License Check + GDPR Check

The usage of wrong licenses can cause a huge damage to an app project but is still an often seen situation as programmers tend to grab libraries that help them solve a task without checking its license. Violation of GDPR can lead to high fees and removal from the App store. Our legal department takes care of the following:

- Are viral licenses used
- Are license requirements satisfied
- Are necessary license information provided to the user
- What kind of data is processed and is the GDPR correct

Manual Test

Our destructive manual tester will torture your app in a way that it will crash to make it stop. As programmers are a constructive type of people, they make bad testers. Our tester identifies problems that your users might step on but are not always found in the code review. This process is a complementary part of code analysis. We perform:

- Developing and executing test cases
- Testing on multiple devices
- Testing of different scenarios
- Review user comments and reproduce bugs users report

Automated Testing aka Code Coverage

Automated tests are run after every build to ensure that changes to code and app do not cause errors. Often solving one bug can introduce another especially when changing functionality used in many areas of the app. We inspect:

- Are Unit tests existing and how is the code coverage
- Are UI tests existing and how much of the app is tested
- Which framework is used for the UI tests
- How is the quality of the tests

Documentation

The documentation of the app project should leave no questions unanswered. It should be possible to hand over the project to another team which will get all necessary information to continue work. The architecture needs to be documented, the build process and variants, the code should be documented, there should be a document describing the app and its functionalities. In detail:

- Is the code documented according to JavaDoc, KDoc or Swift-DocC rules
- Does the formal architecture document exists
- Is there a readme file that documents the build process, dependencies and deployment
- Is there a document describing the app's functionalities and features

Build tools

A well set-up app project should build and test with a single click. A well set-up project should automatically build and test all components at every code merge. If building and testing of the app is a manual labor taking time and effort, it needs to be optimized. Therefore we check:

- CI/CD pipeline configuration
- Automatic testing
- Automatic versioning

Security Analysis

Usage of outrun and no more safe crypto algorithms, the usage of unsafe protocols or saving passwords as clear text in the database or the source code are only a few reasons an application's security can be violated and sensitive user data stolen. Therefore we check:

- Used algorithms
- Protocol safety
- Right use of cryptography
- Storage of sensitive data

UX / Usability analysis

Great UX can make great apps but bad UX will definitely end up with a bad app and unhappy customers. Our UX expert will inspect the flow of all UI components for usability and accessibility. In detail:

- Accessibility
- Modern paradigms
- Usability

Process evaluation

A great development process will help avoid bugs. Missing code reviews, non-existent git flow, bad traceability, and unclear processes will result in low quality and unstable apps. We analyze not only the development process itself but also the used git flow, and team metrics to reveal optimization potentials. Areas we cover:

- Traceability: Why a change was done, by whom and who ordered it
- Git flow: Branching, commit messages, code reviews
- Team metrics: Burn rate, output of each team member
- Tools: are the right tools for the job used