**SAST: Сложность выбора.**

Когда число разработчиков в компании начинает исчисляться сотнями, а количество строк написанного ими кода – миллионами, тестирование такого ПО становится весьма нетривиальной задачей. Вместе с ростом числа программистов, растёт и отдел тестирования ПО, множатся тесты, применяются новые методики. Но даже код, покрытие которого близко к 100% может, и как правило, содержит в себе ещё массу ошибок, отловить которые не под силу ни разработчику, ни тестировщику. Связано это с тем, что автотесты и функциональное тестирование не могут выявить большое количество специфических проблем, таких как, например, взаимная блокировка потоков или состояние гонки, связанных с ошибками проектирования многопоточных систем. Поэтому на борьбу с багами выходят относительно новые типы ПО под общим названием SAST – системы статического анализа исходного кода. Сложность этих систем достаточно высока, учитывая их специфику. Из этого вытекает их высокая стоимость. Кроме того, интеграция такой системы в цикл разработки дело отнюдь непростое, поэтому от её выбора напрямую зависит, вырастет ли эффективность работы технических отделов, или же наоборот она станет ненужной обузой или дорогой игрушкой.

**SAST: Choosing problem.**

When count of developers in company becomes to hundreds and count of lines in their sources are millions testing becomes a big part of application lifecycle. QA should create some new tests and implement new testing techniques. But even if code coverage near the 100% it’s not means that this code hasn’t some issues which can’t be found using unit tests or manually by QA staff. There is some specific error can be existing in sources such as deadlocks or race conditions related to mistakes made in the design of multi-threaded applications. One of the tools which can helps QA and programmers in their bug-fighting is a SAST – Static Application Security Testing. It’s a very difficult and complicated systems which price not too low. Also adding such system as a part of CI not a trivial task. So right SAST can increase productivity of QA teams several times but bad SAST may do it many times lower.