本の競争戦略

- 2-1 3つの開発パラダイム
- 2-2 SDxとAIの相互作用
- 2-3 AIエコシステムにおける「ラストワンマイル戦略」の勝機
- 2-4 生成AI民主化と組込み業界への影響
- 2-5 3つのパラダイムの統合戦略と未来への道筋

第3章：ベンダーの新たな道筋と変革

- 3-1 メーカー単独での内製化の困難とベンダーの必要性
- 3-2 従来のビジネスモデルの限界
- 3-3 共通SDxプラットフォームの構築者という道
- 3-4 『人』から『知』へ：AI時代におけるソフトウェアベンダー成長の道
- 3-5 二層構造による共存共栄モデル

第4章：JASAの提供価値とアクションプラン

- 4-1 課題の明確化と戦略的提言
- 4-2 技術的解決策の提示
- 4-3 ビジネスマネジメントの支援
- 4-4 人材育成とコミュニティ形成
- 4-5 共通プラットフォーム開発の促進

おわりに

序章:ソフトウェア駆動型社会への変革

日本の製造業は、経済産業省が掲げる「PIVOT」戦略を背景に、国内中心の事業構造からグローバル市場志向へと大きく舵を切りつつあります。この転換期において、製品価値の源泉はハードウェア単体から、ソフトウェアの継続的なアップデートによって創出される「サービス価値」へと移行しています。こうした変革は「SDx(Software Defined Everything)」として定義され、従来の“売り切り型”モデルを、ソフトウェアによる継続的価値提供を軸とした“サブスクリプション型”または“サービス型”ビジネスへと進化させています。

SDxの潮流は、自動車、ロボティクス、医療機器、産業機械など、あらゆる分野の“ものづくり”的定義を再構築しています。製品に内包されるソフトウェアサービスが絶えず進化し続けることで、新たな顧客価値が創出され、企業競争力の持続性が確保されるのです。すなわち、SDxはもはや単なるITインフラの革新ではなく、製造業の構造そのものを変革する概念です。製品の価値源泉をハードウェアからソフトウェアへと移し、顧客に“継続的な体験価値”を提供する仕組みこそがSDxの本質といえます。

かつての製造業は、製品販売によって価値が固定化される「売り切り型」ビジネスを前提としていました。しかし現代の顧客は、製品そのものに加え、利用体験やサービスの継続的改善にも価値を見出しています。この“顧客起点の価値創出”こそがDX(デジタルトランスフォーメーション)の核心であり、SDxはその実現を支えるテクノロジー基盤なのです。製品をソフトウェアで定義し、制御・更新を可能にすることで、企業は市場の変化に即応し、継続的な価値提供を実現します。

SDxによって製品は、ハードウェアの静的な機能を超えて、ソフトウェアの更新を通じて進化し続ける「動的なプラットフォーム」へと変貌します。

たとえば、自動車はもはや単なる移動手段ではなく、インフォテインメントや自動運転、安全支援、パーソナライズド機能などをソフトウェア経由で提供する「Software Defined Vehicle(SDV)」へと進化しています。この継続的アップデートの仕組みが、顧客との関係深化と新たな収益機会を同時に生み出しているのです。

SDx時代の開発と生成AIの必然性

SDxの前提となるのは「継続的な価値提供」です。開発は一度きりではなく、顧客や市場からのフィードバックを迅速に反映し、短いサイクルで改良・更新を重ねることが求められます。しかし、この“高速かつ持続的な開発”は、従来の人手依存の開発手法ではもはや限界に達しています。ここで登場するのが、生成AIという技術革新です。

生成AIは、SDxが要求する開発スピードと品質を両立させるための鍵となります。コードやテストケース、ドキュメントの自動生成により反復作業を効率化し、エンジニアを創造的な課題解決に集中させます。すなわち、SDxという「目的地」に向かうために、生成AIは不可欠な「推進力」となるのです。

製品機能を変革するAI技術

AI技術の進化は、開発プロセスの効率化にとどまりません。ディープラーニングやエンド・ツー・エンド学習により、従来のルールベース制御では不可能だった複雑な挙動を実現します。ヒューマノイドロボットの自律動作、完全自動運転、柔軟な組立作業を行う産業用ロボットなどは、その典型例です。これらの製品は膨大なデータ学習を通じて環境適応性を獲得し、AIモデルの継続的更新によって進化し続けます。

したがって、AIを搭載した製品は必然的にSDx化されます。AIモデルは継続的学习と改善を前提とするため、実運用データをフィードバックし、モデルを更新し、OTA(Over-The-Air)で配信するサイクルが不可欠です。AIとSDxは相互補完的な関係にあり、AIが製品価値を創出し、SDxがその価値を継続的に届けるアーキテクチャを支えます。

SDx × AI — 二重の革新軸

こうして見していくと、SDxとAIは独立した技術トレンドではなく、「開発プロセスの革新」と「製品機能の革新」という二重の相互依存関係を形成しています。両者が一体となって製造業の価値創造プロセスを再構築し、かつてないスピードと柔軟性で市場変化に応答する時代を切り開こうとしているのです。

日本の課題と変革の方向性

一方、日本のソフトウェア開発現場はいまだに「人月ビジネス」という旧来の構造に依存しています。労働投入量を価値基準とするこの仕組みは、効率化や自律的イノベーションを阻害し、SDxが要請する高速かつ継続的な開発サイクルとの間に構造的な乖離を生じさせています。

このギャップを克服できなければ、日本の製造業はグローバル競争において持続的な優位性を維持することが難しくなるでしょう。

しかし、SDxと生成AIの融合は、この停滞を打破する革新的な契機でもあります。両者の統合は、従来の労働集約型モデルから知識集約型モデルへの転換を促し、製造業全体の生産性構造を根本から変革する可能性を秘めています。短期的には体質改善の痛みを伴うものの、長期的には従来市場を凌駕する新たな成長エコシステムを形成するでしょう。

本レポートでは、この構造転換の本質を明らかにし、日本企業がこの変革の潮流を捉え、次世代の競争優位を確立するための戦略を提示します。

レポート構成

- 第1章: SDx時代における組込みソフトウェア開発の課題と解決策を記述します。
- 第2章: 生成AI時代における組込みソフトウェア開発の課題と解決策を記述します。
- 第3章: 労働集約型から知識集約型へのビジネスモデル転換を記述します。
- 第4章: これらの変革に対してJASAが果たすべき役割と提供価値を記述します。

序章・エグゼクティブサマリ・はじめに 参考文献一覧(APA形式)

経済産業省. (2023). PIVOT戦略—ソフトウェア駆動型社会への転換. 東京: 経済産業省.

独立行政法人 情報処理推進機構 (IPA). (2025). ソフトウェアモダナイゼーション委員会 最終報告書. 東京: IPA.

経済産業省. (2024). デジタル経済レポート 2024: データ駆動型社会における産業構造変革. 東京: 経済産業省.

一般社団法人 組込みシステム技術協会 (JASA). (2024). ソフトウェア駆動型社会への変革と日本の競争戦略. 東京: JASA.

経済産業省. (2023). 製造業DXレポート 2023: サイバーフィジカルシステムへの挑戦. 東京: 経済産業省.

日本経済新聞社. (2024). 製造業のソフトウェア化がもたらす産業構造転換. 日経産業新聞, 2024年6月号.

国立研究開発法人 産業技術総合研究所 (AIST). (2023). CPS(サイバーフィジカルシステム)ロードマップ2023. 東京: AIST.

内閣府. (2023). Society 5.0実現に向けた統合戦略 2023. 東京: 内閣府.

OpenAI. (2024). The Impact of Generative AI on Software Development Productivity. San Francisco, CA: OpenAI.

Microsoft Research. (2024). Copilot and Software Development Transformation. Redmond, WA: Microsoft.

経済産業省. (2024). 生成AIの産業活用に関する検討会報告書. 東京: 経済産業省.

日経クロステック. (2024). 生成AIが変える開発現場——人月ビジネスの終焉. 日経クロステック, 2024年5月号.

Gartner. (2024). Software-Defined Everything (SDx) Market Forecast 2024–2030. Stamford, CT: Gartner Inc.

IDC Japan. (2024). 国内SDx市場予測2024–2028年: オープンプラットフォーム化の進展. 東京: IDC Japan.

Red Hat. (2023). From Monolithic to Modular: Modern Software Architectures for Industry 4.0. Raleigh, NC: Red Hat Inc.

Siemens AG. (2024). Digital Transformation in Manufacturing: The Role of SDx and AI. Munich: Siemens AG.

World Economic Forum. (2024). The Future of Manufacturing and Software-Defined Industry. Geneva: WEF.

McKinsey & Company. (2024). Reinventing Industrial Competitiveness through Software and AI. New York, NY: McKinsey Global Institute.

Accenture. (2024). Software-Driven Manufacturing: Pathways to Digital Value. Dublin: Accenture.

OECD. (2024). Digital Transformation and Industrial Policy 2024. Paris: OECD Publishing.

第1章:SDx時代の組込みソフトウェア開発

1-1 SDxによる「クローズドからオープンへ」の流れ

SDxは、ソフトウェアの「アーキテクチャ(構造)」と「ライフサイクル(開発・運用)」に関する根本的な変革です。従来の組込みシステムが、一度出荷したらほとんど変わらない固定的なクローズドシステムだったのに対し、SDxはOTA(無線通信)を通じて機能が継続的に更新・追加されることを前提としています。これにより、システムは常に変化・進化し、新しい機能やサービスを柔軟に取り入れられるオープンなプラットフォームとなります。

従来の組込みシステムは、動作するハードウェア、接続される機器、実行される環境のすべてが事前に厳密に定義された、クローズドな環境で動作することを前提に設計されてきました。このような環境では、システムの振る舞いを完全に予測し、すべての状況を事前にテストすることが可能でした。開発段階で想定されるすべての状況をテストし、仕様通りに動作することを確認するという静的なアプローチが機能していました。

しかし、SDxによって実現されるシステムは、まったく異なる性質を持ちます。システムはOTAを通じて継続的にアップデートされ、外部のクラウドサービスと連携し、他のデバイスと動的に接続します。ユーザーの使い方も多様化し、想定外の環境や状況でシステムが動作することも珍しくありません。つまり、システムはオープンな環境で動作するようになり、事前にすべての状況を網羅的に予測・検証することは現実的ではなくなります。ただし、これは「予測と制御を完全に放棄する」ことを意味するのではありません。重要なのは、予測可能な範囲とその境界条件を明確に定義し、その範囲内での動作を保証しつつ、境界を超えた予期せぬ事態に対する適応メカニズムを組み込むことです。

SDx時代における組込みソフトウェア開発は、従来の開発手法の延長線上にはありません。数週間から数日単位での迅速なソフトウェアアップデートが求められるため、これまでの人海戦術に依存した品質保証手法や開発プロセスは非効率的となります。この「クローズドからオープンへ」という変化は、システムの品質や信頼性をどう保証するかという問題に、根本的な問い合わせかけます。

1-2 ソフトウェアアーキテクチャの大転換

SDxプラットフォームの実現には、サービス指向型アーキテクチャ(SOA)に基づき、オープンなAPIを提供し、そのプラットフォームとセットになって構成されるツールチェーンによる開発環境の標準化、そしてその二つによってもたらされるCI/CDを含むDevOpsプロセスがモダンな組込みシステム開発基盤となります。

システムを独立性の高いサービスコンポーネントに分割し、それをアプリケーションインターフェースとすることでSDxで重要となるハードウェア・ソフトウェア分離と、オープンなAPIを実現します。従来のモノリシックなソフトウェア構造では、機能が密結合しており、一部の変更が全体に影響を及ぼす可能性がありました。しかし、サービス指向アーキテクチャでは、各サービスが明確に定義されたインターフェースを通じてのみ相互作用するため、個別のサービスを独立して更新・テスト・デプロイできます。

特に、ミッションクリティカルな組込みシステムで重要となるリアルタイム性などを担保できるソフトウェアフレームワークを導入することが重要となります。このアーキテクチャの大転換により、ソフトウェアの再利用性が飛躍的に向上し、開発効率が大幅に改善されます。また、複数のチームが並行して異なるサービスを開発できるため、開発スピードも向上します。

このアーキテクチャ変革は、単なる技術的な選択ではなく、SDx時代の継続的な価値提供を実現するための必須要件です。オープンなAPIを通じて、サードパーティの開発者や他のシステムとの連携も容易になり、エコシステムの形成にもつながります。

1-3 生成AIによる開発プロセスの効率化

開発プロセスの抜本的な効率化を目的として、「生成AI」の積極的な活用が求められます。生成AIは、要件定義からコーディング、テストケース考案、API仕様書生成といった反復的なタスクを自動化し、エンジニアがより創造的な問題解決に集中することを可能にします。

従来、ソフトウェア開発の多くの時間は、定型的な作業に費やされてきました。たとえば、APIの仕様書を書く、テストケースを網羅的に列挙する、データベーススキーマに対応したCRUD(Create、Read、Update、Delete)操作のコードを書く、といった作業です。これらの作業は重要ですが、高度な創造性を必要としません。

生成AIは、このような反復的なタスクを劇的に効率化します。たとえば、要件定義の段階では、自然言語で記述された要求を解析し、それを実現するために必要な機能の一覧や、システムアーキテクチャの初期案を生成できます。コーディングの段階では、関数の仕様を説明するだけで、その実装コードを生成できます。テストの段階では、関数の仕様から網羅的なテストケースを自動生成し、さらにそのテストコードも自動生成できます。

重要なのは、生成AIは人間のエンジニアを置き換えるのではなく、その能力を拡張するツールであるということです。生成AIが生成したコードやテストケースは、必ず人間のエンジニアがレビューし、プロジェクトの文脈に合わせて調整します。しかし、ゼロから作成するのではなく、AIが生成した初期案をベースに改善していくことで、生産性は飛躍的に向上します。この効率化によって生まれた時間を、より高度な設計判断や、ユーザーエクスペリエンスの創出に振り向けることができます。

1-4 迅速に開発を回すための要素技術

プラットフォームにより、デプロイ後のシステムから各種アプリケーションの振る舞いやユーザの利用状況などのデータを適切な形式と量で得ることが可能になります。これにより、システムのオペレーションデータを継続的に行われるシステム、ソフトウェア、プラットフォームの開発に活用し、継続的に価値を高めることができます。

迅速な新たなソフトウェアのデプロイを実現するために、自動化されたテスト、静的解析、安全認証への適合性検証をCI/CDに組み込み、品質を維持しながら開発スピードを向上させます。このCI/CDに不可欠となるツールチェーンの統合や標準化などはプラットフォーム化の重要な効果の一つです。

IoTからAI、生成AI、CPS、OTAまで、先端技術が連携し、製品に継続的な進化と価値をもたらす革新的なサイクルを実現します。SDxプラットフォームがAPIの利用を介してシステムの利用状況

や必要な物理データの収集を行います。これは単にセンサの物理データなどに限らず、特にAPIの上で動作するアプリケーションの振る舞いや、その入出力データがプラットフォームを介して収集可能となり、システム全体の振る舞いやユーザの利用データが入手できることが、次のバージョンのシステムやソフトウェアの価値向上に大きく寄与します。

収集されたデータは機械学習モデルによって解析され、製品のパフォーマンス改善や新機能開発の根拠となるインサイトが生成されます。解析結果に基づき、生成AIはソフトウェアのコードを迅速かつ効率的に生成・修正を行います。生成されたソフトウェアはCPS環境(デジタルツイン)でシミュレーションされ、その有効性や安全性が検証・調整されます。検証済みのソフトウェアアップデートはOTAを通じて製品に遠隔でデプロイされ、継続的な機能向上と価値提供を実現します。物理空間でエッジAIがアクチュエータ(機械)に指令を出します。このサイクルを高速で回すことで、SDx時代の製品は、顧客のニーズや市場の変化に迅速に対応し、継続的に価値を提供し続けることができます。

1-5 オープンシステム・ディペンダビリティ

従来の「品質」や「信頼性」という言葉自体が、「工場で製品を作り、テストして、出荷したら終わり」という静的なシステムを前提としたニュアンスを持っています。SDx時代のシステムは、出荷後も進化し続け、常に変化する環境の中で動作します。このような動的なシステムに対しては、新しい概念が必要です。

ここで注目すべき概念が「ディペンダビリティ」です。ディペンダビリティとは「依存できる性質」「頼りにできること」を意味し、その本質は動的で継続的なものです。システムが常に変化し、未知の状況に直面する中でも、「このシステムに依存できる」という状態を維持し続けることを表現しています。従来の品質や信頼性が「正しく作られているか」という製品の属性を表すのに対し、ディペンダビリティは「継続的に頼りにできるか」というサービスの性質を表します。

SDx時代のシステムに求められるのは、「オープンシステム・ディペンダビリティ」です。これは、オープンな環境下で動作するシステムが、未知の状況に直面しても、継続的に依存できる状態を維持し続ける性質を指します。その本質は、事前の完全な予測や制御を諦め、代わりに継続的な監視と適応を通じて信頼性を確保するという考え方になります。

オープンシステム・ディペンダビリティを実現するための代表的なアプローチが、DEOSプロセスや関連する国際規格であるIEC 62853です。DEOSプロセスは、システムの稼働中に継続的に監視を行い、想定外の状況を検出し、適切に対処するというサイクルを継続的に回すことで、動的な環境下での信頼性を保証します。IEC 62853は、システムが外部のサービスやコンポーネントに依存する場合に、その依存関係を明確に管理し、依存先の変化や障害に適切に対処する方法を定めています。これらに共通するのは、「完全な予測と制御」から「継続的な監視と適応」へという発想の転換です。

オープンシステム・ディペンダビリティという概念において、日本の製造業が長年にわたり培ってきた「安全・安心」に関するナレッジは、重要な競争優位性となります。自動車の機能安全規格ISO 26262や医療機器のソフトウェアライフサイクルプロセスIEC 62304といった業界固有の知識を、オープンシステム・ディペンダビリティという新しい枠組みの中で再定義し、SDxプラットフォームに組み込むことで、日本独自の競争優位性を確立できます。

グローバルなクラウドプラットフォームやソフトウェアフレームワークは、汎用性と拡張性を重視して設計されていますが、特定の業界の厳格な安全要件や規制要件に対応するには、追加の開

発やカスタマイズが必要になることがあります。一方、日本の製造業は、自動車、医療機器、航空宇宙、産業機械など、高い安全性と信頼性が要求される分野で世界をリードしてきました。重要なのは、日本が持つ安全・安心に関する深い知見を、静的なクローズドシステムの枠組みから解放し、オープンシステム・ディペンダビリティという動的な枠組みの中で再構築することです。これにより、日本の製造業は、SDx時代においても「依存できるシステムを作る力」で世界をリードし続けることができるのです。

第1章 参考文献一覧(APA形式)

- Gartner. (2024). Software-Defined Everything (SDx): From Concept to Practice 2024. Stamford, CT: Gartner Inc.
- IDC Japan. (2024). 国内SDx市場予測2024–2028年: IoT・クラウド・エッジの統合動向. 東京: IDC Japan.
- Red Hat. (2023). Modern Application Architectures: From Monolith to Microservices. Raleigh, NC: Red Hat Inc.
- Docker Inc. (2023). Docker and Kubernetes for Embedded Systems. San Francisco, CA: Docker Inc.
- ThoughtWorks. (2024). Software Architecture for Continuous Delivery. London: ThoughtWorks Publishing.
- Humble, J., & Farley, D. (2010). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Boston, MA: Addison-Wesley.
- Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A Software Architect's Perspective. Boston, MA: Addison-Wesley.
- GitHub. (2024). GitHub Copilot for Embedded Development: AI-assisted Productivity Report. San Francisco, CA: GitHub Inc.
- Microsoft Research. (2024). Generative AI for Software Engineering: The Copilot Revolution. Redmond, WA: Microsoft Research.
- OpenAI. (2024). Code Generation and Reasoning Models: A Technical Summary. San Francisco, CA: OpenAI.
- 経済産業省. (2024). 生成AIがもたらす開発業務効率化と人材変革. 東京: 経済産業省.
- Laprie, J. C. (1995). Dependable Computing: Concepts, Limits, and Challenges. Proceedings of the 25th International Symposium on Fault-Tolerant Computing, 42–54.
- Tokoro, M., & Ishikawa, Y. (2014). Open Systems Dependability. Tokyo: Springer Japan.

Ishikawa, Y., Tokoro, M., & Matsuno, Y. (2019). Dependability Engineering for Open Systems. Tokyo: Springer Japan.

経済産業省. (2023). DEOS(Dependable Embedded Operating System)プロジェクト成果報告書. 東京: 経済産業省.

IEC. (2023). IEC 62853: Open Systems Dependability. Geneva: International Electrotechnical Commission.

一般社団法人 組込みシステム技術協会 (JASA). (2023). 組込みシステム開発の現状と将来動向2023. 東京: JASA.

独立行政法人 情報処理推進機構 (IPA). (2023). ソフトウェア・エンジニアリング白書 2023. 東京: IPA.

経済産業省. (2024). 製造業DXレポート2024: 安全性と信頼性の再定義. 東京: 経済産業省.

国立研究開発法人 産業技術総合研究所 (AIST). (2023). サイバーフィジカルシステム(CPS)実現に向けたロードマップ2023. 東京: AIST.

ISO. (2022). ISO/IEC 30141: Internet of Things (IoT) Reference Architecture. Geneva: ISO.

Siemens AG. (2024). Digital Twin and CPS Integration Framework. Munich: Siemens AG.

Tesla, Inc. (2023). OTA Software Update Platform Architecture. Palo Alto, CA: Tesla Inc.

第2章:組込みAI開発の新たな潮流と日本の競争戦略

近年、生成AIは急速な進化を遂げ、その普及は社会のあらゆる側面を変えつつあります。この技術の進化は、組込み開発のあり方を根本的に変革し、新たな潮流を生み出しています。これまで実現不可能とされたヒューマノイドロボットや完全自動運転のような複雑なシステムが、AIによって現実のものとなりつつあります。最終的に社会実装される局面では、これらの技術は必ずデバイスに搭載される組込みソフトウェアとして形になります。このため、組込みソフトウェアの市場は爆発的に拡大し、その価値はさらに増していくことになります。

2-1 3つの開発パラダイム

3つの開発パラダイム

AIの活用度合いによって、組込みソフトウェア開発は、従来の厳格なアプローチから、よりデータ駆動型のアプローチへと進化しています。この進化は、以下の3つの主要なパラダイムに分けることができます。

第一のパラダイムは、従来型開発、すなわちルールベースによる開発です。このパラダイムは、厳格な検証プロセスと確立された認証体系に支えられた、決定論的アプローチによる安全性と信頼性の確保を特徴とします。システムの振る舞いは、人間があらかじめ定めたルールに基づいて予測可能であり、医療機器や航空宇宙分野、産業用制御システムなど、最高レベルの安全性が要求される分野で不可欠な技術基盤です。このアプローチは今後も、人命に直結する制御や、確定的な動作が法的に要求される領域において中心的な役割を果たし続けます。

開発には、形式手法やモデルベース開発(例:MathWorks社のSimulink/Stateflow)が用いられ、システム全体の振る舞いを厳密に記述・検証します。品質保証は、すべての入力パターンに対する網羅的なテストを中心とする演繹的な手法によって行われます。この手法は、予測可能な振る舞いを要求されるタスクには最適ですが、不確定な外部環境に対応する柔軟性には限界があります。

たとえば、産業用ロボットの安全制御システムでは、すべての動作は事前に定義されたルールに基づいて実行されます。人間がロボットの動作範囲に入った場合、センサーがそれを検出し、事前に定義されたルールに従ってロボットは即座に停止します。このような決定論的な動作は、安全性を保証する上で極めて重要です。

第二のパラダイムは、AI支援型開発です。このパラダイムでは、人間とAIが協力して開発を進めます。生成AIを活用して、従来のルールベース開発における反復的なタスクを効率化します。生成AIは、コード生成(例:GitHub Copilot)、テストスクリプトの作成、ドキュメントの自動生成といったタスクを支援し、エンジニアがより創造的な問題解決に集中することを可能にします。

AIはあくまで人間の能力を増強するツールとして機能し、最終的な意思決定と責任は人間が担います。このアプローチは、従来の開発プロセスを維持しつつ、生産性を向上させる現実的な手段であり、すでに多くの現場で導入が進んでいます。

たとえば、組込みシステムのデバイスドライバー開発において、ハードウェアレジスタの仕様書を入力すると、生成AIがそのレジスタにアクセスするための初期コードを生成します。エンジニアはそのコードをレビューし、プロジェクト固有の要件に合わせて調整します。このプロセスにより、従来は数日かかっていた作業が数時間に短縮されます。

第三のパラダイムは、統合AI活用型開発、すなわちエンド・ツー・エンド学習による開発です。このパラダイムは、ヒューマノイドロボットや完全自動運転といった次世代技術を現実化する、最も革新的なアプローチです。このアプローチでは、テキスト、画像、音声、センサーデータなど多岐にわたるデータを統合的に処理する帰納的データ駆動アプローチへの根本的な転換が求められます。

これは、事前にすべてのルールを定義するのではなく、大量のデータからパターンを学習し、未知の状況にも対応できるシステムを構築する手法です。AI部分の開発では、強化学習や模倣学習といった手法が用いられ、モデルの性能はデータの質と量に大きく依存します。また、エッジデバイス上でのAIモデルの軽量化(例:モデル量子化、枝刈り)や、推論の最適化(例:TensorRT)が重要な技術課題となります。

たとえば、ヒューマノイドロボットが人間と同じ環境で歩行するためには、平坦な床だけでなく、階段、傾斜、不整地など、無数のバリエーションに対応する必要があります。これらすべての状況に対してルールを定義することは事実上不可能です。代わりに、AIは大量の歩行データから学習し、新しい環境でもバランスを保って歩行する方法を発見します。

AIによる制御革命:ルールベースからエンド・ツー・エンドへ

AI技術の進化は、制御のロジック、すなわち制御方法を根本的に変革し、「ルールベースからエンド・ツー・エンドへ」という新たな潮流を生み出しています。従来のシステムは、人間があらかじめ定めたルールベースのロジックで制御されていました。しかし、AI、特に生成AIの進化により、大量のデータから適切な制御を学習するエンド・ツー・エンドの制御が、特定の適用領域において実用段階に入りつつあります。

これにより、複雑な状況でも人間には予測できないような、より高精度で効率的な制御が、一部の領域で実現し始めています。ただし、すべてのシステムがエンド・ツー・エンド制御に移行するわけではありません。安全クリティカルな基本制御や、説明責任が厳格に求められる領域では、今後もルールベース制御が主流であり続けるでしょう。この革新の真の意味は、従来のルールベースでは制御が困難だった、複雑かつ大規模なシステムの一部に、新たな制御手法の選択肢が加わったという点にあります。

その結果、これまで実現不可能だった自律的なロボットやインテリジェントな製造ラインなど、新たな価値を創造する市場が立ち上がりつつあり、組込みソフトウェア市場全体の拡大を牽引しています。ただし、この市場拡大は一様ではなく、適用領域によって進展速度が大きく異なることを認識する必要があります。

従来のルールベース制御では、エンジニアが状況を分析し、「もしAならばBをする」というルールを定義します。たとえば、自動車のクルーズコントロールでは、「前方車両との距離が一定値以下になつたら減速する」というルールを定義します。このアプローチは、状況が限定されている場合には有効ですが、状況の組み合わせが爆発的に増えると、すべてのケースをカバーするルールを定義することが困難になります。

エンド・ツー・エンドでは、センサーの生データ(例:カメラ画像、LiDARデータ)を入力として、制御出力(例:ステアリング角、加速度)を直接出力するニューラルネットワークを学習させます。このネットワークは、人間が明示的にルールを定義することなく、大量の運転データから、安全で快適な運転を実現する制御を学習します。

この学習ベースのアプローチの利点は、人間が気づかないような微妙なパターンも学習できることです。たとえば、熟練ドライバーは無意識のうちに、路面の状態、天候、周囲の交通状況などを総合的に判断して運転しています。エンド・ツー・エンド学習は、このような暗黙知を、大量のデータから自動的に抽出できます。

2-2 SDxとAIの相互作用

SDx時代の開発と生成AIの必然性 — 開発プロセスの革新

SDxの根幹にあるのは、「継続的な価値提供」という思想です。

製品開発はもはや“一度きりの完結型”ではなく、顧客や市場からのフィードバックを短いサイクルで反映し、継続的に改良・更新していくことが前提となります。こうした“高速かつ持続的な開発”は、従来の人手依存型開発では限界に達しています。ここで登場するのが、生成AIという新たな技術的推進力です。

生成AIは、SDxが要求するスピードと品質を両立させる鍵となります。

コード生成やテストケースの網羅化、ドキュメント自動化など、反復的なタスクを自動化することで、エンジニアは創造的な問題解決やアーキテクチャ設計に専念できるようになります。つまり、SDxという「目的地」に向かうために、生成AIは不可欠な「推進エンジン」として機能するのです。

製品機能を変革するAI技術 — インテリジェントな進化サイクル

AIの影響は、開発工程の自動化にとどまりません。

ディープラーニングやエンド・ツー・エンド学習の発展により、従来のルールベース制御では到達できなかった複雑で高度な挙動を実現します。

ヒューマノイドロボットの自律動作、完全自動運転、複雑な組立工程を柔軟にこなす産業用ロボットなどはその象徴的な例です。これらのシステムは膨大なデータを学習しながら環境適応性を高め、AIモデルの継続的更新によって常に進化を続けます。

この構造は、SDxアーキテクチャと密接に連動しています。

AIモデルは継続的な学習と改善を前提としており、実運用で得られたデータをフィードバックし、再学習によってモデルを更新し、OTA(Over-The-Air)を通じて製品に再配信するというサイクルが不可欠です。

AIは製品価値を創出し、SDxはその価値を持続的に提供する仕組みを支える——両者はこのように、補完的な役割分担のもとで共進化しているのです。

SDx × AI — 二重の革新軸がもたらす構造転換

以上のように、SDxとAIは独立した技術トレンドではありません。

SDxはAIを活かす開発基盤として、AIはSDxの価値を最大化する知能として機能します。

両者は「開発プロセスの革新」と「製品機能の革新」という二重の相互依存関係を形成し、製造業の価値創造構造を根底から再定義しています。

この相互作用によって、製品はハードウェア中心の静的な存在から、ソフトウェアとデータによって進化し続ける“生きたシステム”へと変貌します。

SDxとAIの融合は、開発と運用、提供と利用の境界を曖昧にし、かつてないスピードと柔軟性で市場変化に応答する、新たな製造業の姿を形づくろうとしているのです。

2-3 AIエコシステムにおける「ラストワンマイル戦略」の勝機

AI時代の「ラストワンマイル」と4種の主要プレイヤー

AI駆動型産業エコシステムを理解するためには、まず「ラストワンマイル」という概念を正確に定義し、なぜこの困難な領域に多くの企業が挑戦するのかを理解する必要があります。

ラストワンマイルとは何か、そしてなぜ重要なのか

ラストワンマイルとは、汎用的なAI基盤技術を、人命や社会インフラを支えるリアルな現場に実装・統合する、最も困難なプロセスを指します。具体的には、クラウド上で開発・訓練された強力なAIモデルを、製造現場で働く産業用ロボット、病院で使用される医療機器、物流施設で稼働する自律移動システムといった、失敗が許されない環境のエッジデバイス上で動作するように最適化・実装することを意味します。このプロセスでは、AIの知能とハードウェアの身体を、安全性・信頼性・リアルタイム性といった組込み技術固有の厳格な要求の下で統合する必要があります。

この作業は極めて困難です。クラウド上で優れた性能を示すAIモデルも、ミリ秒単位の応答が求められる産業機器、通信が途絶する可能性がある遠隔地、あるいは人間のすぐそばで動作する協働ロボットといった環境では、全く異なる挑戦に直面します。データセンターでの理想的な条件と、泥や油にまみれた工場の床、振動する車両の中、温度や湿度が変化する屋外環境とでは、要求される技術水準が根本的に異なるのです。

では、なぜ多くの企業がこの困難な領域に挑戦するのでしょうか。理由は三つあります。

第一に、市場規模の爆発的拡大です。製造業、物流、医療、建設、農業といった実世界の産業でAIが本格的に活用されれば、その経済インパクトは計り知れません。様々な調査機関がフィジカルAI関連市場の急速な成長を予測しており、クラウド上で動作する情報処理AIの市場と比較しても、物理世界で実際に動き、価値を生み出すフィジカルAIの市場は極めて大きな成長ポテンシャルを持っています。

第二に、社会的課題の解決という使命です。日本を含む先進国は、深刻な労働力不足に直面しています。製造業では熟練工の高齢化と後継者不足、物流業界ではドライバー不足、医療現場では看護師不足が深刻化しています。AIとロボティクスの統合によってこれらの課題を解決できれば、社会全体に計り知れない価値を提供できます。これは単なるビジネスチャンスではなく、社会的責任でもあるのです。

第三に、日本企業にとって既存の強みを活かせる領域だという点です。日本の製造業は、五十年以上にわたり、品質、安全性、信頼性を追求してきました。ミッションクリティカルな環境でのシステム統合、現場との綿密な擦り合わせ、長期にわたる保守サポート。これらは、まさにラストワンマイルで求められる能力です。クラウドAIの基盤技術開発では米中の巨大資本に勝てなくても、それを安全に実世界に実装する技術では、日本企業が競争優位を持つ可能性があります。

この困難だが巨大な機会を実現するため、AI駆動型産業のエコシステムが形成されています。このエコシステムには、それぞれ異なる役割と責任範囲を持つ、主に四種類のプレイヤーが存在します。ただし、ラストワンマイル、つまり最終製品への実装と社会実装を直接担うのは、これから説明するプレイヤー3と垂直統合例外のTeslaです。プレイヤー1とプレイヤー2は、そのラストワンマイルを実現するために不可欠な技術基盤や高度なAIを提供する、いわば支援的な役割を担っ

ています。これから、各プレイヤーを、彼らが持つ役割、戦略、そして最も重要な責任範囲という視点から見ていきましょう。

プレイヤー1:技術基盤プロバイダー(Technology Foundation Providers)

技術基盤プロバイダーは、AIの開発・実行に必要な汎用的なツール、素材、そしてハードウェアを提供する水平分業のプレイヤーです。このカテゴリーには、NVIDIAのGPUやIsaac、Driveといったプラットフォーム、GoogleのCloudサービスやTensorFlow、OpenAIのAPI、MicrosoftのAzure、そしてエッジAI向けの専用プロセッサなどが含まれます。

彼らのビジネスモデルの核心は水平分業にあります。NVIDIAのGPUを例に取れば、同じ製品が様々な産業で使われています。製造業の検査システム、物流の自律移動ロボット、医療の画像診断システム、さらにはゲーム業界まで。一つの技術基盤を多様な産業に展開することで、研究開発への巨額投資を回収し、莫大な利益を上げることができます。特定の産業に深く入り込むのではなく、全産業を横断して広く技術を提供する、これが彼らの戦略です。

プレイヤー2:ドメインAIプラットフォーマー(Domain AI Platformers)

ドメインAIプラットフォーマーは、特定の高度なタスク領域に特化し、そのタスクを実行するAI、つまり脳そのものをプラットフォームとして開発・供給します。このカテゴリーには、自動運転分野のWayve Technologies、汎用ロボット操作のPhysical Intelligenceといった企業が含まれます。

前述の定義に基づけば、プレイヤー2は高度なAIという「脳」を開発・提供する重要な役割を担いますが、それ自体は最終製品への統合と社会実装という意味でのラストワンマイルの手前の段階です。彼らが開発したAI技術は、プレイヤー3によって最終製品に統合され、初めて顧客の手に届きます。

彼らは、AI時代の新しい専門家集団です。重要なのは、彼らの製品が物理的な製品ではなく、AIモデル群そのものだという点です。Physical Intelligenceを例に取れば、彼らは特定のロボットを製造するのではなく、汎用的なロボット操作を可能にするAI基盤モデルπ0を開発しています。このモデルは、様々な種類のロボットに適用でき、物体の把持、移動、操作といった複雑なタスクを実行できます。彼らの価値は、膨大な学習データと、それを使った継続的な改善サイクルにあります。

彼らの戦略の中核は、プレイヤー1の技術基盤を最大限に活用しながら、特定ドメインにおける膨大なデータ収集、学習、シミュレーションのサイクル自体をプラットフォームとして構築・保有することにあります。このデータと学習サイクルこそが、彼らの競争力の源泉です。新規参入者が同じレベルのAIを開発しようとしても、この蓄積された経験がなければ困難なのです。

彼らはこの高度なAIの脳を、プレイヤー3に、製品を構成する主要コンポーネントとして供給します。

プレイヤー3:AIプロダクトメーカー(AI Product Manufacturers)

AIプロダクトメーカーは、AIつまり脳と、最終製品つまり身体を統合し、実際に顧客に届ける製品として完成させる企業群です。このカテゴリーには、大手自動車メーカー、産業機器や医療機器のグローバルリーダー、ヒューマノイドロボットのスタートアップ、サービスロボットメーカー、特定の中小医療機器メーカー、専門的な産業機械メーカーなど、極めて多様な企業が含まれます。

プレイヤー3の本質的な役割は、高度なAI技術を自社製品に統合し、最終的に顧客の手に届く形で実装することです。大手企業の中には、独自の統合プラットフォームを構築する企業もあれば、スタートアップのようにプレイヤー1やプレイヤー2の技術基盤を直接活用する企業もあります。しかし規模や戦略の違いにかかわらず、共通しているのは、最終製品が引き起こすあらゆる問題に対して社会的・法的責任を負うという点です。

だからこそ、プレイヤー3はAIを単純に買って載せるのではなく、自社の製品の中で厳格に管理し、監視し、異常時には安全システムが確実に作動するように設計する必要があります。AIが異常な判断をした場合に備えて、フェイルセーフ機構が即座に介入し、システムを安全に停止させる。この全体設計を担うのがプレイヤー3の役割です。

プレイヤー3の競争力の源泉は、特定のドメインに対する深い理解にあります。製造現場の検査ロボット、病院の薬剤搬送システム、自動車の安全制御。それぞれのドメインでは、求められる精度、安全基準、法規制が全く異なります。製造業と長年協業してきた検査システムメーカーは、製造現場の微妙な要求を知り尽くしています。どの程度の精度で不良品を検出すべきか、誤検出と見逃しのバランスをどう取るか、ラインのスピードに合わせてどう最適化するか。この現場知識は、グローバルなAI企業には獲得できないものです。

医療機器メーカーも同様です。日本の医療現場特有の運用方法、厚生労働省の認可プロセス、医師や看護師との協働のノウハウ、医療安全に関する深い理解。これらは何十年もかけて築いた独自の資産であり、たとえ同じAI基盤を使っていても、他社が短期間で真似できるものではありません。

プレイヤー3は、最終製品への統合と社会実装という、眞のラストワンマイルに直接向き合っています。そして、製品が引き起こすあらゆる問題に対して、社会的・法的責任を負います。この責任の重さこそが、彼らの存在意義であり、同時に日本企業が強みを発揮できる領域でもあるのです。

プレイヤー4: 垂直統合例外(Vertical Integration Exception)

このプレイヤー構造には重要な垂直統合例外が存在します。それがTeslaです。

Teslaは、プレイヤー1、プレイヤー2、プレイヤー3の役割を、AI学習基盤DojoやFSDチップから自動運転AI、車両OS、OTA、顧客データ収集、充電網に至るまで、ほぼ全て自社で垂直統合しています。

このモデルは、ハードウェアとソフトウェアの完璧な統合と圧倒的な開発スピードを実現し、他社が真似できない体験を創出できます。しかし同時に、全てのレイヤーで最高の技術を維持し続けるための莫大な投資とリスクを伴います。これはほとんどの企業にとって現実的な選択肢ではありません。しかし重要なのは、プレイヤー3の大手企業が目指しているのも、方法は違えどハードウェアとソフトウェアを自社の設計思想の下で深く統合し、顧客体験をコントロールするというゴールはTeslaと共通している、という点です。

JASA会員(ベンダー)の戦略的ポジション: プレイヤー3を支える「成功請負人」

では、本レポートの読者であるJASA会員、つまりシステムインテグレーターやソフトウェアベンダーは、この構造のどこに位置するのでしょうか。

JASA会員、つまりベンダーは、特定のプレイヤーに属するのではありません。エコシステム全体を横断的に支援し、技術と現場を結びつける成功請負人なのです。ベンダーは、主にプレイヤー3を顧客として支援し、プレイヤー3がプレイヤー1やプレイヤー2の技術を活用して最終製品を完成させるプロセスを支援します。

具体的には、プレイヤー1の技術基盤やプレイヤー2の高度なAIを、プレイヤー3の製品に統合する作業を支援します。これには、AIと従来型制御システムとの安全な統合、プラットフォーム上で動作するアプリケーションの開発、暗黙知のデータ化、AIの最適化、そして厳格な検証といった、ラストワンマイルを実現するための技術的課題の解決が含まれます。

独自の統合プラットフォームを構築する大手企業に対しては、そのプラットフォーム設計やアーキテクチャ構築、さらにはサプライヤーのアプリケーション統合を支援します。既存の技術基盤を活用するスタートアップや中小企業に対しては、限られた資源の中で効率的にAIを製品に組み込み、高い品質を実現するための実践的な支援を提供します。

戦略実行上の課題

ただし、この戦略を実行する上では、いくつかの重大な課題が存在します。

変革への覚悟と実行として、特にプレイヤー3の大手企業にとって、ハードウェア中心の組織文化をソフトウェアネイティブへと変革する覚悟が問われます。これは単なる資本力や技術力の問題ではなく、失敗を許容するアジャイルな文化、減点主義に代わる人事評価、トップクラスのソフトウェア人材を惹きつける魅力といった、組織の根本的な変革を実行できるかにかかっています。

暗黙知のデータ化の壁として、プレイヤー3が競争力を維持するには、現場の深いドメイン知識が不可欠です。しかし、AIはドメイン知識そのものではなく、データしか学習できません。ベテラン作業員の阿吽の呼吸といった暗黙知を、AIが学習可能な形式知、つまり高品質なデータセットに変換するプロセスは地道でコストがかかり、ここの実行力が企業の明暗を分けます。

ベンダー自身の人材の崖として、成功請負人であり続けるべきJASA会員にも、深刻な人材の崖があります。従来の仕様書に基づき高品質な組込み開発を行うスキルと、プラットフォーム上で動作するアプリケーション開発スキル、あるいはAI基盤を使いこなすシステム・アーキテクトのスキルは全く異なります。この巨大なスキルの崖を乗り越えるための大規模なリ・スキリングが、喫緊の課題です。

なぜ米中の巨大企業が「ラストワンマイル」に手を出せないのか

米中の巨大企業、つまりプレイヤー1にとって、ラストワンマイルは経済合理性の観点から参入が困難な領域です。この構造的な非対称性こそが、日本企業にとっての希望となります。

まず、個別対応の非効率性があります。一つ一つの工場や製品への対応は、スケールメリットを追求する巨大企業にとって経済的に非効率です。彼らの事業戦略と相容れません。

次に、現場知識の習得困難性があります。製造現場の暗黙知や業界特有の商慣習は、現地に根ざした企業でないと理解困難です。

そして、長期関係構築の困難性があります。十年から二十年の製品ライフサイクルでの継続サポートは、短期的な収益最大化を目指す巨大企業には適さないビジネスモデルです。

したがって、日本企業は基盤モデル開発という莫大な投資が必要な領域で米中と競争するのではなく、既存のグローバルAI基盤を安全に社会へ実装する翻訳層として価値を発揮できます。これこそが、現実的かつ戦略的なラストワンマイル戦略なのです。

スマートフォン時代との決定的な違い

ここで、「なぜスマートフォン時代ではラストワンマイルが価値にならなかったのか？」という疑問に答える必要があります。かつて日本のメーカーは、Androidというプラットフォームに主導権を奪われ、苦戦しました。なぜ今回は同じことが起きないのでしょうか。

スマートフォンにおける「ラストワンマイル」の短さ

スマートフォンのOSという点だけを取ってみれば、それは所詮デジタルデバイスです。Androidというプラットフォームが提供する標準機能の上で、メーカーがカスタマイズできる部分は極めて限定的でした。GUIのデザイン、プリインストールするアプリの選択、多少のシステム設定の調整。メーカーが独自に手を加えられる領域は、実質的にこの程度しかありませんでした。

つまり、スマートフォンにおける「ラストワンマイル」は、距離がほとんどないのです。Googleが提供するAndroid OSという完成されたプラットフォームと、最終製品との間に、メーカーが深いドメイン知識を活かして付加価値を生み出せる余地が、構造的にほとんど存在しませんでした。

さらに決定的なのは、製品の機能拡張がアプリという形で完全にメーカーの手を離れてしまったことです。カメラの性能を向上させたい、新しいコミュニケーション機能を追加したい、ゲーム体験を豊かにしたい。こうした機能はすべて、Google Playストアを通じてサードパーティの開発者が提供するアプリによって実現されます。メーカーは、ハードウェアの筐体とディスプレイを提供する以外、製品の進化に関与できなくなってしまいました。

この構造では、メーカー間の差別化は困難です。「同じAndroid OS、同じアプリストア、ならば安い方がいい」という価格競争に巻き込まれるのは必然でした。

製造業製品における「ラストワンマイル」の長さと深さ

一方、製造業の製品、つまり産業機器、自動車、医療機器、ロボットにおいては、状況がまったく異なります。

第一に、ラストワンマイルが極めて長いのです。たとえNVIDIAのIsaacやGoogleのTensorFlowという強力なAI基盤があっても、それを実際の工場の生産ライン、病院の手術室、走行する自動車の中で、安全に、確実に、リアルタイムで動かすためには、膨大な統合作業が必要です。

センサーからのデータ取得、ノイズ除去、前処理。エッジでのリアルタイムAI推論。機械制御への指令変換。異常検知とフェイルセーフ。これら一つ一つが、その産業特有の物理法則、安全規格、運用フローと深く結びついています。トヨタの工場で培われた生産管理の思想、病院での医療安全の多重チェック体制、自動車の車両動特性と路面状況の複雑な相互作用。これらのドメイン知識がなければ、どんなに優れたAI基盤があっても、現場では使い物になりません。

第二に、製品の中核機能がアプリという形で外部化されることはありません。産業用ロボットの制御アルゴリズム、自動車のパワートレイン制御、医療機器の診断ロジック。これらはすべて、メーカーが自社で設計し、実装し、長期にわたって保守する責任を負う領域です。「アプリストアから

機能を追加」という構造にはなりません。なぜなら、これらの機能は製品の安全性と直結しており、メーカーが最終責任を負わなければならないからです。

つまり、製造業の製品におけるラストワンマイルは、スマートフォンとは比較にならないほど長く、深く、専門的なものです。ここに、メーカーが何十年もかけて蓄積してきたドメイン知識を活かし、他社には真似できない価値を提供できる広大な領域が存在します。

構造的な非対称性が生む競争優位

この構造的な違いこそが、スマートフォン時代とは異なる競争環境を生み出します。GoogleやNVIDIAのような巨大企業は、強力な汎用プラットフォームを提供できます。しかし、その先の長いラストワンマイル、つまり日本の工場、日本の病院、日本の道路という具体的な現場で、何十年も安全に動き続けるシステムを作り上げることは、現地に根ざした企業でなければ困難です。

スマートフォンでは「ラストワンマイルが短すぎた」ために日本企業の強みが活きませんでした。しかし製造業では「ラストワンマイルが長く深い」からこそ、現場を知り尽くした日本企業の力が不可欠なのです。この翻訳層、つまりラストワンマイルこそが、日本企業が勝てる戦場なのです。

2-4 生成AI民主化と組込み業界への影響

GPU依存からの脱却：民主化への道筋

現在のAI技術は、NVIDIAのGPUに大きく依存する構造となっています。大規模言語モデルの学習には数万枚のGPUが必要であり、推論においても高性能GPUなしには実用的な応答速度を得ることができません。この状況は、AI技術の利用を資本力のある大企業に限定し、中小企業や組込み業界にとって大きな参入障壁となっていました。

しかし、複数の技術革新により、この構造が根本的に変わりつつあります。生成AI民主化の波は、組込み業界に新たな事業機会をもたらす可能性を秘めています。

DeepSeekによる技術革新：効率性革命の始まり

2024年末から2025年初頭にかけて、中国のDeepSeekが発表したR1モデルは、AI業界に衝撃を与えるました。このモデルは、OpenAIのGPT-3と同等の性能を、はるかに少ない計算資源で実現したのです。従来、高性能なAIモデルを構築するには膨大な計算資源が必要であるという前提が覆され、「大規模＝高性能」という常識が疑問視されることになりました。

DeepSeekの革新は、技術的な効率化が可能であることを実証しました。これは単なる一企業の成功事例ではなく、AI技術開発における新しいパラダイムの始まりを示しています。計算資源の効率的な利用により、GPU依存度を劇的に下げることが可能であることが明確になったのです。

SLMの台頭：小型高性能モデルの実用化

大規模言語モデルと並行して、SLM(Small Language Model)と呼ばれる小型モデルの開発が急速に進んでいます。MicrosoftのPhi-3シリーズやGoogleのGemini Nanoなど、数億から数十億パラメータ規模の小型モデルが、特定タスクにおいては大規模モデルに匹敵する性能を発揮することが実証されています。

これらのSLMは、スマートフォンや組込みデバイス上で直接動作することができます。クラウドへの通信を必要とせず、デバイス単体でAI処理を完結できることは、リアルタイム性が求められる組込みシステムにとって革命的な変化です。工場の生産ラインにおける品質検査、医療機器での診断支援、自動車の運転支援システムなど、遅延が許されない用途において、エッジデバイス上でのAI処理が現実的な選択肢となっていました。

技術的ブレークスルー：モデルの小型化と効率化

SLMの実用化を支えているのは、複数の技術的ブレークスルーです。これらの技術革新により、従来はデータセンターでしか動作しなかったAIモデルが、組込みデバイスで動作可能になってきています。

量子化技術の進化により、モデルの表現精度を下げることで、サイズを大幅に圧縮できるようになりました。従来の32ビット浮動小数点から16ビット、さらには8ビットや4ビット整数へと量子化することで、モデルサイズを4分の1から8分の1に削減しながら、性能をほぼ維持することが可能になっています。これにより、メモリ容量が限られた組込みデバイスでも、実用的なAIモデルを搭載できるようになりました。

知識蒸留技術の発展も重要です。大規模な教師モデルの知識を小規模な生徒モデルに転移することで、巨大モデルの能力を小型化して組込み機器に搭載できるようになってきています。教師モデルが学習した複雑なパターンを、より効率的な形で生徒モデルに継承させることで、サイズと性能のバランスを最適化することが可能になります。

効率的なアーキテクチャの開発も進んでいます。Mixture of Experts (MoE) のような、必要な部分だけを動的に活性化するアーキテクチャにより、見かけ上は大規模でも実行時の計算量を大幅に削減できます。全てのパラメータを常時使用するのではなく、入力に応じて最適な専門家モジュールのみを選択的に活用することで、計算効率が劇的に向上します。

GPU以外の選択肢：専用チップの台頭

AI処理に特化した専用チップの登場により、NVIDIA GPUへの依存度が下がりつつあります。Google TPU、Apple Neural Engine、QualcommのAI Engineなど、各社が独自のAI専用プロセッサを開発しています。これらのチップは、特にエッジAI向けには、消費電力とコストの面で優位な選択肢となっています。

汎用的なGPUと異なり、これらの専用チップは特定のAI処理に最適化されているため、同じ性能をより少ない電力で実現できます。組込みシステムにおいては、消費電力の制約が厳しいため、この効率性は決定的な優位性となります。バッテリー駆動の機器や、冷却能力が限られた環境での使用において、専用チップは実用的な選択肢として急速に普及しつつあります。

オープンソースモデルの普及：プラットフォーム依存からの解放

MetaのLlamaシリーズや、Mistralのオープンモデルなど、商用利用可能なオープンソースの基盤モデルが次々と公開されています。これらのモデルにより、企業が自社で学習やカスタマイズを行える基盤が整ってきました。クラウドプラットフォーマーのAPIに依存することなく、自社のニーズに合わせたモデルを構築できる環境が整いつつあります。

オープンソースモデルの普及は、技術の透明性を高めることにもつながります。モデルの内部構造や学習プロセスが公開されることで、安全性の検証や、特定用途への最適化が容易になります。

す。医療機器や自動車など、高い安全性が要求される分野において、ブラックボックスではないAIシステムを構築できることは、大きな意味を持ちます。

三段階シナリオ：民主化の進展と組込み業界の機会

生成AI民主化は、段階的に進展していくと予想されます。各段階において、組込み業界が取るべき戦略と得られる機会が異なります。

短期(現在から2年程度)においては、クラウドAPI依存の状況が継続します。大規模な統合AIモデルは依然として巨大資本による基盤開発が中心であり、組込み企業はこれらのプラットフォームを活用する立場が続きます。この段階では、クラウドとエッジのハイブリッド構造を最適化し、通信遅延やセキュリティの課題に対処しながら、AIの恩恵を製品に取り込むことが重要です。

中期(2年から5年後)においては、SLMによるエッジAI実用化が本格化します。組込み機器が自律的にAI処理を実行できるようになり、クラウドへの依存度が大幅に低下します。この段階では、特定用途に最適化された小型モデルの開発が競争優位の源泉となります。医療機器向け、製造装置向け、自動車向けといった産業特化型のSLMを開発し、エッジデバイス上で高速に動作させる技術が重要になります。

この段階における市場機会は劇的に拡大します。エッジAIデバイス市場は大幅に成長すると予想されており、組込み企業にとって新たな成長領域となります。巨大プラットフォームからニッチ特化型への転換が始まり、特定のニーズに最適化されたAIにより、ビッグテックのGPU支配からの脱却が可能になります。

長期(5年から10年後)においては、統合AI民主化への希望的シナリオが展開される可能性があります。LLMでの成功パターンが他の分野にも展開され、統合AI基盤モデルもドメイン・製品特化型として小型化・民主化される可能性があります。産業特化統合モデル(自動車専用、医療専用、製造業専用の統合AI)、機能限定統合モデル(画像生成特化、音声処理特化の小型統合AI)、エッジ統合AI(リアルタイム処理に最適化された統合AI)という展開により、真の「エッジファースト」時代が到来するのです。

この段階では、ドメイン特化統合AIの小型化が実現し、製品組込み可能な統合AIプラットフォームが登場します。組込み業界企業は民主化時代のリーダーシップを確立し、技術主導者としての地位を確固たるものにできます。

組込み業界への戦略的インプリケーション

この民主化の波は、組込み業界にとって技術受容者から技術主導者への転換の機会を意味します。各段階で最適なポジションを取る柔軟性が重要であり、短期では現実的な技術活用(API利用、応用開発)に注力しながら、中期では民主化への準備投資(技術調査、共同研究、人材育成)を進め、長期では民主化時代のリーダーシップ確立(標準化主導、エコシステム構築)を目指す戦略的アプローチが求められます。

重要なことは、この変化を受動的に待つのではなく、能動的に準備を進めることです。SLMの技術動向を継続的に追跡し、自社製品への適用可能性を評価することが必要です。エッジAI向けの専用チップやツールチェーンの評価を進め、将来の製品開発に向けた技術基盤を整えることが重要です。オープンソースコミュニティへの参加を通じて、最新の技術トレンドにアクセスし、自社の知見を蓄積することも有効です。

生成AI民主化は、組込み業界にとって大きな機会の窓を開きます。GPU依存からの脱却により、資本力に依存しない競争が可能になり、日本企業が培ってきた現場知識や品質文化を活かした差別化が実現できるようになります。この機会を最大限に活用するために、今から準備を始めることが求められています。

2-5 3つのパラダイムの統合戦略と未来への道筋

パラダイムの共存と最適組み合わせ

重要なことは、これらの3つのパラダイムが独立したものではなく、相互に補完し合う関係にあることです。最適なシステムでは、それぞれのパラダイムを適材適所で組み合わせることが重要です。

ルールベースによる開発は信頼性と効率性の基盤として、今後も安全クリティカルな部分での継続的重要性を保ちます。他のパラダイムとの組み合わせによる安定性確保に貢献し、特に医療機器、航空宇宙、原子力制御など、最高レベルの安全性が求められる分野では必須の技術です。

AI支援型開発は開発プロセスの革新により、従来型開発の効率性を劇的に向上させます。エンド・ツー・エンドの基盤技術として機能し、人間の創造性とAIの処理能力を最適に組み合わせます。

エンド・ツー・エンドは制御技術の革命として、適用可能な領域において全く新しい応用領域と市場の創造を可能にし、学習・進化機能により継続的な価値向上を実現します。

ハイブリッドアプローチの実践と限界の認識

特に安全クリティカルなシステムでは、従来型の信頼性とAI技術の柔軟性を融合させたハイブリッドアプローチが有効です。安全クリティカル部分ではルールベースによる確実性を重視し、基本制御ロジック、フェイルセーフ機能、緊急停止システムなどを担当します。非クリティカル部分ではAI支援型開発による効率性を重視し、ユーザーインターフェース、データ分析・最適化、予測・学習機能などを担当します。そして統合部分ではエンドツーエンドによる革新性を重視し、全体最適化、適応制御、予測保全などを担当する構造が、現段階での現実的なアプローチとなります。

ただし、このハイブリッドアプローチにも限界があることを認識しておく必要があります。エンドツーエンドAIの真の価値は、システム全体を統合的に学習・最適化することで、人間が設計したルールを超える性能を発揮することにあります。ハイブリッドアプローチは、従来のシステムからの移行期における実用的な解ではありますが、特定の領域では、より純粋なエンドツーエンド型のシステムへと進化していく可能性があります。

そのとき、日本企業が培ってきた安全性や品質への取り組みは、形を変えて活かされる必要があります。従来のような事前検証型ではなく、継続的な監視と学習による適応型の安全確保、説明可能性を高めるためのAIアーキテクチャの工夫、予期せぬ事態への対応プロトコルの確立など、新しいパラダイムに適した安全設計の手法を開発していくことが求められます。

この変革期において、日本企業が取るべき戦略は、世界が作り上げた強力な基盤モデルを最大限に活用し、自社の強みである「現場の知恵」を新しいパラダイムに適応させながら「ラストワン

マイル」を担うことです。同時に、中国を含む先進的な取り組みから謙虚に学び、グローバルな技術進展を注視しながら、日本独自の価値を再定義していく姿勢が不可欠です。新しい技術の可能性を積極的に探求し続けることが、次世代の組込みシステムを創造する鍵となります。

フィジカルAIの時代が幕を開けた今、私たちは単なる技術者ではなく、物理世界とデジタル世界を融合させる新時代の創造者として、人類の未来を切り拓いていく使命を担っています。ただし、その道のりは、従来の成功体験の延長線上にあるのではなく、複数のパラダイムを適材適所で選択し、組み合わせるという新しいアプローチを通じて切り開かれるものであることを、常に心に留めておく必要があります。

第2章 参考文献一覧(APA形式)

MIT Computer Science and Artificial Intelligence Laboratory (CSAIL). (2023). End-to-End Learning for Robotics and Autonomous Systems. Cambridge, MA: MIT Press.

NVIDIA Corporation. (2024). NVIDIA Isaac Robotics Platform Technical Overview. Retrieved from <https://developer.nvidia.com/isaac>

Tesla, Inc. (2024). Dojo Supercomputer and Full Self-Driving (FSD) Architecture Whitepaper. Palo Alto, CA.

Google Research. (2024). Efficient LLMs and Edge Inference: Toward Scalable On-device AI. Mountain View, CA.

Microsoft Research. (2024). Phi-3 Technical Report: Small Language Models for Edge Intelligence. Redmond, WA.

DeepSeek. (2025). R1 Model Efficiency Benchmark Report. Beijing: DeepSeek AI Lab.

IDC Japan. (2024). 国内組込みAI市場予測2024–2028年：デバイスエッジにおけるAI推論の拡大. Tokyo: IDC Japan.

McKinsey & Company. (2025). Generative AI and Edge Computing: Global Industry Outlook 2025. New York, NY.

Gartner. (2024). Emerging Tech: Edge AI 2025 Outlook. Stamford, CT: Gartner Inc.

経済産業省. (2025). デジタル経済レポート2025：データに飲み込まれる世界、聖域なきデジタル市場の生存戦略. 東京: 経済産業省.

独立行政法人 情報処理推進機構 (IPA). (2024). AIシステムにおける安全性・信頼性確保の指針 Ver.3. 東京: IPA.

一般社団法人 組込みシステム技術協会 (JASA). (2023). 組込みAI開発の現状と課題レポート. 東京: JASA.

経済産業省. (2024). 製造業DXレポート2024：現場力とAIによる生産革新. 東京: 経済産業省.

PwC. (2025). Global AI Investment Outlook 2025: Return on Intelligence. London: PwC Global.

OECD. (2024). AI Policy Observatory 2024 Update: Global Governance of AI and Industrial Policy. Paris: OECD Publishing.

中国国家工業情報化部 (MIIT). (2023). 中国製造2025:AIとロボティクスの国家ロードマップ. 北京: MIIT.

IEC. (2023). IEC 62304: Medical Device Software — Software Life Cycle Processes. Geneva: International Electrotechnical Commission.

ISO. (2023). ISO 26262: Road Vehicles — Functional Safety (2nd ed.). Geneva: International Organization for Standardization.

IEEE Robotics and Automation Society. (2024). Human–Robot Interaction and Safety in AI-driven Systems: Technical Standards Overview. New York: IEEE.

進藤智則. (2025年9月). Physical Intelligenceと組んだ日本スタートアップ 275億円調達 Telexistence、ロボAI実用化見据える. 日経*Robotics*, 2025年9月号, 4–14. 日経BP社.

進藤智則. (2025年6月). AIによるソフト開発エージェントの進展と展望. 日経*Robotics*, 2025年6月号, 30–31. 日経BP社.

進藤智則. (2025年6月). 躍進する中国のロボAI、ヒューマノイドのAgiBotが100万件データ公開、潜在空間で行動生成のVLA. 日経*Robotics*, 2025年6月号, 4–13. 日経BP社.

進藤智則. (2025年5月). オールジャパンでロボAI開発へ、AIロボット協会発足 メカ偏重への危機感後押し、協調領域で基盤モデル. 日経*Robotics*, 2025年5月号, 3–6. 日経BP社.

進藤智則. (2025年5月). グーグルがついにGeminiをロボットAI向けに投入 数千時間の学習データ適用、蒸留して軽量化も. 日経*Robotics*, 2025年5月号, 8–14. 日経BP社.

進藤智則. (2025年1月). グーグル辞めたロボットAIの精銳達が革新的な成果 ロボットAI革命近し、汎用的基盤モデルの入り口へ. 日経*Robotics*, 2025年1月号, 3–19. 日経BP社.

進藤智則. (2024年8月). 1000台のヒューマノイドで学習加速のノルウェー1X OpenAIも出資し家庭向け狙う、スケール化に懸ける. 日経*Robotics*, 2024年8月号, 3–13. 日経BP社.

第3章:ベンダーの新たな道筋と変革

3-1 メーカー単独での内製化の困難とベンダーの必要性

日本のメーカーがSDxへの移行をすべて内製でやり遂げることは、非常に困難であるというのが一般的な見解です。SDxの実現には、クラウド、AI、サイバーセキュリティ、組込みソフトウェアなど、高度なスキルを持つ専門家が多数必要となります。しかし、必要な人材をすべて自社で確保することには、多くの障壁が存在します。

この課題を乗り越えるには、自社の強みであるハードウェアと密接に関わるコア技術を内製しつつ、基盤となるソフトウェアプラットフォームや高度なAIアルゴリズムといった領域は外部の専門家や企業と協業・活用する「内製と協業のハイブリッド戦略」が現実的です。この戦略において、組込みソフトウェア開発ベンダーとの協業は不可欠となります。

3-2 従来のビジネスモデルの限界

しかし、SDxと生成AIの時代においては、単に委託・請負型の支援では、協業パートナーとして機能することはできません。この新しい潮流は、顧客起点での継続的な価値開発と、爆発的な生産性を前提とします。そのため、これまでの「人月による労働力提供」や、与えられた仕様を単に開発するだけの労働集約型のビジネスモデルは、SDx時代に求められる要件に対して長期的には持続不可能となります。

この持続不可能性には、以下の2つの明確なロジックが存在します。ただし、これは即座の破綻を意味するのではなく、段階的な移行が必要な構造的課題であることを認識する必要があります。

継続的な価値開発との非互換性

人月モデルは、あらかじめ定義された固定的なスコープと、それに必要な労働時間を前提としています。人月モデルは、完成した時点で役割を終える「売り切り」の発想であり、常に進化し続ける「サービス」を提供するというSDxの思想と長期的には対立します。SDxでは、ベンダーは単なる労働力の提供者ではなく、製品のライフサイクル全体を通じて共に価値を創出する「パートナー」としての役割が求められます。

爆発的な生産性とのインセンティブの対立

生成AIの普及により、ソフトウェア開発の生産性は飛躍的に向上しています。例えば、以前は数週間かかっていたコード生成やテストスクリプトの作成が、AIの活用によって数時間で完了するようになりました。人月モデルは、労働時間に直接報酬を紐づけるため、このような効率化を推進するインセンティブが働きません。むしろ、生産性を高めて作業を短時間で終えると、ベンダー側の売上が減少してしまうという、本質的なインセンティブの対立が発生します。SDx時代に不可欠な「爆発的な生産性」は、時間をかけた労働を評価する人月モデルを過去のものとします。

日本の製造業がソフトウェア駆動型社会へと移行する中、組込みソフトウェア開発ベンダーに求められるのは、従来の「労働力提供」を価値とする人月ビジネスからの脱却です。これは、単にコスト削減や業務効率化の問題にとどまらず、新たな時代における競争力と生存をかけた根本的

な戦略転換を意味します。ただし、この転換は一夜にして実現するものではなく、移行期には人月モデルと新しいモデルを併用しながら、段階的に進めていく現実的なアプローチが必要です。

この変革期において、ベンダーが取るべき戦略は一つではありません。特に、これまでのソフトウェア開発を委託という形で、開発の労務提供を主軸としてきたSIベンダー(System Integration vendor)にとっては、自社の強みや経営判断によって、大きく二つの道に分かれるでしょう。一つは、「SDxプラットフォームの開発者」として日本の製造業の基盤を支える道。もう一つは、「高度なナレッジ提供者」として、個別企業の課題解決に深くコミットする道です。この章では、この二つのアプローチを組み合わせた、持続可能な共存共栄モデルを提言します。

3-3 共通SDxプラットフォームの構築者という道

SDx時代では、製品価値の源泉が、継続的にアップデートされるソフトウェアプラットフォーム、そしてそのアプリケーションへと移行します。このパラダイムシフトは、個々のベンダー・メーカーが単独で実現することは非常に困難です。プラットフォームの開発と、その継続的な進化には、メカエレに強みを持つメーカーと、組込みソフトウェアに強みを持つベンダガタッグを組んで実現する必要があります。

これまで開発の労務提供を主軸としてきたSIベンダーのうち、特定の技術や業界で深い専門性を培ってきた企業は、自社の強みを活かしてこの道を進むことができます。JASAの各会員企業が、それぞれの専門領域で培ってきた技術や知見を統合し、「安全性」「信頼性」といった日本独自の強みを体現する日本製SDxプラットフォームをOne Teamで開発することを提言します。

なぜOne Teamでプラットフォームを開発するのか

個々の企業が独自にSDxプラットフォームを開発するアプローチには、いくつかの本質的な限界があります。まず、開発コストとリスクの問題です。SDxプラットフォームの開発には、莫大な投資が必要となります。モダンなソフトウェア開発への変換、モノリシックなソフトウェアからサービス指向アーキテクチャのコンポーネントベースのプラットフォームの構築と導入、クラウドネイティブ技術、CI/CDパイプライン、セキュリティ基盤、OTA機能など、多岐にわたる技術要素を統合する必要があります。これを単独の企業が担うには、技術的にも財務的にも大きな負担となります。

次に、人材確保の困難さがあります。SDxプラットフォームの開発には、ソフトウェアアーキテクチャ、リアルタイムシステムや分散システム、クラウド技術、DevOps、サイバーセキュリティ、AI/ML、データ分析など、幅広い専門性を持つ人材が必要です。これらの人材をすべて自社で確保し、維持することは、大企業であっても容易ではありません。

さらに、標準化と相互運用性の欠如という問題もあります。各社のみで独自プラットフォームを開発すると、業界全体で互換性のない複数のプラットフォームが乱立することになります。これは、サプライチェーン全体での効率を低下させ、エコシステムの形成を阻害します。

そして、グローバル競争における不利です。海外では、AWSやAzure、Googleなどのクラウドプラットフォーム、あるいはTeslaやAppleのような企業が、巨額の投資によって高度なプラットフォームを構築しています。日本企業が個別に対抗しようとしても、規模の経済で不利になります。

一方で、最初から全メーカーが一つの共通プラットフォームを合同開発するのは現実的ではありません。最初のフェーズでは、同じ全体アーキテクチャのブループリント(青写真)を共有しつつ、個

別または複数のメーカやベンダのアライアンスでプラットフォーム開発を進め、各社のプラットフォーム技術の基盤を獲得し、ある段階で産業ドメイン単位でのアライアンスによる同一プラットフォームに収束させていく流れを取るのが現実的です。ただし、だからこそ同じブループリントを共有して進めないと、いざ手を組む段において、言葉も通じない、まったく再利用が出来きないとなり、産業としてのロスが大きくなってしまうことは防がねばなりません。また、ブループリントとそれに従った主要技術要素を共有することにより、アライアンスを促進、加速することが、結果的にプラットフォームの共通化の実現をより速やかに引き起こしつつ、各社の異なる特質や状況を合理的な範囲で織り込み整合させることに繋がると考えます。

SDxプラットフォームの戦略的意義

SDxプラットフォームは、単なる技術的な基盤以上の戦略的意義を持ちます。まず、これは日本の製造業の競争力の基盤となります。自動車、医療機器、産業機械など、日本が強みを持つ分野において、高い安全性と信頼性を備えたプラットフォームを提供することで、日本製品の差別化要因となります。

また、サプライチェーン全体の効率化にも寄与します。完成品メーカーからTier1、Tier2のサプライヤーまで、共通のプラットフォームを使用することで、開発の重複を避け、相互運用性を確保し、全体最適を実現できます。例えば、自動車業界において、OEMとサプライヤーが共通のSDxプラットフォームを使用すれば、部品レベルでのソフトウェア統合が容易になり、開発期間の短縮とコスト削減が実現します。

さらに、中小企業の参入障壁を下げる効果もあります。共通プラットフォームがあれば、大企業だけでなく、リソースが限られた中小企業も、高度なSDx機能を活用した製品を開発できるようになります。また、プラットフォームはコンポーネントベースなため、強みがある領域のコンポーネントを提供するというビジネスも可能になります。これにより、業界全体のイノベーションが加速します。

そして、人材育成の効率化も重要な意義です。共通プラットフォームがあれば、教育機関や企業の研修プログラムは、そのプラットフォームに特化したカリキュラムを提供できます。これにより、業界全体で必要なスキルを持つ人材を効率的に育成できます。また、エンジニアが企業間を移動する際も、共通のスキルセットを活用できるため、労働市場の流動性が高まります。

SDxプラットフォームのブループリントとその開発

SDxプラットフォームは、メカ、エレ、そしてそれらを制御するコンピュータの包括的な開発プラットフォームです。その中心となるのは、システムに組込まれるコンピュータのソフトウェアプラットフォームとシステム開発のためのツールチェーンです。ブループリントでは、このシステムをソフトウェア化するためのシステムモデリングのアプローチ、ソフトウェアプラットフォームのリファレンスアーキテクチャ、ツールチェーン構成、そして開発とシステムオペレーションの空間の両方を包含するDevOpsプロセスとなります。

このブループリントは、単に図や文章のみで表現できるものではありません。図と文章だけで、そのブループリントの妥当性や有効性を示すことは出来ませんし、考えるだけで作れる簡単なものでは決してありません。ブループリントの開発をリファレンス実装的なプラットフォームの開発とセットで行う必要があります。

リファレンスSDxプラットフォーム開発の推進体制

リファレンスSDxプラットフォームの開発は、JASAが中心となり、会員企業の協力のもとで推進します。ただし、これは単なる技術開発プロジェクトではなく、業界全体のエコシステムを構築する取り組みです。

まず、ガバナンス構造を明確にする必要があります。技術選定、優先順位の決定、予算配分といった重要な意思決定を行う運営委員会を設置します。この委員会には、参画企業の代表者だけでなく、最終ユーザーであるメーカー、学識経験者なども含めることで、多様な視点を取り入れます。

次に、開発体制の整備です。プラットフォーム全体を、コア基盤、ミドルウェア、ツールチェーン、AI基盤といった機能領域ごとに分割し、それぞれにワーキンググループを設置します。各ワーキンググループは、その領域に専門性を持つ企業が主導し、他の企業がサポートする形で開発を進めます。

重要なのは、オープンイノベーションの原則です。プラットフォームのフレームワークやインターフェース仕様は、積極的にグローバルなオープンスタンダードも活用しつつ、それに不足するものについては独自に定め、オープンに公開し、広く業界全体が利用できるようにします。一方、業界特有の機能や、差別化要因となる部分については、参画企業が独自に開発できる余地を残します。Open/Closed原則に従い、何をいつどこまでオープンにし、クローズにするかを適切に判断します。

また、継続的な改善の仕組みも必要です。プラットフォームは一度作ったら終わりではなく、技術の進化、規制の変更、市場のニーズに応じて継続的に改善していく必要があります。そのため、フィードバックループを確立し、実際の利用状況や課題を収集し、次のバージョンに反映させる体制を整えます。

さらに、知的財産権の扱いも重要な論点です。共同開発された技術の特許権や著作権をどのように管理するか、明確なルールを定める必要があります。一般的なアプローチとしては、フレームワークは共有財産とし、各社が平等に利用できるようにする一方、各社が独自に開発した応用技術については、その企業が権利を保持するという形が考えられます。

業界別のプラットフォームバリエーション

SDxプラットフォームは、すべての業界に対して完全に同一なプラットフォームを提供するのではなく、ブループリントやフレームワーク部分は共通にしつつ、業界ごとの特性に応じたバリエーションを用意します。

自動車業界向けプラットフォームでは、ISO 26262(機能安全)、ISO/SAE 21434(サイバーセキュリティ)、AUTOSAR(車載ソフトウェアアーキテクチャ標準)への準拠が重要です。また、車載ネットワーク(CAN、Ethernet、FlexRay)のサポート、リアルタイム性の保証、車両診断機能(UDS)、V2X通信といった、自動車特有の機能が必要です。さらに、ティア1サプライヤーとOEMの協業を支援するために、部品レベルでのソフトウェア統合を容易にする仕組みも重要です。

医療機器業界向けプラットフォームでは、IEC 62304(医療機器ソフトウェアライフサイクルプロセス)、IEC 60601(医療機器の安全規格)への準拠が必須です。また、FDA(米国食品医薬品局)やPMDA(医薬品医療機器総合機構)といった規制当局への申請を支援する文書生成機能、トレー

セキュリティ管理(要求から実装、テストまでの追跡)、監査証跡の記録といった機能が重要です。医療機器は人命に直結するため、極めて高い安全性と信頼性が要求されます。

産業機械・ロボット業界向けプラットフォームでは、ISO 10218(産業用ロボットの安全)、IEC 61508(機能安全の基本規格)への準拠が重要です。また、PLCとの連携、産業用ネットワーク(EtherCAT、PROFINET、CC-Link)のサポート、予知保全機能、デジタルツイン連携といった機能が求められます。製造現場では、24時間365日の安定稼働が求められるため、高可用性とメンテナンス性も重要な要件です。

これらの業界別バリエーションは、コアプラットフォームの上に構築されるため、コア部分の改善はすべての業界に波及します。一方、業界固有の機能は独立して開発・改善できるため、各業界のニーズに柔軟に対応できます。

エコシステムの形成

SDxプラットフォームの真の価値は、それ自体だけでなく、そのまわりに形成されるエコシステムにあります。プラットフォームを中心に、様々なプレイヤーが価値を提供し、相互に補完し合う関係を構築することが重要です。

まず、プラットフォーム上で動作するアプリケーションやサービスを開発するアプリケーションプロバイダーが重要な役割を果たします。これには、既存のソフトウェアベンダーだけでなく、スタートアップ企業や個人開発者も含まれます。プラットフォームが標準化されることで、一度開発したアプリケーションを、そのプラットフォームを採用しているすべての製品で利用できるようになります。これは、アピリストアのような形で実現される可能性があります。

次に、教育・研修サービスプロバイダーも重要です。プラットフォームの使い方を教える研修プログラム、認定資格制度、オンライン学習コンテンツなどを提供します。これにより、プラットフォームを使いこなせる人材が業界全体で育成されます。

また、コンサルティングサービスプロバイダーも重要な役割を担います。プラットフォームを実際の製品に適用する際のアーキテクチャ設計支援、移行計画の策定、ベストプラクティスの提供といったサービスを提供します。特に、既存システムからSDxプラットフォームへの移行は複雑なプロセスであり、専門家の支援が不可欠です。

さらに、ツールベンダーも重要です。プラットフォームと連携する開発ツール、テストツール、シミュレーターなどを提供します。これらのツールがプラットフォームに最適化されることで、開発効率がさらに向上します。

そして、学術機関との連携も重要です。大学や研究機関がプラットフォームを教育や研究に活用することで、最新の研究成果をプラットフォームに取り込むことができます。また、学生がプラットフォームに慣れ親しむことで、将来の人材供給にもつながります。

このようなエコシステムが形成されることで、プラットフォーム自体の価値が加速度的に高まっていきます。これは、いわゆるネットワーク効果であり、利用者が増えるほど、プラットフォームの価値が高まるという好循環を生み出します。

実現に向けたロードマップ

リファレンスSDxプラットフォームの開発は、段階的に進めることができます。一度にすべてを完成させようとするのではなく、MVP(Minimum Viable Product)から始めて、徐々に機能を拡張していくアプローチが有効です。

第一フェーズでは、コアアーキテクチャの定義と、基本的なミドルウェアコンポーネントの開発に注力します。まず、参画企業間での合意形成を行い、プラットフォームの基本設計を固めます。この段階では、一つの業界(例えば自動車)にフォーカスし、その業界の基本的な要件を満たすプラットフォームを構築します。また、パイロットプロジェクトを実施し、実際の製品開発に適用して課題を洗い出します。

第二フェーズでは、ツールチェーンの整備と、他の業界への展開を進めます。パイロットプロジェクトからのフィードバックを反映し、開発ツール、CI/CDパイプライン、テストフレームワークを充実させます。また、最初の業界での知見を活かして、医療機器や産業機械といった他の業界向けのバリエーションを開発します。この段階で、プラットフォームを採用する企業が増え、エコシステムが形成され始めます。

第三フェーズでは、生成AI基盤の本格的な統合と、グローバル展開を目指します。生成AIの技術は急速に進化しているため、最新の技術を取り込み、開発プロセス全体を革新します。また、日本国内での実績を基に、海外市場への展開も視野に入れます。日本の安全・安心の強みを活かしたプラットフォームとして、グローバル市場でのポジションを確立します。

このロードマップは、市場の変化や技術の進化に応じて柔軟に調整していく必要があります。重要なのは、早期に実用的なプラットフォームを提供し、実際の使用を通じて継続的に改善していくというアプローチです。

SDxプラットフォームの構成要素の例

具体的なSDxプラットフォームの構成は、前述のブループリントの開発によって決定されます。ここでは、あくまでも部分的な一例としてプラットフォームの構成要素を列挙しています。

SDxプラットフォームは、複数の階層から構成される包括的なシステムです。それぞれの階層が特定の役割を担い、全体として統合されたソリューションを提供します。

SDxプラットフォームのブループリント、即ちリファレンスアーキテクチャは、プラットフォーム全体の設計思想と構造、そしてサービス/コンポーネント構成を決めるシステムモデリングのアプローチを定義します。これは、自動車の機能安全規格であるISO 26262、医療機器のIEC 62304、産業機械の安全規格といった、各業界の規制要件を満たすように設計されます。

セキュリティアーキテクチャも重要な要素です。ゼロトラストの原則に基づき、すべての通信を暗号化し、認証と認可を厳格に管理するアーキテクチャを定義します。特に、OTA更新の際のセキュリティは重要であり、更新パッケージの署名検証、ロールバック機能、段階的展開といった機能を標準で提供します。

さらに、安全性設計のパターンも含まれます。フェイルセーフ、フォールトレランス、冗長化といった安全性を確保するための設計パターンを、業界ごとの要件に合わせて提供します。例えば、ASIL-D(最高レベルの自動車機能安全)を要求される機能に対しては、デュアルコア・ロックステップ構成、独立した監視機構、診断機能といった具体的な実装パターンを提示します。

共通ツールチェーンは、開発者の生産性を飛躍的に向上させる統合開発環境です。これには、複数の重要なコンポーネントが含まれます。

メカエレ開発に利用されるCADツールや検証ツールで用いられるシミュレーションモデル群をソフトウェア開発で必要となるシステムのプラントモデルとして接合し、バーチャルな環境でコンカレントにメカエレとソフトウェアが開発できる仕組みをツールチェーンとして構成する必要があります。

統合開発環境(IDE)は、コーディング、デバッグ、テストを一貫して行える環境を提供します。重要なのは、単なるコードエディタではなく、プラットフォームの構造を理解し、適切な実装をサポートする知的な環境であることです。例えば、マイクロサービス間のインターフェース定義を自動生成したり、セキュリティ要件を満たさないコードに警告を出したりする機能を持ちます。

CI/CDパイプラインは、コードの変更から本番環境へのデプロイまでを自動化します。これには、ビルトの自動化、単体テスト・統合テスト・システムテストの自動実行、静的コード解析、セキュリティスキャン、性能テスト、そしてOTAパッケージの生成と配信までが含まれます。重要なのは、これらがプラットフォームの構造に最適化されていることです。例えば、マイクロサービスのどの部分が変更されたかを自動的に検出し、影響を受ける範囲だけをテストすることで、テスト時間を大幅に短縮できます。

テストフレームワークは、包括的な品質保証を可能にします。単体テストから統合テスト、システムテスト、さらにはハードウェア・イン・ザ・ループ(HIL)テストまで、一貫したフレームワークで実施できます。特に重要なのは、シミュレーション環境との統合です。デジタルツインを活用することで、実際のハードウェアを用意することなく、様々な状況でのテストが可能になります。

モニタリングと診断ツールも不可欠です。稼働中のシステムから、性能データ、エラーログ、セキュリティイベントなどを収集し、可視化します。これにより、問題の早期発見と迅速な対応が可能になります。また、このデータは、次の開発サイクルでの改善にも活用されます。

バージョン管理とコンフィギュレーション管理も重要です。マイクロサービスが多数存在する環境では、どのバージョンのサービスがどの環境に展開されているかを正確に把握することが重要です。また、ハードウェアのバリエーションに応じた設定管理も必要です。同じソフトウェアでも、搭載されるハードウェアによって設定を変える必要がある場合、それを効率的に管理する仕組みが求められます。

高信頼なSDxプラットフォーム向けOS(RTOS)も重要となります。マイコンから各種プロセッサーアーキテクチャを含むメニーコアのSoCまでスケーラブルに対応できるOSが必要となります。また、Linuxなどの汎用OSをアプリケーションドライバとして利用可能にしつつ、ミッションクリティカルな安全安心に関わるソフトウェアをガードできるハイパー・バイザ機能もこのOSには求められます。

汎用ミドルウェアコンポーネントは、多くのアプリケーションで共通に必要となる機能を、再利用可能な形で提供します。

通信ミドルウェアは、サービス間通信、車載ネットワーク(CAN、Ethernet、FlexRayなど)通信、クラウドとの通信を統一的に扱える抽象化層を提供します。開発者は、低レベルの通信プロトコルの詳細を意識することなく、メッセージの送受信を行えます。また、通信の信頼性を確保するために、再送制御、エラー検出、フロー制御といった機能も標準で提供されます。

セキュリティミドルウェアは、認証・認可、暗号化・復号化、鍵管理、セキュアブートといった機能を提供します。特に重要なのは、ハードウェアセキュリティモジュール(HSM)やトラステッド実行環境(TEE)といったハードウェアベースのセキュリティ機能を活用する際の抽象化層です。これにより、開発者はハードウェアの違いを意識することなく、高度なセキュリティ機能を実装できます。

OTA更新ミドルウェアは、ソフトウェアの遠隔更新を安全かつ効率的に実行する機能を提供します。これには、差分更新(変更された部分だけを送信)、圧縮、暗号化、署名検証、ロールバック(更新に失敗した場合に前のバージョンに戻す)、段階的展開(一部のデバイスで先行展開し、問題がないことを確認してから全体展開)といった高度な機能が含まれます。

データ管理ミドルウェアは、エッジデバイス上でのデータ収集、前処理、ローカルストレージ、クラウドへの同期といった機能を提供します。IoTデバイスは、大量のセンサーデータを生成しますが、すべてをクラウドに送信するのは非効率です。エッジ側で前処理を行い、重要なデータだけをクラウドに送信する仕組みが必要です。また、通信が一時的に途絶えた場合でも、ローカルにデータを保持し、通信が回復したら自動的に同期する機能も重要です。

診断とロギングミドルウェアは、システムの状態監視、エラー検出、ログ記録といった機能を統一的に提供します。これは、開発時のデバッグだけでなく、運用時の障害解析にも不可欠です。特に、分散システムでは、複数のサービスにまたがる処理の流れを追跡する分散トレーシング機能が重要になります。

生成AI活用基盤は、開発プロセス全体にわたって生成AIを活用するための環境です。これは、単にAI APIを呼び出すだけでなく、組込み開発の文脈に最適化された形で生成AIを活用できるようにします。

コード生成支援では、自然言語での機能説明から、プラットフォームの構造に適合したコードのスケルトンを生成します。重要なのは、生成されるコードが、プラットフォームのアーキテクチャパターン、コーディング規約、セキュリティ要件を満たしていることです。また、既存のサービスとのインターフェースも自動的に考慮されます。

テストケース生成では、関数やサービスの仕様から、網羅的なテストケースを自動生成します。境界値テスト、異常系テスト、性能テストなど、様々な観点からのテストケースを生成し、テスト漏れを防ぎます。

ドキュメント生成では、コードやアーキテクチャ図から、技術文書を自動生成します。これには、API仕様書、設計書、ユーザーマニュアルなどが含まれます。重要なのは、コードの変更に追従してドキュメントも自動的に更新されることです。これにより、ドキュメントとコードの乖離を防ぎます。

コードレビュー支援では、生成AIがコードを分析し、潜在的な問題を指摘します。これには、バグの可能性、セキュリティ脆弱性、性能問題、コーディング規約違反などが含まれます。人間のレビューは、AIが指摘した点を中心に確認することで、レビューの効率と品質を向上できます。

また、プラットフォーム固有の知識ベースも重要です。業界の規制要件、過去のトラブル事例との解決策、ベストプラクティスといった情報を生成AIに学習させることで、より文脈に適した支援が可能になります。

3-4 『人』から『知』へ:AI時代におけるソフトウェアベンダー成長の道

すべてのベンダーがプラットフォーム開発に参画できるわけではありません。また、プラットフォームという器を構築するだけでは、その真の価値は發揮されません。そこで、個別の課題解決に強みを持つシステムインテグレーションベンダーは、共通プラットフォームを活用するメーカーに対し、その潜在能力を最大限に引き出すための専門知識やノウハウを「ナレッジ提供型サービス」として提供することで、ビジネスモデルを転換します。

この転換は、一足飛びに実現するものではありません。生成AIを段階的に活用することで、以下の3つの活動層を経て、着実に実現します。重要なのは、これらは段階的に移行する「ステップ」ではなく、成熟度に応じて同時並行で回る「循環モデル」であるという点です。

ステップの概要

ステップ1:AI駆動 — AIスキル投資

- 目的:人の"時間"を"知"に変換
- 要素:AIリスキリング、プロンプト活用、自動化ツール導入
- キーワード:脱属人化・生産性向上・AIアシスト開発
- 象徴フレーズ:「AIが時間を拡張する時代へ」

まず、生成AIを実践的に活用することでスキルを習得し、開発力の基盤を構築することから始めます。この段階では、効率化によって生まれた時間を収益化するのではなく、将来の競争力強化のための投資期間として位置づけます。

ステップ2:IP駆動 — IPビジネス

- 目的:知の再利用と知的財産の収益化
- 要素:コード・テンプレート・ナレッジの部品化
- キーワード:再利用・知の資産化・ナレッジキャピタル
- 象徴フレーズ:「知を積み上げるほど強くなる開発組織へ」

次に、その習得したスキルと開発力の向上を根拠として、蓄積したナレッジを体系化されたIP(知的財産)として整備し、付加価値の高いサービスの提供と適切な価格設定を実現します。

ステップ3:ストック駆動 — ストック収益化

- 目的:知的資産を持続的な収益源に転換
- 要素:SaaSモデル、プラットフォーム化、継続課金型サービス
- キーワード:持続収益・価値循環・顧客共創
- 象徴フレーズ:「知が稼ぎ続ける組織へ」

最終的には、蓄積したナレッジをサービスとして自動提供することで、フロービジネスからストックビジネスへと移行し、ビジネスをスケールさせます。

変革の本質

AIで知を生み、IPで蓄え、ストックで稼ぐ。

— 人月からの脱却は、「知の循環経済」への進化である

従来の派遣やコンサルティングが労働力の補填や一時的な助言であったのに対し、このナレッジ提供型サービスの本質は、SDxプラットフォームやAIを駆使した新たな価値創出に必要な専門知識やノウハウを、メーカーの組織に継続的に埋め込み、定着させることです。これまで熟練エンジニアの属人的な作業に依存し、スケールしにくいという課題がありました。しかし、生成AIを活用することで、この課題を解決し、ナレッジ提供を自動化・サブスクリプション化することが可能になります。

戦略の本質的理解：段階的移行ではなく継続的循環

ここで重要な認識を明確にしておく必要があります。この3つのステップは、ステップ1からステップ2、ステップ2からステップ3へと一方向に移行し、最終的にすべてがストックビジネスになるという単純な段階的移行モデルではありません。

むしろ、これはフロー・IP・ストックという3つの活動が同時並行で回り続ける動的な循環モデルです。新しい課題解決で得た知見をAIでIP化し、成熟したIPの一部をストック化する。しかし常に新しいフロー案件で次のIPネタを生み出し続ける。この循環構造こそが、この戦略の核心なのです。

フロー案件は決して消滅しません。それは組織の「知のフロンティア」であり、ここで得られる新しい知見こそが継続的な競争力の源泉になります。すべての案件をストック化しようとすれば個別ニーズに対応できなくなり、すべてをフローのままにすれば規模が拡大しません。重要なのは、標準化可能な共通部分はストック化し、差別化要因となる個別部分はフローとして残すという、戦略的な切り分けです。

ハイブリッドモデルの現実性

組込みソフトウェア開発の本質的な特性を考えると、完全なストック化は非現実的です。顧客ごと、製品ごとにハードウェア構成や制約条件が大きく異なるこの領域では、個別性への対応が常に必要とされます。

したがって、実際のビジネスモデルは以下のような三層構造のハイブリッドモデルになります。

基盤層(ストック型)では、業界横断的な共通技術基盤、標準的な開発ツールチェーン、汎用的なプロンプトライブラリなどをサブスクリプション型で提供します。これは比較的標準化が可能で、多くの顧客に共通して価値を提供できる部分です。

IP活用層(フロー型+IP課金)では、個別プロジェクトでの技術支援を提供しますが、蓄積されたIPを活用することで高い生産性と品質を実現します。人月単価には、AI活用スキルとIP活用能力が反映され、従来より高い単価設定が正当化されます。ここでは、フロー型の収益に加えて、特定のIP利用に対する課金も可能になるでしょう。

先端開発層(純粋フロー型)では、まだIPが確立していない新しい技術領域や、極めて個別性の高い課題に対して、従来型の受託開発を継続します。しかしここで得られた知見は、次のサイクルでIP化され、やがて上位層に移行していきます。

この三層構造により、安定した継続収益(ストック)、高付加価値サービス(IP活用フロー)、新しい知見の創出(純粋フロー)という、バランスの取れた収益構造を実現できます。各企業の成熟度や

市場ポジションに応じて、この三層の比率は異なります。スタートアップは先端開発層の比率が高く、成熟企業は基盤層の比率が高くなるでしょう。

ステップ1:生成AI活用スキルの実践的習得

この段階は、従来の人月モデルや派遣モデルを維持しながら、ベンダー自身が生成AIを実践的に活用するスキルを習得することに専念するフェーズです。重要なのは、この段階では効率化の成果を収益増に転換することを目指すのではなく、将来的な競争力強化のための投資期間として位置づけることです。

実践を通じた学習の重要性

生成AIの活用スキルは、座学やトレーニングだけでは十分に習得できません。実際のプロジェクトにおいて試行錯誤を繰り返し、どのようなプロンプトが効果的か、どのような場面でAIを活用すべきか、どのような制約や限界があるかを体験的に学ぶ必要があります。この学習プロセスには一定の時間とコストがかかりますが、それは将来への不可欠な投資です。

委託ビジネスにおいては、従来通りの契約形態と価格設定を維持しながら、開発プロセスの中で積極的に生成AIを活用します。例えば、従来10人月かかっていた作業が生成AIの活用により6人月で完了できるようになったとしても、この段階では顧客に対して追加のプレミアムを請求したり、納期を短縮しながら従来の価格を維持したりすることは試みません。

なぜなら、この段階ではまだ生成AI活用のスキルが十分に成熟しておらず、効率化の程度も安定していないためです。また、顧客にとって生成AI活用の価値がまだ十分に可視化されていない段階で、追加費用を求めるることは信頼関係を損なうリスクがあります。むしろ、従来通りの価格で従来以上の品質や納期を実現することで、顧客満足度を高め、次のステップへの基盤を築きます。

委託ビジネスにおける実践

委託プロジェクトでは、生成AIを以下のような場面で積極的に活用しながら、そのノウハウを蓄積します。

コード生成支援では、関数やモジュールの初期実装において生成AIを活用し、どのようなプロンプトが高品質なコードを生成するかを学びます。テストコード生成では、単体テストや統合テストのコード作成に生成AIを活用し、テストの網羅性を高めるためのアプローチを習得します。ドキュメント生成では、技術仕様書や設計書の下書き作成に生成AIを活用し、効率的な文書化の手法を確立します。

重要なのは、生成AIの活用によって短縮された時間を、戦略的に配分することです。一部の時間はより上流の設計や品質向上といった顧客価値の向上に投入します。効率化で生まれた余裕を利益として享受するのではなく、より良い成果物の提供に振り向けることで、顧客からの信頼を獲得し、次のステップでの単価向上の基盤を築きます。

同時に、もう一部の時間は、生成AI活用のナレッジを自社のIP(Intellectual Property:知的財産)として体系化する作業に充てます。具体的には、効果的なプロンプトのパターン化と分類、特定のドメイン要件に対する技術的解決策の整理、アーキテクチャ設計における判断基準の明文化、過去のトラブル事例とその対処方法の構造化といった活動です。このIPの構築こそが、ス

ステップ2における独自の価値提案の源泉となります。単に個々のエンジニアのスキルが向上するだけでなく、組織として再利用可能なIPを構築することで、ステップ2での差別化を実現する基盤が整います。

派遣ビジネスにおける実践

派遣エンジニアの場合も、派遣先において生成AIを積極的に活用し、個人の生産性向上を実現します。派遣の単価は従来通りですが、派遣先企業にとっては、より高い生産性を持つエンジニアを受け入れることになり、結果的に満足度が向上します。

この段階で重要なのは、派遣エンジニアが個別に習得したスキルやノウハウを、派遣元企業の組織的な知見として蓄積する仕組みを構築することです。派遣エンジニアには定期的な報告や知見の共有を義務づけ、どのような場面でどのようにAIを活用したか、どのような成果が得られたか、どのような課題に直面したかを詳細に記録します。

この記録は、単なる日報ではなく、構造化されたナレッジベースとして整理されるべきです。例えば、特定の技術領域における効果的なプロンプトのパターン、特定の顧客業界における開発上の留意点、生成AIの出力を効果的に活用するためのレビュー手法などを体系的に蓄積します。

派遣先で得られた知見は、委託ビジネスへの転換や、次なるステップである専門性の向上へと繋がる重要な財産となります。特に、顧客のビジネス課題や組織文化への理解は、派遣という形態だからこそ得られる貴重な情報です。

スキル習得の評価指標

この段階の成功を測る指標は、収益や利益率ではなく、スキルの習得度合いと知見の蓄積量です。具体的には、エンジニアがどの程度生成AIを日常的に活用できているか、生成AIによる効率化の程度は安定しているか、効果的なプロンプトのパターンがどれだけ蓄積されたか、顧客満足度は維持または向上しているかといった指標で評価します。

また、この段階で重要なのは、生成AI活用による効率化の「再現性」を確立することです。特定のエンジニアだけが高い効率化を実現できるのではなく、チーム全体で一定水準の効率化を安定的に実現できる状態を目指します。そのためには、効果的なプロンプトやワークフローを組織内で共有し、標準化していく取り組みが必要です。

ステップ2への移行判断

ステップ1からステップ2への移行は、以下のような条件が満たされた時点で検討します。生成AI活用による開発効率の向上が安定的に40パーセント以上を達成していること、効果的なプロンプトのパターンやワークフローが組織内で標準化されていること、顧客満足度が従来と比較して維持または向上していること、そして最も重要なこととして、生成AI活用の価値が顧客にも認識され始めていることです。

この最後の点は特に重要です。顧客が、御社のエンジニアは生成AIを効果的に活用しており、それによって高品質な成果物が迅速に提供されていると認識していれば、次のステップでの価格交渉がスムーズに進みます。そのため、ステップ1の後半では、顧客に対して生成AI活用の取り組みとその成果を適切に説明し、理解を得ておくことが推奨されます。

ステップ2:生成AIによる付加価値の価格転換とIPの活用

ステップ1で生成AI活用のスキルを確立したベンダーは、この段階でその価値を適切に価格に反映させることができます。ここでの中心的な考え方は、ベンダーが提供しているのは単なる労働力ではなく、生成AIとIPを駆使した高度な技術的価値であるという認識を、顧客と共有することです。

IPの体系的整備

ステップ1で蓄積してきたナレッジを、より体系的なIPとして整備します。IPの形式は多様ですが、以下のような要素が考えられます。

プロンプトライブラリでは、特定のタスクや業界に最適化されたプロンプトのパターン集を構築します。例えば、医療機器開発における安全性要件の分析プロンプト、自動車制御システムのコード生成プロンプトといった、メイン特化型のプロンプトを体系化します。

技術的解決パターン集では、頻出する技術的課題とその解決策をパターン化します。例えば、リアルタイムシステムにおける性能最適化パターン、組込みLinuxのセキュリティ強化パターンなど、汎用性の高い解決策を整理します。

アーキテクチャテンプレートでは、特定のメインや製品カテゴリに対する標準的なシステムアーキテクチャの雛形を作成します。これにより、設計の初期段階を大幅に加速できます。

品質保証チェックリストでは、生成AIの出力を効果的にレビューするためのチェックポイントを整備します。AIが生成したコードやドキュメントに対して、どのような観点で品質を確認すべきかを明確化します。

これらのIPは、単に保存されているだけでは価値がありません。生成AIと組み合わせて効果的に活用できるように、検索可能・再利用可能な形で整備されている必要があります。

委託ビジネスにおける価格転換

委託ビジネスにおいて、ステップ2では人月単価の引き上げ、もしくは成果物ベースでの価格設定への転換を追求します。重要なのは、単に「生成AIを使っているから高い」という説明ではなく、顧客にとっての具体的な価値を明確に示すことです。

価値提案の構造としては、まず開発期間の短縮を示します。ステップ1で蓄積したIPと生成AIの組み合わせにより、従来と比較して開発期間を30から50パーセント短縮できることを実績データで示します。次に品質の向上を訴求します。体系的なレビュープロセスとAI活用により、バグの発生率が低減し、後工程での手戻りが減少することを定量的に示します。さらにリスクの低減として、過去のトラブル事例を体系化したIPにより、同様の問題の再発を防止できることを説明します。

具体的な価格設定手法としては、いくつかのアプローチがあります。人月単価アップ方式では、従来の人月単価に対して20から30パーセントのプレミアムを設定します。これは、生成AI活用とIP適用により、実質的な生産性が大幅に向上していることを根拠とします。

固定価格方式への移行では、人月単価ではなく、成果物単位での価格設定に移行します。例えば、特定の機能モジュールの開発に対して固定価格を設定し、生成AIとIPにより効率化できた分は、ベンダーの利益として確保します。

価値ベース価格設定では、顧客が得る価値に基づいて価格を設定します。例えば、開発期間短縮により顧客が得る市場投入のタイミング優位性、品質向上により削減される保守コストなどを定量化し、その一部を価格に反映させます。

ただし、価格転換は顧客との信頼関係を前提とします。ステップ1で十分な実績を示し、顧客の信頼を獲得していることが前提条件です。また、すべての顧客に対して一律に価格を引き上げるのではなく、まずは技術的理解度が高く、長期的な関係を築いている顧客から段階的に進めることができます。

派遣ビジネスにおける展開

派遣ビジネスにおいても、ステップ2では単価の向上を追求できます。ステップ1で生成AI活用スキルを確立したエンジニアは、派遣先において明らかに高い生産性と品質を実現できるため、その価値に見合った単価設定が可能になります。

派遣単価の引き上げの根拠としては、生成AIを活用した高度な開発スキル、業界特化型のIPに基づく迅速な問題解決能力、最新の開発手法への精通といった要素を提示します。従来の派遣単価に対して10パーセントから15パーセント程度の上乗せが現実的な目標となります。

単価アップの根拠

ベンダーは、単なる技術提供者から、顧客のビジネスを共に解決する技術パートナーへとポジションをシフトします。生成AIとIPを活用することで、従来よりも短期間で、より高品質な技術的提案や設計を提供できるため、人月単価の引き上げが正当化されます。

顧客に対しては、生成AIとIPを活用することで、設計品質が向上し、手戻りが減少し、結果的にプロジェクト全体のコストが削減されるという価値を明確に示します。単価は上がるかもしれません、プロジェクト全体での投資対効果は向上する、という説明です。

また、この段階で蓄積した知見やノウハウを生成AIで活用することで、ステップ1の委託ビジネスと同様に、期間短縮による売上増も見込めます。要求分析からアーキテクチャ設計、詳細設計までの期間が大幅に短縮されるため、同じ費用でより多くのプロジェクトを担当できるようになります。

ステップ3への移行判断

ステップ2からステップ3への移行は、以下のような条件が満たされた時点で検討します。生成AI活用基盤が十分に成熟し、安定的に高品質なサービスを提供できていること、人月単価が従来比で20パーセント以上向上していること、形式知化されたIPが一定規模に達し、体系的に整理されていること、そして複数の顧客において、ベンダーの生成AI活用とIPの価値が認識され、継続的な関係が構築されていることです。

特に最後の点は重要です。ステップ3でナレッジのサービス化を行う際には、既存の顧客基盤が不可欠です。ステップ2の段階で、生成AI活用とIPの価値を理解し、高い単価を支払ってくれる顧客を十分に獲得できていることが、ステップ3への移行の前提条件となります。

ステップ3:ナレッジのサービス化によるストックビジネスへの転換

ステップ1や2のアプローチでは、人月を効率化するだけなので、委託元にとっても委託先にとっても、所詮数倍にしかスケールしないのが課題です。この最終段階では、これまでに蓄積したナレッジをサービスとして提供し、ビジネスを大幅にスケールさせます。ただし、このステップは段階的に進めることが重要です。

段階的なサービス化のアプローチ

いきなり完全なSaaSを目指すのではなく、まず社内向けのナレッジベース構築から始めます。ステップ2で形式化したナレッジを、社内のエンジニアが効率的に活用できるシステムとして整備します。これにより、社内の生産性向上と品質の標準化を実現します。

次に、既存顧客向けのクローズドなサービスとして提供します。特定の顧客に対して、その顧客の案件で蓄積したナレッジを、カスタマイズされた形で提供します。例えば、特定顧客向け医療機器開発ナレッジベースといった形で、その顧客に特化した情報を生成AIを活用して提供します。このフェーズでは、まだ人間のコンサルタントが関与し、顧客の質問に対して適切なナレッジを提示したり、必要に応じてカスタマイズしたりします。

この段階で重要なのは、顧客からのフィードバックを収集し、ナレッジベースを継続的に改善することです。どのようなナレッジがよく利用されるか、どのような質問が多いか、どこに不足があるかといった情報を分析し、サービスの質を高めていきます。

十分な実績と改善を重ねた後、最終的にオープンなサービスプラットフォームに展開します。ただし、組込みソフトウェア開発という領域の特性を考えると、完全に自動化された汎用サービスよりも、特定の業界や技術領域に特化したサービスの方が現実的です。

ターゲット市場の明確化

組込みソフトウェア開発のナレッジは、顧客ごと、製品ごとにハードウェア構成や制約条件が大きく異なります。この個別性の高い領域で汎用的なサービスを提供するのは困難です。そのため、ターゲットとする顧客層を明確にする必要があります。

一つのアプローチは、特定の業界に特化することです。例えば、医療機器開発に特化したナレッジサービスであれば、IEC 62304やFDAの規制要件、典型的な開発プロセス、よくある課題とその解決方法など、医療機器業界に共通するナレッジを体系的に提供できます。同様に、産業用ロボット、自動車、航空宇宙など、業界ごとに特化したサービスが考えられます。

もう一つのアプローチは、特定の技術領域に特化することです。例えば、機能安全に特化したサービス、リアルタイムOSに特化したサービス、無線通信プロトコルに特化したサービスなど、技術的な専門性を軸にしたサービスも有効です。

スケーラビリティの本質

ステップ3におけるサービスモデルは、従来の労働集約型ビジネスとは次元の異なるスケーラビリティを実現します。この本質は「限界費用ゼロ」の実現にあります。

従来のビジネスモデルでは、顧客が増えれば増えるほど、提供するエンジニアの数も増やす必要があり、収益は線形にしか成長しません。一方、ナレッジサービスでは、一度IPを構築すれば

ば、それを複数の顧客に提供する際の追加コストはほぼゼロです。10社に提供しても、100社に提供しても、サービス提供の限界費用は極めて小さいのです。

ただし、完全に人手を介さないサービスは現実的ではありません。実際には、基本的なナレッジ提供はシステムが自動で行い、より高度な相談や個別のカスタマイズが必要な場合には、専門家が関与するハイブリッドモデルが効果的です。

このハイブリッドモデルでは、自動化層として標準的な質問への自動回答、過去事例の自動検索、基本的なコード生成支援を行い、人間支援層では複雑な課題の分析と解決策提案、カスタマイズされたIPの開発、新規ドメインへの展開支援を担います。

重要なのは、自動化層の範囲を徐々に拡大していくことです。最初は人間が対応していた質問の中で、パターン化できるものを見つけ出し、それを自動化層に移行していきます。この継続的な改善により、スケーラビリティは徐々に向上していきます。

収益モデルの設計

ステップ3の収益モデルは、従来の人月単価とは全く異なります。代表的な収益モデルとしては以下のようなものがあります。

サブスクリプションモデルでは、月額または年額の固定料金でナレッジベースへのアクセスを提供します。顧客の規模(エンジニア数、プロジェクト数)に応じた段階的な料金体系を設定します。

ティア型モデルでは、基本ティア(標準的なナレッジベースへのアクセス)、プロフェッショナルティア(高度なIP、優先サポート)、エンタープライズティア(カスタマイズIP、専任コンサルタント)といった段階を設けます。

使用量課金モデルでは、生成AIの活用回数、ダウンロードしたIPの数、コンサルテーション時間などに応じて課金します。

成果連動型モデルでは、顧客の開発効率向上やコスト削減の成果に応じて報酬を受け取ります。これは最も難易度が高いですが、顧客との利害を完全に一致させることができます。

実際には、これらを組み合わせたハイブリッド型の収益モデルが効果的です。例えば、基本的なサブスクリプション料金に加えて、高度なコンサルティングサービスは使用量課金、大規模なカスタマイズプロジェクトは成果連動型、といった形です。

戦略の本質的理解: 段階的移行ではなく継続的循環

ここで重要な認識を明確にしておく必要があります。この3ステップ戦略は、ステップ1からステップ2、ステップ2からステップ3へと一方向に移行し、最終的にすべてがストックビジネスになるという単純な段階的移行モデルではありません。

むしろ、これはフロー・IP・ストックという3つの活動が同時並行で回り続ける動的な循環モデルです。新しい課題解決で得た知見をAIでIP化し、成熟したIPの一部をストック化する。しかし常に新しいフロー案件で次のIPネタを生み出し続ける。この循環構造こそが、この戦略の核心なのです。

フロー案件は決して消滅しません。それは組織の「知のフロンティア」であり、ここで得られる新しい知見こそが継続的な競争力の源泉になります。すべての案件をストック化しようとすれば個別ニーズに対応できなくなり、すべてをフローのままにすれば規模が拡大しません。重要なのは、標準化可能な共通部分はストック化し、差別化要因となる個別部分はフローとして残すという、戦略的な切り分けです。

ハイブリッドモデルの現実性

組込みソフトウェア開発の本質的な特性を考えると、完全なストック化は非現実的です。顧客ごと、製品ごとにハードウェア構成や制約条件が大きく異なるこの領域では、個別性への対応が常に必要とされます。

したがって、実際のビジネスモデルは以下のような三層構造のハイブリッドモデルになります。

基盤層(ストック型)では、業界横断的な共通技術基盤、標準的な開発ツールチェーン、汎用的なプロンプトライブラリなどをサブスクリプション型で提供します。これは比較的標準化が可能で、多くの顧客に共通して価値を提供できる部分です。

IP活用層(フロー型+IP課金)では、個別プロジェクトでの技術支援を提供しますが、蓄積されたIPを活用することで高い生産性と品質を実現します。人月単価には、AI活用スキルとIP活用能力が反映され、従来より高い単価設定が正当化されます。ここでは、フロー型の収益に加えて、特定のIP利用に対する課金も可能になるでしょう。

先端開発層(純粋フロー型)では、まだIPが確立していない新しい技術領域や、極めて個別性の高い課題に対して、従来型の受託開発を継続します。しかしここで得られた知見は、次のサイクルでIP化され、やがて上位層に移行していきます。

この三層構造により、安定した継続収益(ストック)、高付加価値サービス(IP活用フロー)、新しい知見の創出(純粋フロー)という、バランスの取れた収益構造を実現できます。

組織構造の根本的変革の必要性

この循環モデルを実装するには、実は組織構造そのものの変革が不可欠です。

従来の組込みソフトウェア開発組織は、プロジェクト単位で動いています。プロジェクトチームが立ち上がり、開発が終われば解散し、メンバーは次のプロジェクトに移っていく。この構造では、プロジェクトで得られた知見が組織に蓄積されにくいのです。

この3つの活動を回すためには、プロジェクト組織とは別に、組織横断的なIP管理機能が必要になります。この機能の役割は、各プロジェクトから知見を吸い上げ、形式化し、品質を保証し、IPベースに登録し、次のプロジェクトで活用できるようにすることです。具体的には、以下のような体制が考えられます。

IPキュレーションチームは、各プロジェクトで生まれた知見を収集し、体系的に整理し、再利用可能な形に加工します。彼らは、どの知見がIP化する価値があるか、どのように分類・タグ付けするか、どのメタデータを付与するかを判断します。

IP品質管理チームは、登録されるIPの品質を検証し、適用範囲や制約条件を明確にし、定期的な見直しと更新を担当します。彼らは、IPの有効性を継続的にモニタリングし、陳腐化したIPを特定し、更新または廃棄を推奨します。

AIプラットフォームチームは、生成AIツールの選定と導入、社内向けRAGシステムの構築と運用、プロンプトエンジニアリングのベストプラクティス共有などを担当します。さらに重要な役割として、蓄積したIPを顧客向けストックサービスとして提供するための技術基盤の構築と運用も担います。具体的には、ナレッジベースのSaaS化に必要なAPI設計と実装、セキュアなマルチテナント環境の構築、IPの検索・推薦機能の開発、サービスのスケーラビリティとパフォーマンス最適化などを行います。彼らは、技術的な基盤を整備することで、各プロジェクトチームが社内でIPを効率的に活用できる環境を提供するとともに、そのIPを外部顧客に対して安定的かつ高品質なサービスとして届けるためのプラットフォームを構築します。

この組織変革は、従来の日本企業が得意としてきた「現場の暗黙知」を重視する文化とは、ある意味で対立する概念です。むしろ、知見を積極的に外部化し、誰でもアクセス可能な形にすることを評価する文化が必要になります。

希望のメッセージ:誰でも始められる変革

この3ステップ戦略の複雑さを見て、「こんな高度なことは、うちの会社には無理だ」と感じた方もいるかもしれません。しかし、最も重要なメッセージは、この変革は誰でも、今日から始められるということです。

完璧を目指さず、小さく始める

ストックビジネスへの転換、IPの体系的整備、品質保証体制の構築…これらすべてを最初から完璧に実現する必要はありません。むしろ、完璧を目指して動けなくなることの方が問題です。

まず始めるべきは、極めてシンプルなことです。今日のプロジェクトで、生成AIを1つだけ使ってみる。それだけで十分です。GitHub Copilotでコードを書いてみる、ChatGPTで設計書の下書きを作ってみる、何でも構いません。

うまくいったプロンプトを、1つだけメモしておく。Excelでも、Notionでも、紙のノートでも構いません。「このプロンプトを使ったら、こういうコードが生成された。使えた」というメモを1つ残すだけです。

これがIPの始まりです。最初から完璧なナレッジベースを作る必要はありません。1つのメモから始めて、それが10個になり、100個になり、やがて体系化されていきます。

失敗は財産:試行錯誤を恐れない文化

AI活用において、失敗は避けられません。プロンプトがうまく機能しないこともあれば、AIが的外れな回答をすることもあります。しかし、それらの失敗こそが最も価値ある学びです。

「このプロンプトではうまくいかなかった」という記録も、立派なIPです。なぜなら、同じ失敗を繰り返さずに済むからです。また、「なぜうまくいかなかったのか」を分析することで、より深い理解が得られます。

重要なのは、失敗を責めない組織文化を作ることです。「AI使ってみたけど、全然ダメだった」という報告が、堂々とできる環境。そして、「じゃあ、なぜダメだったのか、みんなで考えてみよう」という前向きな雰囲気。それが、AI時代の強い組織を作ります。

業界全体で支え合う:孤独な戦いではない

この変革は、1社だけで成し遂げる必要はありません。むしろ、業界全体で知見を共有し、互いに支え合うことで、より大きな成果が得られます。

JASAのような業界団体の役割は、まさにここにあります。各社が試行錯誤で得た知見を持ち寄り、「これはうまくいった」「これは失敗だった」という情報を共有することで、業界全体のレベルが上がります。

特に重要なのは、成功事例だけでなく、失敗事例も共有することです。「うちはこういうアプローチでストックビジネスを試したけど、顧客に受け入れられなかった。理由はおそらく…」という率直な情報交換が、他社の失敗を防ぎます。

競争すべき領域と協力すべき領域を明確に区別することが重要です。AI活用の基本的なスキルやツールの使い方は、業界全体で協力して底上げする。一方で、各社独自のドメイン知識やIP、顧客との関係性といった部分で競争する。このバランスを取ることで、業界全体が強くなりながら、個社の独自性も保てます。

後から参加しても遅くない:むしろ学習コストが下がる

「今から始めて、先行している企業に追いつけないので」という心配は無用です。むしろ、少し遅れて参加する方が、先行者の試行錯誤から学べるという利点があります。

生成AI技術自体も日々進化しています。一年前には最先端だったツールが、今では標準になつたり、より使いやすい新しいツールが登場しています。先行者は古いツールの使い方も学ばなければなりませんでしたが、後発組は最新のツールから始められます。

また、業界内で知見が共有されていけば、「どこから始めればいいか」「どんな落とし穴があるか」といった情報が蓄積されていきます。それは後から参加する企業にとって、大きな財産になります。

重要なのは、完璧なタイミングを待つのではなく、今できることから始めることです。一年後に「あの時始めていればよかった」と後悔するよりも、今日から小さな一步を踏み出す方が、はるかに価値があります。

この変革は義務ではなく機会:自分たちのペースで、自分たちのやり方で

最後に、最も重要なことを強調しておきます。この3ステップ戦略は、すべての企業が必ず実行しなければならない義務ではありません。これは、人月ビジネスから脱却し、より高い付加価値を提供したいと考える企業にとっての、一つの道筋です。

人月ビジネスが悪いわけではありません。多くの顧客は、信頼できるエンジニアの労働力を必要としています。その需要に応えることは、立派な価値提供です。ただ、もし「人手不足で案件を受けられない」「単価が上がらず厳しい」「もっと技術的に面白い仕事がしたい」と感じているなら、この戦略は一つの選択肢になり得ます。

そして、この道を選ぶとしても、自分たちのペースで、自分たちのやり方で進めていけば大丈夫です。レポートに書かれているのはあくまで一つのモデルであり、実際の実装は各社の状況に応じて柔軟に調整できます。大事なのは、完璧に真似ることではなく、この考え方のエッセンスを理解し、自社に合った形で取り入れることです。

一社一社が小さな一步を踏み出し、その経験を業界で共有し合うことで、やがて大きなムーブメントになっていきます。完璧な企業だけが参加できる変革ではなく、不完全な私たちが、一緒に学び、一緒に成長していく変革。それこそが、この戦略の真の姿なのです。

実践上の注意点

ステップ間の移行判断

各ステップ間の移行は、明確な条件に基づいて判断すべきです。無理に次のステップに進むと、準備不足により失敗するリスクがあります。一方で、過度に慎重になって同じステップに留まり続けると、市場の変化に遅れを取る可能性もあります。

移行判断のためには、各ステップで定量的な目標を設定し、それを定期的、例えば四半期ごとに評価します。目標を達成していれば次のステップへの準備を始め、達成していなければ現在のステップでの取り組みを強化します。

特に、ステップ1からステップ2への移行は慎重に判断する必要があります。生成AI活用のスキルが十分に確立されていない段階で価格転換を試みると、顧客との信頼関係を損なうリスクがあります。逆に、十分なスキルが確立されているにもかかわらず、価格転換を躊躇していると、競合に先を越される可能性があります。

また、ステップ2とステップ3は必ずしも段階的に進める必要はありません。ステップ2で専門性を高めながら、並行してステップ3の準備、ナレッジベースの構築やサービスのプロトタイプ開発を進めることも可能です。重要なのは、各ステップの本質的な目的を理解し、それを達成することです。

失敗のリスクとその対処法

各ステップには固有のリスクがあり、それらへの対処法を事前に考えておく必要があります。

ステップ1では、生成AI活用のスキル習得が想定より時間がかかり、プロジェクトの収益性が悪化するリスクがあります。対処法としては、生成AI活用の習熟度が低い初期段階では、リスクの低い小規模プロジェクトから始め、徐々にスケールを拡大することです。また、社内での研修やナレッジ共有を充実させ、チーム全体のスキル向上を加速させます。

ステップ2では、価格転換がうまくいかず、顧客から受け入れられないリスクがあります。対処法としては、既存顧客の中でも、特に技術的な理解度が高く、長期的な関係を築いている顧客から価格交渉を始めることです。また、価格アップの根拠を明確に説明できる資料を準備し、定量的なメリットを示します。

ステップ3では、サービス開発に予想以上のコストがかかったり、顧客の反応が期待より低かったりするリスクがあります。対処法としては、大規模な投資をする前に、MVPを開発し、少数の顧客で検証することです。市場の反応を見ながら、徐々に機能を拡張し、投資を増やしていきます。

組織的な要件と文化変革

このステップを実行するためには、技術や戦略だけでなく、組織体制や企業文化の変革も必要です。

ステップ1でスキル習得を組織的に推進するためには、エンジニアが安心して生成AIを試行錯誤できる環境を整備する必要があります。失敗を許容する文化、学びを共有する仕組み、適切な評価制度などが重要です。生成AI活用の取り組みを評価指標に組み込み、積極的に活用したエンジニアを適切に評価します。

ステップ2で専門性を高めるためには、エンジニアが特定のドメインに長期間専念できる体制が必要です。多くのベンダーは案件ごとにアサインメントを変える運用をしていますが、これを変えるには、組織構造を案件ベースからドメインベース、あるいは顧客ベースに再編する必要があるかもしれません。例えば、医療機器チーム、自動車チームといった形で組織を作り、各チームが特定の領域に特化して深い専門性を築きます。

ステップ3でサービスを開発・運用するには、従来のプロジェクト型開発とは全く異なるスキルセットが必要です。プロダクトマネジメント、顧客ニーズを継続的に把握し、製品をアップデートする、DevOps、継続的なデプロイとインフラ管理、カスタマーサクセス、顧客の成功を支援し、解約を防ぐといった役割が重要になります。これらの人材を外部から採用するか、既存のメンバーを再教育するか、あるいは外部パートナーと協業するかを検討する必要があります。

また、評価制度や報酬体系も見直す必要があるでしょう。従来の人月ビジネスでは、稼働時間や案件数が評価の中心でしたが、新しいモデルでは、顧客満足度、専門性の向上度、ナレッジの貢献度、サービスの利用率向上など、異なる指標が重要になります。

3-5 二層構造による共存共栄モデル

このアプローチは、JASAが中心となり共通プラットフォームを開発・提供し、ベンダー各社はその上でそれぞれの強みを活かしたナレッジサービスを生成AIを活用して提供するという持続可能な共存共栄モデルを構築します。

共通プラットフォームは、個々のベンダーが単独で開発するには負担が大きすぎる基盤技術を、業界全体で共有することで効率化します。一方、ナレッジサービスは各ベンダーの差別化要因であり、それぞれが独自の専門性や顧客関係を活かして価値を提供します。

この二層構造により、ベンダー間の健全な競争と協調が両立します。プラットフォーム層では協力し、サービス層では競争するという構図です。これは、単発の受託開発に依存しない、安定したストック型の収益基盤を築くとともに、日本の製造業全体のSDxへの移行を強力に後押しすることが可能となります。

重要なのは、この変革は一朝一夕には実現しないということです。ステップ1でスキルを習得し、ステップ2で価値を価格に転換し、ステップ3でサービス化するという段階を着実に進め、失敗から学び、継続的に改善していく姿勢が必要です。

また、すべてのベンダーが同じ道を進む必要はありません。自社の強みや市場環境に応じて、プラットフォーム構築に注力する企業、ナレッジサービスに特化する企業、両方を並行して進める企業など、多様なアプローチがあつてよいでしょう。

JASAの役割は、このような多様な取り組みを支援し、成功事例や失敗事例を共有し、業界全体の知見を高めることにあります。理論だけでなく、実践から得られた生の知見こそが、業界の真の変革を推進する原動力となるはずです。

知の循環経済への進化

AIで知を生み、IPで蓄え、ストックで稼ぐ。

これは単なるビジネスモデルの転換ではなく、人月という労働集約型モデルから、知的資本を基盤とした循環経済への根本的な進化です。この進化を通じて、日本のソフトウェア産業は持続可能な成長基盤を確立し、グローバル競争においても優位性を発揮できるようになるでしょう。

第3章 参考文献一覧(APA形式)

経済産業省. (2024). 生成AIの産業活用に関する検討会報告書. 東京: 経済産業省.

日経クロステック. (2024). 「人月ビジネスの終焉」生成AIが変える開発現場. 日経クロステック, 2024年5月号.

独立行政法人 情報処理推進機構 (IPA). (2024). ソフトウェア産業構造転換のためのDX推進報告書. 東京: IPA.

Gartner. (2024). Market Guide for Software Development Service Providers in the Age of Generative AI. Stamford, CT: Gartner Inc.

McKinsey & Company. (2024). The Next Software Economy: AI, Platforms, and Value Creation. New York, NY: McKinsey Global Institute.

Waseda University Business School. (2023). 日本のSI産業における「人月モデル」からの脱却に関する実証研究. 早稲田大学経営研究紀要.

OpenAI. (2024). Generative AI and the Future of Developer Productivity. San Francisco, CA: OpenAI.

Microsoft Research. (2024). AI-Powered Software Engineering Transformation. Redmond, WA: Microsoft.

Red Hat. (2023). Open Hybrid Cloud Strategy and Software-Defined Transformation. Raleigh, NC: Red Hat Inc.

一般社団法人 組込みシステム技術協会 (JASA). (2024). SDx時代の組込みソフトウェア産業変革ロードマップ. 東京: JASA.

渡辺博之. (2025). ベンダー変革論—一人から知への移行. 未発表原稿 (eXmotion社内資料).

第4章:JASAの提供価値とアクションプラン

本レポートで提示した課題は、個別企業が単独で解決できる性質のものではありません。SDxへの対応、生成AIの戦略的活用、人月ビジネスからの脱却、共通プラットフォームの構築といった課題は、いずれも業界全体で取り組むべき構造的な問題です。個社が独自に投資を行い、リスクを負担しながら新しい技術やビジネスモデルに挑戦することは、経営資源が限られる中小企業にとっては極めて困難です。また、大企業であっても、業界標準の不在や人材の流動性の低さといった業界構造的な問題には、単独では対処しきれません。

このような状況において、業界団体であるJASA(一般社団法人組込みシステム技術協会)の果たすべき役割は、これまで以上に重要性を増しています。JASAは単なる情報提供者や交流の場の提供者にとどまらず、業界全体の変革を促進する触媒として、会員企業の実践的な取り組みを支える存在へと進化する必要があります。

触媒としてのJASAの役割とは、具体的には三つの機能を果たすことを意味します。第一に、変革の方向性を明確に示す羅針盤機能です。技術動向や市場環境の変化を継続的に分析し、業界が進むべき方向を提言することで、個社の戦略立案を支援します。第二に、実践の場を提供するプラットフォーム機能です。新しい技術やビジネスモデルを実証し、成功事例と失敗事例を共有することで、会員企業のリスクを低減し、変革への一歩を踏み出しやすくなります。第三に、協業を促進するハブ機能です。メーカーとベンダー、大企業と中小企業、産業界と学術界といった異なるステークホルダーを結びつけ、単独では実現困難なプロジェクトを推進します。

この役割を果たすため、JASAは5つの核心的価値を提供するとともに、具体的な活動を推進する新たな委員会組織を立ち上げます。これらの取り組みは相互に連携し、理論から実践へ、個社の努力から業界全体のエコシステム形成へと、段階的に変革を深化させていくための包括的なフレームワークを構成しています。各取り組みは独立して機能するのではなく、相乗効果を生み出すように設計されています。たとえば、技術的解決策の提示と人材育成は一体的に進められ、ビジネスモデル変革の支援は共通プラットフォーム開発と連動します。このような統合的なアプローチにより、表面的な変化ではなく、業界の構造そのものを変革することを目指しています。

JASAが提供する5つの核心的価値

4-1 課題の明確化と戦略的提言

第一に、課題の明確化と戦略的提言を行います。経済産業省の「PIVOT」戦略やグローバルなSDxの潮流を深く分析し、日本の製造業が直面する本質的な課題を明確に提示します。本レポートはその第一歩であり、今後も継続的に業界動向を分析し、産業界が進むべき方向性を指示する羅針盤としての役割を果たします。

変化の激しい技術環境において、個社が独自に全体像を把握することは極めて困難です。生成AIの進化は月単位で加速し、クラウドサービスの機能は週単位で更新され、オープンソースのエコシステムは日々変容しています。このような環境下で、自社の業務に追われながら最新動向を追跡し、それが自社のビジネスにどのような影響を与えるかを分析することは、大企業であっても容易ではありません。ましてや中小企業にとっては、専任の調査担当者を配置することすら難しい状況です。

JASAは業界横断的な視点から俯瞰的な分析を提供し、各企業が戦略的判断を行うための基盤情報を整備します。具体的には、海外の先進事例の調査、国内外の技術動向レポートの作成、政府の産業政策との整合性分析、競合となる海外プラットフォーマーの戦略分析などを継続的に実施します。これらの情報は、単なるデータの羅列ではなく、日本の組込みソフトウェア業界にとっての意味を解釈し、実践的な示唆を抽出した形で提供されます。

さらに重要なのは、業界全体として取り組むべき優先順位を明確にすることです。すべての課題に同時に対処することは不可能であり、限られた資源をどこに集中投資すべきかという判断が求められます。JASAは会員企業との対話を通じて、業界として最も重要な課題を特定し、段階的なロードマップを提示します。本レポートもその一環であり、2025年から2030年に向けて、どのような順序で変革を進めるべきかという指針を示しています。

また、政府や関連機関への政策提言も重要な役割です。業界の実態を最もよく理解しているJASAが、現場の声を集約し、実効性のある政策の形成に貢献することで、産業界全体の環境整備を促進します。税制優遇、研究開発支援、人材育成プログラム、国際標準化への参画など、個社では働きかけが難しい領域において、業界団体としての影響力を発揮します。

4-2 技術的解決策の提示

第二に、技術的解決策の提示を行います。SDxの基盤となるオープンシステム開発、モダンなアーキテクチャ、そして生成AIの戦略的活用など、具体的な技術的ソリューションを提示します。これには、業界横断的な参照アーキテクチャの策定、ベストプラクティスの文書化、実践的な技術ガイドラインの作成が含まれます。

技術的解決策の提示において重要なのは、理論的な正確性と実践的な適用可能性のバランスです。学術的には優れたアプローチであっても、現実の開発現場で適用できなければ意味がありません。逆に、短期的には機能しても、長期的な保守性や拡張性に問題があるアプローチでは、将来的な競争力を失います。JASAは、理論と実践の両面から検証された、真に有用な技術手法を提供します。

参照アーキテクチャの策定では、自動車、ロボティクス、医療機器、産業機械など、各産業分野の特性を考慮しながらも、共通化できる部分を見極めます。たとえば、センサーからのデータ取得、エッジでの前処理、クラウドへの送信、AIによる分析、制御へのフィードバックという基本的なデータフローは、多くの産業に共通しています。しかし、リアルタイム性の要求レベル、安全性の認証基準、通信プロトコルの選択などは産業ごとに異なります。参照アーキテクチャは、この共通性と個別性を適切に抽象化し、各企業が自社の要件に合わせてカスタマイズできる柔軟性を持たせる必要があります。

ベストプラクティスの文書化では、成功事例だけでなく失敗事例も重要な知見として共有します。特定の技術を適用しようとして失敗した場合、何が問題だったのか、どうすれば回避できたのかという分析は、他の企業が同じ失敗を繰り返さないために貴重です。JASAは、会員企業が安心して失敗事例を共有できる場を提供し、業界全体の学習を加速させます。

実践的な技術ガイドラインには、具体的なコード例、設定ファイルのサンプル、チェックリスト、評価基準などが含まれます。たとえば、生成AIを開発プロセスに導入する際のガイドラインでは、どのような開発タスクがAI支援に適しているか、どのようなプロンプトが効果的か、生成されたコードをどのように検証すべきか、といった実務的な内容を提供します。これらは、実際にJASA会員

企業が試行錯誤した結果を集約したものであり、教科書的な知識ではなく、現場で即座に活用できる実践知です。

4-3 ビジネスマネジメントの支援

第三に、ビジネスモデル変革の支援を行います。「人月ビジネス」から脱却し、高付加価値な「ナレッジ提供型サービス」へと転換するための知見と具体的な事例を提供します。ビジネスモデルの転換は経営判断の核心に関わる難題であり、多くの企業が必要性を認識しながらも、具体的な一歩を踏み出せずにいます。

人月ビジネスからの脱却が困難な理由は、技術的な問題だけではありません。むしろ、組織文化、顧客との関係性、リスク管理、キャッシュフロー管理といった、経営全般に関わる複雑な課題が絡み合っています。たとえば、長年の取引関係がある顧客に対して、突然価格体系を変更することは容易ではありません。また、安定した人月単価による収入から、成果報酬型や継続課金型のビジネスモデルに移行することは、短期的には収益の変動リスクを高めます。さらに、エンジニアの評価制度や報酬体系も、労働時間ベースから価値創造ベースへと転換する必要があります。

JASAは、こうした多面的な課題に対して、包括的な支援を提供します。まず、ビジネスモデル変革に成功した企業の事例を収集し、どのような順序で変革を進めたか、どこで困難に直面しどう乗り越えたか、どのような成果が得られたかを詳細に分析します。これらの事例は、単なる成功物語ではなく、失敗と試行錯誤の過程も含めて共有されます。たとえば、ある企業が価値ベースの価格設定に移行しようとした際、最初の提案が顧客に受け入れられず、どのように説明を工夫し、段階的な移行プランを設計したかといった、実践的な知見が蓄積されます。

価格設定の考え方については、具体的なフレームワークを提供します。従来の工数積算による価格設定から、顧客が得る価値に基づく価格設定へと移行するためには、価値をどのように定量化し、顧客と合意形成するかが鍵となります。たとえば、生成AIを活用した開発支援サービスであれば、開発期間の短縮効果、品質向上による不具合削減効果、技術者のスキル向上効果などを、具体的な数値で示す必要があります。JASAは、このような価値の定量化手法、顧客との価値共有プロセス、契約書のひな型などを整備し、会員企業が新しいビジネスモデルを実践しやすい環境を整えます。

契約形態の工夫も重要な支援領域です。従来の請負契約や準委任契約に加えて、成果報酬型契約、ライセンス契約、サブスクリプション契約など、多様な契約形態の選択肢を提示します。それぞれの契約形態には、リスク配分、収益認識のタイミング、知的財産権の扱いなど、法務的な検討事項があります。JASAは弁護士や公認会計士などの専門家と連携し、技術者だけでは対応困難な領域についても、実務的なガイダンスを提供します。

顧客との関係性構築については、単発のプロジェクトベースの関係から、長期的なパートナーシップへの転換を支援します。これには、顧客との定期的な戦略対話の場の設定、技術ロードマップの共同策定、継続的な価値提供のための組織体制構築などが含まれます。従来のベンダーとしての立場から、顧客の事業成長に貢献する戦略的パートナーへと進化するためのノウハウを、JASAは体系化し共有します。

4-4 人材育成とコミュニティ形成

第四に、人材育成とコミュニティ形成を推進します。SDx時代に不可欠なクロスファンクショナル・エンジニア(ドメイン知識とソフトウェア技術を併せ持つ人材)の育成を強力に支援します。技術セミナー、ワークショップ、勉強会を通じて最新技術を学ぶ機会を提供するだけでなく、メーカーとベンダーが対等な立場で知見を交換し、連携・共創できるオープンなエコシステムを形成します。

人材育成における最大の課題は、単なる技術知識の習得ではなく、それを実際のビジネス価値に変換できる実践力の養成です。多くの技術セミナーや資格制度は、特定の技術の知識や操作方法を教えることに焦点を当てていますが、それだけでは不十分です。重要なのは、顧客のビジネス課題を理解し、適切な技術を選択・組み合わせて解決策を設計し、実装から運用まで責任を持つ総合力です。

JASAの人材育成プログラムは、この実践力の養成に重点を置きます。たとえば、生成AIの活用に関するワークショップでは、単にAIツールの使い方を学ぶのではなく、実際のプロジェクトを題材として、要件定義から設計、実装、テスト、デプロイまでの一連のプロセスをAI支援で行う体験を提供します。参加者は、AIが得意なタスクと不得意なタスクを実感し、人間の判断が必要な局面を理解し、AIと協働する最適なワークフローを自ら設計する経験を積みます。

クロスファンクショナル・エンジニアの育成では、技術とビジネスの両面に精通した人材を系統的に養成します。従来の組込みエンジニアは、特定のハードウェアやリアルタイムOSに深い専門性を持つ一方で、クラウド技術やデータ分析には不慣れな場合が多くあります。逆に、クラウドやAIの技術者は、エッジデバイスの制約やリアルタイム制御の要求を十分に理解していないことがあります。SDx時代には、これらの異なる領域を横断的に理解し、全体最適を図れる人材が求められます。JASAは、組込みエンジニア向けのクラウド・AI研修、クラウドエンジニア向けの組込み・制御研修など、それぞれの専門性を基盤としながら視野を広げるプログラムを提供します。

コミュニティ形成も重要な取り組みです。人材不足が深刻化する中、企業の枠を超えた学びと交流の場を提供することで、業界全体の人材基盤を強化します。JASAが提供するコミュニティには、いくつかの重要な機能があります。第一に、情報交換の場としての機能です。最新技術の動向、プロジェクトでの成功事例や失敗事例、効果的なツールやライブラリの情報などが、会員間で自然に共有されます。第二に、人脈形成の場としての機能です。将来的な協業パートナーの発見、転職やキャリアチェンジの際のネットワーク、技術的な困難に直面した際の相談相手など、人的なつながりが構築されます。第三に、モチベーション維持の場としての機能です。同じ課題に取り組む仲間の存在は、困難な変革に挑戦する際の心理的な支えとなります。

メーカーとベンダーの関係性も、コミュニティを通じて変化します。従来は、メーカーが発注者、ベンダーが受注者という明確な上下関係がありました。SDx時代には両者が対等なパートナーとして協働する必要があります。JASAのコミュニティでは、企業の規模や立場に関わらず、技術的な知見や課題を率直に共有できる文化を醸成します。メーカーの技術者がベンダーの先進的な取り組みから学び、ベンダーの技術者がメーカーのドメイン知識を深く理解する機会を提供することで、相互の理解と信頼を深めます。

4-5 共通プラットフォーム開発の促進

第五に、共通プラットフォーム開発の促進を行います。日本の製造業全体が利用できる「共通SDxプラットフォーム」の開発を、会員企業と共に推進します。この取り組みには、共同での要件定義、アーキテクチャ設計、実装、検証といった各開発フェーズにおけるハブとしての機能と、プロジェクト推進の支援が含まれます。

共通プラットフォームの必要性は、グローバル競争の現実から生じています。欧米や中国の巨大プラットフォーマーは、クラウドからエッジまでを統合したエコシステムを構築し、膨大な開発投資を行っています。日本企業が個別にこれらに対抗することは、資源的に困難です。しかし、業界で力を合わせれば、対抗可能な規模の投資を実現できます。

共通プラットフォームの開発において重要なのは、共通化すべき部分と差別化すべき部分の見極めです。すべてを共通化してしまえば、企業間の競争優位が失われます。逆に、すべてを個別開発すれば、コストとリスクが高すぎて実現できません。JASAは、各産業分野の特性を分析し、共通化によって効率を高められる基盤部分と、各企業が独自性を発揮すべきアプリケーション層を明確に区分します。

たとえば、センサーデータの収集・前処理、通信プロトコルの実装、セキュリティ機能、OTA更新機能、ログ管理機能などは、多くの産業に共通するニーズがあり、個別開発の必要性は低い領域です。これらを共通プラットフォームとして提供することで、各企業は自社のコア技術である制御アルゴリズムやユーザーインターフェース、業界特有の機能開発に資源を集中できます。

共通プラットフォームの開発プロセスでは、JASAはプロジェクトマネジメントとステークホルダー調整の中核を担います。異なる企業、異なる産業分野からの要求を統合し、全体最適を図りながら仕様を策定することは、技術的な課題以上に難しい組織的な課題です。各社の利害関係を調整し、公平で透明なプロセスを確保し、参加企業すべてが納得できる成果物を生み出すために、中立的な立場のJASAが調整役を果たします。

また、オープンソースソフトウェアとの関係も重要な検討事項です。ゼロから独自に開発するのではなく、既存の優れたオープンソースプロジェクトを活用し、日本の産業界特有のニーズに対応する部分を追加開発するアプローチが、効率的かつ現実的です。JASAは、どのオープンソースプロジェクトを採用すべきか、どのような貢献をコミュニティに対して行うべきか、ライセンスやガバナンスをどう設計すべきかといった、オープンソース活用の戦略を策定します。

共通プラットフォームの持続可能性も考慮します。初期開発だけでなく、継続的な保守・更新、新技术の取り込み、セキュリティパッチの適用など、長期的な運用体制が必要です。JASAは、プラットフォームの運用を担う組織体制、資金調達メカニズム、貢献企業へのインセンティブ設計など、持続可能なエコシステムの構築を支援します。

4-6 アクションプラン:新委員会の設置

前節で述べた5つの提供価値を、抽象的な理念にとどめず、具体的な成果として実現するため、JASAは新たに4つの専門委員会を発足させます。これらの委員会は、本レポートが掲げる「人月からの脱却」と「SDxによる新価値創造」、そしてJASA会員企業の持続的成長を実現するための実行組織として機能します。単なる調査研究にとどまらず、実証プロジェクトの推進、ガイドラインの策定、会員企業への直接的な支援など、実践的な活動を展開します。

委員会の設計においては、いくつかの重要な原則を採用しています。第一に、オープンで包摂的な参加体制です。これらの委員会は、JASA会員企業のみならず、メーカー、ベンダー、大学、国の機関など、業界の垣根を超えた広い視野を持つメンバーで組織します。特定の企業の利益を目的としない、フラットで柔軟な業界団体としてのメリットを最大限に活かし、日本産業全体の発展に貢献する活動を推進します。大企業と中小企業、老舗企業とスタートアップ、発注側と受注側といった立場の違いを超えて、共通の課題に向き合う場を提供することで、従来の業界構造では生まれなかった革新的なアイデアや協業関係が創出されることを期待します。

第二に、委員会間の連携と相乗効果の創出です。4つの委員会は独立して活動するのではなく、密接に連携します。たとえば、AI支援型開発委員会で開発されたツールやプロセスは、SDxプラットフォーム委員会のプラットフォーム開発に適用されます。ナレッジサービス委員会で体系化されたビジネスモデルは、企画型人材育成委員会の教育コンテンツとして活用されます。このような横断的な連携により、技術、ビジネスモデル、人材育成という異なる側面からの取り組みを統合的に進め、相乗効果を生み出します。

第三に、段階的かつ反復的なアプローチです。最初から完璧な成果を目指すのではなく、小規模な実証実験から始め、得られた知見をもとに改善を重ねながら、徐々に規模を拡大していきます。失敗を許容し、そこから学ぶ文化を醸成することで、リスクを抑制しながら革新的な取り組みに挑戦できる環境を整えます。

AI支援型開発委員会:生産性革命から価値革命へ

第一の委員会は、AI支援型開発委員会です。この委員会は、AI技術の戦略的活用による生産性の飛躍的向上と、高付加価値化を実現することを目的とします。単にAIツールを使えるようにするという表面的な目標ではなく、AI活用によって創出された時間を、エンジニアの専門性向上に投資し、最終的には人月単価の引き上げと収益向上につなげるという、包括的な変革プロセスを確立します。

この委員会の活動は、三つの段階で展開されます。第一段階は、AI技術の実践的習得です。生成AIによるコード生成、設計支援、ドキュメント作成、テスト自動化など、開発プロセスの各段階でAIをどのように活用できるかを、実際のプロジェクトを通じて学びます。重要なのは、単なるツールの操作方法ではなく、AIの得意な領域と不得意な領域を見極め、人間とAIの最適な役割分担を設計する能力です。たとえば、定型的なコード生成はAIに任せ、システム全体のアーキテクチャ設計やトレードオフの判断は人間が行うといった、実践的な協働パターンを確立します。

第二段階は、創出された時間の戦略的活用です。AIによって開発効率が30パーセントから50パーセント向上したとして、その時間をどう使うかが決定的に重要です。単に労働時間を削減するのではなく、顧客のドメイン知識の深化、新技術の習得、品質向上活動、技術的負債の解消など、長期的な競争力強化につながる活動に投資します。この委員会では、時間投資の優先順位付け、効果測定の方法、組織的な支援体制の構築など、創出された時間を真の価値に変換するためのフレームワークを開発します。

第三段階は、高付加価値化とビジネスモデル転換です。エンジニアの専門性が向上し、より複雑な課題を解決できるようになれば、人月単価の引き上げや、成果報酬型の契約への移行が可能になります。この委員会では、技術力の向上を具体的な価値として顧客に説明する方法、新しい価格体系の設計、契約形態の工夫など、ビジネスモデル転換の実践的な手法を研究し共有します。

委員会の活動には、実証プロジェクトも含まれます。複数の会員企業が参加するパイロットプロジェクトを立ち上げ、AI支援開発の効果を定量的に測定します。開発生産性の向上率、品質指標の変化、エンジニアの満足度、顧客の評価など、多面的な評価を行い、成功要因と課題を分析します。これらの知見は、ベストプラクティスガイドとして体系化され、他の会員企業が同様の取り組みを始める際の参考となります。

さらに、AI技術は急速に進化しているため、最新動向の継続的なキャッチアップも重要な活動です。海外の先進事例の調査、新しいAIモデルやツールの評価、学術研究の産業応用の検討など、常に最新の知見を取り込み、会員企業に提供します。

SDxプラットフォーム委員会:共通基盤で競争力を結集

第二の委員会は、SDxプラットフォーム委員会です。この委員会は、日本の製造業がグローバル競争に打ち勝つための、共通SDxプラットフォームの構築を主導します。個社では負担しきれない大規模な開発投資を業界全体で分散し、深刻化する人材不足に対応します。同時に、業界標準化と相互運用性を確保することで、海外の巨大プラットフォームに対抗し得る、強固で信頼性の高い(ディペンダブルな)開発基盤の構築を目指します。

共通プラットフォームの開発は、単なる技術プロジェクトではなく、業界の構造改革を伴う戦略的な取り組みです。従来、各企業は独自のソフトウェアプラットフォームを開発し、それを競争優位の源泉としてきました。しかし、ソフトウェアの規模と複雑さが増大し、クラウドからエッジまでの統合が求められる現在、すべてを自社開発することは現実的ではありません。差別化につながらない基盤部分は共通化し、各社はその上で独自の価値を創出するという、新しい競争のあり方への転換が必要です。

この委員会の最初の重要な活動は、共通化の範囲と方針の策定です。自動車、ロボティクス、医療機器、産業機械など、各産業分野の特性を分析し、どの部分を共通化すべきか、どの部分は産業固有のカスタマイズを許容すべきかを明確にします。たとえば、通信プロトコルの実装、セキュリティ機能、ログ管理、OTA更新機能などは、多くの産業で共通のニーズがあります。一方、リアルタイム制御のアルゴリズムや、産業固有の安全規格への対応は、各産業の特性に応じたカスタマイズが必要です。この境界線を適切に引くことが、プラットフォームの成功を左右します。

次に、参照アーキテクチャの設計を行います。ハードウェア抽象化層、リアルタイムOS層、ミドルウェア層、アプリケーションフレームワーク層という階層構造を定義し、各層で提供すべき機能、インターフェース仕様、性能要件などを策定します。この設計では、既存のオープンソースプロジェクトとの整合性も考慮します。ゼロから独自開発するのではなく、LinuxカーネルやROS(Robot Operating System)、Autosarといった実績のあるOSSを基盤とし、日本の産業界特有のニーズに対応する部分を追加開発するアプローチを取ります。

実装フェーズでは、優先順位の高い機能から段階的に開発を進めます。最初は、基本的な通信機能とデータ収集機能を持つ最小限のプラットフォームを構築し、実際のデバイスで動作検証を行います。この初期版をパイロット企業に提供し、フィードバックを収集します。そのフィードバックをもとに改善を加え、機能を拡張していくという、アジャイル的な開発プロセスを採用します。一度に完璧なものを作ろうとするのではなく、小さく始めて反復的に改善することで、リスクを抑制しながら実用的なプラットフォームを構築します。

この委員会では、品質保証とセキュリティも重要な検討事項です。共通プラットフォームに脆弱性があれば、それを使用するすべての製品に影響が及びます。したがって、厳格なセキュリティレビュー、継続的な脆弱性診断、迅速なパッチ提供体制など、高い品質基準を維持する仕組みが必要です。ISO 26262やIEC 62304といった安全規格への適合性も、設計段階から考慮します。

さらに、エコシステムの形成も委員会の重要な役割です。プラットフォームが広く採用されるためには、開発ツール、テストフレームワーク、ドキュメント、トレーニング教材など、周辺環境の整備が不可欠です。また、サードパーティによるコンポーネント開発を促進し、プラットフォーム上で動作する多様なソリューションが提供される状況を作り出します。JASAは、このエコシステムの中核として、技術仕様の管理、適合性認証、開発者コミュニティの運営などを担います。

ナレッジサービス委員会:知的資本への転換を支援

第三の委員会は、ナレッジサービス委員会です。この委員会は、SIベンダーのビジネスモデルを「人月」から「知」へと転換するチャレンジを支援します。ベンダー各社が持つセグメントナレッジ(専門知識やノウハウ)をIP(知的財産)化し、「ナレッジ提供型サービス」として収益化するビジネスモデルへの転換を推進します。

この委員会の活動の核心は、暗黙知を形式知に変換し、それを再利用可能な知的資産として体系化するプロセスの確立です。従来、熟練エンジニアが長年の経験を通じて蓄積してきた知識は、その多くが個人の頭の中にあり、組織として活用することが困難でした。しかし、生成AI時代においては、この状況が根本的に変わります。生成AIとの対話を通じて問題を解決するプロセスそのものが、自然な形で文書化されるからです。効果的なプロンプト、生成されたコード、それに対するレビュー、修正プロセス、これらすべてが形式知として記録されます。

この委員会では、こうして蓄積された知識を、どのように体系的に整理し、検索可能にし、適切に分類するかという情報設計の方法論を開発します。たとえば、特定のドメイン要件を技術的に実現するためのパターン集を構築します。自動車の機能安全要件ASIL-Dを満たすソフトウェアーキテクチャのパターン、医療機器におけるOTAアップデートの実装方法と規制対応、産業用ロボットのリアルタイム制御とクラウド連携を両立させる設計手法といった、ドメイン知識とソフトウェアエンジニアリングを統合した知見をデータベース化します。

知的財産化されたナレッジを、どのように価格設定し、どのような契約形態で提供するかも、この委員会の重要な研究課題です。従来の労働時間ベースの価格設定から、提供する価値や成果に基づく価格設定への移行は、発注側と受注側の双方にとって大きな意識改革を伴います。価値をどのように定量化するか、顧客とどのように合意形成するか、リスクをどのように分担するかといった、実務的な課題を一つずつ解決していく必要があります。

この委員会では、具体的な契約書のひな型を開発します。たとえば、基本契約で最低限の保守・運用費用を確保し、効率化や売上向上の効果に応じた成果報酬を上乗せする契約形態や、ナレッジへのアクセスをサブスクリプション形式で提供する契約形態など、多様なモデルを設計します。それぞれの契約形態について、リスク配分、収益認識のタイミング、知的財産権の扱い、紛争解決のメカニズムなどを詳細に規定し、法務的な検証も行います。

さらに、ビジネスモデル転換に成功した企業の事例研究も重要な活動です。どのような順序で変革を進めたか、既存顧客との関係をどのように再構築したか、社内の評価制度や報酬体系をどのように変更したか、といった実践的な知見を収集します。成功事例だけでなく、失敗事例や困難に直面した際の対応も含めて、包括的な事例データベースを構築します。会員企業は、自社の状況に近い事例を参照することで、変革への具体的な道筋を描きやすくなります。

この委員会の活動は、AI支援型開発委員会とも密接に連携します。AIによって創出された時間を、ナレッジの形式知化とIP化に投資することで、人月ビジネスからナレッジビジネスへの転換を加速できます。また、SDxプラットフォーム委員会とも連携し、共通プラットフォーム上で提供するナレッジサービスの形態を検討します。

企画型人材育成委員会:次世代リーダーの養成

第四の委員会は、企画型人材育成委員会です。この委員会は、次世代の経営層および事業リーダーの育成を推進します。技術の詳細を理解しながらも、それをビジネス価値に変換できる視点を持ち、経営層と技術者の橋渡しができる「企画型人材」が、AI時代の組込みソフトウェア業界には不可欠です。

企画型人材とは、単なる技術マネージャーでもなく、単なる経営企画担当でもありません。それは、技術トレンドを深く理解し、それが自社のビジネスにどのような影響を与えるかを分析し、具体的な戦略と実行計画に落とし込める人材です。たとえば、生成AIの技術動向を理解し、それを自社の開発プロセスにどう適用すれば競争力が高まるかを企画し、経営層に投資を提案し、技術者を説得して実行に移し、成果を測定して改善するという、一連のプロセスを主導できる人材です。

この委員会の教育プログラムは、従来の座学中心の研修とは異なるアプローチを採用します。経済産業省「PIVOT」戦略やIPA「ソフトウェアモダナイゼーション委員会」など、国や業界有識者の発信する情報を深く考察することから始めますが、それは単なる知識の習得ではありません。これらの政策提言が、なぜ今このタイミングで出されているのか、グローバルな競争環境の変化とどう関係しているのか、自社にとってどのような意味を持つのかを、徹底的に議論します。

プログラムの中核は、企業の枠を超えた参加者によるワークショップです。異なる企業、異なる産業、異なる立場(メーカー、ベンダー、スタートアップ、大学など)の参加者が集まり、共通の課題に対するソリューションを共同で企画します。たとえば、「日本の組込みソフトウェア業界が、2030年にグローバル市場で競争力を維持するための戦略」というテーマで、参加者がチームを組み、市場分析、技術トレンド予測、競合分析、自社のポジショニング、具体的なアクションプランまでを策定します。

この過程で重要なのは、技術とビジネスの両面から俯瞰する視点を養うことです。技術的に優れたソリューションであっても、ビジネスとして成立しなければ意味がありません。逆に、ビジネスモデルとして魅力的であっても、技術的な実現可能性がなければ絵に描いた餅です。参加者は、技術的制約とビジネス要求のトレードオフを理解し、現実的かつ革新的な解決策を見出す訓練を積みます。

プログラムには、経営層との直接対話の機会も組み込まれています。参加者が策定した企画を、実際の経営者やベテラン役員にプレゼンテーションし、フィードバックを受けます。経営判断の視点、リスク評価の考え方、ステークホルダーとの調整方法など、実務的な知見を直接学ぶことができます。また、成功した企業変革の事例について、その変革を主導した経営者やプロジェクトリーダーから直接話を聞く機会も設けます。

さらに、この委員会では、参加者のネットワーク形成も重視します。企画型人材として活動する際には、社内外の多様なステークホルダーを巻き込み、協力を得る必要があります。このプログラムを通じて構築された参加者同士のネットワークは、将来的な協業の基盤となります。異なる企業の企画型人材が連携することで、業界横断的なプロジェクトの推進や、ベストプラクティスの共有が促進されます。

この委員会の活動は、他の3つの委員会すべてと連携します。AI支援型開発委員会、SDxプラットフォーム委員会、ナレッジサービス委員会で得られた知見は、企画型人材育成の教材として活用されます。逆に、この委員会で育成された人材は、他の委員会の活動に参加し、実践的なプロジェクトをリードします。このような循環により、理論と実践が統合された、真に実効性のある人材育成が実現されます。

活動の可視化と業界への波及

4つの委員会の活動内容および成果は、JASAホームページ等を通じて積極的に公開し、本提言レポートの認知度向上と、業界全体を巻き込んだ活動の活性化につなげていきます。情報公開

は、単なる活動報告にとどまらず、業界全体の変革を促進するための戦略的な取り組みとして位置づけられます。

公開する情報には、複数の層があります。第一に、委員会での議論内容です。どのような課題が議論され、どのような解決策が提案され、どのような判断基準で優先順位が決定されたかというプロセスを、可能な限り透明化します。これにより、JASA会員企業だけでなく、業界全体が変革の方向性を共有できます。特に、失敗した試みや、期待した成果が得られなかつた実験についても率直に公開することで、他の企業が同じ過ちを繰り返さないように支援します。

第二に、策定したガイドラインや参照アーキテクチャです。技術的な仕様書、ベストプラクティス集、契約書のひな型、チェックリストなど、実務に直接活用できる成果物を公開します。これらは、JASA会員企業が優先的に利用できますが、業界全体への波及を考慮し、一定期間後には広く公開することも検討します。オープンな情報共有により、日本の組込みソフトウェア業界全体の底上げを図ります。

第三に、実証プロジェクトの結果です。各委員会が実施するパイロットプロジェクトや実証実験の成果を、定量的なデータとともに公開します。たとえば、AI支援開発によって開發生産性が何パーセント向上したか、品質指標がどう変化したか、エンジニアの満足度やスキルレベルがどう改善したかといった具体的なデータは、他の企業が同様の取り組みを始める際の重要な参考情報となります。ただし、個別企業の機密情報には十分配慮し、匿名化や集約化を適切に行います。

情報公開の形態も多様化します。従来の報告書やホワイトペーパーに加えて、ウェビナー、技術ブログ、ポッドキャスト、YouTubeチャンネルなど、さまざまなメディアを活用します。特に、若い世代のエンジニアにリーチするためには、テキストだけでなく動画やインタラクティブなコンテンツが効果的です。委員会メンバーや実証プロジェクトに参加した企業の担当者が直接語る形式にすることで、リアルな実践知を伝えます。

また、年次での活動報告と、本レポート自体のアップデートを継続的に行うことで、変化する技術環境と市場動向に対応した提言を提供し続けます。2025年版の本レポートは出発点に過ぎず、毎年新しい章を追加し、既存の章を改訂し、最新の事例と知見を反映させていきます。このような継続的な更新により、レポートは常に業界の現状を反映した、実用的な指針であり続けます。

JASAは、一度きりの提言ではなく、継続的な伴走者として、日本の組込みソフトウェア業界の変革を支援していきます。委員会活動を通じて得られた知見は、新たな課題や機会の発見につながり、それが次の活動計画に反映されます。このような継続的な改善のサイクルを回すことで、業界全体の進化を加速させます。

さらに、国内だけでなく、国際的な情報発信も強化します。日本の組込みソフトウェア業界の取り組みを、英語で発信することで、海外の企業や研究機関との連携の機会を広げます。また、海外の先進事例を国内に紹介する逆の情報流通も促進します。グローバルな知識の循環に積極的に参加することで、日本の組込みソフトウェア業界の国際的な存在感を高めます。

会員企業に対しては、定期的なフィードバックの機会を設けます。四半期ごとの報告会、年次総会、テーマ別のワークショップなどを通じて、委員会活動の進捗を共有し、会員企業からの要望や提案を収集します。JASAの活動は、会員企業のニーズに応えるものでなければ意味がありません。常に現場の声に耳を傾け、実効性の高い支援を提供できるよう、双方向のコミュニケーションを重視します。

最後に、これらの活動の究極的な目標は、個別企業の競争力強化にとどまらず、日本の組込みソフトウェア業界全体のエコシステムを強化し、グローバル市場での競争力を高めることです。JASAは業界団体として、個社の利益を超えた視点から、業界全体の持続的な発展を追求します。AIとSDxが主導する新しい時代において、日本の組込み技術の強みを活かしながら、新しい価値創造の形を確立する。その実現に向けて、JASAは会員企業とともに、着実に歩みを進めていきます。

おわりに

変革の時代に立ち向かう

最近の技術進化は著しく、まさに激動の時代を迎えています。とくにSDx(Software-Defined Everything)による製品価値の創出は、オープンで迅速なソフトウェア開発を通して実現されるようになりました。これは、従来の組込みソフトウェアの十八番であった、クローズドで効率・品質優先の開発から大きく変化しています。垂直統合による緻密な最適化と品質保証で差別化を図ってきた日本の組込みソフトウェア業界にとって、この変化は単なる技術トレンドの変化ではなく、ビジネスモデルそのものの根本的な見直しを迫るものです。

また、生成AIによるソフトウェア開発の生産性向上は、私たちの想像を遥かに超えています。コード生成、設計支援、テスト自動化、ドキュメント作成—開発プロセスの至るところでAIが人間の能力を拡張し、あるいは代替しつつあります。この変化の速度は加速度的であり、この先、ソフトウェアエンジニアの仕事が従来の形で存在し続けるのかという岐路に立たされているといつても過言ではありません。

この間も、さまざまな技術進化に合わせてソフトウェア開発のやり方およびビジネスモデルについては数多くの提言・批判がなされてきました。2025年問題、DXレポート、デジタル庁の発足、経済産業省による「PIVOT」戦略—政府や業界団体から発信される警鐘は絶えることがありませんでした。しかし、ここにきて、いよいよ待ったなしとなった感が強くあります。経済産業省が公表した「デジタルエコノミーレポート」では、日本のGDPに占めるデジタル経済の割合が主要国と比較して低く、2030年に向けた本格的な構造転換が急務であることが示されています。もはや、将来的課題ではなく、今この瞬間に取り組むべき喫緊の課題となっています。

JASA改革プロジェクトの立ち上げ

私の会社もそうですし、それよりもさらに私の会社が所属しているJASA自体、みな同じこの問題に向き合うことになります。すでに自分自身、この数年間、個社としてこの課題に向き合ってきました。生成AIツールの導入、アジャイル開発手法の実践、ビジネスモデルの転換—試行錯誤を重ねながら、変革の道を模索してきました。しかし、実際に取り組んでみて痛感したのは、これはもはや個社レベルでやるべき内容ではないということです。

AI時代の変革は、技術の問題だけではありません。それは、人材育成、組織文化、ビジネスモデル、業界構造、さらには社会システム全体に関わる包括的な課題です。一社単独で対応できる範囲を超えて、特に、共通プラットフォームの構築、標準化の推進、人材の流動性確保、新しい取引慣行の確立といった課題は、業界全体で協力しなければ実現できません。

この揺れ動く、危機的状況の中で、業界団体として大きなメッセージを出すべきと考えました。個社の利害を超えて、業界全体の未来を見据えた変革の方向性を示す必要があります。そのような話を周囲のメンバーとしている中で立ち上がったのがJASA改革プロジェクトであり、その中で、この考えを実現すべく活動した結果がこのJASAレポートです。

レポート作成の歩み

2025年の初頭から、われわれなりの仮説をまとめ始めました。組込みソフトウェア業界が直面している課題は何か、SDxと生成AIがもたらす変化の本質は何か、日本の強みを活かしながら競争力を維持・向上させるための道筋はどこにあるのか—多くの問い合わせに対して、データ収集、分析、議論を重ねました。

さらにレポートをまとめると、ぜひこの方に話を伺わなければならぬという方々へのヒアリングを実施しました。マイクロソフトの田丸様には、生成AI全般の今後の展望について、最前線で技術開発とビジネス展開に携わる立場から、実践的かつ先進的な知見を共有していただきました。特に、企業がAIをどのように活用し、組織変革を進めるべきかについて、具体的な事例とともに貴重なアドバイスをいただきました。

山下技術開発事務所の山下様には、生成AIの活用がソフトウェア開発をどのように変化させていくのか、そしてその変化の中で私たちが取るべき重要な行動は何かについて、示唆に富んだお話を伺いました。特にAIの民主化がもたらす可能性と、それに向けて今から準備すべきことについてのご意見は、私たちの思考の枠を大きく広げてくれました。

日経ロボティクス編集長の進藤様には、本レポートで提示した技術的な仮説や内容の記述について、ロボティクス産業の専門家としての立場から率直かつ厳しいレビューをいただきました。特に、われわれが描いた技術トレンドや市場展望に対して、産業界の現実や技術的な実現可能性の観点から具体的な指摘と改善提案をいただいたことで、レポート全体の精度と説得力が大きく向上しました。専門家による忌憚のない批評的検討を経たことで、本レポートはより現実的で実践的な内容になったと確信しています。

これらのヒアリングを通じて、当初の仮説を大いにアップデートすることができました。さらに多くの方の意見を伺いながら、何度も書き直し、議論を重ね、なんとかここまでたどり着きました。

協力者への感謝

JASA改革プロジェクトメンバーにも、大いに助けられました。山田専務理事と鴨林理事は、このプロジェクト設立時から一緒に活動を続け、本レポートの完成に向けてともに活動してきました。困難な局面でも前向きな姿勢を失わず、常に建設的な議論を重ねてくれたことに、心から感謝しています。大いに勇気づけられました。

また、権藤理事には副リーダーとして、特にPIVOTレポートからのつながりやSDxに対する深い知見など、有効な意見をいただきました。経済産業省の政策動向と業界の実態を橋渡しする視点は、本レポートの実効性を高める上で不可欠でした。

そして竹内会長には、プロジェクトの立ち上げから今日に至るまで、温かく見守り、時には的確な助言をいただき、会長としての強力なリーダーシップのもと、私たちは安心して活動を進めることができました。深く感謝申し上げます。

その他にも、多くの会員企業の方々、有識者の方々、関係団体の方々から、貴重なフィードバックをいただきました。ここに記しきれないほど多くの方々の協力があって、このレポートは完成しました。改めて、深く感謝申し上げます。

継続的なアップデートと実践への期待

これからは、ぜひこのレポートの内容を年々アップデートしながら、われわれの活動指針として役立てることができれば幸いです。技術は進化し続け、市場環境は刻々と変化します。2025年に正しかった戦略が、2026年、2027年にも通用するとは限りません。だからこそ、継続的な見直しと改善が必要です。

ここで重要なことを明確にしておきたいと思います。今回のレポートで提言した内容は、現在の時点でベストと思われる仮説であり、必ずしも正解ではない可能性があります。AI時代の変革は前例のない領域であり、誰も確実な答えを持っていません。しかし、この状況の中で、必ず正しいものをやるという姿勢だけでは先に進むことができません。

たとえ間違った可能性があっても、その時点でベストと思われる仮説を立て、それを実行し、結果を検証するという仮説検証のアプローチで進むべきです。完璧な戦略を待ち続けていては、変化の激しい技術環境に取り残されてしまいます。「完璧な地図を待つより、不完全でも羅針盤を持って一步を踏み出す」—これが、技術革新の波の中で生き抜くための基本的な姿勢だと考えています。

JASAは、このレポートを単なる調査報告書として終わらせるのではなく、生きた指針として進化させていく覚悟です。会員企業からのフィードバック、技術動向の変化、実践の成果と失敗—これらすべてを取り込みながら、毎年内容を更新し、より実効性の高いものにしていきます。

そして何より重要なのは、実践です。ここに書かれている内容について各社で実践をし、その知見を業界全体で共有しながらともに前進していくことを強く望みます。成功事例も失敗事例も、すべてが業界の共有財産です。オープンに情報を交換し、互いに学び合うエコシステムを構築することが、AI時代を生き抜く鍵となります。

未来への決意

機械や建築のように1000年単位の歴史を通じてエンジニアリングを確立してきた分野とは異なり、ソフトウェアは100年程度の短い期間でエンジニアリングを進めてきました。さらに最近の変化を見ればわかるように、四半期単位、月単位、時には週単位で技術が変わっています。こうした中でエンジニアリングを確立することは極めて困難です。しかし、ソフトウェアが社会の基盤となる現状をとらえると、その難しい課題を解決することは必須です。

エンジニアリングとして最低限実現しなければならないことは、再現性、検証性の確立であり、さらには相互運用性やモデリング&シミュレーションができます。残念ながら、属人的な開発や、オフィスソフトウェアで設計している世界では、これらを実現することは難しい状況です。生成AIと近代的な開発ツールの活用により、ようやくこれらの実現可能性が見えてきました。

また、人月からの脱却は、長い間言われてきましたが、まだ十分には実現していません。社会の多くのものが、その製品やサービスに対していくら払ってもよいという価値をベースに取引をしているのに、ソフトウェアではいまだに、何人で作りましたという原価をベースとして価格が形成されています。

一方、ソフトウェアもアプリケーションやサブスクリプションによって、価値によって評価される兆候が見え始めています。AI活用や組み立て産業化など、サービス実現方法が変わってきている中で、やっと、特殊な業界慣行から抜け出せそうです。本レポートで提示した「AI駆動」「IP駆動」「ストック駆動」の3ステップは、まさにこの転換を実現するための具体的な道筋です。

AI、クラウドサービス、組み立て産業化、デジタルツイン、さらにはグローバルマーケット化など、社会環境が大きく変化しています。しかも人口減少などの社会課題が限界に達している今こそ、ソフトウェア業界全体でエンジニアリングの本質を見つめなおし、解決に向けた変革が求められています。

組織の変革も重要ですが、人材流動化も進んできています。人材自らが環境変化に合わせてスキルをピボットして変革を図っていくことも可能です。組織の変革に先駆け、人材の変革が始まるかもしれません。組織も人材もこれからは選ばれるという意識が求められてきます。そのために、環境変化を迅速にとらえ変革を図っていく必要があります。

2030年に向けた第一歩として

本レポートは、広く国内外の状況を分析し、議論を重ねてきたものです。グローバルに競争力を持つユーザー企業、ベンダー企業、そして個々のエンジニアにとって、本レポートが2030年に向けた変革の第一歩になれば幸いです。

組込みソフトウェア業界は、これまで数多くの技術的挑戦を乗り越えてきました。マイコンの黎明期、インターネットの普及、モバイル革命、そしてIoTの時代—その都度、業界は適応し、進化し、新たな価値を創造してきました。AI時代の変革も、この延長線上にあります。しかし、今回の変革は過去のどれよりも大規模で根本的です。

だからこそ、JASAは単なる業界団体を超えて、変革の触媒として機能する必要があります。会員企業間の知識共有、産学官連携の推進、国際的なパートナーシップの構築、人材育成プログラムの提供—JASAが果たすべき役割は多岐にわたります。

日本の組込み技術の深さと品質は、AI時代においてこそ真価を発揮します。安全性、信頼性、効率性への徹底的なこだわり、現場での地道な改善活動、顧客との長期的な信頼関係—これらの強みを、AIという新しいツールと組み合わせることで、世界に類を見ない価値を創造できるはずです。

「あきらめない」精神と継続的な努力により、必ず道は開けます。技術の進歩は社会の豊かさと人間の幸福につながり、JASAとその会員企業はこの変革を主導できる力を持っています。

変革は既に始まっています。待っている時間はありません。今こそ、共に行動を起こす時です。AI時代の組込みソフトウェア業界の未来は、私たち一人ひとりの手にかかります。JASAという共通のプラットフォームを通じて、個社の枠を超えた価値創造に挑戦し、世界に誇れる技術とソリューションを生み出していきましょう。

技術革新の波の中で、完璧な答えはありません。しかし、確実な道筋がないからこそ、仮説を立て、実行し、学習し、改善するという姿勢が重要です。共に新しい道を探し、共に未来を創造していきましょう。

2025年11月

一般社団法人組込みシステム技術協会(JASA)副会長
JASA改革プロジェクトリーダー
渡辺 博之

執筆・編集者

執筆

渡辺 博之

一般社団法人組込みシステム技術協会(JASA)副会長
JASA改革プロジェクトリーダー
株式会社エクスマーション 代表取締役社長

横浜国立大学卒業後、メーカー勤務を経て、1996年より組込み分野におけるオブジェクト指向技術の導入支援に従事。コンサルタントとしてFA装置や自動車、デジタル家電など多くの分野において現場支援や人材育成を手掛ける。2008年9月に(株)エクスマーションを設立し現在に至る。並行して、JASAにおけるETロボコンやUMTPにおける組込みモデリング部会をはじめとする様々なモデリング推進活動を通じた、組込みソフトウェア業界全体の技術底上げにも従事。現在、(独)情報処理推進機構(IPA)ソフトウェアモダナイゼーション委員、ETロボコン共同企画委員長。

JASA改革プロジェクトメンバー

権藤 正樹

一般社団法人組込みシステム技術協会(JASA)理事
JASA改革プロジェクトサブリーダー
イーソル株式会社 代表取締役社長CEO兼CTO

1996年にeSOL入社。以来社OS及びツール関連の開発、それらを用いた車載、産業機器、家電機器などの各種システムプラットフォーム開発に取組む。近年はシングルコアからメニーコアまで対応したOSであるeMCOS、ドメイン知見と機械学習を組合せたドライバモデル eBRAD、AUTOSAR Adaptive Platform仕様策定アーキテクト、マルチコア向けアーキテクチャ記述仕様IEEE Std. 2804 SHIMのWG Chair、社内の開発プロセス含む技術インフラ、プロダクトマネジメントを推進、2022年に専務ソフトウェア事業部長、2025年より現職。他には組込みマルチコアコンソーシアム副会長、IEC TC91/WG13メンバ、早稲田大学アドバンスドマルチコアプロセッサ研究所招聘研究員、COOLChips TPC等。

鴨林 英雄

一般社団法人組込みシステム技術協会(JASA)理事
株式会社DTSインサイト 代表取締役常務

横河電機システム会社入社後、集中治療室監視システム、超音波診断装置の開発等に従事。1990年代の半導体成長期からは、横河デジタルコンピュータにて開発支援ツールと受託開発で「ものづくり日本」を根底から支え、顧客の総合開発パートナーとしての地位を確立する。現在、SDxと生成AIの潮流を捉え「人月ビジネスからの脱却」を牽引。JASA ET事業本部長として製造業の競争力を高めつつ、経営者として自動車(SDV)や医療分野で、機能安全規格に準拠した「開発プロセス」自体をソリューションとして提供。

山田 敏行

一般社団法人組込みシステム技術協会(JASA)専務理事

大手鉄鋼メーカーの研究所で電子機器開発に従事後、1991年に横河デジタルコンピュータ株式会社(現・株式会社DTSインサイト)に入社。インサーチットエミュレータをはじめとするマイコン開発環境のビジネスを担当。2014年に株式会社日新システムズに入社。スマートシティを構成する生活支援・交通・建設など、幅広い産業分野への事業展開を推進。JASAでは2009年から「ET & IoT Technology / Edge Tech+展示会」の実行委員長、カンファレンス委員長などを歴任。2025年より現職。
