



Guademy 2008

aleixpol@gmail.com



# Developing on GNU

Aleix Pol i Gonzàlez

25th April 2008





- 1 Introduction
  - The Problem
- 2 What to do?
  - Language
  - Buildtool
  - Debuggers
- 3 Conclusions



- 1 Introduction
  - The Problem
- 2 What to do?
  - Language
  - Buildtool
  - Debuggers
- 3 Conclusions



- Aleix Pol González
- I study Computer Science at FIB/UPC
- I collaborate in KDE by developing KAlgebra (KDE-Edu) and KDevelop



## We want an Integrated Development Environment

- We want a program that understands what we are working on and helps us with this knowledge
- Not a program for people that does not know how to develop software



# The Problem



## Playground

- Huge developer base
- Huge code base
- Programs not meant to communicate to other programs (unidirectional)
- It is very hard to get it reliable



## What do we want to integrate?

- Buildtool - to know about the whole project
- Language - is what we are writing on
- Debugger - we will use it (probably :)



- 1 Introduction
  - The Problem
- 2 What to do?
  - Language
  - Buildtool
  - Debuggers
- 3 Conclusions





## How to support C++?

- Not possible to integrate g++
- Created a preprocessor
- Created a parser from scratch
- Some effort duplication



## Providing the necessary information

- We only integrate CMake as the Makefile representation
- We created a parser to get to know the system information
- It was not possible to use CMake as a library because of Kitware policies
- Some effort duplication



### Actually supporting a tool

- GDB/MI is a line based machine oriented text interface to GDB
- GDB integration needs speed, less commands to be executed



- 1 Introduction
  - The Problem
- 2 What to do?
  - Language
  - Buildtool
  - Debuggers
- 3 Conclusions



## Present and future

- Benefit from the KDevelop3 experience
- Reliable
- The less configuration possible
- Do not support if we cannot fully support



## Way to go

- We are duplicating efforts
- We are doing same/similar work many times
- If we don't duplicate code, we will have a more reliable support
- If we don't duplicate execution, we will have a much better user experience
- If we partially support a feature, nobody really knows what can we do with it



Any questions?

