



SMALLTALK-DRIVEN GUI TESTING

By Andreas Hiltner





```
23
24
25     var requestUrl = app.baseUrl + "/w/api.php";
26     $.ajax({
27       type: 'GET',
28       url: requestUrl,
29       data: {
30         action: 'opensearch',
31         search: term,
32         format: 'json'
33       },
34       success: function(data) {
35         renderResults(data);
36       }
37     });
38   } else {
39     chrome.showNoConnectionMessage();
40     chrome.showContent();
41   }
42 }
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
```

```
<script type="text/html" id="search-results-template">
  {{#pages}}
  <div class="listItem" data-page-url="{{key}}" data-page-title="{{value}}">
    <a class="listItem searchItem">
      <span class="iconSearchResult">
        <span class="text deletedResult">{{title}}</span>
      </span>
    </a>
  </div>
  {{/pages}}
  {{#pages}}
  <div class="listItem" data-page-url="{{key}}" data-page-title="{{value}}">
    <a class="listItem searchItem">
      <span class="iconSearchResult">
        <span class="text deletedResult">{{title}}</span>
      </span>
    </a>
  </div>
  {{/pages}}
  <div class="listItemContainer classSearch">
```

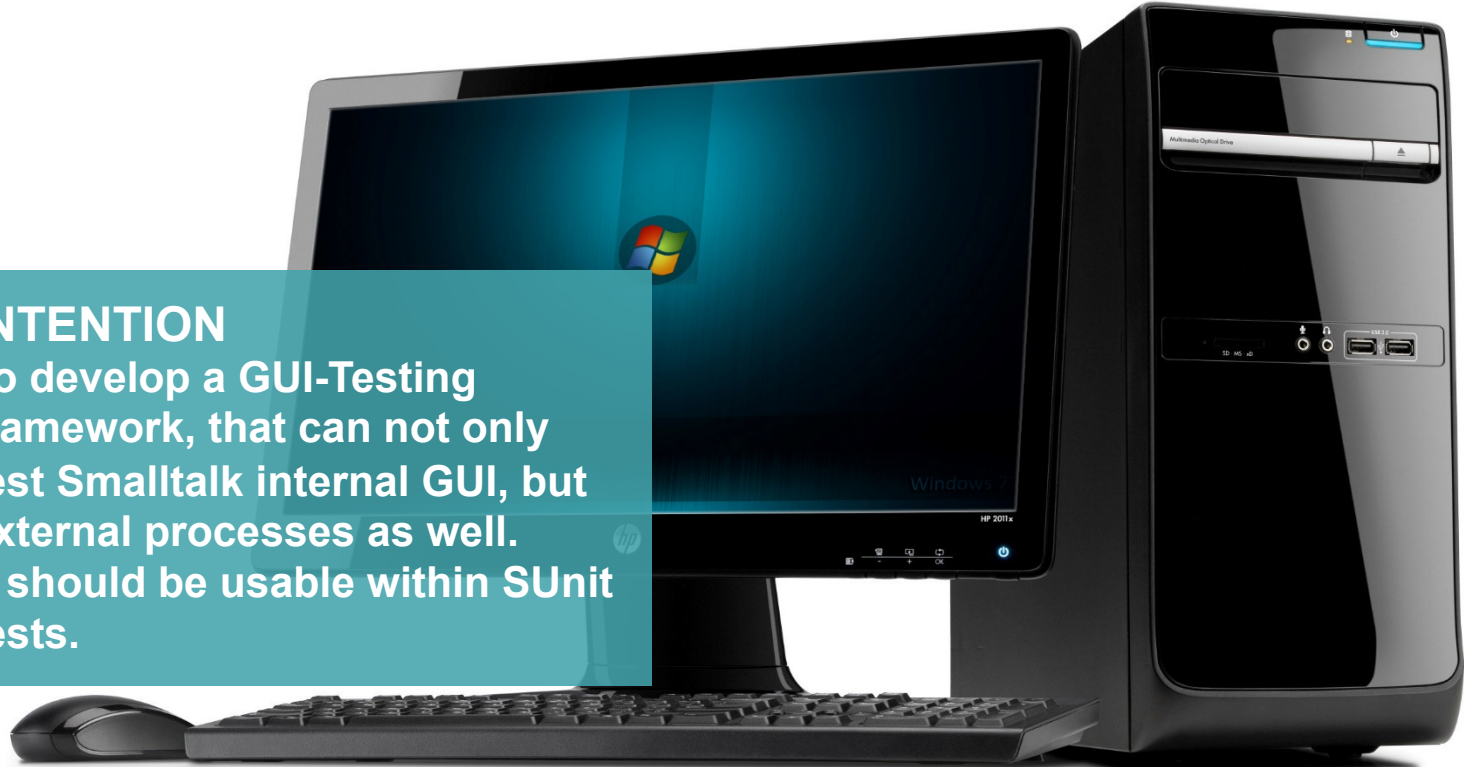
MacBook Pro

Overview

- **Intention**
- **Record and Replay Events**
- **Script Classes**
 - Screenshot
 - SendInput
 - OSWindow
- **Script Example**
- **Questions**

INTENTION

To develop a GUI-Testing framework, that can not only test Smalltalk internal GUI, but external processes as well. It should be usable within SUnit tests.



Record and Replay Events

Record all events (keyboard, mouse) with timing.
Events can be saved to a file and replayed later.

Beware of applications, that save the state in
between, e.g. Calculator

Record and Replay Events

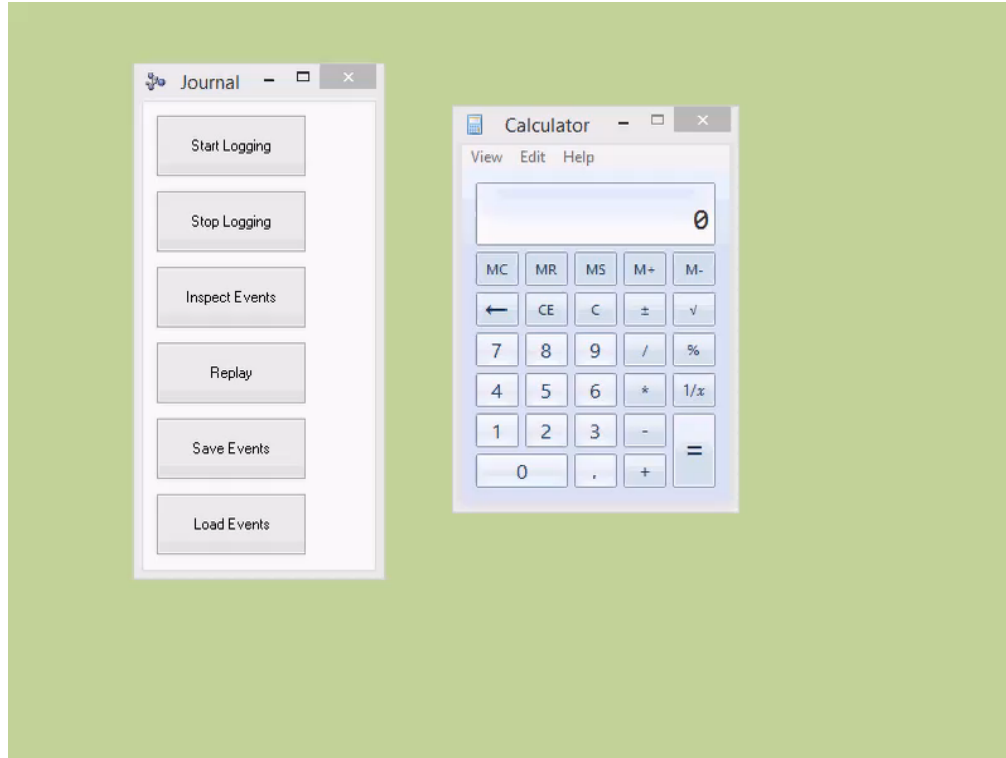
Advantage:

- What-You-See-Is-What-You-Get

Drawback:

- Uses absolute coordinates
- Recording has to be redone, if the test changes even the slightest bit

Record and Replay Events



Script

- Windows
- Mouse Input
- Keyboard Input
- Screenshot

Script - Window

- Retrieve Window with Title
- Wait for Window with Title
- Retrieve Window at a certain location
- Get the (Windows) menu
- Send a menu item event
- Retrieve Titlebar-Information
(location of Minimize/Maximize/Close button)

Script – Mouse Input

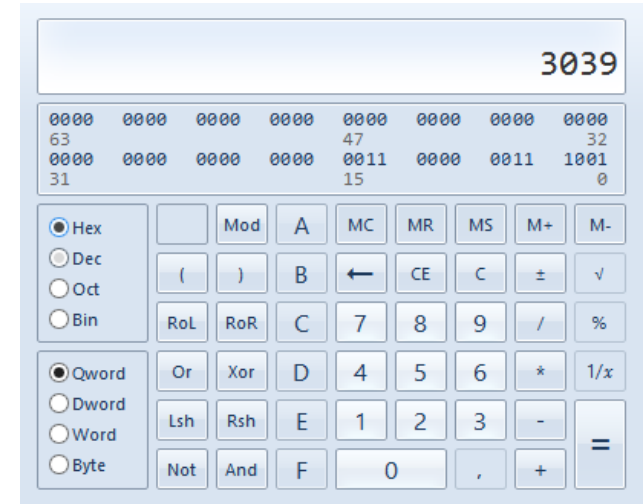
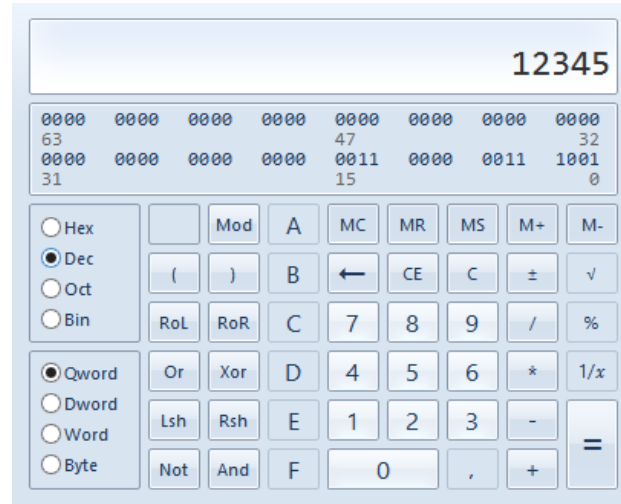
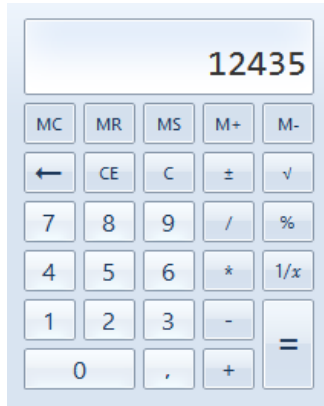
- Click left/middle/right button
- Scroll Mouse wheel up/down
- Move cursor
- Hover

Script – Keyboard Input

- Send String or individual Characters
- Send Shortcut (e.g. CTRL+C, ALT+1)
- Send Keys like LeftArrow, RightArrow, PageUp, PageDown, Return, etc.

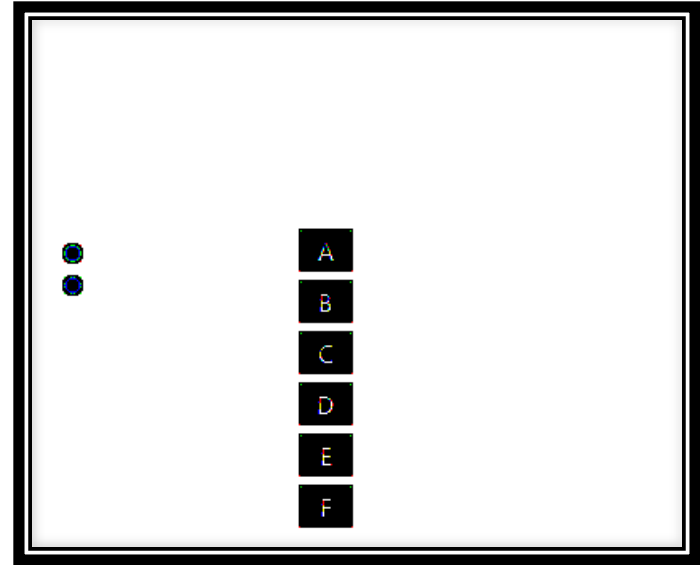
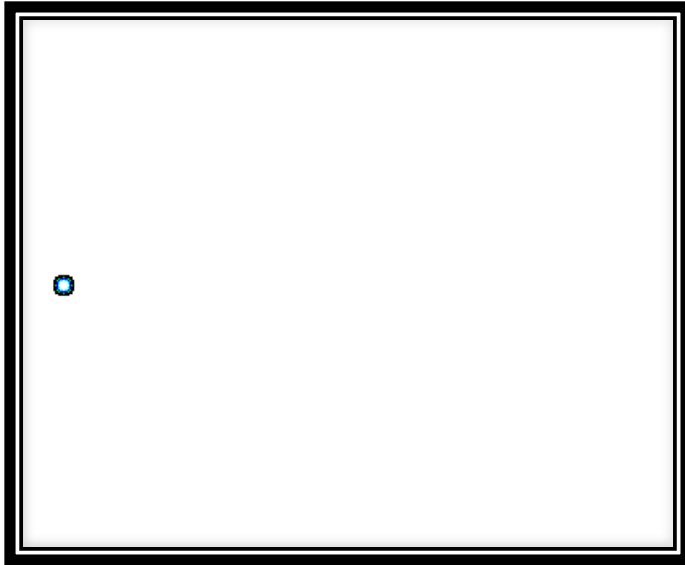
Script – Class Screenshot

- Take screenshot with / without window frame
- Save screenshot as Bitmap / PNG / JPEG

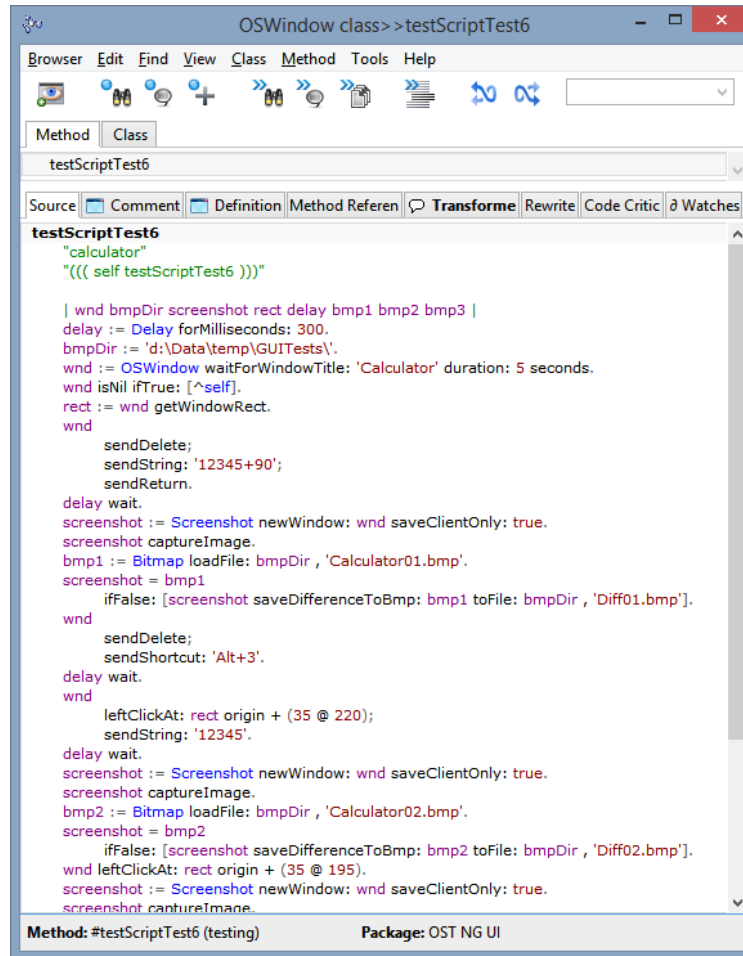


Script – Class Screenshot (con't)

- Compare Screenshot with existing Bitmap
- Save differences to a Bitmap in a file



Script



The screenshot shows an IDE window titled "OSWindow class>>testScriptTest6". The window contains a script for testing the OSWindow class. The script is written in a language that uses objects and methods like Delay, Screenshot, Bitmap, and OSWindow. It performs several actions: capturing screenshots of a calculator window, comparing them to find differences, and clicking on the calculator window.

```
OSWindow class>>testScriptTest6
Browser Edit Find View Class Method Tools Help
Method Class
testScriptTest6
Source Comment Definition Method Referen Transforme Rewrite Code Critic Watches
testScriptTest6
"calculator"
"((( self testScriptTest6 )))"

| wnd bmpDir screenshot rect delay bmp1 bmp2 bmp3 |
delay := Delay forMilliseconds: 300.
bmpDir := 'd:\Data\temp\GUI\Tests\'.
wnd := OSWindow waitForWindowTitle: 'Calculator' duration: 5 seconds.
wnd isNil ifTrue: [^self].
rect := wnd getWindowRect.
wnd
    sendDelete;
    sendString: '12345+90';
    sendReturn.
delay wait.
screenshot := Screenshot newWindow: wnd saveClientOnly: true.
screenshot captureImage.
bmp1 := Bitmap loadFile: bmpDir, 'Calculator01.bmp'.
screenshot = bmp1
    ifFalse: [screenshot saveDifferenceToBmp: bmp1 toFile: bmpDir, 'Diff01.bmp'].
wnd
    sendDelete;
    sendShortcut: 'Alt+3'.
delay wait.
wnd
    leftClickAt: rect origin + (35 @ 220);
    sendString: '12345'.
delay wait.
screenshot := Screenshot newWindow: wnd saveClientOnly: true.
screenshot captureImage.
bmp2 := Bitmap loadFile: bmpDir, 'Calculator02.bmp'.
screenshot = bmp2
    ifFalse: [screenshot saveDifferenceToBmp: bmp2 toFile: bmpDir, 'Diff02.bmp'].
wnd leftClickAt: rect origin + (35 @ 195).
screenshot := Screenshot newWindow: wnd saveClientOnly: true.
screenshot captureImage.
```

Method: #testScriptTest6 (testing) Package: OST NG UI

Script – Part One

```
wnd := OSWindow waitForWindowTitle: 'Calculator' duration: 5 seconds.  
wnd isNil ifTrue: [^self].  
rect := wnd getWindowRect.  
wnd  
    sendDelete;  
    sendString: '12345+90';  
    sendReturn.  
delay wait.  
screenshot := Screenshot newWindow: wnd saveClientOnly: true.  
screenshot captureImage.  
bmp1 := Bitmap loadFile: bmpDir , 'Calculator01.bmp'.  
screenshot = bmp1  
    ifFalse: [screenshot saveDifferenceToBmp: bmp1 toFile: bmpDir , 'Diff01.bmp'].
```

Script – Part Two

```
wnd
    sendDelete;
    sendShortcut: 'Alt+3'.
delay wait.
wnd
    leftClickAt: rect origin + (35 @ 220);
    sendString: '12345'.
delay wait.
screenshot := Screenshot newWindow: wnd saveClientOnly: true.
screenshot captureImage.
bmp2 := Bitmap loadFile: bmpDir , 'Calculator02.bmp'.
screenshot = bmp2
    ifFalse: [screenshot saveDifferenceToBmp: bmp2 toFile: bmpDir , 'Diff02.bmp'].
```

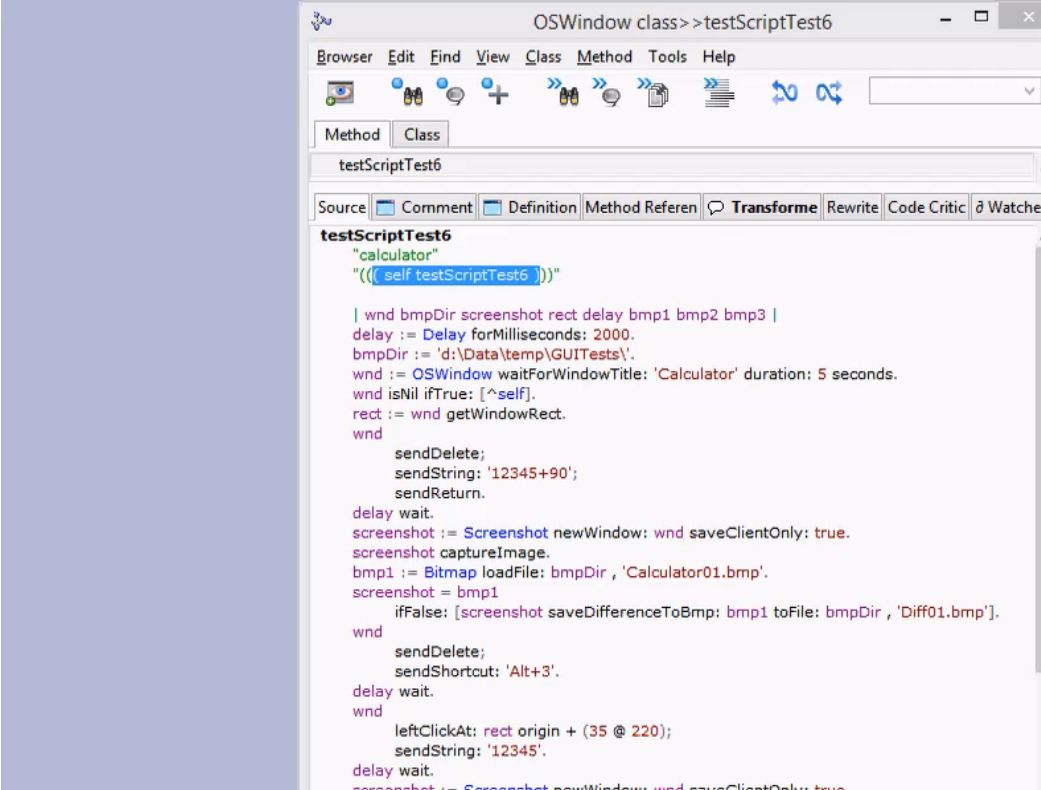

Script – Part Three

```
wnd leftClickAt: rect origin + (35 @ 195).  
delay wait.  
screenshot := Screenshot newWindow: wnd saveClientOnly: true.  
screenshot captureImage.  
bmp3 := Bitmap loadFile: bmpDir , 'Calculator03.bmp'.  
screenshot = bmp3  
    ifFalse: [screenshot saveDifferenceToBmp: bmp3 toFile: bmpDir , 'Diff03.bmp'].
```

Script – Part Four

```
wnd
    sendDelete;
    leftClickAt: rect origin + (35 @ 220);
    sendShortcut: 'Alt+1'.
delay wait.
wnd
    sendDelete;
    sendString: '3+4'.
delay wait.
wnd sendString: '='.
delay wait.
wnd
    sendDelete;
    leftClickAt: wnd getTitleBarInfo rectCloseButton center
```

Script – Video



```
OSWindow class>>testScriptTest6
Browser Edit Find View Class Method Tools Help
Method Class
testScriptTest6
Source Comment Definition Method Referen Transforme Rewrite Code Critic Watches
testScriptTest6
"calculator"
"((( self testScriptTest6 )))"

| wnd bmpDir screenshot rect delay bmp1 bmp2 bmp3 |
delay := Delay forMilliseconds: 2000.
bmpDir := 'd:\Data\temp\GUITests\'
wnd := OSWindow waitForWindowTitle: 'Calculator' duration: 5 seconds.
wnd isNil ifTrue: [^self].
rect := wnd getWindowRect.
wnd
    sendDelete;
    sendString: '12345+90';
    sendReturn.

delay wait.
screenshot := Screenshot newWindow: wnd saveClientOnly: true.
screenshot captureImage.
bmp1 := Bitmap loadFile: bmpDir, 'Calculator01.bmp'.
screenshot = bmp1
    ifFalse: [screenshot saveDifferenceToBmp: bmp1 toFile: bmpDir, 'Diff01.bmp'].

wnd
    sendDelete;
    sendShortcut: 'Alt+3'.

delay wait.
wnd
    leftClickAt: rect origin + (35 @ 220);
    sendString: '12345'.

delay wait.
screenshot := Screenshot newWindow: wnd saveClientOnly: true
```

Questions?

Contact Information

Star Team (Smalltalk Strategic Resources)

- **Suzanne Fortman** (sfortman@cincom.com)
Cincom Smalltalk Program Director
- **Arden Thomas** (athomas@cincom.com)
Cincom Smalltalk Product Manager
- **Jeremy Jordan** (jjordan@cincom.com)
Cincom Smalltalk Marketing Manager
- **Suzanne Fortman** (sfortman@cincom.com)
Cincom Smalltalk Engineering Manager

Try Cincom Smalltalk

Evaluate Cincom Smalltalk:

 try.cincomsmalltalk.com

Join our Cincom Smalltalk Developer Program:

 develop.cincomsmalltalk.com