

Open-Sourcing a Major Commercial Technology

Bertrand Meyer
Till G. Bay
Bernd Schoeller

ETH Zurich, Switzerland

also known as

How to dump 2.300.000 lines of “industrial strength” code onto undergrad students (a lessons learned session)

Bertrand Meyer

Till G. Bay

Bernd Schoeller

ETH Zurich, Switzerland

Overview

1. What is EiffelStudio?
2. Why did we open source it?
3. How did we open source it?
4. What can we learn?

Eiffel

- Object-Oriented Programming Language
 - 1986: First compiler
 - 1991: Full language definition
 - 2005: ECMA 367
 - 2006: ISO 25436
- Bondage and discipline
- Design by Contract

Industrial applications

- **Axa Rosenberg:**
Investment manager
managing more than
US\$40 billion,
2m LoC
- **Chicago Board of Trade:**
Eiffel + Solaris + Windows
+ CORBA + ...
- **Boeing:**
Large scale simulations
- **Swedish social security:**
accident reporting and
management



EiffelStudio

- Developed by Eiffel Software, Santa Barbara
- Compiler for Eiffel language
- Modern IDE
- Graphical UML editor with full round-trip engineering
- Refactoring support
- Highly portable:
 - Windows (32 and 64 bit)
 - Linux (32 and 64 bit)
 - FreeBSD
 - Solaris
 - Mac OS X
 - VMS
 - Embedded devices

EiffelStudio

The screenshot displays the EiffelStudio IDE interface. The main window shows the source code for the `ROOT_CLASS` in the `root_cluster` project. The code includes an `indexing` block with a description and note, a `class` declaration for `ROOT_CLASS`, a `create` block for `make`, and a `feature` block for `Initialization`.

```
indexing
description : "System's root class"
note       : "Initial version automatically generated"

class
  ROOT_CLASS

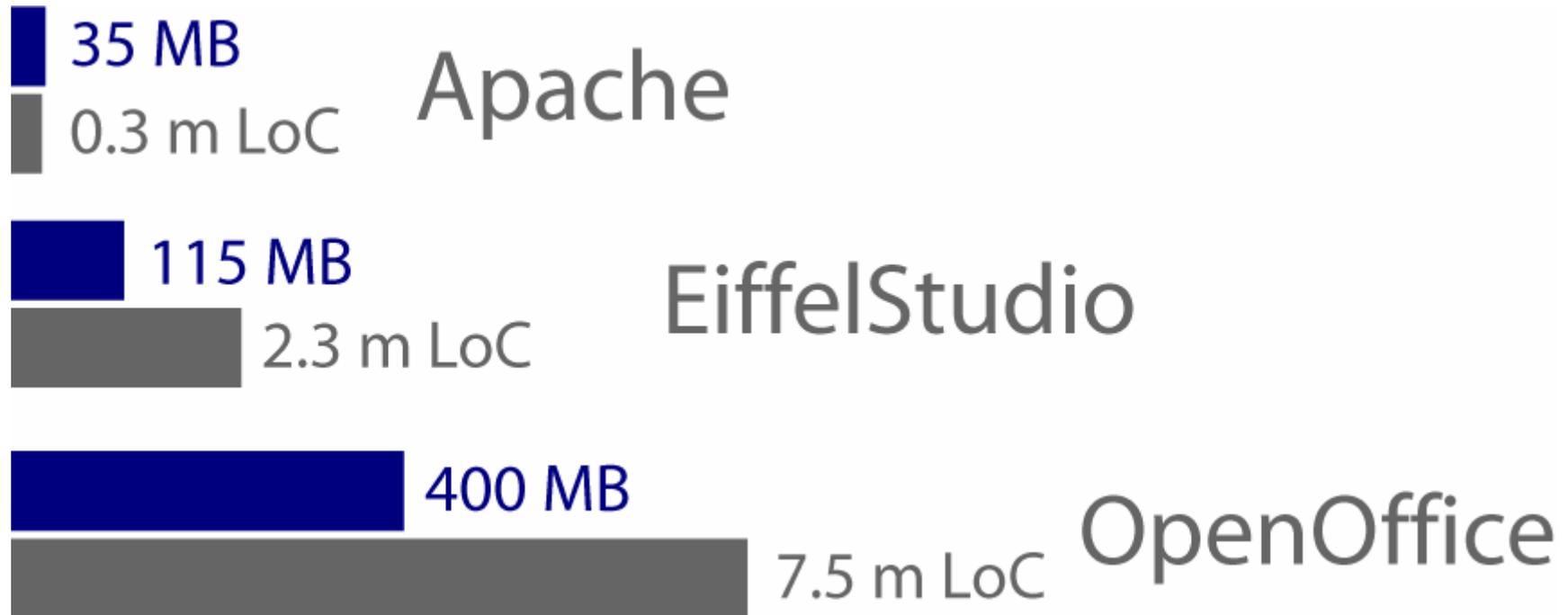
create
  make

feature -- Initialization
```

The UML diagram in the Context window shows a class hierarchy. The `{root} ROOT_CLASS` class has a `- <<Initialization>> + make` feature. It is inherited by the `{reused} ANY` class, which has several features: `+2 <<Access>>`, `+2 <<Status report>>`, `+5 <<Comparison>>`, `+6 <<Duplication>>`, `+3 <<Output>>`, `+1 <<Platform>>`, `+2 <<Basic operations>>`, `+3 <<Duplication>>`, `+1 <<Retrieval>>`, `+1 <<Output>>`, `+1 <<Initialization>>`, and `+2 <<Basic operations>>`.

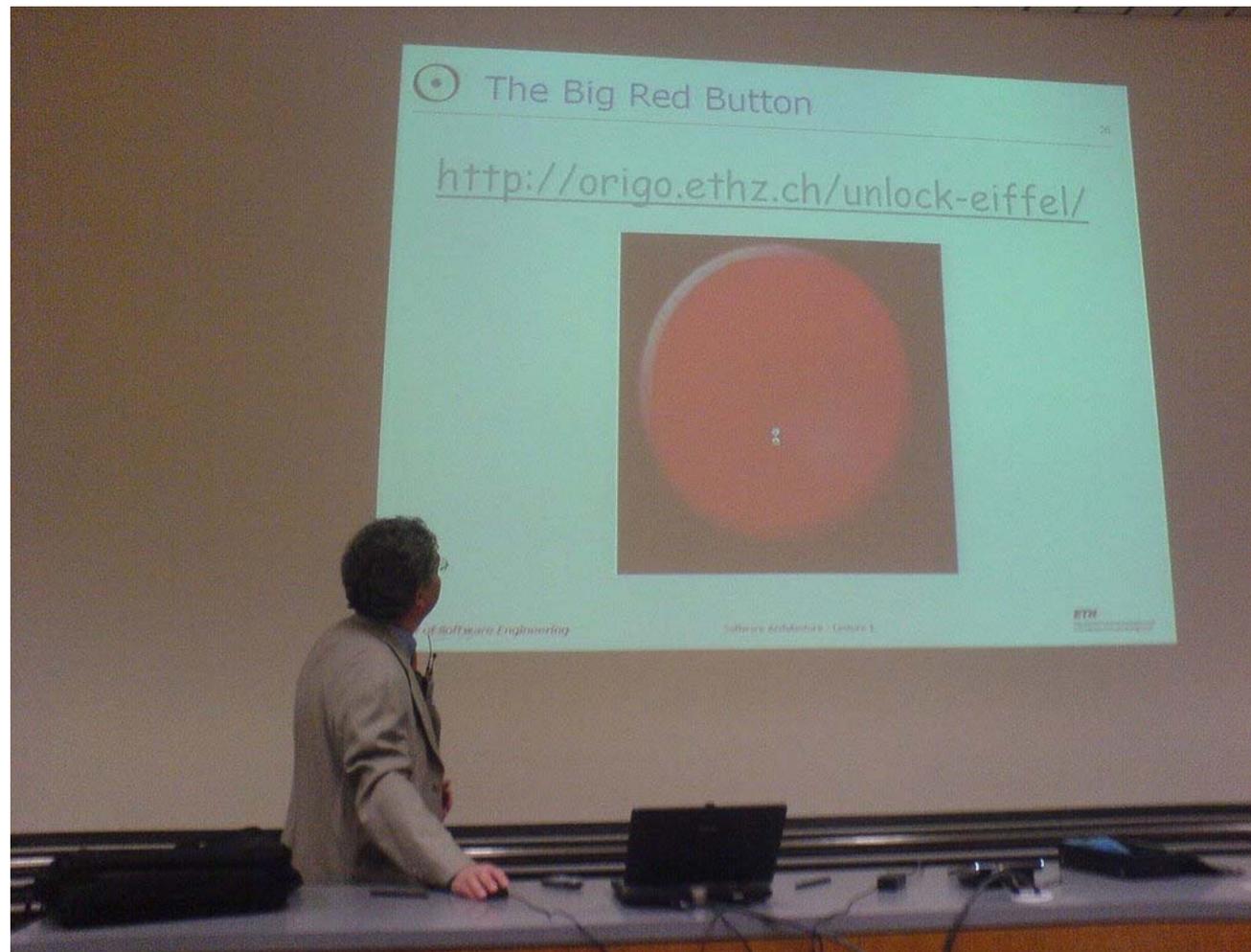
The interface also shows a Search panel on the left with search options and a Clusters panel showing the project structure.

EiffelStudio facts



Open Sourcing EiffelStudio

April 5th, 2006

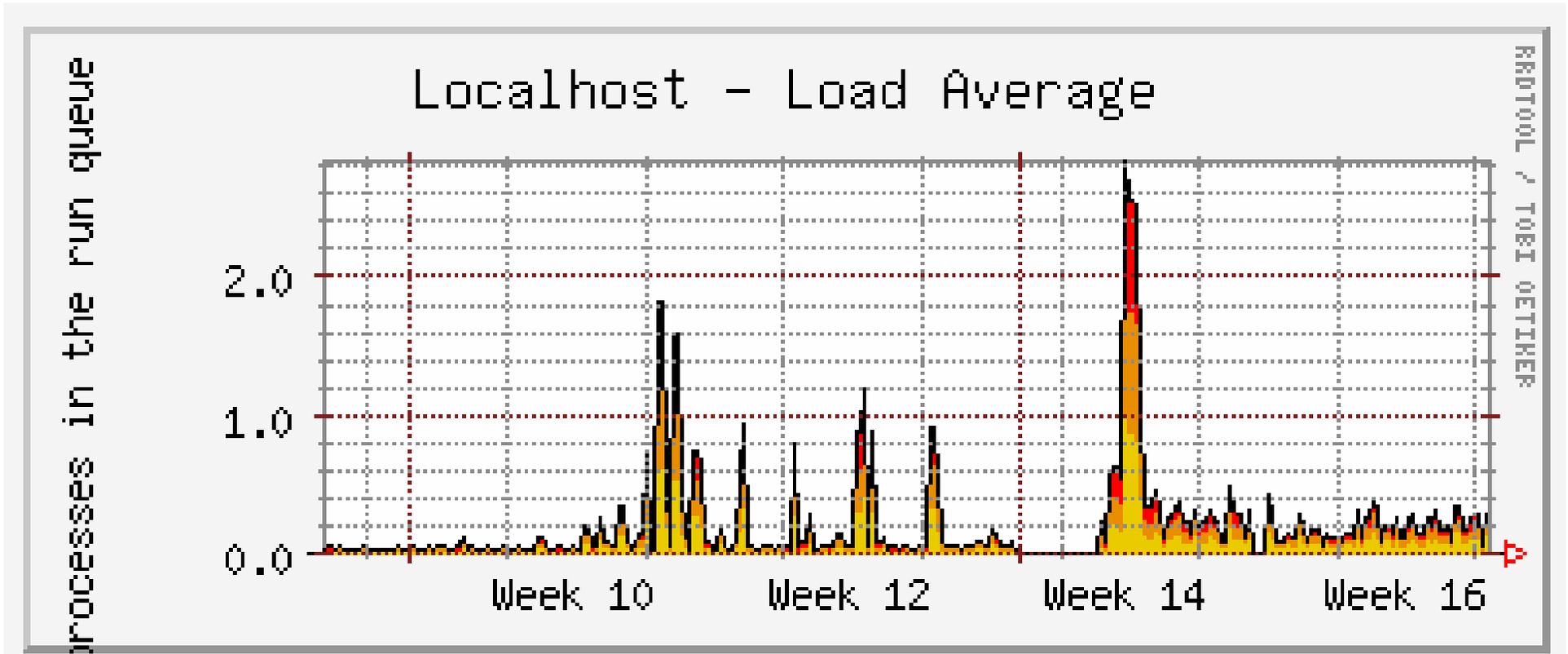


In the media

- Mentions on
 - Slashdot
 - Heise
 - Symlink.ch
 - Standard.at (!)
 - And many others



...and the Slashdot effect...



- 95 GB of transferred data on release day
- 100.000 Visitors on the project website

Overview

1. What is EiffelStudio?
2. Why did we open source it?
3. How did we open source it?
4. What can we learn?

Why?

Eiffel is the best thing since the invention of sliced bread.

(our modest and subjective opinion)



Why?

- Compilers are the first technology where open source has become the standard.
(Cambridge Technology Partners, 2006)
- Any programming language needs a critical mass. How?
 - ECMA / ISO standardization
 - Teaching Eiffel
(ETH mandatory first year course,
new textbook under development)
 - Open sourcing the technology

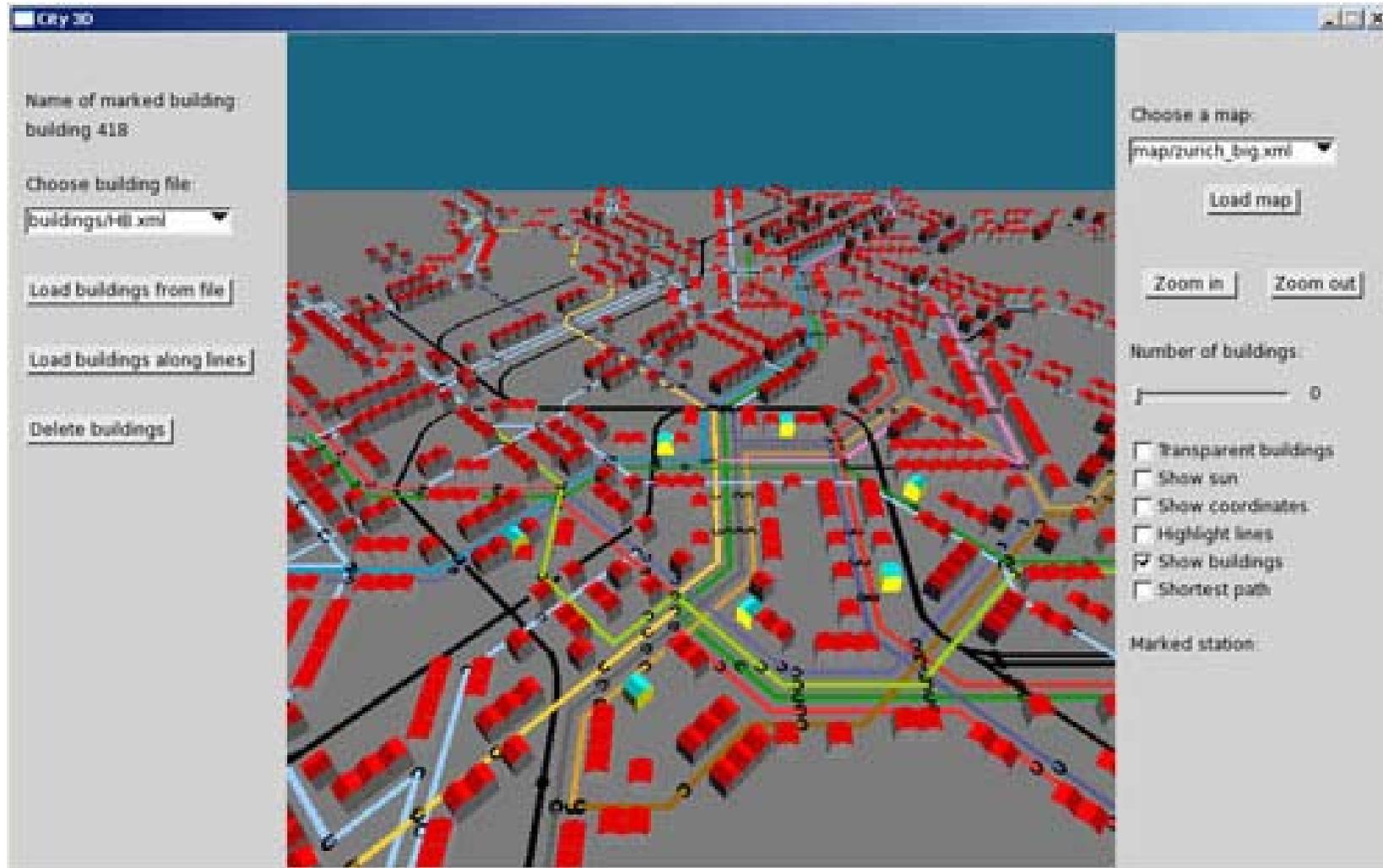
Overview

1. What is EiffelStudio?
2. Why did we open source it?
3. How did we open source it?
4. What can we learn?

Inverted Curriculum

- Good bye “Hello World”
- Large code base that is discovered “piece by piece”
 - Traffic (traffic simulation) 100.000 LoC
 - EiffelMedia (multimedia library) 500.000 LoC

Teaching Software Engineering



EiffelMedia

- 500.000 LOC
- 1.350 Classes
- 12.000 CVS Revisions
- 2.000 Messages
- 30 People
- 2 Years
- More than 150 Client Apps



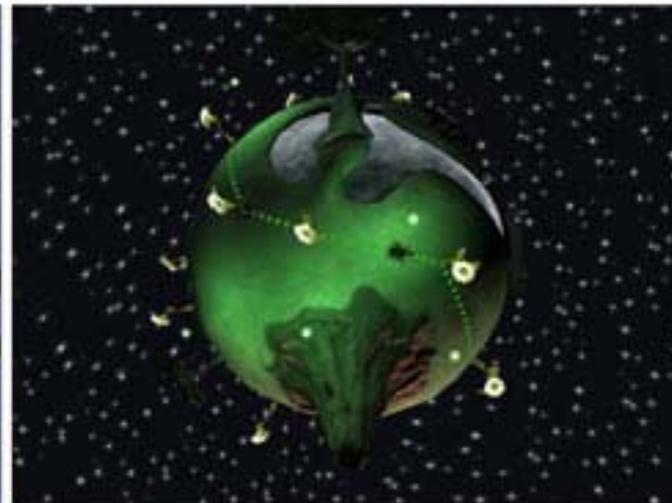
EiffelMedia matured over time...

- X Adventure Engine (1 year ago)

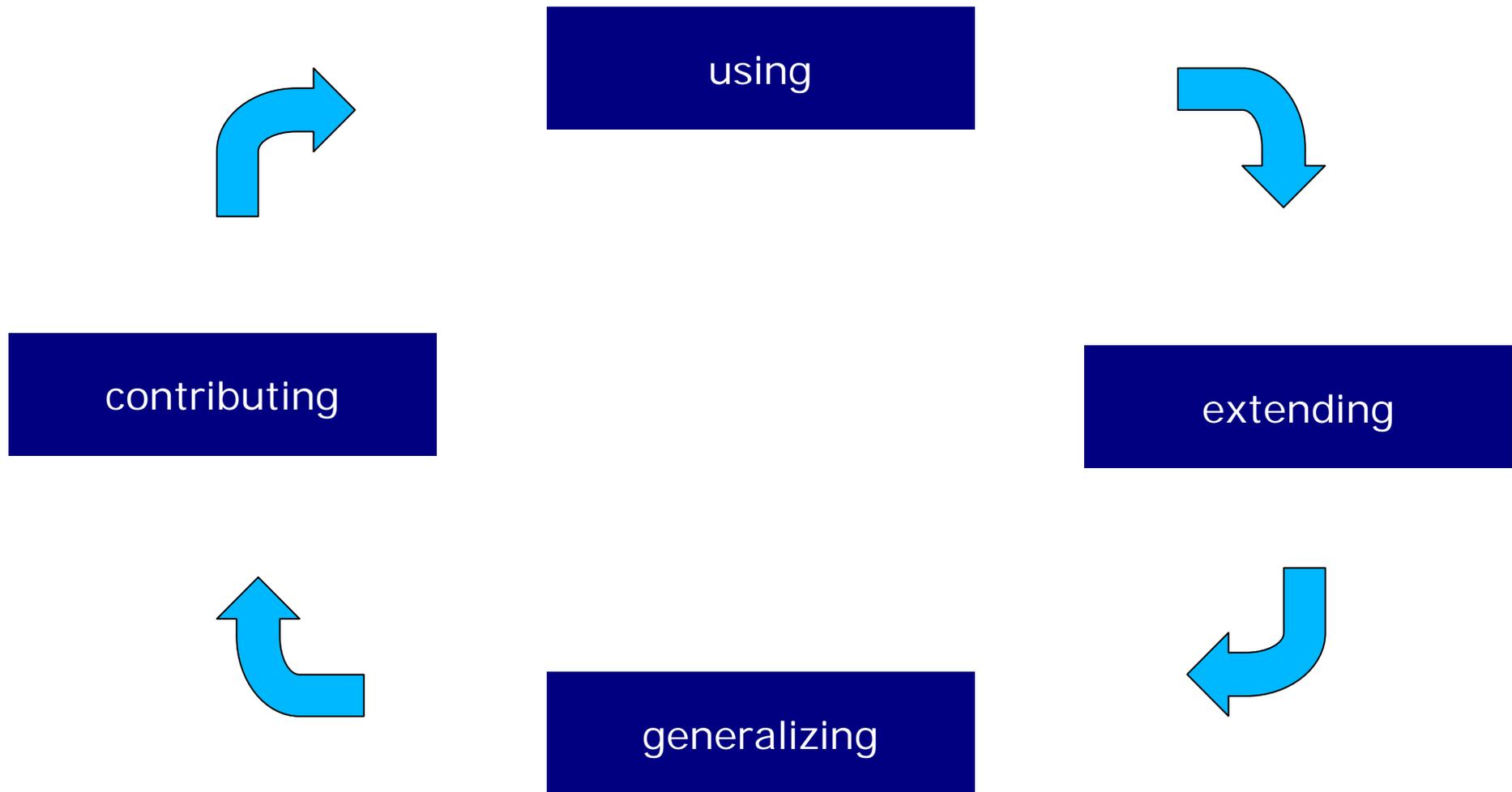


EiffelMedia matured over time...

- Antworld (this years best game)



The development process in the inverted curriculum



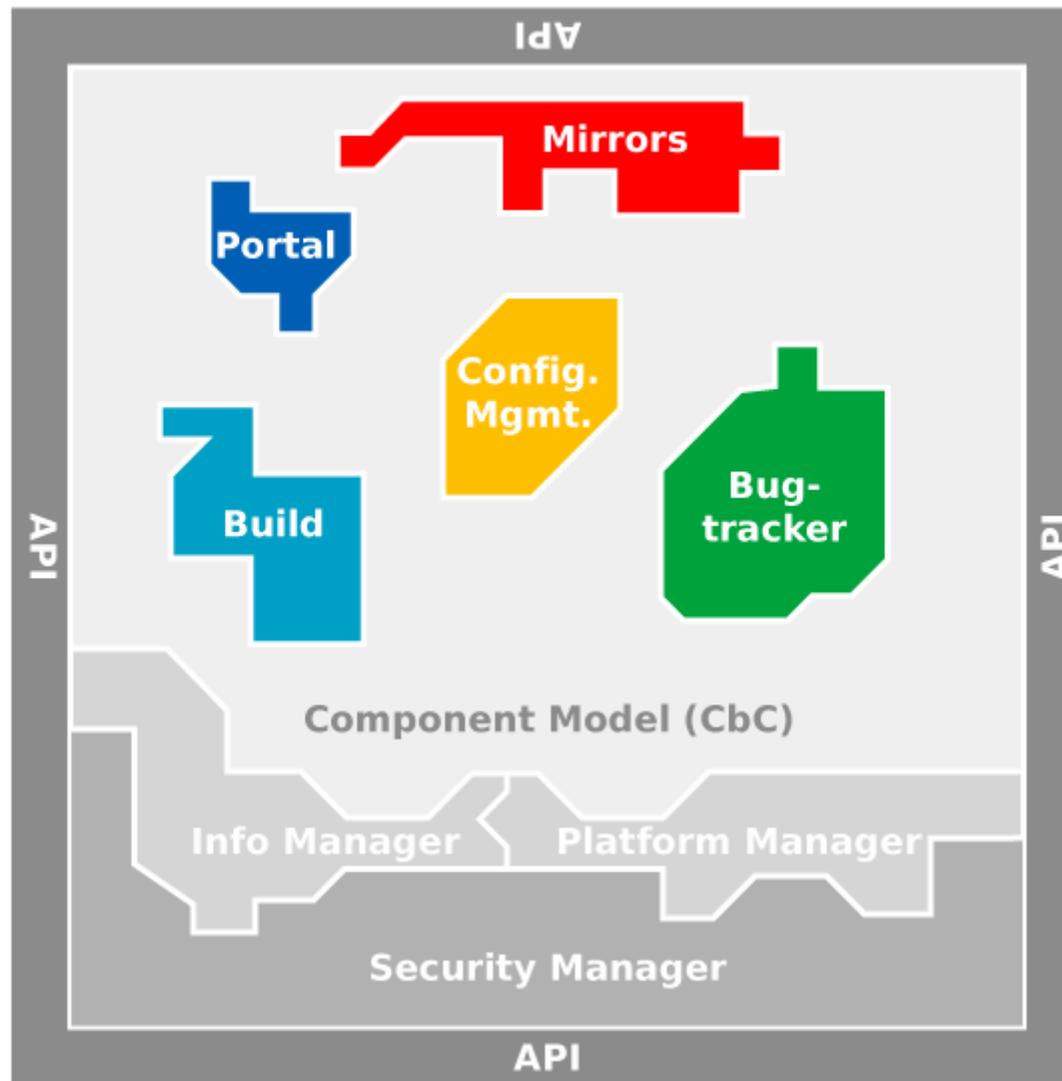
Overview

1. What is EiffelStudio?
2. Why did we open source it?
3. How did we open source it?
4. What can we learn?

How to host this process?

- Servers
- Configuration management
- Communication
- Presentation

The Origo Platform



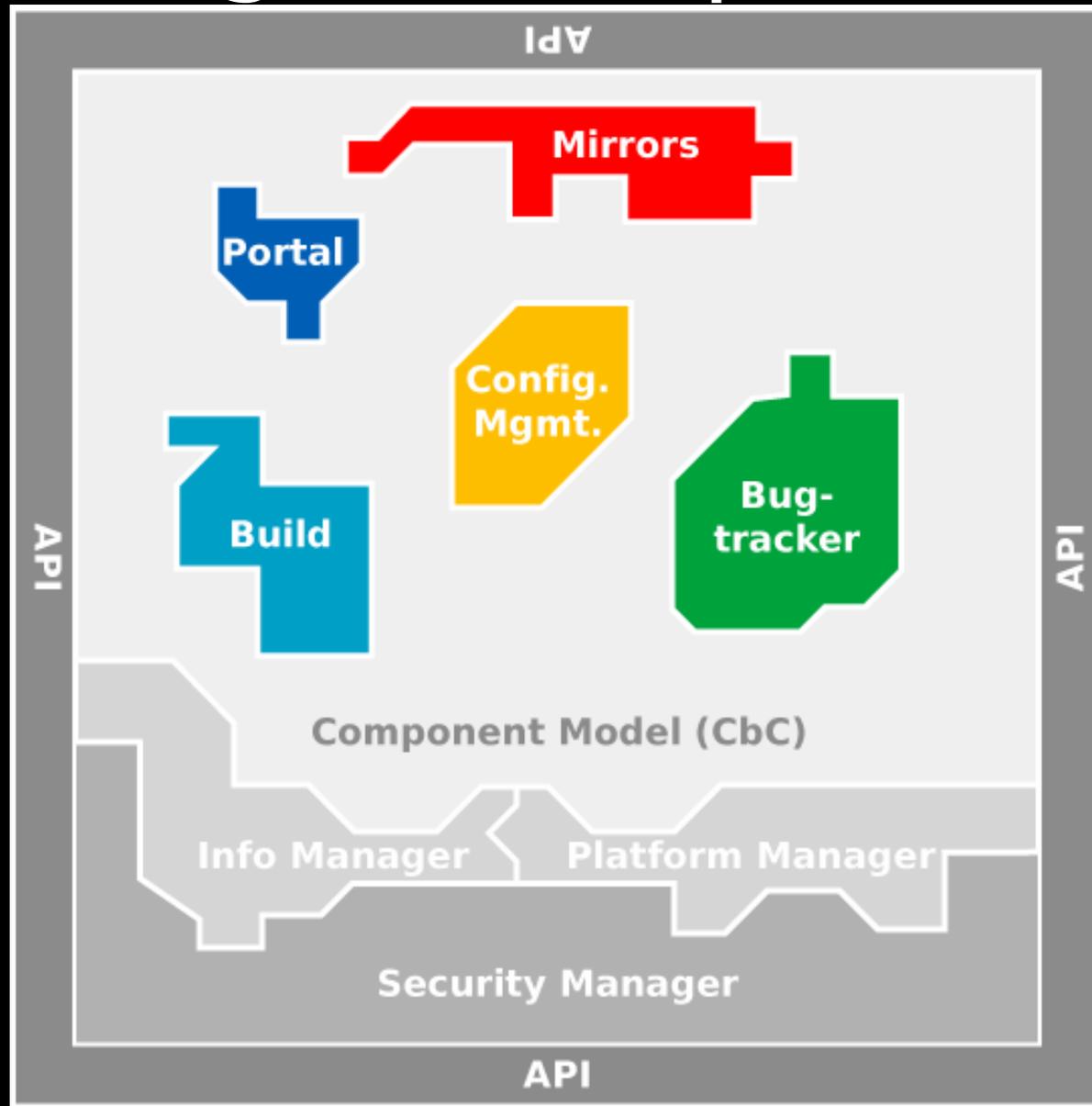
What Origo is

- A platform for distributed software development
- A model for integrating components

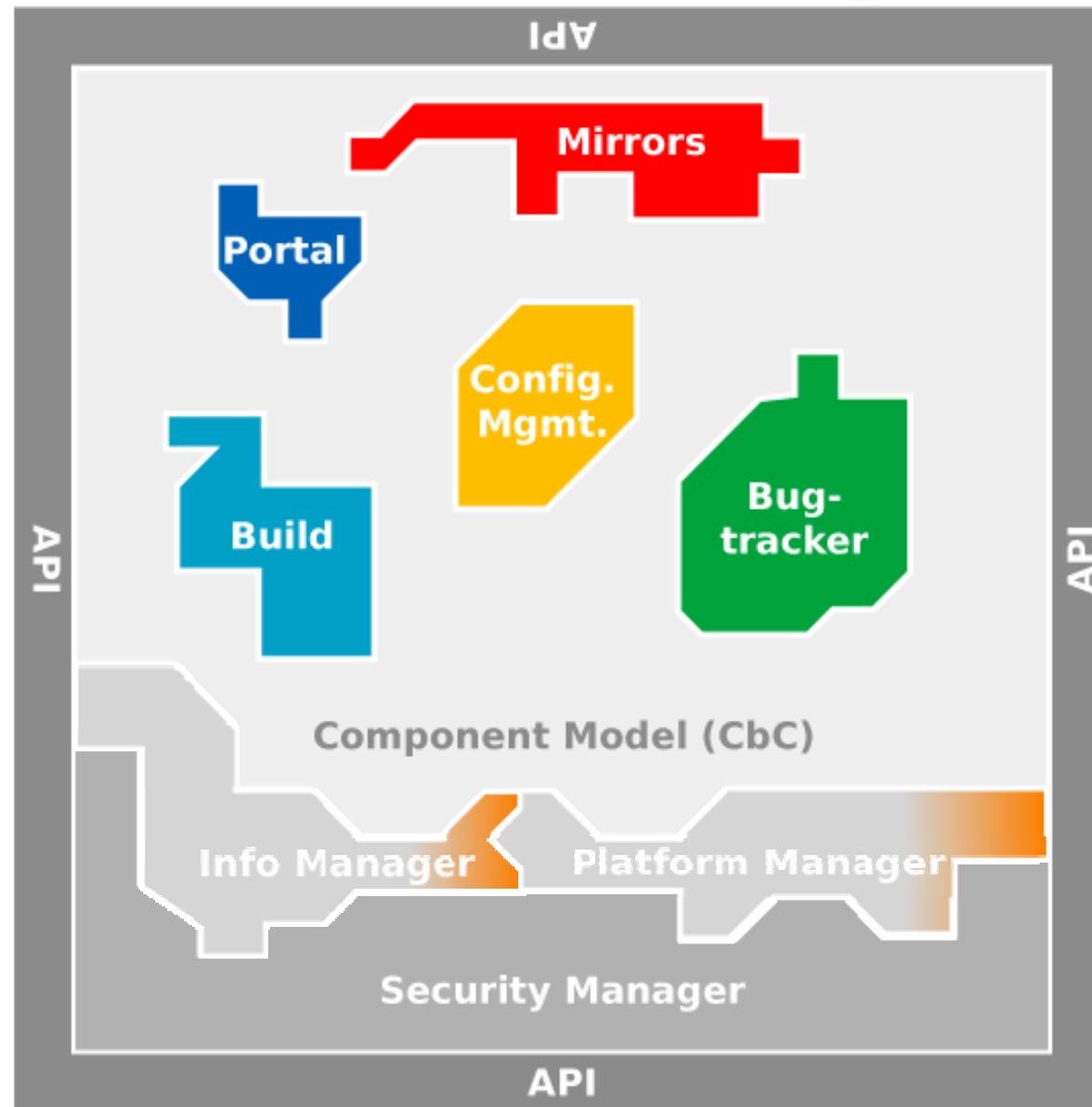
What Origo is not

- A project management tool
- A build server, configuration management repository etc.

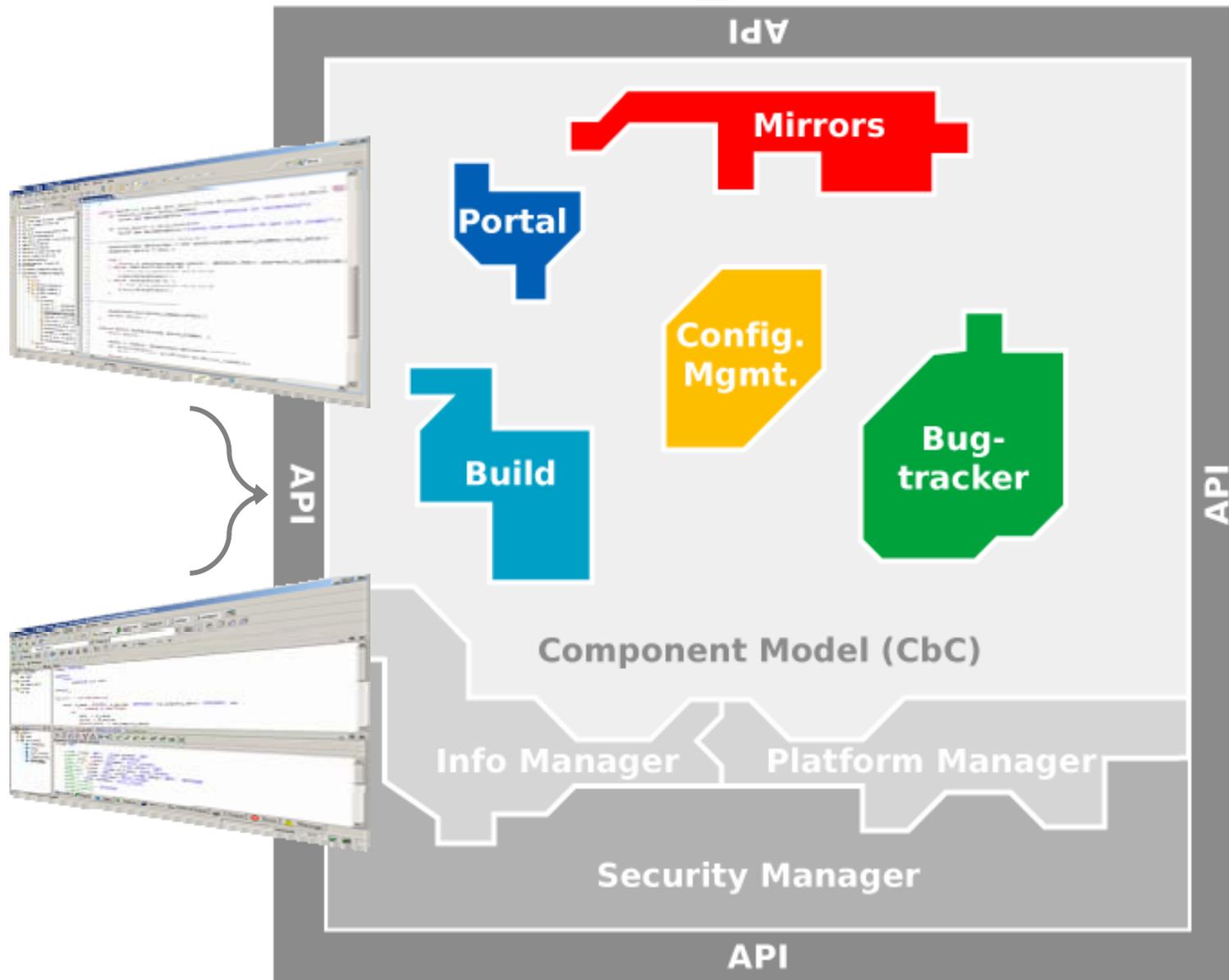
Origo's components



Extendibility



Integration



What is the added value of Origo?

- Technology independent
- Integrated into any process (API)
- Secure
- Extensible

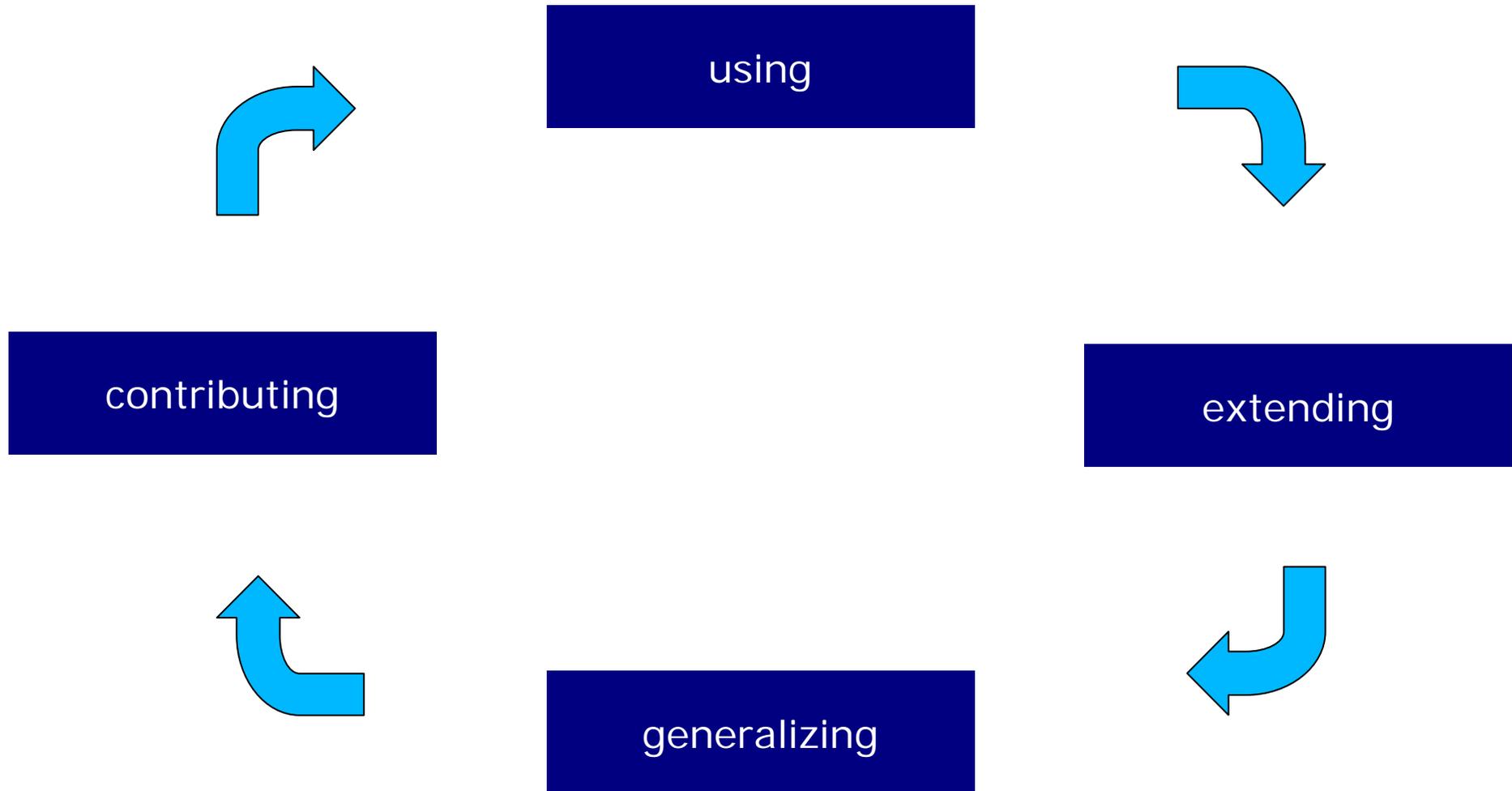
Coming back to EiffelStudio

- Project Start: April 5th
- Student Projects:
 - Folding editor support
 - Carbon port of GUI toolkit
 - Internationalization
 - Stripped down version for teaching
 - ...
- Common development tree for all projects

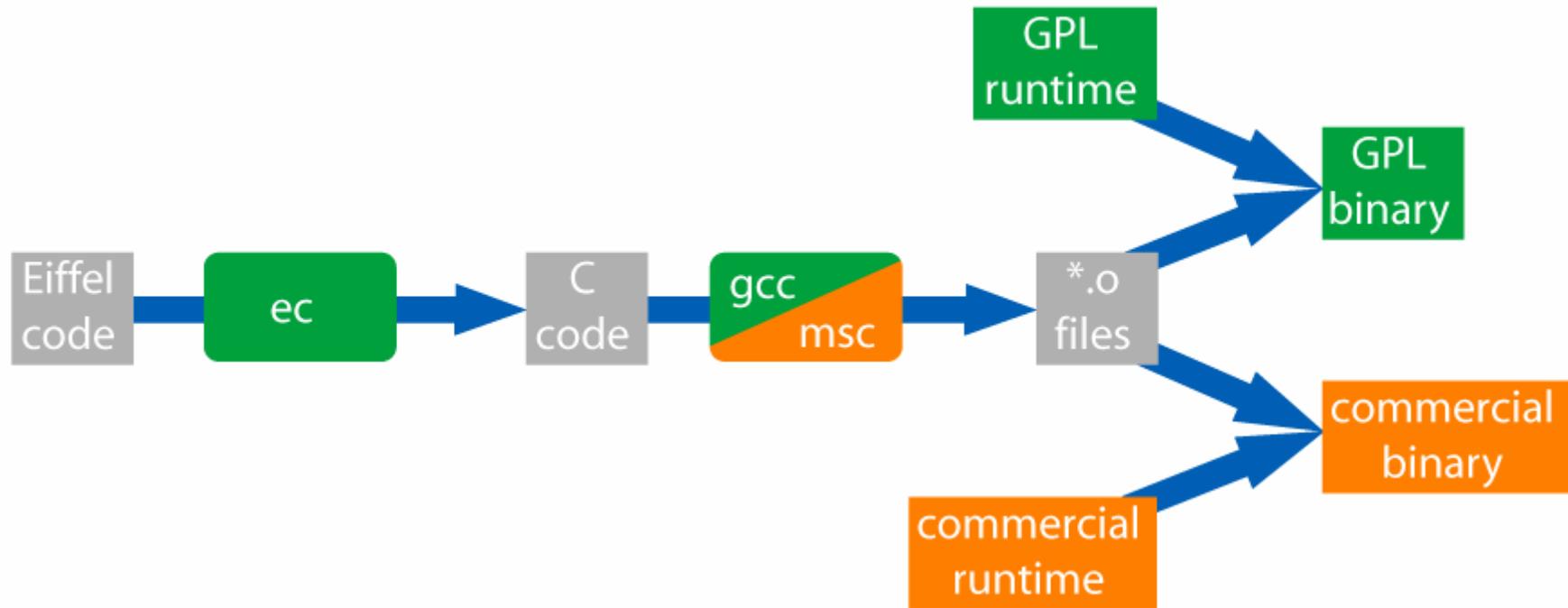
Which license?

- Continued commercial support
- Closed source development should use paid version
- Free for open source development
- No new open source license

Extending EiffelStudio



Dual License



Overview

1. What is EiffelStudio?
2. Why did we open source it?
3. How did we open source it?
4. What can we learn?

Hazards ahead



Lessons learned the hard way

Lesson Learned (Part I)

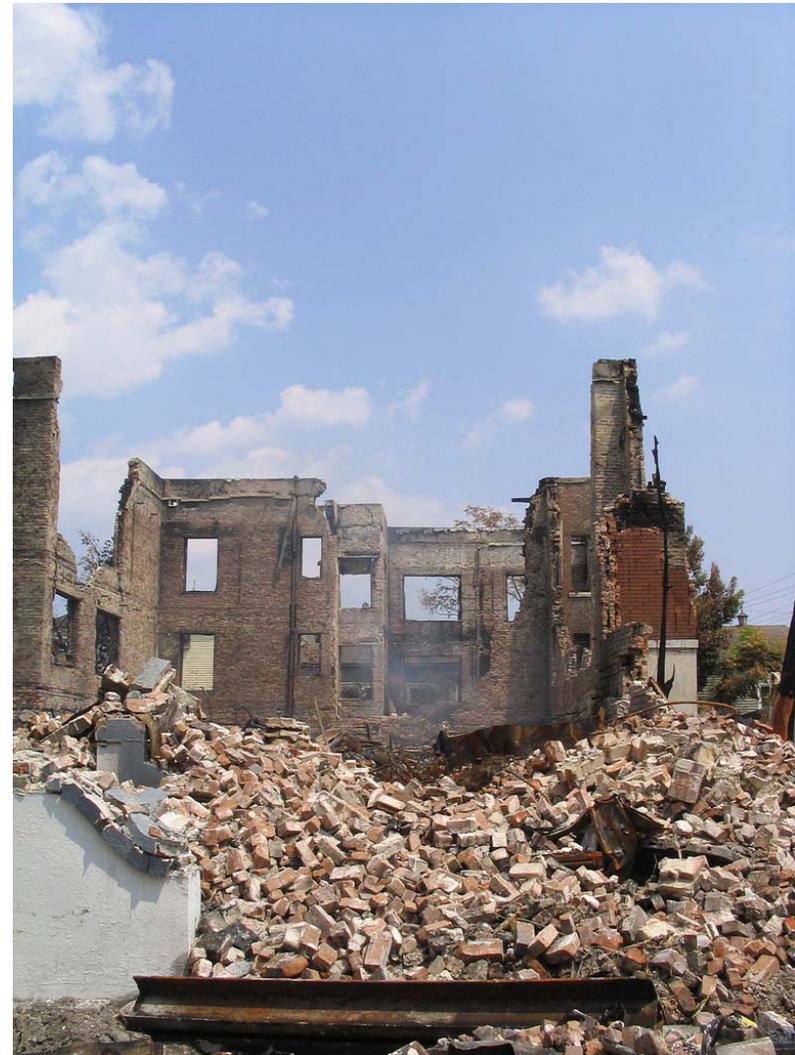
Make sure it is easy
to build your stuff
(alone, at home)



Lesson Learned (Part II)

Fear regressions

Example:
do not change
build file format



Lesson Learned (Part III)

Have a development plan (and publish it)



Lesson Learned (Part IV)

Eat your own dogfood



If you are interested

- EiffelStudio

<http://eiffelsoftware.origo.ethz.ch>

<http://www.eiffel.com>

- Games

<http://games.ethz.ch>

- All other open source libraries

<http://se.ethz.ch>