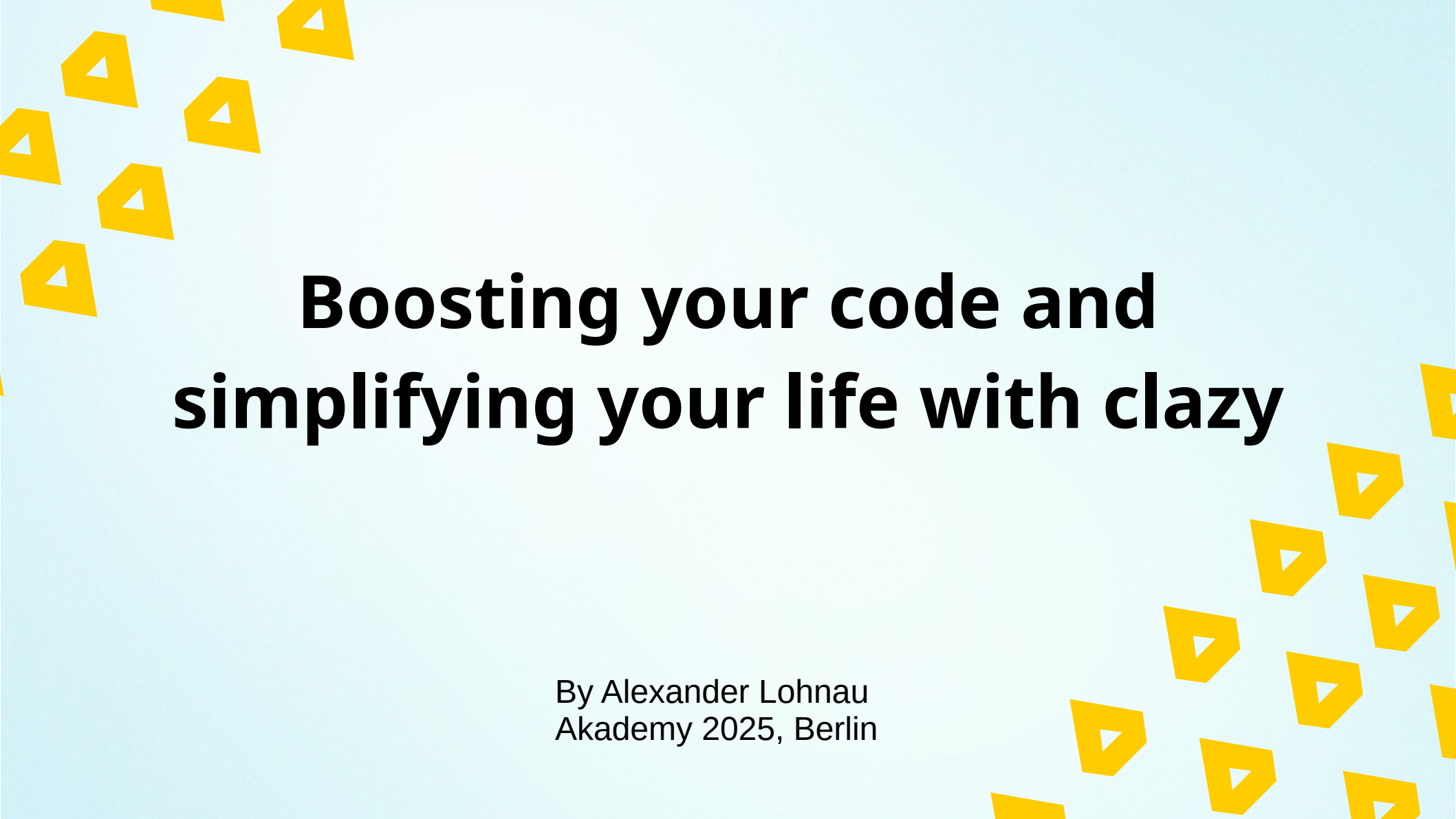


The background is a light blue gradient. It is decorated with numerous yellow triangles, each containing a white right-pointing triangle inside. These triangles are arranged in two main clusters: one in the top-left corner and another in the bottom-right corner, both appearing to radiate from the corners.

Boosting your code and simplifying your life with crazy

By Alexander Lohnau
Akademy 2025, Berlin

What is Clazy?

- A Clang-based static analysis tool for C++
- Developed primarily for Qt code
- Provides compile-time checks to catch common performance and correctness issues
- Runs as a Clang plugin or via command-line interface



What is Clazy?

- Checks are organized in levels: level0, level1, level2 and manual level
- Indication for how critical the issue and how reliable the check is
- Some checks provide replacement snippets called “fixits”



My history with Clazy

- Initially started playing as part of dev process
- Had to compile from source
 - Broken Neon packages with incompatible ABI ;)
- First real contact with clazy development



My history with Clazy

- First patch in December 2023 improving docs
 - ... and a few days later fixing a false positive
- Snippet from plasma-workspace:
- `QRegularExpression(QString("Panel %1$").arg(panelId()))`



Starting the Dev-Streak

- First steps to compile and run tests were rather hard, no checks on CI other than compilation
- Mission: Make it easier to get started



Starting the Dev-Streak

- Improve test setup, run test via “ctest” without install
 - Parallel execution makes CI blazing fast 
- Ensure proper formatting and freshness of generated files
- Better safety-net for contributors



Starting the Dev-Streak

- Many checks did not work with Qt6
- ... or C++ 17 due to changed syntax tree
- Version 1.12 with full Qt6 support, first release since 2.5 years



Maintaining & Improving

- Lots of check could use improvements and fixes for false-positives
- Used KDE codebase to look for real-world cases 🕶️
- Focus on providing new fixits to existing checks



Maintaining & Improving

- Example “qcolor-from-literal” fixits
- `QColor("#001020") => QColor(0x001020)`
- `QColor("#00112233") => QColor(0x11, 0x22, 0x33, 0)`
- Invalid string-patterns are also reported!



Integrating & running crazy

- Two main ways:
 - As a clang plugin during compilation
 - Using a crazy-standalone executable



Integrating & running crazy

- Clang-plugin is wrapped in “crazy” shell-script
- Can be used as normal compiler:
 - `-DCMAKE_CXX_COMPILER=crazy`
 - `export CXX=crazy`



Integrating & running crazy

- crazy-standalone does same checks, but does not compile code
- Utilizes compile_commands.json file
- Example:
 - `crazy-standalone -p build src/**/*.cpp`



Integrating & running crazy

- Checks are configured using “CLAZY_CHECKS” environment variable
- Combination of levels and individual check names
- level0,level1,used-qunused-variable,no-qproperty-without-notify



clang-tidy, if all
people only run crazy



Clazy & Clang-Tidy

- Clang ecosystem has its own linter: clang-tidy
- Offers performance, modernization and bug checks
- Operates like clazy-standalone on compile commands file



Clazy & Clang-Tidy

- Clang-Tidy supports external plugins
 - KDE loooves plugins ;)
- API is far more restrictive than compiler plugin/standalone



Clazy & Clang-Tidy

- Refactorings proceeded plugin integration, 9 pull requests in total
- Improving flexibility of internal API
- Check registration is more restricted in clang-tidy
 - No concept of check-levels, only wildcards



Clazy & Clang-Tidy

- Clang-Tidy command needs plugin file
 - -load=ClazyClangTidy.so
 - If not in library path, absolute path to plugin



Clazy & Clang-Tidy

- All checks of clazy plugin have “clazy-” prefix
- Example command line flag to enable some:
 - `-checks=clazy-qdatetime-utc,clazy-qgetenv`



Clazy & Clang-Tidy

- Instead of levels, define check list as ENV variable
 - <https://alex1701c.github.io/2025/07/27/clazy-clangtidy.html>
- Enable/disable checks after level is specified
 - `-checks="$CLAZY_LEVEL0,-clazy-use-static-qregexexpression"`



Clazy & Clang-Tidy

- “run-clang-tidy” executable allow parallel execution 
- Warnings outside of source file are hidden by default
 - Specify header-filter to view all
- Use full power of NOLINT commands – supported by other modes of clazy too 



A light blue background with a pattern of yellow triangles. The triangles are arranged in a grid-like fashion, with some triangles pointing up and some pointing down. They are located in the top-left and bottom-right corners of the image.

Thank you & Happy Hacking

Read more on my blog:
<https://alex1701c.github.io/>