Technical Consortium on High Performance Computing: Software Engineering Initiative (TCHPC/SE)

George K. Thiruvathukal
Professor of Computer Science, Loyola University Chicago
Visiting Faculty, Argonne National Laboratory

About TCHPC and TCHPC/SE

The IEEE Computer Society Technical Consortium on High Performance Computing (TCHPC) advances and coordinates the work in the field of high performance computing networking, storage, and analysis concepts, technologies and applications.

The TCHPC Software Engineering Initiative (TCHPC/SE) aims to identify and embrace best practices in software engineering to ensure robust HPC solutions.

What is Software Engineering?

Software engineering is the establishment and use of software engineering principles in order to obtain economically software that is **reliable and works efficiently on real machines**. (Pressman and Maxim, 2016)

“the application of a s**ystematic, disciplined, quantifiable approach to the development, operation, and maintenance of software**; that is, the **application of engineering to software**).”

The **study of approaches [to software engineering]** above. (SWEBOK 3.0)

Margaret Hamilton

Coined the term Software Engineering to address mission critical software development and the need for *engineering*.

“There was no second chance. We all knew that.”

Directed software engineering for the entire Apollo Space program

Presidential Medal of Freedom, 2016

SE Need not be Difficult

Workshop on Sustainable Software for Science: Practice and Experience position paper by Carver and Thiruvathukal

- Source code management (e.g. svn, hg, git and hosting solutions)
- wikis
- issue tracking
- testing
- automatic builds
- project management

http://ecommons.luc.edu/cs_facpubs/75/

Can we do more?

- capturing high-level **requirements**
- understanding functional vs. non-functional requirements
- **design** prior to coding of complex software
- **metrics**: in-process metrics, code metrics (looking beyond number of commits)
- greater integration of **coding, testing, and documentation** (what agile actually emphasizes)
- using **code reviews** to identify / eliminate defects at the source

Possible Community Trajectories

- Coordination
- Referencing
- Advocacy
- Conferences/Workshops
- Reports
- Resources

Discussion on Slack will help to finalize the list.

Call to Action

E-mail to George K. Thiruvathukal (gkt@cs.luc.edu) to join the TCHPC slack.

We welcome *everyone* but ask you to include the following for verification purposes:

- full name and affiliation
- title (optional) - we welcome academics, researchers, and practitioners
- web page or link to professional profile

Please use subject “Join TCHPC/SE channel on Slack”

Related Efforts

- Computing in Science and Engineering Software Engineering Magazine SE Track: www.computer.org/cms/Computer.org/ComputingNow/docs/2016-software-engineering-track.pdf
- Software Engineering for Science Workshop Series, www.se4science.org/workshops/
- Software Sustainability Institute, www.software.ac.uk/
- Jeff Carver, Neil P. Chue Hong, and George K. Thiruvathukal, Software Engineering for Science, CRC Press (free chapter), ecommons.luc.edu/cs_facpubs/128/