

Why do Modern Web Applications Fail and What Can We Do About It ?



Karthik Pattabiraman¹

Frolin Ocariza¹

Kartik Bajaj¹

Ali Mesbah¹

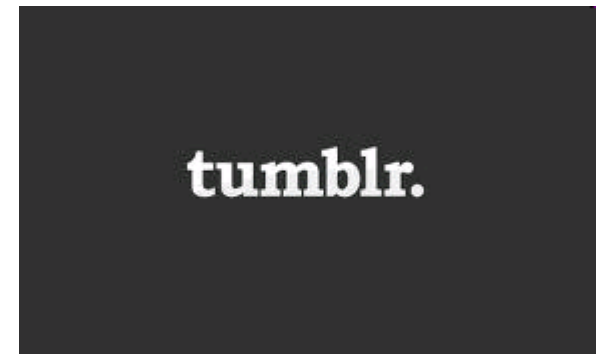
Benjamin Zorn²

¹ University of British Columbia (UBC), ²Microsoft Research (MSR)

My Research

- **Building fault-tolerant software systems**
- **Compiler & runtime techniques for resilience**
 - Error detection in parallel programs [DSN'12]
 - Error detection in soft-computing applns. [DSN'13]
- **This talk**
 - Reliability of modern web applications
 - “VISIONS for SEC&DEP R&D (FUTURE)”

Modern Web Applications



Modern Web Applications: Features



Asynchronous



Multiple Elements

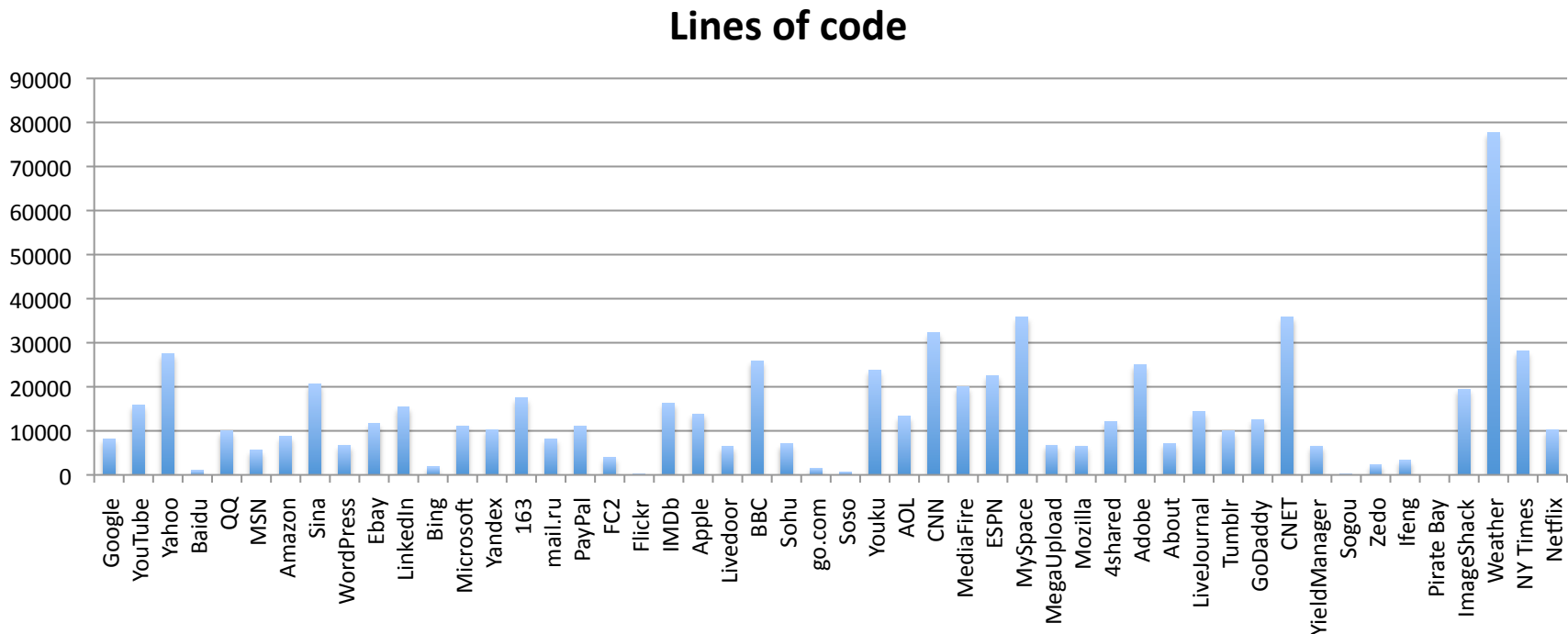


JavaScript

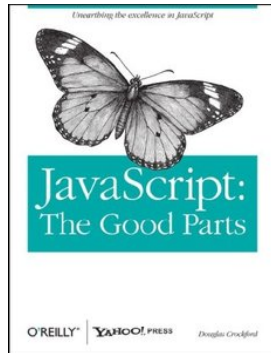
- **JavaScript:** Implementation of ECMAScript standard
- Executes in the web browser – sends AJAX messages
- Responsible for web application's core functionality
 - Not only for appearance and stylistic elements

JavaScript: Prevalence

- 97 of the Alexa top 100 websites use JavaScript
- Thousands of lines of code, often > 10,000



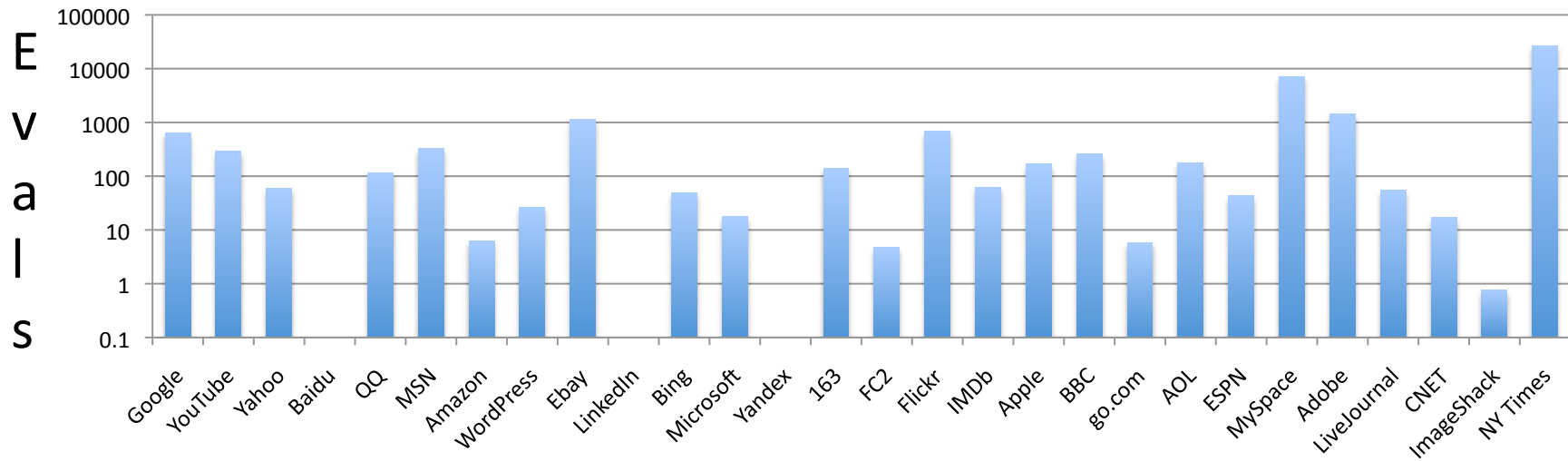
JavaScript: “Good” or “Evil” ?



Vs.



Eval Calls (from Richards et al. [PLDI-2009])



Does Reliability Matter ?

- Snapshot of iFeng.com: Leading media website in China

an error occurred when processing this directive

[an error occurred while processing this directive]

李克强宣布广州亚残运会开幕

火炬手攀登点燃主火炬|数开幕式十宗“最”
亚残运开幕解密|广州亚残运会开幕式特写

广州亚运会圆满闭幕 高清图

[组图]仁川十分钟: Rain连唱三曲|暖场演出
童谣《月光光》拉开序幕|大郅出任中国旗手

女排上演绝地逆转战胜韩国夺冠

周苏红发威女排逆转|韩国输球再斥裁判丑陋
女排逆转令洪钢哽咽|俞觉敏: 我为队员骄傲

[高清]冠军球员搭讪礼仪小姐

裁判引导韩朝摔跤手赛场握手|摔跤精彩瞬间
男篮绝杀伊朗进决赛|朝鲜女足失冠背向升旗

- “铁血女将”黄蕴瑶暂列亚运英雄榜之首
- 中华台北选手罹癌参赛 携奖牌返家无遗憾
- 日本男女足亚运齐称霸 统治亚洲足坛获证
- 霍启刚温尔雅态度和蔼 与郭晶晶差别大
- 快讯: 广州亚运会发生第二起兴奋剂事件
- 阿联首绝杀韩国队 将与日本争男足金牌
- 韩朝射箭选手只关注比赛 不知两国冲突



Studies of JavaScript Web Applications

Performance and parallelism:

[Ratanaworabhan-WebApps2010],
[Richards - PLDI2009], [Fortuna –
IISWC2011], [Mehrara – HPCA2011]

Reliability

?

Security and Privacy:

[Yue-WWW 2009],
[Guarnieri-Usenix 2009],
[Jang-CCS 2010]



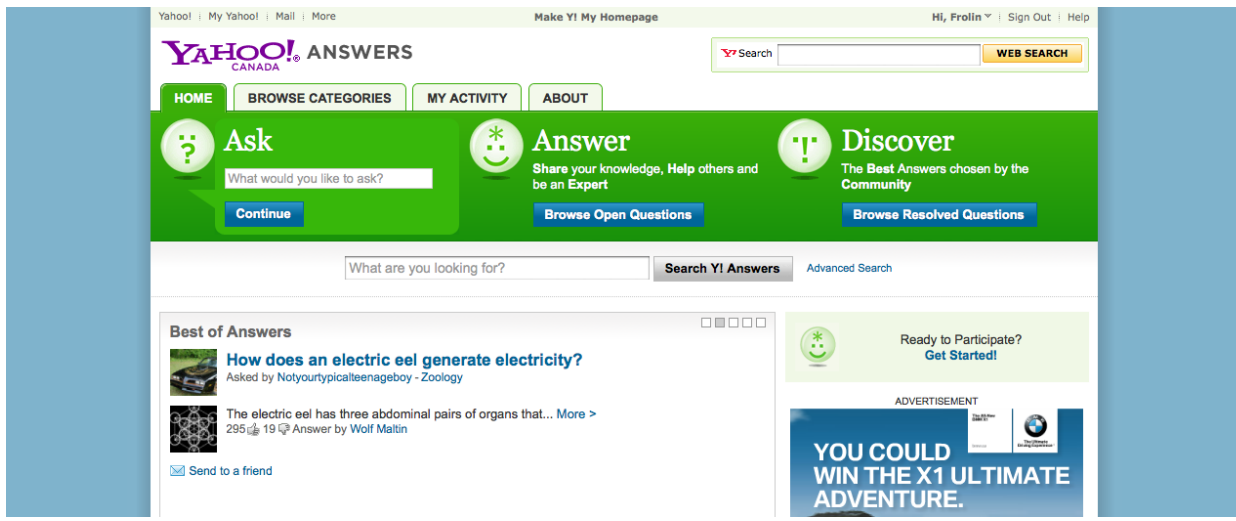
Goal: Study and measure the reliability of JavaScript web applications

This Talk

- Motivation and Approach
- Two approaches for studying JS Reliability
 - Log Messages [ISSRE 2011] – With F. Ocariza and B.G. Zorn
 - Bug Reports [ESEM 2013] – With F. Ocariza, K. Bajaj, and A. Mesbah
- Future Directions and Conclusions

JSER: JavaScript Error Messages

- All exceptions thrown are logged to JS console



The screenshot shows the Yahoo! Answers website interface. At the bottom, a JavaScript error console is open, displaying three error messages:

```
✖ C is null
  if (window.yzq_p==null) document.write("../d/lib/bc/bcr_2.0.5.js"</scr+"ipt>");

✖ Permission denied for <http://ad.yieldmanager.com> to call method Location.toString on <http://ca.answers.yahoo.com>.
  document.write("<scr+"ipt type='text/...rc='"+prHost+pr_s+"'</scr+"ipt>");

✖ E("set_hp_firefox_instructions") is undefined
  if("undefined"!==typeof YAHOO&&YAHOO){...etTimeout(arguments.callee,100)}});
```

Multiple exceptions

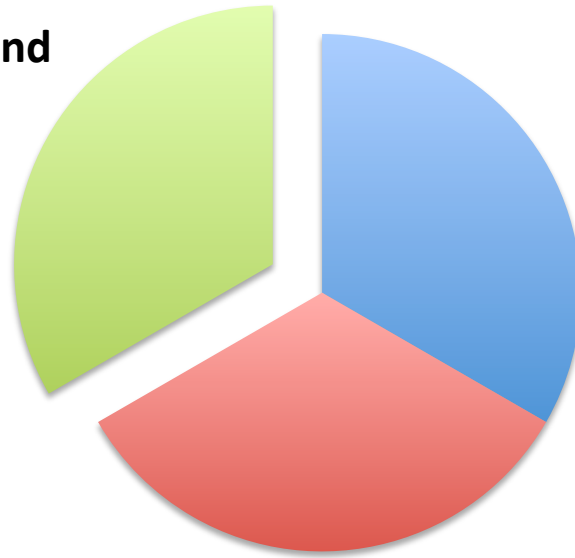
JSER: Tools

- Chose 50 web applications from Alexa top 100
- Created Selenium tests (15 per application)
- Capture JavaScript Errors printed to Firebug



JSER: Research Questions

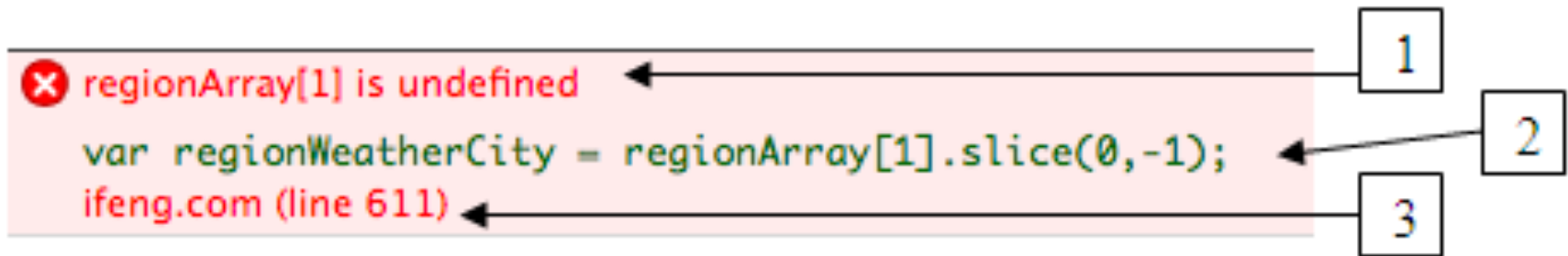
Do errors occur in web apps and if so, what categories do they fall in ?



How do errors correlate with static and dynamic characteristics of the app?

How do errors vary by speed of testing ? Are they all deterministic ?

JSER: Method

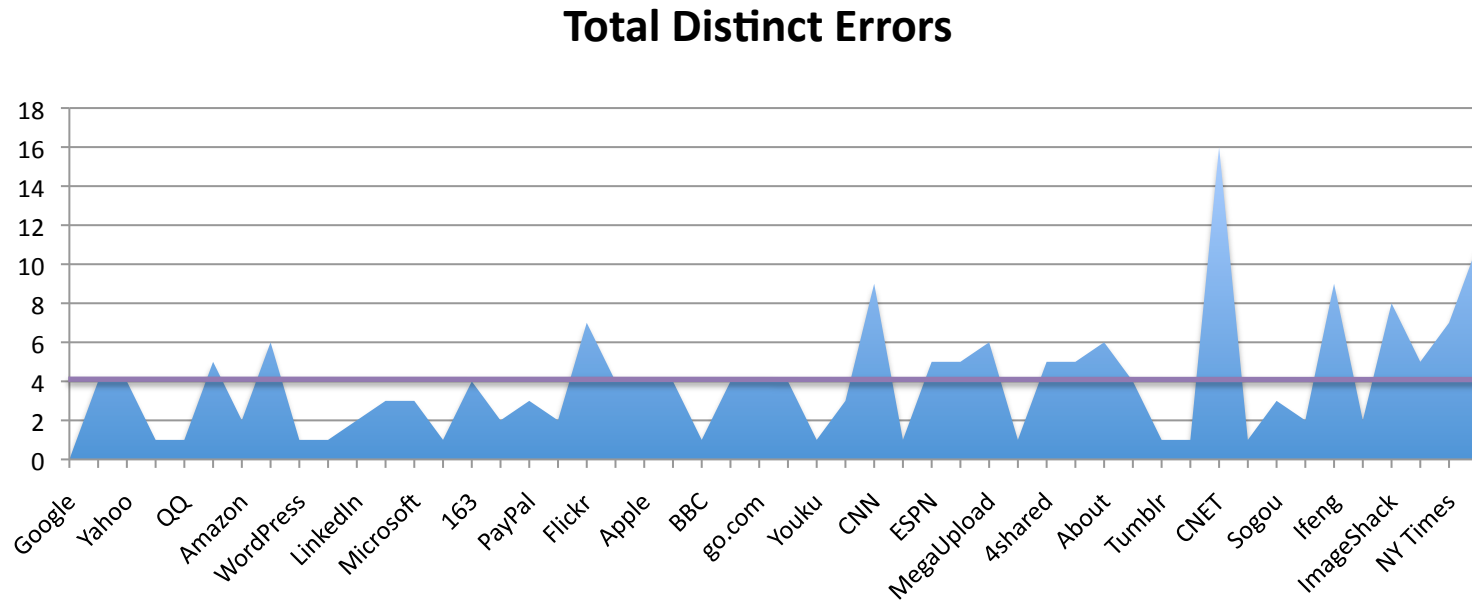


1. Description of error message
2. Line of code corresponding to error
3. Domain name and line number

Two errors are different if any attribute is different

JSER: Error Frequencies Results

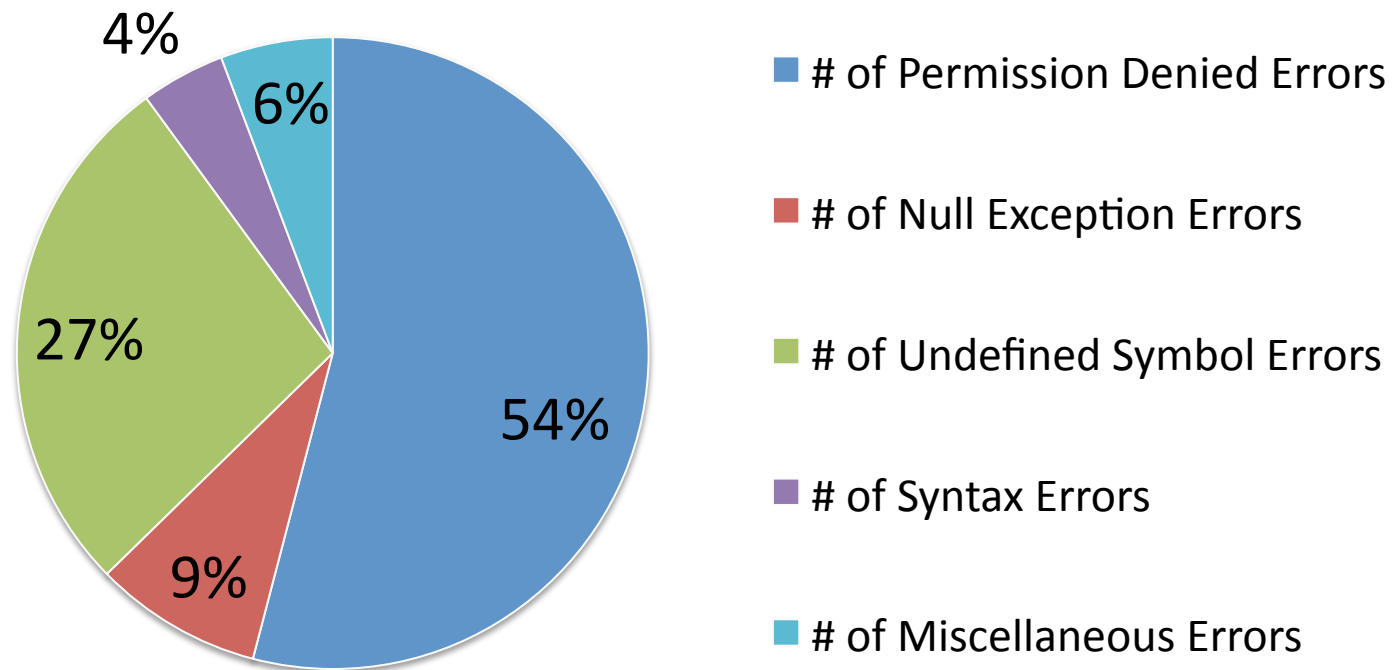
- Average of **4 distinct error messages/application**
 - Conservative estimate (only distinct messages)
 - Fifteen test cases per application



JSER: Error Classification Results

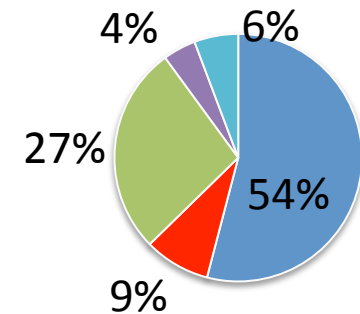
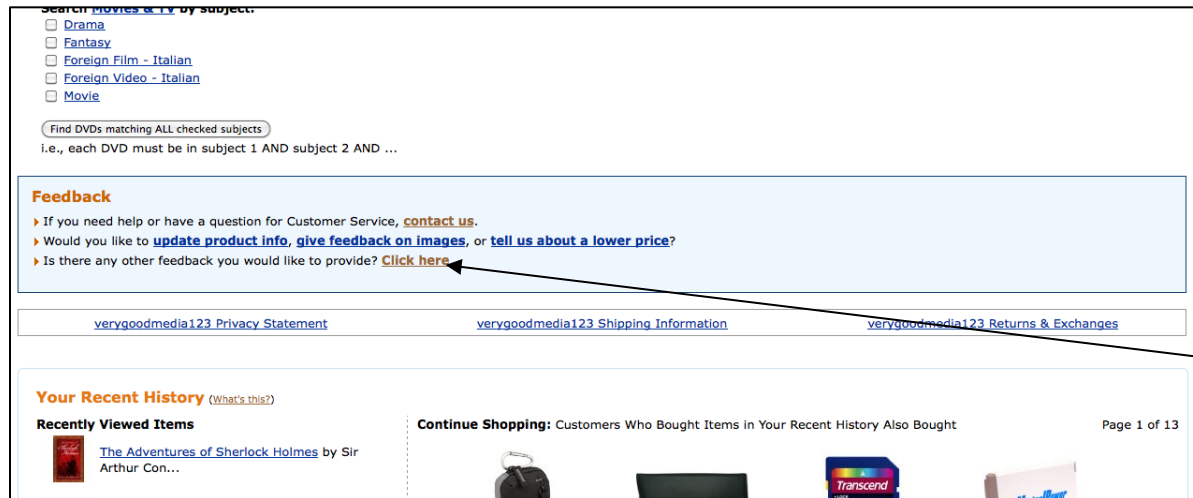
- 94 % of errors fall into four predominant categories

Distribution of Error Messages



JSER: Null Exception Example

Taken from
amazon.com



Causes error
on click

- **Error Message:** `document.getElementById("inappDiv")` is null

```
document.getElementById("inappDiv").style.display = 'none';
```

- **Explanation:** `inappDiv` was only defined for users who are logged in

- **Bottom Line:** JS errors may expose security issues

JSER: Research Questions

Do errors occur in web apps and if so, what categories do they fall in ?

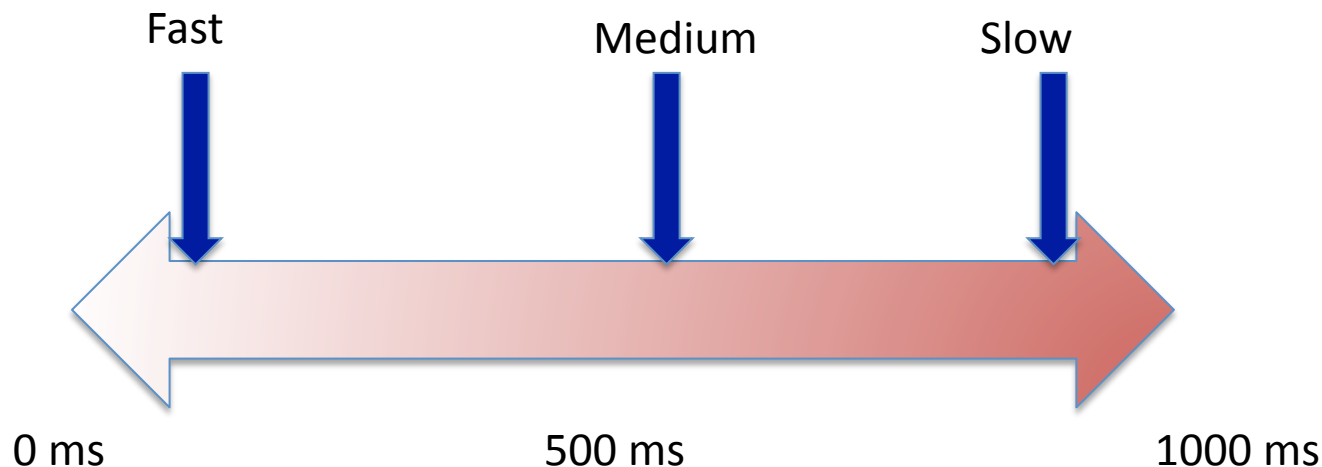


How do errors correlate with static and dynamic characteristics of the app?

How do errors vary by speed of testing ? Are they all deterministic ?

JSER: Effect of Testing Speed

- Varied testing speed for replaying events in Selenium
- Performed three executions in each testing speed



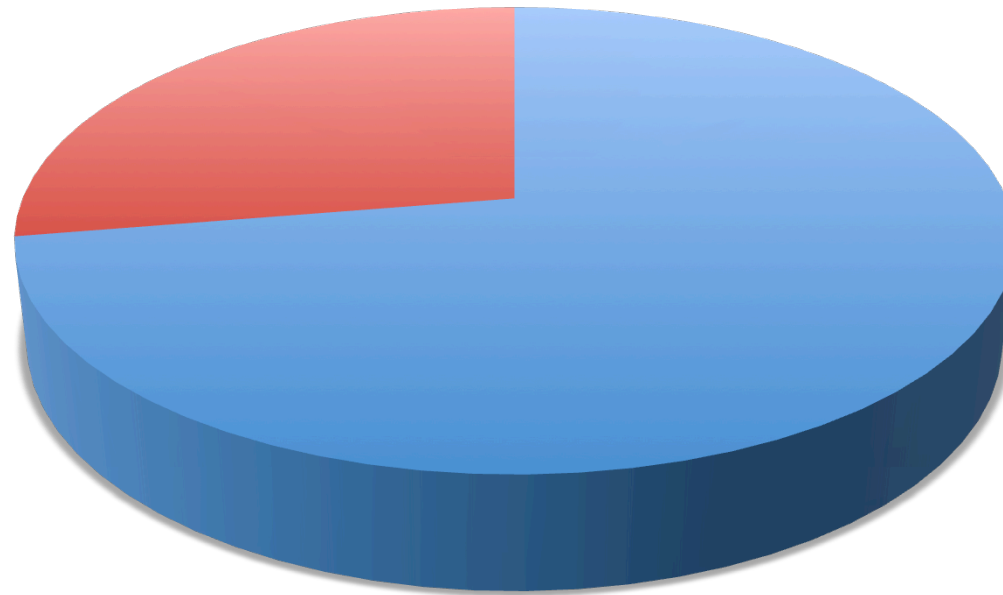
JSER: Testing Speed Results (CNN)

Error Message (shortened)	F	F	F	M	M	M	S	S	S
	1	2	3	1	2	3	1	2	3
Permission Denied for view.atdmt.com to call <fname> on marquee.blogs.cnn.com	4	4	4	1	3	3	2	2	3
window.parent.CSManager is undefined	0	0	0	0	0	0	1	1	0

Many Errors are Non-Deterministic

JSER: Non-Determinism

- More than 70% of errors are non-deterministic



■ Total non-deterministic errors

JSER: Research Questions

Do errors occur in web apps and if so, what categories do they fall in ?



How do errors correlate with static and dynamic characteristics of the app?

How do errors vary by speed of testing ? Are they all deterministic ?

JSER: Static/Dynamic Characteristics

Static Characteristics

Measured using Phoenix & Firebug plugins

- Alexa Rank
- Bytes of JavaScript code
- Number of domains
- Number of frameworks

Dynamic Characteristics

From Richards et al. [PLDI – 2010]

- Number of called functions
- Number of eval calls
- Properties deleted
- Object inheritance overridings

JSER: Correlations Summary

Static Characteristics

- **Alexa Rank**
- Bytes of JavaScript code
- **Number of domains**
- **Number of frameworks**

Dynamic Characteristics

- Number of called functions
- Number of eval calls
- Properties deleted
- Object inheritance overridings

JSER: Summary

- JavaScript errors **abound** in production web applications under normal operations
- JavaScript errors vary by speed of testing; majority of the errors are **non-deterministic**
- JavaScript errors are **not correlated** with code size or with eval calls, but with other metrics

This Talk

- Motivation and Approach
- Two approaches for studying JS Reliability
 - Log Messages [ISSRE 2011] – With F. Ocariza and B.G. Zorn
 - Bug Reports [ESEM 2013] – With F. Ocariza, K. Bajaj, and A. Mesbah
- Future Directions and Conclusions

JavaScript Bug Report Study

- What are the root **causes of** JavaScript faults?



- What **impact** do JavaScript faults have?



Bug Report Study of twelve popular, Open Source JavaScript Applications



Bug Report Study: Results

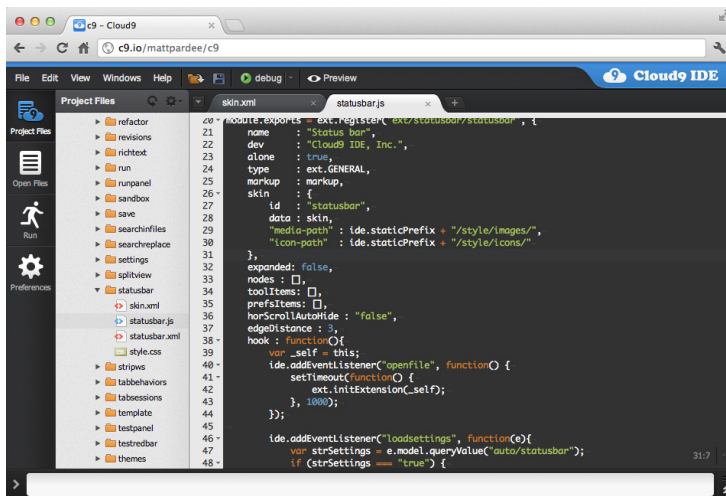
- **Bug report study of 12 web applications**
 - Over 300 bug reports analyzed; only fixed bugs
- **DOM-related errors dominate JavaScript errors (66% of total errors)**
 - DOM related errors often lead to wrong output rather than exceptions (89% versus 39%)
 - DOM related errors responsible for 80% of highest impact faults (e.g., major functionality loss)

This Talk

- Motivation and Approach
- Two approaches for studying JS Reliability
 - Log Messages [ISSRE 2011] – With F. Ocariza and B.G. Zorn
 - Bug Reports [ESEM 2013] – With F. Ocariza, K. Bajaj, and A. Mesbah
- **Future Directions and Conclusions**

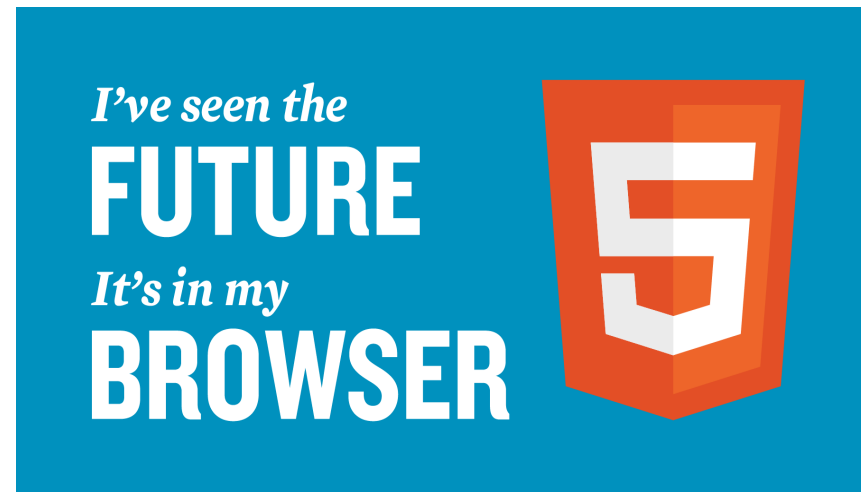
Future Directions

- **Techniques to build dependable web applications**
 - Fault Localization [ICST'12 – best paper award nominee]
 - Mutation testing [ICST'13 – best paper award runner up]
 - Robustness testing [ISSRE'10]
- **Effect of HTML5 primitives and features**
 - Canvas interactions, local storage, database etc.



The screenshot shows the Cloud9 IDE interface. The main editor displays JavaScript code for a status bar component. The code includes a jQuery plugin registration, a constructor function for the status bar, and event listeners for 'openfile' and 'loadsettings'. The 'openfile' listener sets a timeout to initialize the extension. The 'loadsettings' listener checks for a specific setting and updates the status bar accordingly.

```
21 jQuery.extend(exports, {
22   name: "Status bar",
23   dev: "Cloud9 IDE, Inc.",
24   alone: true,
25   type: ext.GENERAL,
26   markup: markup,
27   skin: {
28     id: "statusbar",
29     data: skin,
30     media-path: ide.staticPrefix + "/style/images/",
31     icon-path: ide.staticPrefix + "/style/icons/"
32   },
33   expanded: false,
34   nodes: [],
35   toolItems: [],
36   prefItems: [],
37   horScrollAutoHide: "false",
38   edgeDistance: 0,
39   hook: function() {
40     var self = this;
41     ide.addListener("openfile", function() {
42       setTimeout(function() {
43         ext.initExtension(self);
44       }, 1000);
45     });
46     ide.addListener("loadsettings", function() {
47       var strSettings = e.model.queryValue("auto/statusbar");
48       if (strSettings === "true") {
```



Conclusions

- **Modern web applications increasingly important**
 - Reliability (and security) is a significant challenge
- **Measured the reliability of modern web applications**
 - Based on console log messages [ISSRE'11]
 - Based on openly available bug reports [ESEM'13]
- **This is an area that we in the dependability community should be leading**
 - Significant impact possible if we jump into this now !
 - Other communities will do this, and we may not like it !

Backup Slides

Modern Web Application: Example



Modern Web applications provide rich functionality without leaving the page

Experimental Objects

Eight JavaScript Web Applications



TYPO3

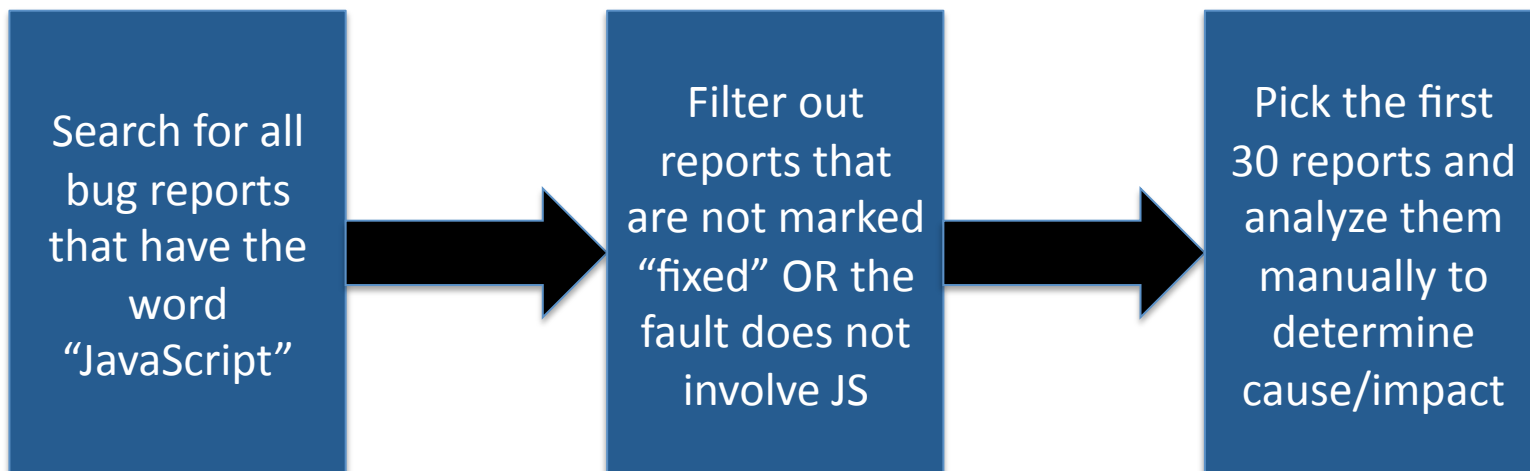


Four JavaScript Libraries



Experimental Methodology

- Collect bug reports from bug repositories
 - Focus on bugs that are marked fixed
 - Organized into a uniform format (XML file)



DOM-Related Error: Example

ID of element to retrieve: hello_world

```
1  var toggle = 1;
2  var x = "hlelo_";
3  var y = "world";
4  var elem = document.getElementById(x + y);
5  var dis = "";
6  if (toggle == 1) {
7      dis = "block";
8  }
9  else {
10     dis = "inline";
11 }
12 elem.style.display = dis;
```

Fault: "hello_" is misspelled

Error: Code would attempt to retrieve the DOM element using wrong ID. *elem* becomes **NULL**

Failure: NULL EXCEPTION!

Research Questions

- **RQ1:** What types of JavaScript *errors* occur in web apps ?
- **RQ2:** What is the nature of *failures* stemming from JS errors ?
- **RQ3:** What is the impact of JS errors ?

Research Questions

- **RQ1:** What types of JavaScript *errors* occur in web apps ?
- **RQ2:** What is the nature of *failures* stemming from JS errors ?
- **RQ3:** What is the impact of JS errors ?

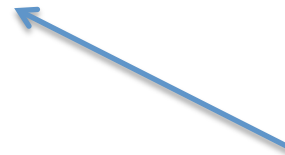
Research Questions

- **RQ1:** What types of JavaScript *errors* occur in web apps ?
- **RQ2:** What is the nature of *failures* stemming from JS errors ?
- **RQ3:** What is the impact of JS errors ?

AutoFlox [Ocariza – ICST 2012]

- **AutoFlox**: Automatic fault localization tool for JS
 - Find origin of the null value
 - i.e., find the *direct DOM access*

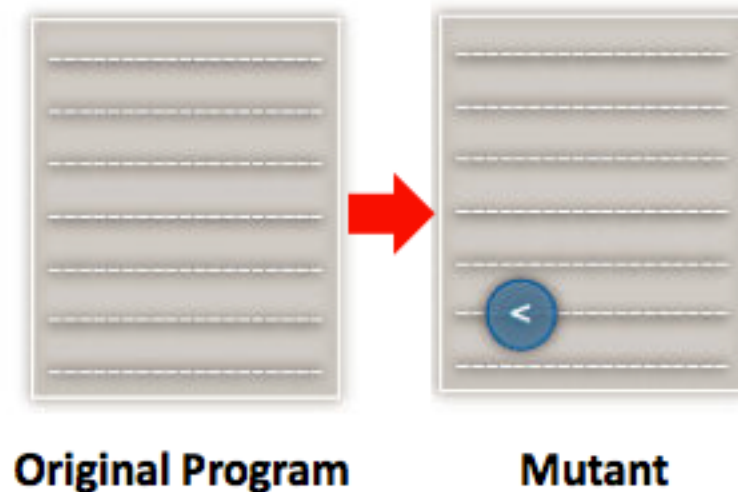
```
1  var toggle = 1;
2  var x = "hlelo_";
3  var y = "world";
4  var elem = document.getElementById(x + y);
5  var dis = "";
6  if (toggle == 1) {
7      dis = "block";
8  }
9  else {
10     dis = "inline";
11 }
12 elem.style.display = dis;
```



Direct DOM Access
(This is where the NULL
value came from)

Mutandis [Mirshokraie - ICST 2013]

- Mutates original program to test quality of test suites
- Problem: Equivalent mutants obscure the value
- Generate only a few equivalent mutants by ranking functions by their importance: FunctionRank



Modern Web Application: JavaScript



```
109.     }
110.   }
111. }
112.
113. S9MultiPackLayout.prototype.makeVisible = function() {
114.   var numProposedVisibleItems = this.numProposedVisibleItems();
115.   var lastVisibleCol = this.firstVisibleCol + numProposedVisibleItems - 1;
116.   var width = ((100 / numProposedVisibleItems)-1);
117.
118.   if (this.seedItem) {
119.     this.seedItem.style.width = width + "%";
120.     this.itemChildren[0].style.display = "";
121.     this.itemChildren[0].style.width = "100%";
122.     this.otherItems.style.width = (98 - width) + "%";
123.     var widthWithoutSeed = ((100 / (numProposedVisibleItems-1))-1);
124.     for (var i = 1; i < this.itemChildren.length; i++) {
125.       if ((i >= this.firstVisibleCol) && (i <= lastVisibleCol)) {
126.         this.itemChildren[i].style.display = "";
127.         this.itemChildren[i].style.width = widthWithoutSeed + "%";
128.         if (this.itemImages[i].getAttribute("url")) {
129.           this.itemImages[i].src = this.itemImages[i].getAttribute("url");
130.           this.itemImages[i].setAttribute("url", "");
131.         }
132.       } else {
133.         this.itemChildren[i].style.display = "none";
134.       }
135.     }
136.   } else {
137.     for (var i = 0; i < this.itemChildren.length; i++) {
138.       if ((i >= this.firstVisibleCol) && (i <= lastVisibleCol)) {
139.         this.itemChildren[i].style.display = "";
140.         this.itemChildren[i].style.width = width + "%";
141.         if (this.itemImages[i].getAttribute("url")) {
142.           this.itemImages[i].src = this.itemImages[i].getAttribute("url");
143.           this.itemImages[i].setAttribute("url", "");
144.         }
145.       } else {
146.         this.itemChildren[i].style.display = "none";
```

Significant amount of JavaScript code executing in the browser

Modern Web Application: Console

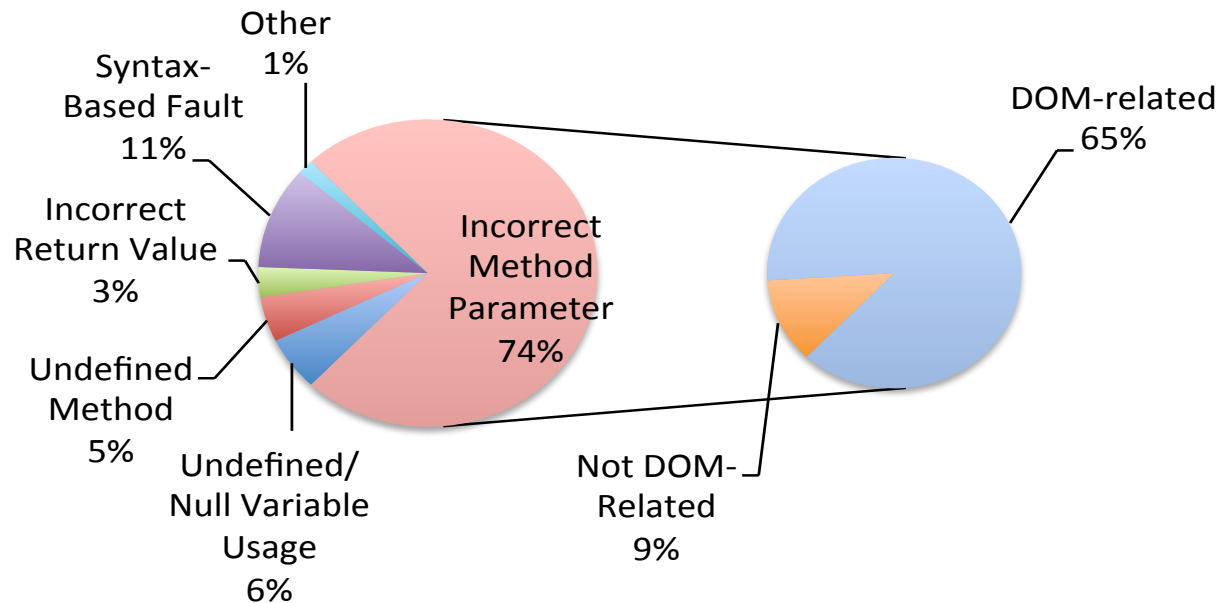
```
✖ unterminated string literal
zGPU='http://www.about.com/'" search.about.com (line 22)
✖ ▶ ss is not defined
return ss("go to wildtangent.com"); search.about.com (line 2)
✖ ▶ cs is not defined
return ss("go to www.toys-games.best-price.com"); search.about.com (line 2)
✖ ss is not defined
✖ ▶ cs is not defined
cs(); search.about.com (line 2)
✖ unterminated string literal
zGPU='http://www.about.com/'" search.about.com (line 22)
✖ Permission denied for <http://ad.yieldmanager.com> (document.domain has not been set) to call method Location.toString on <http://search.about.com> (document.domain=<http://about.com>).
✖ unterminated string literal
zGPU='http://www.about.com/'" search.about.com (line 22)
✖ ▶ ss is not defined
return ss("go to ask.com"); search.about.com (line 2)
✖ ▶ cs is not defined
return ss("go to ask.com"); search.about.com (line 2)
✖ ss is not defined
✖ ▶ cs is not defined
cs(); search.about.com (line 2)
✖ ▶ ss is not defined
return ss("go to wildtangent.com"); search.about.com (line 2)
✖ ▶ cs is not defined
return ss("go to wildtangent.com"); search.about.com (line 2)
✖ ss is not defined
✖ ▶ cs is not defined
```

Modern Web Apps experience errors even during normal execution !

Research Questions

- **RQ1:** What types of JavaScript *faults* occur in web apps ?
- **RQ2:** What is the nature of *failures* stemming from JS faults, and what is the *impact*?
- **RQ3:** What is the root cause of JS faults?
- **RQ4:** Are JS faults browser-specific?
- **RQ5:** How long does it take to fix a JS fault?

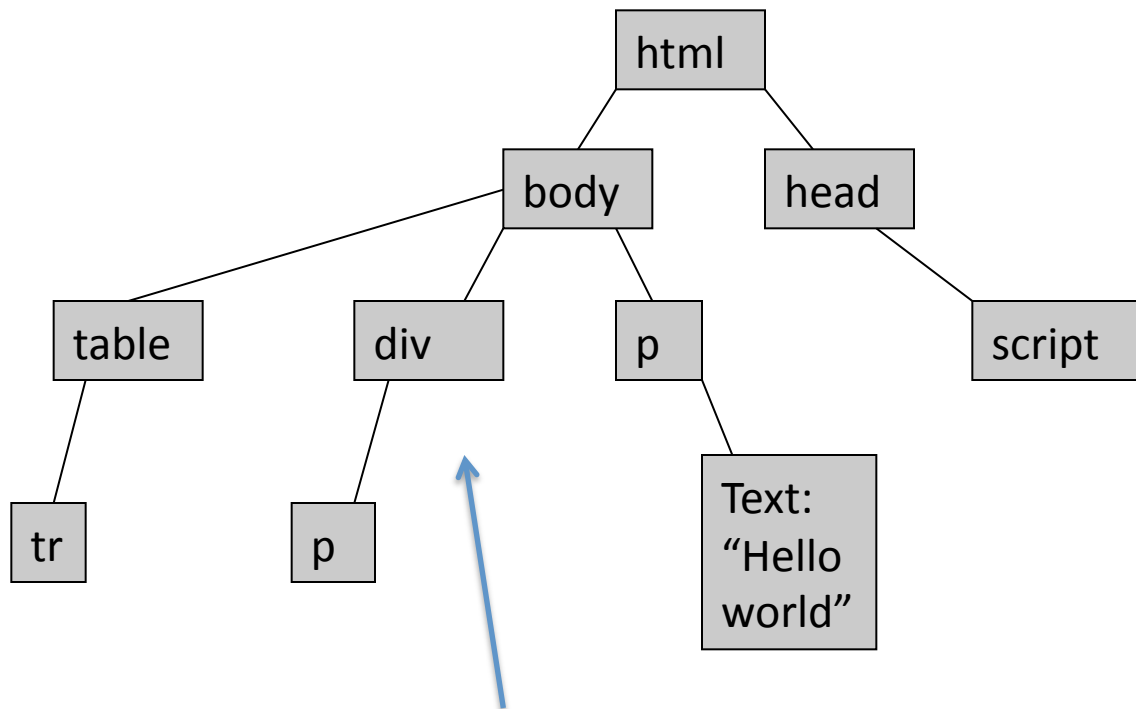
Results: Error Categories



Incorrect Method Parameter Error: Unexpected or invalid value passed to JS method or assigned to JS property.

DOM-Related Error: The JS method is a DOM API method
- Account for around two-thirds of JavaScript Errors

DOM-Related Errors



Want to retrieve this element

DOM-Related Errors

JavaScript code: `var x = document.getElementById("elem");`



DOM-related
JavaScript error

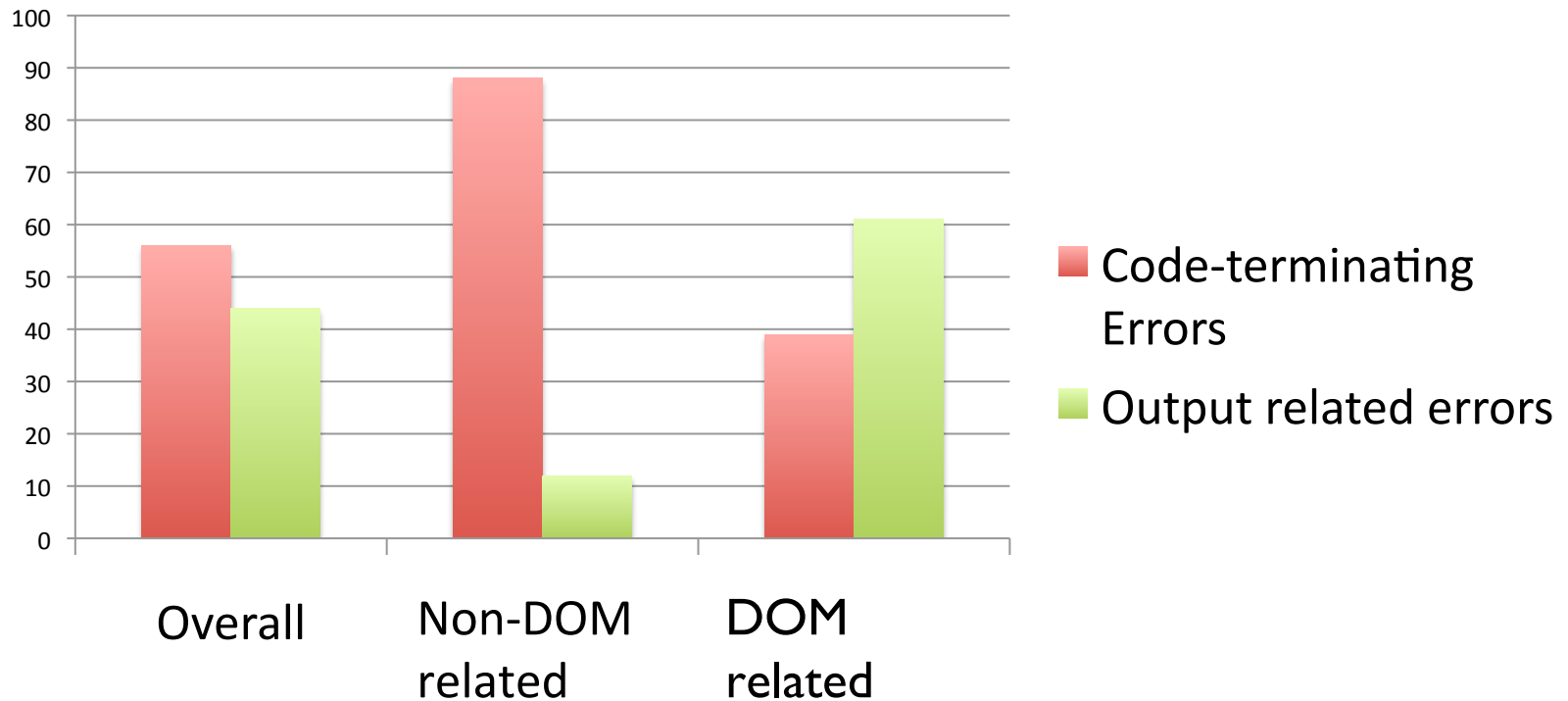
...with ... the ... existent ID

DOM:

id: elem

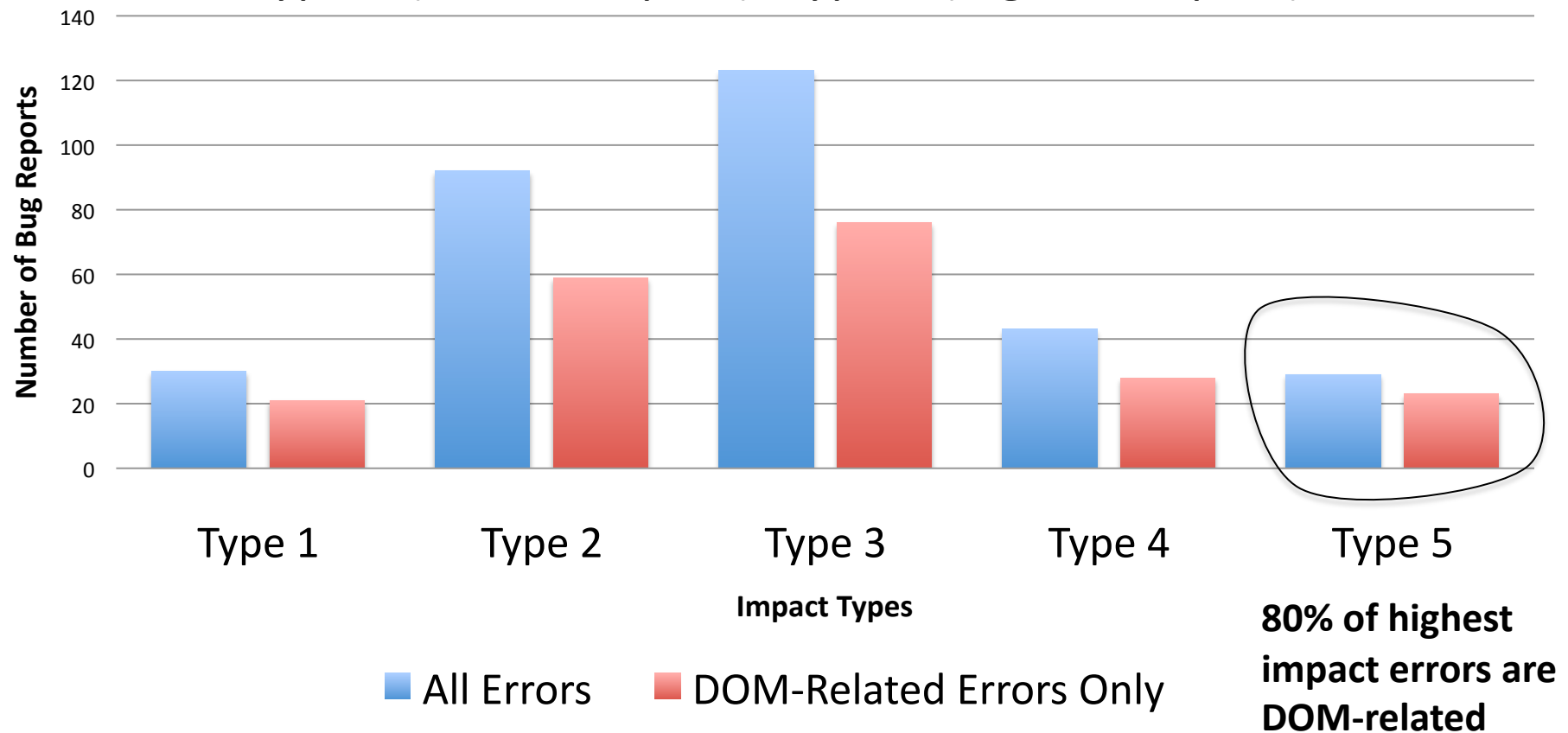
Results: Nature of Failures

- DOM related errors are less likely to be code terminating i.e., exceptions (39% Vs. 89%)



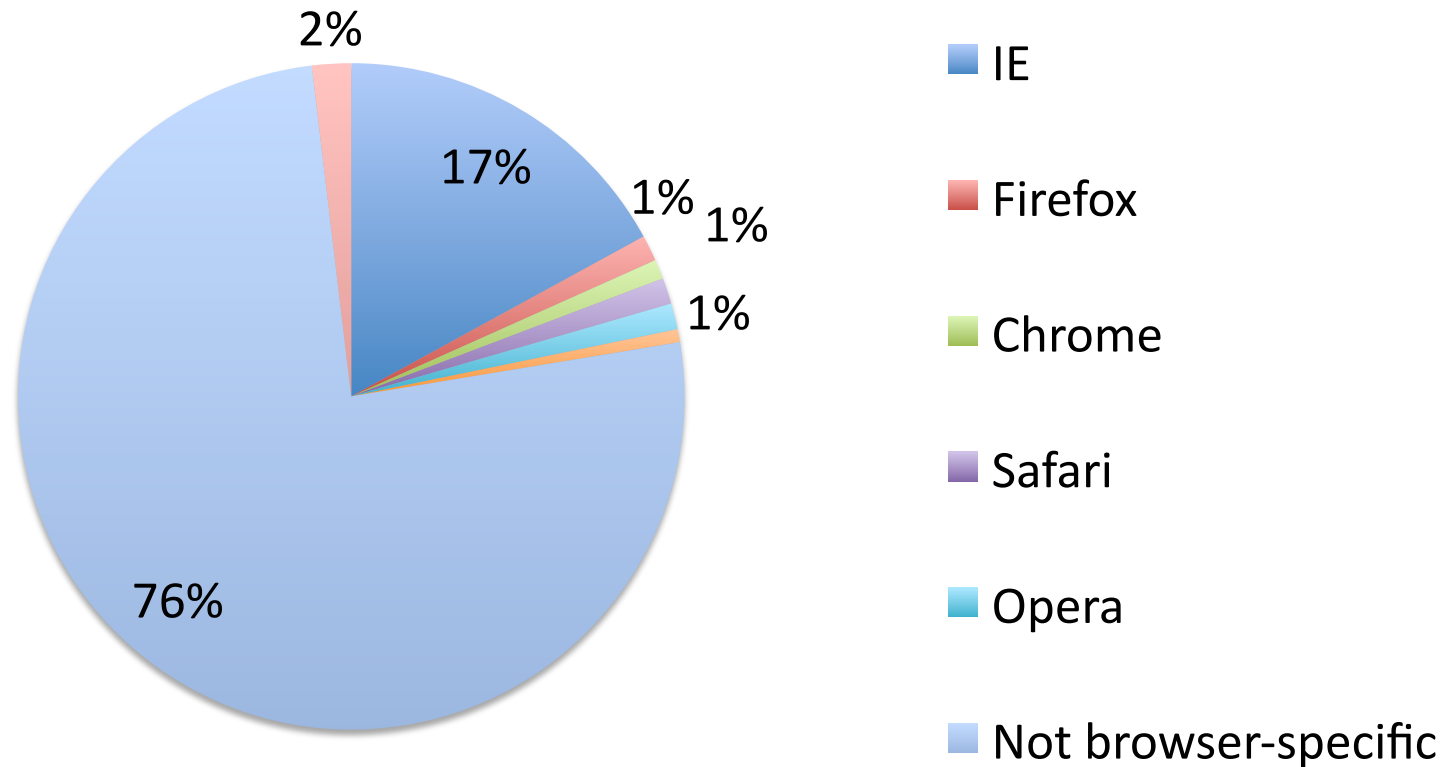
Results: Impact of JavaScript Errors

- Impact Types – Based on Bugzilla classification
 - Type 1 (lowest impact), Type 5 (highest impact)



Results: Browser Specificity

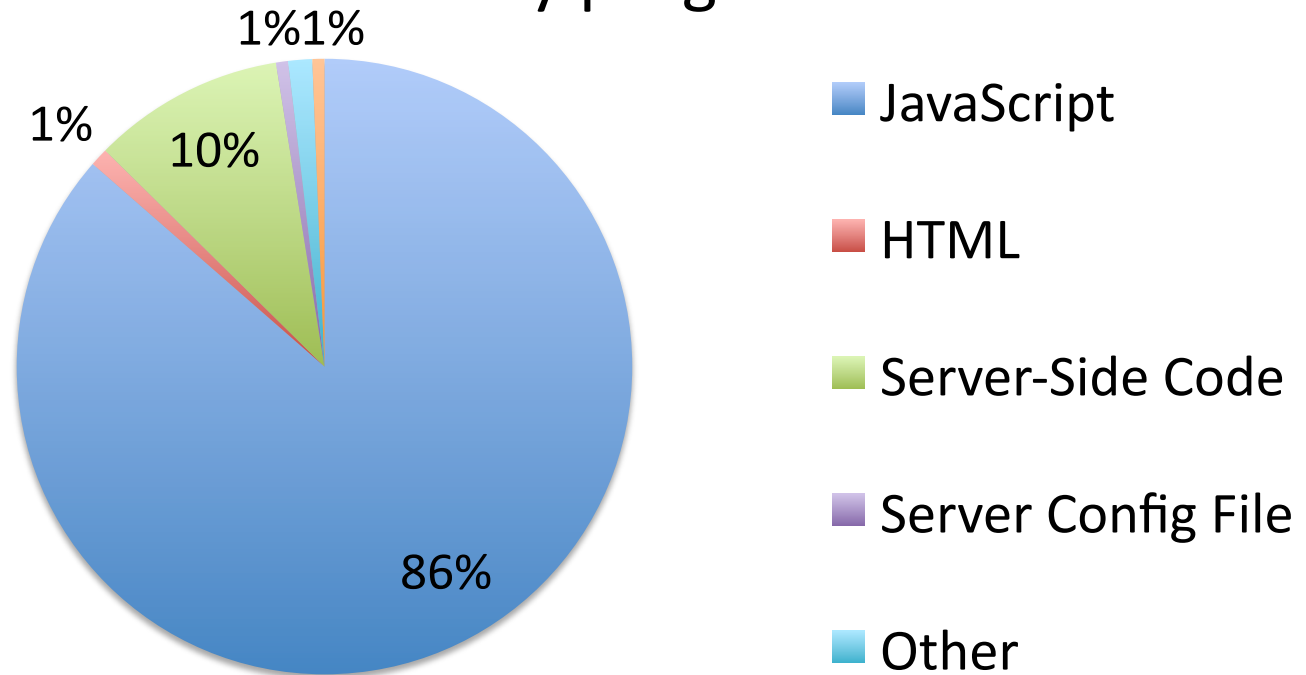
Most JavaScript faults are not browser-specific



Results: Causes of JS Faults

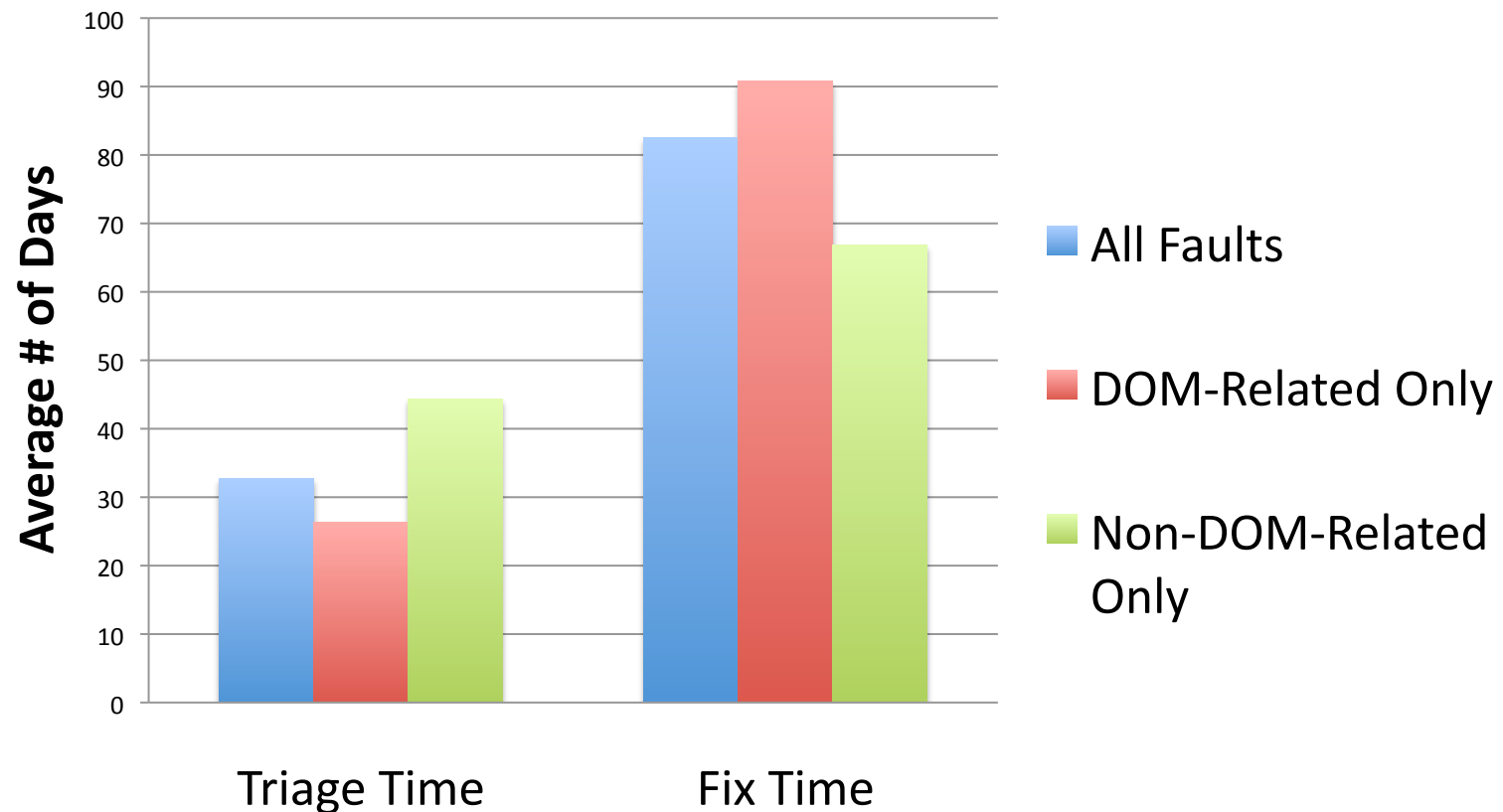
- **Error Locations**

- Most errors committed by programmer in JS code



Results: Triage and Fix Times

- **Triage Time:** Time it took to assign/comment on bug
- **Fix Time:** Time it took to fix the bug since it was triaged



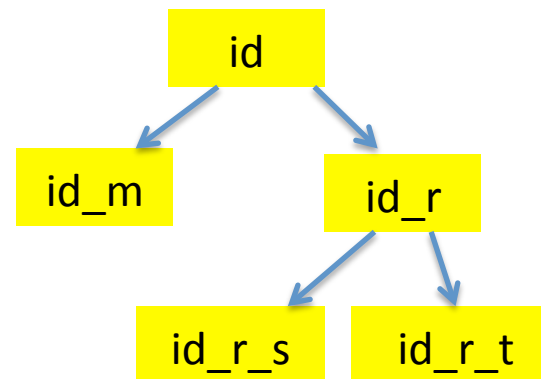
Vejovis [Ocariza – To be submitted]

- **Vejovis:** automatic fix suggestion of DOM-related errors
 - Starts at direct DOM access
 - Provide fix suggestions based on common patterns in DOM

```
1 var toggle = 1;
2 var x = "hlelo_";
3 var y = "world";
4 var elem = document.getElementById(x + y);
5 var dis = "";
6 if (toggle == 1) {
7     dis = "block";
8 }
9 else {
10    dis = "inline";
11 }
12 elem.style.display = dis;
```

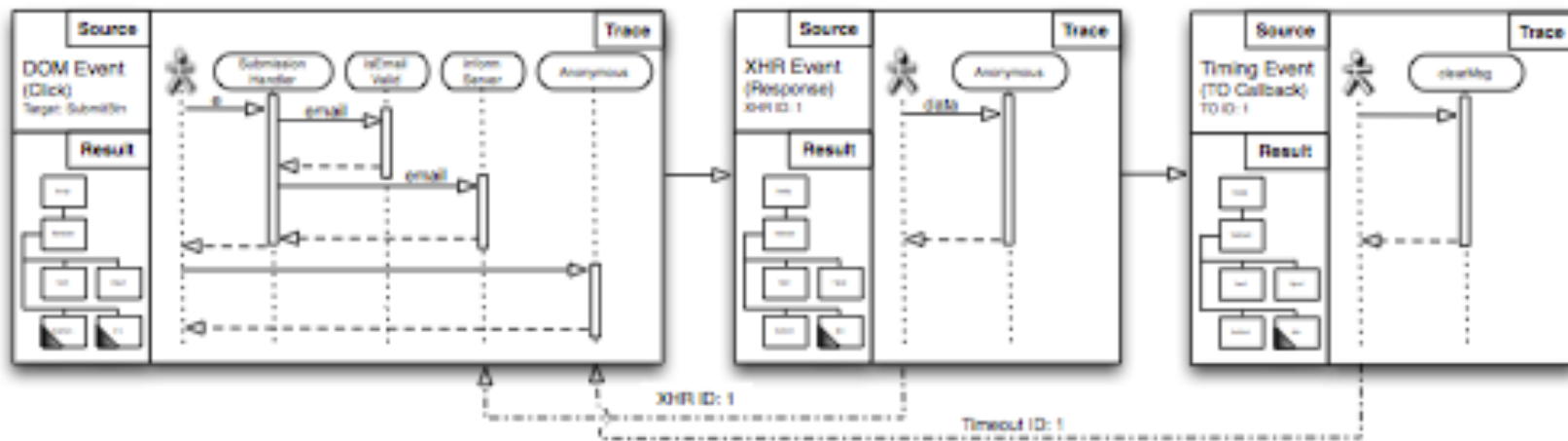
Parameter

Find "potential replacements" in DOM



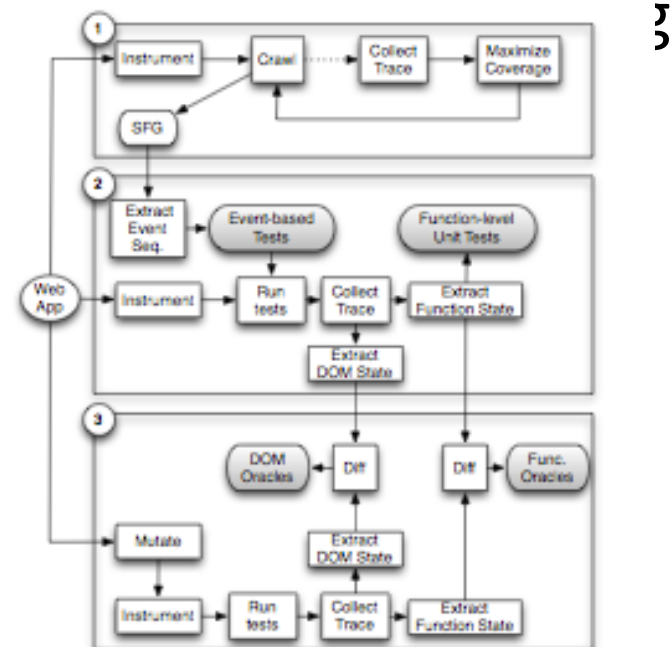
Clematis [Alimadi – under review]

- Challenge: Web applications are complex, and consist of DOM interactions, AJAX messages and timeouts
- Difficult to trace the links between events and JS code
- Clematis allows users to visualize causal dependencies between events and code, and between asynchronous events



Pythia [Mirshokraie – under review]

- Automated unit test and oracle generation for web apps.
- First, crawls application to generate event sequences
- Extracts unit tests from sequences with high coverage
- Creates Oracles for unit tests



Nature of Failures

Code-terminating

```
element = getElementById("elem");
```

```
b = element.getAttribute("badAttr")
```

```
element.innerHTML = "text";
```

```
b.value = "newValue";
```



exception

Output DOM-related

```
function changeToBlue(elem) {
```

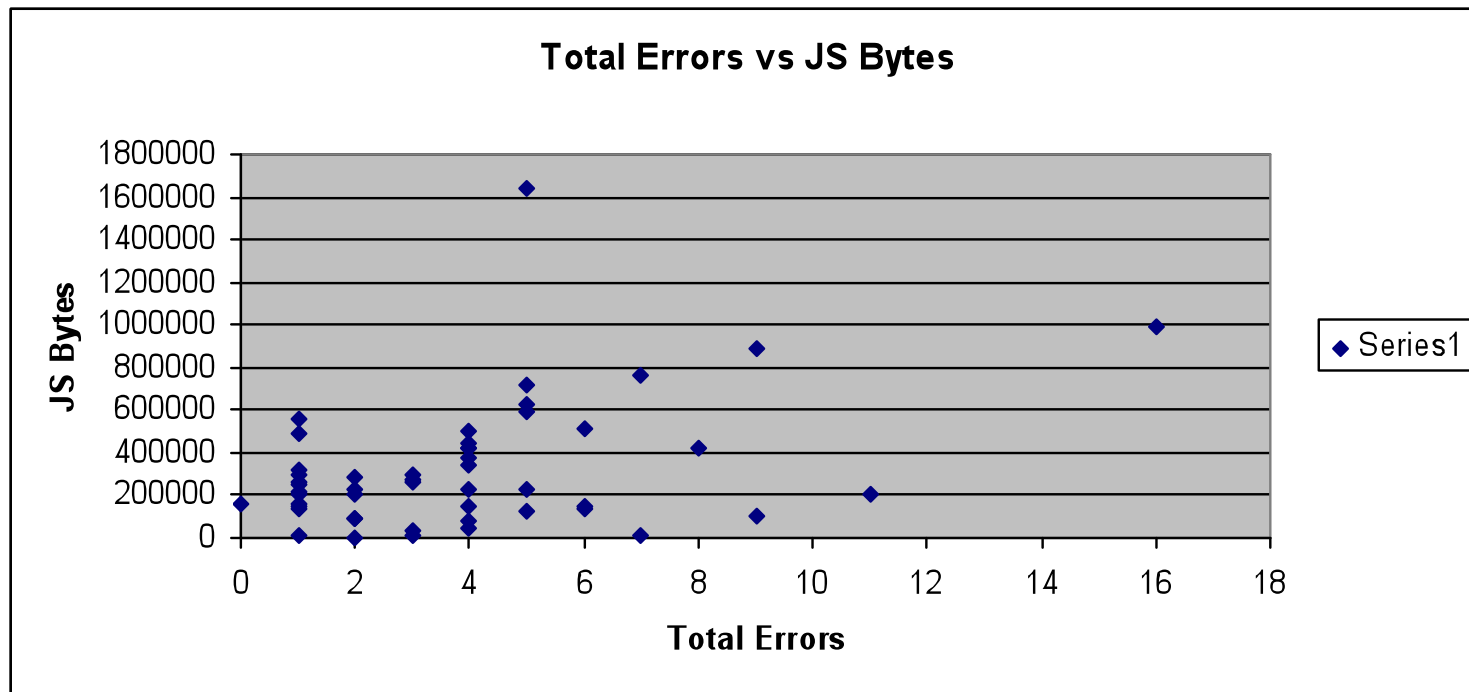
```
    element = getElementById(elem)
```

```
    element.style.color = "red"; Wrong colour change
```

```
}
```

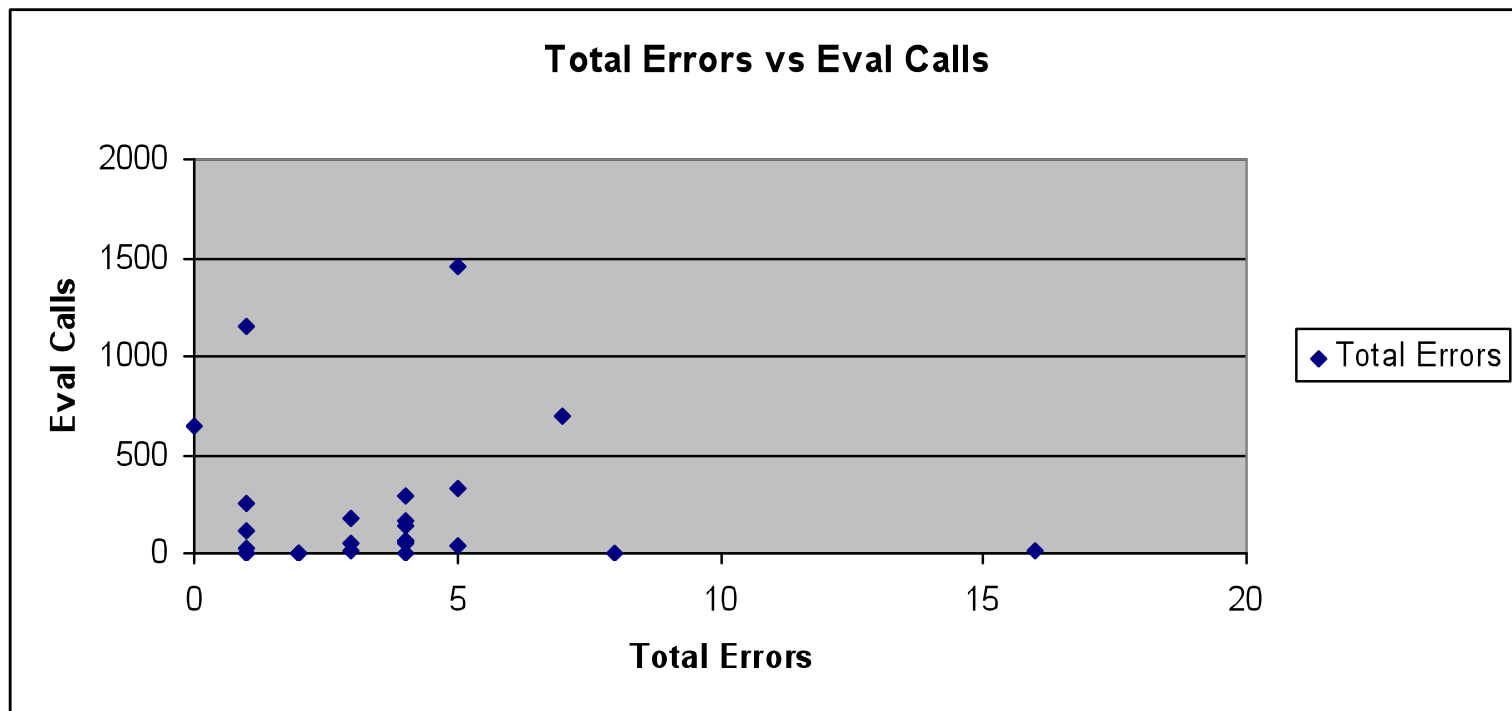
JSER: JS Code Size Correlations

- Low correlation
 - JS reliability not correlated with code size



JSER: Eval Calls Classification

- Low correlation
 - Eval calls do not seem to influence reliability



Towards Dependable Web Applications

- **Focus on understanding, testing and fixing errors in web applications automatically**
 - Web application understanding behavior (Clematis)
 - Test generation (**Mutandis – ICST'13 best paper runner up**, Pythia – under submission to ASE'13)
 - Fault localization and suggesting fixes (**AutoFLox – ICST'12 best paper nominee**, Vejovis)