

モデルベースデザインの投資利益率 (ROI) の測定

サイバーフィジカルシステム (CPS) の複雑化に伴い、従来のアプローチによるシステム開発では、品質の維持やコストのコントロールが困難になりつつあります。この課題に対応し、競争力を高めるために、企業ではシステムやソフトウェアの開発用にモデルベースデザイン (MBD、モデルベース開発) を採用しています。ただし、投資を決定するには、モデルベースデザインを採用するメリットを、それを十分に発揮するためのサポートプロセスとともに正当化する必要があります。本紙で紹介するモデルベースデザインの投資利益率 (ROI) フレームワークは、従来の開発手法と比較してモデルベースデザインを採用した場合に想定されるコスト削減効果を定量化することで、モデルベースデザインへの投資を正当化するための解析ツールを提供します。

はじめに

顧客要件の範囲や複雑度の増大に伴い、システムの論理および制御ソフトウェアもその範囲や複雑さを増しています。航空機や自動車に必要な何百万行ものコードをますます厳しいスケジュールのもとで開発する際に、従来の開発プロセスでは品質やスケジュール目標を満たすのに十分ではないことが分かっています。モデルベースデザインを使用したシステム開発では、開発プロセスの初期段階で欠陥を特定し、潜在的な欠陥の総数を減らすことでコストを削減できます。産業機械、ロボティクス、無線システムなどの複雑なシステムを、より高品質に、低コストかつ短期間で提供することで、モデルベースデザインは企業の競争力を高めます。

従来の開発プロセスとモデルベースデザインの比較

従来の開発プロセスでは、各段階のタスクが、異なるツール環境で順次実行されます。このプロセスには、問題を引き起こす可能性のある多くの手作業が含まれます (図 1)。モデルベースデザインの手法が採用されていない場合は、システム要件の定義からシステムの運用まで、開発の各段階において非効率性が生じます。

システム要件は、Microsoft® Word® や IBM® Engineering Requirements Management Doors® などのツールを使用してテキストで把握できますが、システム アーキテクチャは描画ツールで指定されるため、変更時の解析、解釈、および管理が困難になります。サブシステムの設計は、ドメイン固有のツールを使用して作成されるため、ソフトウェアやハードウェアに実装されるまでシステムレベルのテストを行うことができません。その後、手作業で設計をコードに変換しますが、これは時間がかかり、欠陥が発生しやすいプロセスです。各フェーズで何らかの欠陥が発生し、テストフェーズでは、それまでのフェーズを通して蓄積された欠陥がすべて集まります。共通のツール環境の欠如、複数の手作業、および欠陥の発見の遅れが、すべて開発時間とコストの増加の原因となります。また、運用データやそれに伴う分析情報も有効に活用されず、システムの運用中にシステムの効率や稼働率を向上させる貴重な改善機会は見送られることとなります。

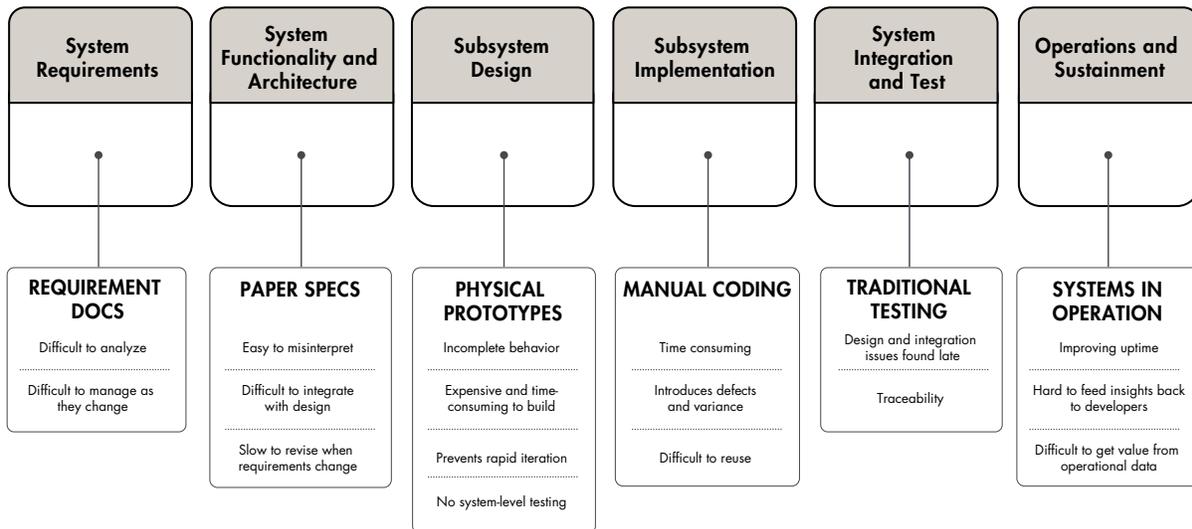


図 1. 従来のソフトウェア開発プロセス。この手法では、欠陥を発生させる可能性のある多くの不必要な手作業が必要になります。

モデルベースデザイン (図 2) では、開発プロセス全体でモデルを体系的に使用します。従来のプロセスと同じ連のシステム要件から開始します。ただし、要件はテキストによる仕様の基礎となるものではなく、動作モデルやアーキテクチャモデルといった実行可能な仕様であるシステム アーキテクチャを開発するために使用されます。エンジニアは、これらのアーキテクチャモデルを使用して、要件や仕様を明確化します。次に、このモデルをベースにして、サブシステムの詳細設計を開発します。

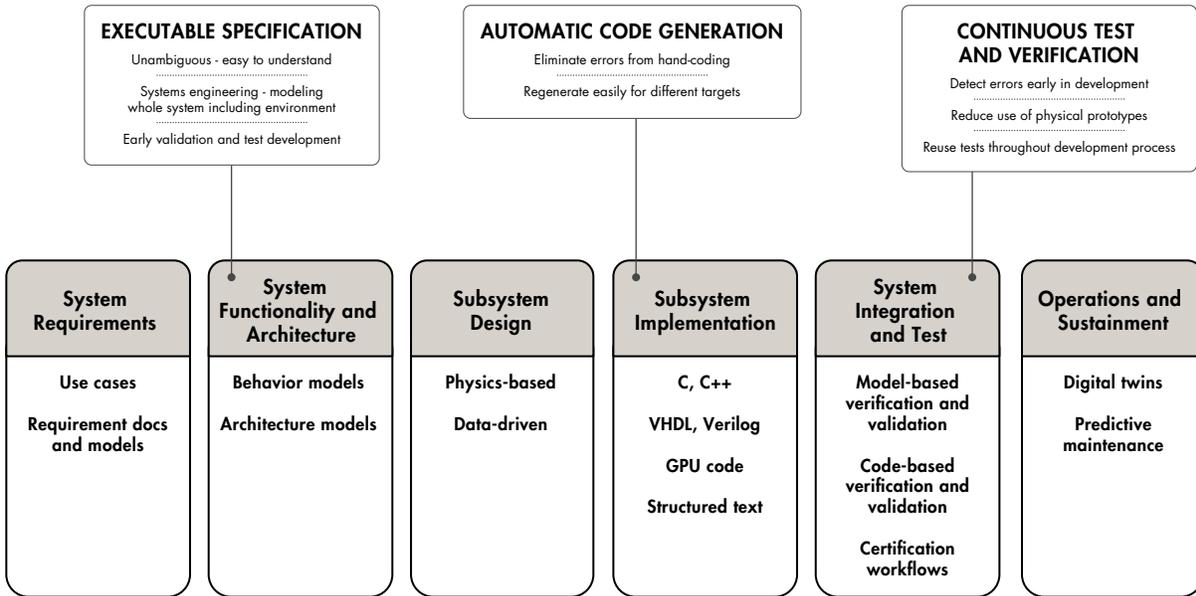


図 2. モデルベースデザインを用いたソフトウェア開発プロセス。この手法では、システムレベルモデルを開発全体で実行可能な仕様として使用し、システムレベルおよびコンポーネントレベル設計とシミュレーション、自動コード生成、および継続的なテストと検証をサポートします。

MATLAB® および Simulink® によるモデルベースデザインを使用すると、エンジニアはシステムレベルで設計をシミュレーションし、実装前にインターフェイスの欠陥を発見することができます。設計が完了すると、エンジニアはモデルから量産品質のコードとテストケースを自動的に生成し、コードを手書きすることによるエラーを回避することができます。このワークフローでは、エンジニアはシステム要件の定義からシステムのテストまで同じ環境で行うことができ、手作業を最小限に抑えることができます。さらに、実行可能な仕様をモデルでシミュレーションし、要件を満たしていることを検証する要件フェーズでテストを開始することができます。その結果、欠陥を早期に発見して除去し、開発の総コストを減らすことができます。

システム エンジニアリングの観点からのコスト削減

Jerry Krasner 氏が Embedded Market Forecasters のために行った調査によると、モデルベース システムズ エンジニアリングにおいてモデルを体系的に使用することで、2 年後には全体で最大 55% のコスト削減効果があるとされています [1]。モデルを使用することで、過去のコミュニケーション不備により過剰に指定されがちなテキスト要件への依存を軽減することができます。モデルベースデザインを使用する場合、モデルが開発プロセス全体での共通言語になります。この手法により、製品仕様のあいまいさが軽減され、シミュレーションの使用により要件の妥当性を確認することができます。製品の作成前にモデルを使用した製品仕様のシミュレーションを行い、調整することが、コスト削減の主な理由です。

また、モデルベースデザインでは、システム要件やアーキテクチャを開発する際に複数のソリューションを探求することができ、設計プロセスの初期段階でより速やかに設計の反復を行ったりコンセンサスを形成したりすることができるようになります。モデルベースの手法を使用すると、システムエンジニアリング フェーズからプロジェクトの開発フェーズおよび運用フェーズまで、単一のツールチェーンを使用し、システム アーキテクチャをシステム設計にシームレスかつ効率的に接続することができます。

開発の観点からのコスト削減

モデルベースデザインを採用した組織では、従来の手法と比較して、20 ~ 60% のコスト削減を実現しています [2, 3]。これらのコスト削減の大部分は、優れた要件の解析と、早期かつ継続的なテストと検証の組み合わせによって生み出されます。モデルを使用して要件や設計をシミュレーションすることで、開発プロセスの初期段階で欠陥が発見され、修正にかかるコストを桁違いに少なくすることができます (図 3)。

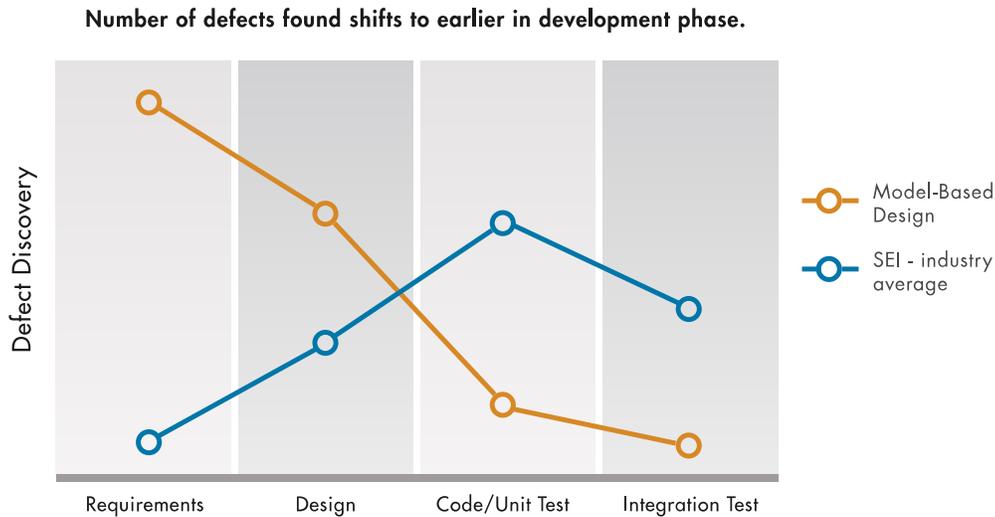


図 3. 開発フェーズで発見された欠陥。モデルベースデザインでは、プロセスの初期段階で欠陥を発見できるようになります。

モデルベースデザイン ROI フレームワークを使用したコスト削減効果の定量化

モデルベースデザイン ROI フレームワークは、特定のプロジェクトでモデルベースデザインを採用した場合の ROI を推定するために設計されたものです。このフレームワークでは、プロジェクトの規模やチームの規模などの要因をもとに、基本的な COCOMO (Constructive Cost Model) モデルを用いて従来の開発にかかるコストを算出し、モデルベースデザインによるコスト削減分を差し引いてモデルベースデザインの開発コストを算出します。(このフレームワークでは、基本となる COCOMO モデルを選択しました。これは、調達コストの説明責任を果たすために、ソフトウェアのコストを推定するための厳密なモデルが要求される航空宇宙および防衛産業で幅広く使用されている、一般的なパラメトリック コスト推定ツールであることが理由です。)

次に、モデルベースデザインを開発に使用した結果得られるコスト削減分を考慮して ROI が算出されます。この計算には、Software Engineering Institute (SEI)、Institute of Electrical and Electronics Engineers (IEEE)、および業界調査による指標が使用されています [1、4、5、6]。プロジェクトの範囲、既存のプロセス、モデルベースデザインを使用するチームの専門性は、産業分野や企業によって異なるため、モデルベースデザイン ROI フレームワークは、特定のプロジェクトやチームに合わせてカスタマイズすることができます。

5 万行のコードを扱うソフトウェア プロジェクトのベースラインケースについて考えてみます。基本的な COCOMO モデルを使用して、従来の手法で開発した場合のコストは約 600 万ドル (米国ドル) となります。従来の手法と比較したモデルベースデザインのコスト削減効果を算出するために、要件、設計、実装、およびテストの各開発フェーズを業界の指標に基づいて解析します。次に、コスト削減分をまとめて、従来の開発コストから差し引きます。この場合、モデルベースデザインのコストは 300 万ドル (米国ドル) で、従来の手法と比較して 50% のコスト削減になります。

このフレームワークでは、50% のコスト削減を算出するまでに、モデルベースデザインで排除される従来の開発プロセスの非効率性を検証し、顧客からの経験的データ、顧客へのインタビュー、業界の指標、および平均値に基づいてコスト削減効果を計算しています。このフレームワークでは、モデルベースデザインの段階的な採用に適応するよう、各開発フェーズでのコスト削減効果を個別に算出しています。

要件開発時のコスト削減

このフレームワークがどのように機能するかを理解するために、要件フェーズでの非効率性を見てみましょう。モデルを使用して、あいまいさ、矛盾、またはテスト不可能な要件を明らかにすることにより、エンジニアはより高い割合で欠陥を発見することができます。ベースラインケースでは、要件の 15% に欠陥があるか、または手直しが必要であると想定していますが、これは業界のインタビューから得た推定値です。これらの欠陥を要件フェーズで発見することは、開発フェーズでのコストのかかる手直しを防止することを意味します。要件のコスト削減分は、欠陥のある要件の数に、要件フェーズ後に検出された欠陥のある要件を解決するための平均時間を乗じることで算出されます。ベースラインケースでは、欠陥のある要件 1 件あたりの平均対応時間は 4.5 時間です [6]。この計算により、モデルベースデザインでは 3,375 時間のエンジニアリング時間を削減することができます。表 1 は、要件解析の問題点を取り扱うフレームワークのセクションを示しています。

要件開発とトレーサビリティ	
手直しが必要な要件の割合	15%
不正確な要件の数	750
各要件の手直しに要する時間	4.5
要件フェーズ後の不正確な要件の修正を回避することにより削減される作業時間	3,375

表 1. 不正確な要件を早期に修正することで削減できるエンジニアリング時間数の ROI フレームワーク計算。

テスト時のコスト削減

テストフェーズでの非効率性もこのフレームワークで把握され、コスト削減の大部分を占めることとなります。モデルベースデザインを使用すると、テストを自動的に生成してモデルの検証および妥当性確認を行い、フルテストカバレッジを得ることができます。この手法により、開発プロセスの遅い段階で未達テストカバレッジに対処するために貴重な時間を費やすことを回避できます。これらのテストケースは、デスクトップシミュレーションからハードウェアインザループ (HIL) テストまで、複数のテスト段階を通して使用することができます。テスト結果をまとめたレポートや、業界規格に準拠したレポートを自動生成することも可能です。5,000 件の要件を持つプロジェクトを想定した場合、モデルベースデザインを使用すると、モデリングフェーズ時に包括的なテストケースを生成し、複数のテスト環境でテストケースを再利用できるため、最大で 12,000 時間の削減が可能になります。図 5 は、テストの問題点を取り扱うフレームワークのセクションを示しています。

テストとレポート	
要件ごとの未達テストカバレッジの対処に要する時間	2
テストの自動生成によるコスト削減率	70%
削減される未達カバレッジ対応時間	7,000
要件ごとのラボでのテストの検証またはデバッグに要する時間	2
テストの再利用率 (デスクトップテストからラボまで)	50%
削減されるラボでのテスト時間 (テストの再利用)	5,000

表 2.モデルベースデザインを使用することで、テストワークフローおよびレポートワークフローで削減されるエンジニアリング時間数の ROI フレームワーク計算。

開発全体におけるコスト削減

このフレームワークには、開発プロセスの各段階におけるさまざまな非効率性を取り扱ういくつかのセクションがあります。開発プロセス全体でのコスト削減効果をまとめると、この例ではテストフェーズでのコスト削減効果がほぼ半分を占めています (図 4)。これは、開発プロセスの初期段階で徹底した要件解析を行うことにより、要件やテストの修正と改善にコストがかかるテストフェーズまで持ち越される欠陥の数が減少するためです。簡単に言えば、より良い要件はより良い設計につながります。早期かつ継続的にテストを行うことで、欠陥をそれが発生したフェーズで特定し、対処することができます。これにより、ソフトウェアに残る潜在的な欠陥が減少し、全体的な開発コストの削減につながります。

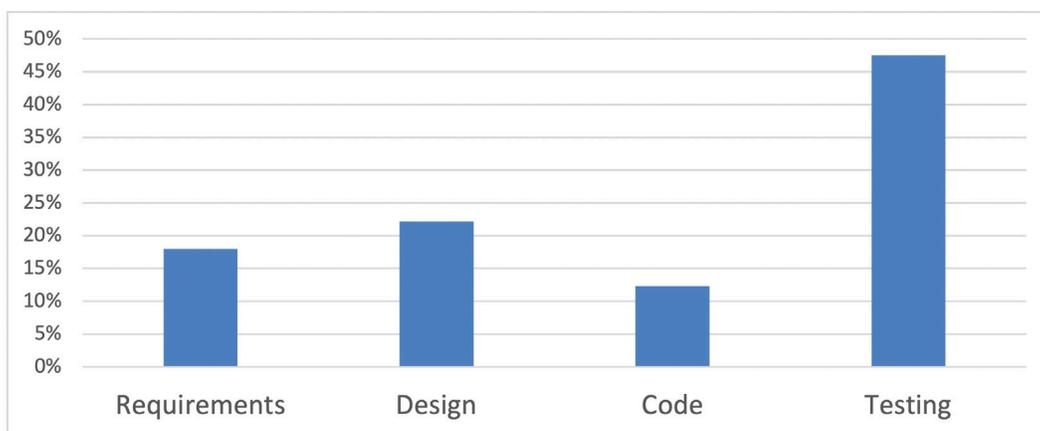


図 4. 開発フェーズ別の総コスト削減の割合。
総コスト削減のほとんどを占めるテストフェーズでのコスト削減。

モデルベースデザインを採用する企業が MathWorks と共同作業する場合、この ROI フレームワークを採用プロセスの指針とすることで、モデルベースデザインへの移行によってすぐに大きなメリットを得ることができる領域を特定することができます。

開発ライフサイクルを通じたコスト削減

ROI フレームワークでは、開発ライフサイクルを通じたコスト削減を定量的に表しますが、モデルベースデザインの採用による経済的な利点は、これらの削減だけではありません。開発のペースが上がることで、市場投入までの時間が短縮され、モデルベースデザインを使用するお客様は競争力を高めることができます。さらに、モデルベースデザインによってオーバーヘッドやコミュニケーション不備が減少し、主要な業務活動や技術革新に必要なリソースを確保できます。

まとめ

多くの企業にとって、新しい技術やプロセスへの投資はリスクを伴います。本紙でご紹介した投資収益の計算は、モデルベースデザインへの投資をサポートするための解析上の根拠となることを目的としています。ROI フレームワークでは、投資の正当化に加え、チームはモデルベースデザインで最もコスト削減が可能な領域を特定し、詳細な調査を進めることで大幅な追加のコスト削減が可能な領域を特定することができます。

ご所属のチームや組織向けにカスタマイズされた ROI 計算を希望される場合は、[営業担当にお問い合わせください](#)。

関連情報

- [モデルベースデザインを採用する理由](#)
- [エンジニアリング チームがモデルベースデザインを採用する方法](#)
- [シミュレーションおよびモデルベースデザイン向け Simulink](#)
- [Model-Based Design Process Assessment および Maturity Framework](#)

参照

1. Krasner, Jerry. How Product Development Organizations Can Achieve Long-Term Cost Savings Using Model-Based Systems Engineering (MBSE). October 2015. https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:how_product_development_organizations_can_achieve_long-term_savings_1_.pdf
2. MathWorks と OHB が自律フォーメーション飛行向け衛星 GNC (誘導、ナビゲーション、および制御) ソフトウェアを開発。 https://www.mathworks.com/company/user_stories/ohb-develops-satellite-guidance-navigation-and-control-software-for-autonomous-formation-flying.html
3. MathWorks と BAE Systems がソフトウェア無線の開発期間を 8 割短縮。 https://www.mathworks.com/company/user_stories/bae-systems-achieves-80-reduction-in-software-defined-radio-development-time.html
4. Over, James W., *Managing Software Quality with the Team Software Process*, Software Engineering Institute. <http://c-spin.net/2010/c-spin201004Managing%20Software%20Quality%20with%20the%20Team%20Software%20Process.pdf>
5. Vallespir, Diego, and William Nichols. *Analysis of Code (and Design) Defect Injection and Removal in PSP*. CARNEGIE-MELLON UNIV PITTSBURGH PA, 2012. https://resources.sei.cmu.edu/asset_files/Presentation/2012_017_001_298088.pdf
6. Tom King, Joe Marasco, *What Is the Cost of a Requirement Error?* StickyMinds. <http://www.stickyminds.com/sitewide.asp?ObjectId=12529&Function=edetail&ObjectType=ARTCOL>