

MATLAB EXPO

 FRANCE

8 octobre 2024 | Paris

MATLAB, les bonnes pratiques *Master Class*

Pierre HAROUIMI, MathWorks

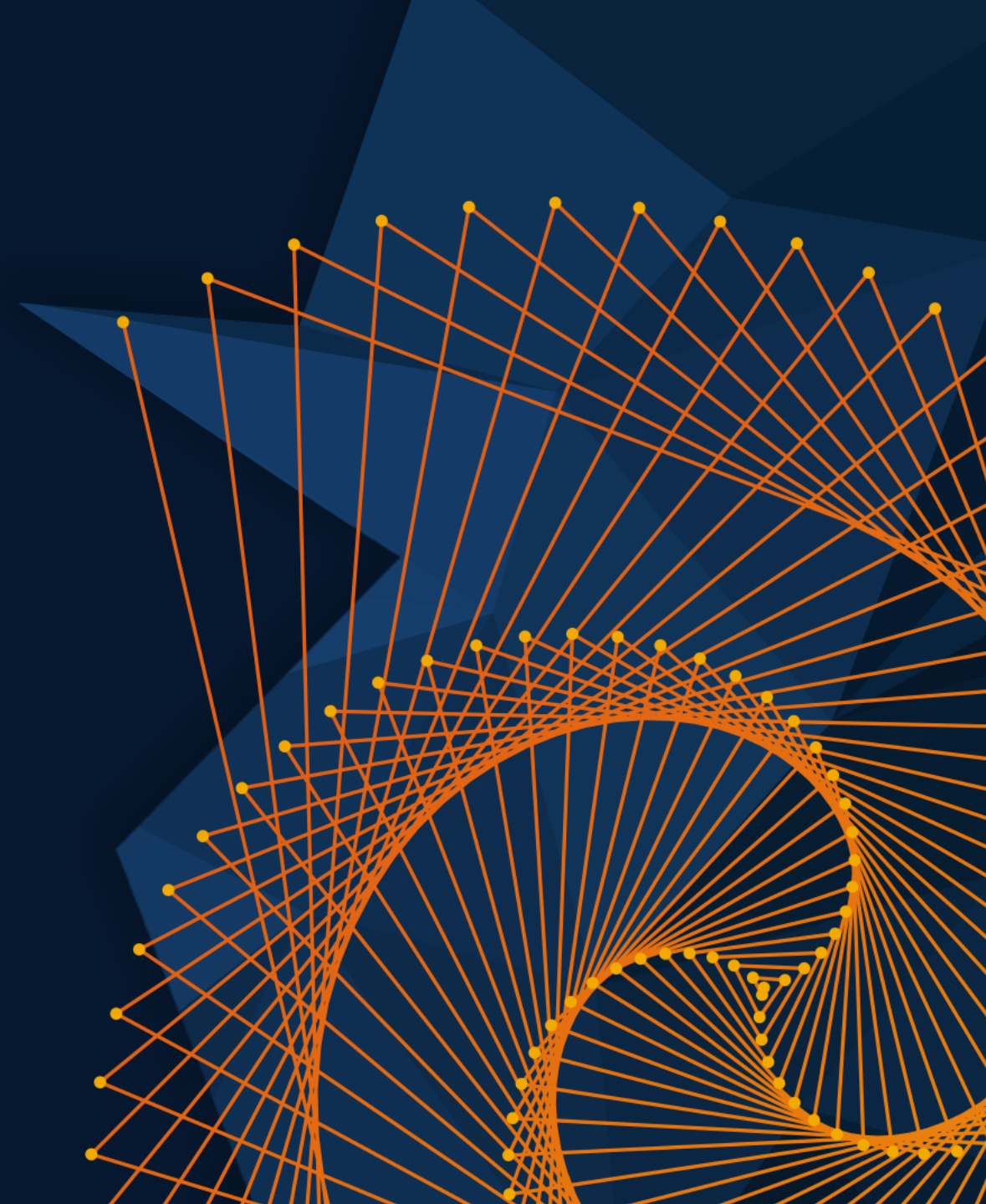


Application Engineer

Romain DUVAL, MathWorks



Application Engineer



Today's goal

Calculate the perceived temperature with the following very complex formula:

$$[T_{\text{perceived}} \approx T_{\text{air}} - 0.7 \times V^{0.5}]$$

wind

MATLAB code example

```
% Read Excel file - Trigger not recommended function
data = xlsread("weatherData.xlsx");

% Interpolate data - Trigger changed behavior
newStepTime = (data.Time(1):hours(1):data.Time(end))';
interpT = interp1(data.Time, data.Temperature, newStepTime, 'cubic');
interpW = interp1(data.Time, data.WindSpeed, newStepTime, 'cubic');
interpW(interpW<0) = 0;

% Trigger unused variable
i = 0;

% Calculate perceived temperature - Trigger preallocating
for i = 1 : numel(interpT)
    perceivedTemp(i,1) = interpT(i) - (0.7*(interpW(i)^0.5));
end

% Evaluate expression - Trigger custom rules
eval("plot(perceivedTemp)");
```


MATLAB code example

```
% Read Excel file - Trigger not recommended function
data = xlsread("weatherData.xlsx");

% Interpolate data - Trigger changed behavior
newStepTime = (data.Time(1):hours(1):data.Time(end))';
interpT = interp1(data.Time, data.Temperature, newStepTime, 'cubic');
interpW = interp1(data.Time, data.WindSpeed, newStepTime, 'cubic');
interpW(interpW<0) = 0;

% Trigger unused variable
i = 0;

% Calculate perceived temperature - Trigger preallocating
for i = 1 : numel(interpT)
    perceivedTemp(i,1) = interpT(i) - (0.7*(interpW(i)^0.5));
end

% Evaluate expression - Trigger custom rules
eval("plot(perceivedTemp)");
```

Info

Warning

Error

What are your main concerns?

« My code is very sensitive to changes »

« I don't know how to optimize my code »

« Each upgrade of a MATLAB release is very painful »

« It is hard to maintain a code with thousands of lines »

« Writing tests seems to be complicated »

Streamlining code development

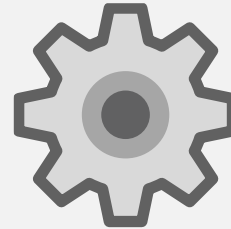
« It is hard to **maintain** a code with thousands of lines »

« Each **upgrade** of a MATLAB release is very painful »



Code quality
Code compatibility

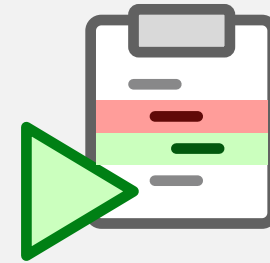
« I don't know how to **optimize** my code »



Code optimization
Code improvement

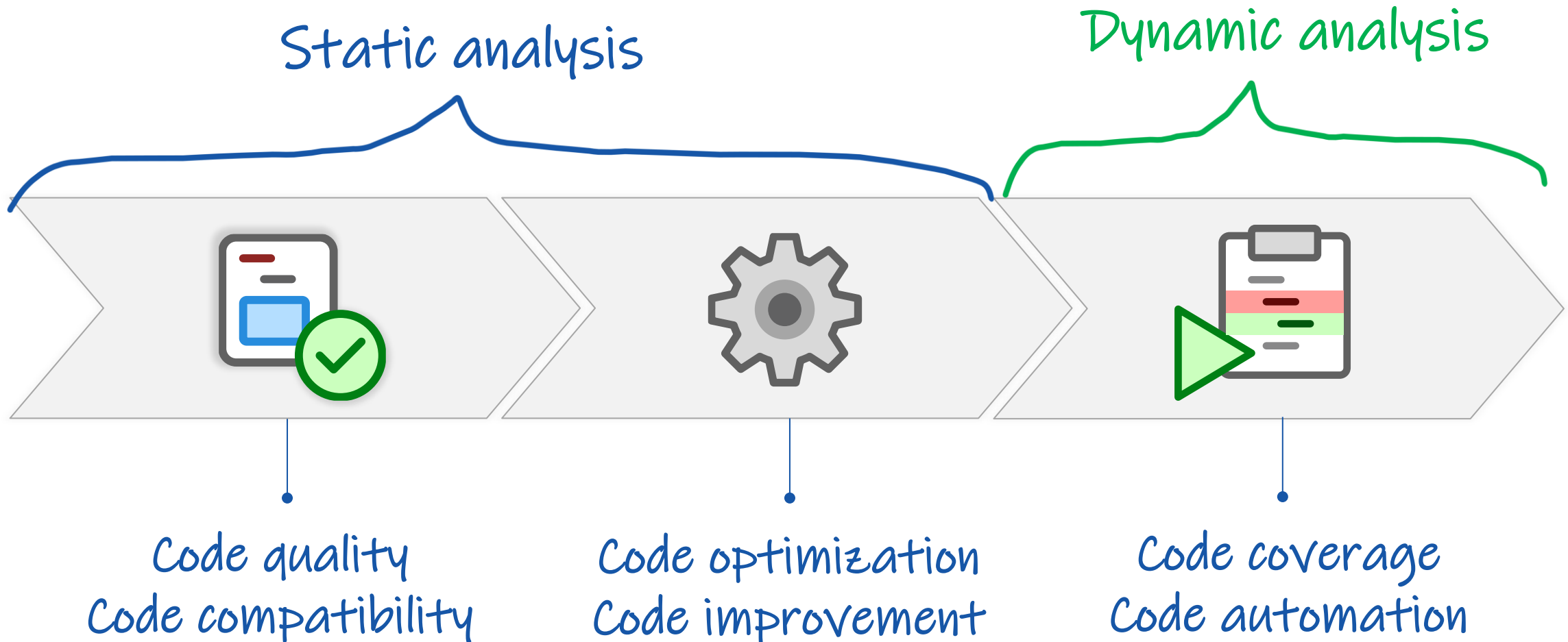
« My code is very sensitive to **changes** »

« Writing **tests** seems to be complicated »



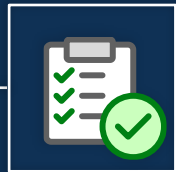
Code coverage
Code automation

Streamlining code development





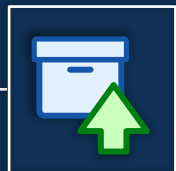
Code analysis



How to ensure code quality?



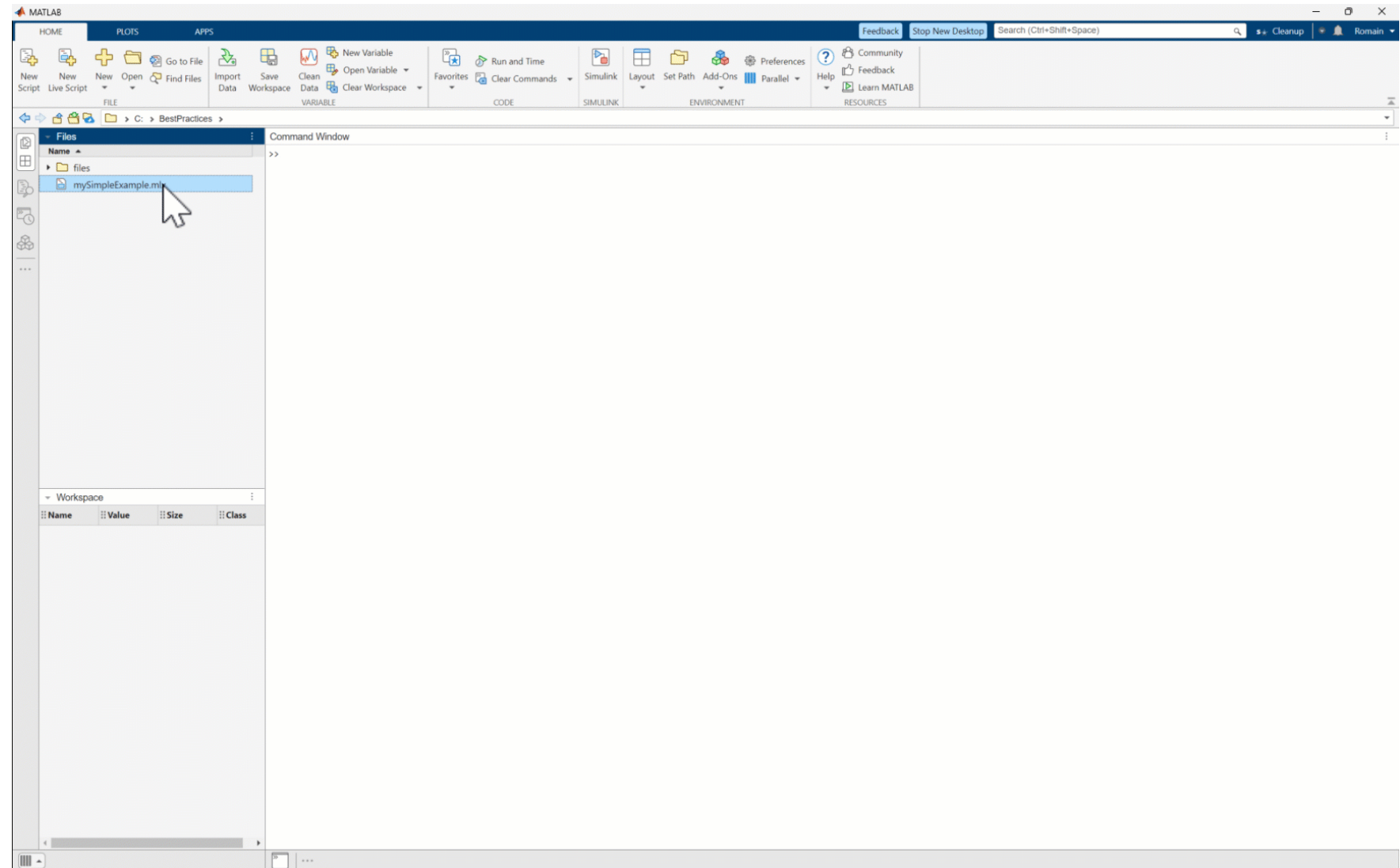
How to standardize code analysis?



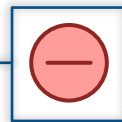
How to properly upgrade my code?

Static analysis with Code Analyzer

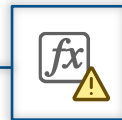
- * Proactive code checking
- * Code review accelerator
- * Code practice standardization
- * Reporting
- * Can be automated



Code **Compatibility** Analyzer: What does MATLAB check?



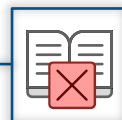
Functionalities that has been or will be removed



Functionalities that has changed or will change behavior



New functionalities that might improve code



Unsupported functionality that might cause errors

Code Compatibility Report



Code Compatibility Analyzer

Analysis Date: 9/24/2024, 5:46:53 PM

Active Issues Only ▼

MATLAB Release: R2024b

1

Total Files Analyzed

!

0

Error

!

2

Warning

i

1

Info

Code Health Details [expand all](#)

☰ Group by Issue Type ▼
🔍 Filter by Severity ▼
🔍 Filter by Issue Type ▼

Suggested Improvements (1)	Help
▶ i 'xlsread' is not recommended. With appropriate code changes, use 'readtable', 'readmatr...	?

Behavior Changes (2)	Help
▶ ! 'interp1(...,'cubic')' changed in R2020b to perform cubic convolution. To continue using s...	?

Code Compatibility Report

Code Compatibility Analyzer

Analysis Date: 9/24/2024, 5:46:53 PM

Active Issues Only

MATLAB Release: R2024b

1

Total Files Analyzed

0

Error

2

Warning

1

Info

Code Health Details [expand all](#)

Filter by Text Group by Issue Type Filter by Severity Filter by Issue Type

Suggested Improvements (1)	Help
<ul style="list-style-type: none"> Info 'xlsread' is not recommended. With appropriate code changes, use 'readtable', 'readmatr... 	?

Behavior Changes (2)

<ul style="list-style-type: none"> Warning 'interp1(...,'cubic')' changed in R2020b to perform cubic convolution. To co
--



(Overlap with Code Analyzer)

New message

Automatization



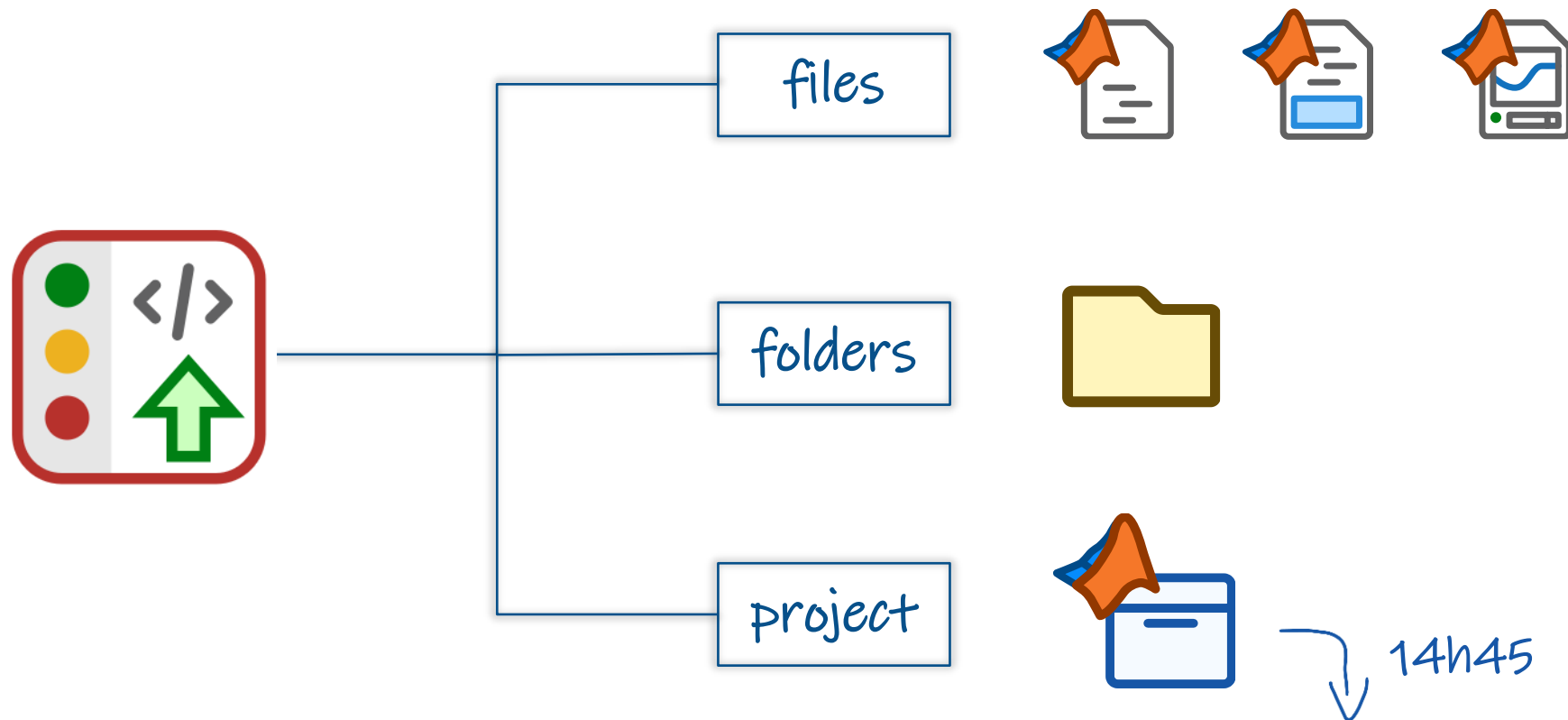
```
results = analyzeCodeCompatibility
```

```
results.Recommendations
```

```
ans = 3x8 table
```

	Identifier	Description
1	INTRPC	'''interp1(...,'cubic')' changed in R2020b to pe
2	INTRPC	'''interp1(...,'cubic')' changed in R2020b to pe
3	XLSRD	'''xlsread' is not recommended. With appropri

Code Compatibility for your projects





Master Class : Usine logicielle, industrialisez vos développements avec MATLAB et Simulink

Michelle Valente, *MathWorks*


Maxime François, *MathWorks*


Upgrade tool








UPGRADE

 Upgrade
UPGRADE

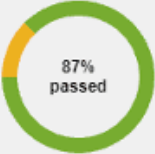
 Rerun Checks
SELECTED FILE

 View Changes
EXPORT

 Save Report
EXPORT

Files
All files ▼
All results ▼

- source
 - ⚠ timesTableGame.m
 - ✔ timestable.mlapp
- tests
- utilities



87%
passed

MATLAB Code	
Passed	7
Passed with fixes	0
Need attention	1

timesTableGame.m

✔ 1971
 i 0
 ⚠ 1

[source\timesTableGame.m](#)

Check Name	Result
'csvread' is not recommended. With appropriate code changes, use 'readtable' or 'readmatrix' inst...	⚠
'adobecset' has been removed. There is no simple replacement for this.	✔
'dbmp' will be removed in a future release. With appropriate code changes, use 'imwrite' instead.	✔
'dbmp16m' will be removed in a future release. With appropriate code changes, use 'imwrite' inst...	✔
'dbmp256' will be removed in a future release. With appropriate code changes, use 'imwrite' inste...	✔
'dbmpmono' will be removed in a future release. With appropriate code changes, use 'imwrite' ins...	✔
'dhdf' will be removed in a future release. With appropriate code changes, use 'imwrite' instead.	✔
'dllf' has been removed. Use Encapsulated PostScript instead.	✔
'dpbm' will be removed in a future release. With appropriate code changes, use 'imwrite' instead.	✔
'dpbmraw' will be removed in a future release. With appropriate code changes, use 'imwrite' inste...	✔
'dpcx16' will be removed in a future release. With appropriate code changes, use 'imwrite' instead.	✔
'dpcx24b' will be removed in a future release. With appropriate code changes, use 'imwrite' instead.	✔
'dpcx256' will be removed in a future release. With appropriate code changes, use 'imwrite' instead.	✔
'dpcxmono' will be removed in a future release. With appropriate code changes, use 'imwrite' inst...	✔
'dpgm' will be removed in a future release. With appropriate code changes, use 'imwrite' instead.	✔
'dpgmraw' will be removed in a future release. With appropriate code changes, use 'imwrite' inste...	✔
'dppm' will be removed in a future release. With appropriate code changes, use 'imwrite' instead.	✔

Showing 20 of 1972 results [Show All](#)

Details

'csvread' is not recommended. With appropriate code changes, use 'readtable' or 'readmatrix' instead.

[Learn more](#)

Warning

Incompatible code found in:

Line	Code
36:	M = csvread ('csvlist.dat',



Code optimization



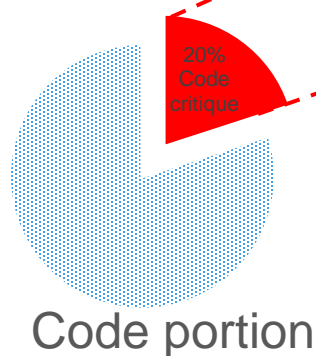
How to target performance critical code?



What are some of the optimization techniques?

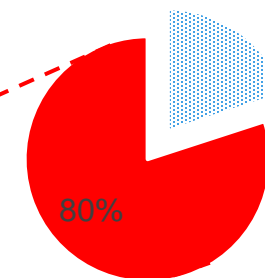
Empirical Pareto principle

- ⊘ Optimizing a complete software is easy
- ⊘ No need for performance-oriented development



- ⊘ Critical code already optimized
- ⊘ Critical code is scattered
- ⊘ Change impact all the code
- ⊘ Initial performance too poor

Execution time
Memory usage



Profiler MATLAB : "A graph is worth a thousand words."

- * Overview of execution by callstack
 - Code structure
 - Quickly identify bottleneck
- * Traceability
- * Programmatical approach

profile on;
 <function to benchmark>
 profile viewer;

The screenshot shows the MATLAB Profiler interface. The main window displays a code example for calculating perceived temperature. The code is as follows:

```

1 % Read Excel file - Trigger not recommended function
2 data = readtable("weatherData.xlsx");
3
4 % Interpolate data - Trigger changed behavior
5 newStepTime = (data.Time(1):hours(1):data.Time(end))';
6 interpT = interp1(data.Time, data.Temperature, newStepTime, 'cubic');
7 interpW = interp1(data.Time, data.WindSpeed, newStepTime, 'cubic');
8 interpW(interpW<0) = 0;
9
10 % Trigger unused variable
11 i = 0;
12
13 % Calculate perceived temperature - Trigger preallocating
14 for i = 1 : numel(interpT)
15     perceivedTemp(i,1) = interpT(i) - (0.7*(interpW(i)^0.5));
16 end
17
18 % Evaluate expression - Trigger custom rules
19 plot(perceivedTemp);
  
```

The Command Window at the bottom shows the following commands:

```

>> clear
>>
  
```

The Profiler interface also shows a file browser on the left with the file 'mySimpleExample.mlx' selected. The workspace is empty. The Command Window shows the current directory as 'C:\BestPractices'.

Benchmark technics

Benchmark should be performed with:

- Clean/ Defragged computer
- No concurrent running programs

The screenshot shows the MATLAB Live Editor interface. The main window displays a code example titled "MATLAB EXPO: Best Practices" with the subtitle "Code example for today". The code calculates the perceived temperature using a complex formula: $T_{\text{perceived}} \approx T_{\text{air}} - 0.7 \times V^{0.5}$. The code includes several comments highlighting performance triggers:

```

1 % Read Excel file - Trigger not recommended function
2 data = readtable("weatherData.xlsx");
3
4 % Interpolate data - Trigger changed behavior
5 newStepTime = (data.Time(1):hours(1):data.Time(end));
6 interpT = interp1(data.Time, data.Temperature, newStepTime, 'cubic');
7 interpW = interp1(data.Time, data.WindSpeed, newStepTime, 'cubic');
8 interpW(interpW<0) = 0;
9
10 % Trigger unused variable
11 i = 0;
12
13 % Calculate perceived temperature - Trigger preallocating
14 perceivedTemp = perceivedTemperature(interpT,interpW);
15
16 % Evaluate expression - Trigger custom rules
17 plot(perceivedTemp);

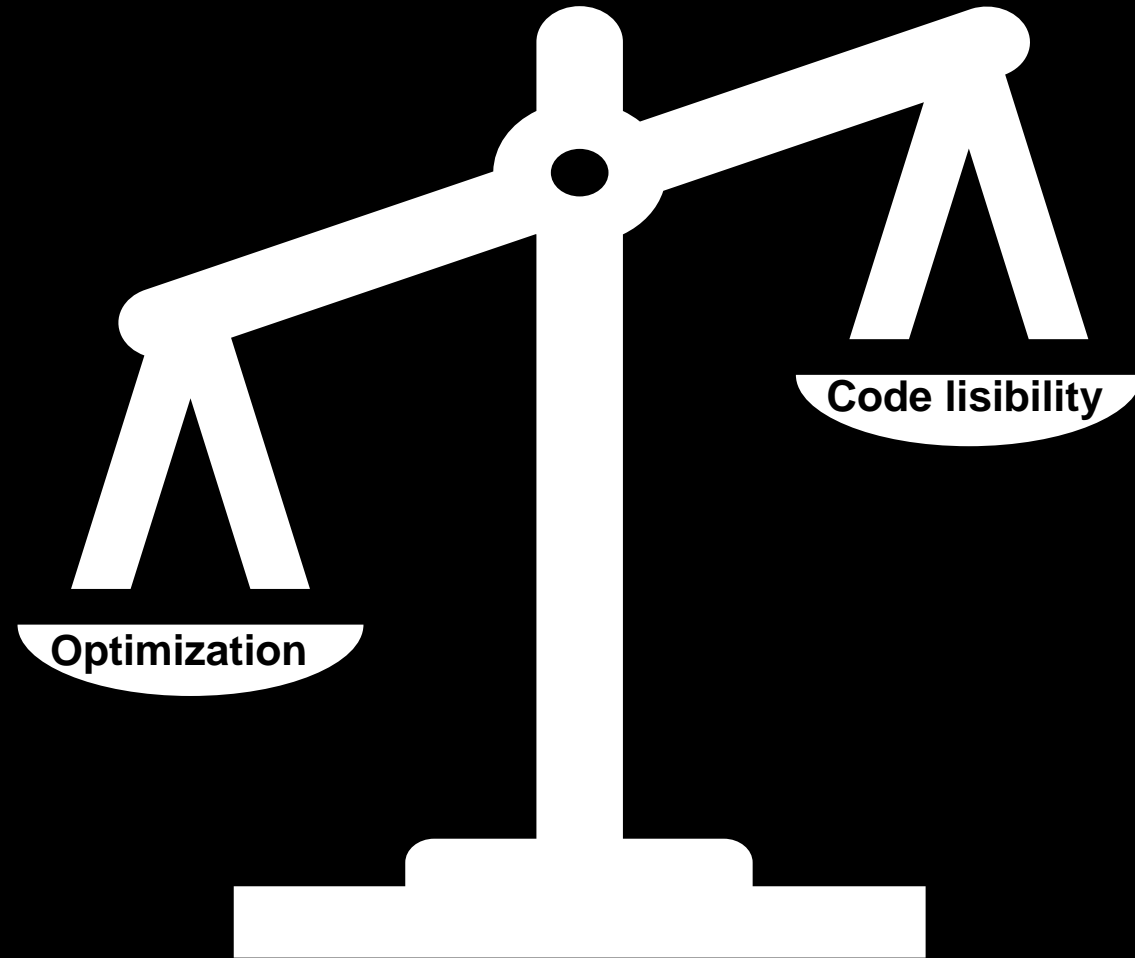
```

The Command Window at the bottom shows the prompt `>>`. The workspace window shows an empty table with columns for Name, Value, and Size.

Preliminary checklist

- ✓ Recent version of MATLAB
- ✓ Create functional code
- ✓ Minimize file I/O
- ✓ Reuse graphical component / libraries when possible
- ✓ Avoid displaying into Command Window
- ✓ Avoid Clear all, introspection, evaluation functions [dbstack | exist | whos | eval | feval(fname)]
- ✓ Do not use data as code

/!\ DISCLAIMER /!



DEVELOPER DISCRETION IS ADVISED

Préallocation

~~x~~ (1) zeros(3,
~~x~~ (2) ≡ 4
~~x~~ (3) ≡ 12
 x(3) = 12

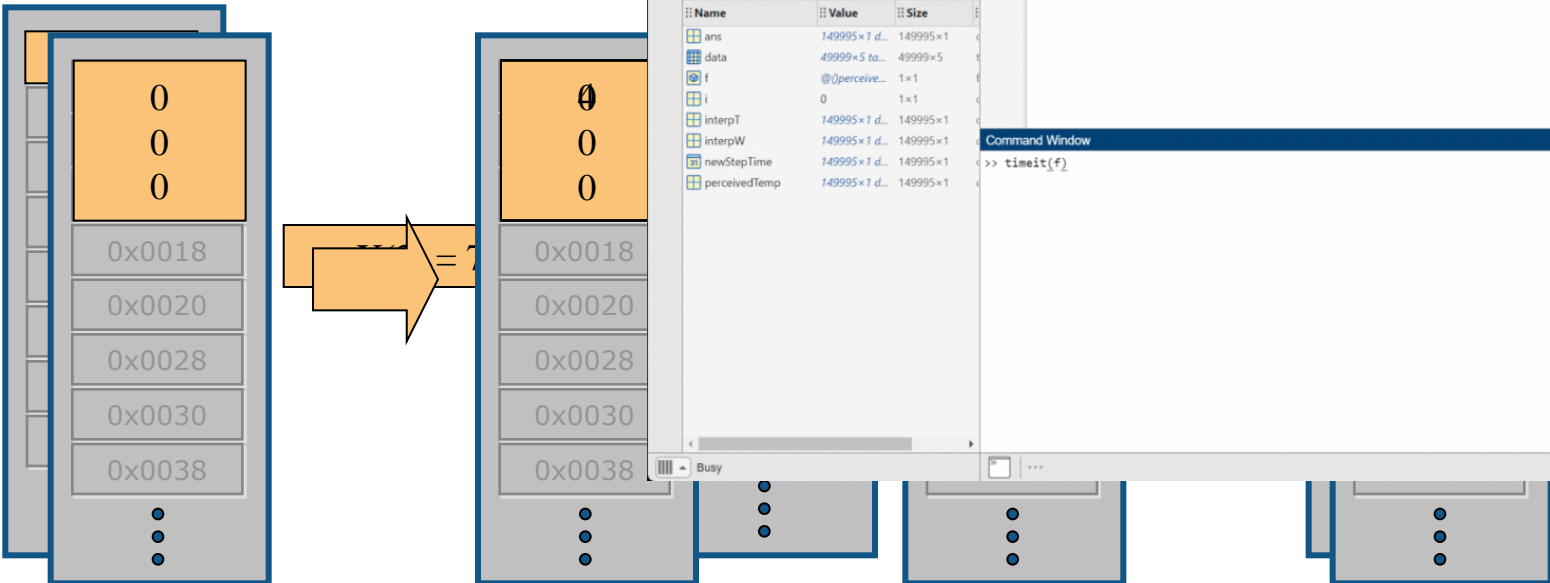
The screenshot shows the MATLAB IDE with the following content:

```

function perceivedTemp = perceivedTemperature(interpT,interpW)
for i = 1 : numel(interpT)
    perceivedTemp(i,1) = interpT(i) - (0.7*(interpW(i)^0.5));
end
end
  
```

Name	Value	Size
ans	149995 × 1 d...	149995 × 1
data	49999 × 5 ta...	49999 × 5
f	@(p)perceive...	1 × 1
i	0	1 × 1
interpT	149995 × 1 d...	149995 × 1
interpW	149995 × 1 d...	149995 × 1
newStepTime	149995 × 1 d...	149995 × 1
perceivedTemp	149995 × 1 d...	149995 × 1

Command Window: `>> timeit(f)`



Loop vs Vectorization



$$h_{\theta}(x^{(i)}) = \theta_0 x_0^{(i)} + \theta_1 x_1^{(i)} + \dots + \theta_j x_j^{(i)}$$



$$\theta^T x^{(i)} = [\theta_0 \theta_1 \dots \theta_n] \begin{bmatrix} x_1^{(i)} \\ x_1^{(i)} \\ \dots \\ x_1^{(i)} \end{bmatrix}$$



```

function perceivedTemp = perceivedTemperature(interpT,interpW)
2   perceivedTemp = zeros(size(interpT));
3   for i = 1 : numel(interpT)
4       perceivedTemp(i,1) = interpT(i) - (0.7*(interpW(i)^0.5));
5   end
6   end
  
```

Name	Value	Size
ans	149995×1 d...	149995×1
data	49999×5 ta...	49999×5
f	@perceive...	1×1
i	0	1×1
interpT	149995×1 d...	149995×1
interpW	149995×1 d...	149995×1
newStepTime	149995×1 d...	149995×1
perceivedTemp	149995×1 d...	149995×1

Arguments validation

The screenshot shows the MATLAB EXPO interface. The main window displays a code editor with the following content:

MATLAB EXPO: Best Practices

Code example for today

Calculate the perceived temperature with the following very complex formula: $[T_{\text{perceived}} \approx T_{\text{air}} - 0.7 \times V^{0.5}]$

Temperatures in Paris for the month of September 2024.

```

1 % Read Excel file - Trigger not recommended function
2 data = readtable("weatherData.xlsx");
3
4 % Interpolate data - Trigger changed behavior
5 newStepTime = (data.Time(1):hours(1):data.Time(end));
6 interpT = interp1(data.Time, data.Temperature, newStepTime, 'cubic');
7 interpW = interp1(data.Time, data.WindSpeed, newStepTime, 'cubic');
8 interpW(interpW<0) = 0;
9
10 % Trigger unused variable
11 i = 0;
12
13 % Calculate perceived temperature - Trigger preallocating
14 perceivedTemp = perceivedTemperature(interpT,interpW);
15
16 % Evaluate expression - Trigger custom rules
17 plot(perceivedTemp);

```

The Command Window at the bottom is empty, showing the prompt `>>`.

* What if the code is meant to be shared?

* Works for outputs too!

Parallelization : divide and conquer

- + Modern computers are multicores
- + OS support multithreading
- + Common tasks are time-consuming
 - .File IO
 - .Result display
 - .Lengthy calculation no vectorizable
 - ...
- + Tasks independent

asynchronous execution

The screenshot shows the MATLAB Live Editor interface. The main window displays a code example titled "MATLAB EXPO: Best Practices" with the following content:

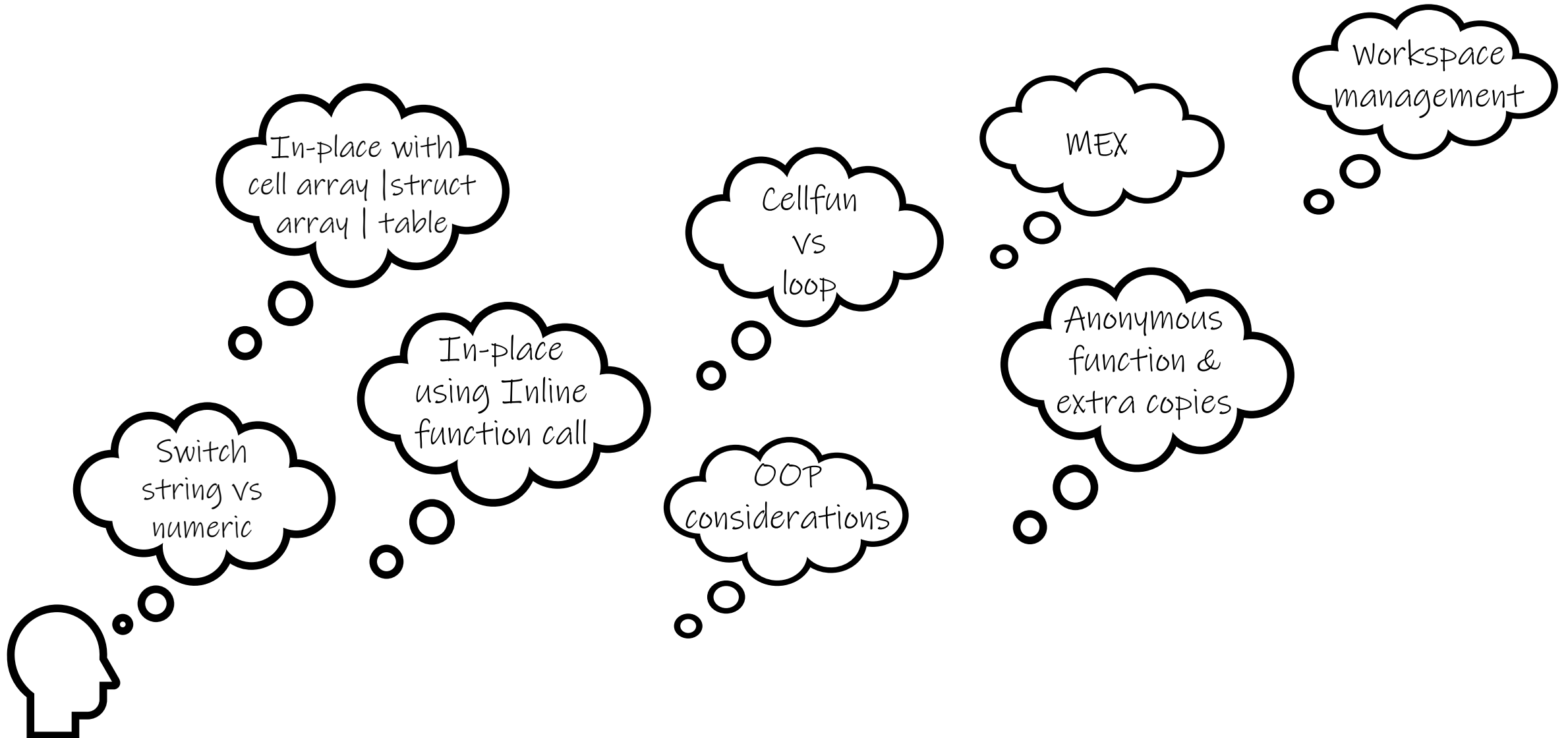
Code example for today
 Calculate the perceived temperature with the following very complex formula: $[T_{\text{perceived}} \approx T_{\text{air}} - 0.7 \times V^{0.5}]$
 Temperatures in Paris for the month of September 2024.

```

1 % Read Excel file - Trigger not recommended function
2
3 data = readtable("weatherData.xlsx");
4
5 % Interpolate data - Trigger changed behavior
6
7 newStepTime = (data.Time(1):hours(1):data.Time(end))';
8 interpT = interp1(data.Time, data.Temperature, newStepTime, 'cubic');
9 interpW = interp1(data.Time, data.WindSpeed, newStepTime, 'cubic');
10 interpW(interpW<0) = 0;
11
12 % Trigger unused variable
13 i = 0;
14
15 % Calculate perceived temperature - Trigger preallocating
16
17 perceivedTemp = perceivedTemperature(interpT,interpW);
18
19 % Evaluate expression - Trigger custom rules
20 plot(perceivedTemp);
  
```

The Command Window at the bottom shows the prompt `>>`.

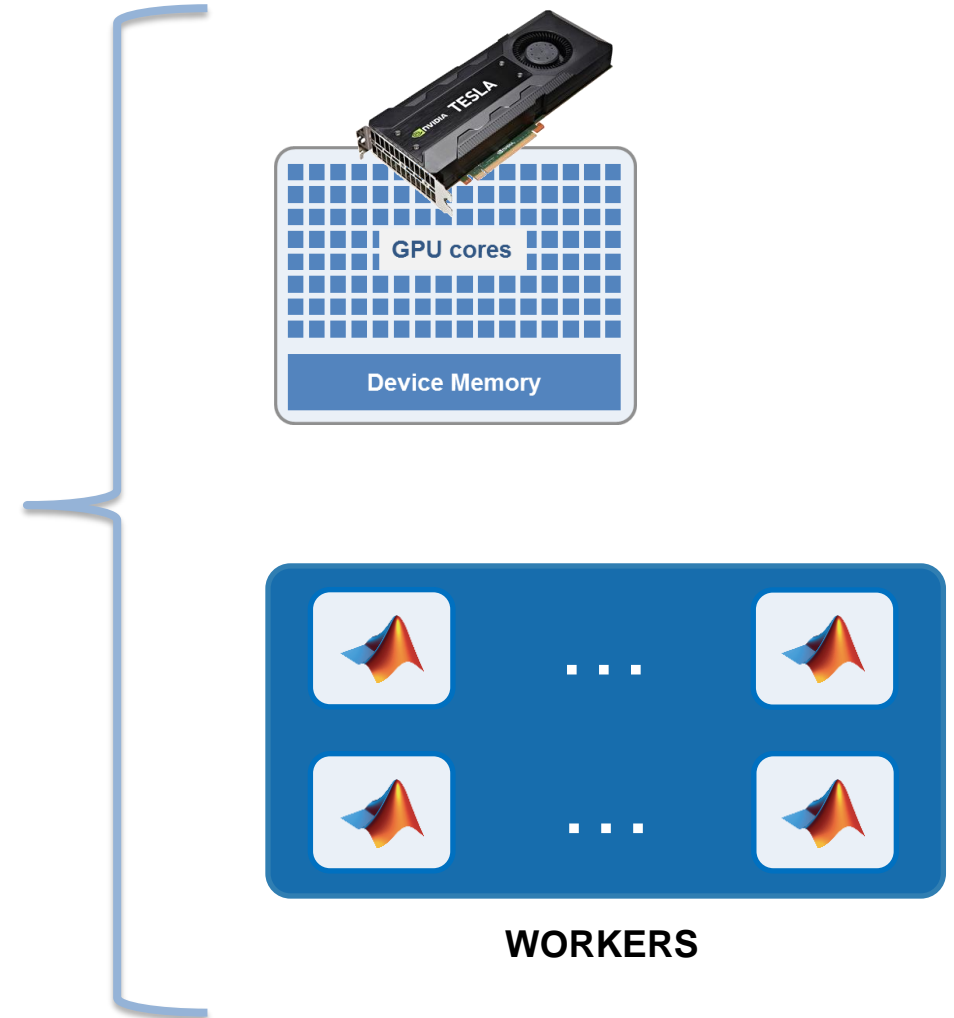
We could also have mentioned....



What else? ...Unleash the power of parallelism



MATLAB
Parallel Computing Toolbox

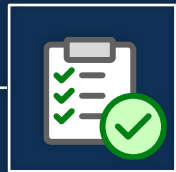




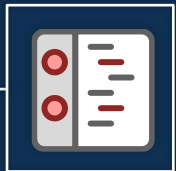
“Make my
code work,
Make my
code work”



Testing



How to continuously maintain my code?

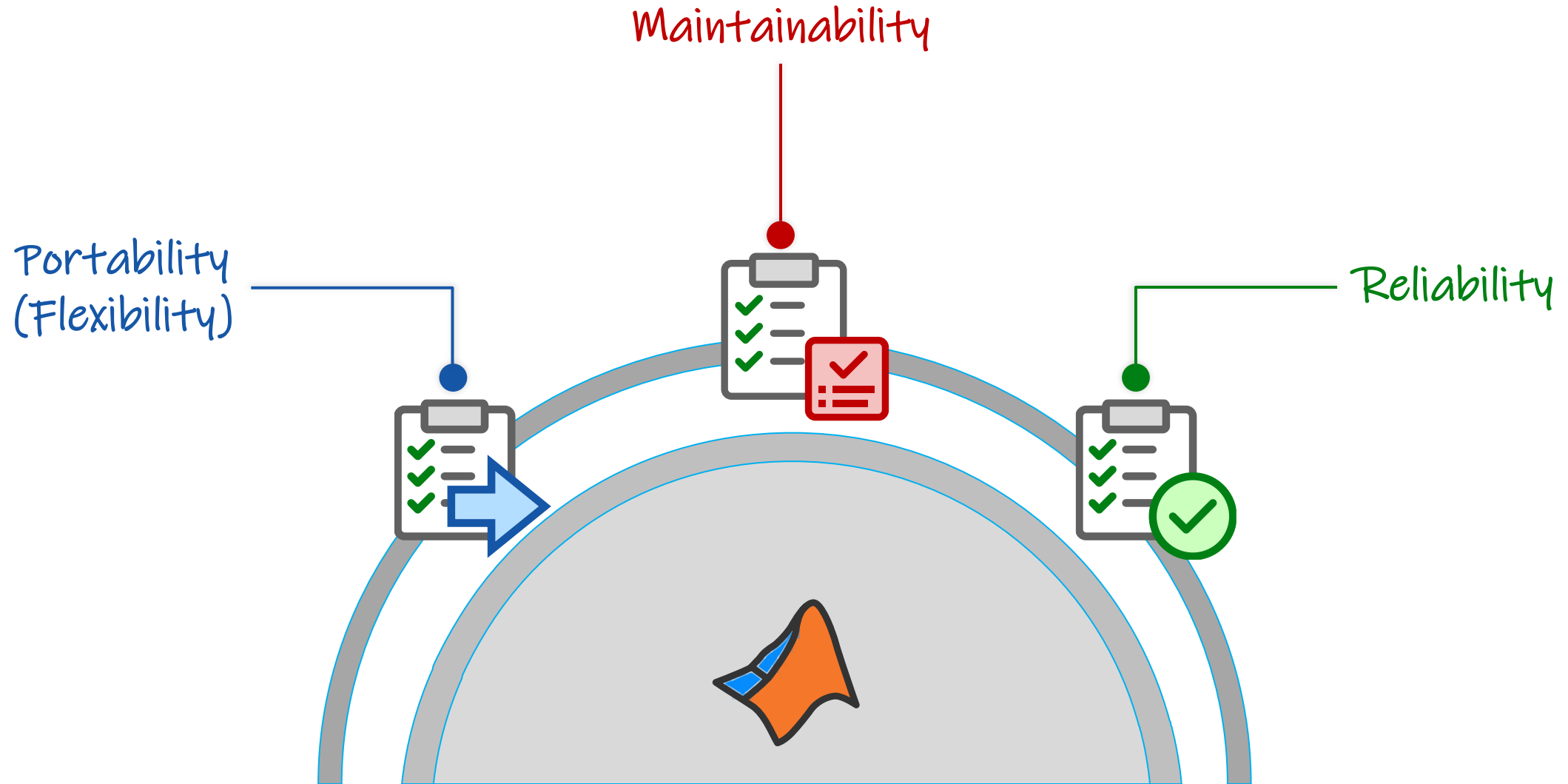


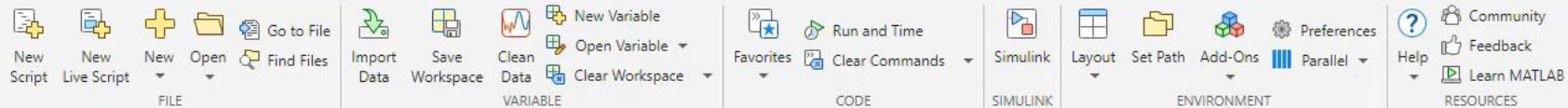
How to prevent breaking changes?



Is my code portable?

Software Product Quality: Maintainability, portability, reliability





OneDrive > ByProject > Events > MATLAB EXPO > MATLAB EXPO 2024 > Presentation_BestPractices >

perceivedTemperature.m x +

C:\Users\pharouim\OneDrive - MathWorks\ByProject\Events\MATLAB EXPO\MATLAB EXPO 2024\Presentation_BestPractices\functions\perceivedTemperature.m

```
1 function temp = perceivedTemperature(T,W)
2 arguments
3     T (1,:) double
4     W (1,:) {mustBeNonnegative}
5 end
6 temp = T - 0.7*W.^0.5;
7 end
```

Workspace

Name	Value
------	-------

Debugger

Breakpoints

- Pause on E...
- Pause on ...
- Pause on N...
- Pause on O...

Function call stack

Not paused in debugger

Command Window

>>

Equivalence testing

=> Test package deployment and code generation in 3 lines

Example with Python package deployment

```
% Equivalence test
classdef tDeployment < matlabtest.compiler.TestCase
    methods(Test)
        function pythonEquivalence(testCase)
            buildResults      = build(testCase,"perceivedTemperature.m","pythonPackage");
            executionResults = execute(testCase,buildResults,{[20 40]});
            verifyExecutionMatchesMATLAB(testCase,executionResults);
        end
    end
end
```

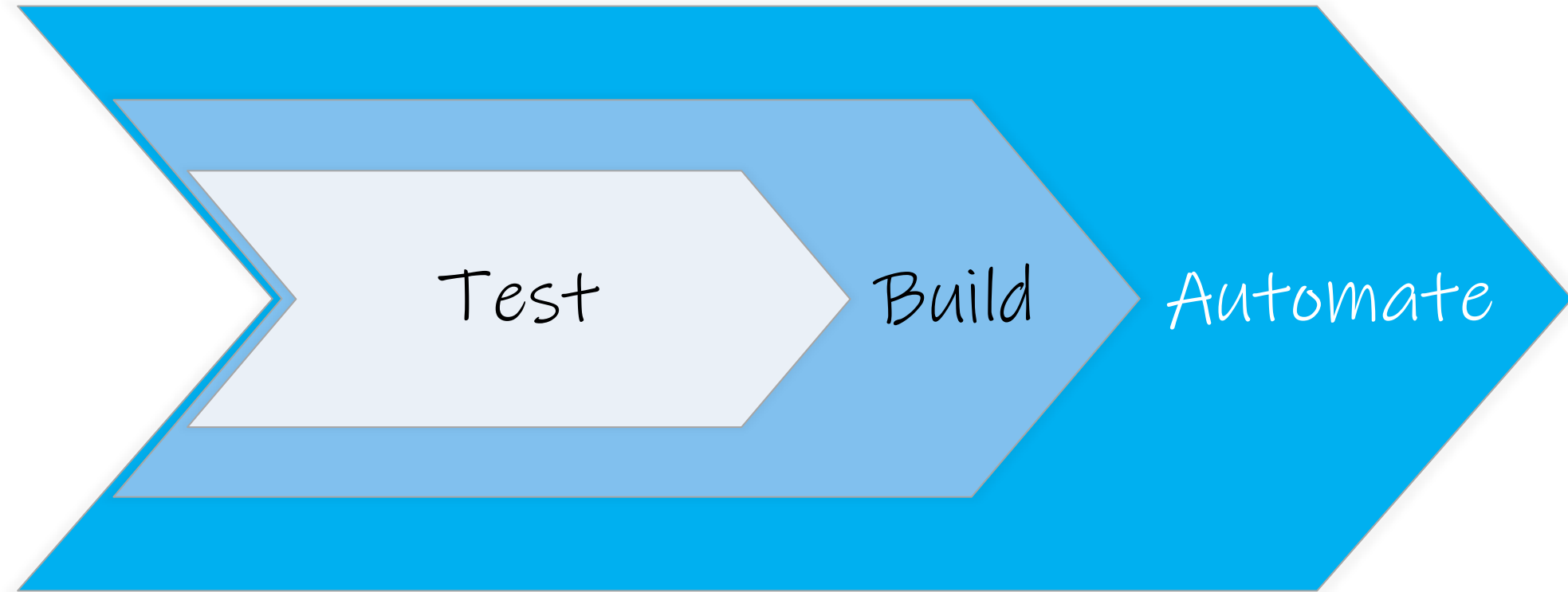


Build

Execute

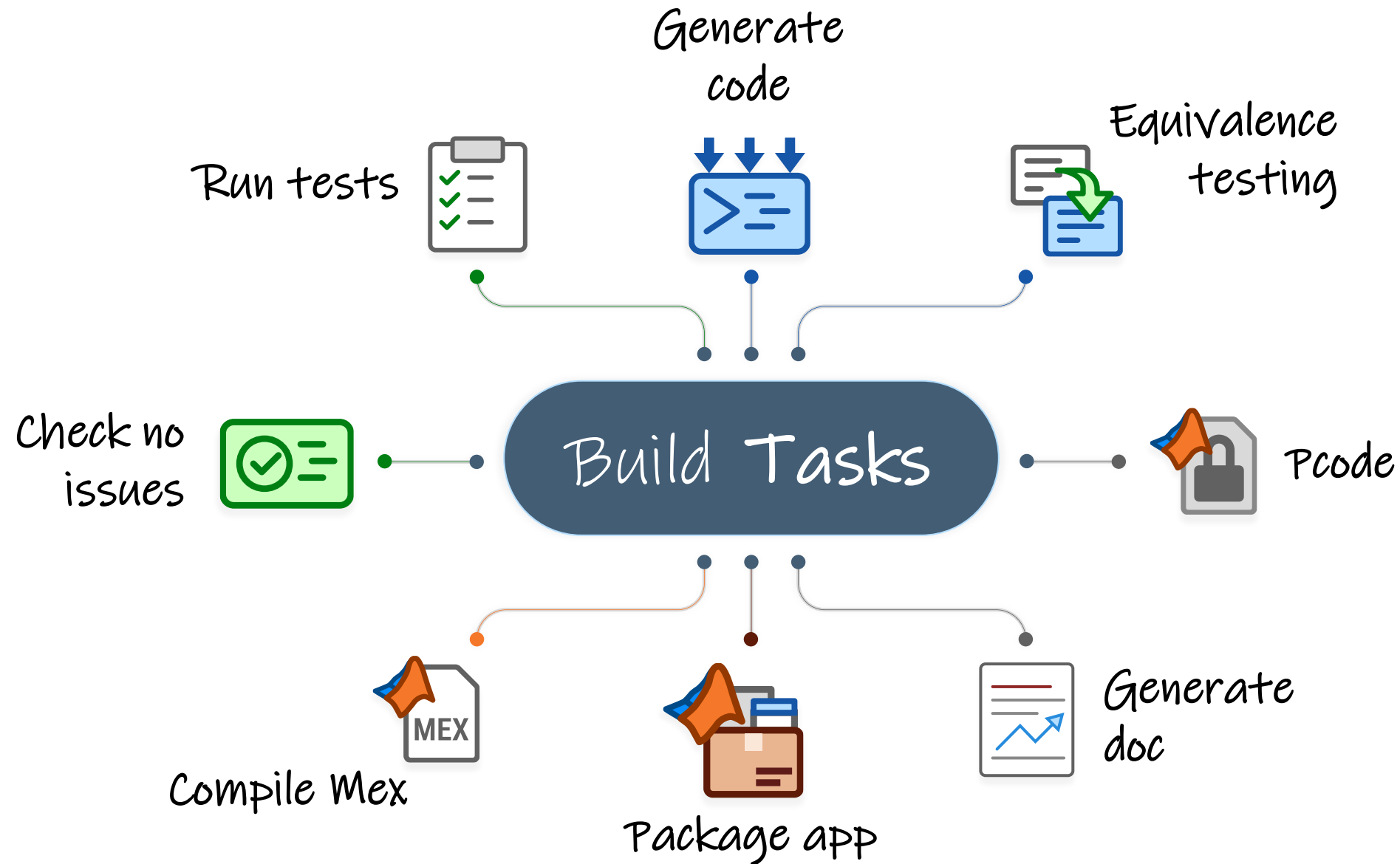
Verify

MATLAB's evolving solutions for test & automation



Increasing Levels of Automation

Build multiple tasks...



Build multiple tasks... and automate them

```
plan = buildplan;  
plan("check")      = CodeIssuesTask();  
plan("test")       = TestTask();  
plan("equivalence") = EquivalenceTask()
```

→ Define tasks

plan =

Plan with tasks:



check – Identify code issues



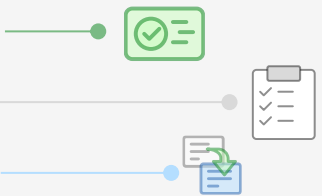
equivalence – Verify deployed code artifact and the MATLAB source code match



test – Run tests

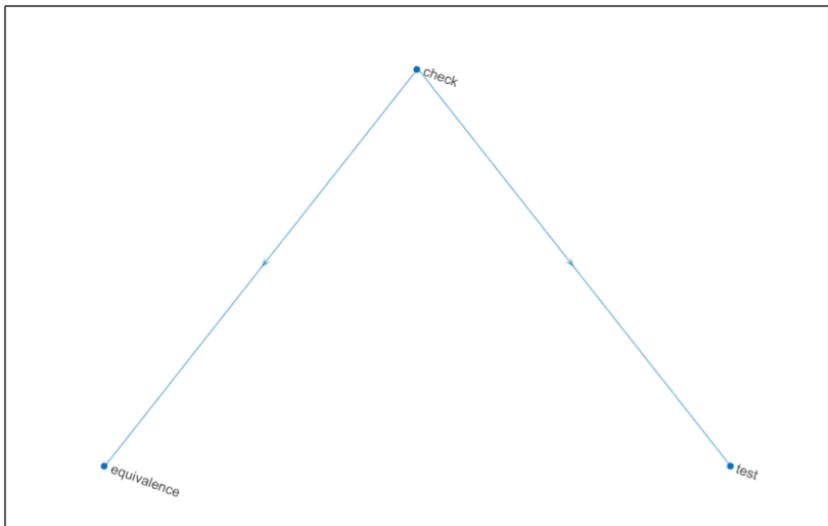
Build multiple tasks... and automate them

```
plan = buildplan;  
plan("check") = CodeIssuesTask();  
plan("test") = TestTask();  
plan("equivalence") = EquivalenceTask();
```



→ Define tasks

```
plan.DefaultTasks = "check";  
plan("check").Dependencies = ["test", "equivalence"];  
plot(plan)
```



→ Define dependencies

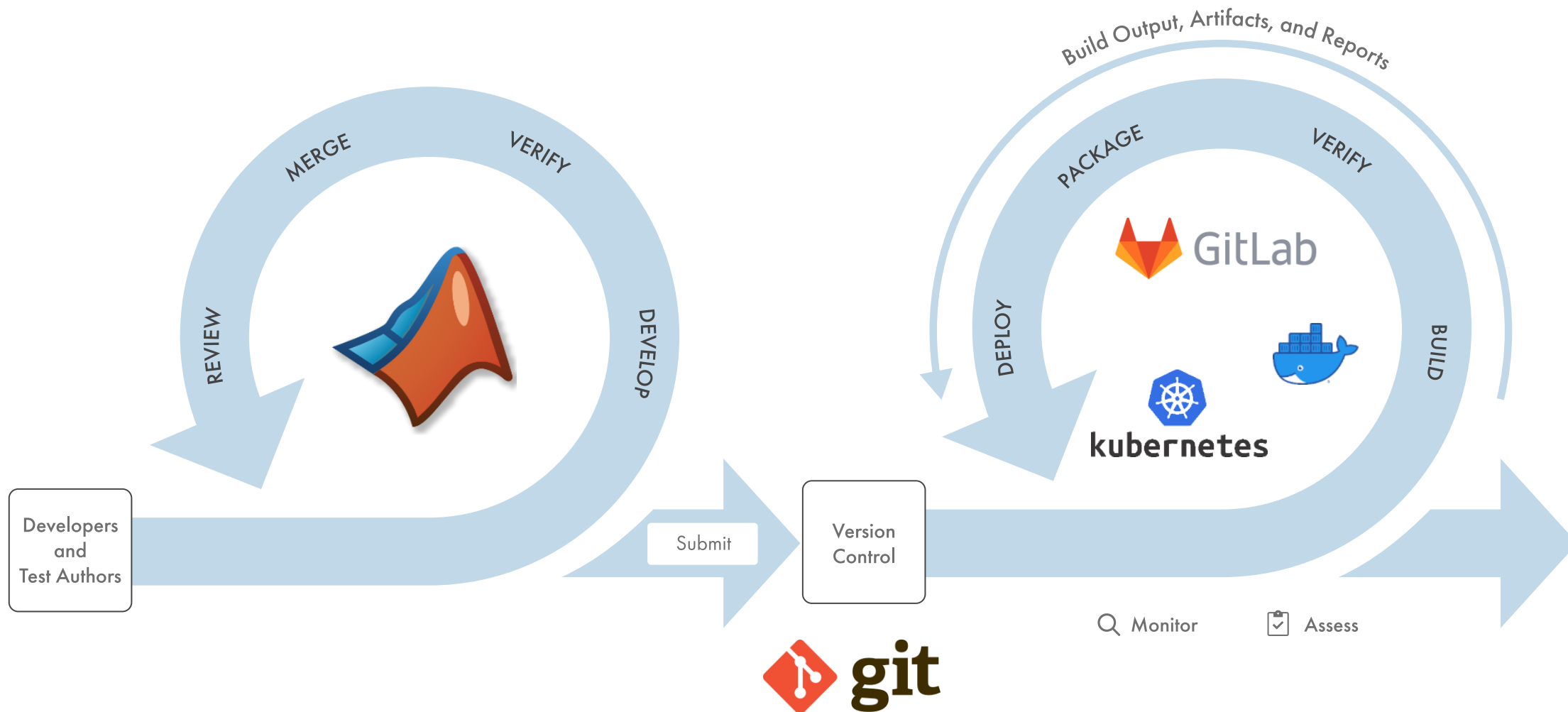
Build multiple tasks... and automate them

```
run(plan)
```

```
** Starting test  
Not possible  
.  
  
Test Summary:  
  Total Tests: 1  
    Passed: 1  
    Failed: 0  
  Incomplete: 0  
  Duration: 0.029041 seconds testing time.  
  
** Finished test  
  
** Done equivalence
```



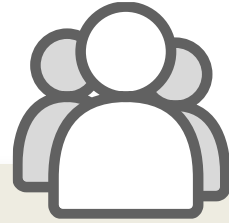
Automatize and industrialize your whole suite of tests in CI



Next Steps



**MathWorks
Engineers**



**MathWorks
Training**



**MathWorks
Consulting**

Reach out for support to implement these practices in your projects



Pierre Harouimi
pharouim@mathworks.com



Romain Duval
rduval@mathworks.com

MATLAB EXPO

 FRANCE

Thank you!



© 2024 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

