

LESSONS LEARNED FROM DESIGN-BUILD-TEST-BASED PROJECT COURSES

J. Malmqvist, P. W. Young, S. Hallström, J. Kutteneuler and T. Svensson

Keywords: design-build-test, engineering education, project-based course

1. Introduction

Project courses in which students design, build and test a device on their own are increasingly being used in engineering education. The reasons include that such projects do not only train design skills but can also be exploited in order to increase student motivation, to give an improved understanding of engineering science knowledge and to practice non-technical skills such as teamwork and communication. However, design-build-test (DBT) experiences may also be costly, time-consuming, require new learning environments and different specialized faculty competence.

The proper set-up of a DBT educational experience generally requires consideration of a large number of factors - identifying goals, selecting projects, budgeting, and so on - that differ from those in classical educational situations. We therefore see a need for a better understanding of DBT-based learning experiences, along with guidelines for their design. However, scientific publications on the topic tend to describe a particular course or learning environment (confer, for example Sullivan & Watkins, 2000, Elger *et al.*, 2000, Andersson and Palmberg, 2003, Jansen 2003) and there is a lack of larger-scale investigations that in a more systematic fashion analyze the outcomes of DBT learning experiences. This has motivated us to survey a number of DBT experiences and to use the results to develop a set of guidelines for designing DBT experience, with particular application to advanced-level project-based courses.

The objectives of the present paper are thus to

- provide a definition of the concept of a design-build-test educational experience
- identify benefits, limitations, enabling conditions, and challenges in the area
- develop a database of design-build-test experience data as a foundation for long-term research
- summarize and generalize experiences from design-build-test experiences
- state guidelines for the design of design-build-test educational experiences

The presented work constitutes a part of the CDIO project (Berggren *et al.*, 2003, CDIO Initiative Homepage, 2004), an international initiative that aims to develop a new model for engineering education, characterized by using the process of conceiving-designing-implementing-operating, i.e. the product lifecycle, as the educational context. Design-build-test projects are a key element of this new educational model.

The remainder of the paper is structured as follows: In section 2, we discuss the methodology used in our survey, including a listing of courses surveyed and discriminators used. In section 3, we discuss our findings, including some basic definitions and we highlight some key observations. Section 4 presents a set of guidelines for designing DBT experiences and section 5 lists conclusions and proposals for future work.

2. Research approach

The study has been of explorative character, the aims being to identify similarities and differences between design-build-test based courses, to develop a terminology supporting discussions on the topic, and to state guidelines for development.

The investigations started by conducting a number of workshop-type meetings aiming at developing basic definitions, and at identifying benefits, limitations, challenges and enabling conditions for design-build-test experiences. These workshops also served to identify a number of key discriminators for DBT experiences that constitute the basis for the survey and more in-depth analysis.

In the survey, DBT learning experiences have been classified using 50 different discriminators, grouped into 9 categories. The discriminators are intended cover all relevant aspects of the design-build-test experience and address issues such as classifying the learning objectives, the position in the curriculum, the characteristics of the design-build-test task, the student team composition, the teaching and learning strategies, assessment and the outcomes from course evaluations, statements from industrial customers and other data. The full listing of discriminators is shown in table 1.

The surveyed courses are run at Chalmers University of Technology, Linköping University, Massachusetts Institute of Technology, and the Royal Institute of Technology (KTH). See table 2. The courses are given for the 3rd or 4th year of a 4½-5 year education leading to a master's degree, or for the senior year of a bachelor's degree, i.e. the DBT experiences are of an advanced level, requiring advanced technical pre-knowledge. For discussions on the role of design-build-test experiences early and across the curriculum, see Gustafsson *et al.* (2002) and Brodeur *et al.* (2002).

Table 1. Design-build-test experience discriminators used in survey

Course basic facts	Design-build-test task	Teaching & learning methods
Name	Number of design-build tasks	Teaching techniques
University	Type	Topics addressed by lectures
Year started	Project planning	Other topics addressed by lectures
# Students	Project selection	Support & intervention
Fraction female students	# Alternative projects	Industry & outside involvement
# Student working hours	Re-use	Teaching materials
Relative cost	Duration	Staff
Course documentation, website	Starting point	Assessment
Other relevant information	End result	Grades
Learning objectives	Deliverables	Individual assessment
Primary	Complexity	Team assessment
Secondary	Typical # parts	Team grade distribution
Curricular position and links	Key technologies	Assessment of cognitive skills
Student year	# Key technologies used	Assessment of affective dimension
Explicit links to other courses	Fraction of work in course	Fail management
Implicit links to other courses	Student teams	Evaluation
Learning environment	Size	Student evaluations
Description	Selection	Teacher evaluations
	Composition	Industry evaluations

Table 2. Design-build-test based courses surveyed

Course	Project	Course	Project
Mechatronics project course	Autonomous vehicle	Biomedical engineering	Laser scanning device
Product development	Company-specific projects	Systems engineering	Company-specific projects
Formula Student	Small racing car	Control project laboratory	Laboratory control processes
Vehicle engineering	Solar-powered aircraft	System design	WLAN physical layer
Vehicle engineering	Waterbike	Computational physics	Application-tailored material
Mechatronics, advanced course	Company-specific projects	Space systems capstone course	Magnetic attitude/position control
Electronics project course	Computer controlled machines/robots		

3. Results

In this section, some theoretical background is provided in the form of a proposal for a definition of a design-build-test-based learning experience. Then, selected findings from the survey are discussed.

3.1 What is a design-build-test learning experience?

In this paper, the follow definition is proposed:

A design-build-test experience is a learning event where the learning takes place through the creation of a product or system. The product that is created in the learning event should be developed and implemented to a state where it is operationally testable by students in order to verify that it meets its requirements and to identify possible improvements.

Design-build-test experiences can range from “basic” (usable in freshman courses) to “advanced” (requires advanced-level technical pre-knowledge) and be used not only to train the core engineering professional skills of system design and implementation but also to serve as a platform for training non-technical skills such as teamwork and communication, to create an understanding for interdisciplinary issues, to reinforce disciplinary knowledge, or to motivate students for engineering. The product can be built of hardware, software, a combination or even a digital model. The media that the product will be built of needs to be carefully chosen but this does not mean that it has to be very close to final product status. Depending on the level of the course it can be a simple functional model or a complex near production-status prototype as long as it meets the basic requirement of being operationally verifiable and thus providing direct feedback to the students.

3.2 Survey findings

Course basic facts

The number of students in the studied courses varied from 7 to 123 students, with a median of 16. The courses are given for students in the 3rd and 4th year of their engineering education, i.e. they have engineering knowledge that can be applied in rather advanced projects. The courses are quite new; With two courses as exceptions, all have started after year 2000. The number of student working hours ranges from 200 to 1170, with a median of 300.

Learning objectives

Concerning the primary learning objectives, the most common are (a) to train conceiving, design, implementing and operating skills in a realistic environment and (b) to give insight into multidisciplinary aspects of engineering systems. As secondary learning objectives some mention strengthening of disciplinary knowledge. It is clear that these learning objectives are vital engineering professional skills that are tightly coupled to the design-build-test task and would be difficult to teach in any other way. Thus, there is a need for these activities in the curriculum.

A specific consideration for these courses is the need to distinguish between success in achieving curricular (learning) goals and objectives, and between “technical progress” success. Some would argue that the first set is more important than the second, i.e. it is possible to have substantial learning benefits even if the project is not a complete success as regards the function of the product.

Design-build-test task characteristics

A majority of the studied projects aim to design, build and test (ground, air or sea) vehicles, based on mechanics, electronics and software, i.e. they are multidisciplinary systems and bring forward integration issues. The problem-solving is of rather technical nature, a “function” is to be realized. A few courses also consider wider product development issues, such as industrial design, ergonomics and manufacturing.

Designing the task for a design-build-test project is a crucial task. It typically takes most of the time that the students devote to the course and informally defines the “real” course requirements, overriding contents of course memos and even lectures. In the end, students will often be assessed based on their product, although examples where the process is assessed are also present among the studied courses. Further, finding and renewing design-build-test tasks are challenging and time-consuming activities for the responsible teacher. The tasks need to have start and end conditions corresponding to the learning objectives, they should preferably be sponsored by a company, the task statement must leave room for alternative solutions and so on. One specific issue that needs to be considered is finding the adequate level of difficulty for the design-build-test experience. A too difficult task may result in an impressive product that is essentially teacher-designed, with student as “implementers”. A too simple task may not promote motivation nor build the self-confidence that results from having met a challenge, which are two of the most significant benefits of design-build-test projects. The students’ high involvement in the design-build-test task can also make them down-prioritize other courses that are given at the same time. The task needs to be carefully thought through and student time spent on the project monitored in order to keep the balance.

Curricular position and links

Viewing the education as a whole, it is also important to realize that one design-build-test experience, no matter how well-designed it is, cannot cover all relevant aspects. There are many dimensions to design problems and all aspects cannot be considered at one occasion. Designing, like any other skill, requires practice. A more reasonable strategy is to include a sequence of design-build-test experiences in the curriculum and to have a systematic plan for variation across these. One project could emphasize creativity, manufacturability, a third multidisciplinary integration issues and so on. Among the studied projects, we also see examples where disciplinary knowledge is taught in the context of a design-build-test experience, and where the DBT is seen as an “umbrella” for several disciplinary knowledge courses.

Teaching and learning methods

Teaching practices in the studied courses are typically based on a limited number of lectures, and a higher fraction of tutoring. Industry guest speakers are common. The course literature is to a high degree based on lecture notes; adaptation to the specific course at hands seems to be necessary. Product development textbooks such as Pahl & Beitz (1996) and Ulrich & Eppinger (2000) are helpful but do not cover the entire process. First, they provide relatively little guidance concerning the building and testing of the design. Moreover, software, which is an integral part of many of the studied design-build-test projects, is not treated at all in these textbooks. Software engineering textbooks, such as Sommerville (2001), consider these aspects better but, on the other hand, do not address the design of mechanical sub-systems.

A changed view on the teacher’s role is declared in many of the DBT courses, where the authorian position is less pronounced and the teacher instead becomes more of a mentor or coach. This enables a less constrained learning environment where the students dare to discuss, reason, and explore issues with support from the teacher, rather than being lectured.

A generally problematic issue is to acquire faculty competence to teach design-build-test experiences. Currently, a low fraction of the faculty in a typical engineering department have personal practical experience from developing a complex system and many of the courses surveyed are highly dependent on one or two individuals. Along with introducing design-build-test experiences major faculty educational and recruitment activities are needed to ensure a stable basis for sustainable operation. An effective strategy, used by some schools, is to use graduate teaching assistants who are involved in funded research that have similar goals and objectives as the undergraduates’ DBT coursework. This approach supplies valuable technical assistance to the undergraduates while also accomplishing pre-established graduate research goals. Another approach is to seek out and use technical advisors (on a no-direct-cost basis) from industry groups with interests in the students’ projects.

Team composition

One of the discriminators that show greatest variation is that of team size and composition. Team sizes vary from 3 to 28, and team composition is done by a variety of methods: student self-selection, teacher selection etc. While there is no “right” team size it must be understood that the team size has a major effect on the learning outcomes. A small team size (3-4) implies emphasis on technical problem-solving, avoids student over-specialization and forces students to be self-propelling. Large team sizes (>10) imply emphasis on project management and communication and enables student teams to take on more realistic, complex and multi-disciplinary tasks. This makes such design-build-test tasks become more authentic, industry-like experiences. In large student teams, there is a risk for unwanted student specialization, in other cases such specialization may be desired. This dimension needs to be considered when planning the curriculum for the whole educational program so that DBT experiences with different team sizes and compositions are planned, similarly to exposing students to a range of DBT tasks with different characteristics.

Assessment

A commonly cited educational truism is that of “what you assess is what you teach”. This applies equally well to DBT-based learning experiences. One particular difficulty here is to concurrently assess product and development process deliverables. What grade should be given for a functionally excellent technical solution but where the student team fails to document its development process? Or what grade should be given to a mediocre technical solution backed up by evidence of a systematic process? Any teacher responsible for a design-build-test-based course needs to do these trade-offs. However, there exist no common, accepted, assessment methods for such experiences and many teachers find this to be a serious challenge. This is apparent from our data. There is a wide range of assessment techniques and principles applied: some based on product results, others on documentation. In some courses, individual grades are given, other give different grades to different student team members. There are examples of peer assessment but also of teacher assessment. In some courses the students give written feedback to their peers, both on individual behaviour and technical contribution. Feedback and assessment are sometimes given half-way into the course to allow for students to react and improve before the final grading. There are also differences between DBT assessment practices at the same university. More research is needed in this area, to find suitable assessment practices and to make them known to faculty.

Learning environments

The introduction of new design-build-test experiences also brings along a need for new learning environments enabling students to manufacture mechanical parts, assemble circuit boards, code and load software etc. For all courses studied, new learning environments have been developed. There are large variations in purpose, facilities, equipment and investment but a detailed discussion of this lies outside of the scope of this paper. For more details, see Hallström *et al.* (2004).

Evaluation

It is clear from the survey data that design-build-test experiences have many benefits. In addition to training design-build-test of products and systems, they are perceived by students to be fun and motivating and to add realism to the education. They also give an overview of the development of a complete product. They further train students’ creative abilities and strengthen their self-confidence. From a teacher perspective, they stimulate learning of technical knowledge, connect theory to practice and illustrate couplings between subjects. From industry, there is evidence that students that have participated in design-build-test projects are very positively received and possess knowledge and skills that are highly demanded by industry employers.

The available data does not give quantitative proof of the success of these courses but is rich with qualitative statements that support this conclusion. Some examples are listed in table 3.

Table 3. Sample quotations from student, faculty and industry evaluations of DBT experiences

Students'	"I would just like to add that this is the most rewarding counsel have ever done as a student!" (KTH) "Creating the prototype raised the quality of the work" (Chalmers) " It is very good to get experience of running a project in an industrial way" (LiU)
Faculty	"The students are more motivated" (Chalmers) "The (vehicle engineering) project generated an urge for knowledge." (KTH)
Industry	"Excellent students and excellent work! Send more of that caliber!" (SKF-ERC, The Netherlands, about students from Chalmers Mechatronics specialization) Fourth-year design-build experience products at MIT reviewed by industrial experts were assessed as being comparable to professional design studies

Reluctance to include design-build-test experiences in engineering education is frequently rooted in suspicions that such experiences are highly resource-consuming. Indeed, new types of learning environments are needed, along with more teacher-intensive teaching methods, and new faculty competence. Cost-wise, our data suggest that design-build-test experiences cost on average 1.5 times that of a traditional course, with a span from 1.0 to 2.5. (This number includes increased teacher time and facility costs.) However, even in a developed mode where there is one major design-build-test experience per student year DBT experiences will probably only be about 20% of the education, resulting in a need for additional funding or budget re-considerations corresponding to 10% of the total educational cost. Further, examples, such as the "1.0" course, show that it is possible to design low-cost DBT experiences without compromising educational objectives. Another university's internal study of the return-on-investment of an advanced-level DBT project concluded that the success of an ambitious DBT project was a key factor in the university bidding for, and winning, follow-on research proposals. Their figures support a conclusion that there was a 6:1 cost benefit from this combined linkage of academic and research goals.

4. Design guidelines

It must be realized that developing and running design-build experiences is different and more complex than traditional course development and teaching. Success is not only related to the workspace or the design-build-test task, but to a combination thereof, adequate teacher support, and to the use of all this in a well-designed course. Based on our experiences from running design-build-test learning experiences and the survey data, a number of design guidelines for developing DBT experiences have been stated. The aim is to support teachers in planning and running DBT experiences as well as developing the learning environment needed. An abbreviated version of these recommendations is given below. For a detailed version including course examples, see Hallström et al. (2004).

Pre-course planning aspects:

- Start the development of a design-build-test experience/course well in time (preferably a year in advance)
- Do the development of the course in a team in order to get more ideas and to find possible traps
- Make a test run of the design-build-test experience on a single project group, prior to using for a large student group with many groups
- Make sure that all supervisors are well educated and are aware of the course goals and design
- Make a time budget of the course from a student perspective, and plan to track time during the course
- Try to find critical situations that can appear in the course, and prepare actions for these
- Plan for renewal of project ideas – this is a key challenge
- Create a project-dedicated space. This is necessary, at least for advanced-level design-build-test tasks
- Make connections between the course and other design-build-test based courses in the curriculum. Ensure that the courses build on each other and create variation around a common

core, and avoid repetition. Vary task characteristics and team composition but use similar assessment practices

Design-build-test task design

- Seek projects with an identifiable and involved customer. Customer involvement and interest is key to student motivation
- State learning objectives clearly - distinguish between product and learning objectives and keep focus on learning outcomes rather than the product to be designed. The CDIO syllabus (Crawley, 2002) can be used to form a basis for the learning objectives
- Set up the design-build-test task so that the stated learning objectives are taught and assessed through project deliverables, or as part of the process
- Carefully consider start conditions and end result, make sure these map well to the learning objectives
- State the design-build-test on the right level of difficulty. A too difficult task may result in an impressive product that is teacher-created, with student as implementers. A too simple task may not promote motivation nor build confidence from having met a challenge.
- Make sure that some amount of innovation is needed and that there are alternative solutions. The solution should not be obvious – training creative skills is a key element
- Make sure that students do the concept design, not the teacher or an outside expert
- Seek projects and set requirements that lead to designs composed of a number of identifiable sub-systems and/or work packages. This will facilitate for all team members to make an identifiable contribution to the project
- Provide all students with similar opportunities to develop their skills. Avoid student over-specialization, e.g. honing their skills as the CAD expert in a team
- Carefully plan the use of the design-build-test task to teach non-technical skills such as teamwork and communication and include these elements in the learning objectives and pedagogical and assessment techniques employed

Course execution aspects:

- Carefully consider student team size and composition: Small team size implies emphasis on technical problem-solving, large team size means that project management and teamwork will be focused
- Use generalized project models and tools and methods for very early and late project phases. Connect these to domain-specific tools and methods used in intermediate phases
- Prepare students to cope with the uncertainty and unpredictability of a development project
- Be prepared to manage conflicts within the student teams
- Introduce methods and tools at timely points in the project
- Decide checkpoints/deliveries to be able to track progress in the project work
- Carefully consider the communication flow in the course
- Teachers need high availability at delivery points in order to give fast feedback and decisions on project continuation
- Be prepared to improvise in terms of e.g. problem solving workshops or extra lectures
- Include assessment tasks as early as possible
- Use frequent individual time reporting to facilitate the early detection of problems in the project
- Include self-evaluation of project success and working practices
- Request feedback on time actually spent
- Include adequate training in use of equipment

5. Conclusions

In this paper, a definition of the design-build-test educational experience concept has been proposed which clarifies the terminology used to describe this important engineering education element.

Data describing a large number of design-build-test experiences has been compiled, enabling comparisons and constituting an idea catalogue. The data indicates that these experiences do indeed motivate students, integrate different engineering disciplines, train system development and non-technical skills such as teamwork and communication, and thus play a key part in engineering education. These educational experiences further receive very positive evaluations from students, faculty and industry stakeholders. However, design-build-test tasks need to be very carefully planned, both as they stand as separate learning events, but also as parts of a planned sequence of design-build-test experiences in a curriculum. Moreover, design-build-test experiences require different faculty competence and learning environments. The paper suggests a set of guidelines that help address these challenges in a course development process. More work is needed in the area. Assessment in design-build-test-based courses is a challenge as well as the development of cost-effective design-build-test experiences.

Acknowledgements

This research was financially supported by the Knut & Alice Wallenberg foundation. The support is gratefully acknowledged

References

- Andersson, J., Palmberg, J.-O. "The Evolve Project – A Mechatronic Project for Final Year Students". *Proceedings of ICED 03, Stockholm, Sweden, 2003.*
- Berggren, K.F., Brodeur, D B., Crawley, E. F., Ingemarsson, I., Litant, W. T. J., Malmqvist, J., Östlund, S. "CDIO: An International Initiative for Reforming Engineering Education". *World Transactions on Engineering and Technology Education , Vol. 2 No. 1, 2003.*
- Brodeur, D. B., Youn, P. W., Blair, K. "Problem-Based Learning in Aerospace Engineering Education". *Proceedings of the ASEE Conference, Montreal, Canada, 2002.*
- CDIO Initiative Homepage, www.cdio.org, 2004, accessed on March 29, 2004.
- Crawley, E. F. "The MIT CDIO Syllabus Report". *Technical Report. www.cdio.org, 2002.*
- Elger, D. F., Beyerlein, S. W., Budwig, R. S., "Using Design, Build, and Test Projects to Teach Engineering". *Proceedings of the 30th ASEE/IEEE Frontiers in Engineering Conference, pp FC9-FC13, Kansas City. MO, USA, 2000.*
- Gustafsson, G., Newman, D., Stafström, S., Wallin, H. P. "First-year introductory courses as a means to develop conceive – design – implement – operate skills in engineering education programmes". *Proceedings of 30th SEFI Annual Conference, Firenze, Italy, 2002.*
- Hallström, S. et al. "CDIO Workspace Starter Kit". *Technical Report, www.cdio.org, 2004.*
- Jansen, A. "Engineering Students and Robot Contests: Learn Life to the Max!". *Proceedings of ICED 03, Stockholm, Sweden, 2003.*
- Pahl, G., Beitz, W., "Engineering Design – A Systematic Approach", 2nd English edition, Springer-Verlag, Berlin, 1996.
- Sommerville, I. "Software Engineering", 6th edition, Addison-Wesley, New York, 2001.
- Sullivan, J. P, Watkins, W. A. "A Design/Build/Test Environment for Aerospace Education". *Proceedings of AIAA 2000, Paper No AIAA 2000 – 0525, 2000.*
- Ulrich, K. T., Eppinger, S. D. "Product Design and Development", 2nd edition, McGraw-Hill, New York, 2000.

Professor Johan Malmqvist

Chalmers University of Technology, Department of Mechanical Engineering

Division of Product and Production Development

SE-412 96, Göteborg, SWEDEN

Telephone: +46 31 772 1382, Telefax: + 46 772 1375, E-mail: johan.malmqvist@me.chalmers.se