Improving Developer Experience at LibreOffice

Hossein Nourikhah
Contents

- What is “Developer Experience” (DX)?
- Efforts to improve LibreOffice DX
  - Culture
  - Tools
  - Processes
- The road ahead
Developer Experience (DX)

- What is the Developer Experience (DX)?
  - Overall experience of developers toward a goal
    - Fixing bugs, implementing features & other changes
    - It’s not only about tools, but also culture and processes

- Main Tasks to think about
  - Build
  - Debug
  - Test
  - Run/Deploy
Culture

- Friendly culture
  - Accepting and helping newcomers
  - Developer community: Mix of volunteers and employees
  - Balance between individuals and companies
- Interviews and calls with the newcomers
- Regular meetings for important decisions
  - ESC Call
- Annual conference (LibOCon)
Tools

- Continuous Integration (CI) ➔ Jenkins
- Support multiple IDEs
- GUI Design tool ➔ Glade
- Static code checking ➔ Coverity scan
- Code review ➔ Gerrit
- Code search ➔ OpenGrok
- Bug tracking ➔ Bugzilla
- Translation ➔ Weblate
- Code coverage ➔ lcov
- Fuzzers ➔ libfuzzer
Processes

- **Mentoring**: Mentoring is provided for the newcomers
- **EasyHacks**: Simple tasks to get started in LibreOffice development
- **Bug bounty**: Security bugs ➔ Intigriti
- **Tenders**: More complicated tasks to be done by a company or an individual
- **Engineering Steering Committee (ESC)**: Important decisions is taken here
Tools and Processes
**Build tools**

- **gbuild**
  - Old system was **dmake**, a custom old build tool
  - **gbuild** is the current build system of LibreOffice, based on GNU make
  - Works on Linux, macOS, BSDs and Windows (cygwin)
  - In conjunction with **LODE** (LibreOffice Development Environment) it can automate most of the build tasks
  - LibreOffice can now be built using Visual Studio 2022
Continuous Integration

- CI with Jenkins
- Build for different platforms
  - Linux, Windows, macOS, Android
- Unit tests
- CppUnitTests on every platform
- UITests on Linux
GUI Design Tools

- **GUI Designer**
  - Every GUI is created visually using Glade interface designer.
  - The result is usable with every UI backend that VCL supports.
Using IDEs

- Many IDEs are supported
  - Visual Studio
  - VS Code
  - Qt Creator
  - XCode
- Debugging
- Code Completion
Mentoring

- We have dedicated mentors
- We provide help via chats, calls and screen sharing (if needed) with the newcomers to provide help for getting started
- We review the manuals and instructions to make keep them correct, understandable and easy to use
EasyHacks

- Simple tasks for the newcomers
- Different difficulty levels
- Different languages and skills
- Designed to help newcomers to grow
IRC and Mailing Lists

- **Rooms**
  - [#libreoffice]: User support / discussion
  - [#libreoffice-dev]: Development discussion
  - [#libreoffice-design]: Design discussion
  - [#libreoffice-qa]: Quality assurance
- **Bridged to Matrix**
  - LibreOffice Space
  - https://matrix.to/#/#libreoffice-space:matrix.org
- **LibreOffice development mailing list**
Video Tutorials

- Getting started to LibreOffice development
- And many more!
• Wiki
  • Main source of information related to development
  • Build instructions
  • Updated regularly to reflect the recent changes
  • Translated

Get Involved

There are multiple and various ways in which you can join and help the LibreOffice project.

• Are you a developer? You can help out with development.
• Do you want to propose a new feature or report a bug? Report it at our bugzilla instance!
• Are you a copywriter or translator?
  • The wiki needs you
  • Translate LibreOffice
  • Translate Ask-bot at transfex.com
• Are you trying to help developers? The QA (Quality Assurance) team needs you.
• Are you a (UX) designer? The design team would love to hear from you.
• Help users answering their questions at ASK.Libreoffice and the localized version (put a /LANGUAGECODE at the end of the url)
Code Review

- Code Review
- Done via Gerrit
- Each and every submission goes through gerrit
- Improves code quality & helps newcomers
New APIs and platforms

• LibreOffice Kit
  • New API for LibreOffice using tile rendering
  • Easy to use API with no dependencies on LO libraries
  • Easier to use compared to UNO API

• WASM
  • Web assembly, allows to run LO in a browser

• ScriptForge
  • New API for automation using Basic and Python
New Build Tools

• The **gbuild** will continue to be in use
• There is no plan to change the main build tools
• There are preliminary works on adding other build tools
  • **meson**: Preliminary works shows the feasibility of using it for the core
  • **cmake**: Used successfully for the SDK examples, no plan for using it in the LibreOffice core
New Developer Documentation

- Java LibreOffice development (JLOP)
- Author: Dr. Andrew Davison
- Focused on UNO API
- Recently released as CC BY-SA 4.0
- Good platform for adding code snippets for other languages
  - Python, BASIC
  - C++ (probably)
Thanks

hossein@libreoffice.org

DX?