Offensive Programming

@ErraticWelshie

What is Defensive Programming?

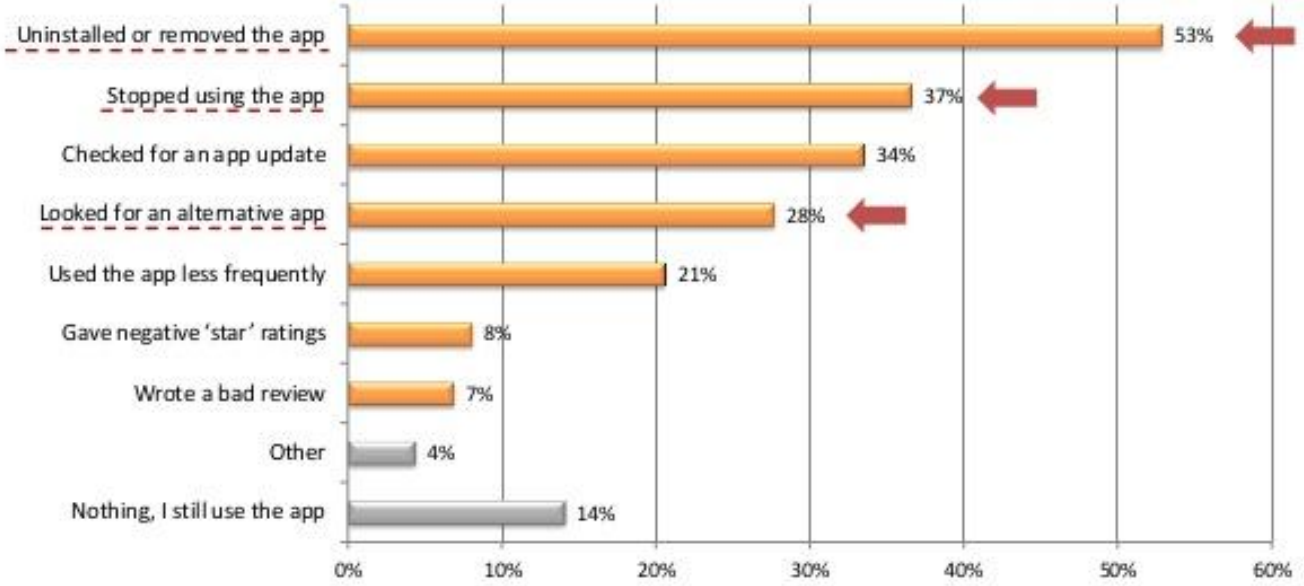
“Defensive programming is a form of defensive design intended to ensure the continuing function of a piece of software under unforeseen circumstances.

Defensive programming techniques are used especially when a piece of software could be misused.”

Severe App Issues Drive Users Away

What have you done when you experienced a mobile app that regularly crashed, stopped responding or had errors?

Select all that apply.



Pointless Code

```
public class Person {
    private String name = null;
    public void setName(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
}
```

```
public Person newPerson(String name) {
    Person person = new Person();
    if (name != null) {
        person.setName(name);
    }
    return person;
}
```

Null Checks

```
if (person != null) {
        List<Friend> friends = person.getFriends();
        if (friends != null) {
            for (Friend friend : friends) {
                if (friend != null) {
                    if (friend.isFavorite()) {
                        favoriteFriends.add(friend);
                    }
                    ...
```

Default Cases

```
switch(viewType) {
    case 0:
        ...
        break;
    default:
        ....
        break;
```

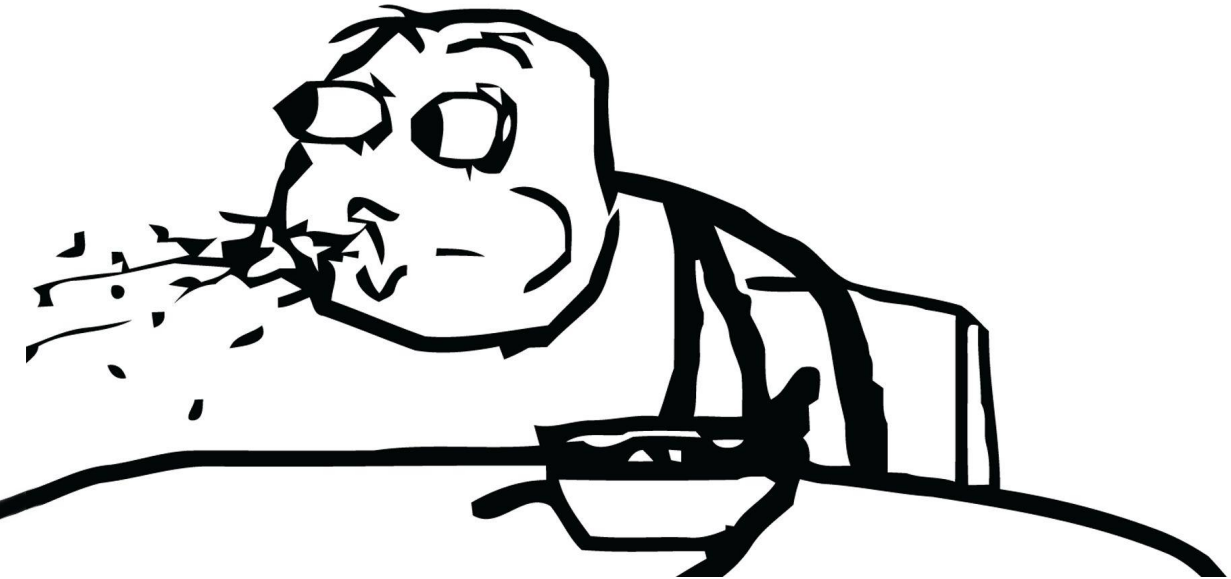
Global Exception Handler

```
Thread.setDefaultUncaughtExceptionHandler(new Thread.UncaughtExceptionHandler() {
        @Override
        public void uncaughtException(Thread thread, Throwable throwable) {
            Crashlytics.logException(throwable);
        }
    });
```

What is Offensive Programming?

Crash Fast!

Fix early...



offensive-example.java

Raw

offensive-example.java

```java
public String badlyImplementedGetData(String urlAsString) {
    URL url = null;
    try {
        url = new URL(urlAsString);
    } catch (MalformedURLException e) {
        logger.error("Malformed URL", e);
    }

    HttpURLConnection connection = null;
    try {
        connection = (HttpURLConnection) url.openConnection();
    } catch (IOException e) {
        logger.error("Could not connect to " + url, e);
    }

    StringBuilder builder = new StringBuilder();
    try (BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()))) {
        String line;
        while ((line = reader.readLine()) != null) {
            builder.append(line);
        }
    } catch (Exception e) {
        logger.error("Failed to read data from " + url, e);
    }
    return builder.toString();
}
```

Offensive!

```
public String getData(String url) throws IOException {
        HttpURLConnection connection = (HttpURLConnection) new URL(url).openConnection();
        StringBuilder builder = new StringBuilder();
        try (BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()))) {
                String line;
                while ((line = reader.readLine()) != null) {
                        builder.append(line);
                }
        ....
```

Offensive Programming: Crash Faster!

- Do not ignore or hide problems
- Assert preconditions
- Throw exceptions
- Fail quickly
- Fix and repeat

Crash Reporting

- Crashlytics
- ACRA
- Bugsnag
- Google

Testing

Testing & Automation

- Unit testing
- Integration tests
- Smoke testing
- Dogfooding
- Staged rollout

Conclusion

- Fail fast
- Fix quicker
- Profit.