A BENCHMARK OF JAVASCRIPT BUGS

Péter Gyimesi, Uni. Szeged
Béla Vancsics, Uni. Szeged
Andrea Stocco, USI
Davood Mazinanian, UBC
Árpád Beszédes, Uni. Szeged
Rudolf Ferenc, Uni. Szeged
Ali Mesbah, UBC
Developer Survey Results 2019
The **strength** of JavaScript is that you can do **anything**.

The **weakness** is that **you will**.

- Reg Braithwaite
ANALYSIS AND TESTING FOR JS
PROPOSED BENCHMARK OF JS BUGS

GOALS, CHALLENGES, CHARACTERISTICS
### Goals
- *Centralized benchmark*
- *Real bugs*
- *Available*

### Characteristics
- Reproducibility
- Isolation

### Challenges
- Bug Tracking
- Testing Frameworks
- Client vs server side
- NodeJS
- Dependencies
10 initial subjects
50 initial subjects
453 final subjects
10 final subjects
795 initial bugs
453 final bugs
complex - refactoring - irrelevant changes -
BugsJS

ARCHITECTURE, FUNCTIONALITIES, USAGE
<table>
<thead>
<tr>
<th>Bug Repository</th>
<th>Subjects</th>
<th>Source Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility Framework</td>
<td>Shields</td>
<td></td>
</tr>
<tr>
<td>Bug Statistics</td>
<td>Eslint</td>
<td>Tests</td>
</tr>
<tr>
<td>Test Commands</td>
<td>Mongoose</td>
<td>Cleaned Patches</td>
</tr>
<tr>
<td>Bug Report Data</td>
<td>PencilBlue</td>
<td>Tagged Bug Fixes</td>
</tr>
</tbody>
</table>

**Docker Image**

Pre-built Environment
Usage of BugsJS

download
$ git clone https://github.com/BugsJS/bug-dataset.git

run
$ git clone https://github.com/BugsJS/docker-environment.git
$ run.sh

artifact
$ python3 main.py -p Bower -b 1 -t checkout -v fixed -o output

coverage
$ python3 main.py ... -t test -v fixed-only-test-change
Usage of BugsJS

**download**

```bash
$ git clone https://github.com/BugsJS/bug-dataset.git
```

**run**

```bash
$ git clone https://github.com/BugsJS/docker-environment.git
$ run.sh
```

**artifact**

```bash
$ python3 main.py -p Bower -b 1 -t checkout -v fixed -o output
```

**coverage**

```bash
$ python3 main.py ... -t test -v fixed-only-test-change
```
Usage of BugsJS

download

$ git clone https://github.com/BugsJS/bug-dataset.git

run

$ git clone https://github.com/BugsJS/docker-environment.git
$ run.sh

artifact

$ python3 main.py -p Bower -b 1 -t checkout -v fixed -o output

coverage

$ python3 main.py ... -t test -v fixed-only-test-change
Usage of BugsJS

download

$ git clone https://github.com/BugsJS/bug-dataset.git

run

$ git clone https://github.com/BugsJS/docker-environment.git
$ run.sh

artifact

$ python3 main.py -p Bower -b 1 -t checkout -v fixed -o output

coverage

$ python3 main.py ... -t test -v fixed-only-test-change
Before Cleaning

<table>
<thead>
<tr>
<th>Production</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>3,685</td>
<td>48</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
</tr>
</tbody>
</table>

After Cleaning (No wp & comments)

<table>
<thead>
<tr>
<th>Production</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>3,632</td>
<td>41</td>
</tr>
<tr>
<td>17</td>
<td>5</td>
</tr>
</tbody>
</table>

**Code Churn**
Hanam et al., FSE 16

bug patterns

9% bugs

BugsJS

83% Uninitialized variables
dereferenced non-values
Pan et al., EMSE 2009

bug-fix patterns

88% fixes

3 novel JS-related

*Bug fixes patterns*

- changing a return statement
- initializing a variable with `empty` literal
- declaring an existing variable
Fault Localization w/ BugsJS

Test results
- Test Commands
- Per-test coverage
  - Utility Framework

Suspiciousness
1. functionAA
2. functionBB
3. functionCC
4. functionDD
5. functionEE

Location of the bugs
- Cleaned Patches

Poster Session
A BENCHMARK OF JAVASCRIPT BUGS

Organization
https://bugsjs.github.io/

Bug Repository
Subjects
Utility Framework | Shields | Express | Hessian
Bug Statistics | Eslint | Karma | Hexo
Test Commands | Mongoose | Bower | PencilBlue | Node-Redis
Bug Report Data

Fault Localization w/ BugsJS

Test results
Test Commands
1 function_f
2 function_g
3 function_h
4 function_i
5 function_j

Per-test coverage
Utility Framework

Location of the bugs
Cleaned Patches

Poster Session

Docker Image
Pre-built Environment