

Joint HSF/WLCG Workshop
Naples, 26 – 29 March 2018

The Bertinoro School Of Computing (ESC)

Francesco Giacomini – INFN
(on behalf of all the people behind the school)

Motivation

- For me: Help physicists spend more time doing physics and less time debugging software
- With INFN since 2000
- Doing some teaching soon after, much more in recent years
- For ESC*: Introduce (young) researchers to the world of modern computer architectures and heterogeneous computing
 - how to deal with them in terms of software code and tools
 - the audience includes also non-particle physicists

* INFN International School on Architectures, tools and methodologies for developing efficient large scale scientific computing applications

Computing fields of interest

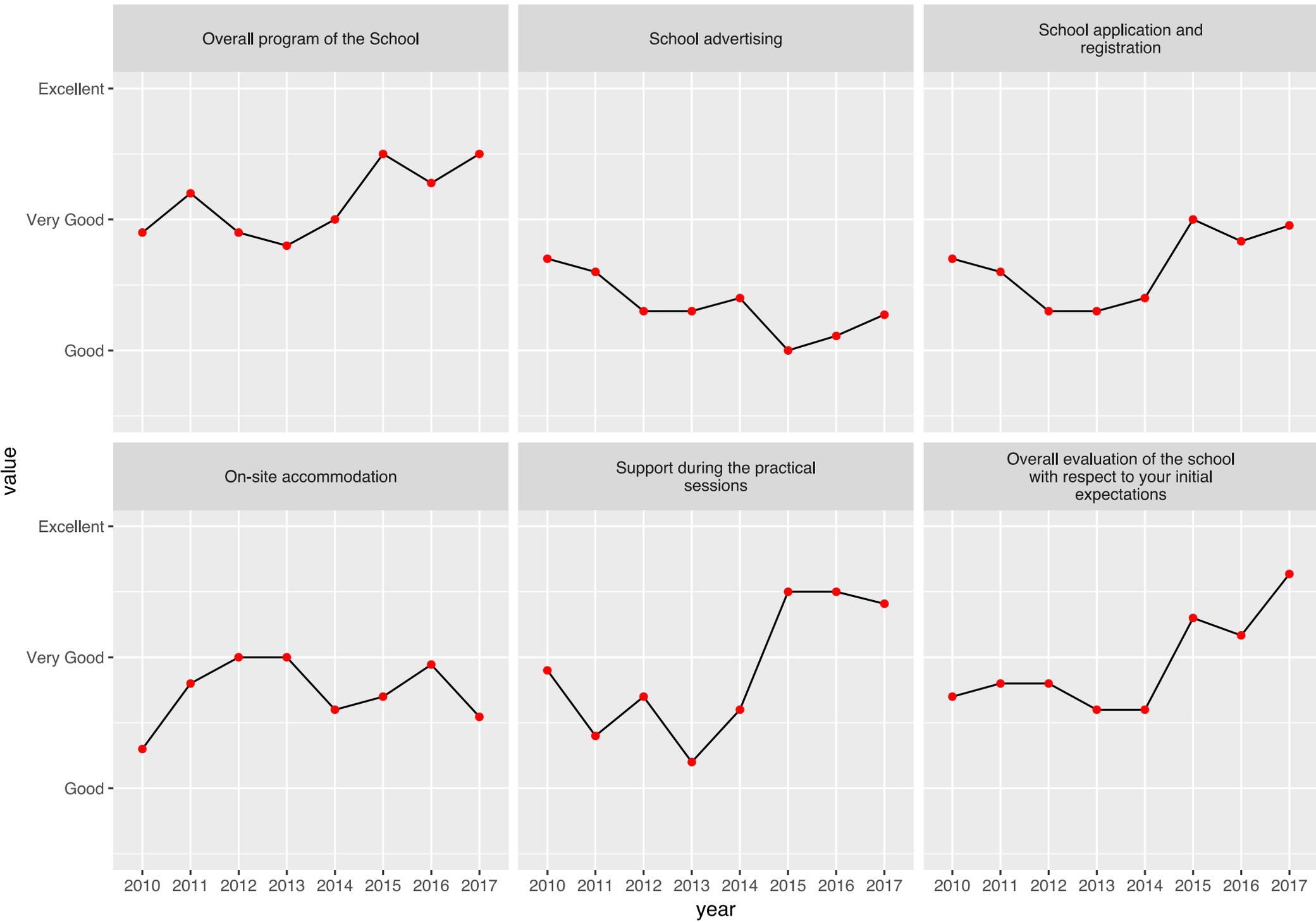
- For me:
 - Modern C++
 - Tools to improve code correctness
 - especially if they help to automate tasks, e.g. for continuous integration and testing
- For ESC:
 - Modern C++
 - Memory-friendly programming
 - Floating-point computation and vectorization
 - Parallel programming (multi-core, GP-GPU, multi-node)
 - And related tools

How the ESC school works

- Five full days, from Monday to Friday
 - registration on Sunday, test and evaluation questionnaire on Saturday morning
- A typical day
 - 8.30 – 19.30
 - 90-minute sessions, lectures and hands-on
 - two 30-minute coffee breaks, 90-minute lunch
 - “consolidation” time, self-managed by students, but with lecturers' and tutors' assistance
- Two evening easy lectures
- Students' lightning presentations

How the ESC school works (cont.)

- Added value of the school: face-to-face interaction among students and lecturers
 - coffee breaks, lunches, hands-on consolidation time, dinners in Bertinoro
- Challenges
 - keeping up with technology advances and user needs
 - have a unique thread through all the lectures
 - no repetitions, C vs C++, ...
 - find the right balance between theory and practice
 - don't overload students with information
 - adapt to the audience
 - ...



Future

- Invest on trainers
 - Being a trainer requires
 - time to learn what you teach → study, follow forums/ mailing lists, write real code, ...
 - time to prepare the material → share experience and material (slides, examples, exercises, ...) among trainers
 - time to give the course → more trainers
 - dedication → motivate people to teach
- Involve trainers both
 - in large experiments → learn from the challenges
 - in small/medium experiments → spread knowledge, e.g. through mentoring on code writing, software quality assurance, ...