

開発プロセスの自動化とDevOpsの実践で 継続的なリリースを支援するソフトウェア開発環境

Software Development Environment Allowing Continuous Delivery of Software Based on DevOps Methodology

前田 尚人 MAEDA Naoto 田原 歩 TAHARA Ayumu 長岡 武志 NAGAOKA Takeshi 山元 和子 YAMAMOTO Kazuko

CPS（サイバーフィジカルシステム）事業を支えるソフトウェア開発には、ビジネス環境や顧客ニーズの変化と技術の進展に対応して、短いライフサイクルでソフトウェアの機能を強化し、継続的にリリースすることが求められる。

東芝グループは、分散拠点間での開発資産の共有や、成果物の品質管理、開発の自動化を実現し、開発と運用を組み合わせることでソフトウェアの継続的なリリースを可能にするDevOpsを適用した独自のソフトウェア開発環境を構築した。これにより、ソフトウェアの要求項目からソースコードまでのトレーサビリティを確保するとともに、不具合の発見から修正までの期間短縮と工数削減が可能になった。

For the successful development of software supporting cyber-physical systems (CPS) services, it is essential to frequently release software updates providing enhanced functions in line with the changes taking place in the business environment and customers' needs as well as the progress of technologies.

The Toshiba Group has constructed a proprietary software development environment for sharing software components between different development bases, managing and controlling the quality of development assets, and automating development processes, through the use of DevOps methodology. This software development environment makes it possible to ensure the traceability of development processes from requirements to source codes and reduce the time and cost required to find and correct defects.

1. まえがき

CPS事業を支えるソフトウェア開発には、ビジネス環境や顧客ニーズの変化、及び技術の進展に対応し、ソフトウェアを迅速に機能強化して継続的にリリースすることが求められる。それらを実現するための手法にアジャイル開発やDevOpsがあり、これらを用いて繰り返し行うソフトウェアのビルド^(注1)、テスト、デプロイ^(注1)といった一連のタスクを自動化し効率的なリリースを支援するソフトウェア開発環境が必要である。更に、開発のグローバル化が進んだことにより、分散した拠点間で開発資産を共有するには、様々な場所からアクセス可能なソフトウェア開発環境が必要である。

ここでは、ソフトウェア開発の動向や、ソフトウェア開発環境に関する東芝グループの取り組みとその事例について述べる。

2. ソフトウェア開発の動向

ソフトウェア開発においては、従来はウォーターフォールモデルと呼ばれる開発プロセスが用いられてきた。ウォーターフォールモデルとは、要件定義、設計、プログラミング、テストのように、逐次的に開発工程が進むプロセスモデ

ルであり、工程の全体像を俯瞰（ふかん）して計画を立てやすいというメリットがあることから大規模開発で用いられることが多い。しかしその一方で、前工程で作成した成果物を後工程のインプットとしていくため、開発工程の途中で要求仕様の見落としや間違いが見付かった場合に前工程に戻らなければならない、このような後戻りを防ぐには各工程で成果物の品質を確保する必要がある。また、大規模開発では要求仕様も多くなり、要求仕様にひも付くタスクや、成果物、レビュー記録、テストが爆発的に増加するため、成果物の品質のばらつきを抑えることが難しい。更に、要求仕様の変更が発生した場合、影響する範囲を特定するために、これらの関連付けを管理するコストが掛かってしまうというデメリットがある。

一方、近年はビジネス環境の変化のスピードが加速しており、それに合わせて必要となる機能も変化し、開発初期に要求仕様の確定が難しいソフトウェア開発では、従来のウォーターフォールモデルでは適合しない場面が出てくる。このような場合は、アジャイル開発が用いられる（この特集のp.15-18参照）。アジャイル開発は、決められた計画の遵守より変化への対応を重視し、包括的なドキュメントより実際に動くプログラムに価値を置くという考え方に基づいている。原則として、事前に開発全体の詳細な計画は作らず、

(注1) ビルドとは、ソフトウェアを実行可能な形式に変換すること。デプロイとは、ビルドしたソフトウェアをテスト環境や運用環境に配置すること。

1～4週間という一定の短い周期で要件定義、開発、テストを繰り返しながら、動作可能なソフトウェアを作り上げていくという特徴がある。

更に、開発したソフトウェアをいち早く運用してサービスとして公開するには、開発(Development)と運用(Operations)を組み合わせたDevOpsを実践する必要がある。2020年6月時点でもDevOpsに厳密な定義は存在していないが、ここでは“システム開発のライフサイクルを短縮し、継続的にリリースする手法”をDevOpsの定義とする(同p.2-6参照)。DevOpsの実践では、例えば数日に一度、ソフトウェアのアップデートが行われるため、これに伴うソースコードのチェック、ビルド、テスト、リリースといった繰り返し行われる一連のタスクを効率良く実行するために、極力自動化する必要がある。

このように、ビジネス環境の変化のスピードが速い分野におけるソフトウェア開発手法は、ウォーターフォールモデルからアジャイル開発やDevOpsの実践に変わってきている。

3. 東芝グループのソフトウェア開発環境

3.1 ソフトウェア開発環境の特徴

東芝グループでは、要件定義、設計、開発、テスト、デプロイといったソフトウェア開発における一連の流れをシームレスに行えるように、複数のツールを組み合わせたソフトウェア開発環境¹⁾を構築、運用している。複数のツールは連鎖的に利用できるように組み合わせられており、ツールチェーンと呼ばれている。東芝グループのソフトウェア開発環境はツールチェーンの考え方を基に、**図1**に示すように、OSS(オープンソースソフトウェア)をベースとして構成しており、OSSだけでは不足する機能については、独自ツールとして開発を行って補完している。そして、ソフトウェア開発プロジェクトの特性に合わせて必要なツールを組み合わせ、グループ内にサービス提供している。

東芝グループは、2013年からソフトウェア開発環境の提供を開始しており、主に、ウォーターフォールモデルによるソフトウェア開発で利用されてきた。2章で述べたように、ウォーターフォールモデルでは、多人数で開発する場合のソフトウェア品質のばらつきを抑え、各工程で成果物に求められる品質を確保する必要がある。これを支援するために、**図2**に示すように、設計間の整合性をチェックする設計情報管理ツールを組み込んでいる。また、要求仕様の変更が発生した場合に影響する範囲を特定する必要があることから、要求項目と要件・設計項目間の追跡を支援する要求管理ツール²⁾を組み込んでいる。

更に、2018年からは、アジャイル開発やDevOpsの実

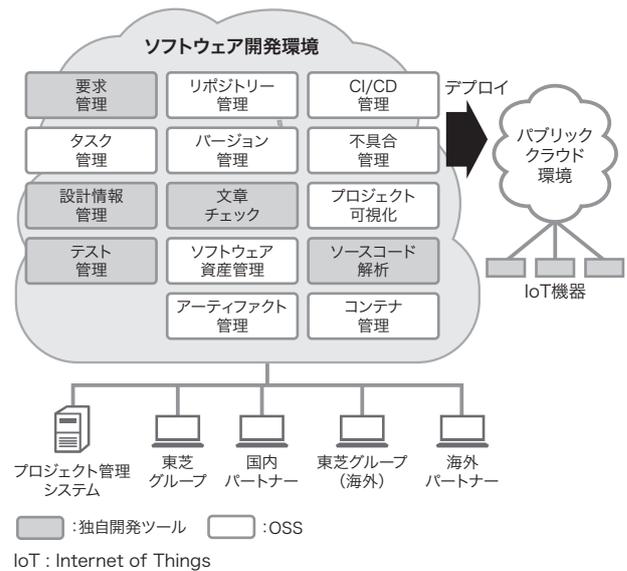


図1. 東芝グループのソフトウェア開発環境の構成

要件定義、設計、開発、テスト、デプロイといったソフトウェア開発における一連の流れをシームレスに行えるように、複数のツールを組み合わせ、連携させたソフトウェア開発環境を構築、運用している。

Configuration of software development environment

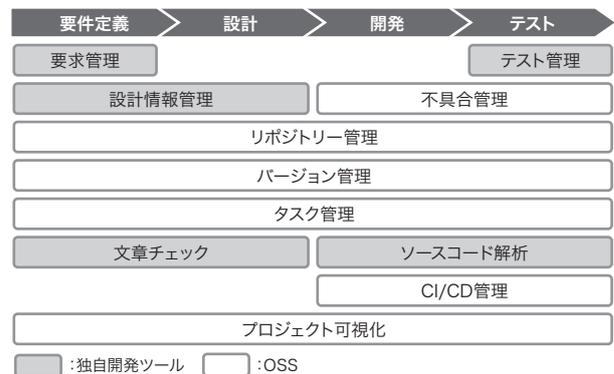


図2. ウォーターフォールモデルでの開発を支援する機能

品質確保と作業の効率化を目的として、独自開発のツールを組み込むことで、ソフトウェア開発環境の機能強化を行っている。

Components of software development environment supporting software development by waterfall model

践、パブリッククラウド上で開発を行うプロジェクトに対応したソフトウェア開発環境の提供を始めている。2章で述べたように、アジャイル開発やDevOpsの実践では極力自動化を行わなければならない。ソースコードのチェック、ビルド、テスト、リリースといったタスクの自動化を支援するツールとして、CI/CD(継続的インテグレーション/継続的デリバリー)管理ツールがある。また、ソフトウェアの部品情報やコンプライアンス情報を管理するOSSを活用したソフトウェア資産

管理ツール⁽³⁾とアーティファクト管理ツール^(注2)を、ソフトウェア開発環境に組み込んでいる(この特集のp.23-26参照)。これらのツールの詳細については、3.2節で説明する。

更に、ソフトウェア開発環境サービスの利用者数の増加に伴いサーバーリソースの柔軟な拡張が必要になったことに加え、開発した成果物をパブリッククラウドへシームレスにデプロイできることが求められたため、ソフトウェア開発環境サービス自体をパブリッククラウド環境に移行した。

3.2 DevOpsを支援する開発環境

2章で述べたように、アジャイル開発及びDevOpsの実践では、1～4週間という一定の短い周期で要求、開発、テストを繰り返しながら、動作可能なソフトウェアを作り上げ、更に短い期間でのリリースを実現する必要がある。これを実現するための手段として、CI⁽⁴⁾やCDがある。これらを組み合わせた“CI/CD”は、ソフトウェアのビルド、テスト、デプロイを自動化し継続的に行う手法を意味し、これを支援するツールがCI/CD管理ツールである。東芝グループは、DevOpsの実践を支援するため、図3に示すCI/CD管理、ソフトウェア資産管理、アーティファクト管理のツールを強化している。

CIにおいては、従来は手作業で実行していたソースコード解析、脆弱(ぜいじゃく)性の有無確認、ビルド、テストといった一連のタスクをCI/CD管理ツールで自動実行することで、効率的に繰り返し実行できるようになる。自動実行のタスクは、CIの実践手段であるビルドパイプラインとして定義している。東芝グループでは、標準的なビルドパイプラインをテンプレート化して開発プロジェクトに提供しており、こ

れによりCIを素早く実現できる。

CDにおいては、従来は手作業で行っていたインフラ環境構築やソフトウェアのデプロイをCI/CD管理ツールで自動実行する。インフラ環境構築やソフトウェアのデプロイは、実行環境となるサーバーが冗長化構成を構築しているような場合や、開発・テスト・運用といった複数の環境がある場合は、同じ作業を繰り返し実施しなければならない。そのため、インフラ環境構築自動化ツールを使い、環境構築をコード化(IaC: Infrastructure as Code)することで、手作業によるミスを排除する。このインフラ環境構築のコードについてもテンプレート化を行い、開発プロジェクトに提供している。更に、DevOpsを実践する際は、ビルドに必要な、大量のライブラリーの依存関係を維持しながら、繰り返し生成されるビルド結果のバイナリーファイルを、ビルド履歴とともに一貫して管理する必要がある。この管理を支援するアーティファクト管理ツールとCI/CD管理ツールを連携させることにより、継続的なデリバリーを可能としている。

このように、ソフトウェア開発環境として提供するCI/CD管理ツールを利用することで、リリースに伴うタスクを自動化できる。しかし、CI/CD管理ツールを使いこなすには、利用しているソースコードの言語に合わせたビルドの設定や、パブリッククラウドで運用を行うのであればそのパブリッククラウド上でのテスト・運用環境構築の設定といった、様々な知見が必要である。そのため、パブリッククラウド上でのCI/CDの知見をガイドとしてまとめ、主要なパブリッククラウドで、典型的なテスト・運用環境構築に対応している。また、CI/CDハンズオン教育やCI/CD環境構築支援といった取り組みも併せて実施しており、普及と展開を図っている。

4. DevOpsを支援する開発環境を利用した開発事例

4.1 ICカードソフトウェア開発への適用事例

この事例では、東芝グループの開発メンバーに加え、インドの開発拠点や国内パートナーも含めた拠点間で連携して開発を行っており、追加・変更される多くの機能要求とともに、シビアなセキュリティや性能に対する非機能要求が、確実に製品に組み込まれなければならないという課題があった。

そこで、図4に示すように、ソフトウェア開発環境を用いて、追加・変更された要求項目からのタスク、レビュー、ソースコードのトレーサビリティを確保している。各分散開発拠点の開発状況は、即座に把握できるように、タスク管理ツール上で“かんばん”として可視化し、拠点間の開発メンバー同士が自律的に連携して開発を行っている。また、変更したソースコードの品質を確認するため、CI/CD管理ツールを利用して、ビルド、ソースコード静的解析、単体テ

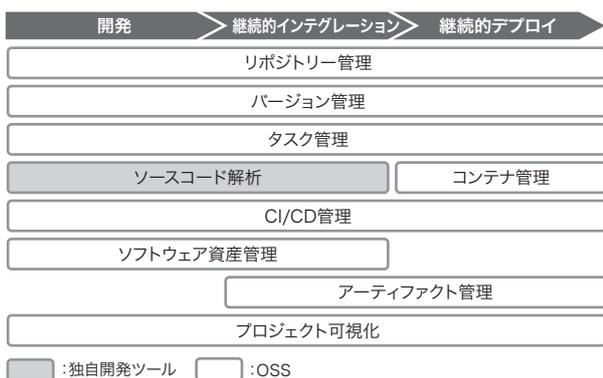


図3. DevOpsの実践を支援する機能

CI/CD管理、ソフトウェア資産管理、アーティファクト管理のツールを強化し、リリース作業の効率化を行っている。

Components of software development environment supporting practice of DevOps

(注2) ビルドに必要なライブラリーやビルドした結果のバイナリーファイルを、一元管理するツール。

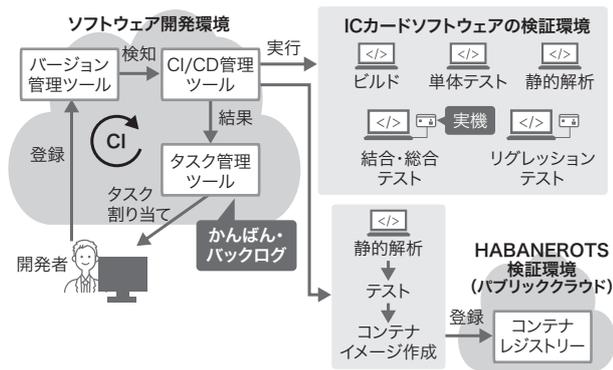


図4. ICカードソフトウェア開発とHABANEROTS開発への適用事例

CI/CD管理ツールを活用し、ビルド、テスト、実機や検証環境へのデプロイを自動化している。

Example of application of software development environment to smart card software and HABANEROTS Toshiba industrial Internet of Things (IoT) platform service development projects

ストを行っている。ビルド状況は、開発室内の共有エリアに設置された大型ディスプレイに表示され、失敗の場合は関係者が迅速に対応できる。更に、実機にデプロイされ、実機を利用したテストも自動的に行われる。

この事例では、これらの取り組みの効果として、要求項目からソースコードへのトレーサビリティ確保と、不具合の発見から修正までの期間短縮・工数削減効果を確認している。

4.2 東芝IoT基盤サービスHABANEROTS開発への適用事例

東芝IoT (Internet of Things) 基盤サービスHABANEROTS (旧名称はHabanero) は、CPSサービスを実現する上で必要となる共通的な機能を、Web API (Application Programming Interface) として東芝グループのCPSサービス開発者に提供することを目的としたWebサービスである (この特集のp.31-34参照)。東芝グループの各種事業は、HABANEROTSをベースにすることで、容易にそれぞれのCPS事業を展開できる。

HABANEROTSの開発は、図4に示すように、ここで述べているソフトウェア開発環境を利用しており、ソースコードのバージョン管理、CI/CD管理ツールを利用したビルド、テスト、及び検証環境へのデプロイを自動化している。これらの種々の自動化により、HABANEROTSの開発を効率化できている。更に、HABANEROTSを構成する自製ソフトウェア・OSSを、ソフトウェア資産管理システムを用いて管理し、自製ソフトウェア部分についてはInnerSource Software (同p.23-26参照) としてグループ内で公開・共同開発していく予定である。

5. あとがき

東芝グループにおける、CPS事業の実現を支えるソフトウェア開発環境構築への取り組みについて述べた。ウォーターフォールモデルの開発を支援するためのソフトウェア開発環境の提供から始まり、アジャイル開発やDevOpsの実践を支援するための独自の機能追加として、開発成果物を管理するアーティファクト管理ツールやOSS利用の管理を効率化するソフトウェア資産管理ツールなどを強化してきた。現在は、東芝グループ全体で2万人がこのソフトウェア開発環境を利用しており、利用者に対して幅広いサービス提供と利用サポートを行っている。

今後も、ソフトウェア開発環境について、実践利用を通じた効果の確認と先進技術に対応した継続的な改善を行い、東芝グループへの展開を続けていく。

文献

- (1) 山元和子, ほか. OSSを活用したソフトウェア開発ツールチェーンの構築. 東芝レビュー. 2012, **67**, 8, p.27-30.
- (2) 山中美穂, ほか. トレーサビリティを確保してソフトウェア開発を効率化する要求管理ツール. 東芝レビュー. 2018, **73**, 5, p.68-71. <https://www.toshiba.co.jp/tech/review/2018/05/73_05pdf/f02.pdf>, (参照 2020-06-10).
- (3) 小林良岳. ソフトウェア開発資産の再利用とOSSコンプライアンスを両立させる管理システム. 東芝レビュー. 2020, **75**, 1, p.52-53. <https://www.toshiba.co.jp/tech/review/2020/01/75_01pdf/r01.pdf>, (参照 2020-06-10).
- (4) Fowler, M. "Continuous Integration". martinowler. <<https://martinfowler.com/articles/continuousIntegration.html>>, (accessed 2020-06-10).



前田 尚人 MAEDA Naoto

東芝デジタルソリューションズ(株) ソフトウェアシステム技術開発センター システム・エンジニアリング開発部
情報処理学会会員
Toshiba Digital Solutions Corp.



田原 歩 TAHARA Ayumu

東芝デジタルソリューションズ(株) ソフトウェアシステム技術開発センター システム・エンジニアリング開発部
Toshiba Digital Solutions Corp.



長岡 武志 NAGAOKA Takeshi, Ph.D.

東芝デジタルソリューションズ(株) ソフトウェアシステム技術開発センター システム・エンジニアリング開発部
博士(情報科学) 情報処理学会会員
Toshiba Digital Solutions Corp.



山元 和子 YAMAMOTO Kazuko

技術企画部 ソフトウェア技術センター
共創ソフトウェア開発技術部
Advanced Collaborative Software Development and Technology Dept.