

# OpenTURNS developers training: agenda

---

## Trainers :

Régis LEBRUN (Airbus) - [regis.lebrun@airbus.com](mailto:regis.lebrun@airbus.com)

Julien Schueller (Phimeca) - [schueller@phimeca.com](mailto:schueller@phimeca.com)

Joseph Muré (EDF) - [joseph.mure@edf.fr](mailto:joseph.mure@edf.fr)

Sofiane Haddad (Airbus) - [sofiane.s.haddad@airbus.com](mailto:sofiane.s.haddad@airbus.com)

## Developers training



# The developers training : objectives

This 4 days training will give you the first elements to:

- Understand the OpenTURNS aim and architecture,
- Discover the coding rules, the development process and the associated infrastructure,
- Make your first steps in the OpenTURNS development by adding a new specialization of an existing concept both in the C++ library and the Python interface,
- Make your first steps in the OpenTURNS module development.

## General organization : agenda

Each of these four days will be organized as follow:

**9.30am** Welcome (coffee & tea)

**9.45am** Theory

**12.15pm** Lunch

**13.15pm** Theory or experimentation

**15.15pm** Break (coffee & tea)

**15.30pm** Experimentation

**17.30pm** End of the day

# Day by day...

## **Day 1: uncertainties, OpenTURNS platform, architecture**

- Uncertainties: quick reminder on analysis, probability and statistics
- OpenTURNS platform: the global picture of OpenTURNS product and the website, short presentation of the development life-cycle.
- Architecture: the general organization of the product with some highlights on the key mechanisms and their implementation.
- The development infrastructure: some elements on CMake.
- The development process: architecture, C++ implementation, doxygen, tests.

## Day 2: development in the C++ library

- Interfacing C++ and Python using SWIG.
- The development process: SWIG interface, python modules, tests.
- Experimentation: C++ implementation of the challenge.
- Experimentation: Python binding of the challenge.

## **Day 3: make the development visible in python**

- Documentation : docstrings, sphinx, examples, & others.
- Experimentation: documentation of the challenge + examples for existing classes (closing existing issues)

## Day 4: create an OpenTURNS module

- The concept of module, its standard structure.
- The key steps in the development of a module.
- How to install and use a module?
- Experimentation: implementation of the second challenge as a module.

### **You are a new OpenTURNS developer!**

- All the developments made during this training session will become part of OpenTURNS sooner or later, depending on their degree of maturity (quality ;) ?).
- The missing steps for a direct integration will be the testing phase, the validation phase and probably parts of the documentation phase.
- The integration will be done using a development branch on GitHub.

Check the upcoming release or the next to see your work in action!