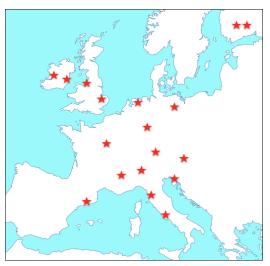# E-CAM Deliverable 5.1 – Guidelines for the Extended Software Development Workshops (ESDWs)

Delivered in Month 3 – December 2015

The European Centre of Excellence for
Software, Training and Consultancy
in Simulation and Modelling

Preamble

These guidelines are intended to be a living document which evolves to reflect experience gained in running the ESDWs and thus this first iteration is not overly prescriptive. In particular, the guidelines are subject to revision based on the outcomes of each year's activities.

This deliverable was co-ordinated by Sara Bonella (EPFL, Lausanne) on behalf of the E-CAM consortium with contributions from Emilio Artacho (UB), Sara Bonella (EPFL), Luke Drury (NUI UCD), Michael Lysaght (NUIG ICHEC), Donal MacKernan (NUI UCD), Godehard Sutmann (Jülich) and Dominic Tildesley (EPFL).

Contents

E-CAM will deliver four workshops every year, each focused on software development in one of its four core scientific areas: classical MD, electronic structure, quantum dynamics, meso and multiscale modeling. The purpose of the workshops, called **Extended Software Development Workshops** (ESDW), is twofold. On the one hand they are a mechanism for generating software modules for inclusion in the E-CAM repository. On the other, they are an integral part of the E-CAM training programme and represent "training by doing".

1   Purpose and structure of the ESDW

The typical size of the workshop will be 10-15 trainees plus staff.

Trainees: The trainees will consist of a balanced mixture of post-doctoral research assistants or senior PhD students in the third or fourth year of their postgraduate studies, and young industrial researchers. These researchers will have already trained in simulation and modelling and would be considered part of the companies expertise base in this field. The ESDW will extend their skill range and produce the required training boost to keep them at the leading edge of their discipline. These workshops will also be of potential value to academic staff from European institutions, in the process of changing or extending their current research fields. Trainees should come from a diverse range of backgrounds and a conscious effort will be made to avoid biases, implicit or explicit, in their selection, especially in the area of gender.

All attendees must commit to the full period of the workshop both in residence and back at their home bases. In the case of industrial participants this will require the written permission of their line-managers and in the case of academics the written permission of their supervisors. ESDWs will normally be available to European applicants but applications from the rest of the world will be considered and accommodations will be made when the non-European applicant can add high value to a workshop. Given the significant time requirement for participation in one of these workshop early and binding commitments will need to be obtained from perspective trainees. It is important that people appreciate that the workshops are indeed "extended" in that significant amounts of work will be done outside the face-to-face meeting.

Staff: The workshop will be led by at least one senior academic member of the E-CAM consortium, who is an expert in the field under discussion and already familiar with the needs of industry or academic users. He or she will be supported by the project software manager, E-CAM software engineers and the PDRAs associated with the science area of the ESDW in that particular year of the project. We will invite other senior stakeholders and guests to lecture and mentor during the workshops, and computer manufacturers, such as IBM, Nvidia and Intel, to provide hands-on

programming support, and when appropriate even participate within training sessions/teams.

1.1 Logistics

Trainees will normally be divided into teams to develop the software, and will need office space, with projectors, white- boards and flip charts. A small resource library will be created of leading edge texts available on-line which can be used with each workshop. Code development will be performed on desktop machines at the location of the ESDW or, preferentially, on portable computers belonging to the trainees. When necessary, remote access to massively parallel and heterogeneous platforms will be ensured, including specialised compilers, profilers and debuggers. Specifications for the minimum requirements in terms of operating system and installed software and compilers will be detailed before each ESDW.

Assistance will be provided, as for usual CECAM schools, to the attendees and staff for accommodation during the residential parts of the ESDW.

Remote collaboration tools will be identified to enable the teams to continue to work on their software projects when dispersed back to their home institutions. Host institutions must provide necessary system level support to enable such remote collaboration (e.g. opening designated ports on firewalls).

1.2 Funding

Funding for the ESDW is provided via the CECAM contribution to E-CAM: 2/3 funded from CECAM Headquarters, 1/3 from the local beneficiary hosting the event including possible funding from external sources and co-sponsors.

Participation is free of charge for academic participants and for industrial participants from one of the current or future industrial partners of E-CAM. An attendance fee of up to 1000 Euro may be charged to industrial trainees from outside the E-CAM partnership. Special provisions for licensing and upload to the repository of software developed by paying trainees will be made if necessary.

2 Output and evaluation of the ESDW

The output of these workshops will be a number of modules, prototypes or final products depending on the precise state of the field. The number of expected modules from each ESDW will be established during the review process leading to the

definition of its contents (see below) and will contribute to meet the projected number of modules for each year in each scientific area.

Outputs from the ESDW may be made available through an on-line Wiki, and will also be uploaded on the E- CAM repository according to the schedule envisaged under the E-CAM Grant Agreement. The software manager is responsible for the evaluation of the software quality of the modules and their upload to the E-CAM library under the open-source software licence conditions agreed by the consortium.

At the end of each ESDW, the software manager and Senior E-CAM scientists leading it will prepare a report on the contents of the workshop and related actions. These reports will be used to monitor and improve the structure and planning of future ESDWs, as anticipated in Work Package 5 (Training) and in Work Package 6 (Software Infrastructure) in the E-CAM Grant Agreement at technically more detailed level.

2.1 Key Performance Indicators

The output of the ESDWs will also be measured via a set of key performance indicator (KPI), such as:

- Number of software modules developed and assessment of the production of working software responding to the needs of the users of the E-CAM infrastructure and its actual usage. The users include academic team members directly associated with ECAM beneficiaries, industrial affiliates, and external users.

- The training impact of the ESDWs, measured both in terms of the number of people trained and the level of technical skill acquired through the ESDWs and associated activities before and after.

- Impact in increasing diversity within the training programme, both in terms of gender, and academic background/ discipline.

- Degree of industrial engagement in the ESDWs.

The precise formulation of the KPIs in quantitative terms will be determined through discussion within the consortium and during the definition of the yearly program.

3 Definition of the yearly program

The locations and indicative contents of the workshops in the first year of the project are as follows

The content and location of the workshops for the first year is established based on the activities planned in Work Packages 1-4 in the Grant Agreement. Starting from

Workshop	Node	Date	Main topic
Classical MD	AT	Months 10-12	Trajectory sampling algorithms
Electronic Structure	UK-MAXWELL	Months 8-10	Wannier function based methods
Quantum MD	FR-IDF	Months 6-8	Sampling quantum initial conditions
Meso Multi-scale	ES	Months 10-12	Identifying reaction coordinates

year two, the program will be decided based on the following procedure. The software modules to be developed in each area will be selected during the year via:

1. a call for proposals for specific development requests published in accordance to the CECAM calendar (open April and close in June);

2. the software development needs of the PDRAs associated to the Pilot Projects in Work Packages 1-4;

3. direct requests from current industrial partners;

4. the outputs of the E-CAM scoping workshop;

5. the outputs from the E-CAM state-of-the-art workshops.

The software development requests will be collected and monitored by the software manager and the technical manager and considered in detail by the ESDW organiser (the academic beneficiary responsible). These requests will be ranked in order of merit and assigned to a specific ESDW via a refereeing process. Referees will be selected by the software and technical manager according to the specific scientific areas of the proposed ESDW within Work Packages 1-4 and members of the Scientific Advisory Committee of CECAM. Within the spirit of modern software management methods, the teams will have significant freedom to choose or prioritise which of the code modules ranked above they wish to develop.

Although the original locations for the ESDW workshop have been decided (see table 1), it will be possible to choose different locations defined by the beneficiaries responsible for their delivery, in agreement with the Software Management Group and CECAM.

The program for each year will be published by Dec. 20 for the following year of the project.

4   Duration and content of the workshops

Depending on the specific subject area, the workshops will have a variable total duration. Generally, they will consist of

- Preliminary work by the attendees based on basic on-line training tools to ensure minimum common expertise

- Two-week residential workshop

- Variable period of on-line work

- Final one-week wrap-up residential workshop

The training tools for the preliminary work will be defined in detail in the software development guidelines (Deliverable D6.1, month 6). These training tools will consist of documentation (theory and examples) and exercises and assume basic programming skills in one or more languages (including C, C++, FORTRAN 2008, PYTHON). Trainees will be required to complete specific exercise sets prior to attendance to the workshop. In this preliminary phase, assistance will be provided via a web based help desk. Contents of the on-line training modules will include: Use of a structured Wiki page; Basic scripting tools; Basic parallel programming (MPI, OpenMP); Version Control (Git [1], Bazaar [2], Launchpad). The software manager, in collaboration with the software engineers and experts from the PRACE centres, will prepare this material. A more detailed description of this tasks will be provided in Deliverable D6.2 (due in month 8). The tools will be hosted on the E-CAM server at CECAM HQ.

In addition to coding sessions, the typical workshop will consist of a number of training lectures in leading-edge technique in software development; lectures on computer hardware and advances in new architecture; parallel programming techniques including MPI; a number of leading-edge scientific presentations to stimulate ideas; a number of presentations from software and hardware vendors. The residential workshops will be intense with attendees working in small teams of between 6-9 members including trainers/teachers. Each attendee is expected to work on one or more software development projects during the course of the ESDW.

The ESDWs will teach and implement general, language independent, skills to ensure that the E-CAM library embodies long term coding best practices.

Software development methods will be close to the approach successfully used in open-source projects and will exploit modern management methods such as Agile [4,5,6] and in particular SCRUM[7].

Software will normally be written in C or C++, modern versions of FORTRAN (e.g Fortran 2008) and PYTHON. The interoperability of the software in each language will be an important component of the development. When appropriate, the codes will be parallelized for use on CPU clusters (and possibly mixed CPU-GPGPU clusters) using distributed, shared or hybrid shared and distributed memory (using MPI or OpenMP). All code will contain extensive in-line documentation. Subroutines and functions will be stored collectively in modules, which can be used to construct a variety of programs in a common application space from the same building blocks. Each program constructed from the modules will come with some realistic test cases, specimen results and timings for implementation on a number of different machine architectures. The underpinning algorithms, input data structures and expected outputs will be carefully documented and stored with the modules. An important part

of the work of ESDWs will be the creation of a number of wrappers which will allow the module to interface to more standard commercial packages such, for example, as LAMMPS, GROMACS, DL_POLY, ABINIT, Quantum Espresso, CP2K, or VASP. Where wrappers are created they will be stored with the code and tested with the packages under consideration.

A more detailed description of the software standards for the workshops is contained in Deliverable D6.1 (month 6).

5 References

[1] https://git-scm.com

[2] http://doc.bazaar.canonical.com/bzr.2.6/en/user-guide/index.html

[3] https://launchpad.net

[4] D Turk, R France, B Rumpe, Limitations of agile software processes (2014) http://arxiv.org/abs/1409.6600

[5] https://en.wikipedia.org/wiki/Agile_software_development

[6] Sliger, Michele, and Stacia Broderick.The software project manager's bridge to agility. Addison-Wesley Professional, 2008.

[7] https://en.wikipedia.org/wiki/Scrum_(software_development)