MATLAB EXPO 2017
Toolbox Development

David Sampson & Amy Koh
MathWorks Consulting
Good names

Basemap: 'colorterrain'
BubbleColorList: [8×3 double]
BubbleWidthRange: [5 20]
ColorData: [162×1 categorical]
ColorLegendTitle: 'Cause'
ColorVariable: 'Cause'
FontName: 'Helvetica'
FontSize: 10
GridVisible: 'on'
HandleVisibility: 'on'
LatitudeData: [162×1 double]
...
Bad names

- **Model**
- **foo_new**
- **calc**
- **ModelClass**
- **IModel**
- Indecipherable abbreviations
- Unconventional capitalisation
Too many files

code

resources

doc

demos

code

resources

doc

demos

fresh

stale

public

internal

MATLAB EXPO 2017
Organising your files

/layout
  /+uix
    /+internal
  /+uiextras
  /resources
/layoutdoc
  /examples
  /images

Credit: www.prettyneatliving.com
MATLAB Apps
A nearest-neighbor classifier that makes finely-detailed distinctions between classes, with the number of neighbors set to 1.
App building tools

GUIDE

MATLAB EXPO 2017
Model-view-controller architecture
Design tips

1. Review your APIs
2. Organise your files
3. Include some apps

Resources:
- “Best Practices – Adapt, then Adopt!” on Developer Zone
- MATLAB naming guidelines
- Model-View-Controller example code from previous MATLAB EXPO proceedings

Credit: www.flaticon.com
Design → Implement → Package → Distribute

Credit: www.flaticon.com
>> 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)
MATLAB EXPO 2017
Validate inputs

- `assert`
- `validateattributes`
- `isempty`, `isnan`, `isfinite` ...
- `narginchk`
- `inputParser`

```matlab
function y = timesTwo( x )

% Validate
validateattributes( x, {'double'}, {'size',[1 3], 'increasing'} );

y = x * 2;
end
```

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

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

- error and warning
- Use identifiers
- MException
- try/catch
- errordlg and warndlg
## Managing many changes: bugs, enhancements, tasks, ...

<table>
<thead>
<tr>
<th>Record</th>
<th>Type</th>
<th>Summary</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1003744</td>
<td>Task</td>
<td>Deprecate uixextras.TabPanel</td>
<td>Waiting for vendor</td>
</tr>
<tr>
<td>1019459</td>
<td>Enhancement</td>
<td>Improve handling of axes with illustrations</td>
<td>Waiting for vendor</td>
</tr>
<tr>
<td>1100294</td>
<td>Bug</td>
<td>Plotting to axes on hidden tab shows if visible tab has axes</td>
<td>Waiting for vendor</td>
</tr>
<tr>
<td>1129702</td>
<td>Enhancement</td>
<td>Improve handling of 3D-rotatable axes</td>
<td>Waiting for vendor</td>
</tr>
<tr>
<td>1155591</td>
<td>Enhancement</td>
<td>Add comments on HandleVisibility to GUI Layout Toolbox</td>
<td>Unreviewed</td>
</tr>
<tr>
<td>1214592</td>
<td>Task</td>
<td>Update GUI Layout Toolbox documentation style</td>
<td>To consider</td>
</tr>
<tr>
<td>1218686</td>
<td>Enhancement</td>
<td>Implement partial redraw for uix.BoxPanel</td>
<td>To consider</td>
</tr>
<tr>
<td>1338793</td>
<td>Enhancement</td>
<td>Refresh the documentation for deploying GUIs for Compiler etc.</td>
<td>Unreviewed</td>
</tr>
<tr>
<td>1344383</td>
<td>Enhancement</td>
<td>Support different tab widths for different title lengths</td>
<td>To consider</td>
</tr>
<tr>
<td>1400079</td>
<td>Enhancement</td>
<td>Provide a mechanism for deferring redraws during GUI creation</td>
<td>To consider</td>
</tr>
<tr>
<td>1422529</td>
<td>Enhancement</td>
<td>Add expandable, collapsible flex containers</td>
<td>To consider</td>
</tr>
<tr>
<td>1426213</td>
<td>Bug</td>
<td>Arrow up and down in documentation are not rendered properly.</td>
<td>Need more info</td>
</tr>
<tr>
<td>1497458</td>
<td>Enhancement</td>
<td>Improve test coverage for uix.ScrollingPanel</td>
<td>To consider</td>
</tr>
<tr>
<td>1499525</td>
<td>Bug</td>
<td>Using data cursors on axes inside GLT containers reorders contents</td>
<td>Investigating</td>
</tr>
<tr>
<td>1508772</td>
<td>Bug</td>
<td>HBoxFlex not spanning complete figure in Mac R2015b</td>
<td>Unreviewed</td>
</tr>
</tbody>
</table>
Test early, test often, test automatically

- Function and class-based tests
- Setup and teardown
- Parameterised tests
- Select and run tests
- Analyse test results

```matlab
classdef ExampleTest < matlab.unittest.TestCase
    methods(Test)
    function testA(testCase)
        testCase.verifyTrue(true)
    end
    function testB(testCase)
        testCaseassumeEqual(0,1)
        % remaining test code is not exercised
    end
    function testC(testCase)
        testCase.verifyFalse(true)
    end
end
```

Write unit tests in MATLAB

```matlab
classdef Rectangle < handle
    %RECTANGLE A simple rectangle class
    properties
        Height  double
        Width   double
    end
    methods
        function obj = Rectangle(height, width) ... %constructor
        function obj = resize(obj, height, width)
            %RESIZE Resize rectangle.
            obj.Height = height;
            obj.Width = width;
        end %resize
        end
end
```
Write unit tests in MATLAB

```matlab
classdef RectangleTest < matlab.unittest.TestCase
    %RECTANGLETEST A simple unit test.

    methods (Test)
        function testResize(testCase)

            % Create rectangle object
            r = Rectangle(2,3);

            % Resize
            r.resize(5,6);

            % Verify
            testCase.verifyEqual( r.Height, 5, ...
                'Height is incorrect.' );
            testCase.verifyEqual( r.Width, 6, ...
                'Width is incorrect.' );

        end
    end
end
```

```
>> tc = RectangleTest;
>> tc.run

Running RectangleTest.
Done RectangleTest.

ans =

TestResult with properties:

    Name: 'RectangleTest/testResize'
    Passed: 1
    Failed: 0
    Incomplete: 0
    Duration: 0.5770
    Details: [1×1 struct]

Totals:
1 Passed, 0 Failed, 0 Incomplete.
0.57699 seconds testing time.
```
Version control

- Giant ‘UNDO’ key
- Change management
- Software release management
- Collaborative development
- Git and Subversion Integration

Credit: http://www.chrismadden.co.uk/
Implementation tips

1. Catch errors early and provide actionable feedback
2. Write some unit tests
3. Use version control

Resources:
- MATLAB Testing framework
- MATLAB Source Control Integration
The bad old days

1. Unzip the zip file
2. Find the instructions and release notes
3. Decide whether you want the thing
4. Remove folders from old versions from the path
5. Add folders to the path
6. Save the path for next time
7. Find the documentation
8. Do work
MATLAB Toolbox

- Single toolbox installer (.mltbx)
- Toolbox metadata
- External dependencies
- Examples, apps, custom doc
- MATLAB path management
- Toolbox upgrade management
MATLAB Toolbox

- Toolbox metadata
- External dependencies
- MATLAB path management
- Examples, apps, custom doc
- Single toolbox installer (.mltbx)
- Toolbox upgrade management
Add-On Manager

MATLAB EXPO 2017
Where is the doc?

- **info.xml**
  - toolbox information and help location

- **helptoc.xml**
  - documentation files and structure

- **demos.xml**
  - custom examples
Don’t fall into these pitfalls

- Path issues
- Resources missing
- Add-on identifier ("GUID") reset
- Copy of dependencies included
Packaging tips

1. Package to .mltbx
2. Bundle documentation and examples with your code
3. Automate the packaging process

Resources:
- “Give Them A Care Package” on Developer Zone
- “Packaging Pitfalls” on Developer Zone
- MATLAB Toolbox Tools on File Exchange
Updating toolboxes

>> minimart.updateToolboxes( 'stores\huge' )
Upgrade 'Badminton Toolbox' from version 2.0 to version 2.4? (y/n) y
Upgrade 'Archery Toolbox' from version 1.1 to version 2.1? (y/n) y
Upgrade 'Bottle Toolbox' from version 1.4 to version 3.2? (y/n) y
Distribution tips

1. Corral your toolboxes
2. Help users browse and update
3. Start simple and scale up

Resources:
- MATLAB Minimart on File Exchange
- “Jaguar Land Rover Standardizes on MATLAB for Developing, Packaging, and Sharing Engineering Tools” on User Stories

Credit: www.flaticon.com
Thank you.

Questions?