UNIVERSITÄT BERN

How do code documentation efforts spread over class hierarchy?

Mattia Pedrazzi

Seminar Software Engineering, FS2022

Supervised by Pooja Rani
Software Engineering Group (SEG)

Motivation

To extend / use the code base, developers need

- Enough Documentation
- Good Documentation Quality
- Well Spread Documentation
  - Interfaces
  - Uses
  - Superclasses

Motivation

To provide good documentation, the project maintainers need to know

- Which are the important classes?
- Which classes need documentation?

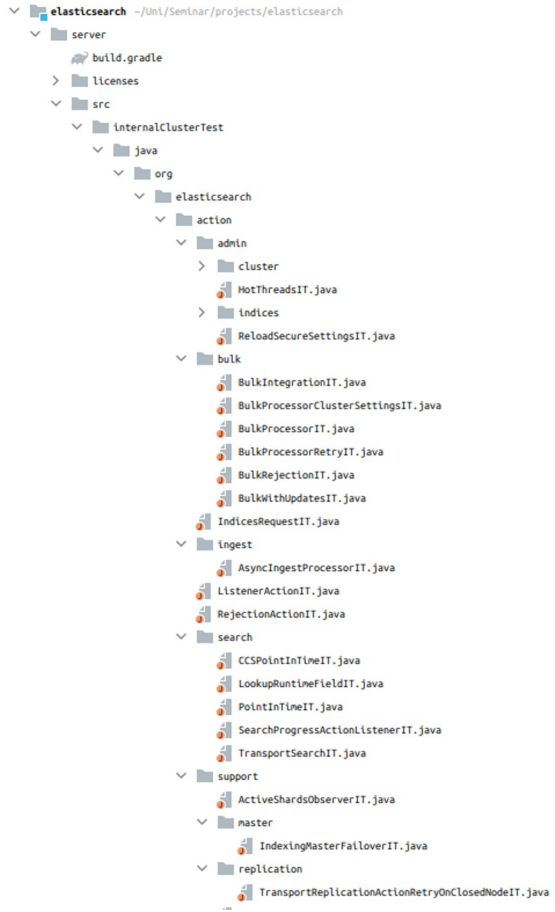
→ High fan-in/out, inherited often

Motivation

Example

Elasticsearch

- 19’931 Classes
- 1’171 Interfaces
- 14’288 Inheriting Entities
- 1’218 Inherited Entities



The Problem

```java
/**
 * Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam
 * nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam
 * erat, sed diam voluptua. At vero eos et accusam et justo duo dolores
 * et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem
 * ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr,
 * sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,
 * sed diam voluptua. At vero eos et accusam et justo duo dolores et ea
 * rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem
 * ipsum dolor sit amet.
 */
1 usage  1 inheritor
public class SomeSuperClass {

    /**
     * Lorem ipsum dolor sit amet, consetetur sadipscing elitr,
     * sed diam nonumy eirmod tempor invidunt ut labore et dolore
     * magna aliquyam erat, sed diam voluptua. At vero eos et accusam
     * et justo duo dolores et ea rebum. Stet clita kasd gubergren, no
     * sea takimata sanctus est Lorem ipsum dolor sit amet.
     */
    1 override
    public void someMethod() {
        // ...
    }
}
```

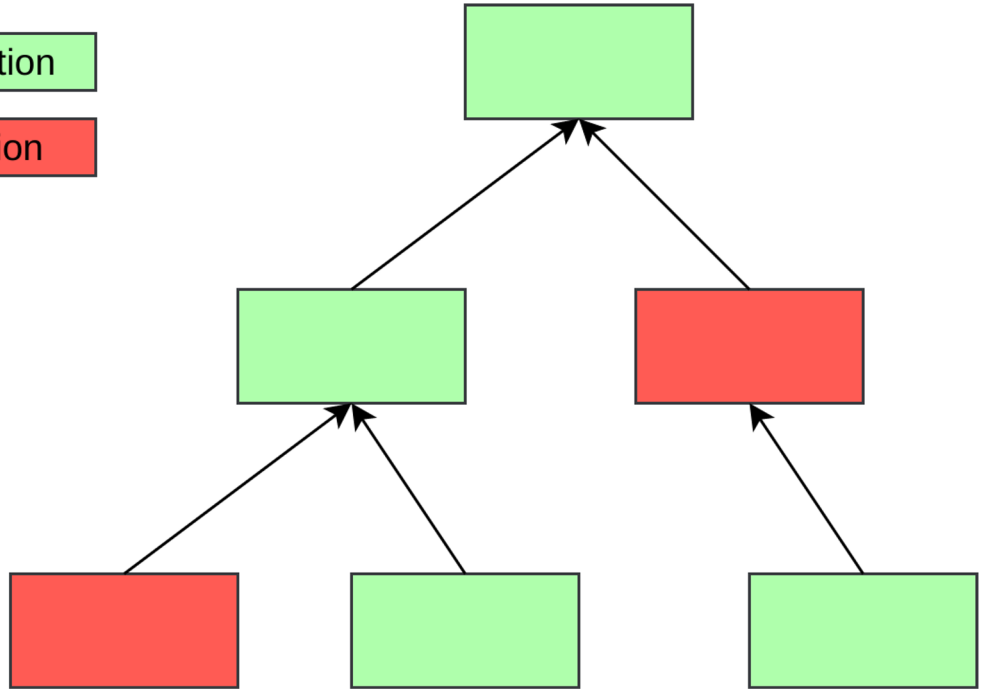
```java
/**
 * Lorem ipsum dolor sit amet, consetetur sadipscing
 * elitr, sed diam
 */
public class SomeChildClass extends SomeSuperClass {

    public void someMethod() {
        // ..
    }

    public void someOtherMethod() {
        // ..
    }
}
```

The Problem

Where is the documentation lacking?

Good Documentation

Bad Documentation



The Goal

- Extract hierarchy and documentation data
- Provide useful visualization
- Identify weak and strong documentation points (Quantity)

Related Work

- Inheritance depth plays a role in the understand of oo software
- six to four levels of inheritance is where problems start

Daly J., Brooks A., Miller J., Roper M., Wood M., “Evaluating Inheritance Depth on the Maintainability of Object-Oriented Software”,Empirical Software Engineering, 1, pp. 109-132, Kluwer Academic Publishers, Boston, 1996

Related Work

Role and Features of Documentation

- Documentation reuse improves productivity
- Documentation inheritance
- Information hiding

Sametinger J. “Reuse Documentation and Documentation Reuse”, Dept. of Computer Science, Texas A&M University, Texas, U.S.A, 1996

Related Work

Metrics

$$\text{ANYJ} = \frac{\text{declarations with any Javadoc comment}}{\text{total number of declarations}} \quad (1)$$

$$\text{DIR} = \frac{\text{documented items}}{\text{documentable items}} \quad (2)$$

$$\text{ANYC}_{\text{method}} = \frac{\text{methods with any kind of comment}}{\text{methods}} \quad (3)$$

$$\text{WJPD} = \frac{\text{words in Javadoc of declarations}}{\text{declarations}} \quad (4)$$

Schreck, D., Dallmeier, V., Zimmermann, T. “How Documentation Evolves Over Time”, Dept. of Computer Science, Saarland University, Saarbrücken, Germany, 2007

Related Work

Eclipse Evaluation

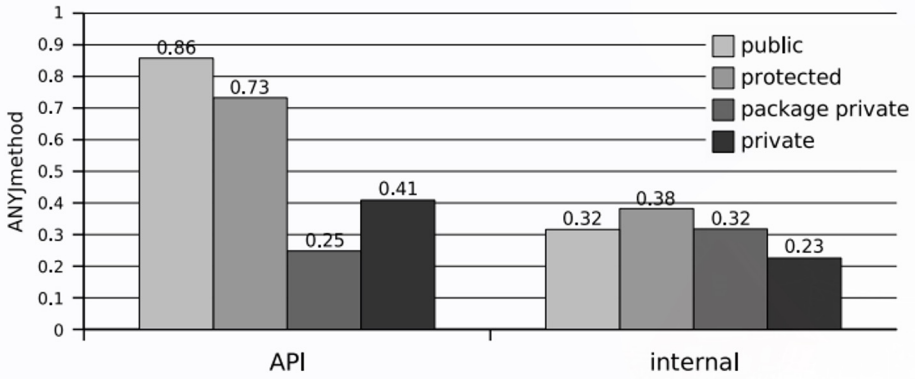
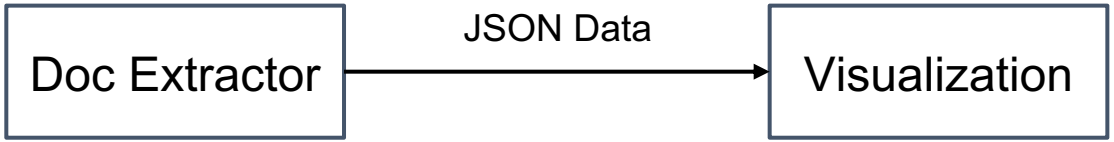


Figure 1: Public methods are documented more frequently ($\mathbf{ANYJ_{method}}$)

Schreck, D., Dallmeier, V., Zimmermann, T. “How Documentation Evolves Over Time”, Dept. of Computer Science, Saarland University, Saarbrücken, Germany, 2007

Our Approach

- Java Doc Extractor (golang)
- Visualize uses and relations with Roassal (Pharo)



Java Doc Extractor Repository: https://github.com/UsernameN0tAvailable/java_doc_extractor

Our Approach

- Projects Analysed
  - Guava (google)
  - Guice (google)
  - Hadoop (Apache)
  - Framework (vaadin)
  - CDT (Eclipse)
  - ElasticSearch
  - P2 Exercise

Results

Extractor Output

```
▼ projectInfos:
    numberOfClasses:                3384
    numberOfInterfaces:             239
    numberOfAnnotations:            69
    numberOfRecords:                0
    numberOfEnums:                  203
    totNumberOfEntities:            3895
    numberOfEntitiesWithSuperClass: 2238
    numberOfEntitesWithChildren:    589
    anyj:                           0.044077136
    dir:                            0.054314267
    anyc:                           0.23980546
    wjpd:                           3.0578172
▼ entities:
  ▶ 0:                              {…}
  ▶ 1:                              {…}
  ▶ 2:                              {…}
  ▶ 3:                              {…}
```

```
documentation: "/**\n * Static utility methods pertaining to {@code float} primitives, that are not already found in\n * either {@link Float} or {@link Arrays}.\n *\n * <p>See the Guava User Guide article on <a\n * href=\"https://github.com/google/guava/wiki/PrimitivesExplained\">primitive utilities</a>.\n *\n * @author Kevin Bourrillion\n * @since 1.0\n */"
name: "com.google.common.primitives.Floats"
extends: "com.google.common.primitives.FloatsMethodsForWeb"
Methods:
  0: {…}
  1: {…}
  2: {…}
  3: {…}
  4: {…}
  5: {…}
  6: {…}
  7:
    signature: "@Beta\n  public static Converter<String, Float> stringConverter()"
    documentation: "/**\n   * Returns a serializable converter object that converts between strings and floats using {@link\n   * Float#valueOf} and {@link Float#toString()}.\n   *\n   * @since 16.0\n   */"
      line: 319
      code: "  @Beta\n  public static… padding) : array;\n  }"
  8: {…}
  9: {…}
  10: {…}
  11: {…}
interfaces: []
path: "/guava/guava/src/com/google/common/primitives/Floats.java"
type: "class"
isTest: false
testClasses:
  0: "com.google.common.primitives.FloatArrayAsListTest"
  1: "com.google.common.primitives.FloatsTest"
subClasses: []
implementedBy: []
uses:
  0: "com.google.common.annotations.Beta"
  1: "com.google.common.annotations.GwtCompatible"
  2: "com.google.common.base.Converter"
  3: "com.google.common.primit…ypesAreNonnullByDefault"
  4: "com.google.common.primitives.Floats.FloatConverter"
usedBy:
  0: "com.google.common.primit…Floats.FloatArrayAsList"
isPrivate: false
InnerClasses:
  0: "com.google.common.primitives.Floats.FloatConverter"
  1: "com.google.common.primit…xicographicalComparator"
  2: "com.google.common.primit…Floats.FloatArrayAsList"
anyj: 0.6666667
dir: 0.72
anyc: 1
wjpd: 54.083332
```

Results

- Visualization Demo

Results

	Guice	Guava	Hadoop	Elasticsearch	Framework	CDT
ANYJ	0.021	0.044	0.157	0.043	0.293	0.126
DIR	0.026	0.054	0.171	0.045	0.340	0.137
ANYC	0.232	0.239	0.336	0.162	0.394	0.320
WJPD	1.031	3.058	1.587	1.596	10.229	3.574

Conclusion

- Is the visualization useful?
- User evaluation needed
- Additional metrics needed