Seppo Sierla 5.5.2011

Constructive alignment of the course: Safety critical automation systems (5cp)

Learning goals

After passing the course, the student will be able to associate activities of the safety process to the system and software engineering process. He/she will be able to work within a role in a team applying these processes to a project in the work machine domain.

More detailed learning goals for teamwork and project work (this does not go to OODI)

- Negotiate decomposition of complex task to smaller subtasks
- Independent development of subtasks according to shared design principles
- Balance between listening to others and influencing the team, according to role
- Identify integrity problems between subtasks *throughout* the project
    - Role of architect must be respected
- Maintain compliance to guidelines required for external audit
    - Role of project manager must be respected
- Controlled integration, testing and fixing
    - Manage concurrent fixes
    - Regression test after fixes
- Tenacity to bring a relatively long project to completion, and to respect project schedule and goals

More detailed learning goals for the safety critical software project (this does not go to OODI)

Improving the safety of software is achieved by improving the development process, and this relies mostly on the same software engineering methods are used in any mature organization aiming at high quality. The major software safety standard IEC 61508-3 makes specific recommendations and requirements on the process itself and on the various techniques that can be used at different phases of the process, depending on the risk level of the application. The students will apply a process (the V-model) that satisfies IEC 61508-3, using several specification and quality assurance techniques (including architecture and module interface specification, tests planning and specification, unit and integration testing and documentation to create evidence that these techniques have been conducted professionally). The standard is very detailed and is aimed squarely at experienced professionals in the area, so it is not read on the course. Students will get a practical experience of a software project that complies with the most fundamental requirements for safety critical applications. The course qualifies students to approach a software safety standard to obtain

detailed requirements for their application, after the safety integrity requirements have been obtained through risk assessment (which I teach on the ALM exercise of the AS-116.3110 course).

Teaching method: PBL

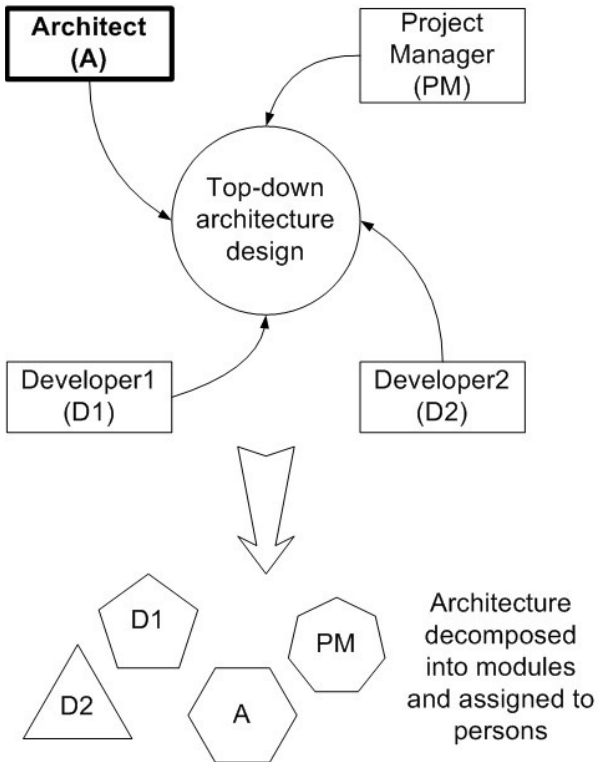
There are many variations of PBL (Problem Based Learning). On this course, the key characteristic of PBL is that each group will explore and produce several design alternatives, which it has to evaluate in order to select the best one. The creation, evaluation and selection will involve both individual and group work, and the team will be required to negotiate and reach a consensus on the best alternative. PBL is combined with project learning, so the phases and team roles are not the same as in some PBL literature, because a professionally relevant project context is pursued.

The role of the teacher in PBL is to facilitate. Teams will be given guidance for the project in a way that integrates the teamwork aspects and technical development process aspects. It is important to note that the nature of teamwork (regarding communication/negotiation needs and special responsibility of roles) will be different in different phases of the development process. During the first lecture, the team process and role responsibilities below will be introduced, after which groups are formed. Each member describes their background and interests, after which roles are chosen. One may either choose a more or less familiar role, as long as the person has sufficient experience to perform the role successfully. The teacher will facilitate this process. Roles will remain the same throughout the semester-long project, because this is the likely situation in a real project.

Below are the phases of the project and the guidelines for each phase, which are explained in the beginning of the course and then facilitated by the PBL instructor as the team works. The guidelines seamlessly integrate engineering and teamwork aspects, and this has 2 advantages:

- no “extra time” is spent practicing teamwork for its own sake
- teamwork skills will be easy to apply in a professional context, since it has been verified with external stakeholders that this engineering process and project roles are well accepted by industry, academia and safety authorities

Architecture design

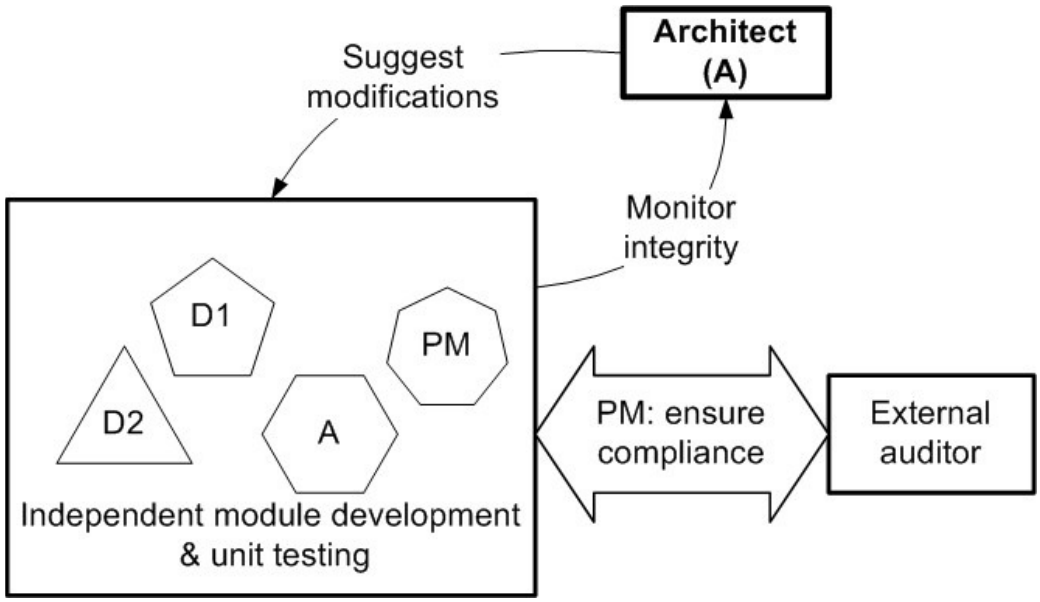


As shown in the figure, all members contribute to identifying modules and their relationships. Ideally, this will be a democratic process, but the architect may and should exert authority to reach a working solution if the democratic process does not seem to converge within the project schedule. The resulting modules are assigned to individuals. The developers should have twice as much responsibility as the other roles.

Role of PBL instructor: observe student process and point out the following questions if it turns out that they are ignoring the issue:

- which modules are responsible for maintaining system state and coordinating sequences?
- where are signal conversions (scaling sensor actuator signals to/from engineering units) made?
- do the modules in combination fully cover the specified functionality?
- is there overlap between module specifications?

Refining module specifications, module development and unit testing



Everyone works on their modules independently, but the architect and project manager have additional responsibility:

Architect ensures that the refined module specifications remain consistent and are detailed enough to enable a judgment that the modules can be integrated.

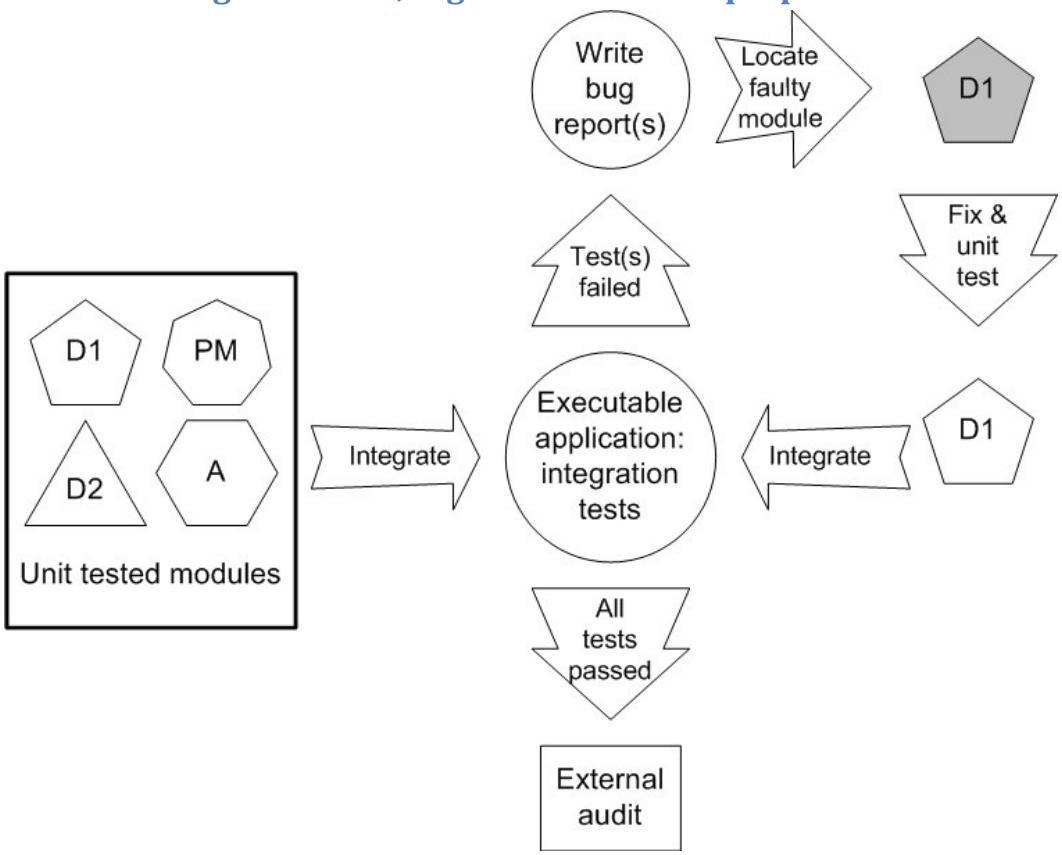
Project manager makes sure that everyone is correctly using the templates for module specifications, unit test specifications and test reports, so that the evidence and documentation required for the audit is generated.

NOTE: formal and documented unit test is not required for all modules, but only for most complex/critical modules. It is enough that each developer role has one module which will be formally unit tested.

Everyone's cooperation under project managers and architects leadership is essential.

Role of PBL instructor: same as in the previous phase. In addition, watch that the roles are being carried out properly and if necessary give advice or enforce the authority of a architect or project manager role.

Software integration test, regression test and preparation for final audit



Note: if an integration test fails, the team may choose either to:

1. Stop testing and begin fault location and fixing.
2. Continue testing in order to find more bugs before starting to fix. Then work as a team to locate the bugs. Then fix each bug one at a time.
3. Continue testing in order to find more bugs before starting to fix, and then make full use of the team resources by fixing bugs concurrently.

In either case, full regression test (i.e. repeat all integration tests) is implied by the central circle. This is because a fix can break code that previously passed a test. Option 1 is easiest as far as teamwork and project management is concerned, but has the following drawbacks:

- heavy integration testing
- hard to fully employ the team for fixing one bug

Option 3 can avoid these problems, but requires good communication in the team especially if several bugs require changes to one module. In this case, option 2 is recommended. Option 3 is recommended only if faults seem to be isolated to different modules, so that each module developer can take full responsibility for one or more faults.

Role of PBL instructor: offer *context aware* advice on choosing between options 1-3 above. Have informal audits with the project manager and give advice on preparing the materials for the audit.

Workload calculation

Work type	Time reserved
Thu sessions: lectures, supervised group work, audits 10x4h	40h NOTE: 4h is upper bound for each session. In some cases it is significantly less.
Independent activity related to contact sessions (preparation, reflection, review)	73h
PC classroom sessions: 10x2h	20h
Peer audit and giving written feedback on group dynamics and the course in general	2h

Total: 5cp and 135 hours. Independent activity is intentionally slack, since I consider this kind of workload realistic if a student is required to take 30cp per semester.

Evaluation

Evaluation method for determining team score

The team performs a PBL project, and the evaluation method for the team is **audits**, which will be performed for intermediate and final results of a project. The audit is similar to a demo, but the emphasis on adherence to process is specific to auditing. Also, since even the evaluation situation should contribute to learning, the experience from being audited is a professional competence that is developed on the course. Evaluation criteria are:

- Working software application, judged by *coverage* of test results presented and systematic employment of coverage techniques of equivalence classes and boundary value analysis
- Adherence to process, demonstrated by appropriate documentation

Evaluation method for determining individual score

Peer audit among team members. Each member fills a paper form which will only be seen by the teacher. Students only see the average score given by the team members.

- Students rate each of their team members on scale 1-5 according to following criteria:
- Shows up on time or presents good reasons for being absent well in advance (too many absences prevent passing)
- Good balance of being active and listening to others
- Behavior in conflict situations:
    - Willingness to seek compromises if there are alternative feasible proposals within the group and/or:
    - Appropriate use of architect/project manager authority to move the project forward
- Supporting integration of individual contributions to common result according to role

- Awareness of project issues: schedule, constraints, scope, resources, documentation, requirements

Justification

Since the evaluation methods of the PBL course do not significantly burden the students or teachers, and since there is no exam, the time of teachers and students can be focused on the PBL project, which is ambitious. The learning goals of the course cannot be evaluated with an exam, since ultimately only a successfully completed project is evidence of meeting these goals. In addition to producing a working application, learning goals are heavily related to process and project issues, so audits are seen as the most appropriate evaluation method, which also promote development of professional competence. Peer evaluations with criteria published at the beginning of the course will influence teamwork into the desired direction and prevent foreseeable teamwork problems.

Collection and utilization of feedback

Collection

After the final demo, students will be required to fill in a peer audit form. This will also contain a text field, in which the student describes his/her experience of the group dynamics. On the other side of the paper, there are feedback questions for the course in general that are designed to encourage descriptions of perceived problems, good practices and ideas for further development. Filling this form and handing it in person during the demo is an excellent way of ensuring that good feedback is obtained from every single student, and I have good experiences of similar arrangements on other courses. A 100% answer rate of rich textual feedback is drastically better than what anyone obtains from electronic submission systems.

I want to have students seriously fill in this one form instead of asking for several different feedbacks (e.g. the paper form and palauteOODI), since it is not reasonable or practical to expect that students will give rich feedback more than once. I do not ask for number ratings for the course or teaching staff, since from my earlier experience with qualitative feedback on other courses, I can say that this information is more accurately obtained in textual form which includes the justification. I am against number ratings for courses and individuals, since a bad score cannot always be attributed to the course or teacher (e.g. mandatory and optional courses are not equally disposed to negative feedback; the course content might be incoherent due to reasons outside the control of the teacher; PC-classroom problems are often due to unsatisfactory quality of service by the computing center). If positive and negative feedback needs to be directed at individuals, this can only be accomplished based on qualitative feedback. One issue with feedback collection and processing is that it must be possible to refine the feedback for the purposes of decision-making institutional levels that cannot be bothered to read what the students actually wrote. Instead of numerical averages, I propose to provide an objective and concise textual summary of the feedback as described in the next subsection.

Utilization

The raw textual feedback will be processed by teaching staff and two other members, who bring a broader perspective and objectivity to the process. These members are one of the professors in charge of the module and a student representative holding a position of trust related to education. These will read through all feedback, focusing especially on recurring issues and justified critique. They will formulate a concise listing of successes, problems and their causes as well as a general evaluation of the course and

interaction with teaching staff. Development ideas are weighed against learning goals and available resources. This group will produce a compact text that is representative of the feedback and which can be used at other institutional levels which might take an interest in judging or changing the course. It is my passionate opinion that even this higher level needs some information about *why* a positive or negative rating was obtained, and what were the successes, problems and their causes, if informed decisions regarding the future of the course and its teachers are to be made. I also believe that this arrangement is a good way of involving student representatives in continuous and constructive development of teaching.