# EXPERIENCES FROM A MULTI-CULTURAL DESIGN, BUILD, TEST PROJECT

**Thomas Gustafsson**

School of Engineering, Jönköping University, Sweden

**Adam Lagerberg**

School of Engineering, Jönköping University, Sweden

## ABSTRACT

To train students and prepare them for industrial projects, where two key elements are communication and planning, we expected a Design, Build, and Test (DBT) project would use available staff efficiently and at the same time give the students a multi-disciplinary view on embedded system development, which is difficult to give in our normal curriculums.

The students taking part in the project have varied cultural background. The project has been given two times and we report our experiences from running the project and trying to adapt it based on experiences from the first time.

Identified key experiences are: it is difficult to motivate the students to work independently in the project, it is difficult to communicate what is a successful project (i.e., the focus should be on how the group is working and not on technical solutions), and model-based development.

## KEYWORDS

DBT project, fellow student assessment, multi-cultural project

## INTRODUCTION

At School of Engineering, Jönköping University, Sweden, a 2 year Master's programme in Embedded Systems is given. The programme is designed based on the CDIO Syllabus (1). Admitted students should have an undergraduate degree in electrical engineering that corresponds to a Bachelor degree. The admitted students have varied educational background, as well as varied cultural background. The students come from north Africa, middle East, east Asia, Europe, and north America.

As part of the programme is a linked project (2) where the course work of two concurrent courses are joined and merged into a Design, Built, and Test (DBT) project. The two courses are Mechatronics and Software Engineering. Mechatronics and Software Engineering constitute two key competencies that are needed in many industrial projects in the embedded systems area.

Designing a project around the development phases design, build, test is not uncommon, and experiences of such activities have been reported elsewhere (3) (4). In this paper, we discuss our experiences with executing such a project with students from many different backgrounds.

By using a linked project of the two courses, we feel that the following from the CDIO Standards (3) are addressed: Standard 3 Integrated Curriculum, Standard 5 Design-Build Experiences, Standard 6 CDIO Workspaces, Standard 7 Integrated Learning Experiences, Standard 8 Active Learning, Standard 11 CDIO Skills Assessment.

In the DBT project work, the students were grouped by the supervisors and then the groups worked independently with product development. The supervisors gave the groups written requirements of two products, where the groups should develop the software for these products.

The DBT project has now been given two times. Based on our experiences from the first round, we made adaptations to the DBT project. Some of the adaptations addressed issues that we believe originates in different learning cultures of the students. We report our first time experiences and the experiences from the second round in the paper.

The conclusions of the paper are that (i) it is hard to make some of the students work independently and creatively with a problem (this may be due to educational culture in their undergraduate studies), (ii) the question 'when is a DBT project successful?' must be well communicated with the students before and during the project, and (iii) working with model-based development of controllers requires a physical system where it is possible to determine control parameters within a short time frame (the project is run during 10 weeks).

The outline of the paper is as follows. The setup of the DBT project is given in Section PROJECT DESCRIPTION. The experiences from round one is described in Section EXPERIENCES ROUND ONE. Section ADAPTATIONS FOR SECOND ROUND discusses the changes that were made for the second time the project was given. The Section EXPERIENCES FROM SECOND ROUND discusses our experiences from applying the adaptations. The paper is concluded in the Sections DISCUSSION and CONCLUSIONS.

## PROJECT DESCRIPTION

### *Project Setup*

As described in Biggs' 3P model (4), it is important to reason about teaching and learning before and during such activities, but also what is the learning product of the activities. Biggs also describes two student types; the Susans that are highly motivated deep learners and the Roberts that are less motivated surface learners. The teacher can be on three 'maturity' levels (i) blaming the student, (ii) what the teacher does, and (iii) what the student does.

We strongly believe that taking an approach of continuous development of courses and choosing a teaching approach that suites the subject best, helps making less motivated surface learners more motivated, and hopefully deep learners. The DBT project being discussed in this paper constitutes the practical course work of the two courses Mechatronics and Software Engineering. A project work is well suited for Software Engineering as this course teaches development processes, and Mechatronics touches upon many problem areas that are relevant for a programme in embedded systems.

Our goal with the DBT project is for students to (i) familiarize themselves with a product lifecycle and a systematic approach to working with (software) development, (ii) reason about their own performance in a bigger context, and (iii) understand the need to plan the work and the benefits to reason early about architecture and design. The systematic development approach should also contain a model-based approach to development.

Thus, we should consider the DBT project to be successful if the students have been able to work according to a systematic development process and reflect upon the successes and failures of the group's performance.

The project task is for each student group to develop and deliver two products:

1. A software for a control equipment that can control a pendulum system such that the pendulum is balanced in a hanging down or inverted position.
2. A software that shows the real-time position of the pendulum system. The software should also be able to log pendulum positions to a file and later play these back.

The courses are taught concurrently during one segment of 10 weeks. The courses constitute 400 effective hours together and 40% of the time should be allotted to the practical course work, i.e., 160 hours. Thus, each student is expected to put in approximately 160 hours of work in the project.

As the curriculums are formulated, each course is supposed to have an individual grading, which usually is based on a written examination. The practical course points of each course have only the grades pass or fail. However, we feel the project so important for the learning of the students, that we wanted each student's performance be reflected in the course grades. We achieved this by fellow student assessments and grading of each group's performance based on each artefact they delivered. The grading is discussed further below.

The students were randomly grouped into project groups of 5-9 students by the supervisors. The reason such a grouping method was used was to simulate that you seldom, in industrial projects, have the possibility to select your own colleagues.

The theoretical knowledge necessary for carrying out the project work was given in the theoretical parts of the courses. The Mechatronics course teaches

- Control theory
- Sensors
- Actuators

Software Engineering course teaches

- Software development processes
- Roles
- Risk management
- Software requirements elicitation
- Software architecture and design
- Software testing

The supervisors prepared written requirements of two products that each group should deliver at the end of the project. The supervisors also required deliverables in the form of documents and presentations during and at the end of the project. Documents were (i) artefacts of the project work and (ii) individual documents concerning the learning outcomes. Examples of the first kind are software requirements specification and software test report. Examples of the second kind are self reflection and fellow student assessment.

Each submitted document and given presentation was graded by the supervisors. The first few artefacts could be resubmitted if the group wanted to address comments and get a higher grade.

Two pendulum systems with electronics hardware enhancements were available at project start. The enhancement is an FPGA that converts angle and position sensor signals to a form readable by a National Instruments I/O card. Furthermore, the FPGA can also communicate the sensor readings to an LPC 2294 board using SPI. The following sample code was given at project start:

- C code implementing real-time operating system with tasks reading sensors using SPI. The sensor readings are also transmitted to a PC using RS232.
- Simulink project reading sensor data from National Instruments I/O card.

The project groups choose a development process and they defined different roles that were allocated to different group members. Moreover, each group also created and maintained a project management document including time plan and risk management.

### *Grading*

Each group's performance was derived by taking a weighted average of all graded artefacts they delivered. Since quite a few documents and presentations were prepared and given, each student has been involved in at least one document and presentation. Thus, we believe that the average quite well matches the overall performance of the group.

Each student was also requested to do a fellow student assessment, where they graded all students in their group. The average grade of a student's assessment should be 60 on a 0 to 100 scale.

## EXPERIENCES ROUND ONE

### *Specifics of project setup*

The first time the DBT project was given was January to March 2008. 16 students followed the Master's programme in embedded systems. 6 students were exchange students only taking the Software Engineering course, and the groups with these students were instructed to allocate half the workload to these students. In total three groups were formed of 6-9 members each.

The following deliverables were required from each group:

- Project management document (2 submissions)
- Software requirements specification (2 submissions)
- Software architecture for embedded control software
- Software design for embedded control software
- Software test specification for embedded control software
- Software test report for embedded control software
- Software architecture for GUI
- Software design for GUI
- Software test specification for GUI
- Software test report for GUI
- Software requirements tracing document
- Hand-over documentation describing how someone could setup development environments
- Presentation of project management
- Presentation of architecture
- Presentation of products

The groups split the work by defining several roles, typically project manager, architect, designer, implementer etc., and constructed a time plan outlining the work of the group. Each group was supposed to have one meeting every week where its assigned supervisor attended.

The groups were supposed to use model-based development by first modelling the pendulum system using Matlab/Simulink followed by a design of a controller. This controller should then be transformed to the embedded control software and implemented in C code.

### *Experiences*

Several problems occurred during the project as described below.

- The students have significant differences in both cultural background but also learning background. Some students where high performers (deep learners) with a high level of independence and creativity, while others seemed most accustomed to follow instructions (surface learners) and did not have a high level of independence and creativity. We believe a DBT project can suite both 'learning styles' and the high performers excel and train the others in being more independent and creative. However, it must be enforced that the high performers do not do the work of the others.
- Connected to the above bullet, we also observed that there is a large difference in students' preconception about embedded system development in general and specifically software development. All aspects of software development cannot be covered in detail in the Software Engineering course, which means that the students must be motivated to try to do some read up on their allocated role's tasks by themselves. Moreover, the students may also need to read up on technical details by themselves. This is a good opportunity for them to get experience in such activities as they are expected to be able to work independently in industry and in their master's thesis.
- All groups were unable to construct a model of the pendulum system that was usable for constructing a controller that could work with Simulink in continuous time with the National Instruments I/O card. The students understand the concepts behind constructing a plant model and connecting a controller to it, but problems seem to be related to difficulties in determining plant parameters.
- All groups were significantly delayed with implementation of the embedded control software. This is because the groups focused on deriving a correct model before looking at and trying to use the embedded software development environments. Therefore the deadline for product delivery was extended by one month.
- Related to the above bullets is that everyone expects a person having a master of science degree to be able to understand the context of a problem, divide it into subproblems, and work on the identified subproblems until the problem is solved. Our experience is that students, even though studying on a master's programme, are unable to perform a systematic way of solving a larger problem. Students' learning background has thus been lacking in giving them experiences in understanding a larger context and identifying subproblems. The DBT project as such, is a way to mitigate this, but as the project is given in a limited amount of time, a certain degree of independence at project start is needed. Perhaps the students need to be prepared better in courses given before the project.
- The vast amount of documents each group was supposed to prepare also seems to have delayed the work on the solutions. This, however, is not a problem per se, as the purpose of the project is to work in a systematic way with product development.

- The documents, all with a deadline, forced the students into working in a waterfall like process since the documents should be delivered in the order specifications, architecture and design, test specification, and test report.
- The question ‘when is a DBT project successful?’ is important and it happened that both some supervisors and the students felt the DBT project was partially a failure. The students found the project a failure because they were not able to deliver a working implementation. However, course evaluations show that the students liked the idea of working systematically in a setting simulating an industrial project. Some supervisors found the project a failure because they felt there was a too big focus on imposing a certain workflow on the students (waterfall process model including the documentation).
- Some students seemed uncomfortable in assessing each other’s performance and being assessed. We believe this is due to a cultural background where they are not used to being assessed by fellow students.

As can be understood from the list above, many of the supervisors’ perceived problems stem from students’ motivation and learning style (deep or surface).

## ADAPTATIONS FOR SECOND ROUND

Based on the experiences acquired during the first round of the DBT project, several changes were made to the project setup. The changes are described in the bullets below:

- Independence and motivation: based on our experiences from the first time the project was given, we are concerned that students may not be working with their allocated tasks for the following reasons (i) they are not accustomed to working independently, and/or (ii) they have other motivations than studies to move to Sweden and have therefore less motivation to spend the necessary time to work in a project, and/or (iii) they do not have the background to work with the task and they are not trained in independently finding information and read up on the subject using the found information. The problem of motivating students is thoroughly discussed by Biggs (4). Since the supervisors have relatively little insight how the work is actually performed by the group, we choose to address this by simulating industry where each group had to set up a time reporting process and each member had a limited number of hours to put into the project. Furthermore, each group must include the risk ‘student is not doing allocated tasks’ in the group’s risk management. The idea being that the students would motivate each other to do their allocated work.
- When is a DBT project successful?: as stated in the section above, some supervisors did not feel the DBT project was successful due to the focus on documents and the development process the delivery of the documents enforced on the groups. Thus, the amount of documents that needed to be handed in was reduced. Also, some new deliverables have been introduced: presentations of working subsystems. Our idea is that a smaller number of documents may give students a bit more time to focus on solving the problems and the focusing on delivering working subsystems would enforce an iterative development process. Modern software development processes, e.g., agile methods, are iterative.
- Model-based development: by showing the students, and motivating them to use, Matlab/Simulink for plant modelling and control algorithm construction, we hoped the students would get an idea how model-based development works in practice. As we reported in the section above, this was a failure since the students never succeeded in deriving control algorithms that worked. We therefore reformulated the problem to reducing the oscillations of a disturbed free hanging pendulum. It should be easier to find parameters of a free hanging pendulum than the inverted pendulum, since the system is more ‘forgiving’.

We have not adapted the grading scheme as we felt it worked well the first round.

## EXPERIENCES FROM SECOND ROUND

### *Project Setup*

The second round started in January 2009 and ended in March 2009. The supervisors prepared a new set of documents with the requirements of the products and the subsystems that should be presented during the project.

The deliverables requested from each group during second round are listed below:

- Project management document (2 submissions)
- Software requirements specification (2 submissions)
- Software test report for GUI and embedded control software
- Hand-over documentation describing how someone could setup development environments
- Presentation of project management
- Presentation of model of free hanging pendulum and of software for control of free hanging pendulum
- Presentation of products

There are 18 students and they have been divided into three groups of 6 members each by the supervisors. When the courses start, the groups are supposed to decide on a development process, define roles, assign roles to group members, start planning the work in a time plan, and also start the risk management.

The project groups were supposed to be working in the same way as in round one where a model should be developed for the plant and then a controller should be constructed and simulated.

The presentations were scheduled to be given 3 and 5 weeks into the project, and at the end of the courses. We hoped that these subsystems that should be ready would help the groups to focus on smaller parts of the problem so that they would not be as delayed as in the previous round.

### *Experiences*

First we discuss what problems occurred during the second round and then we relate them to the adaptations that were introduced.

- The independence issue arose again. The problem seems to be accentuated during the second time the project was given.
- The groups were again unable to derive working controllers even though the free hanging pendulum should be more 'forgiving' when it comes to parameters of the controller.
- The groups were unable to stick to their time plans, which was expected. However, as in round one, the groups were very focused on getting a working plant model and controller, which hindered their progress in other activities even though they had planned independent activities. We do not know if this behaviour boils down to pure stubbornness or due to learning background, but since it has happened twice now, we start to believe it is learning background related. This means that the students are lacking in experiences in dividing a problem into subproblems and working on these independently.

- It seems, paradoxically, that the reduction of documents made it a bit harder for the students to clearly see the benefits of a systematic development approach. A systematic development approach would 'reduce' the effects of cultural and learning background differences.

The adaptations for the second round were in the areas independence and motivation, 'when is a DBT project successful?', and model-based development. Below we discuss how the adaptations worked out.

- Independence and motivation: the supervisors feel a higher percentage of the students are not serious about the project during the second time compared to the first time the project was given. The groups should include the risk 'student is not doing allocated task' and use a time reporting process. The following were observed during the project.
  - The groups misunderstood the formulation and never monitored the work of the project members. If the group allocated a task to two or more group members, all of them should try and contribute equally and not rely on one of them to deliver, since the project is given in the context of a learning activity. The supervisors observed that it occurred during the project that if two members were supposed to do a certain task, only one of them did the work.
  - On direct questions to the project leaders how the members were doing with respect to worked hours, the answer was always that the members worked according to allocated hours. This is very unlikely, as some tasks must have taken longer or shorter amount of hours. Our interpretation is that the group members covered for each other, and that it was accepted that some members worked less than others. We believe that some of the students were working part time and prioritized the paid work over the project work. In the fellow student assessment some students got low assessments, which we think reflected the fact that some students did not do their allocated work. We also noted that for some students it was very difficult to confess that they could not do a specific task or did not put in the number of hours requested. We believe this is partly culturally affected, since in some cultures one cannot lose face.
- When is a DBT project successful?: since there were few deadlines during the project for documents to be submitted, no particular development process was enforced upon the groups. However, the presentations the groups were supposed to give gave a small focus on iterative and incremental development. It seems the students realized this. Removing some documents ended up in that some parts were not documented at all; removing the software architecture document gave the effect that no formal software architecture was developed before the software started to be implemented. There is thus a trade off how much documentation should be part of a course like this.
- Model-based development: as mentioned above, the groups were unable to construct working controllers also the second time the project was given. In order to have meaningful tasks for all roles, most notably the testers, the supervisors prepared a fuzzy logic controller implemented in C code that almost worked in the inverted pendulum case. This code was distributed to all groups and they could manipulate it for the free hanging pendulum case and the inverted case.

## DISCUSSION

The DBT project has enabled the students to reason about embedded system development, working in teams, and project management.

For the Software Engineering course lectures, the experiences the students acquire in the project during the project are valuable and can be used as a common shared understanding where new concepts can be introduced. However, this means that some lectures inevitably need to be prepared solely for each time the project is given since the preconditions, e.g., encountered problems by project groups, may be different from time to time.

The anticipated, by the supervisors, main problems in giving the DBT project as a learning activity are

- Give the students the conditions to execute the project as a learning activity, because the students have so different cultural and learning backgrounds, and many of them do not have the background skills in designing computer programs and implementing them.
- Motivate the students to work independently with allocated tasks since, as it seems, some of the students have other motivations to move to another country than to study. Many of the students need to work to finance their studies, which may have a negative effect on the involvement in a time consuming project.

## CONCLUSIONS

Before starting the project the second time, several changes were made to the project setup. One change worked out for the better, namely, reducing some documents the project groups have to submit as this reduces the effect of a specific process (the waterfall process) on the project group's work.

The main problem, which has not been solved by our adaptations, is motivating the students to work hard in the project and independently on their allocated tasks.

## REFERENCES


1. **Crawley, Edward F.** *The CDIO Syllabus.* s.l. : cdio.org, 2001.
2. *CDIO: An international initiative for reforming engineering education.* **Berggren, Karl-Frederik, et al.** 1, 2003, World Transactions on Engineering and Technology Education, Vol. 2.
3. *Engineering design and rapid prototyping: a rewarding CAD/CAE/CAM and CDIO experience for undergraduates.* **deWeck, O.L., et al.** 2005. 1st Annual CDIO Conference.
4. *Integrated assessment of disciplinary, personal and interpersonal skills - student perceptions of a novel learning experience.* **Edström, Kristina, et al.** 2005. Proceedings of 13th Improving Student Learning.
5. The CDIO Standards. *CDIO.* [Online] [Cited: 9 March 2009.] http://cdio.org/tools/cdio_standards.html.
6. **Biggs, John.** *Teaching for quality learning at unversity.* s.l. : Open University Press McGraw-Hill, 2003. 978-033521168-5.


### *Biographical Information*

Thomas Gustafsson is an assistant professor in computer engineering. He teaches several courses at the Master's programme in Embedded System. His research interests are real-time systems and model-based software development.

Adam Lagerberg is an assistant professor in automatic control. He currently teaches courses in control and mechatronics. His research interests include: Control applications, computer simulation and dynamic system modeling.

***Corresponding author***
Dr. Thomas Gustafsson
School of Engineering, Jönköping University
P.O. Box 1026
SE-551 11 Jönköping
Sweden
+46(0)36101596
thomas.gustafsson@jth.hj.se