MATLAB EXPO 2019

Software Development Practices within MATLAB

Elmar Tarajan (Consulting)
elmar.tarajan@mathworks.de

What are your software development concerns?

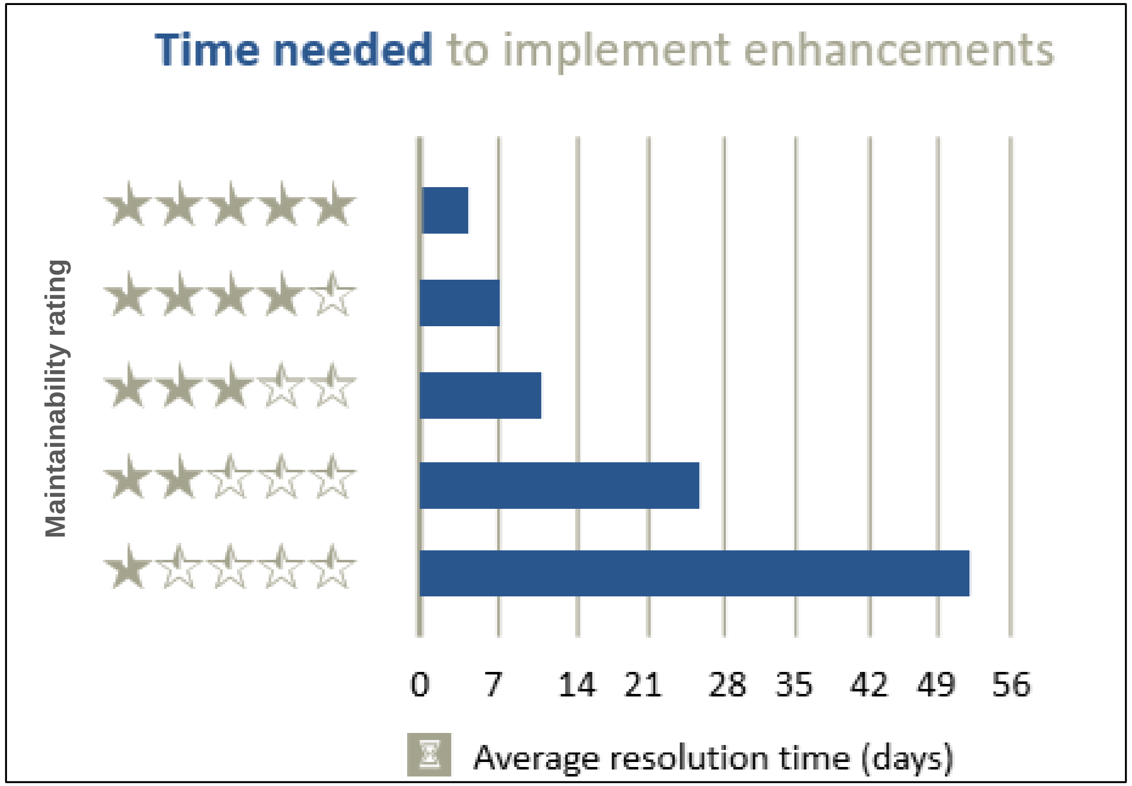
- Accuracy
- Software Speed
- Development Time
- Cost
- Compatibility
- Documentation
- Reusability
- Effective Testing
- Integration
- Ease of Collaboration
- Legacy Code
- Liability
- Maintainability
- Model Risk
- Robustness
- Developer Expertise
- Software Stack Complexity
- …?

Software development practices can help

Treat your software like an asset → reuse it

Developers often spend 4X the effort to maintain vs build software

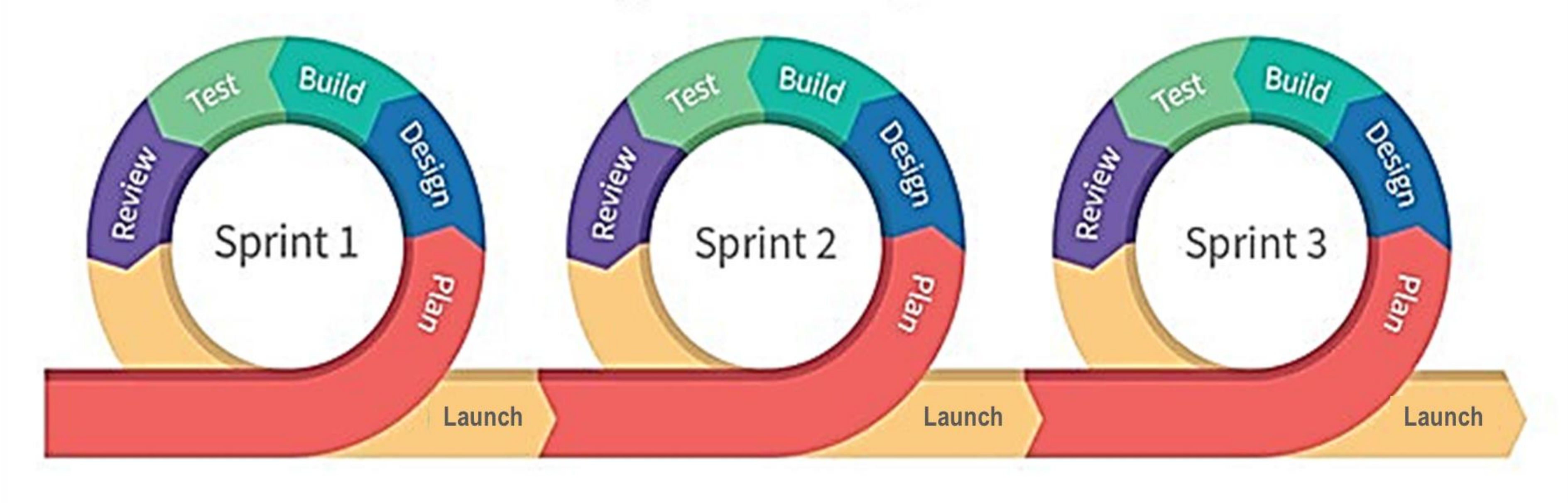
…but this doesn't need to be true!



Journal paper: "*Faster issue resolution with higher technical quality of software*", Software Quality Journal, 201100

Software development practices can help

- Software development approaches like Agile help improve code quality
- The tools and practices we discuss today support Agile development



Agenda

▲	Managing your code
	Tracking code changes and co-authoring workflows
	Writing better, robust, and portable code
	Testing and maintaining your code
	Summary

How do you currently manage your files and paths?

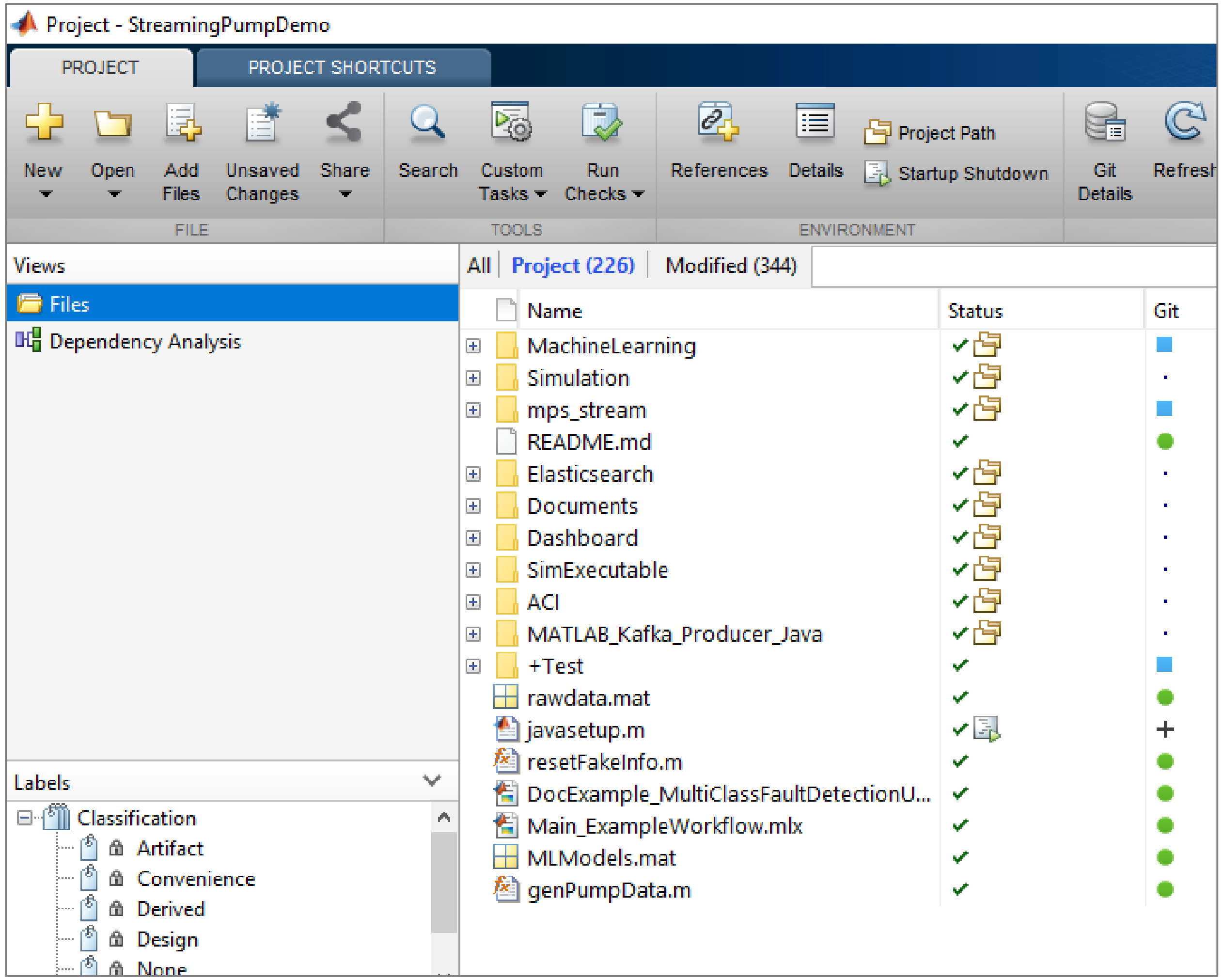
- One big folder of files?
- Many folders of files?
- Organize your code in packages?
- Manual path management?

Successful collaborative development requires …

- Same source code, tests, documentation, requirements, compiler…
- Consistent, shared environment
- Integration with source control

Projects (MATLAB + Simulink Projects)

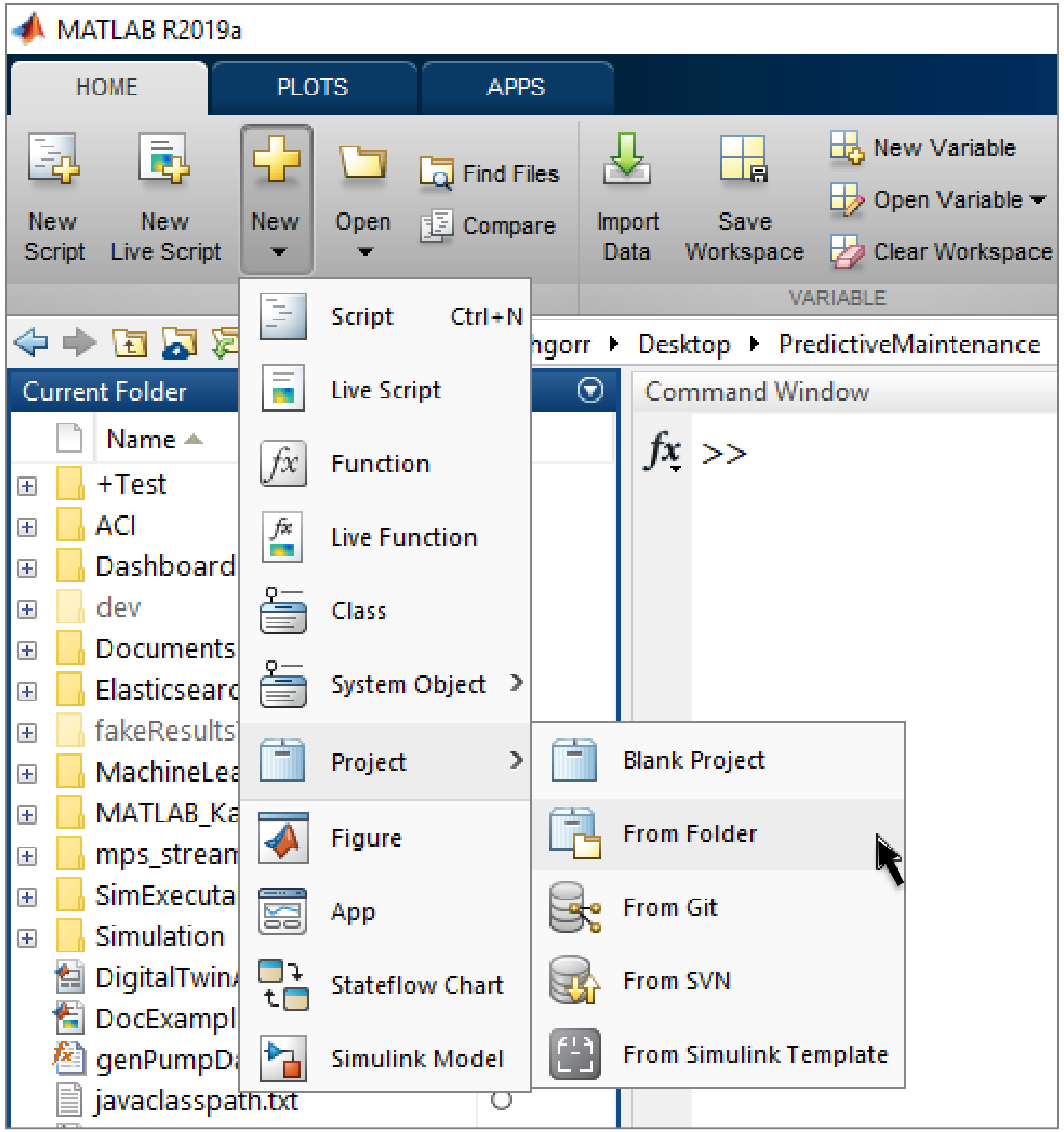
- Manage your files and path
- Analyze file dependencies
- Function refactoring
- Run startup & shutdown tasks
- Create project shortcuts
- Label and filter files
- Integrate source control



Name	Status	Git
MachineLearning	✓	■
Simulation	✓	·
mps_stream	✓	■
README.md	✓	●
Elasticsearch	✓	·
Documents	✓	·
Dashboard	✓	·
SimExecutable	✓	·
ACI	✓	·
MATLAB_Kafka_Producer_Java	✓	·
+Test	✓	■
rawdata.mat	✓	●
javasetup.m	✓	+
resetFakeInfo.m	✓	●
DocExample_MultiClassFaultDetectionU...	✓	●
Main_ExampleWorkflow.mlx	✓	●
MLModels.mat	✓	●
genPumpData.m	✓	●

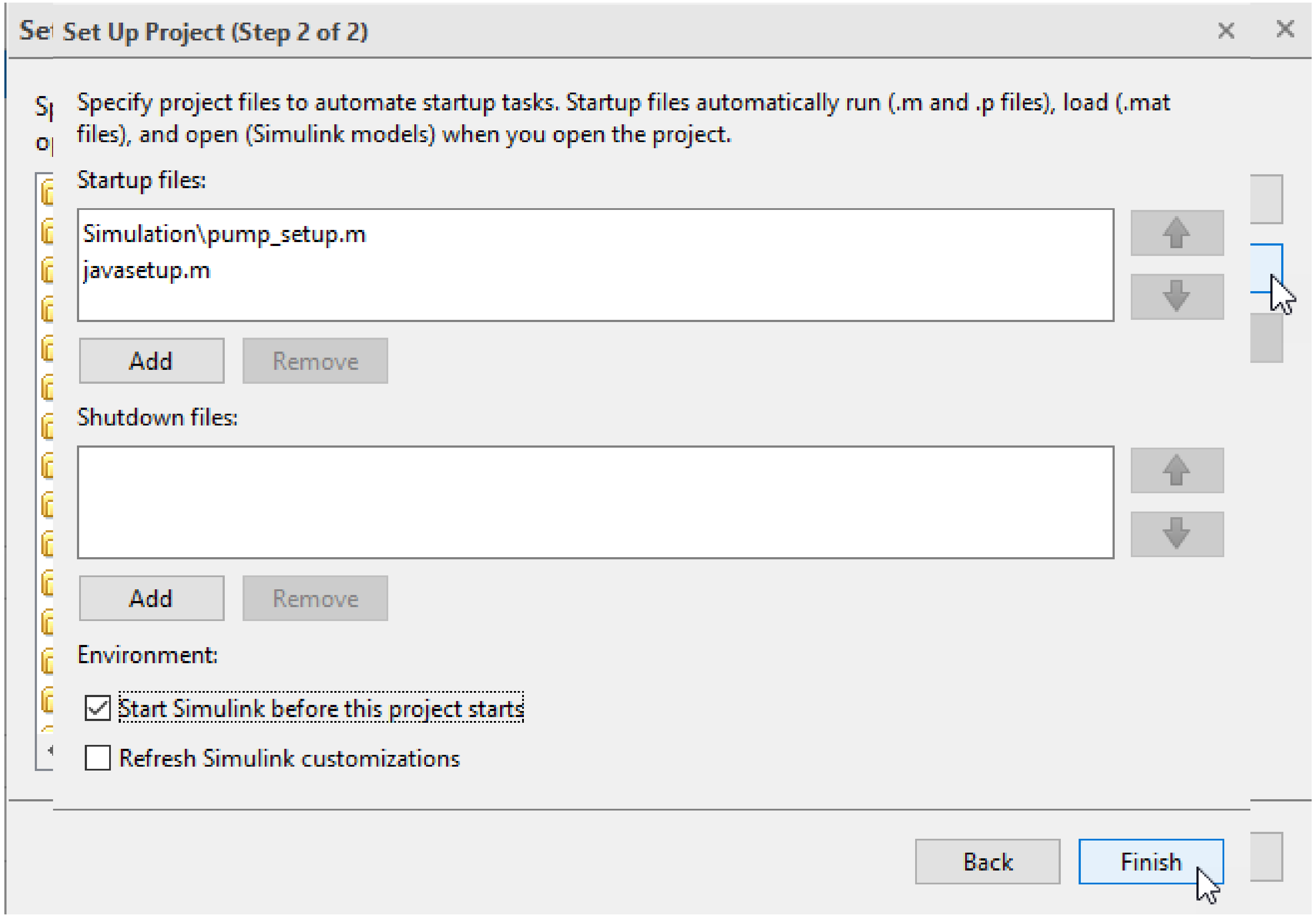
Managing your code with Projects

1. Create project



Managing your code with Projects

1. Create project
2. Set path and startup tasks



Set Up Project (Step 2 of 2)

Specify project files to automate startup tasks. Startup files automatically run (.m and .p files), load (.mat files), and open (Simulink models) when you open the project.

Startup files:

```
Simulation\pump_setup.m
javasetup.m
```

Shutdown files:

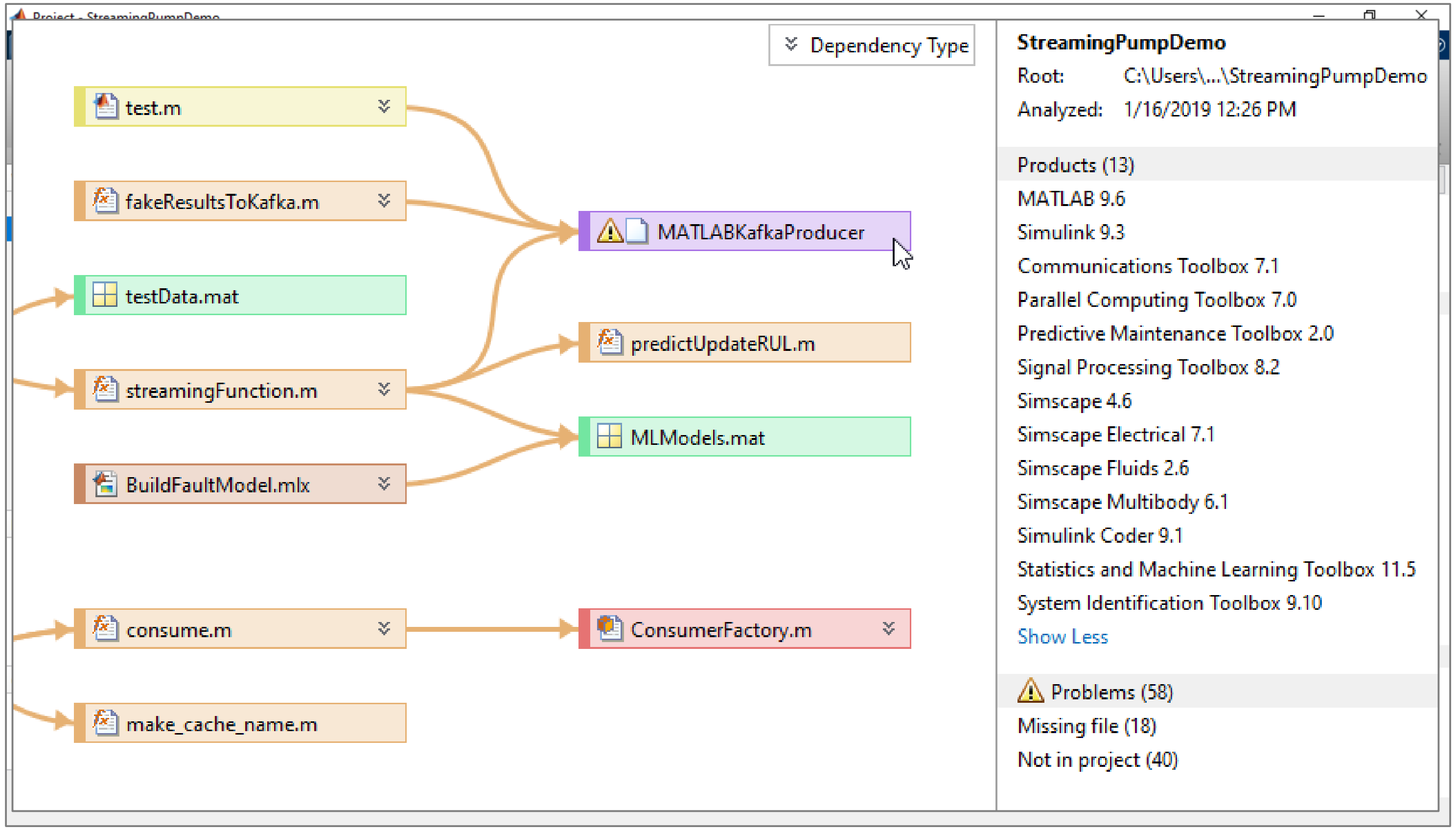
Environment:

☑ Start Simulink before this project starts

□ Refresh Simulink customizations

Managing your code with Projects

1. Create project
2. Set path and startup tasks
3. Explore dependencies



StreamingPumpDemo
Root: C:\Users\...\StreamingPumpDemo
Analyzed: 1/16/2019 12:26 PM

Products (13)
MATLAB 9.6
Simulink 9.3
Communications Toolbox 7.1
Parallel Computing Toolbox 7.0
Predictive Maintenance Toolbox 2.0
Signal Processing Toolbox 8.2
Simscape 4.6
Simscape Electrical 7.1
Simscape Fluids 2.6
Simscape Multibody 6.1
Simulink Coder 9.1
Statistics and Machine Learning Toolbox 11.5
System Identification Toolbox 9.10
Show Less

⚠ Problems (58)
Missing file (18)
Not in project (40)

Managing your code with Projects

1. Create project

2. Set path and startup tasks

3. Explore dependencies

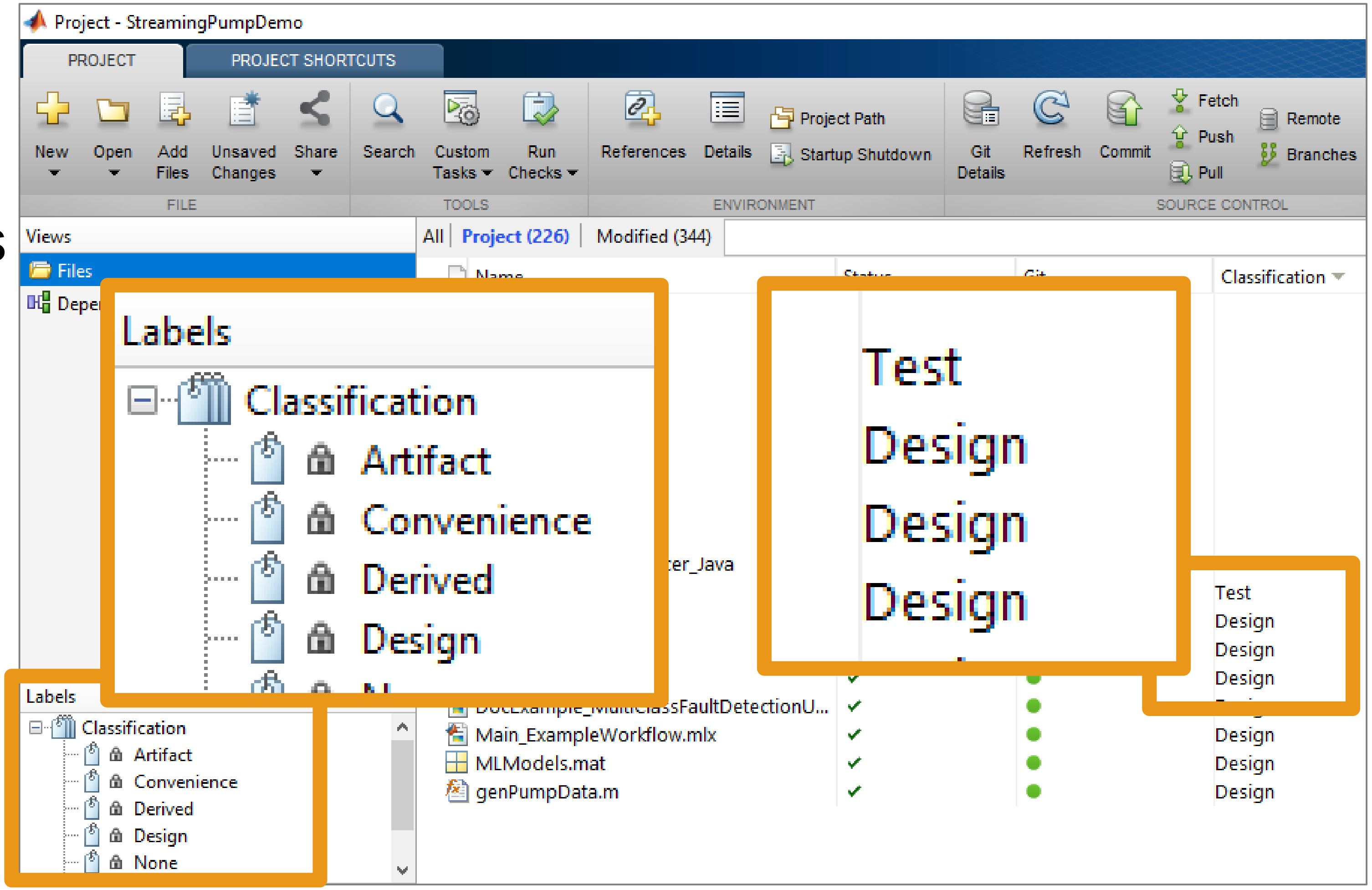
4. Label files

Identify and run tests

…on Continuous Integration (CI) servers

Managing your code with Projects

1. Create project
2. Set path and startup tasks
3. Explore dependencies
4. Label files



Project - StreamingPumpDemo

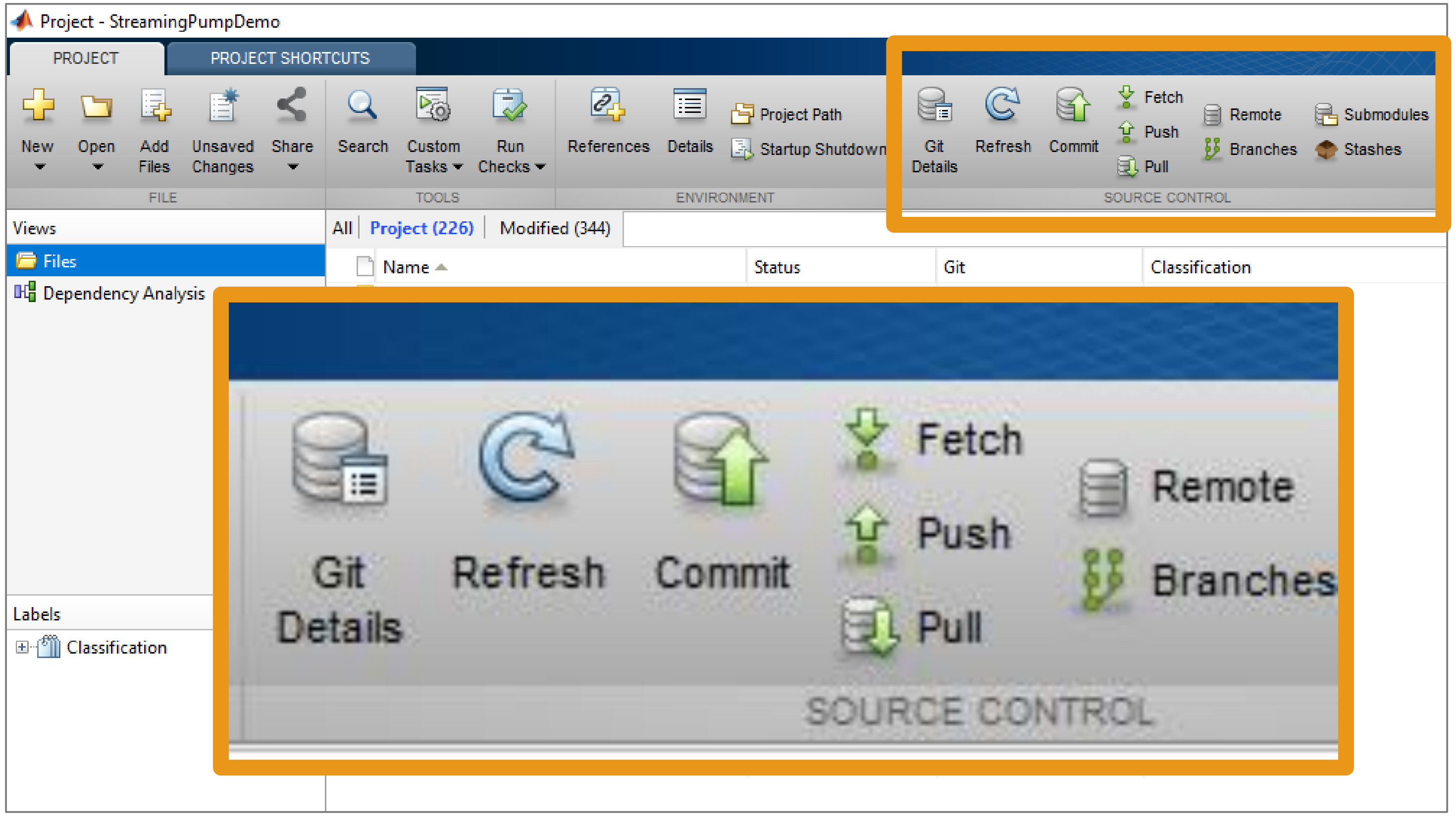
Labels

- Classification
  - Artifact
  - Convenience
  - Derived
  - Design
  - None

Name	Status	Git	Classification
...er_Java			
			Test
			Design
			Design
	✓	●	Design
DocExample_MultiClassFaultDetectionU...	✓	●	
Main_ExampleWorkflow.mlx	✓	●	Design
MLModels.mat	✓	●	Design
genPumpData.m	✓	●	Design

Managing your code with Projects

1. Create project
2. Set path and startup tasks
3. Explore dependencies
4. Label files
5. Integrate source control



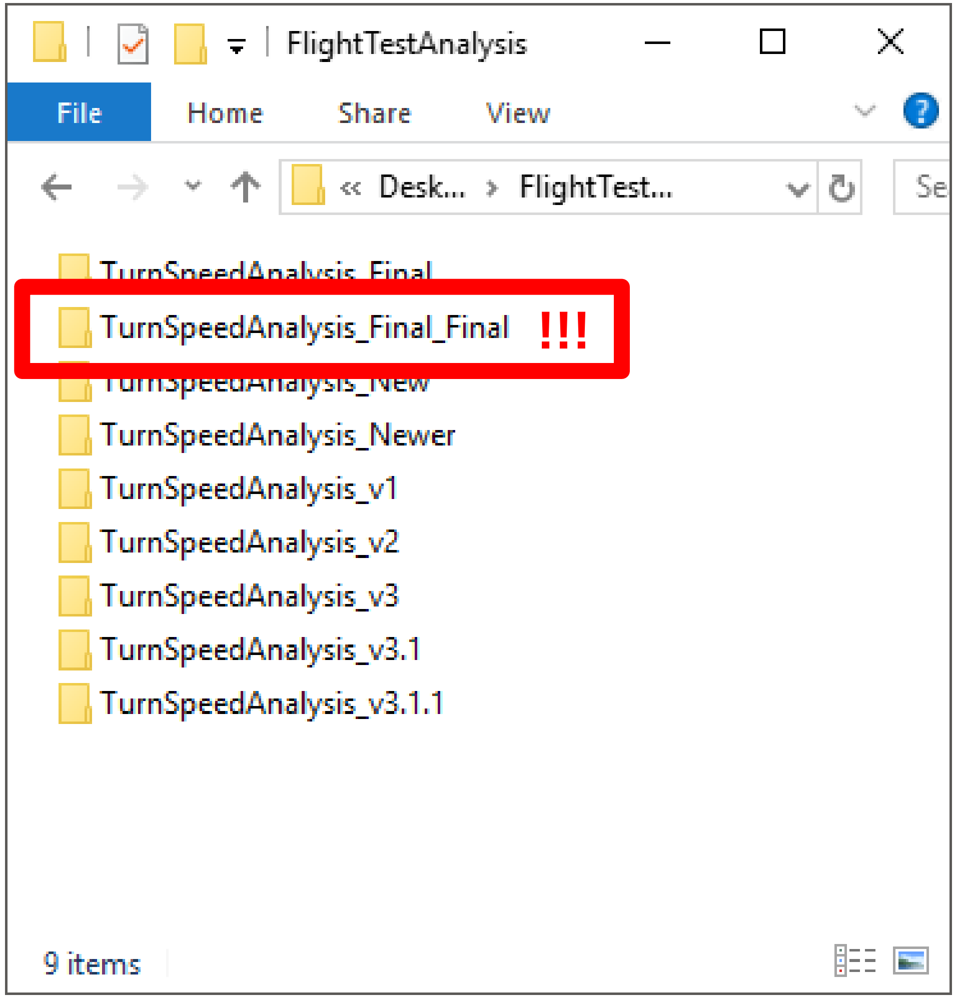
Agenda

	Managing your code
▶	Tracking code changes and co-authoring workflows
	Writing better, robust, and portable code
	Testing and maintaining your code
	Summary

How do you keep track of and share your code as it changes?

- Do you:
  - make copies of your code?
  - e-mail yourself copies of your code?
  - keep a spreadsheet of changes?
- Or do you not keep track of your changes?

There’s a better way!



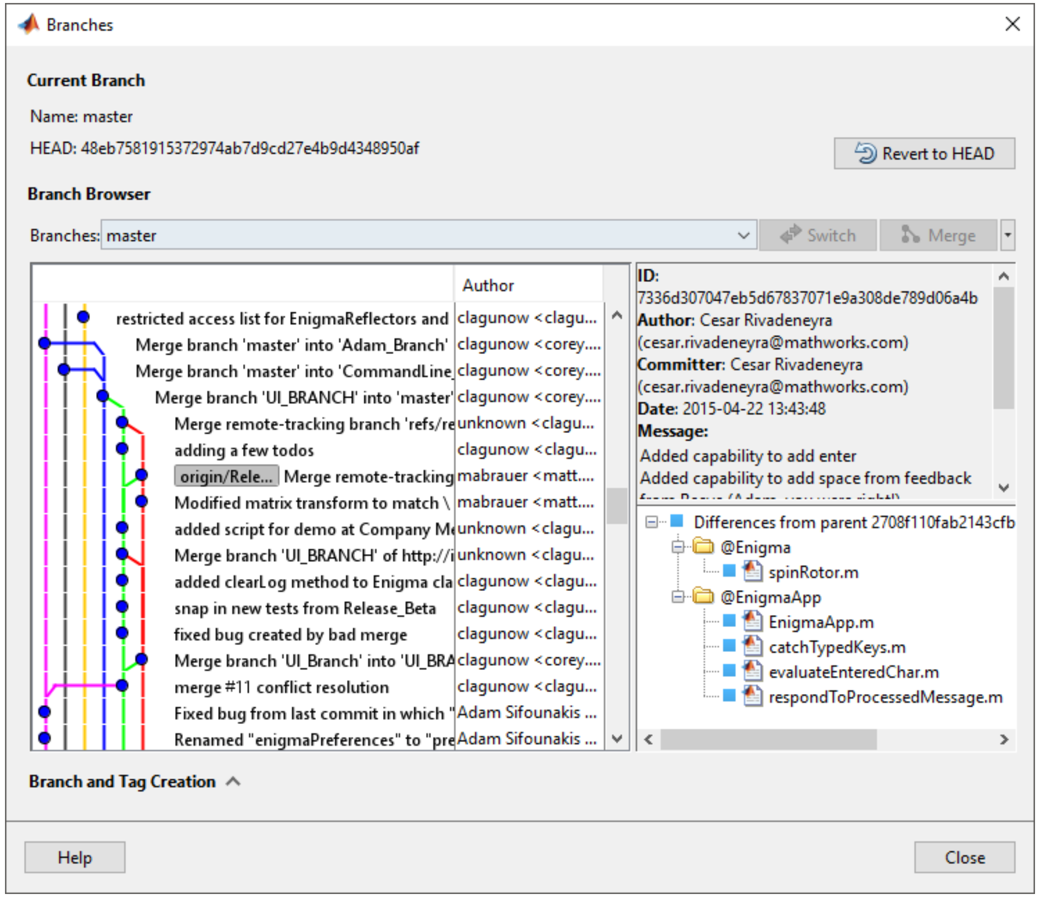
TurnSpeedAnalysis_Final
TurnSpeedAnalysis_Final_Final !!!
TurnSpeedAnalysis_New
TurnSpeedAnalysis_Newer
TurnSpeedAnalysis_v1
TurnSpeedAnalysis_v2
TurnSpeedAnalysis_v3
TurnSpeedAnalysis_v3.1
TurnSpeedAnalysis_v3.1.1

9 items

Source Control

- A system to manage changes to code, documents, etc.

- Benefits of source control:
  - Maintain backups, history, and ability to restore
  - Track changes and responsibility
  - Simplify reconciling conflicting changes
  - Generate discussion
  - Save you from yourself



Current Branch

Name: master

HEAD: 48eb7581915372974ab7d9cd27e4b9d4348950af

Branch Browser

Branches: master

	Author
restricted access list for EnigmaReflectors and	clagunow <clagu...
Merge branch 'master' into 'Adam_Branch'	clagunow <corey....
Merge branch 'master' into 'CommandLine_	clagunow <corey....
Merge branch 'UI_BRANCH' into 'master'	clagunow <corey....
Merge remote-tracking branch 'refs/re	unknown <clagu...
adding a few todos	clagunow <clagu...
origin/Rele... Merge remote-tracking	mabrauer <matt....
Modified matrix transform to match \	mabrauer <matt....
added script for demo at Company Me	unknown <clagu...
Merge branch 'UI_BRANCH' of http://i	unknown <clagu...
added clearLog method to Enigma cla	clagunow <clagu...
snap in new tests from Release_Beta	clagunow <clagu...
fixed bug created by bad merge	clagunow <clagu...
Merge branch 'UI_Branch' into 'UI_BRA	clagunow <corey....
merge #11 conflict resolution	clagunow <clagu...
Fixed bug from last commit in which "	Adam Sifounakis ...
Renamed "enigmaPreferences" to "pre	Adam Sifounakis ...

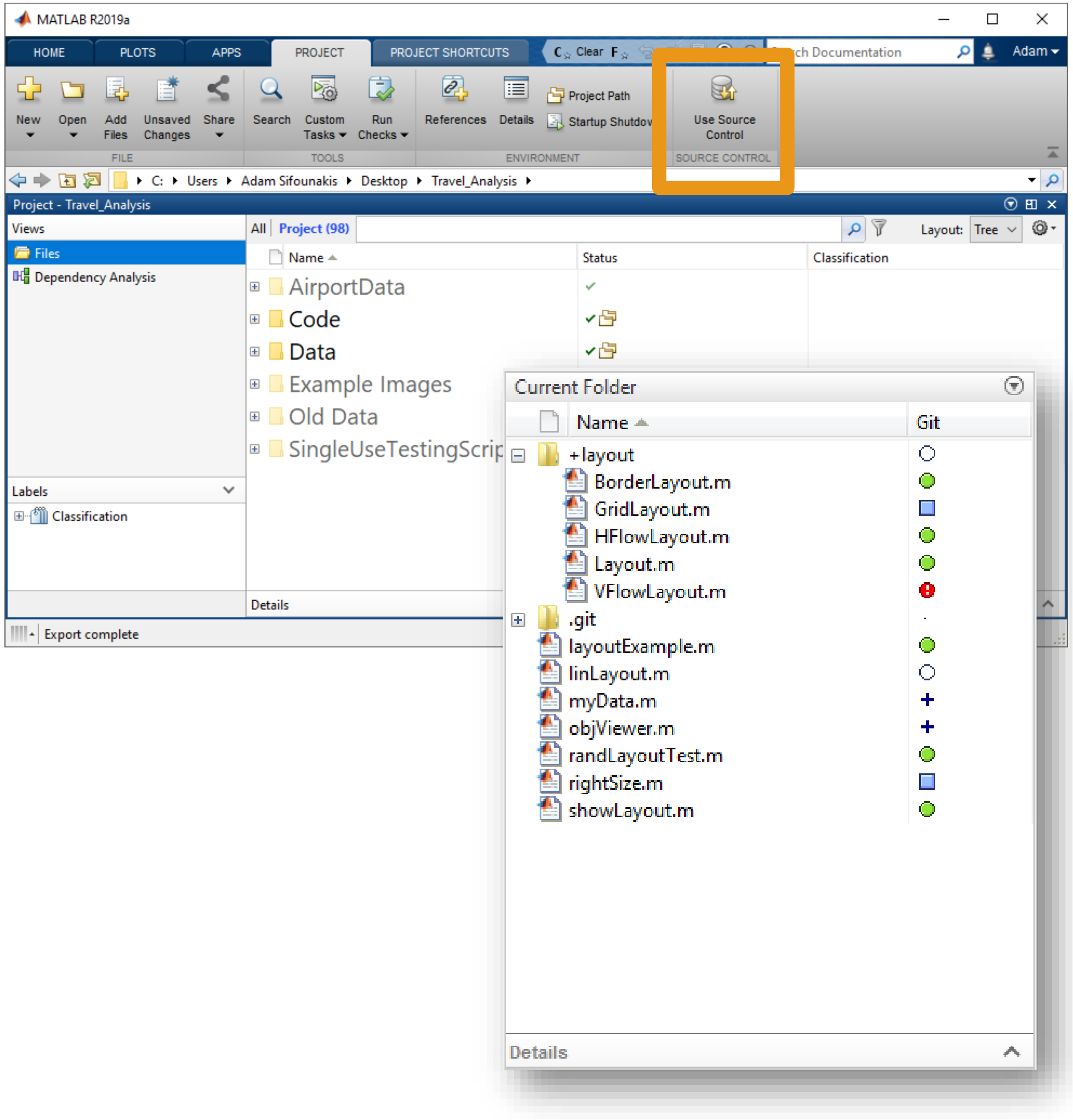
ID:
7336d307047eb5d67837071e9a308de789d06a4b
Author: Cesar Rivadeneyra
(cesar.rivadeneyra@mathworks.com)
Committer: Cesar Rivadeneyra
(cesar.rivadeneyra@mathworks.com)
Date: 2015-04-22 13:43:48
Message:
Added capability to add enter
Added capability to add space from feedback

- Differences from parent 2708f110fab2143cfb
  - @Enigma
    - spinRotor.m
  - @EnigmaApp
    - EnigmaApp.m
    - catchTypedKeys.m
    - evaluateEnteredChar.m
    - respondToProcessedMessage.m

Branch and Tag Creation

Source Control integration

- Manage your code from within the MATLAB Desktop
- Git integrated into:
  - Projects
  - Current Folder browser
- Use Comparison Tool to view and merge changes between revisions



Co-authoring workflows

Creating a repo:

- Initialize
- Add
- Clone

Making changes:

- Commit
- Push
- Branch
- Merge

Repo

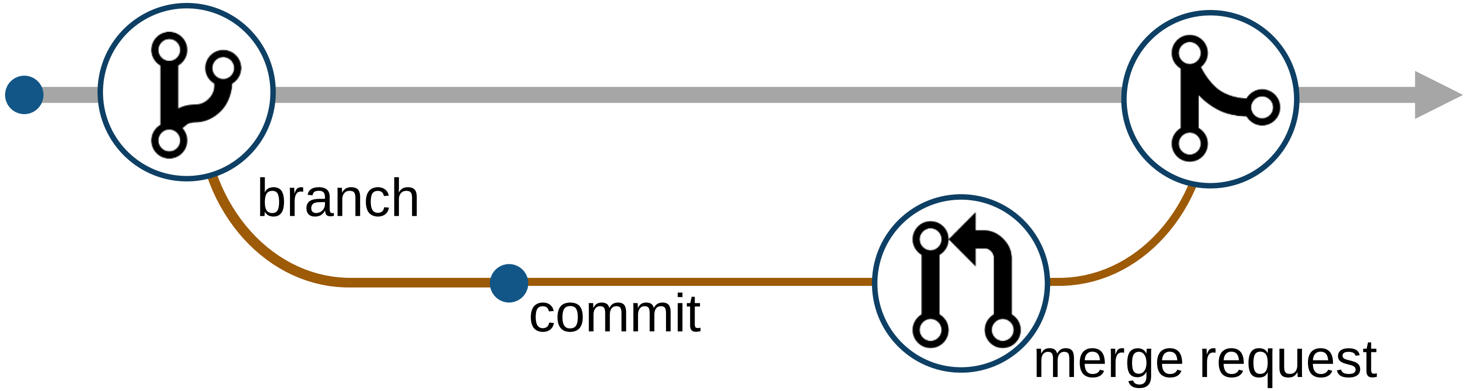
```
% Make prediction
RUL = predictRUL(mdl,data,n);
```

predictUpdateRUL.m ●

predictUpdateRUL.m ●

Repo

```
% Make prediction only when slope changes
if ~isempty(mdl.SlopeDetectionInstant)
    RUL = predictRUL(mdl,data,n);
else
    RUL = old_state.RUL;
end
```



Repo

```
% Make prediction only when slope changes
if ~isempty(mdl.SlopeDetectionInstant)
    RUL = predictRUL(mdl,data,n);
else
    RUL = old_state.RUL;
end
```

predictUpdateRUL.m ●

Agenda

	Managing your code
	Tracking code changes and co-authoring workflows
▶	Writing better, robust, and portable code
	Testing and maintaining your code
	Summary

What defines “better” code?

- Better organized?
- Smaller?
- Faster?
- More stable?
- More portable?
- Easier to maintain?
- …

YES!

Considerations when writing better, robust, and portable code

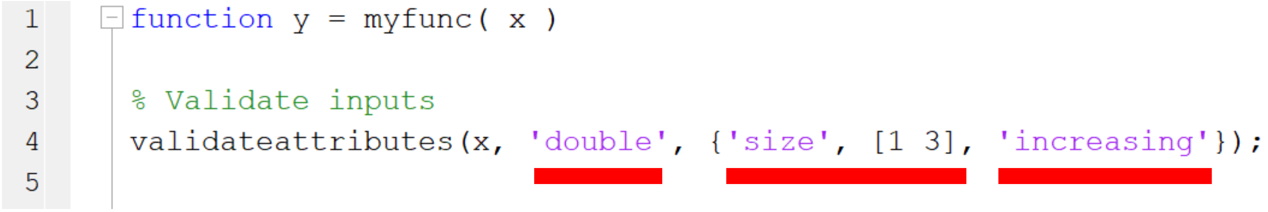
- Input validation
- Error handling
- Writing faster code using the MATLAB Profiler
- Writing code faster using the Live Editor
- Refactoring code to reduce complexity
- Writing code that works on all operating systems
- Sharing your code via apps, toolboxes, and deployment
- Integrating with other languages
- And more…

Writing more robust code

```
>> y = myfunc( 1:5 )
Index exceeds matrix dimensions.

Error in mypkg1.mypkg1a.mypkg1ab.myfunc1 (line 9)
y(idx) = u(idx)*log(u_hat(idx))+(1-u(idx))*log(1-u_hat(idx));

Error in mypkg2.mypkg2a.myfunc2 (line 5)
y = mypkg1.mypkg1a.mypkg1ab.myfunc1( myVar1 .* myVar2 );

Error in mypkg3.mypkg3a.myfunc3>@(x)mypkg2.mypkg2a.myfunc2(x) (line 4)
y = arrayfun( @(x) mypkg2.mypkg2a.myfunc2( x ), myVar );

Error in mypkg3.mypkg3a.myfunc3 (line 4)
y = arrayfun( @(x) mypkg2.mypkg2a.myfunc2( x ), myVar );

Error in myfunc (line 10)
```

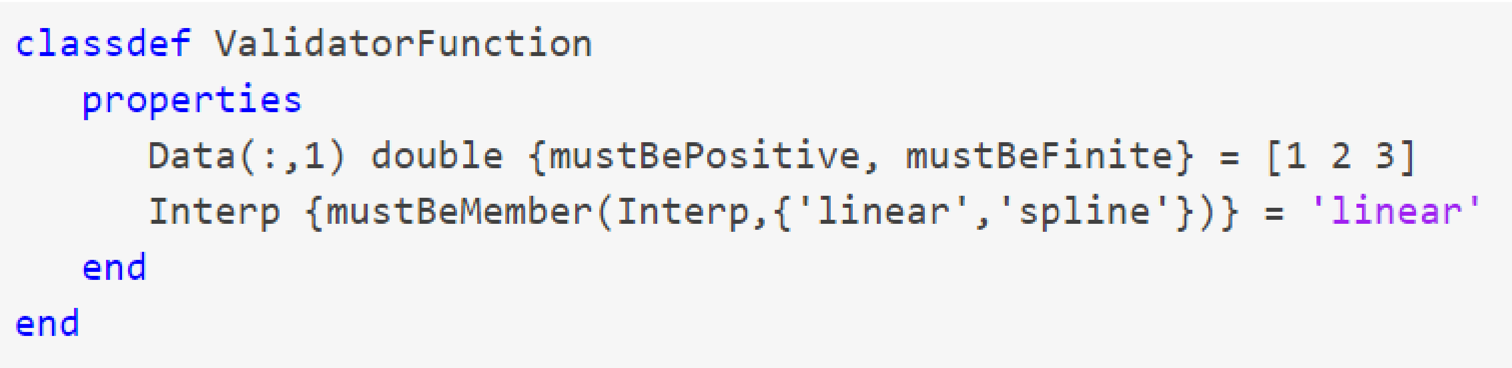
Writing more robust code – Validating inputs

- validateattributes
- assert
- isempty, isnan, isfinite, …
- narginchk
- inputParser
- Property validation for classes



```
1  function y = myfunc( x )
2
3      % Validate inputs
4      validateattributes(x, 'double', {'size', [1 3], 'increasing'});
5
```

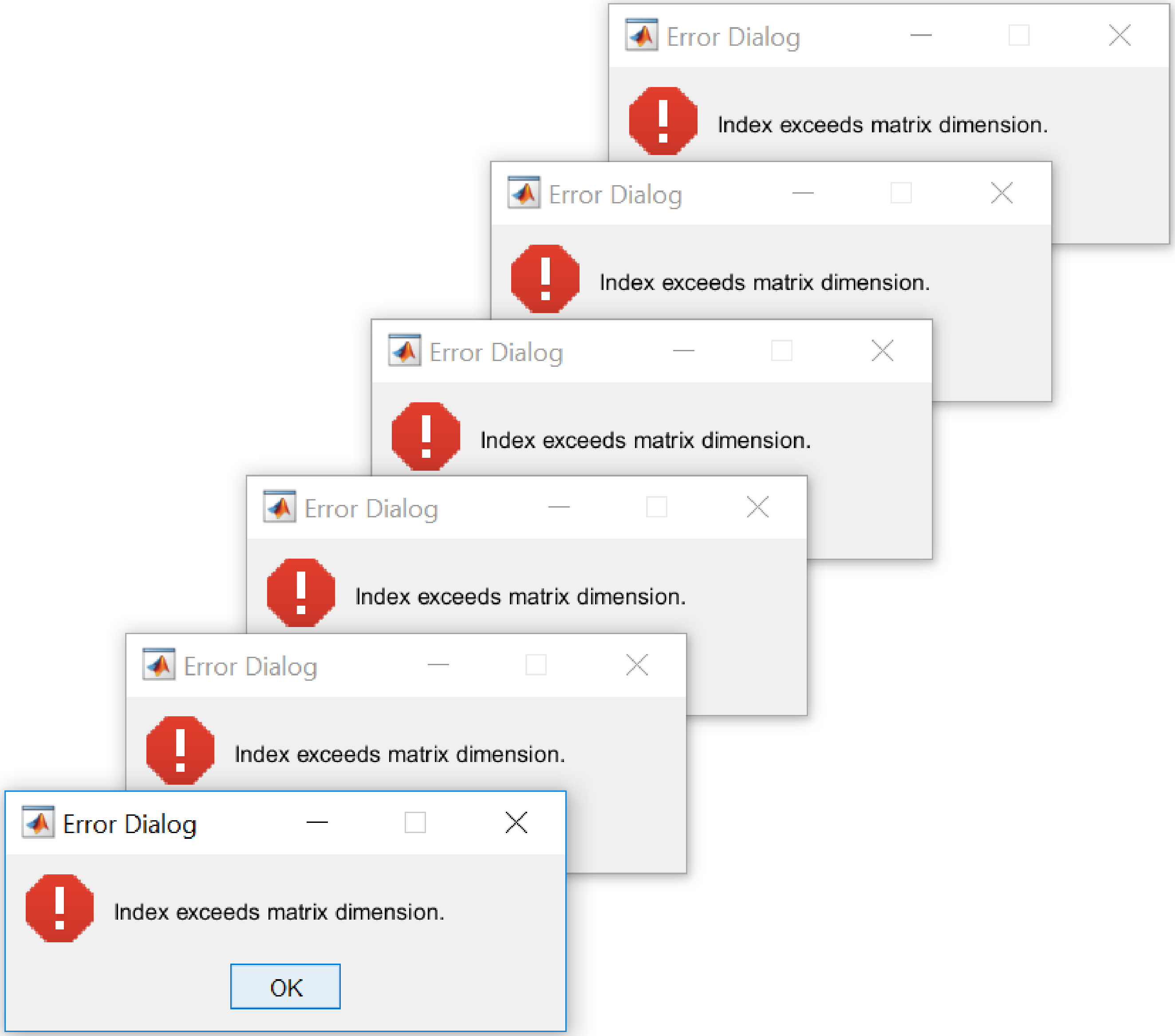
```
>> myfunc( 1:5 )
Error using myfunc (line 4)
Expected input to be of size 1x3, but it is of size 1x5.

>> myfunc( [2 3 1] )
Error using myfunc (line 4)
Expected input to be increasing valued.
```



```
classdef ValidatorFunction
   properties
      Data(:,1) double {mustBePositive, mustBeFinite} = [1 2 3]
      Interp {mustBeMember(Interp,{'linear','spline'})} = 'linear'
   end
end
```

Writing more robust code – Handling errors more elegantly

- `error` and `warning`
  - Use identifiers
- `try/catch`
- `MException`
- `errordlg` and `warndlg`



Index exceeds matrix dimension.

Index exceeds matrix dimension.

Index exceeds matrix dimension.

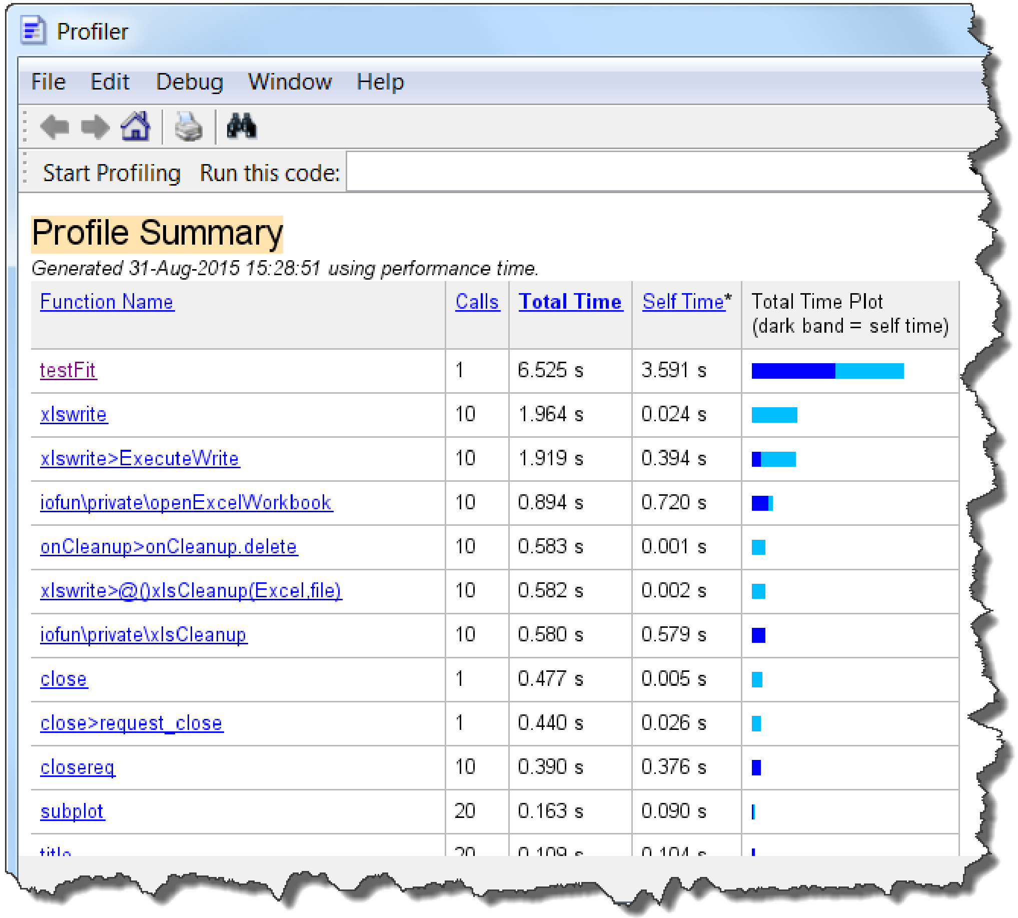
Index exceeds matrix dimension.

Index exceeds matrix dimension.

Index exceeds matrix dimension.

Writing faster code – MATLAB Profiler

- Total number of function calls
- Time per function call
- Highlights largest code bottlenecks
- Statement coverage of code



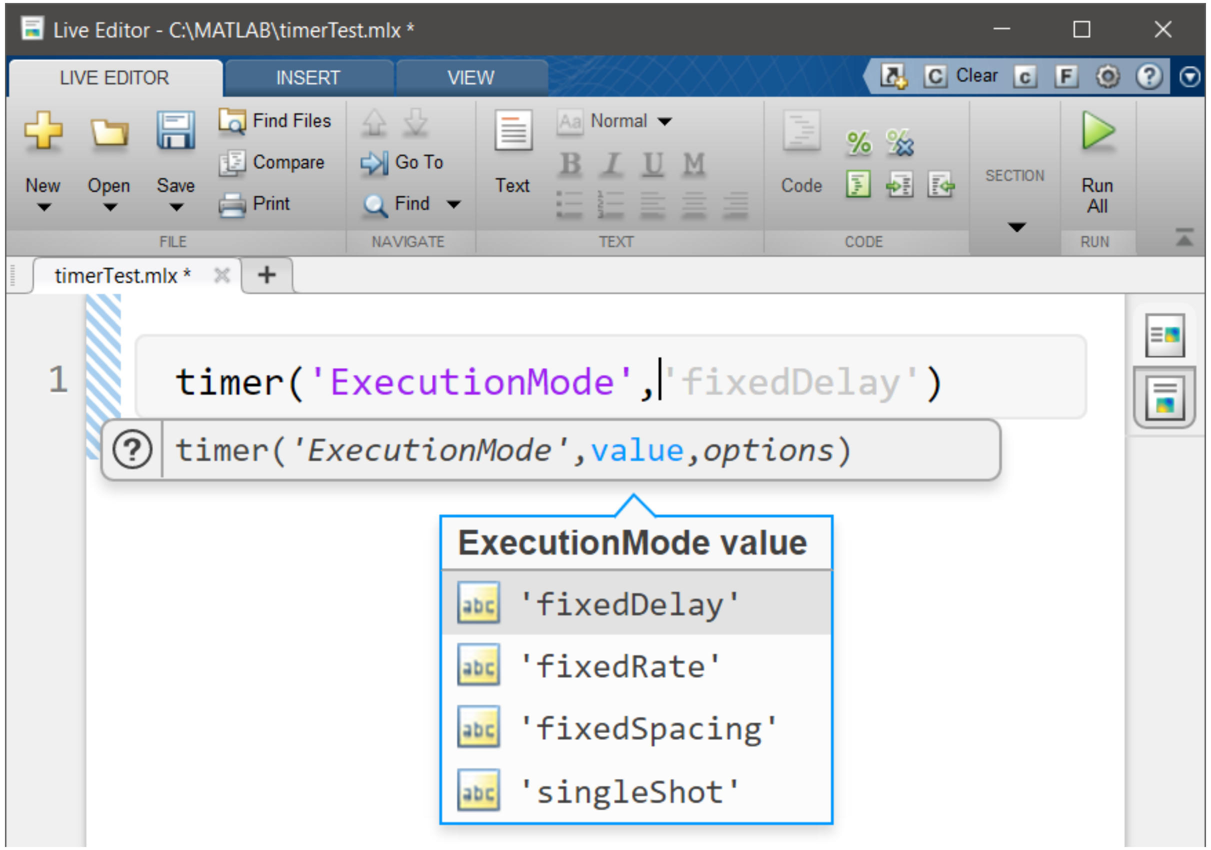
Profile Summary

Generated 31-Aug-2015 15:28:51 using performance time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
testFit	1	6.525 s	3.591 s	
xlswrite	10	1.964 s	0.024 s	
xlswrite>ExecuteWrite	10	1.919 s	0.394 s	
iofun\private\openExcelWorkbook	10	0.894 s	0.720 s	
onCleanup>onCleanup.delete	10	0.583 s	0.001 s	
xlswrite>@()xlsCleanup(Excel,file)	10	0.582 s	0.002 s	
iofun\private\xlsCleanup	10	0.580 s	0.579 s	
close	1	0.477 s	0.005 s	
close>request_close	1	0.440 s	0.026 s	
closereq	10	0.390 s	0.376 s	
subplot	20	0.163 s	0.090 s	
title	20	0.109 s	0.104 s	

Writing code faster – Programming aids in the Live Editor

- Automatically closed parentheses, loops, and conditional blocks
- Context-aware coding guides
  - Automatically suggest function names variables, or file names
  - List available Name/Value pairs



```
timer('ExecutionMode','fixedDelay')
```

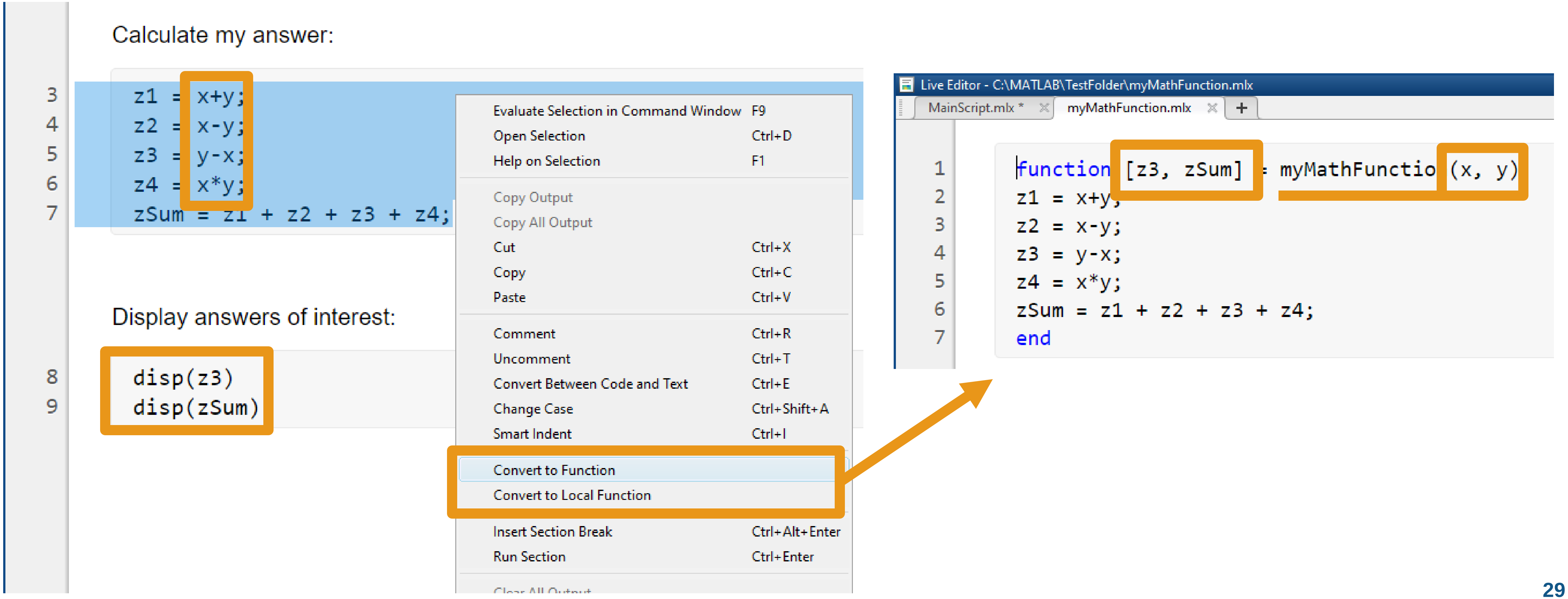
```
timer('ExecutionMode',value,options)
```

ExecutionMode value

- 'fixedDelay'
- 'fixedRate'
- 'fixedSpacing'
- 'singleShot'

Writing code faster – Quickly and safely refactoring code

- Live Editor shortcuts to refactor blocks of code into functions



Calculate my answer:

```
z1 = x+y;
z2 = x-y;
z3 = y-x;
z4 = x*y;
zSum = z1 + z2 + z3 + z4;
```

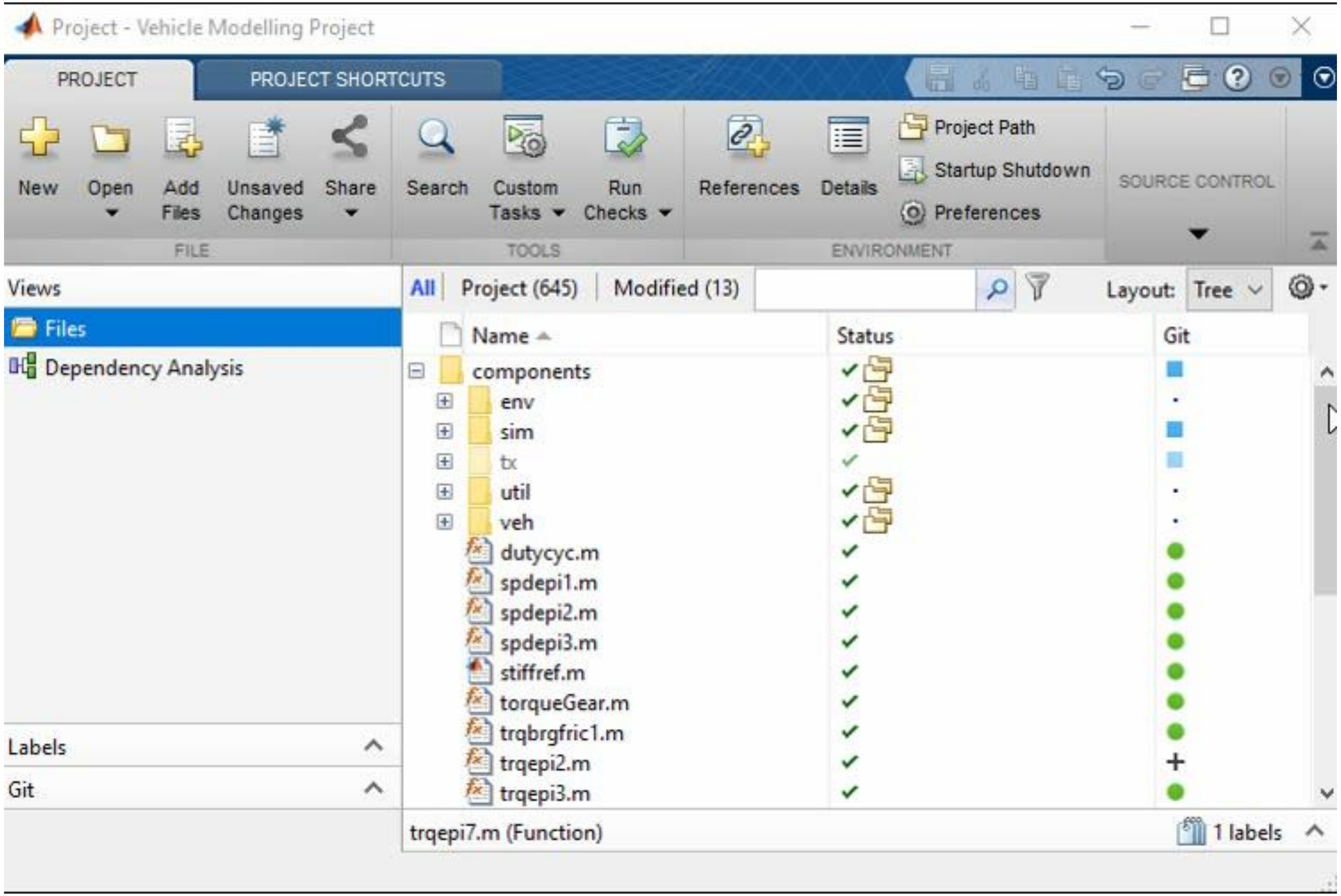
Display answers of interest:

```
disp(z3)
disp(zSum)
```

```
function [z3, zSum] = myMathFunctio (x, y)
z1 = x+y;
z2 = x-y;
z3 = y-x;
z4 = x*y;
zSum = z1 + z2 + z3 + z4;
end
```

Writing code faster – Quickly and safely refactoring code

- Function refactoring across files in Projects



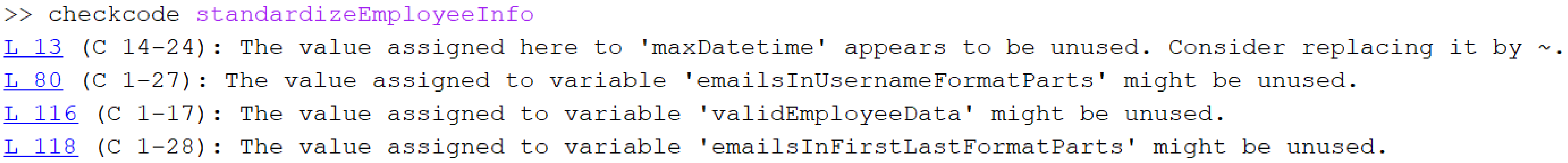
Name	Status	Git
components	✓	■
env	✓	·
sim	✓	■
tx	✓	■
util	✓	·
veh	✓	·
dutycyc.m	✓	●
spdepi1.m	✓	●
spdepi2.m	✓	●
spdepi3.m	✓	●
stiffref.m	✓	●
torqueGear.m	✓	●
trqbrgfric1.m	✓	●
trqepi2.m	✓	+
trqepi3.m	✓	●

trqepi7.m (Function)

1 labels

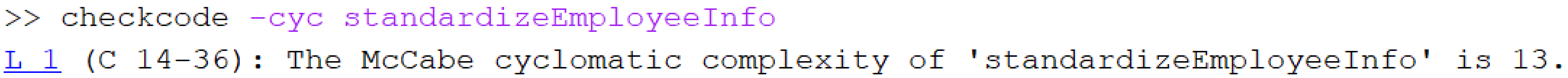
Simple code quality and complexity assessment – `checkcode`

- Analyze all warnings and errors in a code



```
>> checkcode standardizeEmployeeInfo
L 13 (C 14-24): The value assigned here to 'maxDatetime' appears to be unused. Consider replacing it by ~.
L 80 (C 1-27): The value assigned to variable 'emailsInUsernameFormatParts' might be unused.
L 116 (C 1-17): The value assigned to variable 'validEmployeeData' might be unused.
L 118 (C 1-28): The value assigned to variable 'emailsInFirstLastFormatParts' might be unused.
```

- McCabe Cyclomatic Complexity
  - Measures complexity based on the number of linearly independent paths through a code



```
>> checkcode -cyc standardizeEmployeeInfo
L 1 (C 14-36): The McCabe cyclomatic complexity of 'standardizeEmployeeInfo' is 13.
```

Writing more portable code – Code that runs everywhere

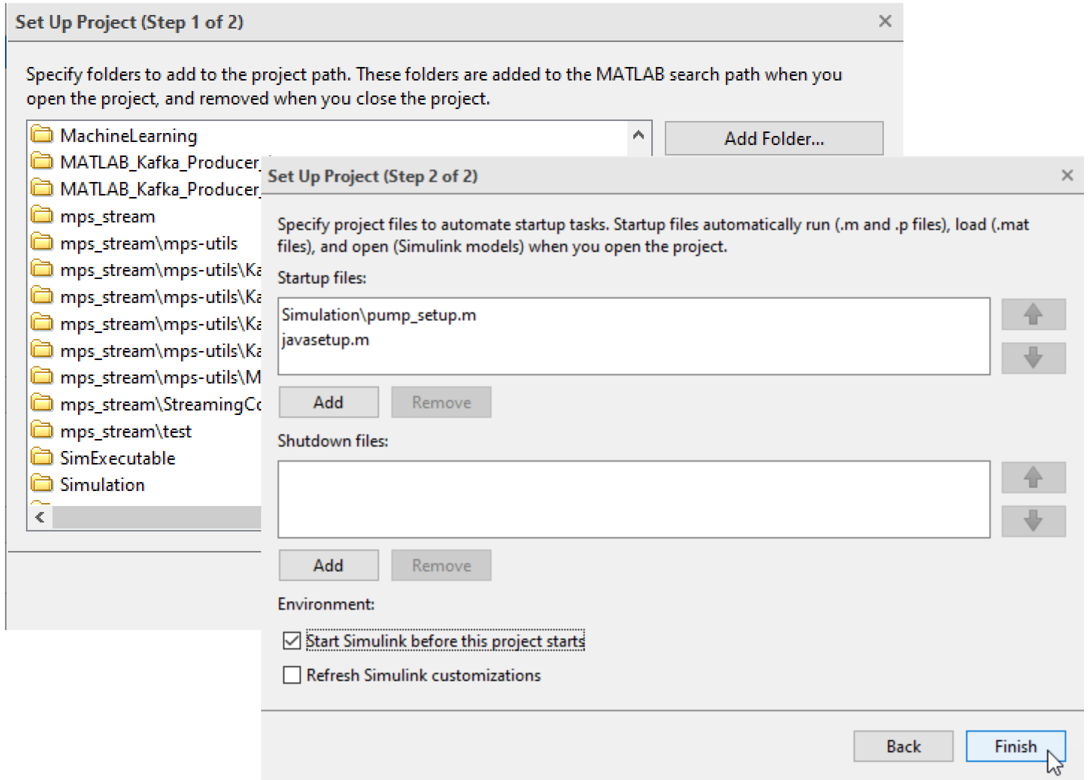
- Operating System-aware code
  - `fullfile`
  - `ispc, ismac, isunix`

```
>> fullfile("..","data","2019","April")
```

Windows: `"..\data\2019\April"`

Mac/Linux: `"../data/2019/April"`

- More reliable portability with Projects
  - Consistent path management
  - Automated startup/shutdown procedures
  - Built-in file dependency analysis



Set Up Project (Step 1 of 2)

Specify folders to add to the project path. These folders are added to the MATLAB search path when you open the project, and removed when you close the project.

MachineLearning
MATLAB_Kafka_Producer
MATLAB_Kafka_Producer
mps_stream
mps_stream\mps-utils
mps_stream\mps-utils\Ka
mps_stream\mps-utils\Ka
mps_stream\mps-utils\Ka
mps_stream\mps-utils\Ka
mps_stream\mps-utils\M
mps_stream\StreamingC
mps_stream\test
SimExecutable
Simulation

Set Up Project (Step 2 of 2)

Specify project files to automate startup tasks. Startup files automatically run (.m and .p files), load (.mat files), and open (Simulink models) when you open the project.

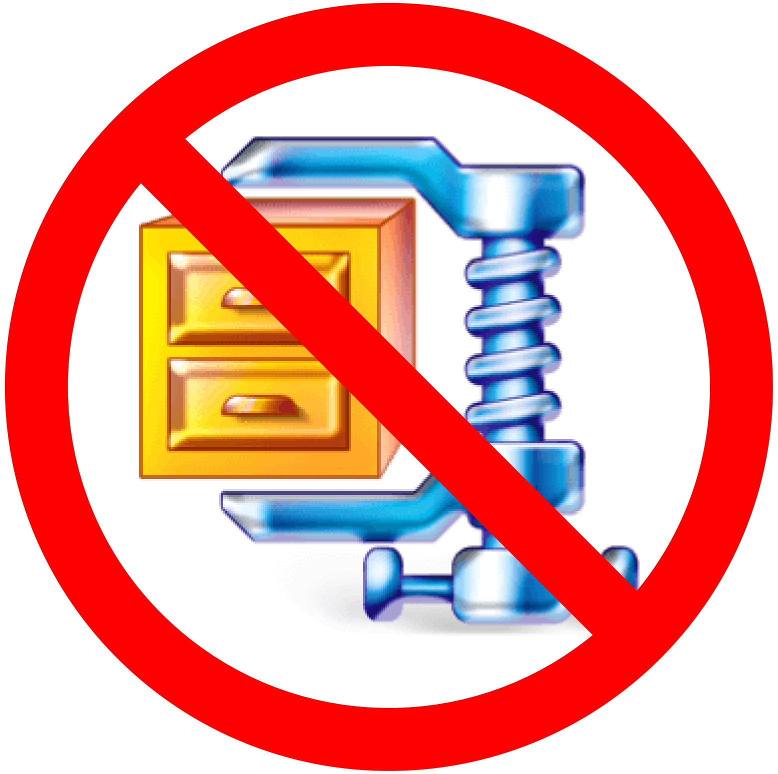
Startup files:
Simulation\pump_setup.m
javasetup.m

Shutdown files:

Environment:
☑ Start Simulink before this project starts
☐ Refresh Simulink customizations

Sharing your code – The traditional way

- Unzip the zip file
- Find the instructions and release notes
- Decide whether you want the thing
- Remove folders from old versions from the path
- Add folders to the path
- Save the path for next time
- Find the documentation
- Do work



Sharing your code – How should you share code?

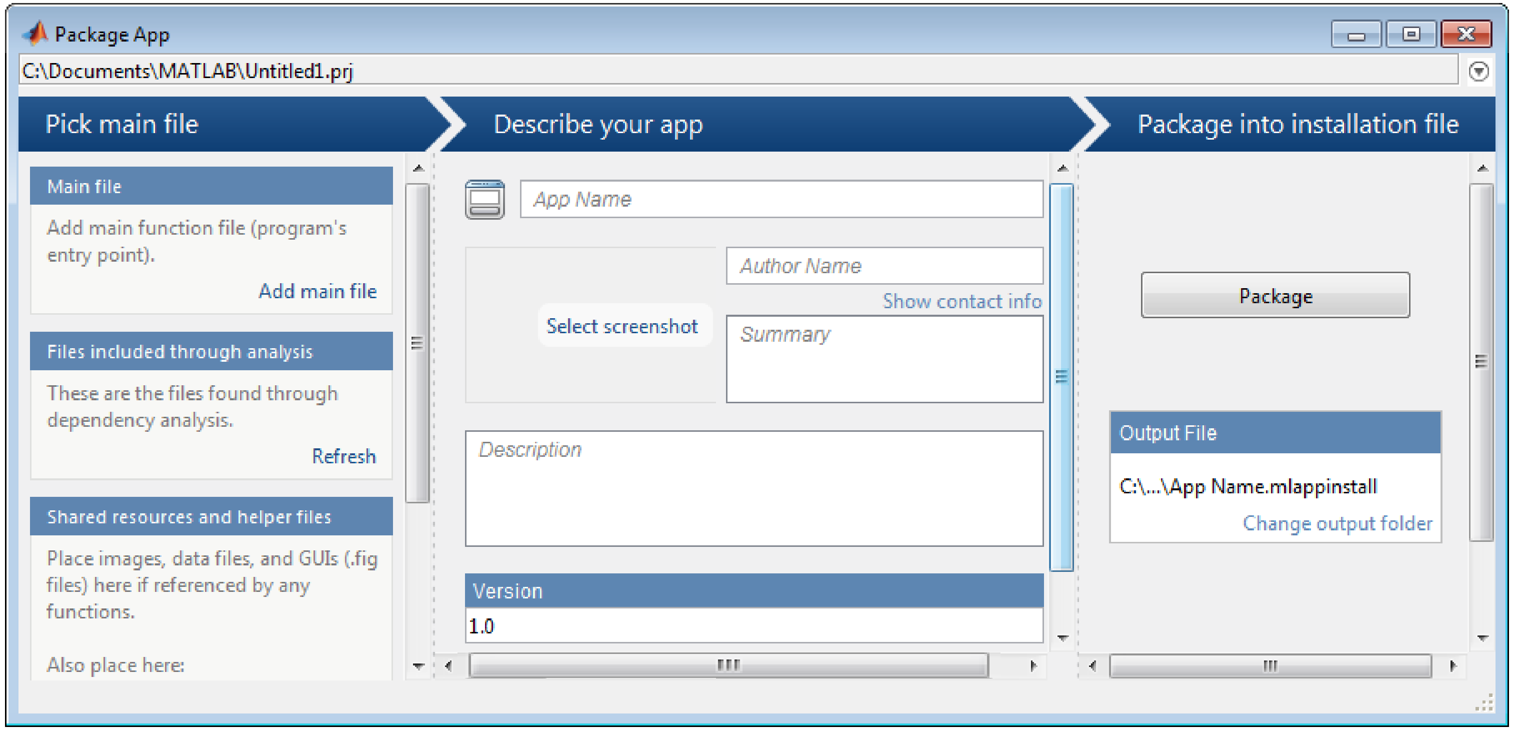
It depends on who you are sharing your code with:

- Co-authors → Project
- End-user with MATLAB → Toolbox or App
- End-user without MATLAB → Deployment (application, library, C code …)

Sharing your code with MATLAB users – Packaging your code

- Toolbox Packaging
- App Packaging

- Combine files into one installation file
- Installs in MATLAB Add-Ons or Apps tab
- Documents required products



Package App

C:\Documents\MATLAB\Untitled1.prj

Main file

Add main function file (program's entry point).

Files included through analysis

These are the files found through dependency analysis.

Shared resources and helper files

Place images, data files, and GUIs (.fig files) here if referenced by any functions.

Also place here:

Version

1.0

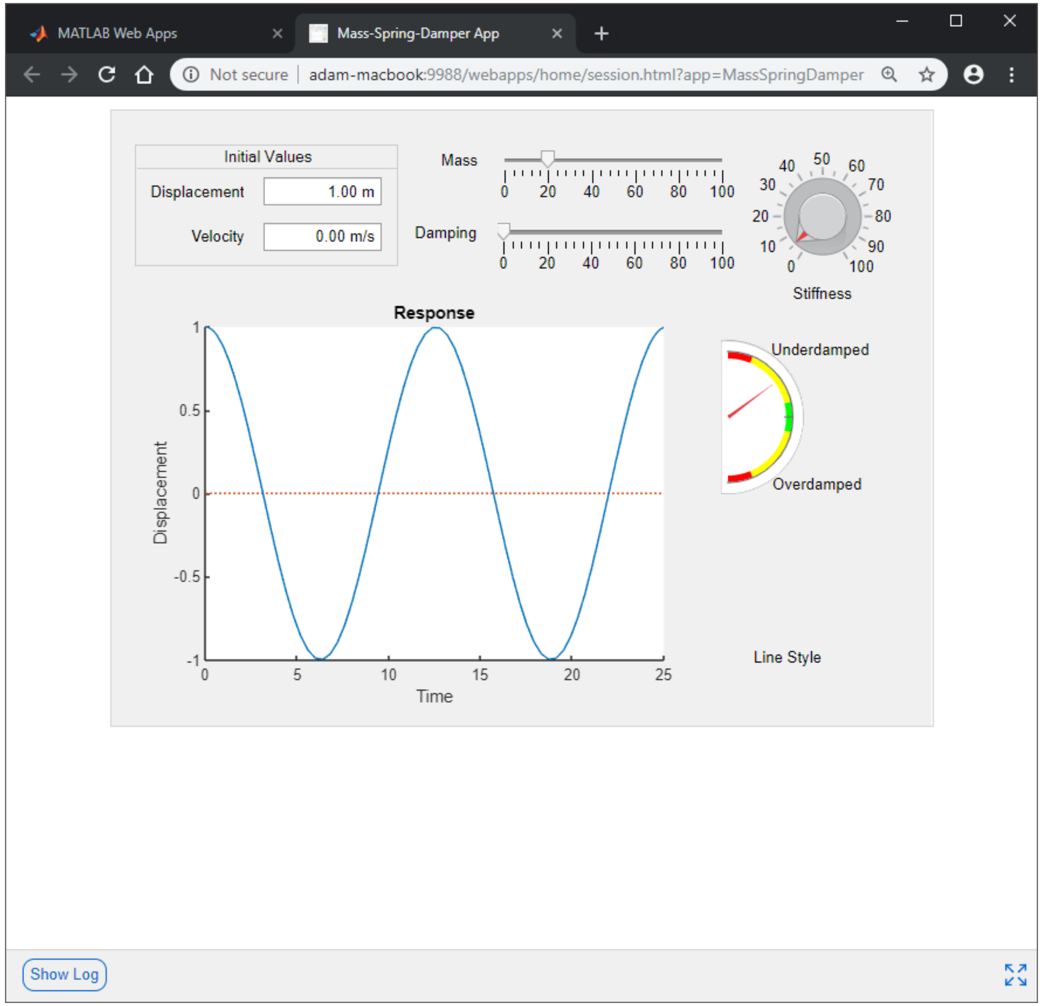
Output File

C:\...\App Name.mlappinstall

Sharing your code outside of MATLAB – Application Deployment

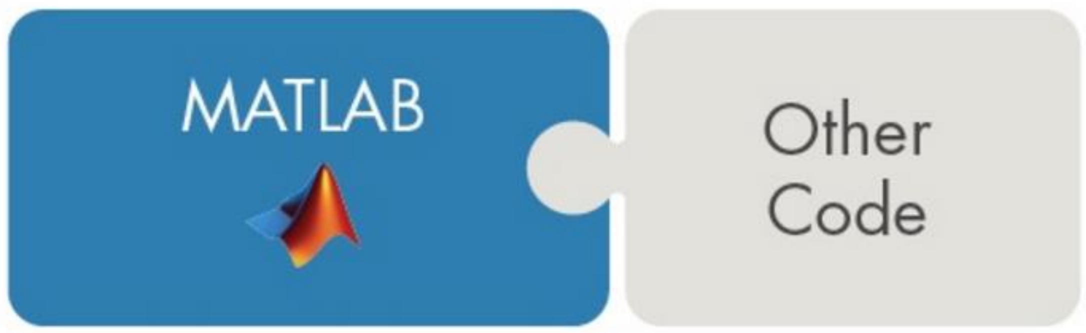
Share your applications as:

- Standalone software — **MATLAB Compiler**
- Web applications — **MATLAB Compiler**
- Language-specific libraries — **MATLAB Compiler SDK**
- Generated code — **MATLAB Coder**



Integrating with other languages – External interfaces

Calling Libraries Written in Another Language



- Java
- Python
- C/C++
- Fortran
- COM components and ActiveX® controls
- RESTful, HTTP, and WSDL web services

Calling MATLAB from Another Language

Your Coding Environment
MATLAB

- Java
- Python
- C/C++
- Fortran
- COM Automation server

Agenda

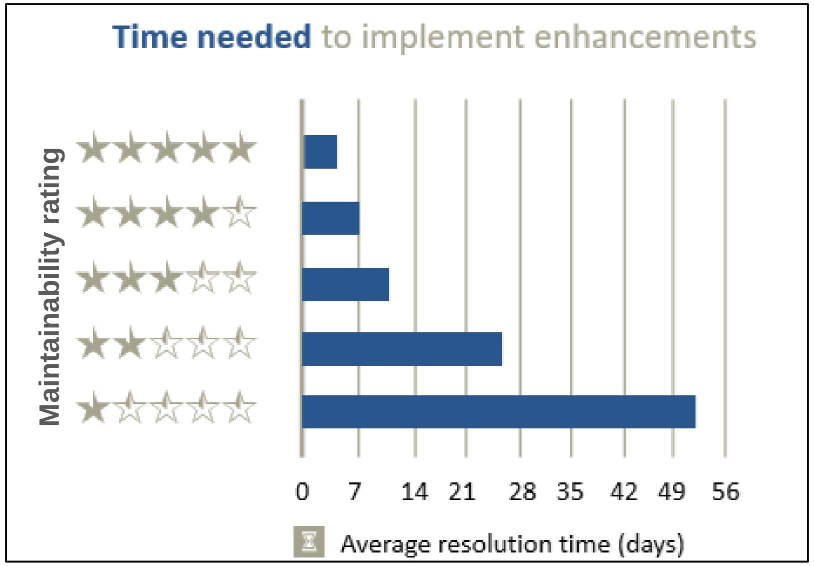
	Managing your code
	Tracking code changes and co-authoring workflows
	Writing better, robust, and portable code
▶	Testing and maintaining your code
	Summary

Code Maintenance – The hidden cost of development

- How do you ensure code doesn't break over time?

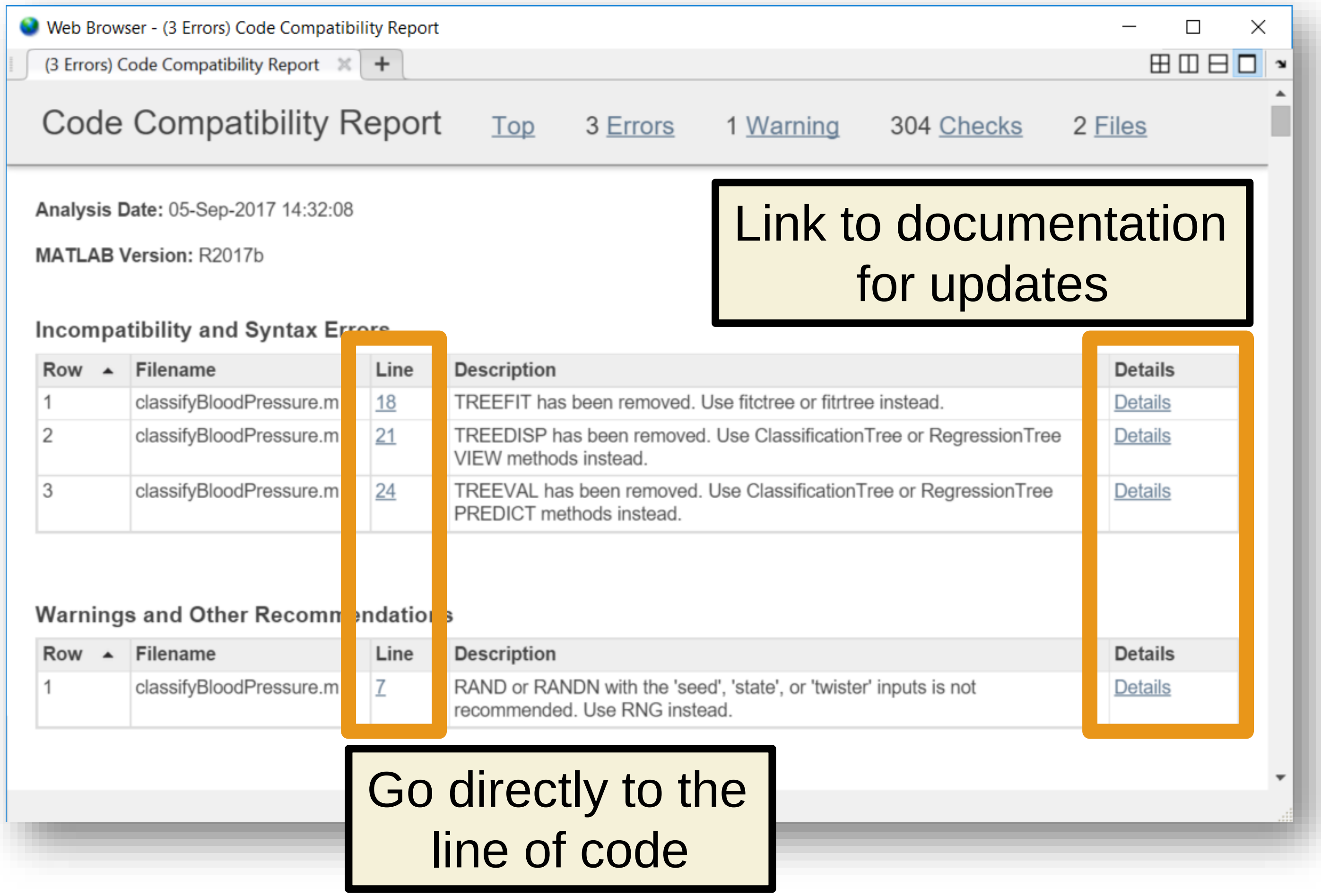
- How do you keep new features from breaking existing features?

- How do you maintain confidence that your code is working as expected?



Upgrading to the latest MATLAB – Code Compatibility Report

- Tool to help upgrade code to latest and greatest MATLAB
- Identifies potential compatibility issues
- Hundreds of checks for incompatibilities, errors, and warnings


Web Browser - (3 Errors) Code Compatibility Report
(3 Errors) Code Compatibility Report
+

Code Compatibility Report   Top   3 Errors   1 Warning   304 Checks   2 Files

Analysis Date: 05-Sep-2017 14:32:08

MATLAB Version: R2017b

Incompatibility and Syntax Errors

Row ▲	Filename	Line	Description	Details
1	classifyBloodPressure.m	18	TREEFIT has been removed. Use fitctree or fitrtree instead.	Details
2	classifyBloodPressure.m	21	TREEDISP has been removed. Use ClassificationTree or RegressionTree VIEW methods instead.	Details
3	classifyBloodPressure.m	24	TREEVAL has been removed. Use ClassificationTree or RegressionTree PREDICT methods instead.	Details

Warnings and Other Recommendations

Row ▲	Filename	Line	Description	Details
1	classifyBloodPressure.m	7	RAND or RANDN with the 'seed', 'state', or 'twister' inputs is not recommended. Use RNG instead.	Details

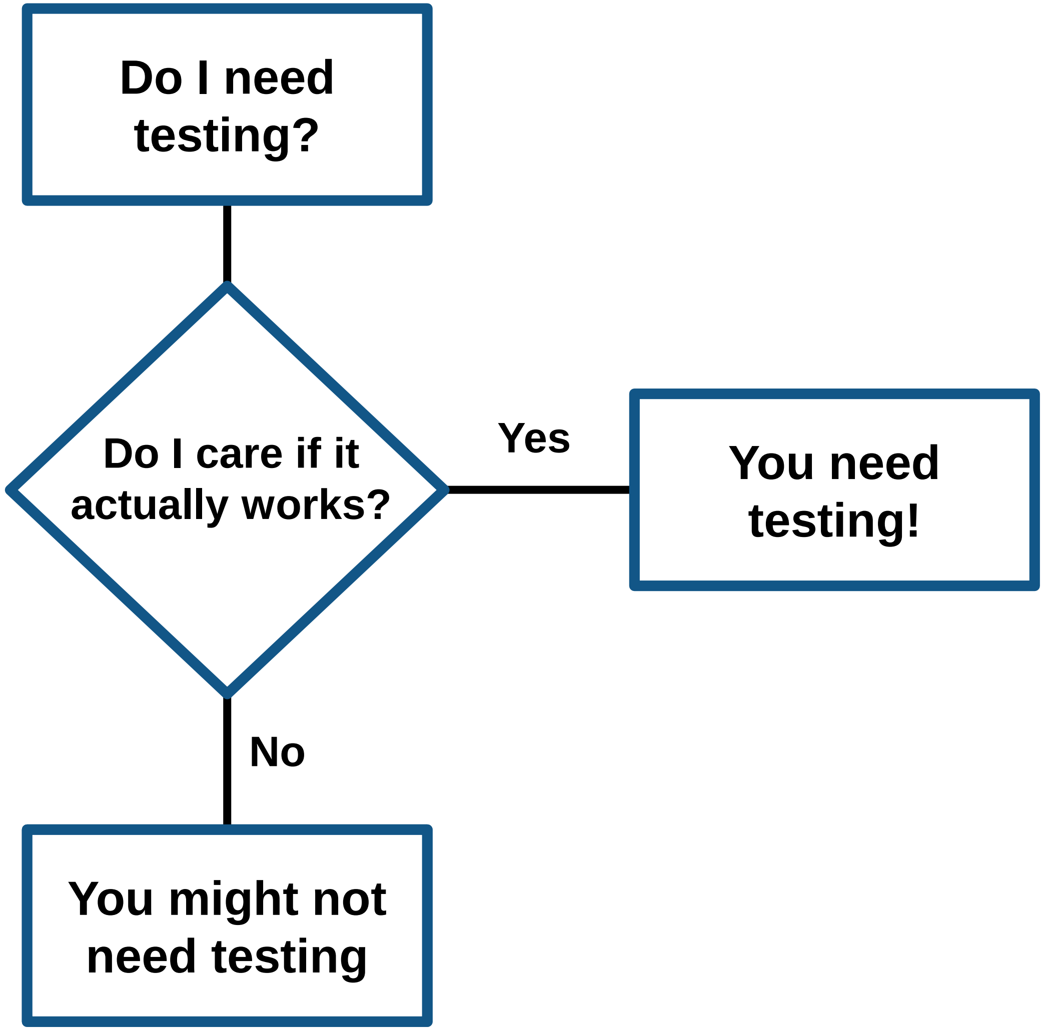
Link to documentation for updates

Go directly to the line of code

Test early, test often, test automatically

- Reduce risk of code breaking
- Catch problems early
- Improve code quality
- Document expected behaviour

Credit: http://geek-and-poke.com/


Do I need testing?

Do I care if it actually works?

Yes

You need testing!

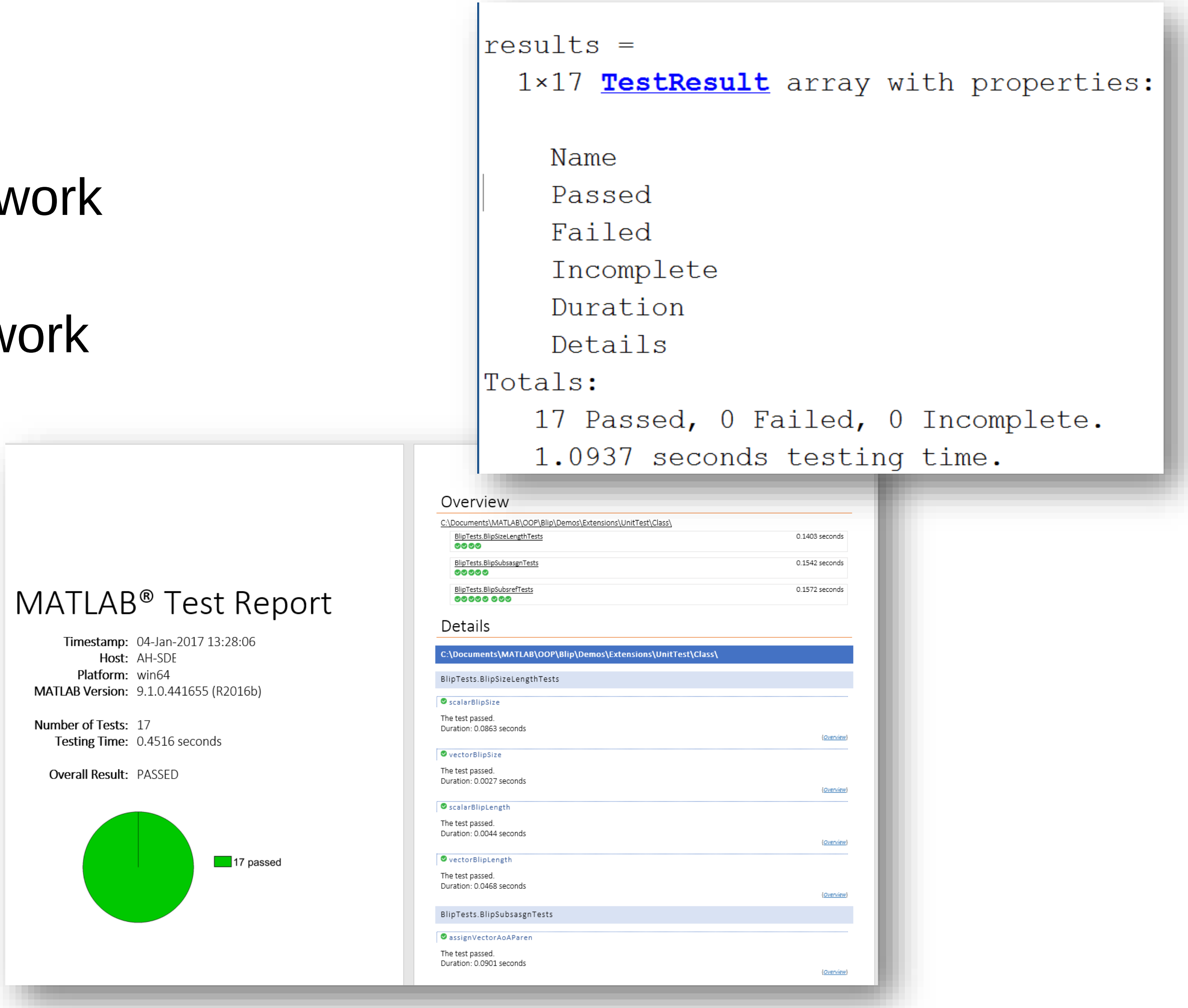
No

You might not need testing

Testing Frameworks

Test your code early and often

- MATLAB Unit Testing Framework
- Performance Testing Framework
- App Testing Framework



```
results =
  1×17 TestResult array with properties:

    Name
    Passed
    Failed
    Incomplete
    Duration
    Details
Totals:
   17 Passed, 0 Failed, 0 Incomplete.
   1.0937 seconds testing time.
```

MATLAB® Test Report

Timestamp: 04-Jan-2017 13:28:06
Host: AH-SDE
Platform: win64
MATLAB Version: 9.1.0.441655 (R2016b)

Number of Tests: 17
Testing Time: 0.4516 seconds

Overall Result: PASSED

🟩 17 passed

Overview

C:\Documents\MATLAB\OOP\Blip\Demos\Extensions\UnitTest\Class\

BlipTests.BlipSizeLengthTests ✓✓✓✓	0.1403 seconds
BlipTests.BlipSubsasgnTests ✓✓✓✓✓	0.1542 seconds
BlipTests.BlipSubsrefTests ✓✓✓✓✓ ✓✓✓	0.1572 seconds

Details

C:\Documents\MATLAB\OOP\Blip\Demos\Extensions\UnitTest\Class\

BlipTests.BlipSizeLengthTests

✓ scalarBlipSize

The test passed.
Duration: 0.0863 seconds

(Overview)

✓ vectorBlipSize

The test passed.
Duration: 0.0027 seconds

(Overview)

✓ scalarBlipLength

The test passed.
Duration: 0.0044 seconds

(Overview)

✓ vectorBlipLength

The test passed.
Duration: 0.0468 seconds

(Overview)

BlipTests.BlipSubsasgnTests

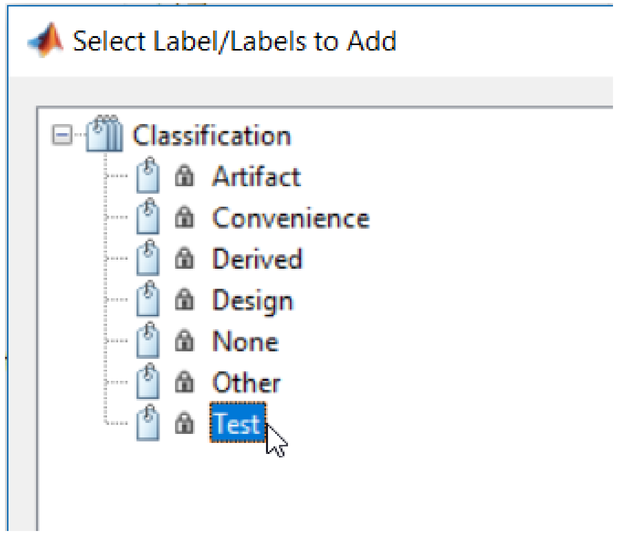
✓ assignVectorAoAParen

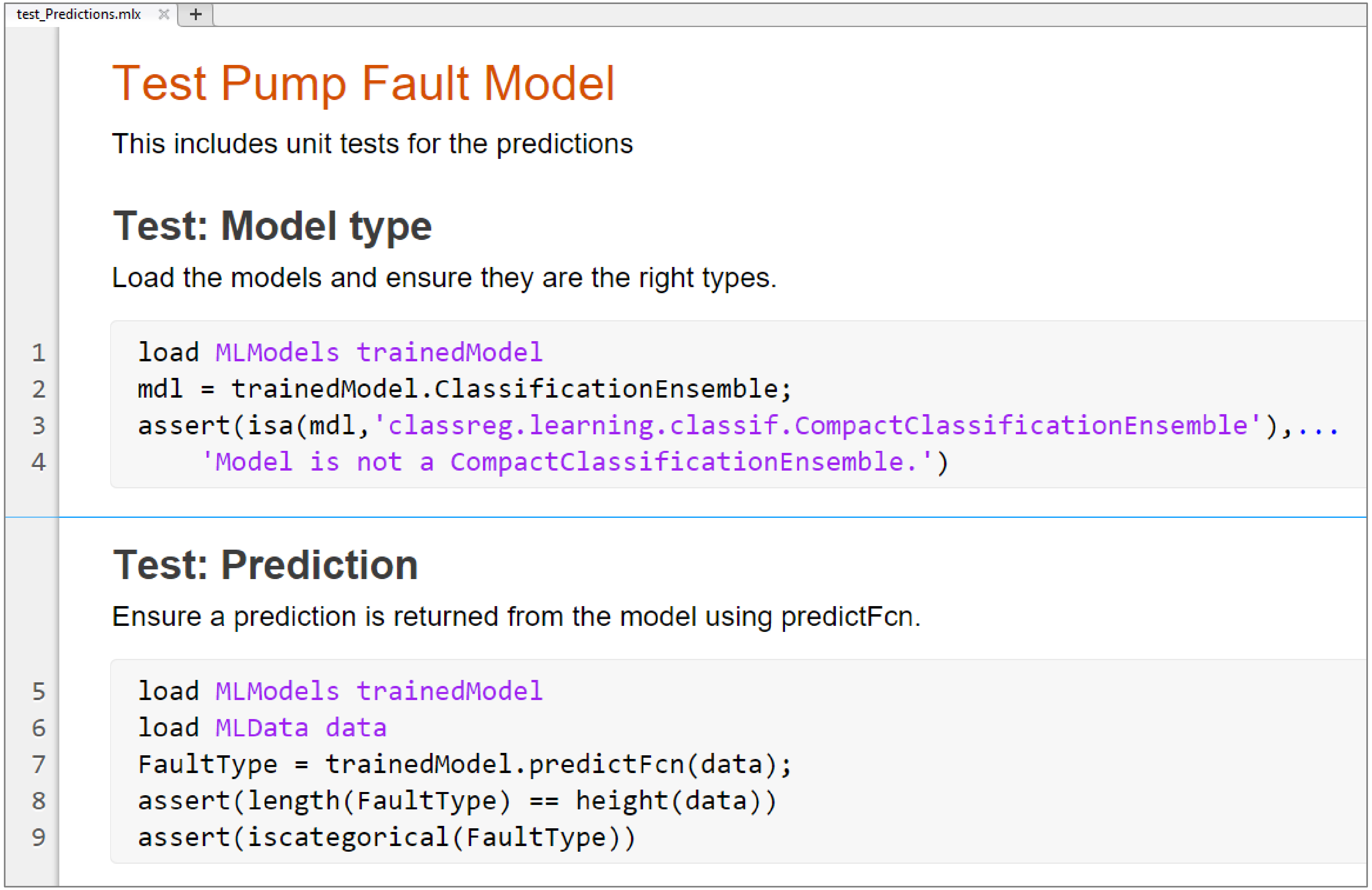
The test passed.
Duration: 0.0901 seconds

(Overview)

Testing Frameworks – Flexible development

- Script-based test
- Function-based test
- Class-based test
- Test integration with Projects





Test Pump Fault Model

This includes unit tests for the predictions

Test: Model type

Load the models and ensure they are the right types.

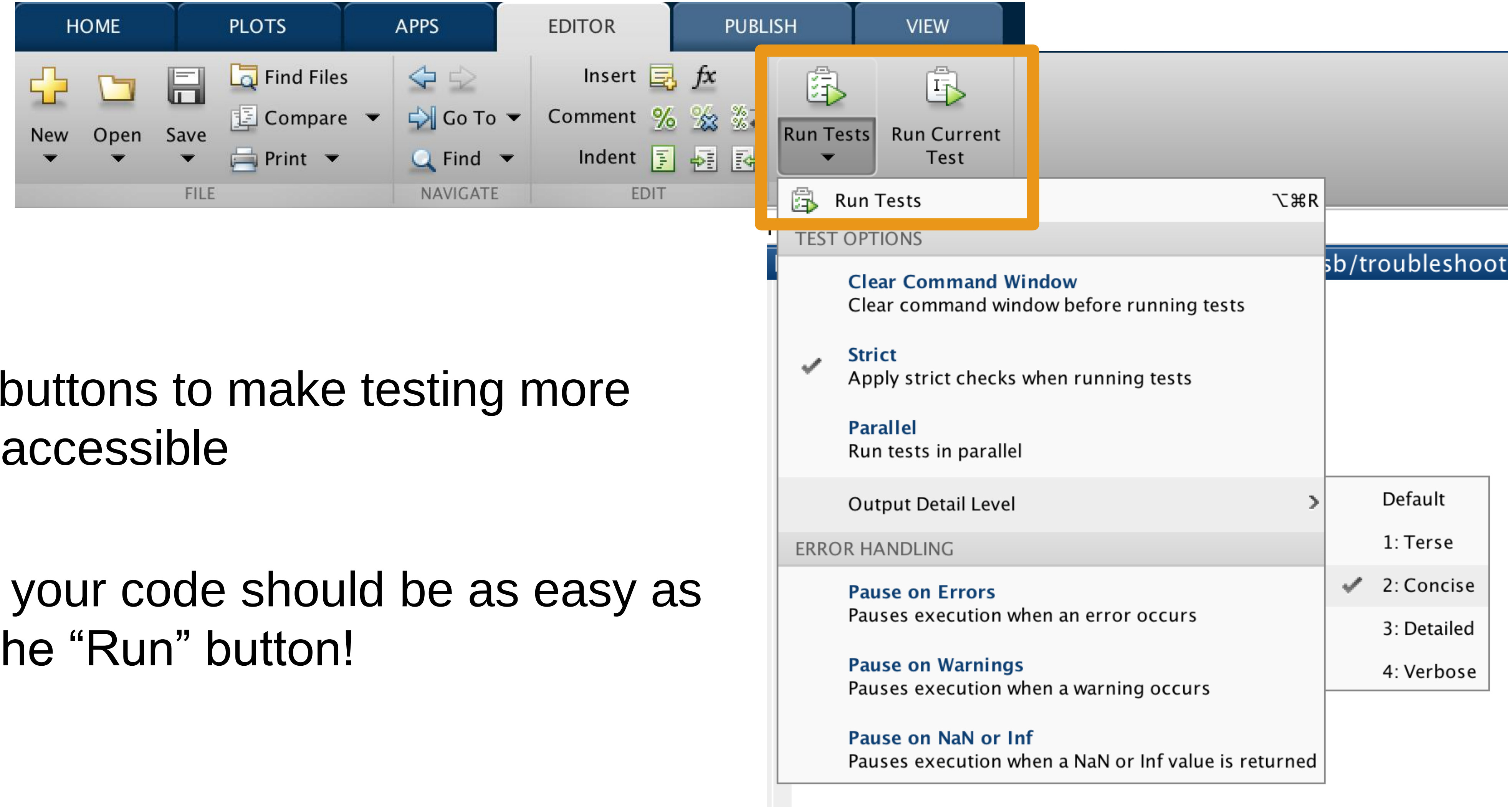
```
load MLModels trainedModel
mdl = trainedModel.ClassificationEnsemble;
assert(isa(mdl,'classreg.learning.classif.CompactClassificationEnsemble'),...
    'Model is not a CompactClassificationEnsemble.')
```

Test: Prediction

Ensure a prediction is returned from the model using predictFcn.

```
load MLModels trainedModel
load MLData data
FaultType = trainedModel.predictFcn(data);
assert(length(FaultType) == height(data))
assert(iscategorical(FaultType))
```

Testing Frameworks – Easily customize and run existing tests



Run Tests ⌥⌘R

TEST OPTIONS

Clear Command Window
Clear command window before running tests

✓ **Strict**
Apply strict checks when running tests

Parallel
Run tests in parallel

Output Detail Level ›

- Default
- 1: Terse
- ✓ 2: Concise
- 3: Detailed
- 4: Verbose

ERROR HANDLING

Pause on Errors
Pauses execution when an error occurs

Pause on Warnings
Pauses execution when a warning occurs

Pause on NaN or Inf
Pauses execution when a NaN or Inf value is returned

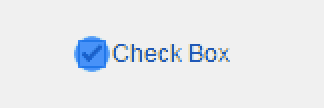
- Added buttons to make testing more readily accessible

- Testing your code should be as easy as hitting the “Run” button!

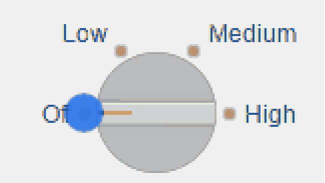
Testing Frameworks – App Testing Framework

- Verify app behavior with tests that programmatically perform gestures on a UI component

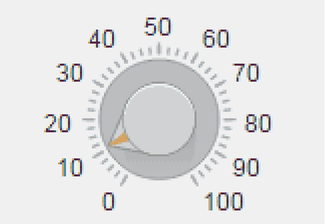
```
testCase.press(myApp.checkbox)
```



```
testCase.choose(myApp.discreteKnob, "Medium")
```



```
testCase.drag(myApp.continuousKnob, 10, 90)
```



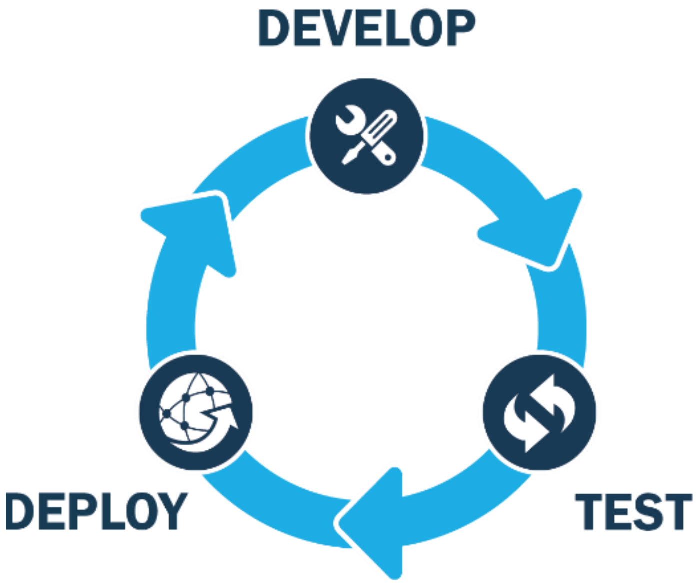
```
testCase.type(myApp.editfield, myTextVar)
```

Automated Testing – Continuous Integration (CI)

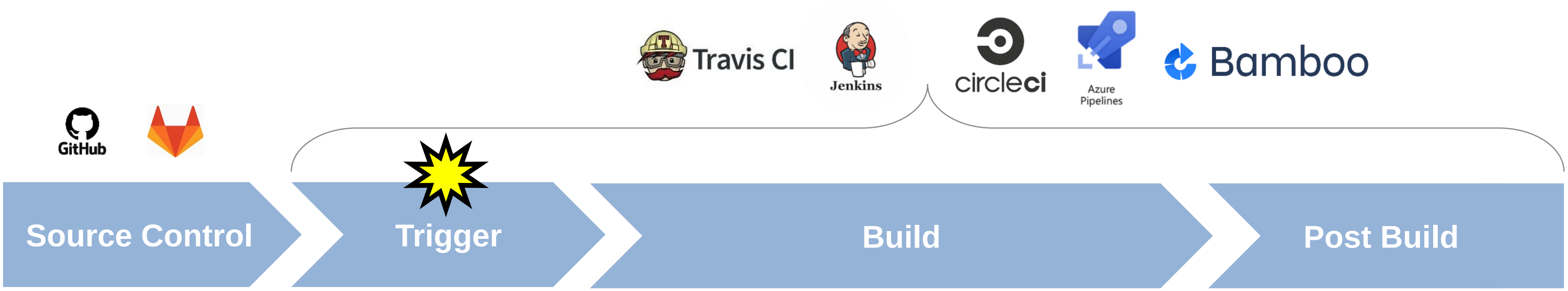
- A system to automate the building, testing, integration, and deployment of code as it is being developed and maintained

- Popular CI systems: Jenkins, Travis, CircleCI , Bamboo, and others…

- Benefits:
  - Detect integration bugs early
  - Allow you to stop bugs from being accepted
  - Track and report testing history
  - Flexible testing schedules and triggers



Automated Testing – Continuous Integration workflow



- Git
- GitHub
- GitLab
- Subversion
- …

- Push
- Merge Request
- Pull Request
- Check In
- Scheduled
- Manual

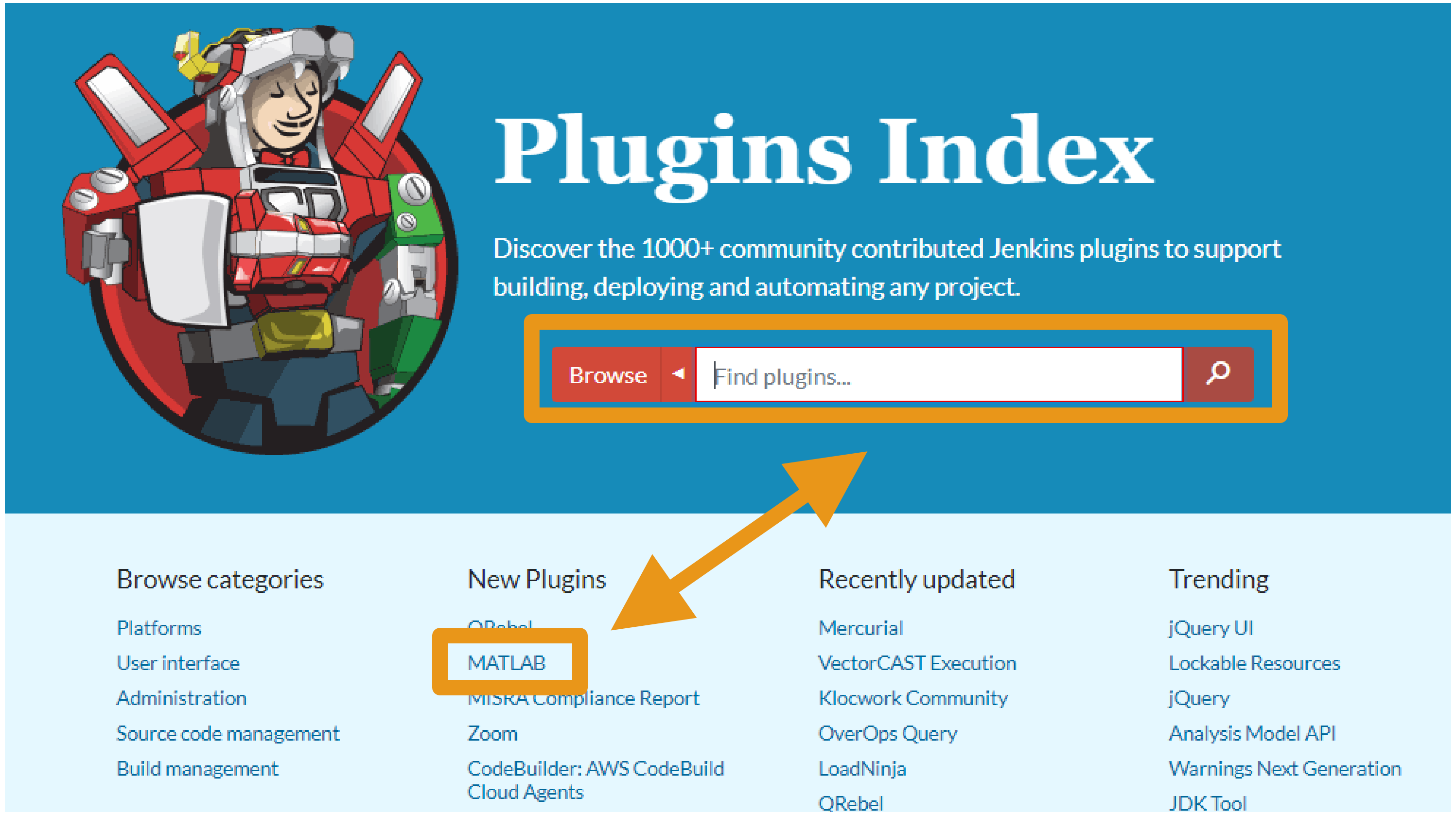
- Run MATLAB / Simulink Tests
- Run Performance Tests
- Compile MEX Files
- Generate Code*
- Package Toolboxes
- Build Components with MATLAB Compiler Stack*
- Integrate with other software

- Publish:
  - Test Results
  - Coverage Results
  - Performance Results
- Accept Merge Request
- Email Notification

* Transformation products may require Client Access Licensing

Automated Testing – Jenkins plugin

- Easily connect and configure MATLAB with Jenkins
- Schedule automatic code execution and testing:
  - based on time of day
  - whenever new code changes are committed



Plugins Index

Discover the 1000+ community contributed Jenkins plugins to support building, deploying and automating any project.

Browse categories
- Platforms
- User interface
- Administration
- Source code management
- Build management

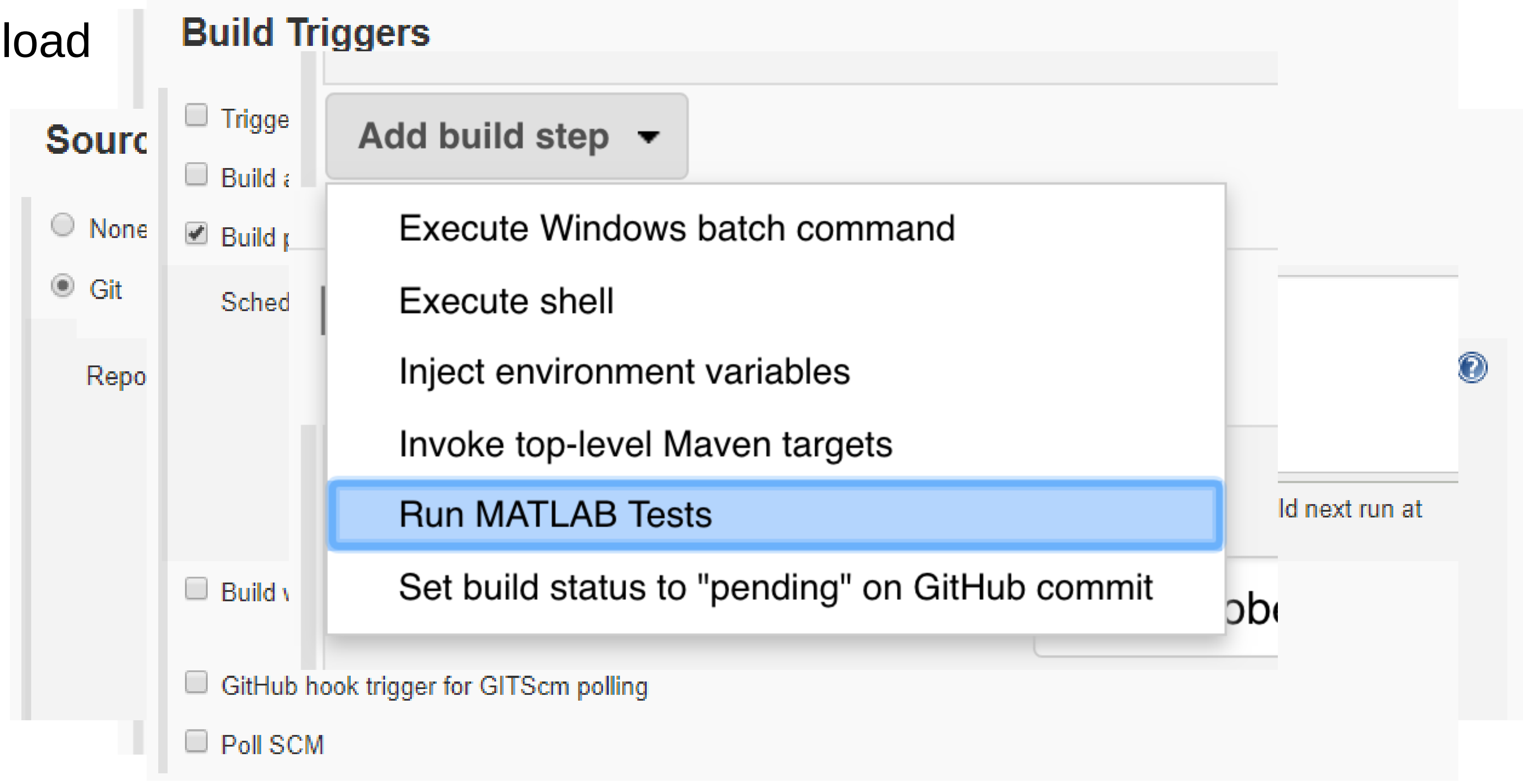
New Plugins
- QRebel
- MATLAB
- MISRA Compliance Report
- Zoom
- CodeBuilder: AWS CodeBuild Cloud Agents

Recently updated
- Mercurial
- VectorCAST Execution
- Klocwork Community
- OverOps Query
- LoadNinja
- QRebel

Trending
- jQuery UI
- Lockable Resources
- jQuery
- Analysis Model API
- Warnings Next Generation
- JDK Tool

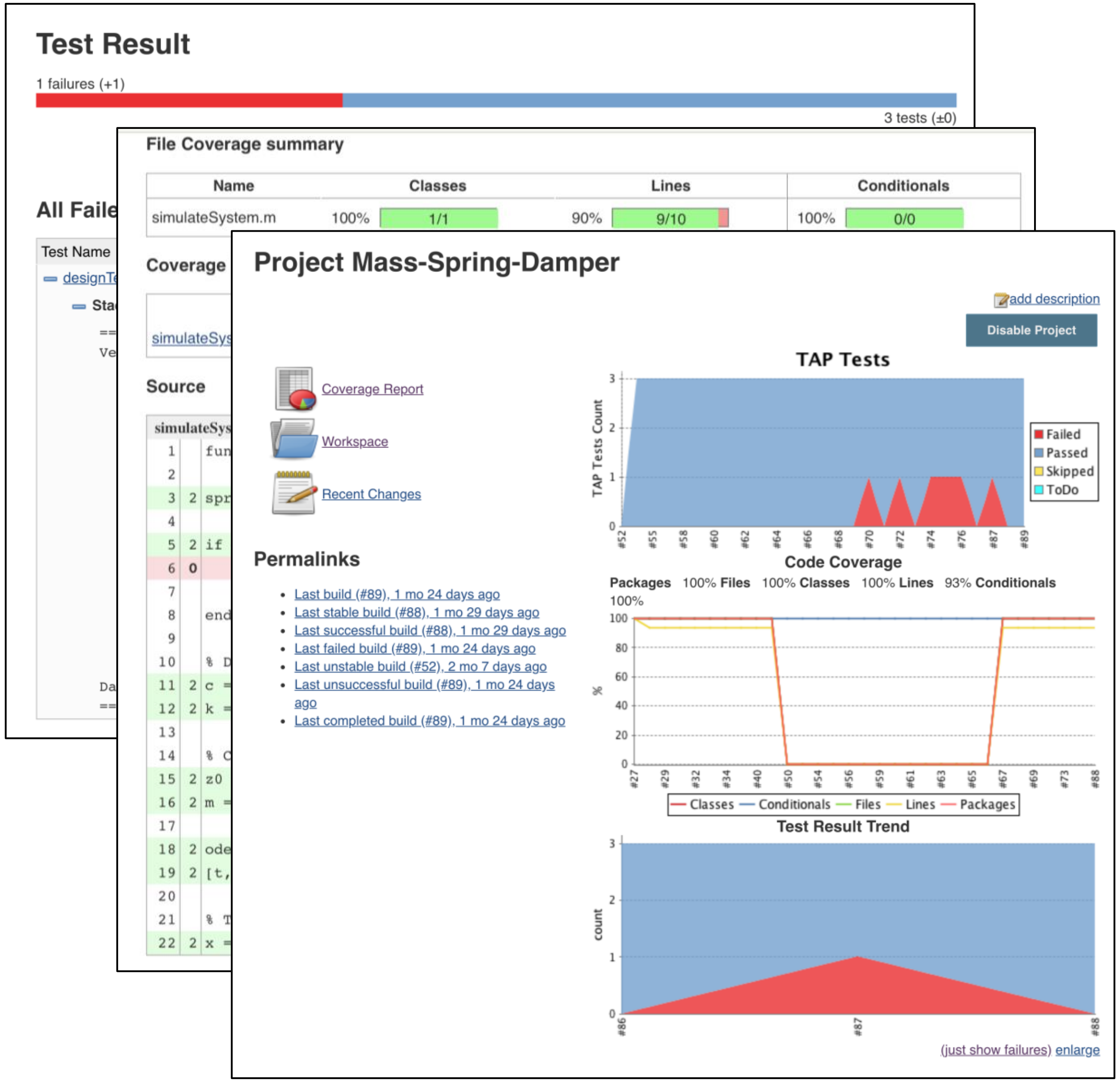
Automated Testing – Jenkins plugin – Configuration

- Easy configuration
  - Locate MATLAB
  - Identify repository to load
  - Set build triggers
  - Add build step


Build Triggers
Sourc
○ None
● Git
Repo
□ Trigge
□ Build a
☑ Build p
Sched
□ Build w
□ GitHub hook trigger for GITScm polling
□ Poll SCM
Add build step ▾
Execute Windows batch command
Execute shell
Inject environment variables
Invoke top-level Maven targets
Run MATLAB Tests
Set build status to "pending" on GitHub commit
ld next run at
obe

Automated Testing – Jenkins plugin – Testing reports

- View testing results
- View code coverage
- View testing reports



Test Result

1 failures (+1)

3 tests (±0)

All Faile

Test Name

designTe

Sta

Ve

Da

File Coverage summary

Name	Classes		Lines		Conditionals	
simulateSystem.m	100%	1/1	90%	9/10	100%	0/0

Coverage

simulateSys

Source

simulateSys

```
fun

2 spr

2 if
0

end

% D
2 c =
2 k =

% C
2 z0
2 m =

2 ode
2 [t,

% T
2 x =
```

Project Mass-Spring-Damper

add description

Disable Project

Coverage Report

Workspace

Recent Changes

Permalinks

- Last build (#89), 1 mo 24 days ago
- Last stable build (#88), 1 mo 29 days ago
- Last successful build (#88), 1 mo 29 days ago
- Last failed build (#89), 1 mo 24 days ago
- Last unstable build (#52), 2 mo 7 days ago
- Last unsuccessful build (#89), 1 mo 24 days ago
- Last completed build (#89), 1 mo 24 days ago

TAP Tests

Code Coverage

Packages 100% Files 100% Classes 100% Lines 93% Conditionals 100%

Test Result Trend

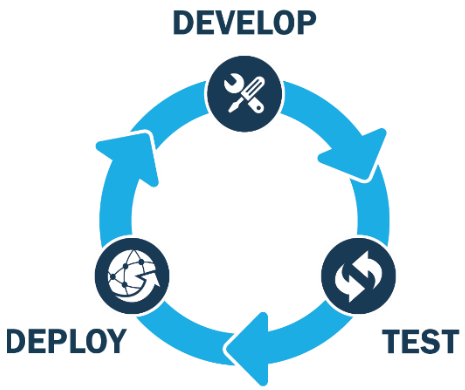
(just show failures) enlarge

Agenda

	Managing your code
	Tracking code changes and co-authoring workflows
	Writing better, robust, and portable code
	Testing and maintaining your code
	Summary

Key Takeaways

- You will save you time, effort, money, and frustration with good software development practices.
- MATLAB provides tools that enable agile software development.
- We're adding more software development tools and features every release!



MATLAB

is the **easiest** and
most **productive** environment
for **engineers** and **scientists**

Training
mathworks.com/services/training

Consulting
mathworks.com/services/consulting