The Evolution of Exceptions

SCG Seminar – Jakob Schaerer
Exceptions

FileReader fr = new FileReader("foo.in");
Exceptions

FileReader fr = new FileReader("foo.in");
Exceptions

```java
FileReader fr;
try {
    fr = new FileReader("foo.in");
} catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```
Evolution of Exceptions
Evolution of Exceptions

As a project evolves:

• Do developers introduce more user-defined exceptions?
• Do developers catch more specific exceptions?
• Do developers change the exception handling (catch block)?

Are the answers to these questions the same for different projects?
Objects of Interest

Exception Class

```java
public class PresentationException extends Exception {
}
```
Objects of Interest

Try-Catch Block

```java
try {
}
} catch (Exception e) {
}
```
Objects of Interest

Throw Statement

```java
throw new Exception();
```
Objects of Interest

Throws declaration

```java
public void sample() throws IOException {
}
```
Dataset

Java Projects from github:

Solr

elastic

and counting ...
Overview
Outlook

• Crunch the numbers
• Parse more projects
Questions?
Mu: number of methods that throw user-defined exceptions only
Md: the methods that throw java default exceptions only
Mb: number of the methods that throw both

Ru: The number of “throw" blocks that throw user-defined exceptions
Rd: The number of “throw" blocks that throw java default exceptions

Tu: The number of user-defined exceptions in “throws” blocks
Td: The number of java default exceptions in “throws” blocks

Cu: The number of user-defined exceptions in “catch” blocks
Cd: The number of java default exceptions in “catch” blocks

Uc: The number of user-defined checked exceptions
Uu: The number of user-defined unchecked exceptions

LOC: the number of lines of code