Teaching Software Design and Programming

Günter Quast (KIT)
Benedikt Hegner (CERN)

HEP Software Foundation Workshop
20.1.2015

Why Teaching in the HSF?

- We have to make sure talented and motivated people join the software projects
  - the time of linux-aware hackers is over and we have to be proactive in teaching students the necessary skills
- At universities good teaching is the key to attract people
- Only a certain level of (design) quality allows sharing of work
  - Lack of proper teaching results in a throw-away code culture!

German SW Teaching

- **Situation at Universities**
  - *No proper teaching* of C++ or even SW design to (under-)graduate students
  - Learning by doing w/o "quality control"
  - Waste of efficiency and potential
- **German-wide community training to mitigate the situation**
  (in the context of the "Physics at the Terascale" alliance)
  - (Advanced) C++ Tutorial (1 week)
  - Advanced SW Design (1 week)
- **GridKa school offered to the international community**
- **Taking advantage of schools offered elsewhere**
  - CERN School of Computing
  - Bertinoro School
- **Such training is a huge investment and needs motivation towards supervisors, funding agencies, etc!**

HSF and Training?

- **HSF as forum for *exchange* and support**
  - Show that software is an essential ingredient to HEP research and deserves more attention at universities!
  - How do other countries tackle the programming and design teaching?
  - What kind of skills should we teach to students to be efficient in the HEP context?
  - Are there any best practices to communicate?
    - Allowing for interoperability between packages, in particular in a multithreaded environment
  - Collect information of tutorial events and whether they fit together
  - Maybe even organize events?
  - Last but not least: *share teaching material* across the community