

MATLAB EXPO

使用MATLAB和Simulink构建无缝协同软件开发流程

吴菁, MathWorks 中国高级应用工程师



项目有多复杂？

- 几百个文件？
- 很多的文件依赖性关系？
- 需要很复杂的设置？
- ...？

项目涉及到多少人参与？

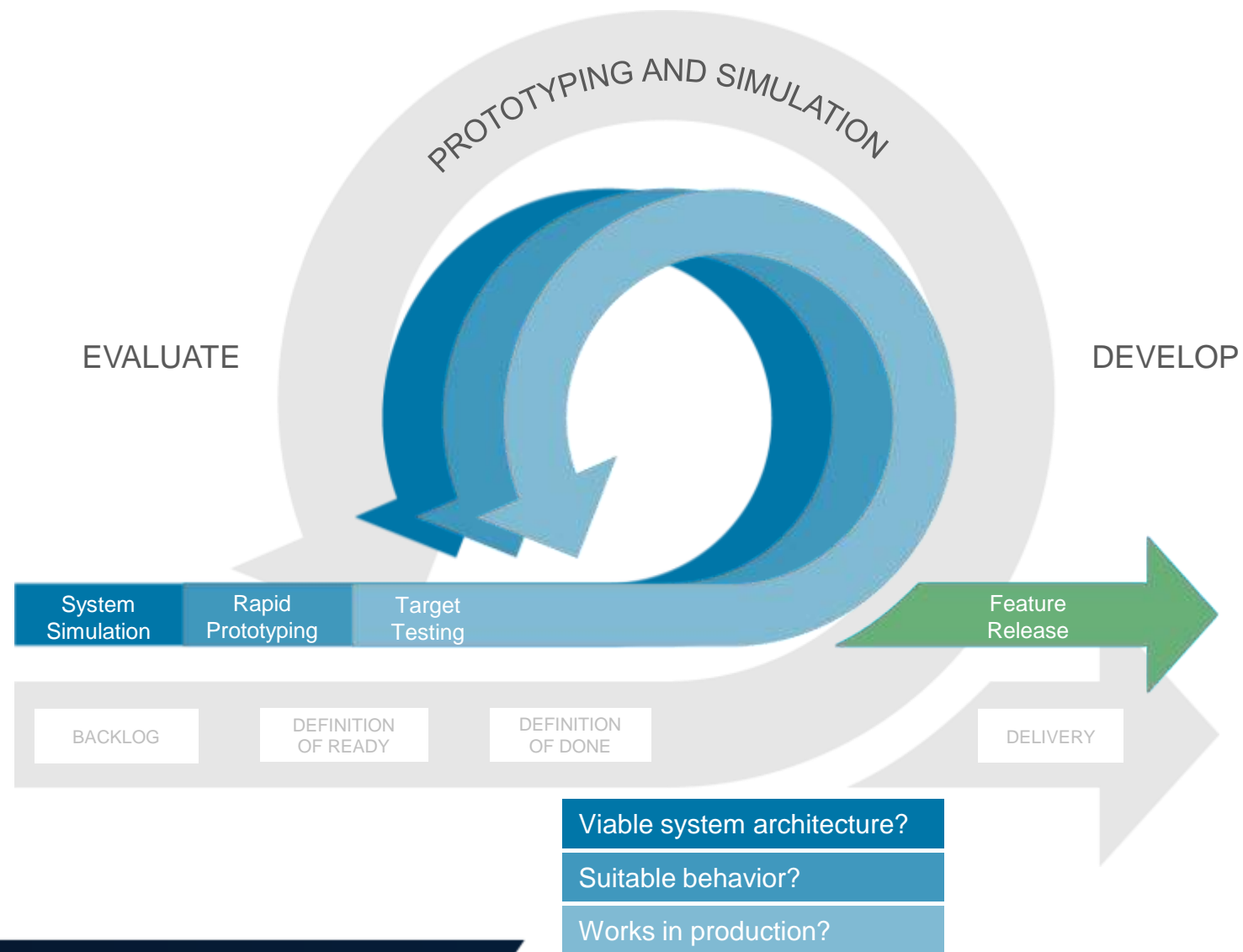
- 很多开发人员？
- 跨学科的团队？
- 团队遍布在世界各地？
- ...？

如何确保项目的质量？

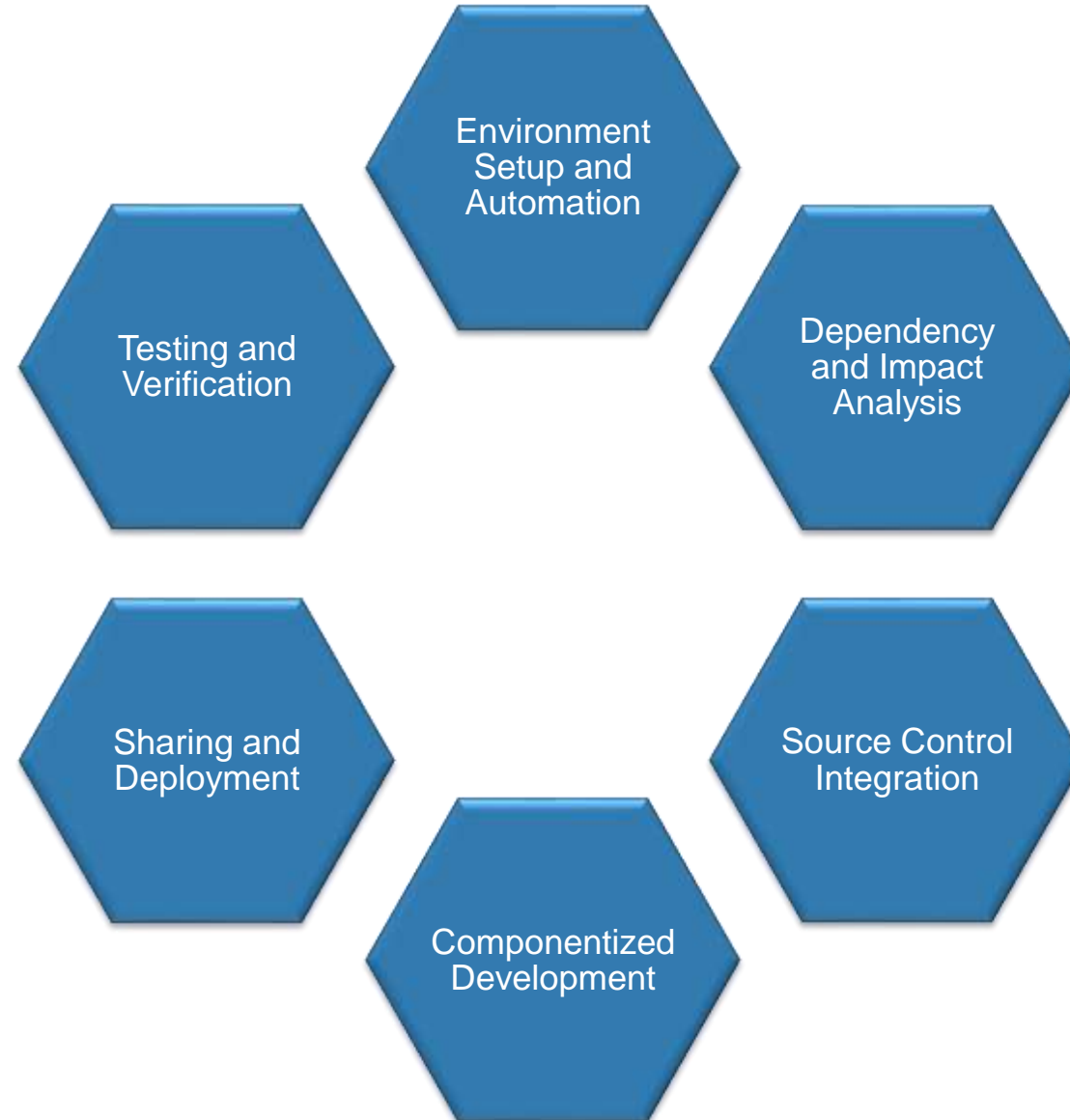
- 系统化测试？
- 编码/建模 遵循标准？
- 监管监督？
- 迷之自信？
- ...？

使用MATLAB和Simulink开发高质量软件

- 良好地软件开发实践有助于提高代码和模型质量
- 今天讨论的工具和实践支持敏捷开发工作流程




坚实的协作开发需要 ...



坚实的协作开发需要 ...



Agenda

	Setting up your development environment
	Managing team workflows
	Developing better code and models
	Testing and verification

开发过程中的挑战

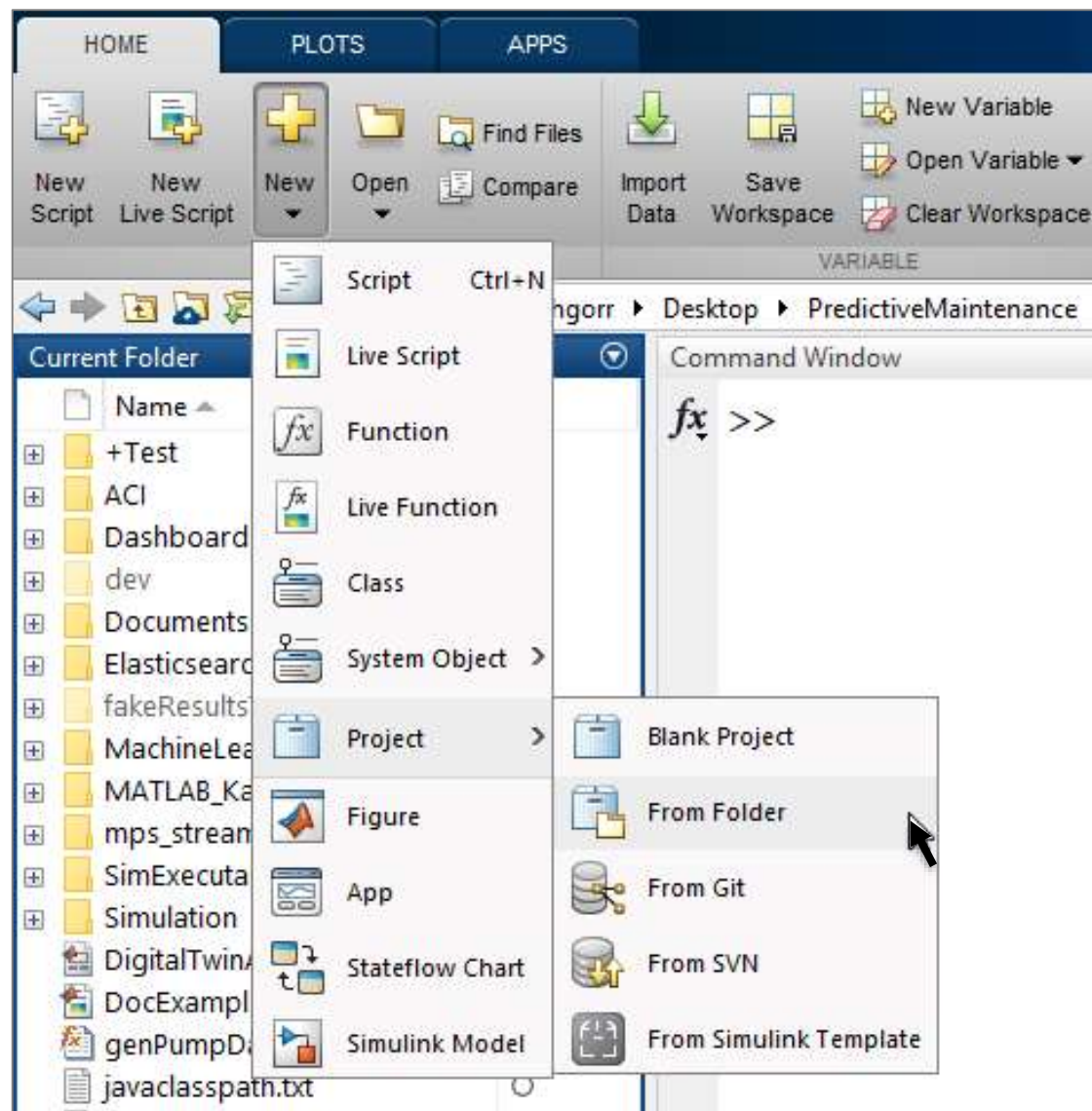
“在我的机器上可以工作,但在你的机器上就不行...”

- 不完整的文件集?
- 缺失了哪些文件?
- 不同的环境?
- 如何开始一个项目?
- ...



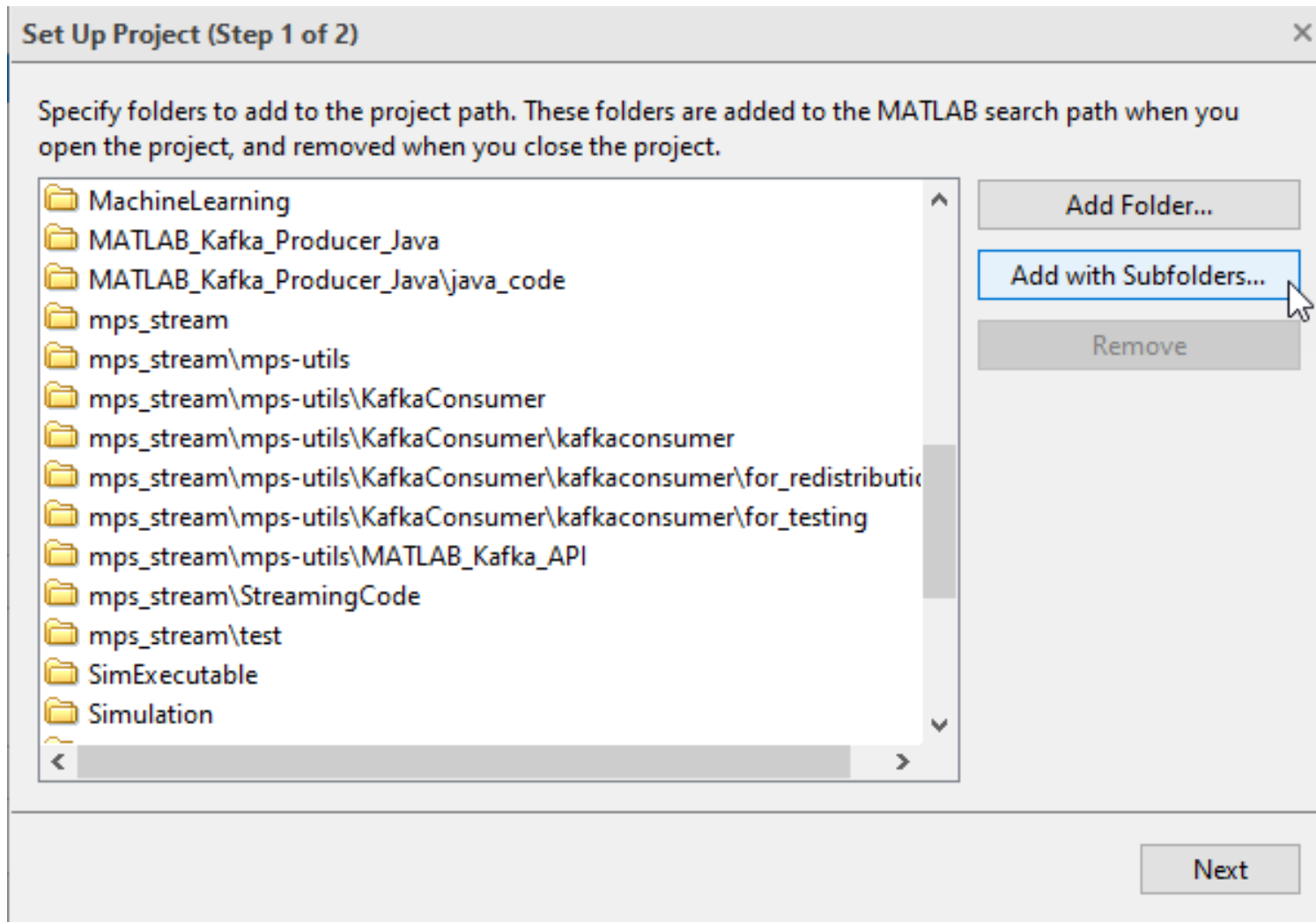
使用Projects管理工作

1. 创建project



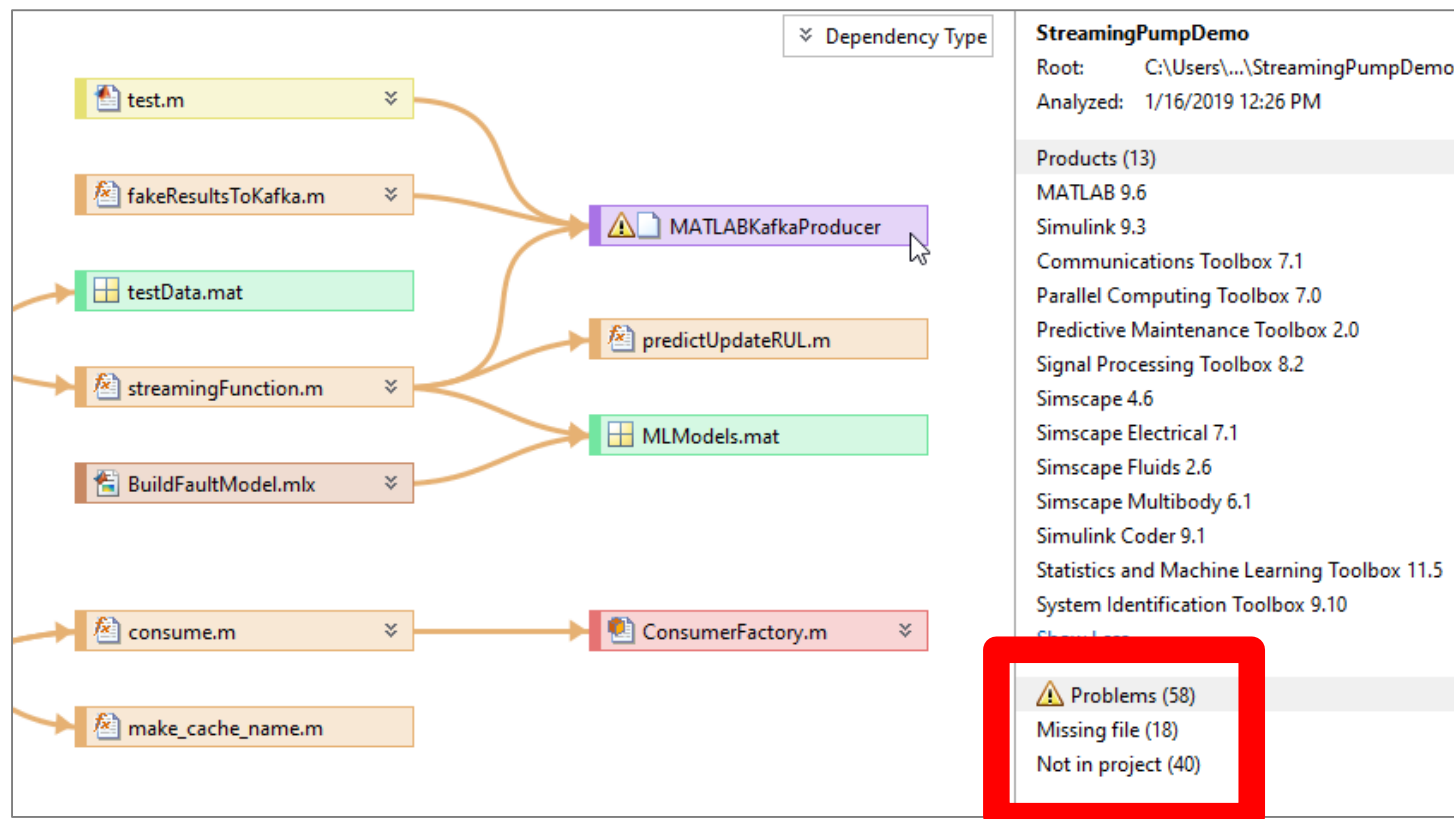
使用Projects管理工作

1. 创建project
2. 设置路径以及startup任务



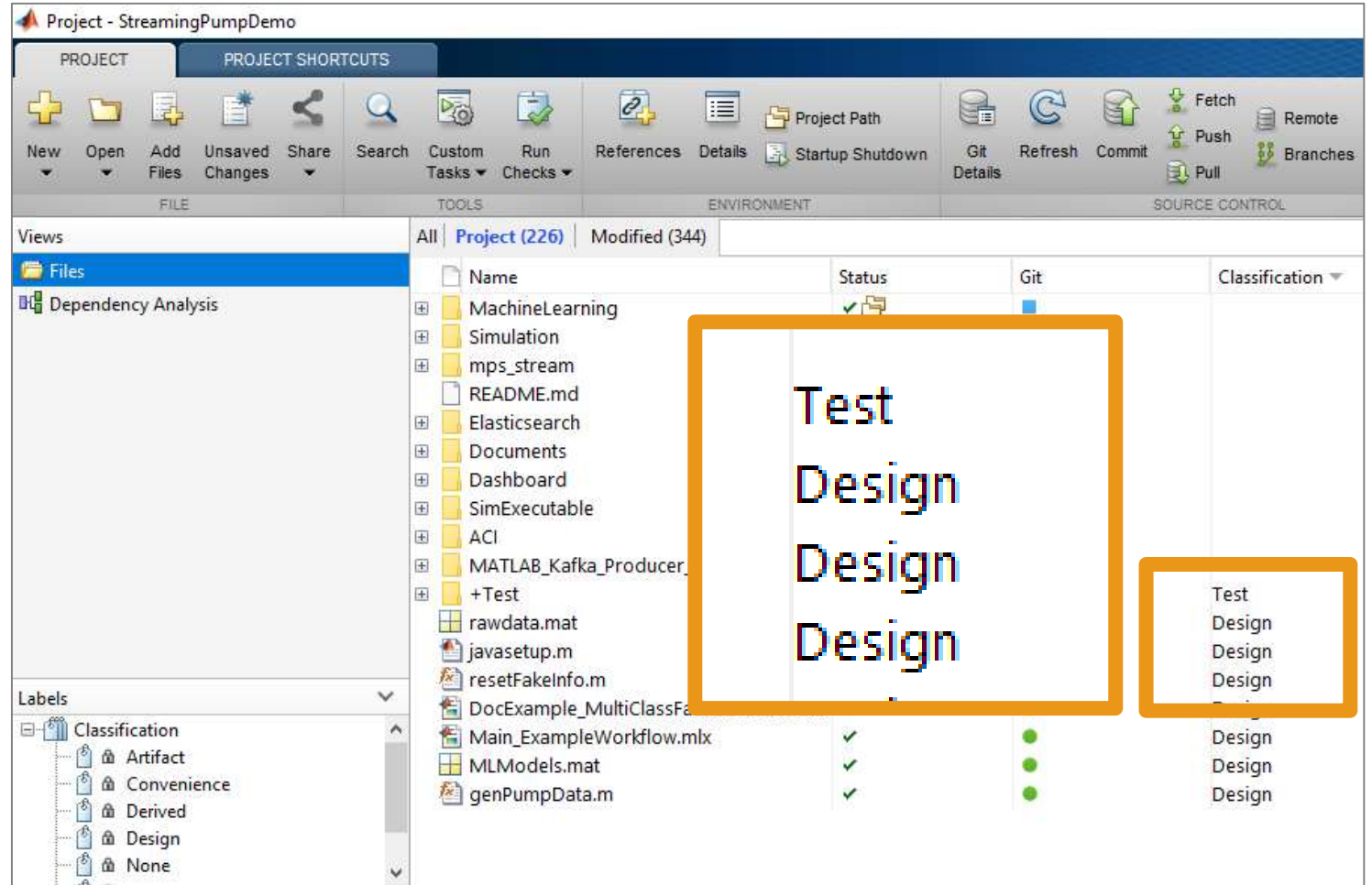
使用Projects管理工作

1. 创建 project
2. 设置路径以及startup任务
3. 查看依赖性关系



使用Projects管理工作

1. 创建 project
2. 设置路径以及startup任务
3. 查看依赖性关系
4. 为文件做标记

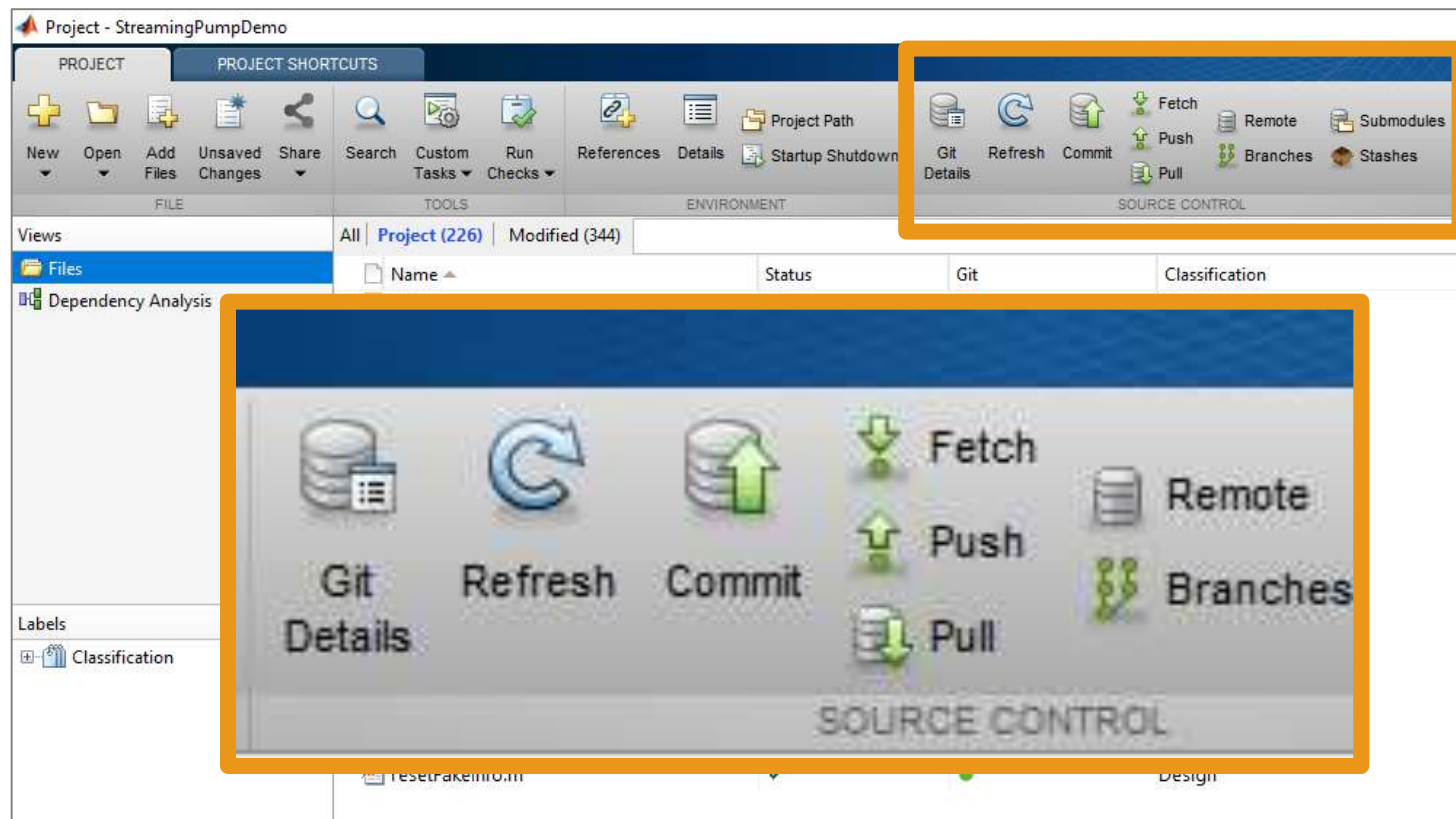


Identify and run tests locally

...and on Continuous Integration (CI) servers

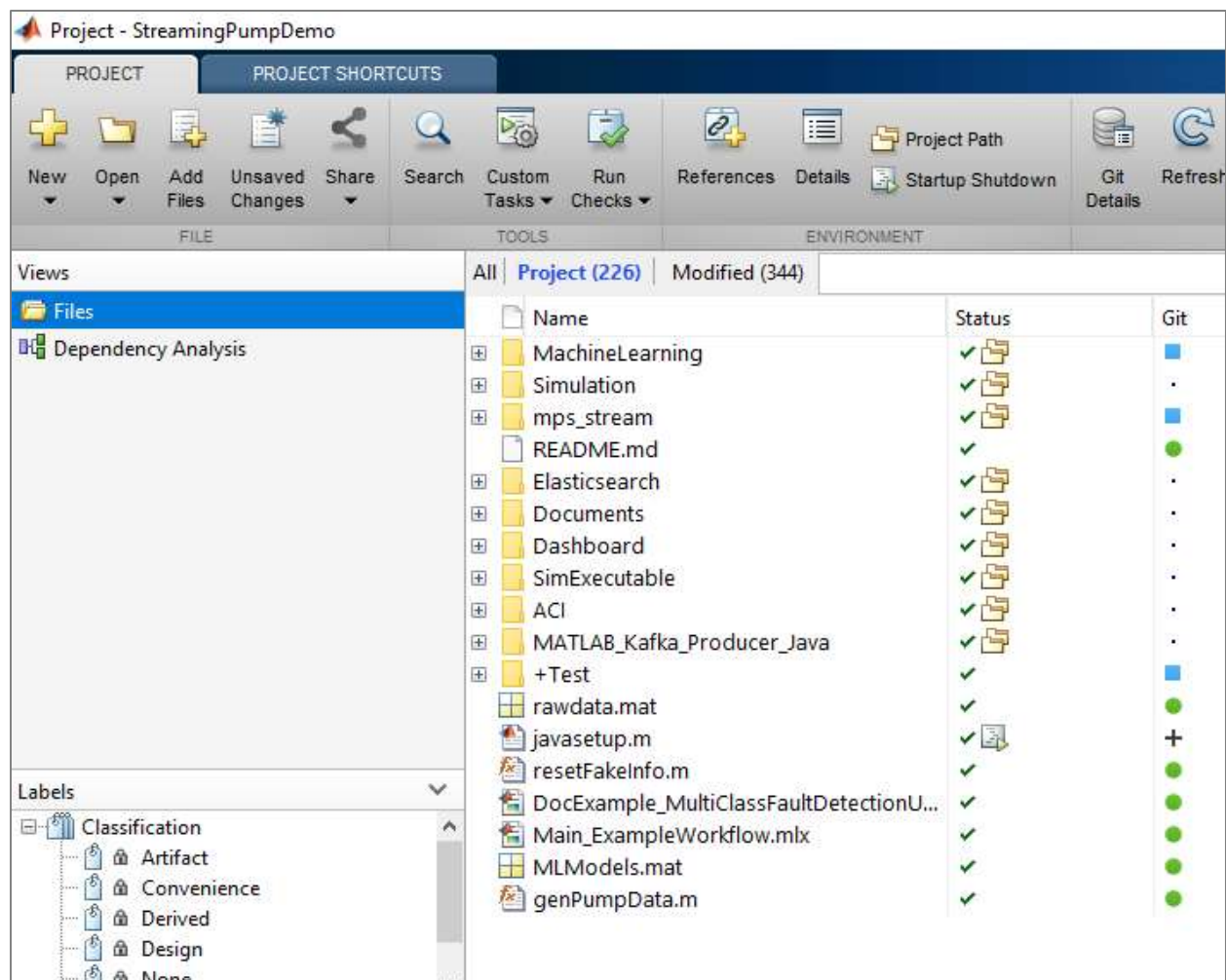
使用Projects管理工作

1. 创建 project
2. 设置路径以及startup任务
3. 查看依赖性关系
4. 为文件做标记
5. 集成源代码控制

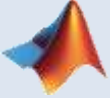


MATLAB和Simulink中的Projects

- 管理文件和路径
- 文件依赖性分析
- 功能重构
- 运行startup & shutdown 任务
- 创建 project shortcuts
- 标记和过滤文件
- 集成源代码控制



Agenda

	Setting up your development environment
	Managing team workflows
	Developing better code and models
	Testing and verification

团队开发中的挑战

“有人破坏了我的代码...”

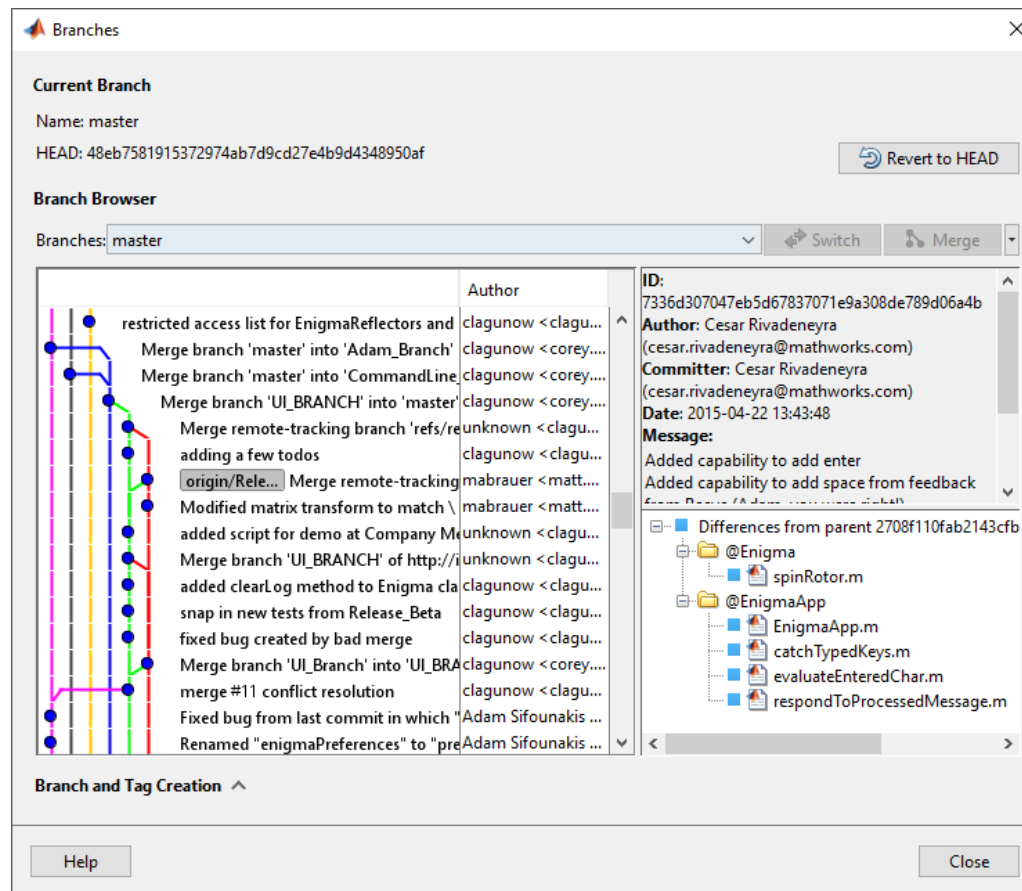
- 在不影响他人的情况下开发代码？
- 识别开发冲突的源头？
- 解决开发中的冲突？
- ...

源代码控制

- 对代码，模型，文档等变更进行管理的系统

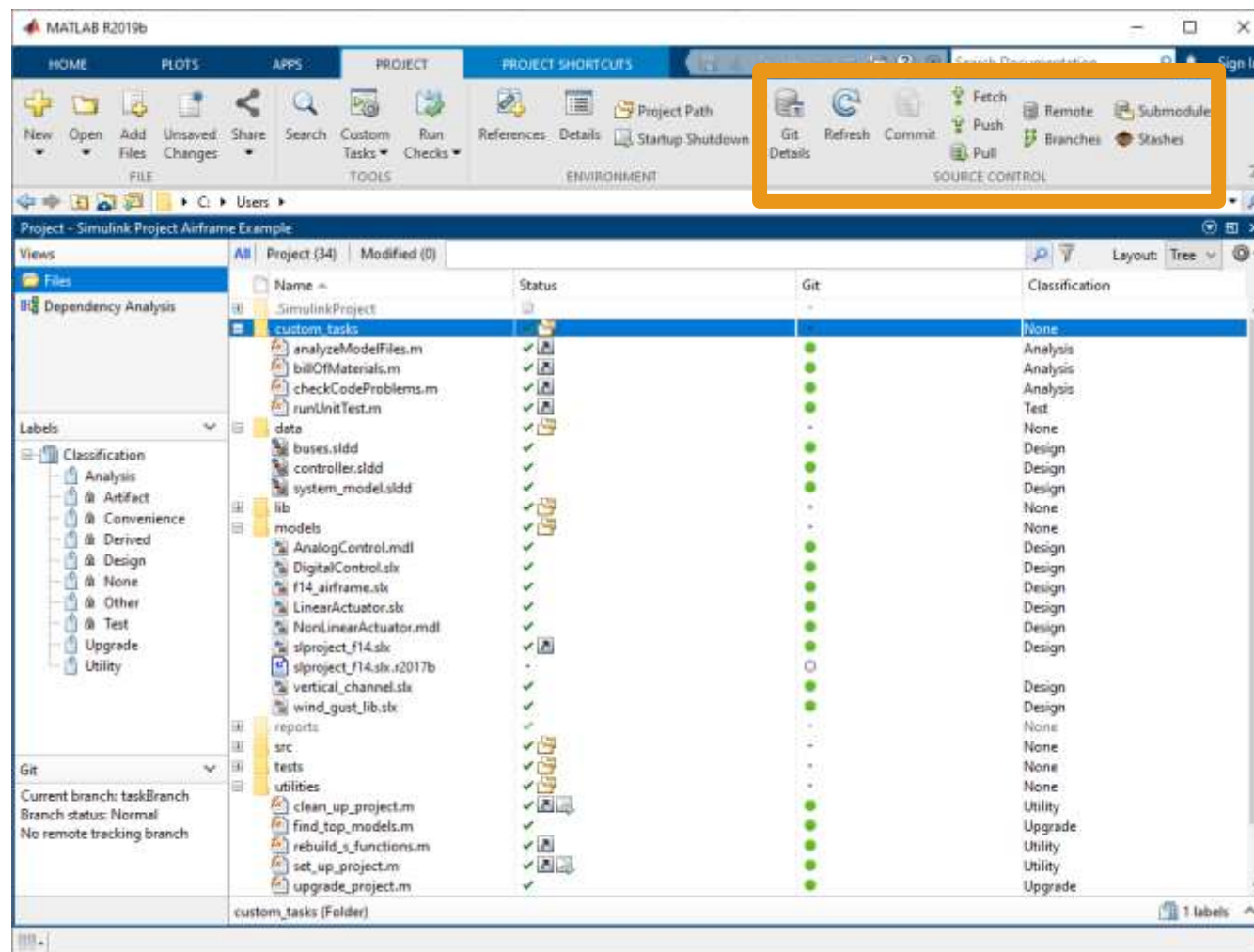
- 优势:

- 维护备份文件, 历史过程, 还可以恢复
- 跟踪变更和责任
- 调停冲突变更
- 讨论澄清
- 自保护



对源代码控制的集成

- 在MATLAB 和 Simulink中管理代码和模型
- Git 集成到:
 - Projects
 - Current Folder browser
- 使用Comparison Tool查看和合并不同版本间的变更



Comparison Tool 和合并

MATLAB

Comparison - Timetable_Example_Rev_521ad33e530dc8fe516d5fa2f1dad27a51bf8245.mlx vs. Timetable_Example.mlx

COMPARISON VIEW

Next Previous Refresh Merge Mode Find

NAVIGATE MERGE

Timetable_Example_Rev_521ad33e530dc8fe516d5fa2f1dad27a51bf8245.mlx vs. Timetable_Example.mlx

Timetable_Example_Rev_521ad33e530dc8fe516d5fa2f1dad27a51bf8245.mlx

Synchronizing with interpolation

synchronize combines data and sorts it, but does not interpolate or fill missing values by default. You can specify which type of interpolation to use for missing data with additional parameters.

```
8 ttLinear = synchronize(indoors,outdoors,'union','linear');
9 ttLinear(1:5,:)
```

Timetable_Example.mlx

Synchronizing with interpolation

synchronize combines data and sorts it, but does not interpolate or fill missing values by default. You can specify which type of interpolation to use for missing data with additional parameters.

```
8 ttLinear = synchronize(indoors,outdoors,'union','spline')
9 ttLinear(1:5,:)
```

+ Insertion - Deletion ≠ Modification

5 Differences

Comparison Tool 和合并

Simulink

The screenshot displays the Simulink Comparison Tool interface. The top menu bar includes 'COMPARISON', 'VIEW', and various icons. Below the menu are sections for 'COMPARISON', 'NAVIGATE', 'PARAMETER', and 'MERGE'. The main area shows two files being compared:

- File 1:** E:\jobarchive\BR2015bd\2015_05_26_h11m55s52_job234054_pass\matlab\toolbox\rptgenext\rptgenextdemos\slxmlcomp\slxml_sfcar_1.slx
Saved in Simulink version: R2015b
E:\...lxmlcomp\slxml_sfcar_1.slx
- File 2:** E:\jobarchive\BR2015bd\2015_05_26_h11m55s52_job234054_pass\matlab\toolbox\rptgenext\rptgenextdemos\slxmlcomp\slxml_sfcar_2.slx
Saved in Simulink version: R2015b
E:\...lxmlcomp\slxml_sfcar_2.slx

Below the file information are two 'Model Hierarchy' panels. The left panel shows the hierarchy for 'slxml_sfcar_1', and the right panel shows the hierarchy for 'slxml_sfcar_2'. The hierarchies are color-coded to show differences between the two models.

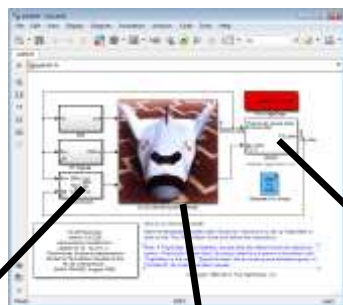
The right side of the image shows two screenshots of the Simulink GUI. The top screenshot is for 'slxml_sfcar_1' and the bottom is for 'slxml_sfcar_2'. Both show the same block diagram of a car model, which includes components like 'User Inputs', 'Engine', 'Transmission', and 'Vehicle'. The diagrams are annotated with text:

- Top Screenshot (slxml_sfcar_1):** Includes the instruction "Choose Run from the Simulation menu to run the simulation." and "Double-click to open the GUI and select an input maneuver".
- Bottom Screenshot (slxml_sfcar_2):** Includes the instruction "Choose Run from the Simulation menu to run the simulation." and "Double-click to open the GUI and select an input maneuver".

The status bar at the bottom of the GUI shows 'Ready', '67%', and 'ode5'.

利用模型引用和项目引用管理复杂性

Model Reference



Project Reference

A screenshot of the Simulink Project - ConVeh interface. The 'References' pane is highlighted with an orange box and contains a tree view of project references:

- AutoTransmission
- Auto Transmission
- Brake System
- Electrical
 - Climate Control
 - Radar
- Engine


Below the references, a table shows the project structure:

Name	Status	SVN	Revision
branches	✓	●	1
data	✓	●	7
lib	✓	●	7
models	✓	■	8
Actuator.mdl	✓	■	7
Actuator.mdl.r201...	·	·	·
AnalogControl.mdl	✓	●	8
Analog Control.mdl	✓	●	7
AutomaticTransmi...	✓	■	7

The 'Actuator.mdl' row is selected. Below the table, the 'Actuator.mdl (Simulink Model)' details are shown:

- Name: AutoTransmission
- Checkpoint: August 7, 2018 3:11:50 PM
- Description:
- Project root: H:\Documents\MATLAB\Automotive Projects\Auto Transmissi

Agenda

	Setting up your development environment
	Managing team workflows
	Developing better code and models
	Testing and verification

“好”设计是由什么定义的？

- 更快？
- 更高效的内存？
- 更好地组织？
- 更稳定？
- 更易维护？
- ...

YES!



在MATLAB 和 Simulink中开发强健的软件系统

- 编写更好更快的代码
- 使用重构降低复杂性
- 和其他语言和工具集成
- 共享和重用

升级到最新版本的 MATLAB 和 Simulink

- 代码兼容性报告

- Upgrade Advisor

The screenshot displays the MATLAB Upgrade Advisor interface. On the left, a 'Code Compatibility Report' window shows a table of 'Incompatibility and Syntax Errors' for the file 'classifyBloodPressure.m'. The table has columns for Row, Filename, Line, and Description. Three errors are listed, with line numbers 18, 21, and 24 highlighted in orange. Below this, a 'Warnings and Other Recommendations' section is partially visible. In the center, the 'Upgrade Advisor' window is open, showing a list of checks with green checkmarks. A 'Model Advisor' panel is also visible at the bottom left. On the right, a 'Check usage of function-call connections' window shows a warning about 'Context-dependent inputs' settings, with a 'Recommended Action' to set the diagnostic to error. This action is highlighted in orange. At the bottom, a yellow box lists other advisors: Model, Code Generation, and Performance.

Row	Filename	Line	Description
1	classifyBloodPressure.m	18	TREEFIT has been replaced by TREEFIT2.
2	classifyBloodPressure.m	21	TREEDISP has been replaced by TREEDISP2.
3	classifyBloodPressure.m	24	TREEVAL has been replaced by TREEVAL2.

Go directly to the line of code


Recommended actions

还有Model, Code Generation, 以及 Performance Advisors

简化函数的参数校验及错误检查

```
% Error check required input arguments
if nargin < 1
    error("rectangle requires width and height values");
elseif ~isnumeric(width) || ~isscalar(width)
    error("width must be a scalar numeric value")
elseif ~isnumeric(height) || ~isscalar(height)
    error("height must be a scalar numeric value")
end

% Process optional inputs xStart and yStart
xStart = 0;
if nargin > 2 && isnumeric(varargin{1}) && isscalar(varargin{1})
    xStart = varargin{1};
end
yStart = 0;
if nargin > 3 && isnumeric(varargin{2}) && isscalar(varargin{2})
    yStart = varargin{2};
end
```

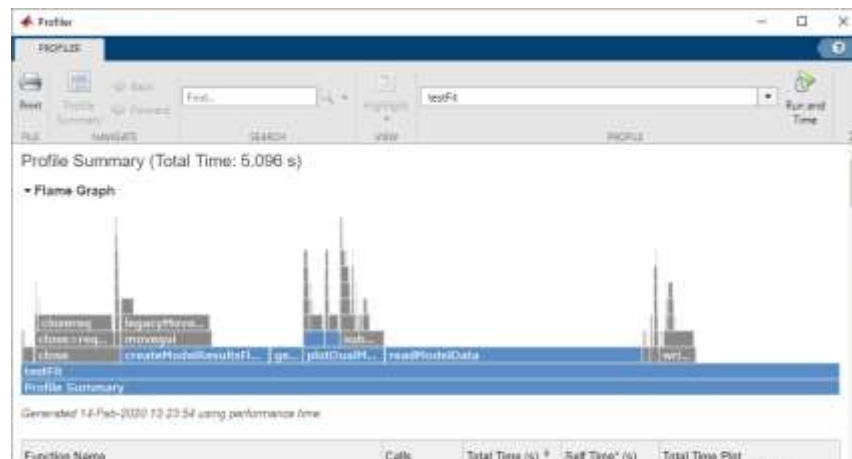


```
arguments
width (1,1) double {mustBeNumeric}
height (1,1) double {mustBeNumeric}
xStart (1,1) double {mustBeNumeric} = 0; % optional
yStart (1,1) double {mustBeNumeric} = 0; % optional
end
```

提升代码和模型的性能

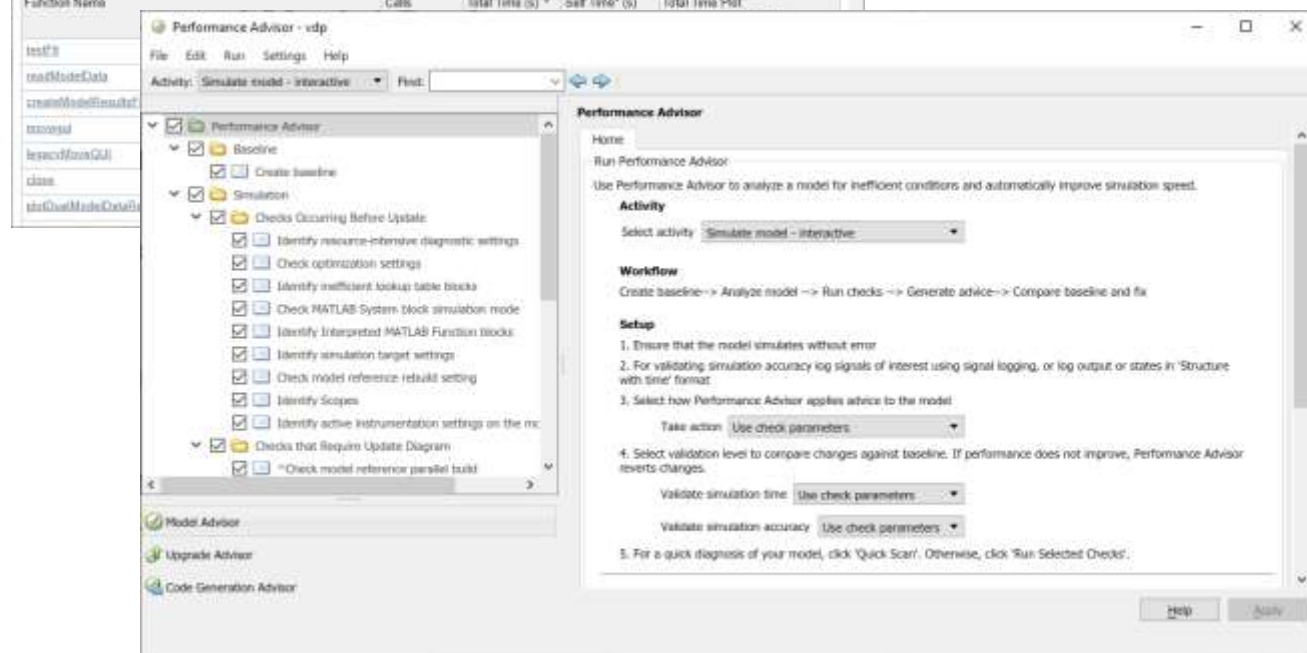
■ MATLAB Profiler

- 火焰图高亮代码瓶颈
- 函数调用的次数
- 每个函数调用所花的时间
- 代码覆盖度



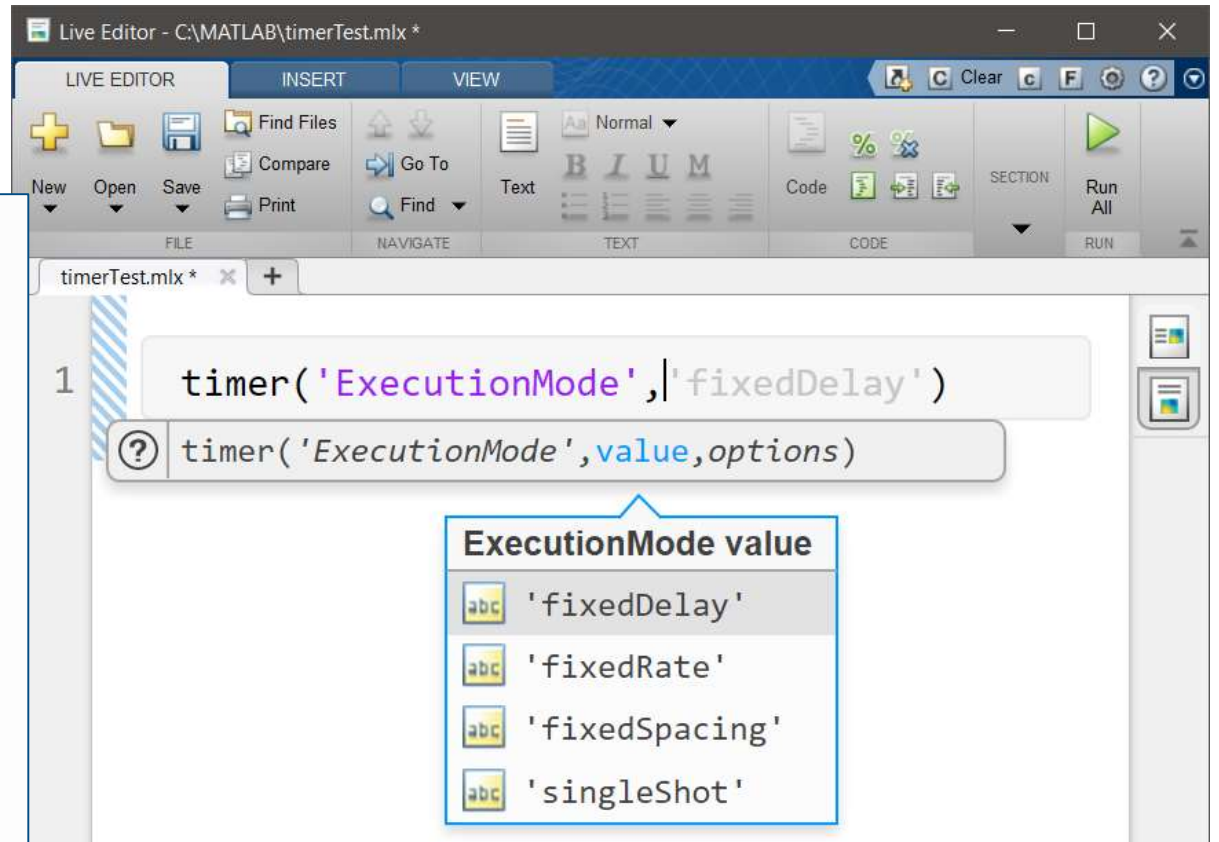
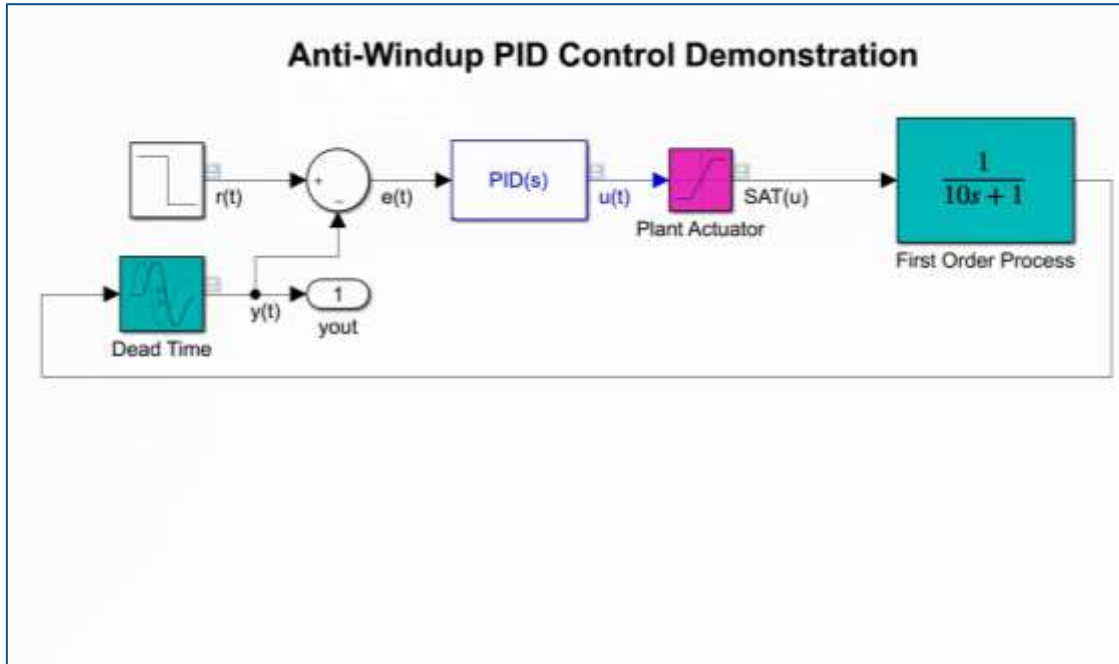
■ Simulink中的Performance Advisor

- 创建基线作为量测对比基础
- 检查推荐选项并自动完成更改



加速开发

- 上下文感知编码指南
 - 自动建议函数，变量，文件以及Name-Value 对
- Model layout tools
 - 自动清理凌乱视图并完成模型的布局

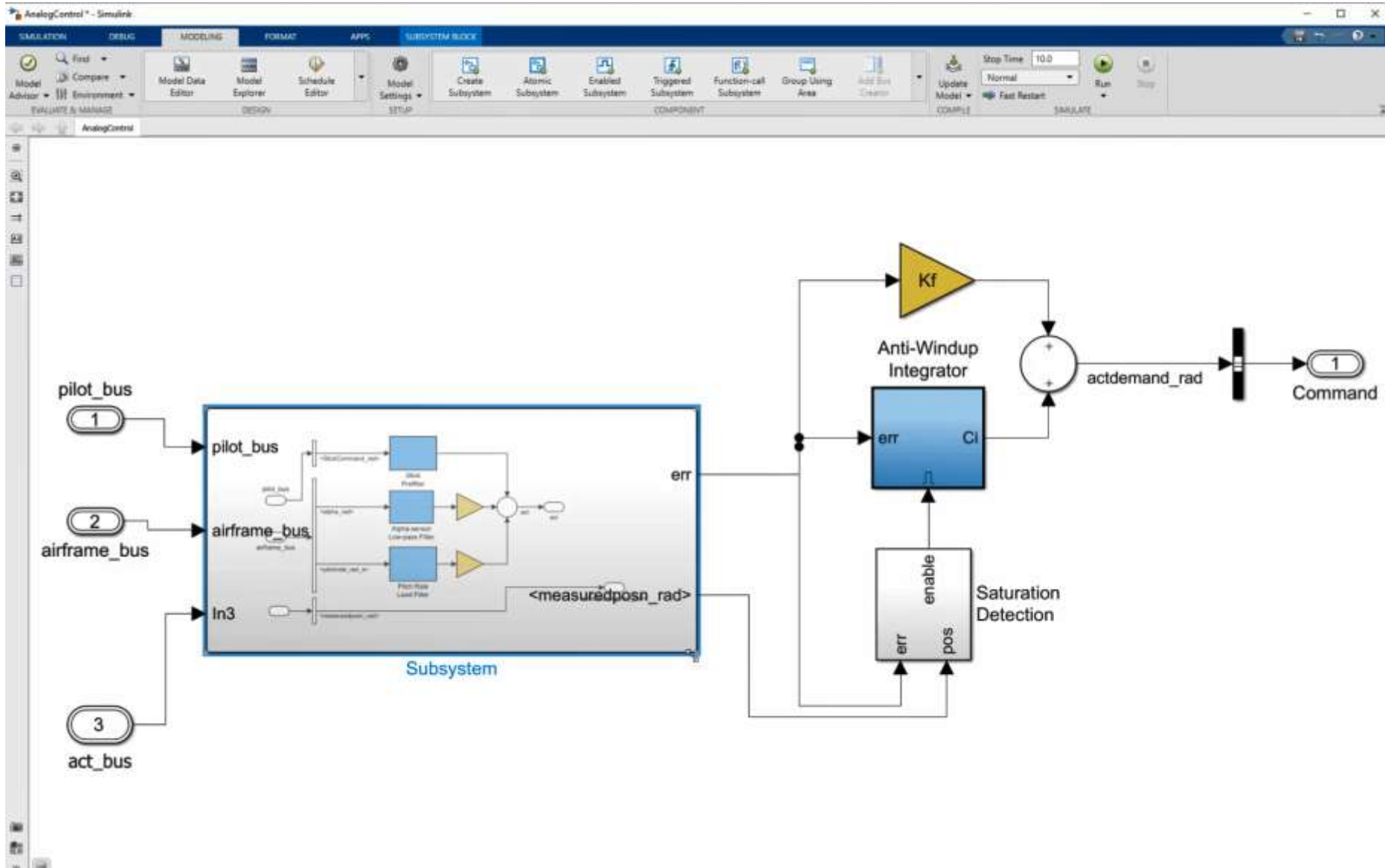


快速安全的重构 – MATLAB 代码

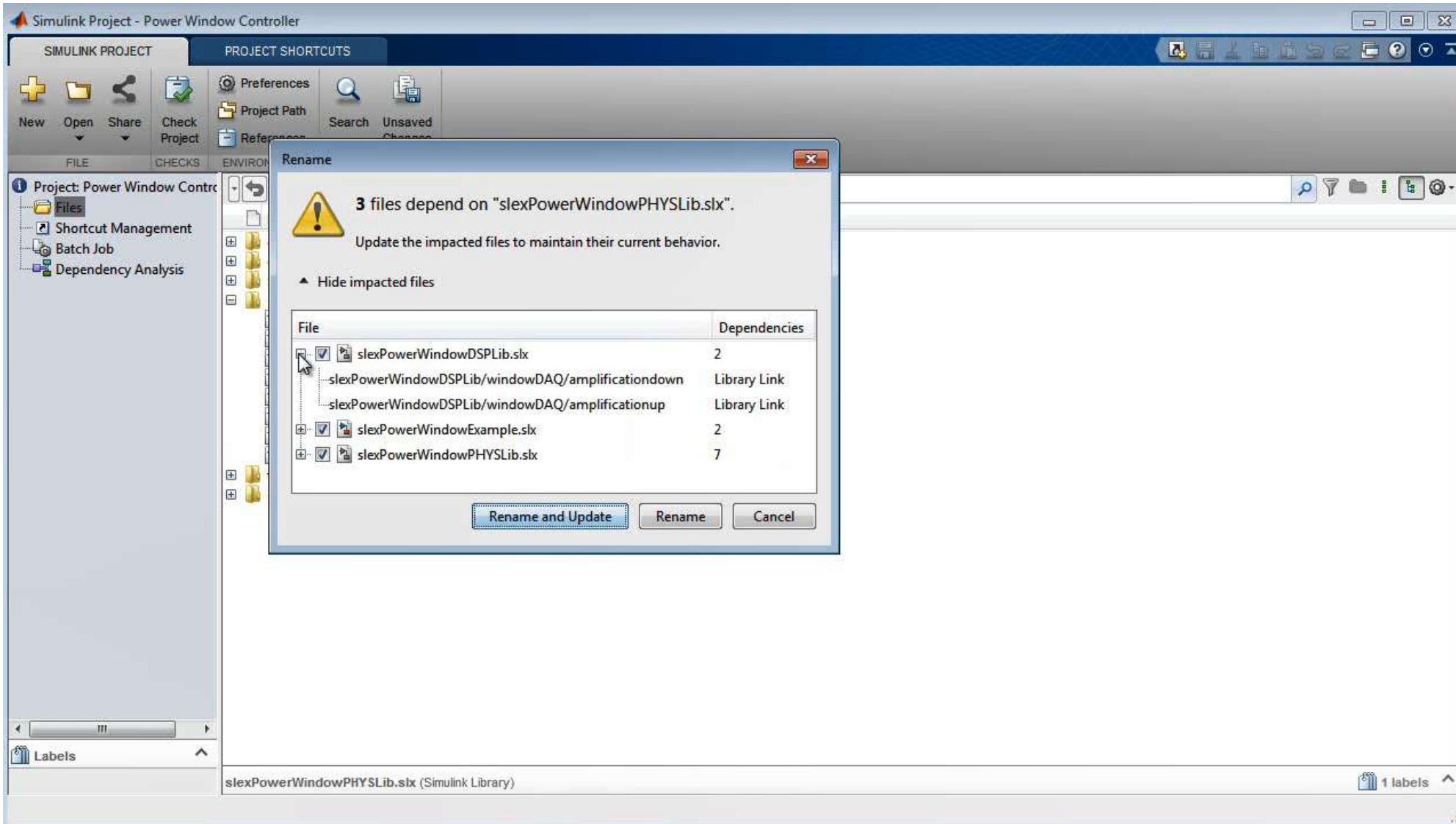
- 将大型复杂的代码和模型分解为可重用且易于维护的组件

The image shows a MATLAB Live Editor window with two code snippets. The left snippet, titled "Calculate my answer:", contains lines 3-7: `z1 = x+y;`, `z2 = x-y;`, `z3 = y-x;`, `z4 = x*y;`, and `zSum = z1 + z2 + z3 + z4;`. The right snippet, titled "Display answers of interest:", contains lines 8-9: `disp(z3)` and `disp(zSum)`. A context menu is open over the code, with "Convert to Function" and "Convert to Local Function" highlighted. An orange arrow points from the "Convert to Function" option to a separate window titled "Live Editor - C:\MATLAB\TestFolder\myMathFunction.mlx". This window shows the resulting function code: `function [z3, zSum] = myMathFunction(x, y)`, `z1 = x+y;`, `z2 = x-y;`, `z3 = y-x;`, `z4 = x*y;`, `zSum = z1 + z2 + z3 + z4;`, and `end`. Orange boxes highlight the function signature and the function name in both windows.

快速安全的重构 – Simulink 模型



快速安全的重构— 函数和模型名



与其他编程语言的集成

调用其他编程语言中的库函数



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

从其他编程语言中调用MATLAB



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

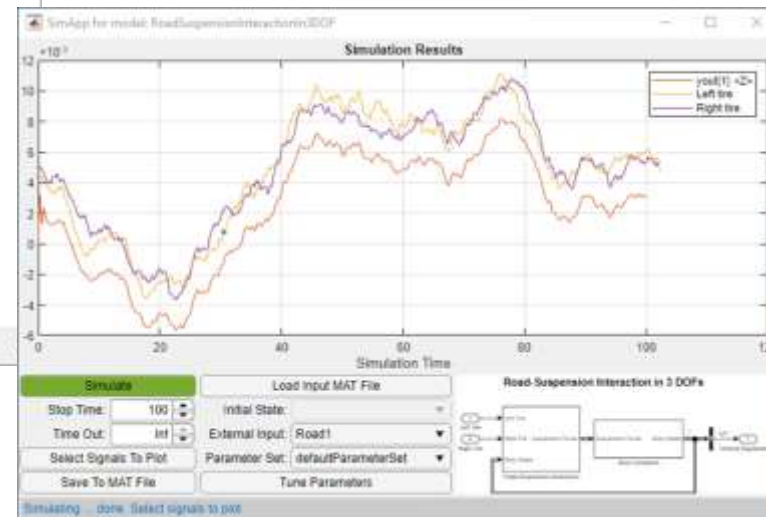
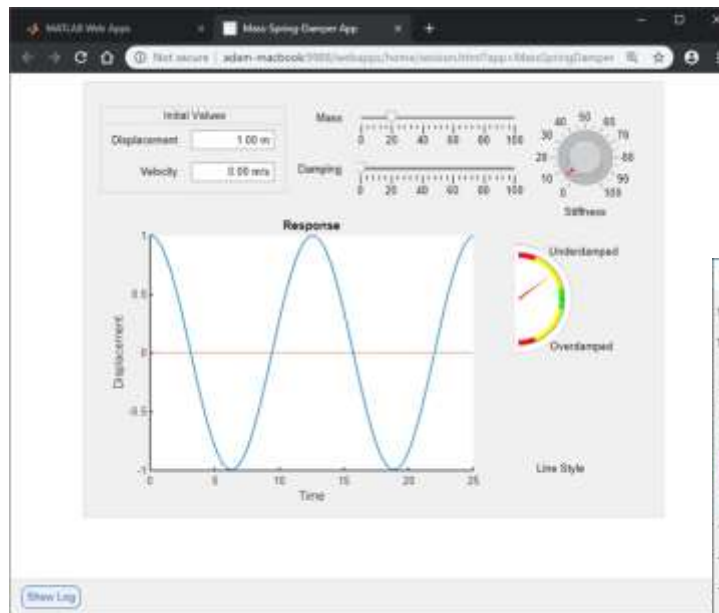
与其他编程语言的集成

The central image features the Simulink logo in blue text, accompanied by a blue triangle and an orange square icon. Below the logo are three 3D models: a transparent car chassis, an Airbus A380 airplane, and a Vestas wind turbine. Surrounding this central image are several logos and icons in rounded rectangular frames:

- Top row: IPG AUTOMOTIVE, a MATLAB toolbox screenshot, the MATLAB logo, the Simscape logo, C++ text, the fmi logo, and the MODELICA logo.
- Second row: GT logo with a red square icon, the Simulink logo with its icon, and the python logo.
- Third row: A blue sports car, a car chassis with a green sensor beam, and a car chassis with a blue sensor beam.
- Bottom row: A car presentation video thumbnail with a '11' icon and the word 'PRESENTATION', and the ROS logo with two robot icons.

工作的共享

- 合作者和开发团队成员
 - Projects
- MATLAB 和 Simulink的最终用户
 - Toolbox 或 App
- 没有 MATLAB 和 Simulink的最终用户
 - 独立可执行和web应用
 - 特定语言的库
 - 自动生成的独立可执行代码
 - 微服务 APIs



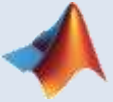
MATLAB Compiler, Simulink Compiler

MATLAB Compiler SDK

Embedded Coder, HDL Coder, PLC Coder, GPU Coder, ...

MATLAB Production Server

Agenda

	Setting up your development environment
	Managing team workflows
	Developing better code and models
	Testing and verification

软件的维护—隐藏的开发成本

- 如何确保代码和模型不会随着时间的推移而被破坏？
- 如何防止新功能破坏现有功能？
- 如何保持信心，系统会一直按预期运行？
- 如何确保软件能够适应未来需求？



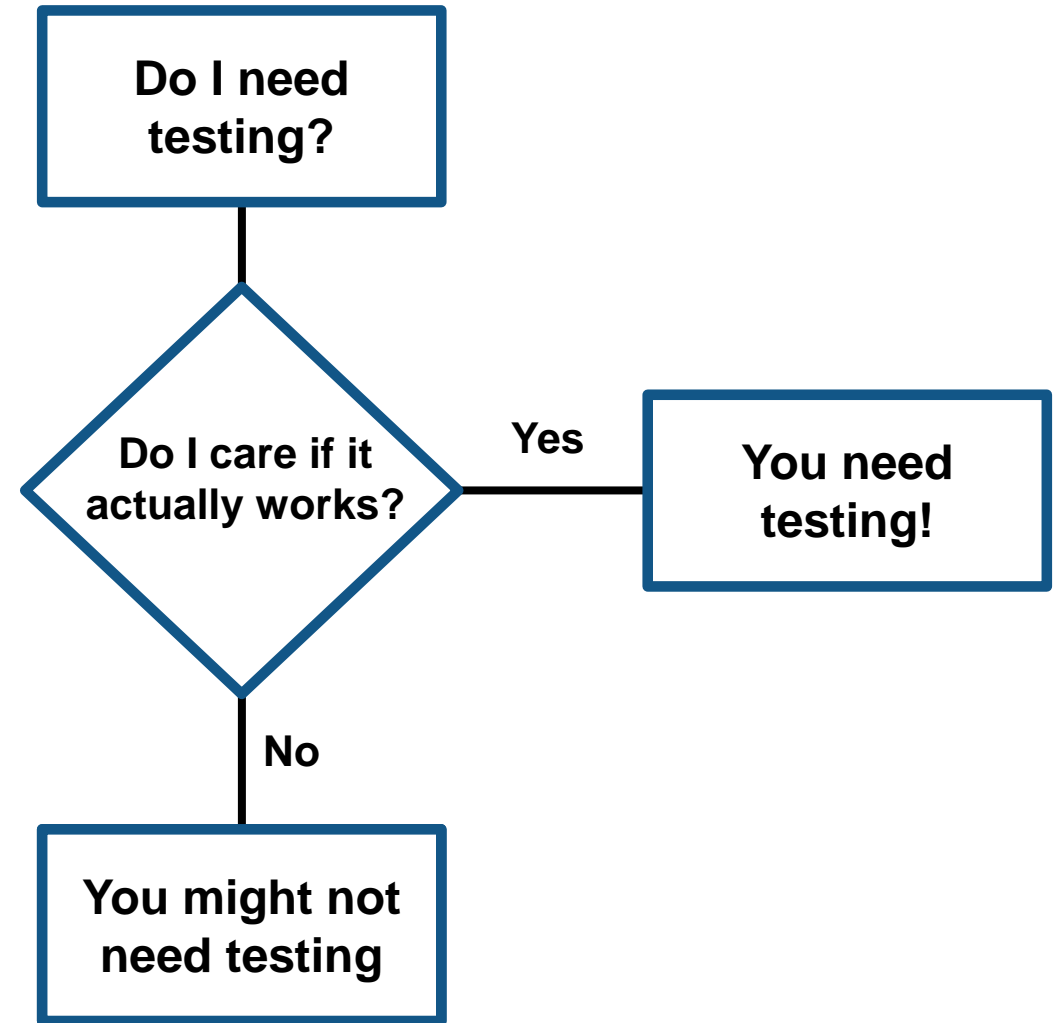
Journal paper: “Faster issue resolution with higher technical quality of software”, Software Quality Journal, 2011

尽早测试，经常测试，自动测试

- 降低软件损坏的风险
- 尽早发现问题
- 提高品质
- 记录预期行为

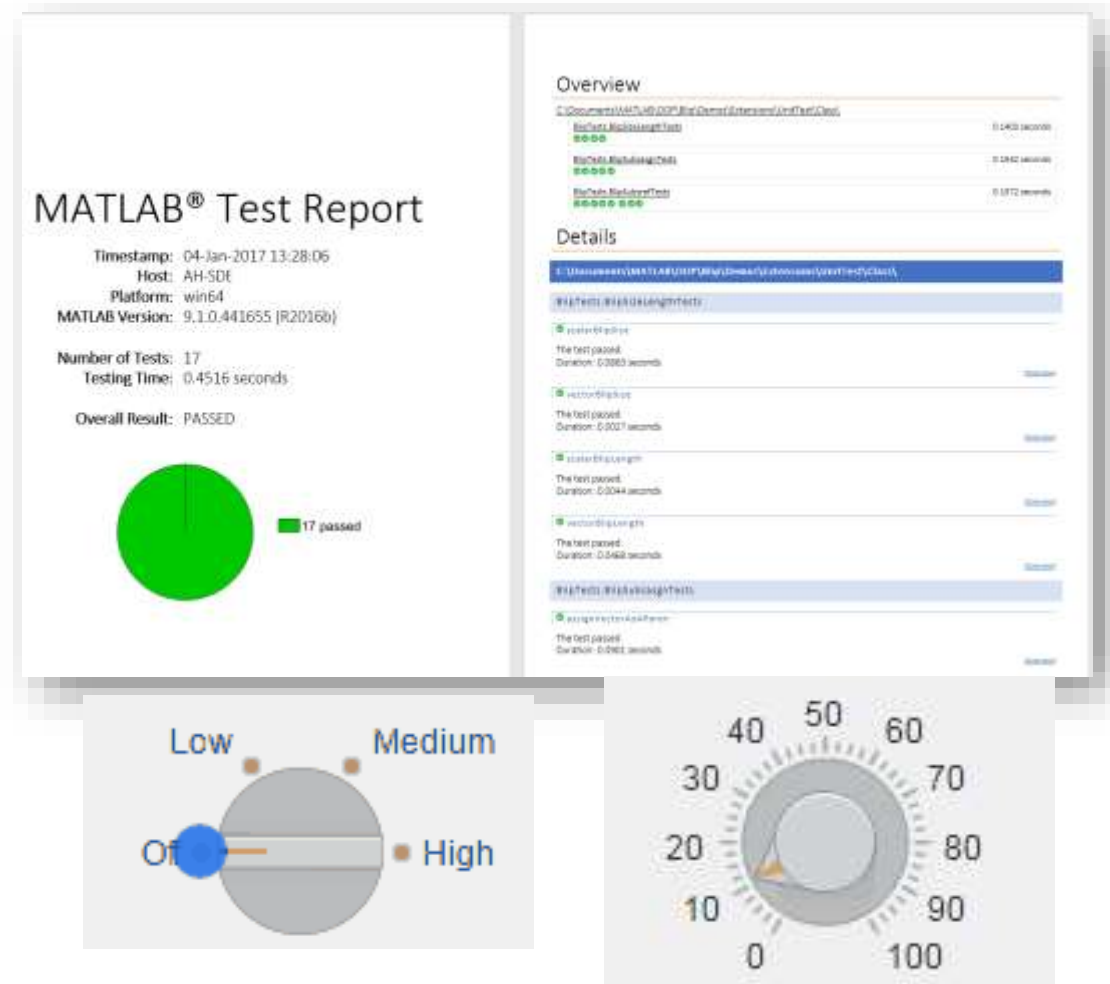


Credit: <http://geek-and-poke.com/>

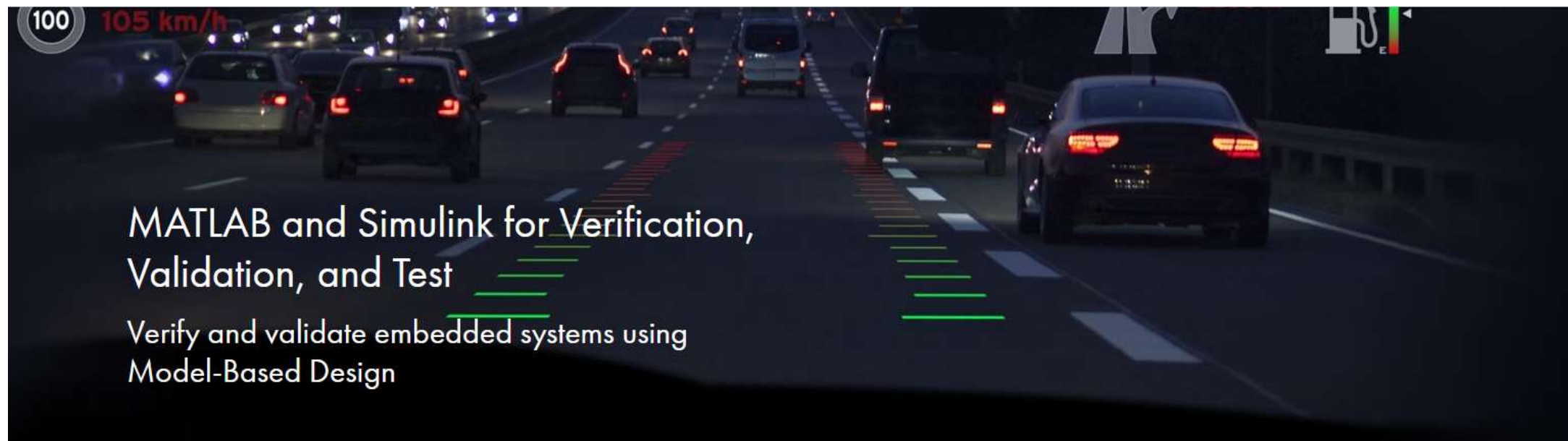


MATLAB 测试框架

- MATLAB 单元测试框架
- 性能测试框架
- Mocking 框架
- App 测试框架



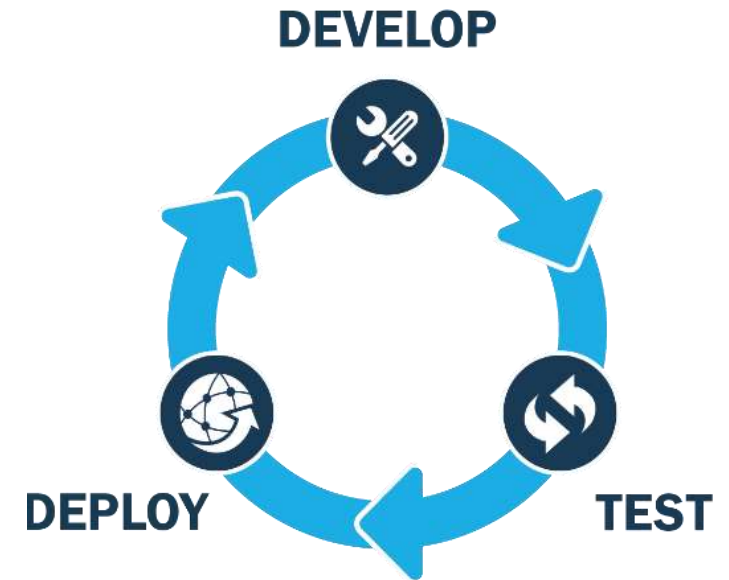
Simulink中的测试和验证



- **需求追溯** 架构，设计，测试和代码
- **验证设计** 符合需求且没有严重的运行时错误
- **检查合规性** 衡量模型和代码的质量
- **自动生成测试用例** 扩大测试范围
- **生成报告** 证明符合标准(比如 [DO-178](#) 和 [ISO 26262](#)).

利用持续集成(CI)进行自动化的测试

- 一个在开发和维护代码时自动编译，测试，集成和部署代码的系统
- 常用的 CI 系统: Jenkins, Travis, CircleCI , Azure DevOps, 等...
- 收益:
 - 早期检测集成bug
 - 阻止错误被接受
 - 跟踪测试历史并生成报告
 - 灵活的测试调度和触发



和 Jenkins集成的MATLAB插件

- 直接从Jenkins插件管理器安装Jenkins的MATLAB插件
- 使用Jenkins轻松连接和配置MATLAB
- 安排自动化的代码和模型测试
 - MATLAB Unit Test Framework
 - Simulink Test



The screenshot shows the Jenkins Update Center interface. The browser address bar displays `https://testserver.com/j/pluginManager/available`. The page header includes the Jenkins logo, the name "Adam Sifounakis", and a "log out" link. Below the header, there are navigation links for "Back to Dashboard" and "Manage Jenkins". A search filter is set to "MATLAB". The "Available" tab is selected, showing a table of plugins. The "MATLAB" plugin is listed with a checked checkbox, a description, and version "1.1.2". Below the table, there are three buttons: "Install without restart" (highlighted with an orange box), "Download now and install after restart", and "Check now". The footer indicates "Update information obtained: 4 min 43 sec ago" and "Page generated: Feb 20, 2020 4:13:08 PM EST".

Install	Name	Version
<input checked="" type="checkbox"/>	MATLAB This plugin integrates MATLAB (R) with Jenkins and provides Jenkins interface to run MATLAB and Simulink (R) tests.	1.1.2



Jenkins中的测试报告

- 查看测试结果
- 查看代码覆盖率
- 查看测试报告

The screenshot displays the Jenkins Test Result page for a project named "Project Mass-Spring-Damper". It includes several sections:

- Test Result:** Shows "1 failures (+1)" with a progress bar. Summary statistics: 3 tests (=0), Took 1.6 sec, and an "add description" link.
- All Failed Tests:** Lists a failed test "designTest.testOvershoot" with a "Stack Trace" showing a "Verification failed" error. It includes framework diagnostic details like "verifyLessThan failed" and "Actual Value: 0.0829432821789317".
- File Coverage summary:** A table showing coverage for "simulateSystem.m":

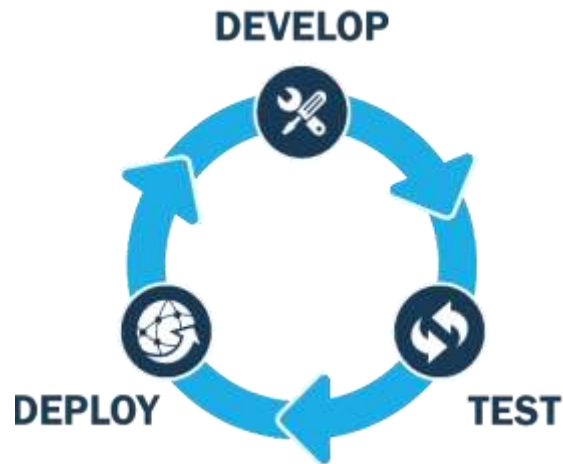
Name	Classes	Lines	Conditionals
simulateSystem.m	100% 1/1	90% 9/10	100% 0/0
- Coverage Breakdown by Class:** Shows "simulateSystem" with 90% coverage.
- Source:** Displays the MATLAB code for "simulateSystem.m", with lines 11-13 highlighted in green to indicate coverage.
- Project Mass-Spring-Damper:** A sidebar with "Coverage Report", "Workspace", and "Recent Changes" links.
- Permalinks:** A list of build links such as "Last build (#89) 1 mo 24 days ago".
- TAP Tests:** A bar chart showing the count of TAP tests (Failed, Passed, Skipped, ToDo) across build numbers #87 to #93.
- Code Coverage:** A line chart showing coverage percentages for Packages, Files, Classes, Lines, and Conditionals across build numbers #87 to #93.
- Test Result Trend:** A bar chart showing the trend of test results (Failed, Passed) across build numbers #88 to #93.



总结

总结

- MATLAB和Simulink可以伴您从构思到产品的全过程
- 通过良好的软件和建模实践节省时间和精力
- Projects使您的协作开发工作流程更加安全



MATLAB and Simulink

are the **easiest** and
most **productive** environments
for **engineers** and **scientists**

