

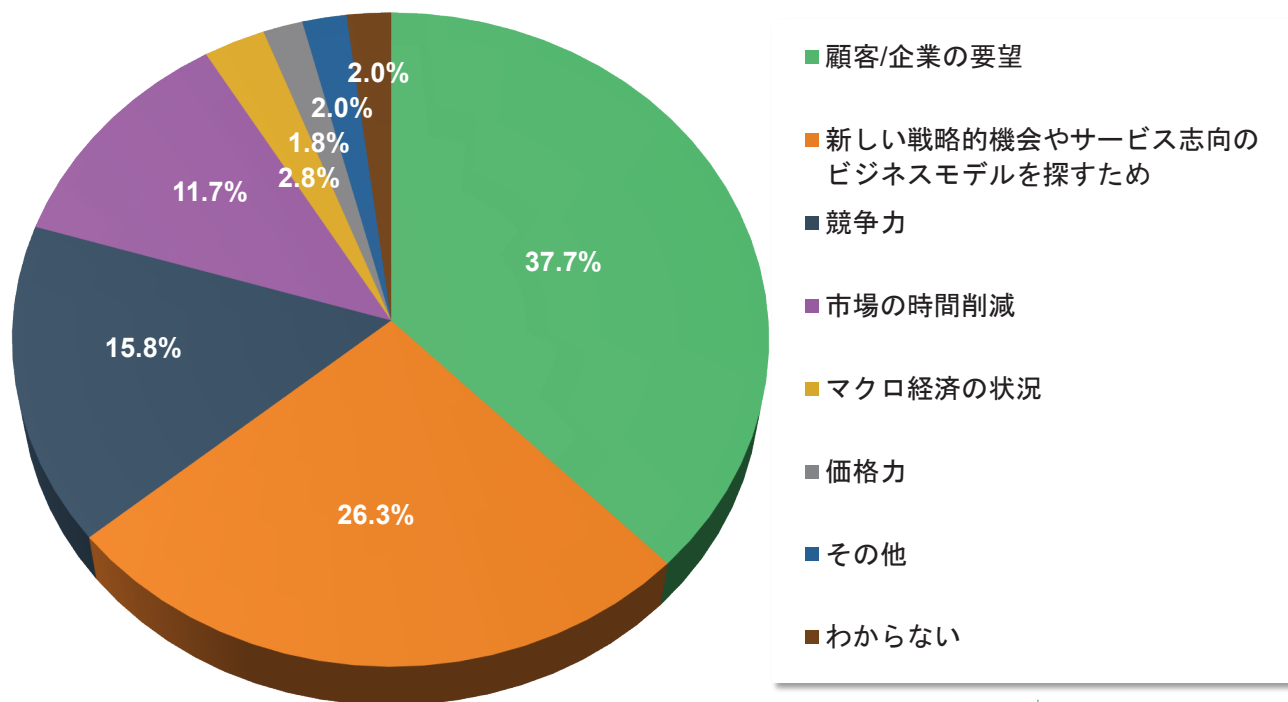
ソフトウェア開発言語の選択 ならびに費用対効果に関して Ada言語の優位性 調査報告



はじめに

IoTの影響については、多くのことが書かれています。新しい顧客要件、新しいビジネスモデル、競争圧力があり、多くのエンジニアリング組織内の改革を推し進めています。要件が変化することで、ミドルウェアの接続、リモート管理機能、機械視覚や環境発電（energy harvesting）などの技術への関心を高めるニーズが増大しています。

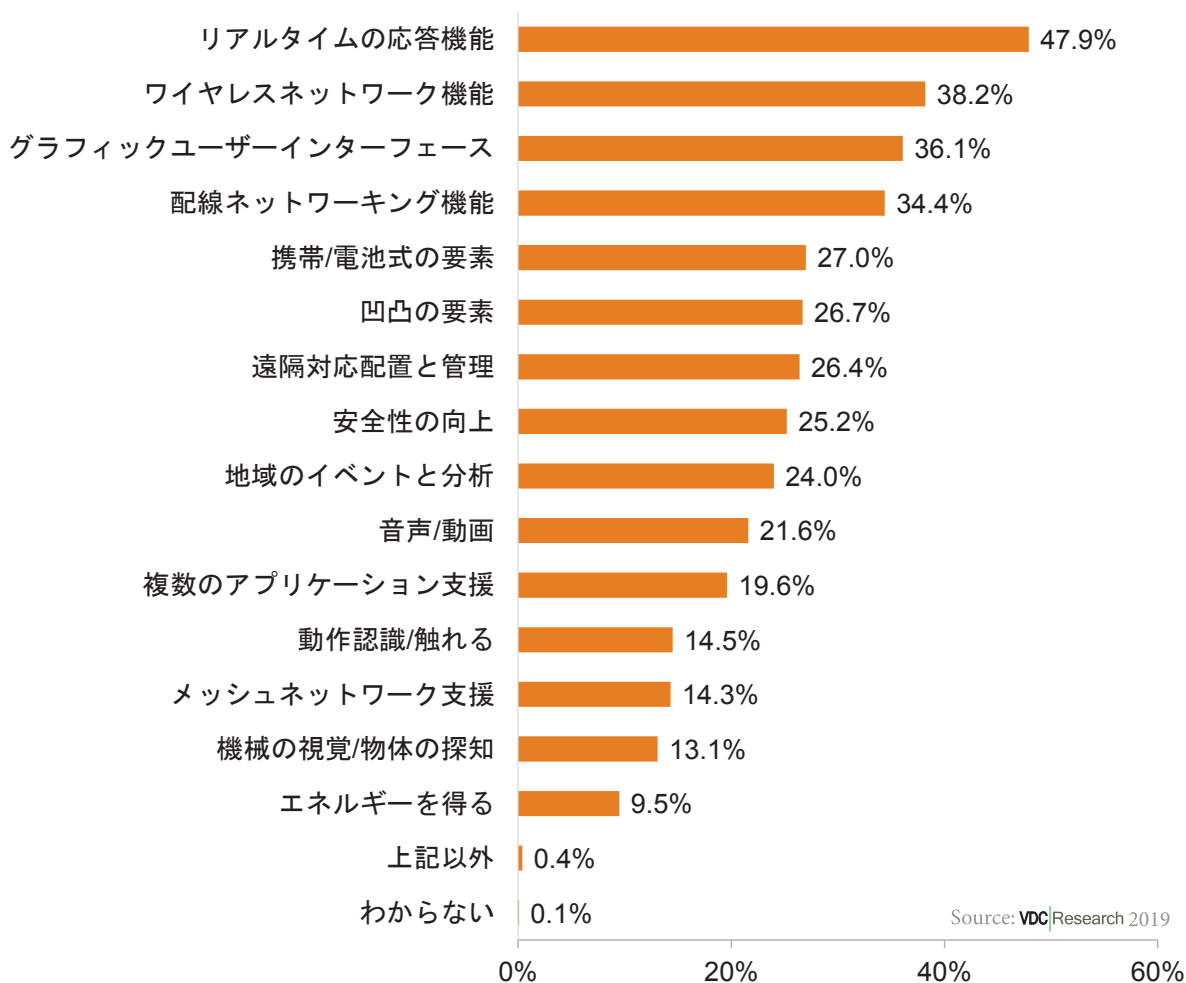
表 1: IoT機能を活用/配置している回答者の組織の第一要因
(回答者の割合)



Source: VDC Research 2019

IoTの周りには過度な期待や展望があり、コネクテッド組込みシステムは新しいものではありません。コネクテッドファクトリーと製造環境を介して多くの相乗効果を引き出すための取組みは何十年も前から存在しています。OnStarのようなテレマティクスとサービスを備えた自動車やトラックも同様に古い概念です。しかし、業界では、増大する企業のビジネスの方向性とビジネスモデルを接続するシステムとの組み合わせにおいて、新しい障壁に直面しています。この技術力とビジネス力の組み合わせは、自己完結サイクルを生み出し、新しいテクノロジーの大規模な採用と投資が生まれています。多くの組織やソリューションベンダーが、新しいIoT技術と機能要件に対応しているにもかかわらず、新たな目標によって従来の組込エンジニアリング要件に関連する課題は軽減されません。例えば、多くの組込み装置では、リアルタイムでセーフティクリティカルなアプリケーションに対応できる解決策へのニーズは衰えていません。実際、調査の回答者のほぼ半数は、現在のプロジェクトにリアルタイム応答の機能が含まれていることを回答しています。多くの場合、ユビキタス接続性、状況認識ならびに適応に期待すると、リアルタイムシステムやセーフティクリティカルが重要なシステムの障害に関連するリスクを増大させるだけです。

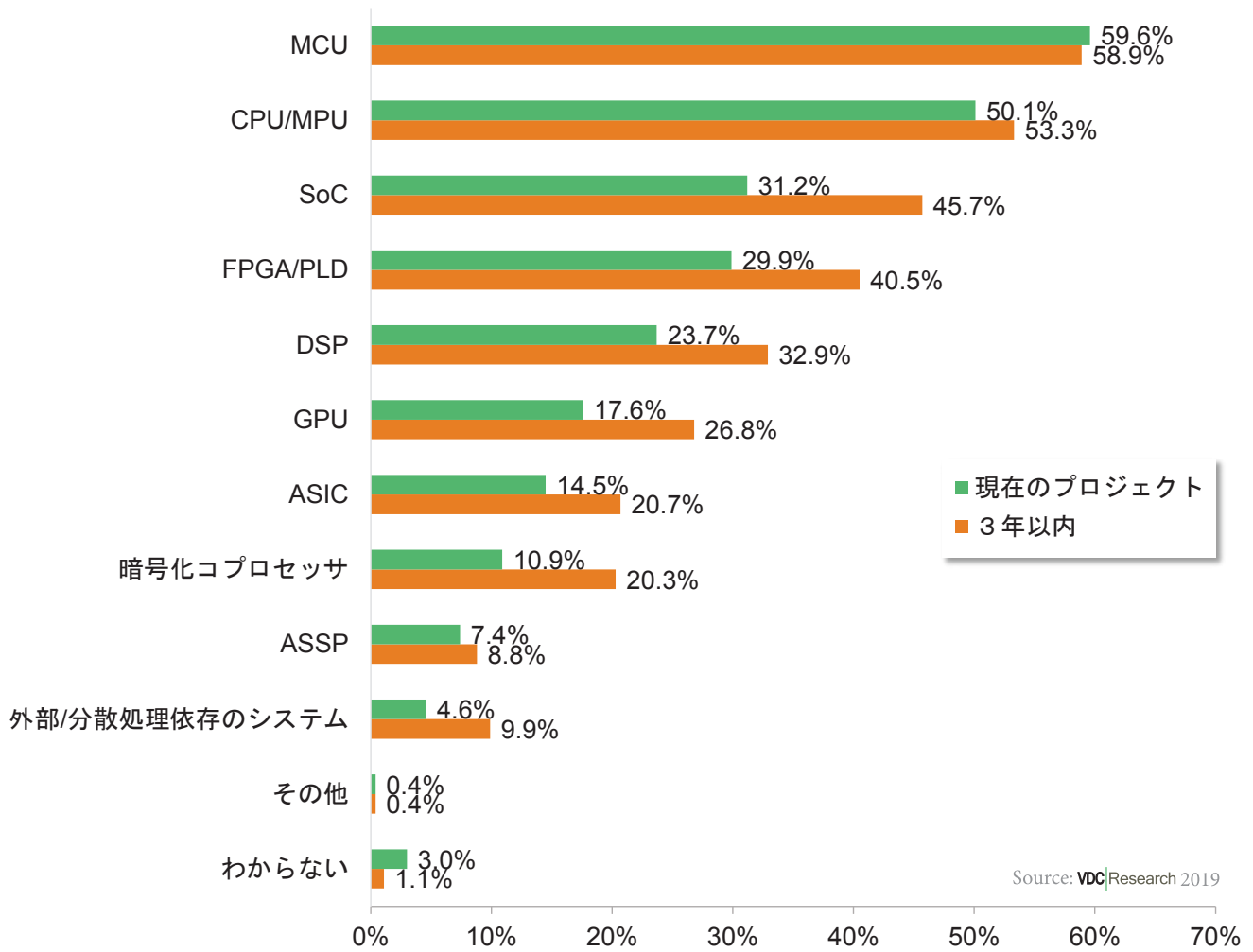
表 2: 現在のプロジェクトに含まれる機能や特徴
(回答者の割合)



注:複数回答により合計は100%以上になります

これらの機能の変更により、新しいハードウェアおよびソフトウェア製品の採用が進んでいます。たとえば、OEM 企業は、より堅牢な（多くの場合、メモリを必要とする）オペレーティングシステムとコンネクティビティスタックを統合しています。これはLinux OSの使用に関する調査への複数の回答です。さらに、新しい設計には、複雑なマルチプロセッサ/SoCが多く組み込まれています。5年前、私たちは、機能と電力性能を最適化するためにCPU + MCU（例えばビック、リトルなど）へ移行するのを見てきました。現在では、ソフトウェアやビジネス要件の複雑化に伴い、機械学習、視覚処理、セキュリティ要件などの新しいテクノロジーでこの変革が加速しています。今日の接続機器のコンピュータ集約型の負荷に対応するため、複雑なSoC、GPU、FPGA、および暗号プロセッサの活用が進んでいます（表3参照）。また、新しいテクノロジーの導入は、開発上の新たな問題を抱える可能性がでてきます。多くの技術変革と新たなビジネスゴールの組み合わせにより、諸問題はより複雑になっています。結果として、エンジニアリング組織は、現在の製品の機能ギャップを解消するだけでなく、新しい開発効率を向上するソリューションを模索する必要があります。

表 3: 現在のプロジェクトで使用しているプロセッサの種類
(回答者の割合)



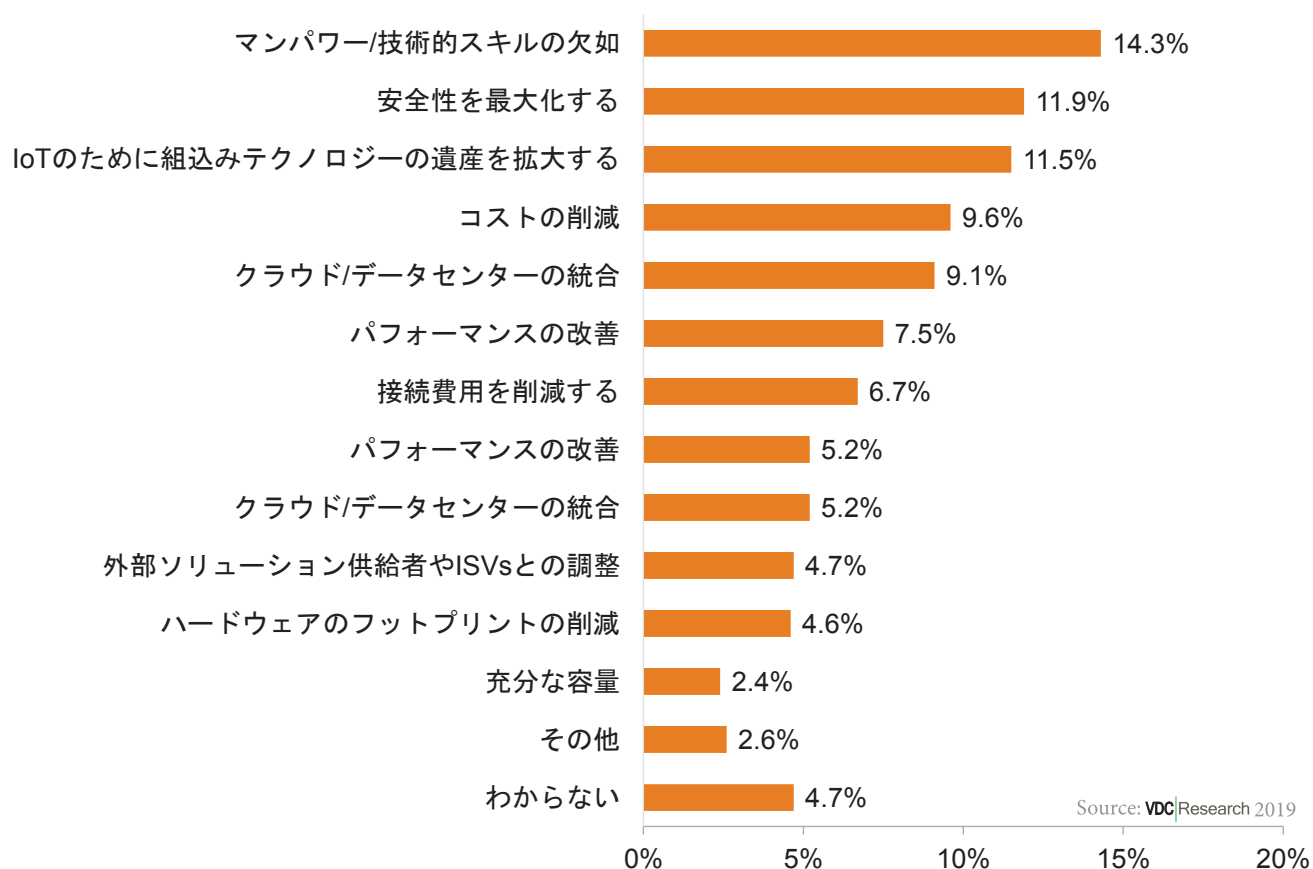
Source: VDC Research 2019

注:複数回答により合計は100%以上になります

経済発展に伴うIoTの再評価

IoTの進化に伴い多くの組織は市場にIoTソリューションを提供するためにリソースあるいは経験の少ない領域に参入することになりました。現在、IoT関連の接続の需要によって、「セキュリティの強化」「クラウド/IT統合の促進」などの新たな大きな課題(表4を参照)が存在しています。この変化、開発リソース、費用は依然としてIoTソリューションの最大の課題の一つです。新しいIoT特有の懸念は従来の問題点を悪化させました。

表4: IoTソリューション開発における組織にとっての課題
(回答者の割合)



組織が新たな開発と人手不足の課題に直面しているため、スケジュールと市場投入までの時間短縮などの課題が拡大してきました。プロジェクトスケジュールの遵守は、製品開発にとって大きな課題であり、プロジェクトの3分の1以上が遅延しているという報告があります。大きな遅延は運用コストを増大させるだけでなく、市場やセールスのチャンスを逸することになり売上に影響がでます。製品の本質的な価値が予測できれば、明らかに、顧客との信頼を獲得し、設備投資への可視性の向上から追加投資が可能となります。プロジェクト遅延率が比較的高いにもかかわらず、この測定基準に対するパフォーマンスは過去数年間で向上しています。この改善は確かに新しい開発技術と方法論からきています。ただし、IoTによるビジネスおよび開発要件の変更の規模により、現在のスケジュール遵守レベルを維持または改善するために新たな対応が必要になります。

VDC の総所有コスト計算機 (以降TCO 計算機)

VDC のIoT&Embedded Engineering Survey には、さまざまな業界の700人以上の回答があり、その結果は、TCO 計算機の作成に使用されました。

TCO 計算機は、次に示す様々な統計を計算に使用します。プロジェクトごとのデバイス、BOM (Bill of Material) コスト、開発コストの配分、エンジニア数、エンジニア一人当たりの労務費の平均、プロジェクトの長さ、フィールドに導入された製品の平均寿命、顧客が報告/要求する欠陥またはソフトウェアパッチの総数、各パッチまたは欠陥修復に必要なIT とエンジニアリングの期間 (時間) の合計、および動作不能になり修復が必要なデバイスの推定割合 または、毎年の交換台数などです。

比較コスト計算では、同じソフトウェア開発言語を使用する同じ垂直市場のプロジェクトの結果と、同じプロセッサアーキテクチャを使用する同じ垂直市場のプロジェクトの結果に重きを置いています。さらに、同じプロセッサアーキテクチャ、同じ言語、同じ垂直市場で使用するプロジェクトの結果も考慮に入れています。

IoT / 組込市場は、非常に複雑で異機種が混在しています。組込市場で必要とされる幅広い機能とフォームファクタは、特定のハードウェアおよびソフトウェアコンポーネントの選択せざるを得ない大きな要因になっています。さらに、テクノロジーの選択を正しく行わないと、現状の要件だけでなく過去のプロジェクトでの投資にも影響され、代用品を用いると新たな人件費やコストが発生する可能性があります。テクノロジーを選択時に実施する多くの評価検討において、決定者は、テクノロジーの潜在的な影響と費用対効果を考慮することが重要です。

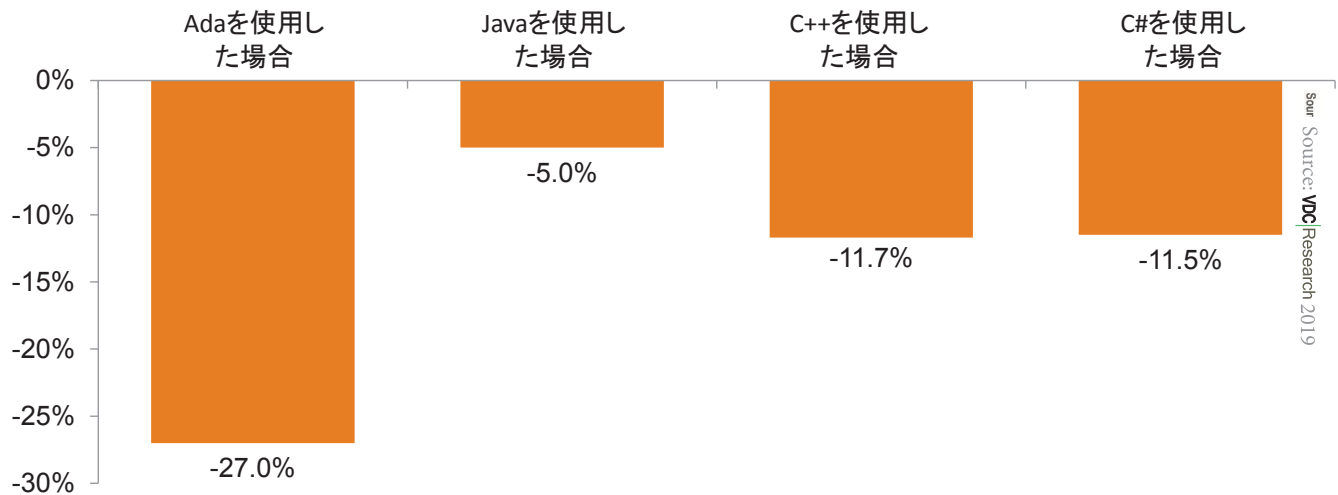
組織やプロジェクト特有の要件を超えて、同様のプロジェクトを比較した時に、特定の傾向や推測がでてきます。BOM (Bill of Material) コスト、開発コスト、運用、欠陥、修復率など、総所有コスト、TCO (Total Cost of Ownership) に影響を与える多くの費用があります。この統計の一部は、業界あるいは人口統計の影響を強く受けていますが、一部の技術選択肢はプロジェクト成果に大きな影響があると言えます。その一つはソフトウェア開発言語の選択でした。

プログラミング言語を選択する複数の要因 (メモリ容量、社内の専門性、従来のソフトウェア資産等) がありますが、多くのプロジェクトはTCO を考慮して言語を選択すべきです。私たちの調査によると、Java やC# ベースのプロジェクトを除き、ソフトウェアの欠陥率とコストは同様のプロジェクト間で一貫していることが分かりました、多くのソフトウェアの欠陥やパッチが出荷後に報告される傾向にありました。しかし、開発コストは、プロジェクトの期間、チームの規模、開発者のコストによって大きく変動し、結果は大きく変動します。技術 (ソリューション) 選択をする際に費用対効果の影響を効果的に測定するために、VDC はTCO 計算機を作成しました。最近の調査結果を見ると、Ada はさまざまなプロジェクトの成功の指標に大きな影響を与えた言語でした。

開発の全費用計算例

開発の総費用に言語選択が与える影響を説明するために、計算機のシナリオとその出力例をいくつか挙げました。最初の例は、x86 プロセッサを使用して10000 台製造し、ソフトウェア開発にはC が使用された通信/ネットワークプロジェクトを評価しています。この計算では、Ada は節約効果が大きく、ソフトウェア開発コストを27%削減できる可能性があることが分かりました。

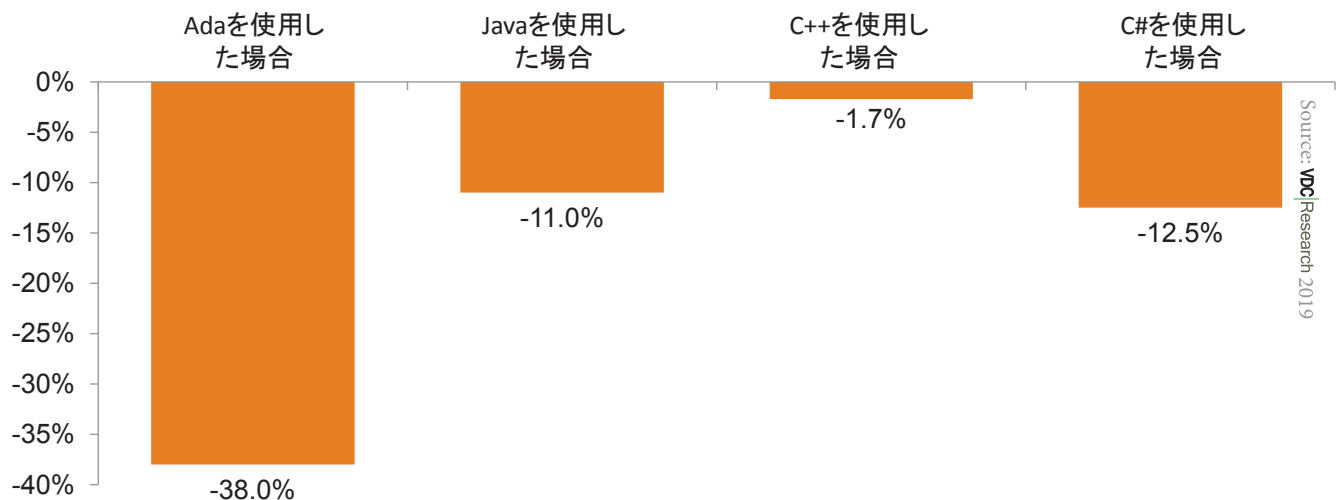
表5：通信/ネットワーク/x86 ベース開発プロジェクトでのソフトウェア開発コストの変化
(C を使用した場合に対するコストの変化率)



ご存知かもしれませんが、航空宇宙と防衛分野ではAda は長年使用されてきましたが、その業界も急速に変化しています。かつてはPowerPC プロセッサを多く使用していましたが、現在はARM に移行しつつあります。

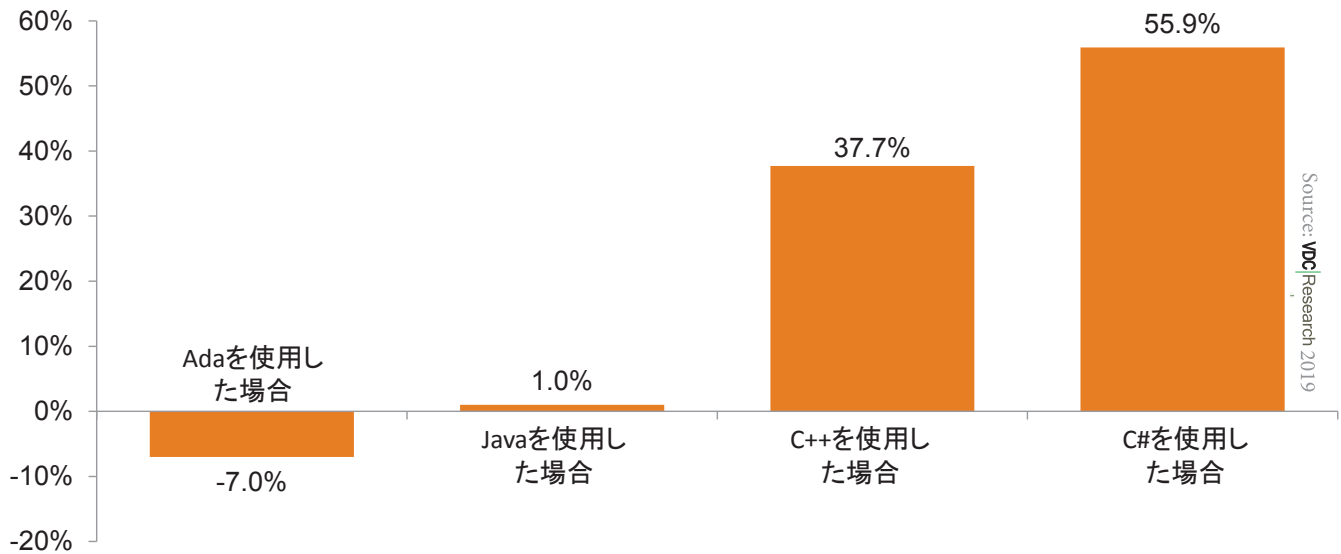
システムインテグレータは、COTS 主導のSWaP-C (Size, Weight Power and Cost) をより重視しています。技術部門では新しい手法とツールを使用してアジリティ (アジャイル開発) を実現することにも新たな焦点を当てています。従来のプロセスが予測可能な結果をもたらさずであろうという信念に基づいて、体系的かつ継続的な開発が行われていました。開発サイクルが長い業界では、明らかに、テクノロジーの選択がさらなる効率向上をもたらす事を理解することが重要です。ARM プロセッサとC 言語をプログラミング言語として使用して、この分野で5,000 台を開発するプロジェクトを検証してみると、Ada は再び最大のコスト削減ができることを示し、ソフトウェア開発コストを38%、50 万ドル以上削減することができます。

表6：航空宇宙および防衛/ ARM ベースの開発プロジェクトでの
ソフトウェア開発コストの変化
(C を使用した場合に対するコストの変化率)



コスト計算の結果は、各分野での係数や考慮事項で大きく変化します。Ada は場合によっては、最良の選択肢ではないかもしれませんが。明確な利便性は見だせないかもしれないし、他の言語を選択した方が良い場合もあります。それでもAda は一般的に良い選択肢と言えます。例えば、自動車/ ARM /C プロジェクトでは、C の継続使用で開発コストの削減が望めない場合でも、あるいはAda による開発リスクが高くない場合でも、諸事情によりAda への変更ができず、C は継続して利用されることがあります。

表7：自動車/ ARM ベースのプロジェクトにおけるソフトウェア開発コストの変化
(C を使用した場合に対するコストの変化率)



Ada が開発コストを削減できる理由は何か？

TCO 計算機で費用削減できる理由を探るために、主な 5 つのAda の優位性を特定しました。

1. 低コストの開発リソース

コスト管理で大変重要なことの一つは人件費です。高度な教育を受けた経験豊富なエンジニアの賃金は高くなる傾向にあり、特に組込システムエンジニアに当てはまります。ソフトウェア開発言語のスキルなどが賃金に大きな影響を与えています。例えば、Ada を使用しているエンジニアの人件費の平均値は、C を使用しているエンジニアの 10% 減、Java を使用しているエンジニアの 20% 減でした (それぞれ 85,000 ドルと 95,000 ドル、105,000 ドル)。

一見すると、この点に関してAdaの相対的な立ち位置は驚くべきものです。長年にわたって、セーフティクリチカル市場と技術は、高コストの労働市場で、大幅なコスト増をもたらしていました。時々この専門性によって高い保険の費用が発生しました。コンシューマ機器と航空宇宙/防衛市場で人件費データを比較しても、多くの場合、それは真実です。しかし、セーフティクリチカルの知識を持ったエンジニアの報酬は過去ほど高いものではありません。現在人気のC++などの言語を修得したエンジニアの需要が高く、場合によっては、一般的ではない言語開発者の市場賃金は低下する傾向にあります。さらに、IT環境で多く利用されているJava等の言語使用が増えているため、労働市場が拡大し全体的には賃金が低下しています。

2. Ada の平均的なプロジェクトの開発期間は短縮される傾向

Ada の開発環境やライブラリが確立されているため、エンジニアは直ちに開発着手できます。投資を軽減し、市場投入までの時間を短縮できます。Ada のプロジェクト期間は、実際、平均的なCのプロジェクトよりも30%短縮され、この点に関して他言語を使用する全てのプロジェクトと比較して有利な調査結果となりました（表8 参照）。結果として、Ada プロジェクトの経費は労働費の割合が低いことが報告されています。つまり、多くの商用プロジェクトで使用するとエンジニアに関連する社内の運用コストのリスクを低減し、予想以上に時間がかかるプロジェクトの潜在的なコストを押し下げています。

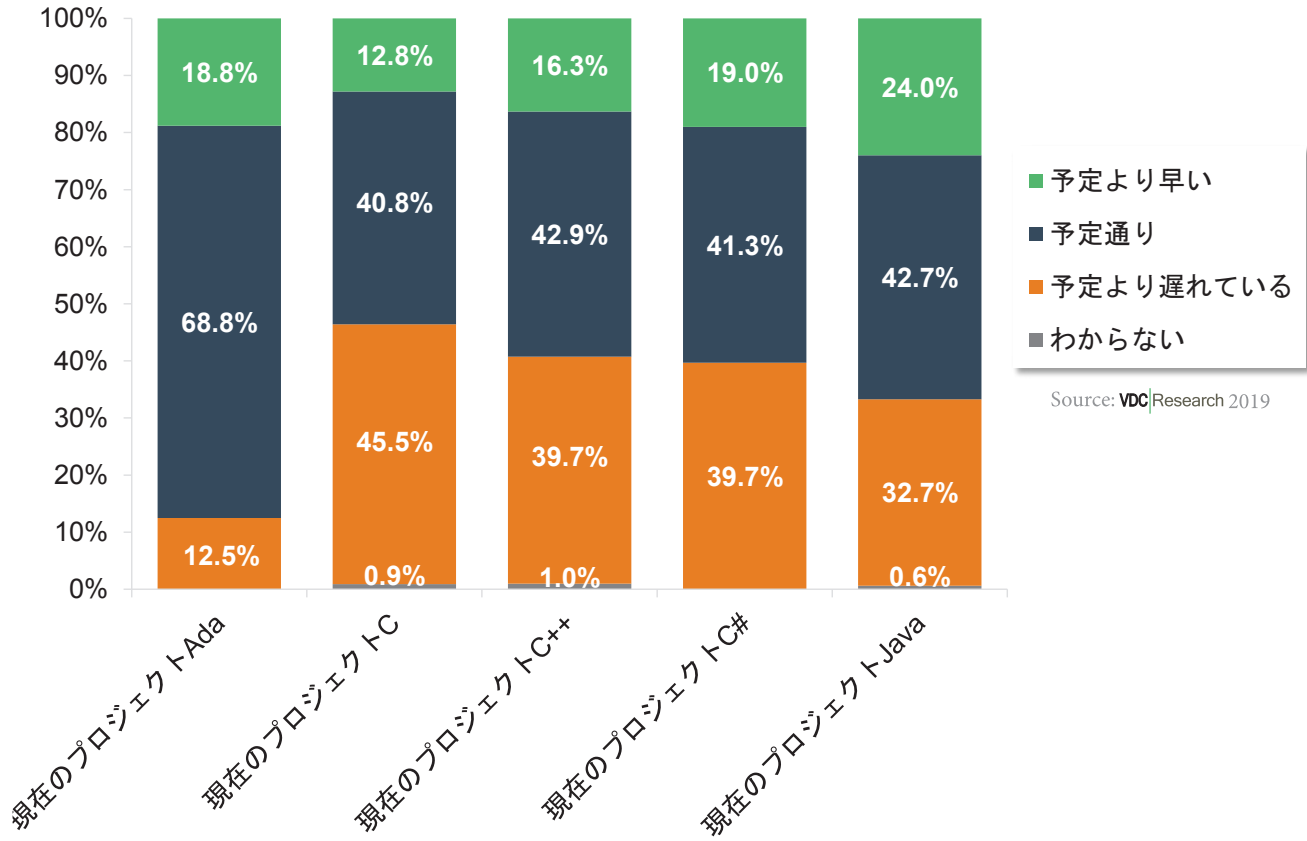
表 8: 全プロジェクト期間の月数（仕様作成から出荷までの時間）
ソフトウェア開発言語使用ごとで分類
（回答の平均値）



3. Ada プロジェクトはスケジュール通りに終了する率が高い傾向。

スケジュール遵守は、エンジニアリングプロジェクト管理にとって重要な点の一つです。遅延の原因となる理由は（判る範囲に絞ったとしても）いくつも存在しますが、OEM はプロセスや技術の変更をしながらスケジュールを管理することに多くの労力を払っています。繰り返しになりますが、ソフトウェア開発言語の選択は、プロジェクトの成果との大きな相関関係を示しています。例えば、C を使用しているエンジニアがAda を使っている人に比べて、予定よりもプロジェクトが遅れたことの報告は3 倍以上ありました。つまり、Ada をソフトウェア開発言語として使用しているとプロジェクト期間は、短縮されるだけでなく遅延のリスクも減少させます。

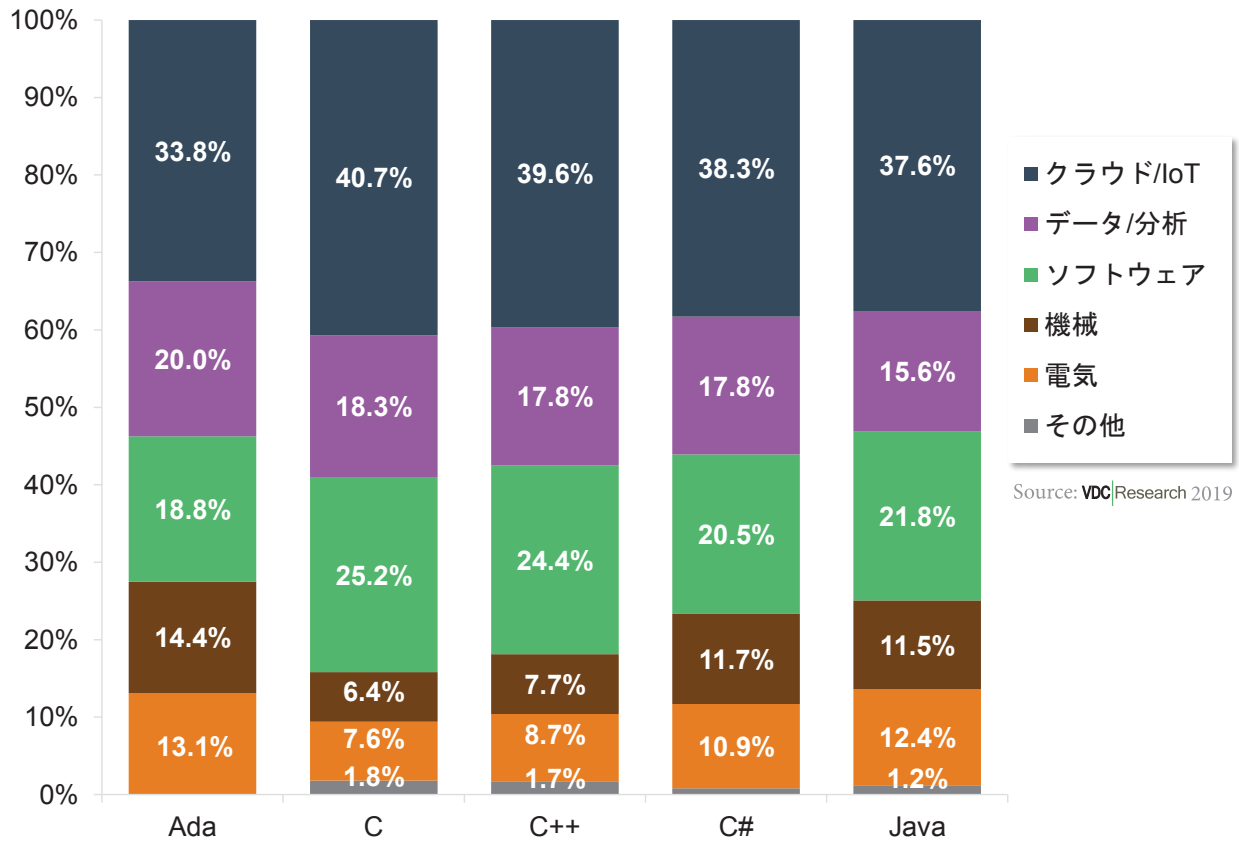
表9：ソフトウェア開発言語の使用別、現在のプロジェクトスケジュールの遵守
ソフトウェア開発言語使用による分類
(回答率)



4. 人的資源の活用

ソフトウェア開発はエンジニアリング組織にとって最大のコストセンタで、開発チームが不必要な費用追加を最小限に抑えることが重要です。この観点で人的資源を選別して、別の分野に充てることができます。実際に、他言語を使用しているユーザよりも、Ada ユーザはプロジェクトチームのソフトウェア開発に占めるコストの割合が少ないと報告されています。さらに、クラウド/ IoT や分析機器など、差別化が必要な分野でAda ユーザの割合が高くなっています。

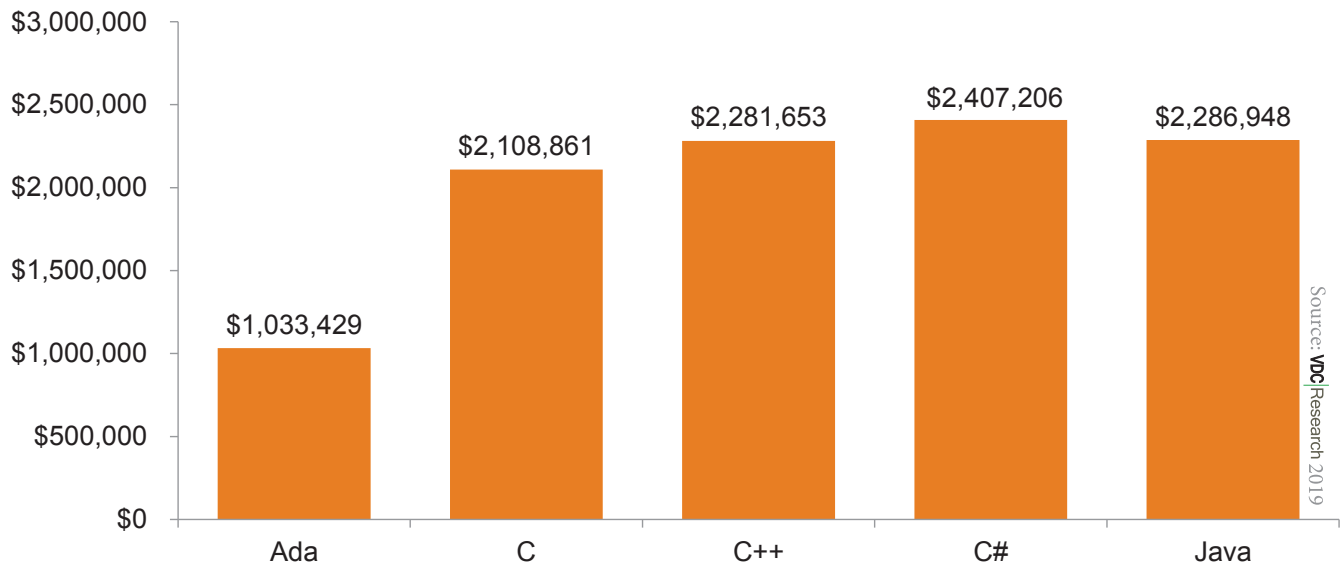
表 10: プロジェクトごとの開発費の推定配分、
ソフトウェア開発言語別の分類
(回答の平均値)



5. ソフトウェア開発のコスト削減

従来報告されているソフトウェアの効率性に関する統計と比較しても、Adaは、プロジェクトの総開発費を軽減できることを示しています。Adaは、実際に他の主要言語と比べて、開発の平均総費用が約半分であったという報告があります。それは、業界の平均に従って結果を正規化するTCO計算機の多くのシナリオにも当てはまっています。

表 11: ソフトウェア開発言語別プロジェクト毎の推定総開発費用
(回答の平均値)



Ada の優位性

前述の定量化された指標とは別に、Ada の開発における利便性、ならびに、その結果となった経緯を理解することが重要です。Ada の優位性を検証するために、主な4つの理由を説明いたします。

1. 信頼性

IoT に注目されていますが、依然として組込システムが成功するために信頼性は最も重要な課題と言えます。多くの点で、IoT がもたらすビジネスでのより深い相互接続で、信頼性の重要度はさらに増します。信頼性が重要な点の一つは、データミニスティック（確定論的）ならびにリアルタイム性によるものです。表2に示すように、多くのエンジニアにとって大きな課題です。長年Ada はセーフティクリチカル市場で利用されてきた実績があり、多くの機能を備えています。例えば、この観点で、認定済のコードジェネレータが利用できれば、認証取得文書の作成が容易になります。そのためAda の実績から組込およびIoT 開発プロジェクトでの選択肢の幅または使用できる範囲が広がります。そのうちの半分は、依然としてリアルタイムの要件が含まれています。さらに、SPARK のようなツールソリューションを使用することで、形式検証の利用ならびにデータ/プログラムフローの正しさを証明するプロセスにより、検証時間とコストをさらに削減できます。そのため、他言語とは異なり、ランタイムエラーの発生を未然に防止、変数を使用前に初期化するため、セキュリティ悪用の脆弱性等も軽減します。

2. 再利用性

前述した市場投入までの時間短縮のため、効率性を重視して再利用に焦点を当てる必要があります。過去の投資を新たな設計に利用できない場合でも、技術開発に貴重な工数を割り当てる余裕がありません。Ada は再利用性を念頭に置いて設計されており、その文法は実装時の依存関係を軽減します。Ada はISO 規格に準拠しており、あいまいさは少なく、再利用性は高くなります。

3. 柔軟性と拡張性

今日の市場では、敏捷性と将来の要求に対応する能力を組織は最大限に高める必要があります。テクノロジーの選択は、導入の速さと費用対効果を最大化するために、同じことを考慮する必要があります。認証が必要な製品では、柔軟性と拡張性がさらに重要となります。認証案件は一般的に時間とコストが増大する傾向にあります。そのため開発手法を簡素化して、認証（ならびに再認証）に係る時間とコストにおいて、新たな変更に対する影響を最小限に留める必要があります。その点でAda の階層化ライブラリは、オリジナルを修正または再コンパイルすることなくモジュールを拡張できます。

4. 使いやすさ

他のテクノロジーと同様に、導入、開発者の投入、そして最終的には、優位性はその使いやすさによって大きく左右されます。上記のように、他の言語に比べてAda のプログラミング規則は厳密で、デバッグ/検証に余分な時間を費やすコーディングエラー等を最小にすることが可能です。この機能によって、特定のビットおよびメモリサイズ等に関して検討する時間が減るため、使いやすさも向上します。そのため、C では、さまざまなコーディングメカニズムを実装する必要があります。使いやすさの理由は、明らかで、従来から一般的に利用されているテクノロジーのスキルセットと経験も満たしているからです。この目的のために、Ada コミュニティは、Ada を習得するためにC/C++ユーザを対象とした広範なライブラリと文献を作成しました（多くのコンピュータサイエンスのカリキュラムに伝統的なAda が含まれることは言うまでもありません）。実際、Ada は他言語とのインタフェースが既に組込まれているという点で優れています。多言語を用いた開発の頻度が高い今日のプロジェクトを考慮すると、今日の環境では開発者の経験の向上に重点が置かれます。

結論

組込システム工学の実務を再定義する必要性と機会はこれまでほとんどありませんでした。IOTは、システム機能とビジネスプロセス要件の広範囲にわたる変化を引き起こしました。残念ながら、多くの場合、要件が変更されても、開発テクノロジーと実務を組合せて十分な再評価は行われていませんでした。特定の機能要件を充足する必要性から多くの場合、開発では部品表のコンポーネントを重視する傾向にあります。また、製品化までの時間が限られているため、現状を維持したまま既存の開発環境を利用し続けることを選択する必要があります。

開発技術（ツール）は、組込システムの開発作業で変動幅のある費用の一つで大きな影響を及ぼします。製品納期が顧客満足度に影響があるだけでなく、サービス収入にも影響を与える可能性があります。そのため、開発チームの効率は非常に重要になります。今回の調査では、プログラミング言語の選択の影響は大きく、プロジェクト期間の短縮、スケジュールの改善、最終的には開発コストの削減につながることを示しています。多くの要因が言語選択に影響を与え、決定を左右しますが、今回の調査によると、Adaの進化は技術的にも財政的にも健全なソリューションを提供しながら、開発組織にとってより魅力的な選択肢であることを示しています。