

MATLAB EXPO

Japan

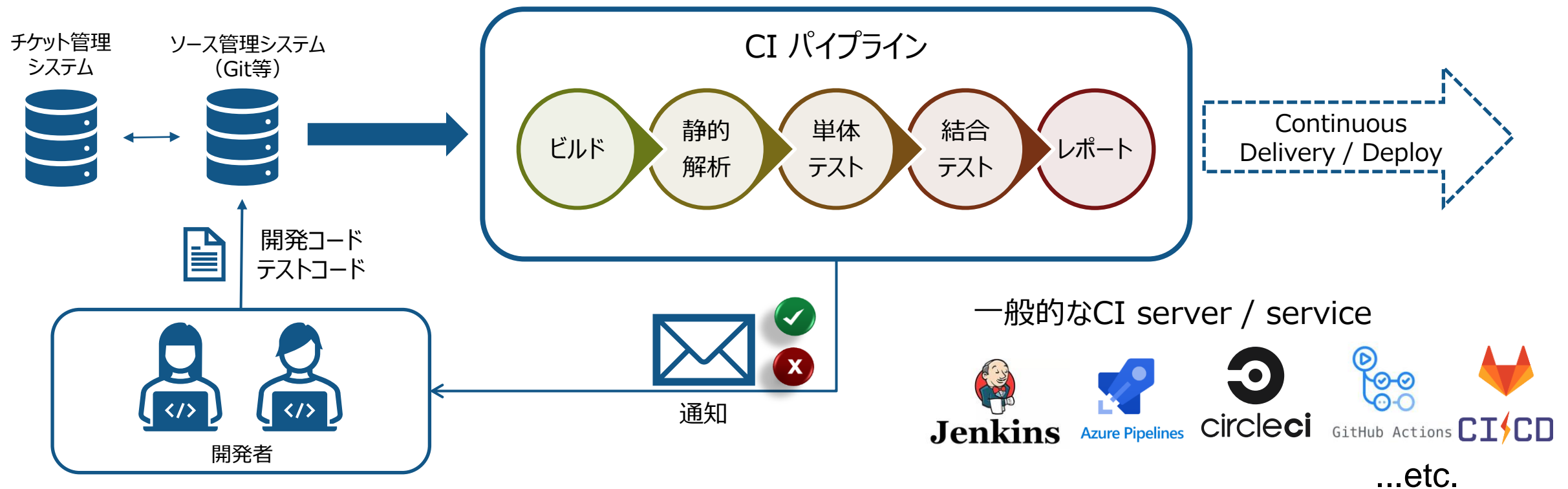
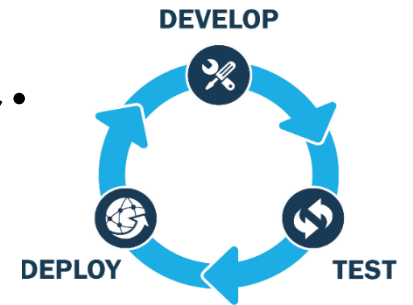
MW-04_モデルベース検証 x CI/CTソリューション(2/2)

MBD×継続的インテグレーションを統合した
アジャイルな開発ワークフロー例のご紹介



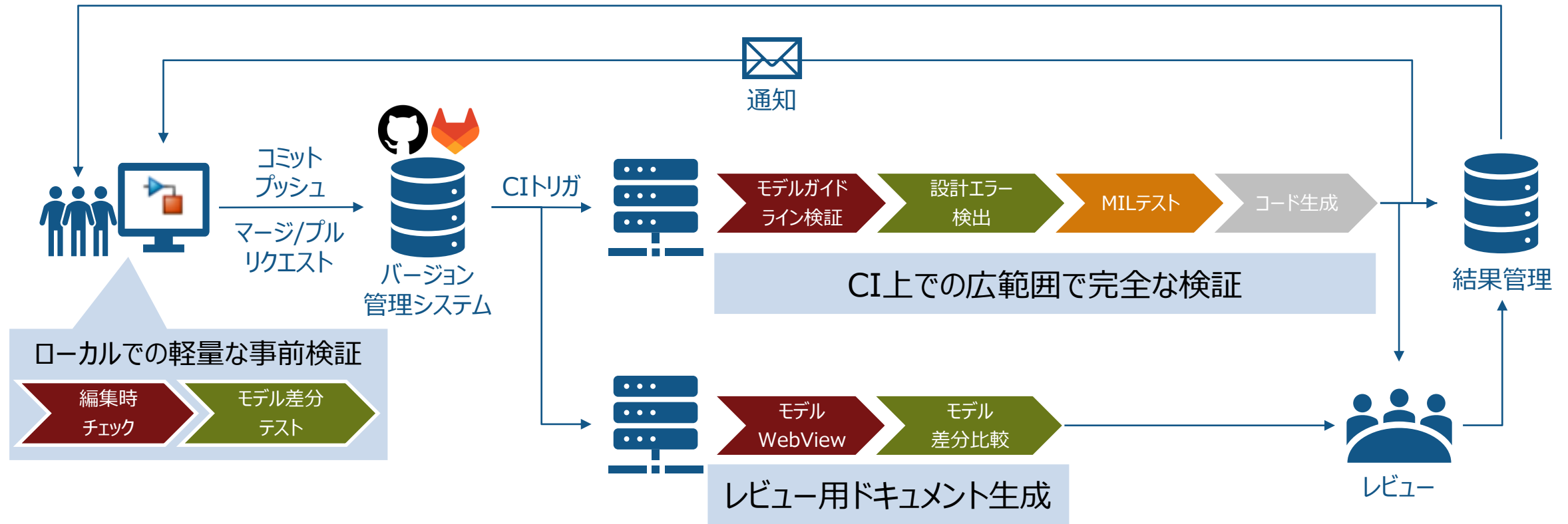
継続的インテグレーション (CI) とは

コードの変更をリポジトリに頻繁にマージし、かつ「定期的・自動的」に「ビルド・テスト」を行うことにより**問題を早期に発見**し、開発の効率化や省力化、納期の短縮などを図る手法であり、**アジャイル開発のベストプラクティス**です



MBD + 継続的インテグレーション

チケット・Issueのアサイン

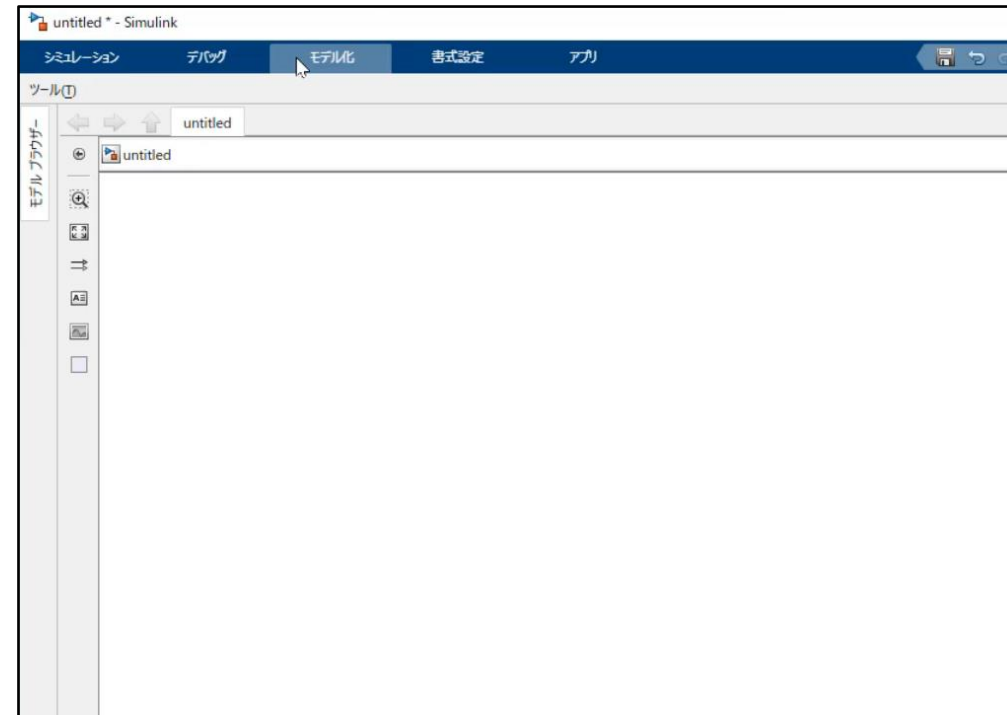
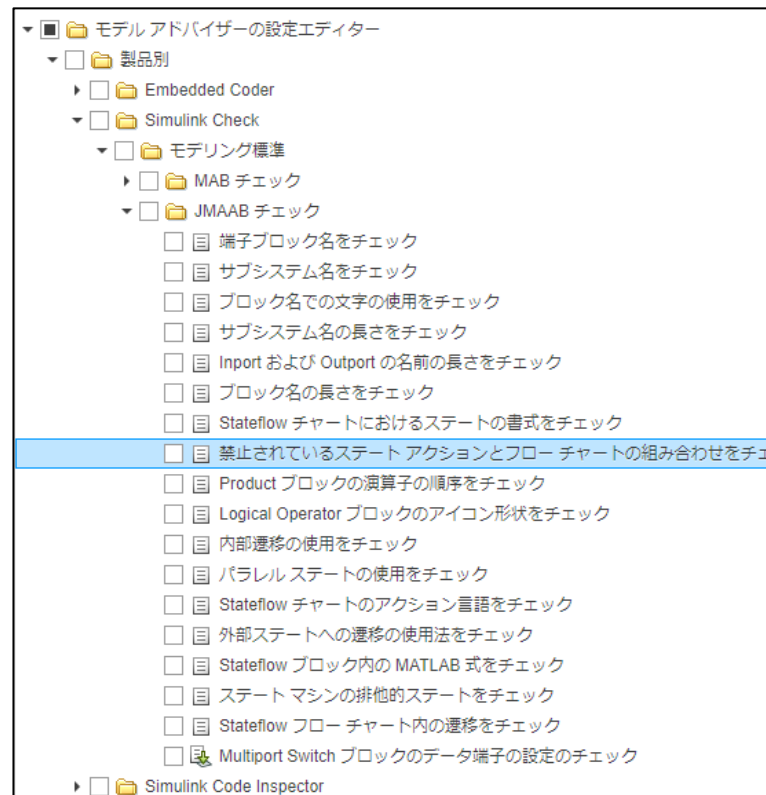


CIを活用してアジャイルなMBDワークフローを実現可能

ローカルでの事前検証

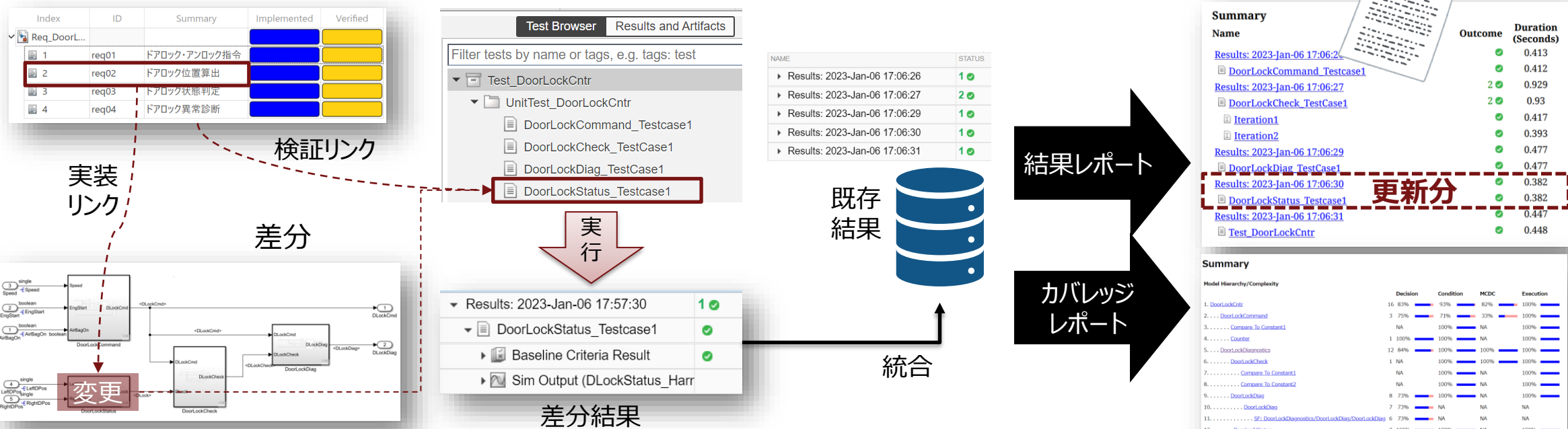
編集時チェックによるモデリングチェック

- モデルの編集とモデリングチェックを同時におこなうことができる
 - ブロックを配置したタイミングなど、SimulinkのEditor上でチェックが自動的に実施される
 - モデリングチェックをより早期に実施できる



ローカルでの事前検証 モデル差分テスト

- テストケースごとに結果を管理し、既存結果と統合したレポートを生成
 - 設計変更時は、関連するテストケースのみ再実行し、既存の結果と統合してレポートする
 - 累積カバレッジ機能と組み合わせることで、モデル全体での結果をレポート可能
 - Ex. 変更したサブシステムのみをテストし、モデル全体でレポート可能



モデルの差分箇所（サブシステム）のみを自動テストするソリューション

ローカルでの事前検証 デモ

The screenshot displays two windows from the MATLAB R2023a environment. The left window shows a test results summary table with columns for '概要名前' (Summary Name), '結果' (Result), and '持続時間(秒)' (Duration in seconds). The right window shows the 'プロジェクト' (Project) browser with a tree view of files and folders.

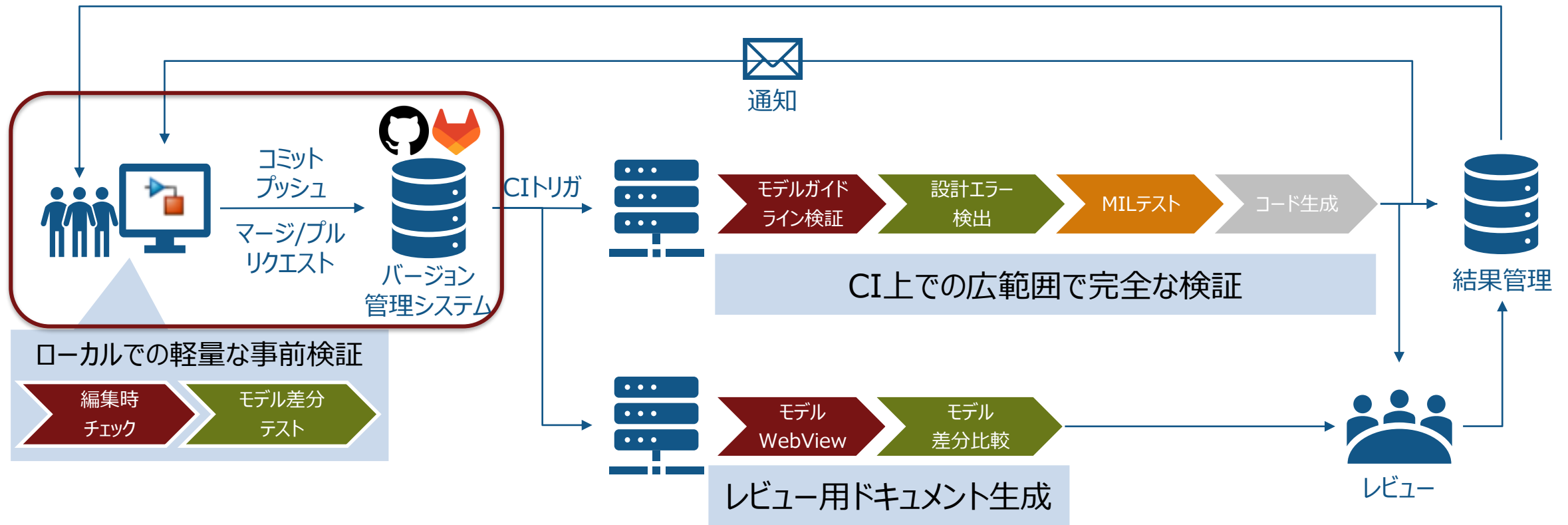
概要名前	結果	持続時間(秒)
結果: 2023-Mar-31 14:13:45	成功	7.342
DoorLockCommand_Testcase1	成功	7.342
Results: 2023-Mar-27 16:10:33	成功	1.713
DoorLockCheck_TestCase1	成功	1.716
Iteration1	成功	0.728
Iteration2	成功	0.838
Results: 2023-Mar-27 16:10:46	成功	1.596
DoorLockDiag_TestCase1	成功	1.595
Results: 2023-Mar-27 16:20:29	成功	0.821
DoorLockStatus_Testcase1	成功	0.822
結果: 2023-Mar-31 14:14:00	成功	7.343
Test_DoorLockCntr	成功	7.343

The project browser on the right shows a hierarchical structure including folders like 'Design', 'test', 'Doc', 'Requirements', 'resources', 'Test_Results', 'scv_images', 'Utility', and 'setup'. The 'Test_Results' folder is expanded, showing various report files, including 'Test_DoorLockCntr_IntegratedReport.pdf'.

編集時チェック、差分テストなど効率のよい軽量な検証をデモします

モデルベースデザイン + 継続的インテグレーション

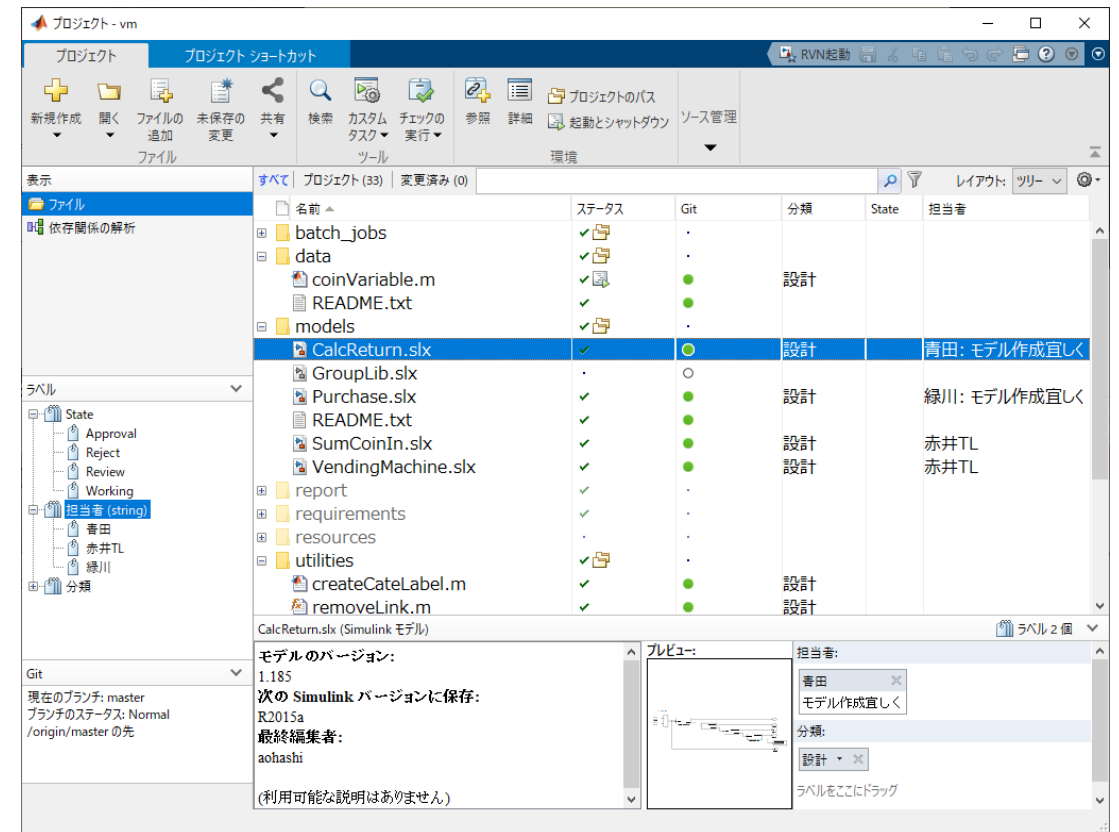
チケット・Issueのアサイン



CIを活用してアジャイルなモデルベースデザインのワークフローを実現可能

プロジェクトとは

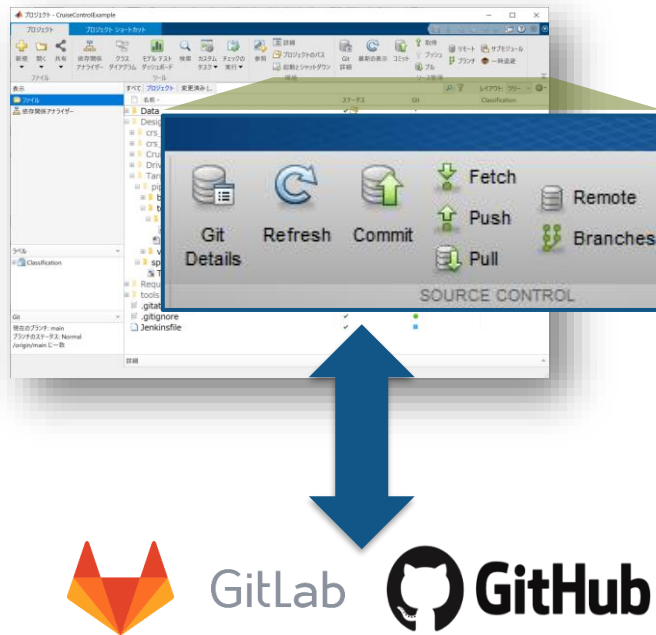
開発における様々な種 of ファイルをパッケージ管理する、チーム開発のための機能



※R2018bまではSimulinkプロジェクトと呼称

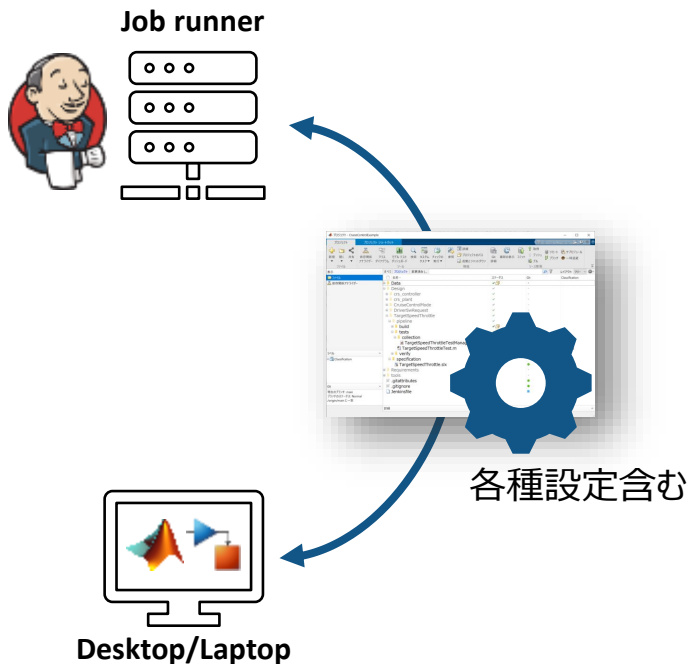
プロジェクトを用いる利点

ソース管理との統合



MATLAB・Simulinkから
直接バージョン管理操作

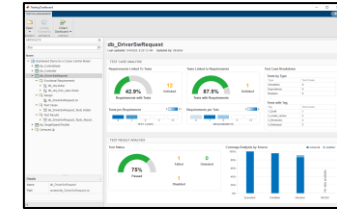
CI環境との統合



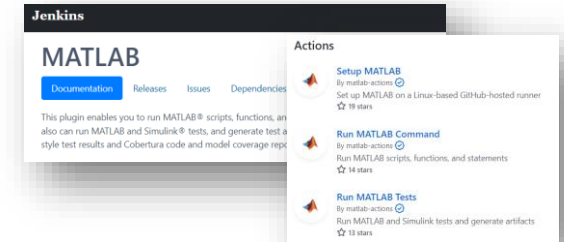
ローカルとサーバ上の
環境差異の縮小

標準機能の活用

■ モデルテストダッシュボード



■ CIプラットフォーム向けアドオン



■ CI/CDオートメーションサポートパッケージ



CIに有用な新機能を
利用する前提条件

プロジェクトとは Source Control Integration



ソース管理コマンド

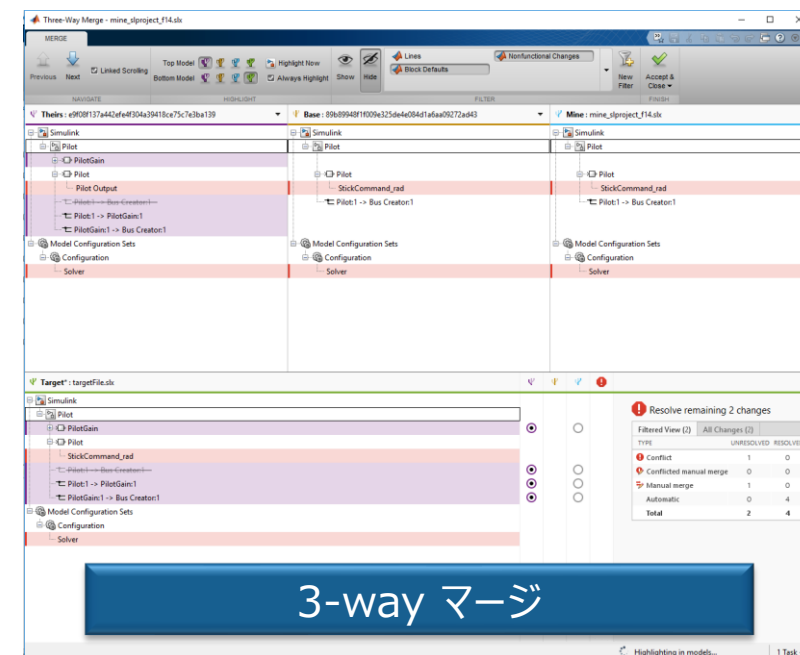
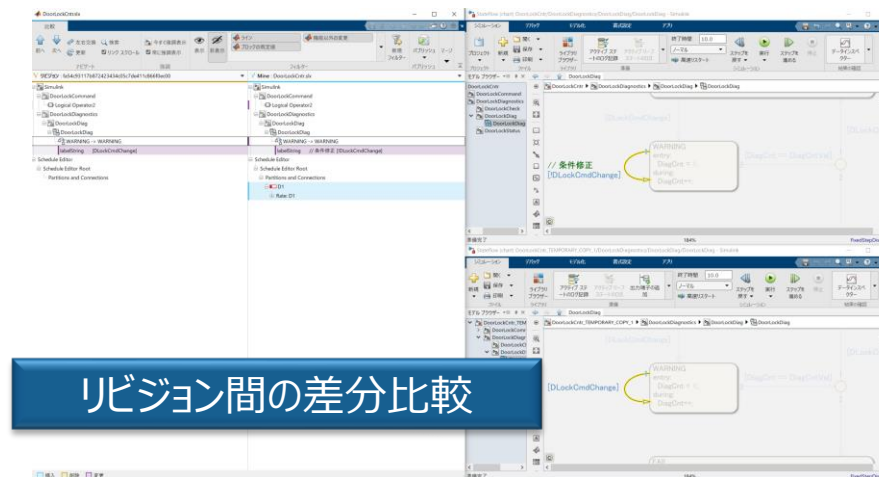
クラウドベースのシステムも対応

ブランチ構成表示

ブランチ・タグ作成

ブランチ	Author	ID
develop_DED	Shintaro Hayashi	f2bc5f1e210933f88ca2f83d07adf4c06add9747
origin/master	Shintaro Hayashi	f2bc5f1e210933f88ca2f83d07adf4c06add9747
HEAD	Shintaro Hayashi	f2bc5f1e210933f88ca2f83d07adf4c06add9747
TEST	Shintaro Hayashi	f2bc5f1e210933f88ca2f83d07adf4c06add9747

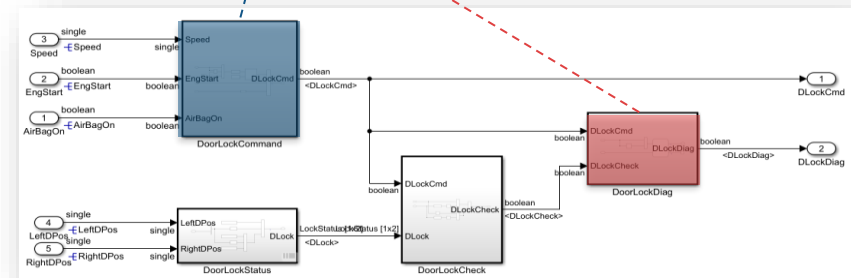
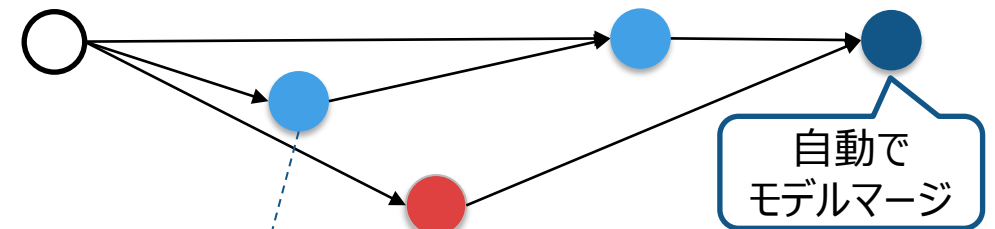
プロジェクトとは compare / diff / merge



プロジェクトとは compare / diff / merge



- オートマージ R2020b**
 - ブランチマージの際、同一モデルファイルでも異なるサブシステムの変更のみの場合は競合せず、自動でマージされる



それぞれのブランチで異なるサブシステムを編集

プロジェクトとは Automatic Environment Configuration



よく使う機能やドキュメントへのショートカット

設定例

設定例

MATLABパス設定

優先順位による潜在的な問題の確認

“slprj” “CodeGen” 場所の設定

チーム内の共通設定

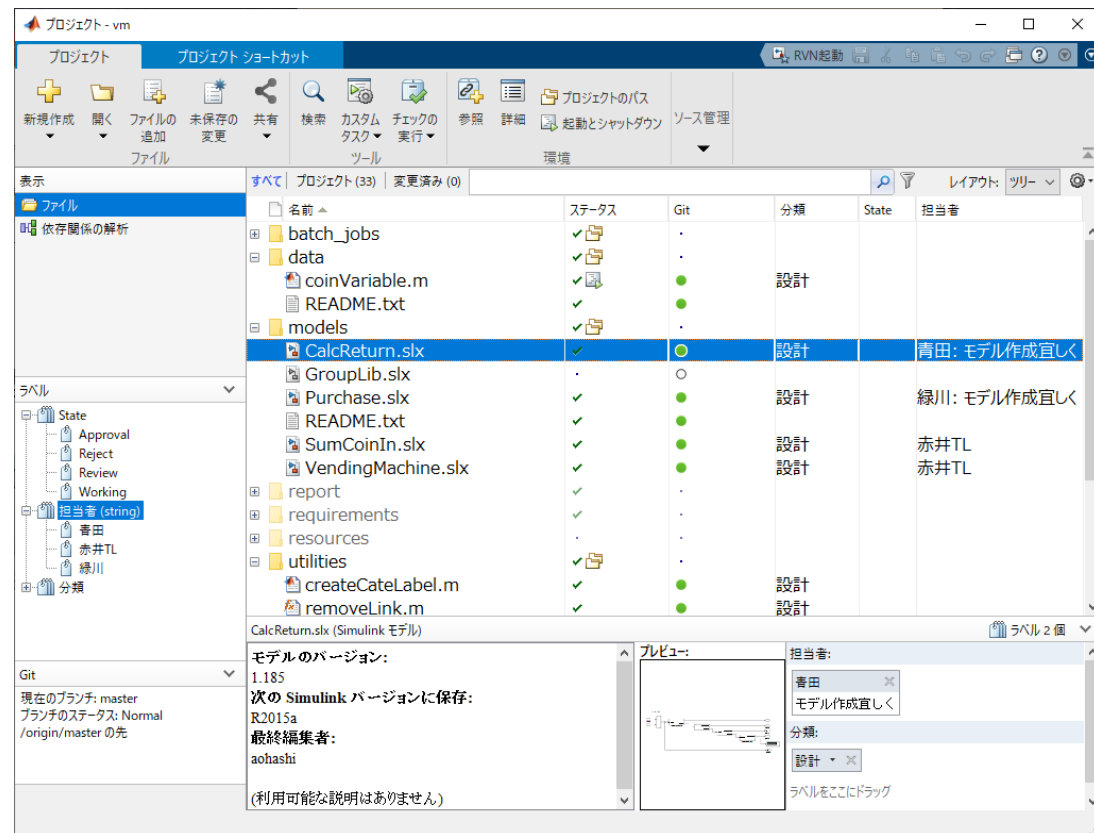
モデルテンプレート設定

プロジェクト固有の設定を起動時に実施
→再利用性、移植性向上

プロジェクトとは 参考リンク

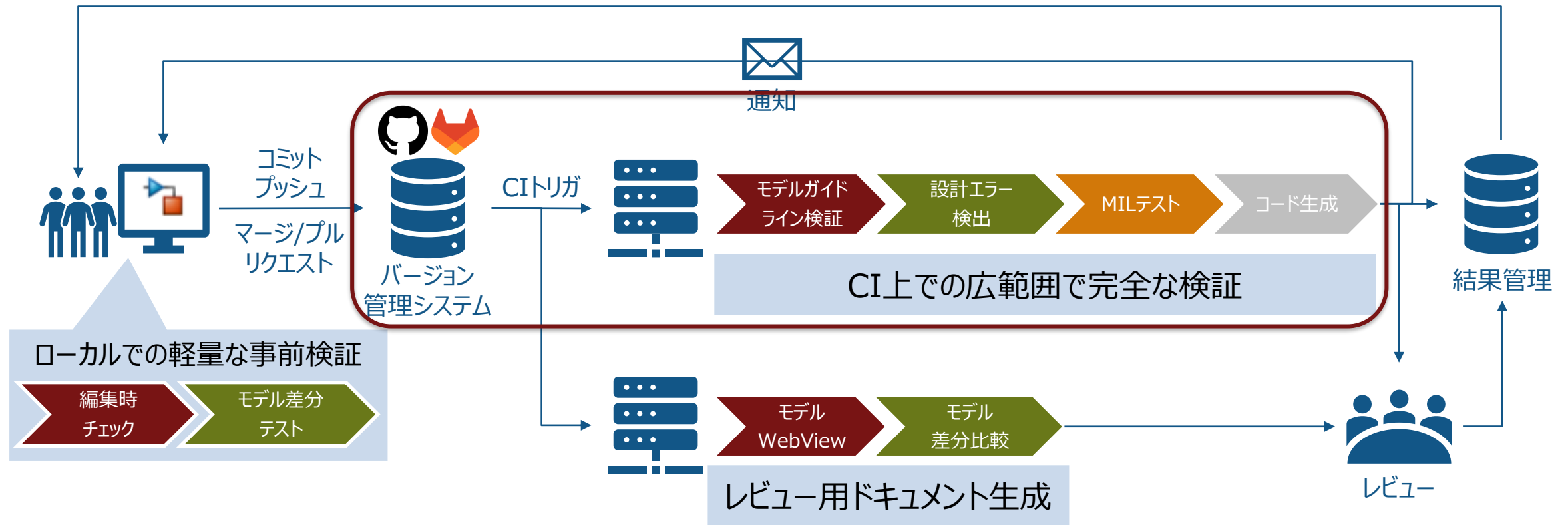
■ 参考リンク

- [プロジェクトのソース管理](#)
- [プロジェクトの派生ファイルの使用](#)
 - .mex等の派生ファイルについて
- [Gitソース管理の設定](#)
 - リポジトリのモデルサイズ低減など
- [シミュレーションを高速化するためのキャッシュファイルの共有](#)
- 比較・マージ
 - [ファイルやフォルダーの比較とファイルのマージ](#)
 - [モデルの比較](#)
 - [API: visdiff](#)
 - [MATLAB を比較とマージに使用するための外部ソース管理のカスタマイズ](#)



モデルベースデザイン + 継続的インテグレーション

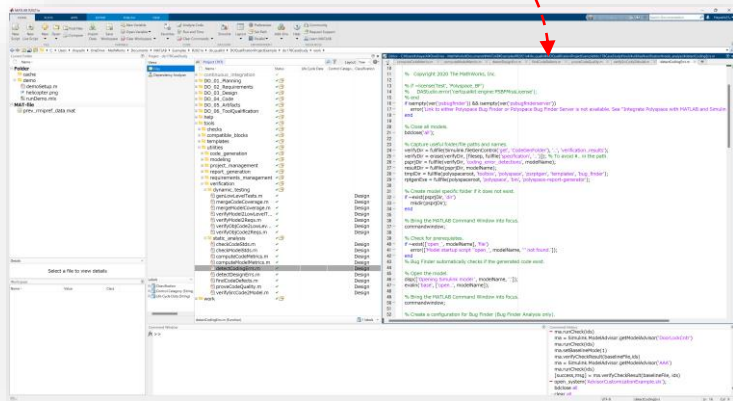
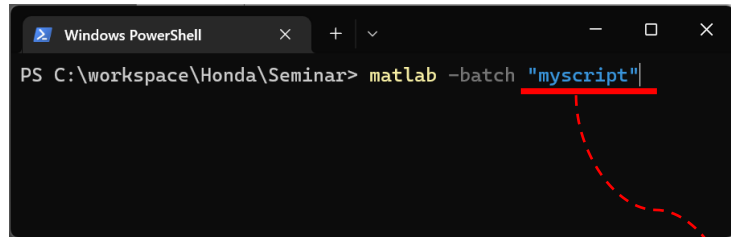
チケット・Issueのアサイン



CIを活用してアジャイルなモデルベースデザインのワークフローを実現可能

CIツールとMATLAB・Simulinkの統合

- システムコマンドによるMATLAB実行
 - 実行するスクリプトを指定して、外部からバッチモードでMATLAB実行が可能



- パイプラインの実行
 - 外部からのMATLAB実行をステージごとに行うことでパイプラインを構成

```

定義
Pipeline script
Script
1 node {
2   stage('preprocess'){
3     bat ''
4     mkdir "AMV_tmp"
5     mkdir "AMV_report"
6   }
7
8   stage('ModelStyleCheck'){
9     bat ''
10    matlab -batch "modelStaticCheck"
11    echo 成功: !ERRORLEVEL!
12    if not !ERRORLEVEL! == 0 {
13      echo 失敗: !ERRORLEVEL!
14    }
15  }
16
17  }
18
19  stage('DesignErrorDetection'){
20    bat ''
21    matlab -batch "modelFunctionalTesting"
22    echo 成功: !ERRORLEVEL!
23    if not !ERRORLEVEL! == 0 {
24      echo 失敗: !ERRORLEVEL!
25    }
26  }
27
28  }
29 }
    
```



モデルのガイドラインチェック
matlab -batch "modelStaticCheck"

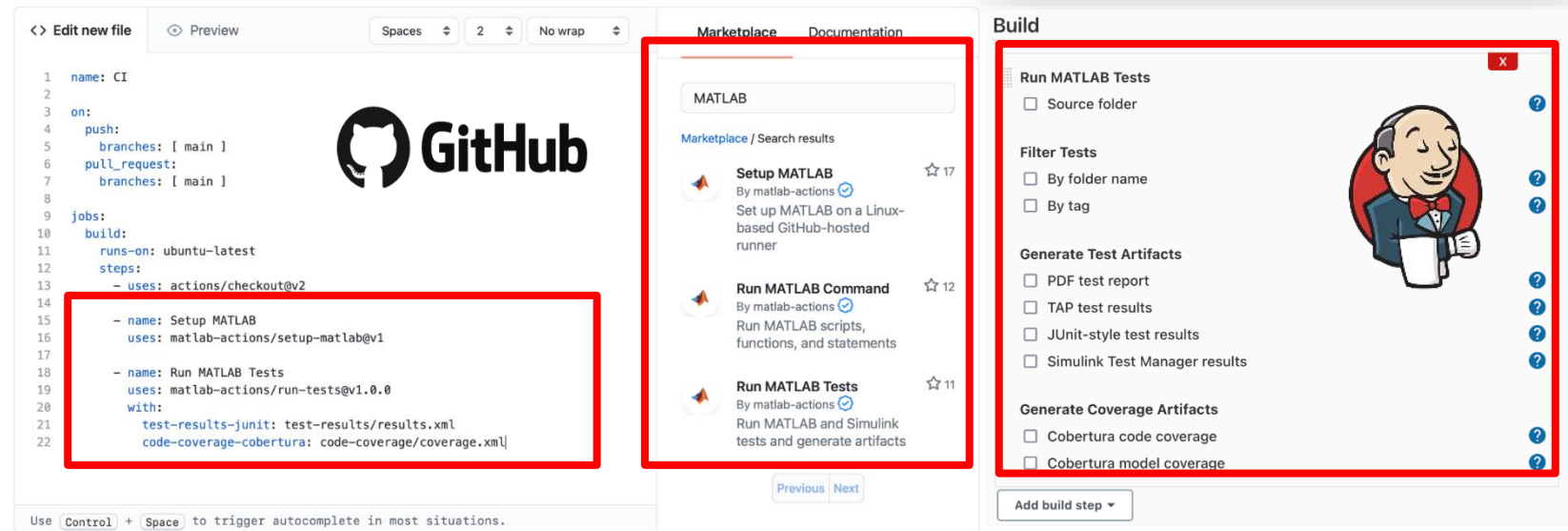
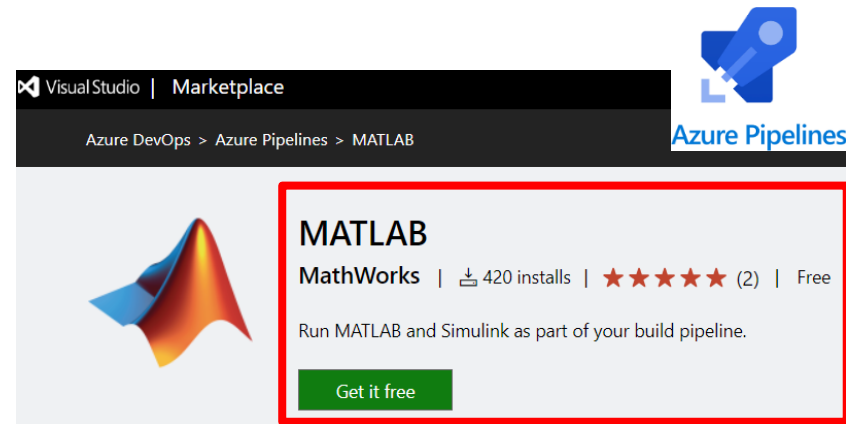
モデルのシミュレーションテスト
matlab -batch "modelFunctionalTesting"

Stage View

	Declarative: Checkout SCM	Model Static Analysis	Unit Model Testing
Average stage times: (Average full run time: ~18min 26s)	6s	10min 26s	3min 37s
Oct 22 10:12 1 commit	7s	10min 37s	4min 56s
Oct 22 11:14 1 commit	6s	11min 33s	5min 7s
Oct 22 10:51 1 commit	6s	13min 34s	180ms
Oct 22 10:18 1 commit	7s	14min 41s	187ms

各種CIプラットフォーム向けアドオンを提供

- MATLAB環境のセットアップ
- 任意のコマンド実行
- テストファイルの実行
 - テスト仕様ファイル(.mldatx)
 - MATLABユニットテスト(.m)
- テスト結果ファイルの生成
 - TAP
 - JUnit
 - Cobertura
- …etc.



CI プラットフォーム上での MATLAB との継続的インテグレーション

https://jp.mathworks.com/help/matlab/matlab_prog/continuous-integration-with-matlab-on-ci-platforms.html

CIでの完全で広範囲な検証 デモ

The screenshot shows a GitHub Actions workflow run for the repository 'shayashi1827 / exCiworkflowExample_WithIncrementalTest'. The workflow is named 'DoorLockCtr_Develop_ci.yml' and was triggered by a push. The workflow consists of four jobs: 'setup' (16s), 'testing' (14s), 'build', and 'package'. The 'testing' job is currently running and contains a 'verify' step. The 'Annotations' section shows a warning about deprecated Node.js actions.

Triggered via push now

Job	Status	Total duration	Artifacts
shayashi1827 pushed -> fd8f3a3 Feature_EXP02023	Queued	-	-

DoorLockCtr_Develop_ci.yml
on: push

Jobs

- ✓ setup
- verify
- testing

Run details

- Usage
- Workflow file

Annotations

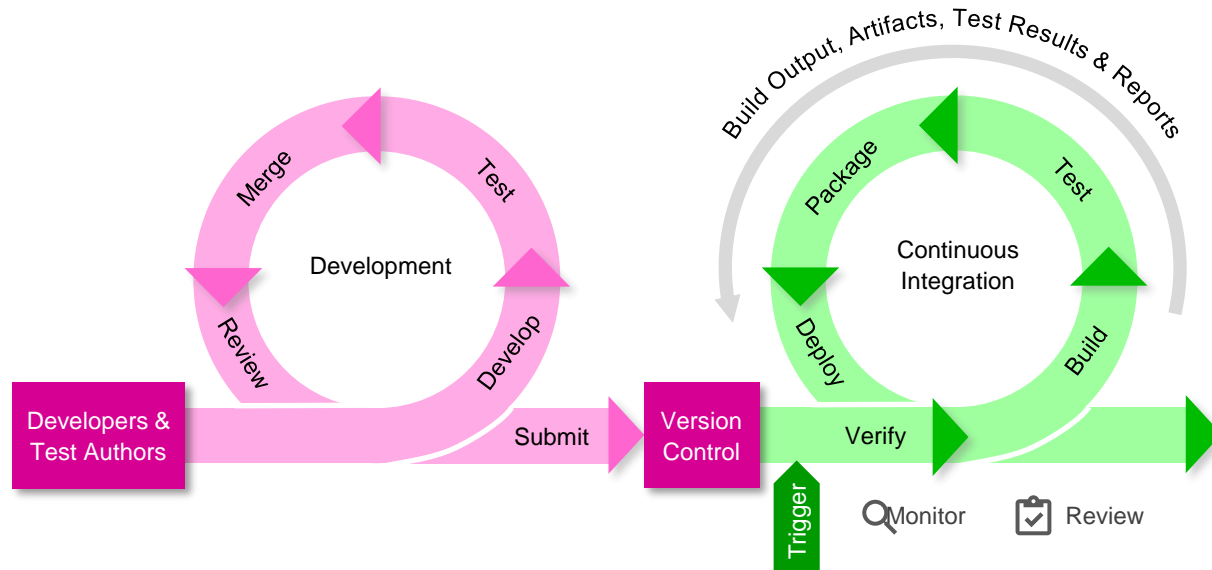
1 warning

⚠️ setup
Node.js 12 actions are deprecated. Please update the following actions to use Node.js 16: actions/checkout@v2. For more information see: <https://github.blog/changelog/2022-09-22-github-actions-all-actions-will-begin-running-on-node16-inst->
[Show more](#)

GitHub等のCIプラットフォームとSimulinkの連携についてデモいたします

Continuous Integration for Verification of Simulink Models

サンプルを用いてCIワークフローについて、環境設定例や実行方法を学べます



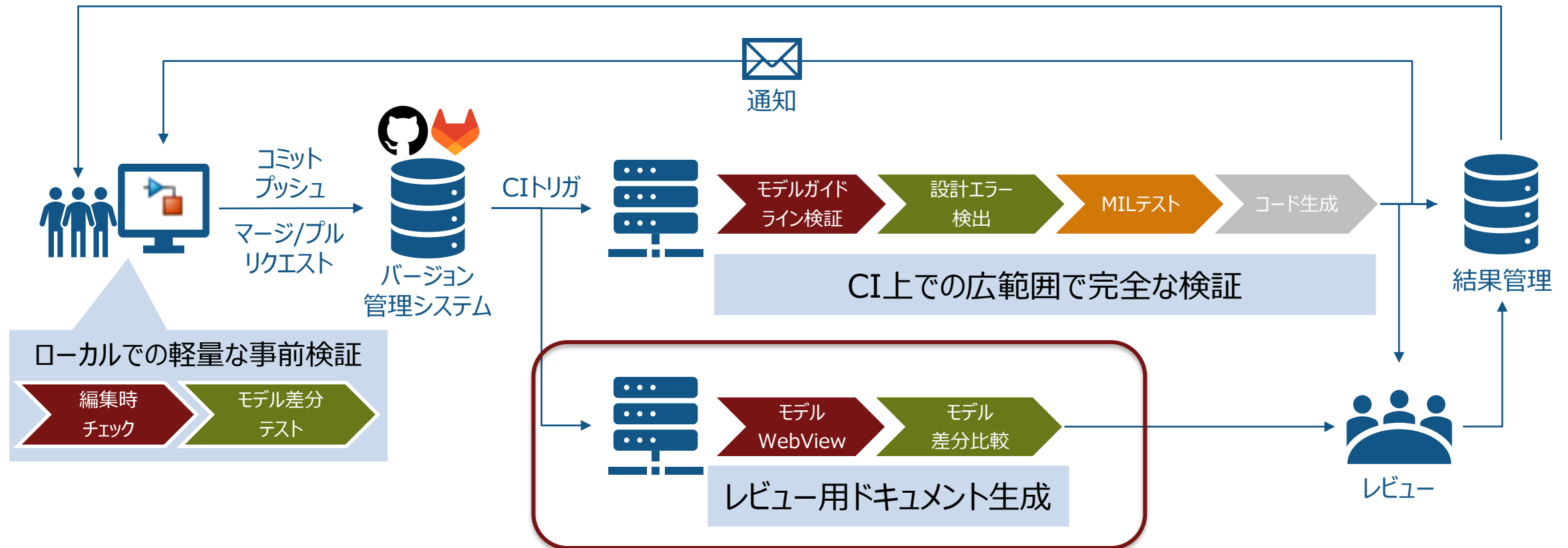
【サンプルの流れ】

1. 環境のセットアップ (MATLAB / バージョン管理 / CI)
2. コミット→CI実行
3. ローカルでのデバッグ及び修正
4. 再度コミット→CI実行
5. マージリクエスト

CI example	Part.1 link	Part.2 link
バージョン管理	GitLab	GitLab
CIツール	Jenkins	GitLab CI/CD
対象	システムテストモデル ソフトウェア統合モデル	ソフトウェア コンポーネントモデル
主なビルド内容	<ul style="list-style-type: none"> ・ガイドラインチェック ・コード生成 ・SIL 	<ul style="list-style-type: none"> ・ガイドラインチェック ・コード生成 ・MIL ・レポート
必要な ツールボックス (MATLAB/Simulink /Stateflow以外)	<ul style="list-style-type: none"> ・Embedded coder ・V and V一式 ・Automated Driving Toolbox ・Sensor Fusion and Tracking ..etc 	<ul style="list-style-type: none"> ・Embedded coder ・V and V一式

モデルベースデザイン + 継続的インテグレーション

チケット・Issueのアサイン



CIを活用してアジャイルなモデルベースデザインのワークフローを実現可能

Pull/Merge Requestとは

- 承認者は変更点などをレビューします
- コード差分はシステム上で確認できます
- 依頼者のコメントやそれに対する返信など、システム上でのやり取りが可能です
 - コードの特定箇所にコメントなども可能
- バイナリファイルはシステム上で差分確認不可
 - モデルを閲覧するためにダウンロードする手間が発生
 - 開発者でなくてもSimulinkインストール必要

The screenshot shows a GitHub Pull Request titled "Changes to my feature branch" for the repository "NewFeature (#1)". It displays a diff for the file "custom_tasks/analyzeModelFiles.m". The diff shows a change in line 3, where a comment "% Add comment" was added. Below the diff, two binary files are listed: "models/AnalogControl.slx" (731 Bytes) and "models/f14_airframe.slx" (583 Bytes). Both files are marked as "Binary file not shown". A red dashed box highlights the binary files section, with a red arrow pointing from the text "バイナリファイルはシステム上で差分確認不可" in the list above.

引用

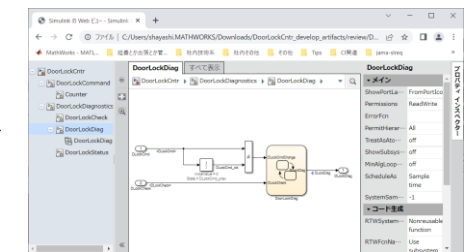
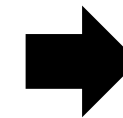
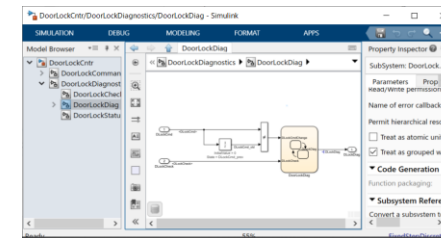
Simulink Model Comparison for GitHub Pull Requests
<https://www.youtube.com/watch?v=PNX7f0kPSYg>

Simulinkはプル（マージ）リクエストを用いたレビューワークフローと相性悪い

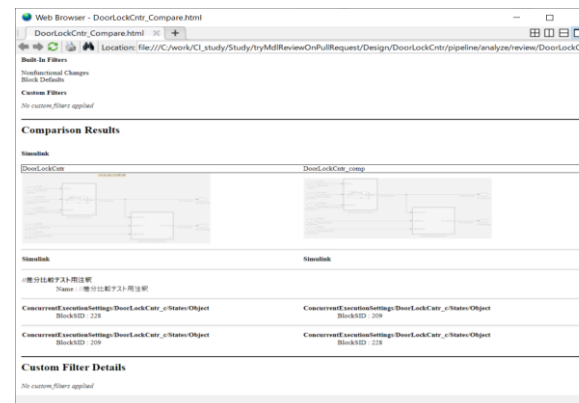
CIを用いたモデルレビュー用ソリューション

Pull Requestを作成した際に
以下を実行するパイプラインを定義します

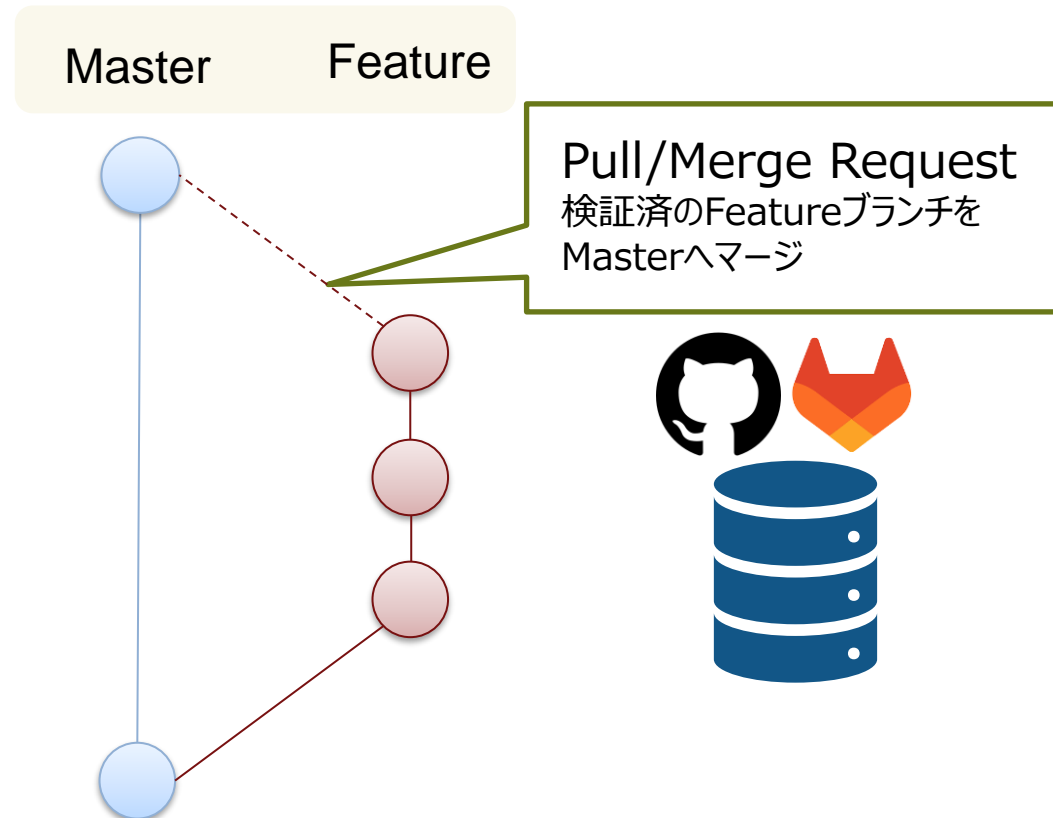
①モデルWebView生成
マージされたモデルのWebView



②モデル差分レポート生成
直前コミットとの差分比較



ブラウザで閲覧
容易にレビュー可能



ブラウザ上でのレビューに必要な
ドキュメントをリクエスト時に生成

CIを用いたモデルレビュー用ソリューション デモ

fixed #15 #16

shayashi1827 wants to merge 1 commit into master from Feature_EXPO2023

Conversation 0 Commits 1 Checks 5 Files changed 6

shayashi1827 commented now

ドアロック位置判定修正

#15

Add more commits by pushing to the Feature_EXPO2023 branch on shayashi1827/exCWorkflowExample_WithIncrementalTest

Some checks haven't completed yet
1 queued and 5 successful checks

- Sample for Model Review in PullRequest / setup (pull_request_target) Queued — Waiting to run this c...
- Model verification example for the DriverSwRequest model / setup (push) Successful in 16s
- Model verification example for the DriverSwRequest model / verify (push) Successful in 5m
- Model verification example for the DriverSwRequest model / testing (push) Successful in 3m
- Model verification example for the DriverSwRequest model / build (push) Successful in 2m
- Model verification example for the DriverSwRequest model / package (push) Successful in 6m

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request

Code Generation Report

Content

- Summary
- Subsystem Report
- Code Interface Report
- Traceability Report
- Static Code Metrics Report
- Code Replacements Report
- Code Assumptions

Code

Main file

- ert_main.c
- Model files
- DoorLockCtrl.c
- DoorLockCtrl_getrate.h
- DoorLockCtrl_types.h
- Data files
- DoorLockCtrl_data.c
- Utility files
- matypes.h

Simulink Web View - Created by: X

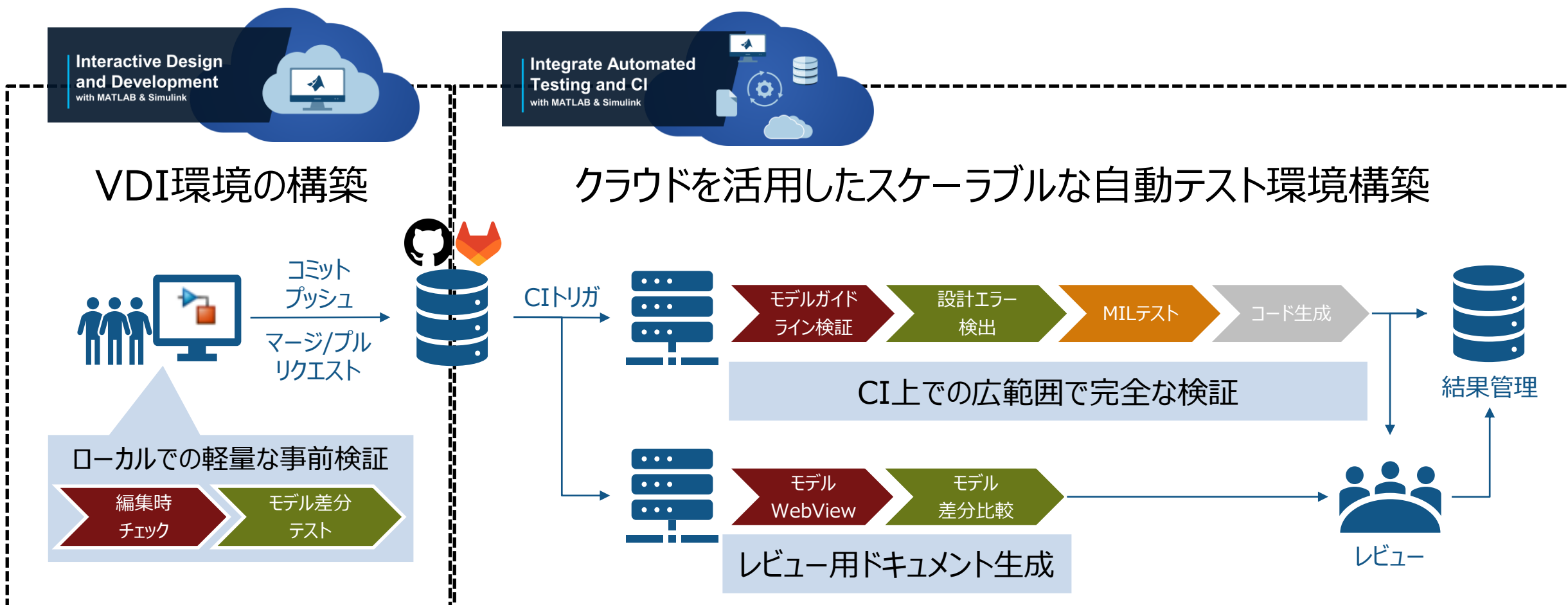
DoorLockCommand

Main

- ShowPortLabels
- Permissions
- ErrorFun
- ParamHierarchical
- FreezeAutomatic
- ShowSubsystemOff
- MinLogLogOccur
- ScheduleAs
- SystemSampleTime
- Code Generation
- RTWSystemCode
- NonExecutable
- RTWUseNameOpts
- Use subsystem name
- RTWUseNameOpts
- Auto
- FunctionInterface
- FunctionWithSiga

プル（マージ）リクエストの際にモデル差分レポート等の
レビュー用ドキュメントを生成するソリューションをデモいたします

クラウドへの拡張



クラウドでもオンプレミス環境と同等の開発環境の提供を目指しています

MATLAB EXPO

Japan

Thank you !

