Occurrence of flow in a first-year engineering design module

T. Burke
Lecturer
School of Electrical and Electronic Engineering
Dublin Institute of Technology
Dublin, Ireland
E-mail: ted.burke@dit.ie

G. Duffy
Lecturer
School of Electrical and Electronic Engineering
Dublin Institute of Technology
Dublin, Ireland
E-mail: gavin.duffy@dit.ie

Conference key areas: Engineering education research; Learning formats specific for engineers; Novel education tools for engineering programs.

Keywords: flow, student-centred learning.

INTRODUCTION

The term *flow* describes a psychological state of intense and sustained concentration on a task, which is highly rewarding and productive. In this paper, we report on our efforts to create conditions conducive to flow in a group design project in the first year of an undergraduate electrical engineering programme in the Dublin Institute of Technology in Ireland. The *Engineering Practice and Design* module, known informally as "RoboSumo", focuses on the design and construction of an autonomous robot vehicle.

The combination of a highly motivational task (building a robot for a competition) together with occasional critical deadlines (competition days) and the ability to carry out practical work at home or in other settings outside special-purpose labs seems to provide students with opportunities for experiencing flow. However, students' descriptions of their own experiences point to factors which still inhibit flow.

Here, we review the relevance of flow to project work in engineering education, describe our efforts to nurture it within the RoboSumo module, and report on interviews carried out with students to explore how the design of the module affected their experiences and work patterns. Additional evidence is drawn from students' project blogs and from the experiences of staff teaching on the module.

1 FLOW

1.1 Introduction to flow

Csikszentmihalyi coined the term *flow* to describe the positive psychological state that occurs when a person is completely absorbed in a task [1]. He described it as the “optimal experience” - highly productive and deeply rewarding. In his extensive research on the phenomenon, Csikszentmihalyi (and others) showed that flow occurs in a broad range of activities, but that nine conditions (listed below) are common to people’s flow experiences. These will resonate with many creative individuals (including engineers) who recognise them in their own practice, especially during the periods of deep concentration when outstanding work is typically produced.

1.2 Flow conditions

Csikszentmihalyi identified nine characteristics of tasks that elicit flow [2]:

1. The goals at each step of the task are clear.
2. The activity is such that a person’s actions result in immediate feedback.
3. The complexity of the task is commensurate with skill level.
4. Action and awareness merge; undivided attention is focused on the task.
5. “Dimensions” are excluded from consciousness. The mind naturally focuses on what is relevant at each moment, disregarding extraneous factors.
6. Fear of failure is absent. What needs to be done is clear and skill is adequate.
7. Self-consciousness disappears as a result of deep immersion in the task.
8. Time becomes distorted. Hours may fly by while immersed in the task.
9. The activity becomes *autotelic* – done for its own sake. A task which satisfies the above conditions is *enjoyable*, providing the motivation to keep doing it.

Item 3 is particularly significant here. When a task is too complex for a person’s skill level, anxiety can result. Conversely, if a challenge falls below a person’s skill level, boredom results. Csikszentmihalyi provides a useful visual representation of the so-called *flow channel* (illustrated in Fig.1), where skill and challenge are matched.

1.3 Flow in engineering education

Csikszentmihalyi’s work has been influential in a broad range of disciplines, but it has particular resonance in creative work [2]. The conditions for flow arguably provide a check list for those designing and delivering project-based learning experiences [3]. Despite this, explicit references to flow in the engineering education literature are surprisingly sparse (notable exceptions include [4, 5]).

In the early years of an engineering degree, project work is seen as addressing a range of important learning needs, including group work skills, project management and critical thinking [6]. Moreover, it has an important *motivational* role, showing students the relevance of theory they encounter elsewhere and introducing them to the substantial personal reward of overcoming a challenge through sustained hard work and the joy of simply creating things. Flow is therefore particularly pertinent to the design of project-based learning experiences in engineering education.

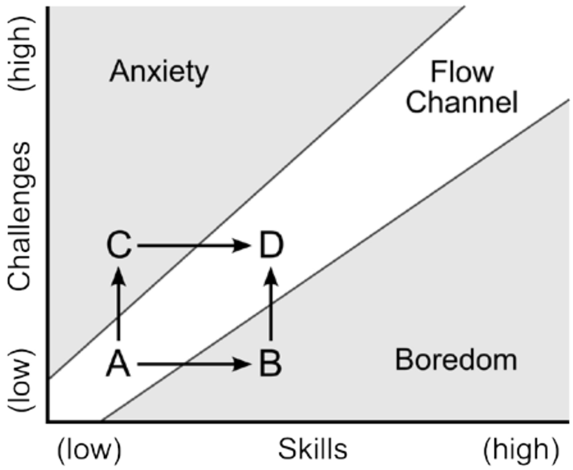

Fig. 1. This diagram is closely modelled on that in [1, p.74], which illustrates the concept of the *flow channel*, in which challenges match skills. Trajectories A→B→D and A→C→D are routes by which a person can grow in ability while experiencing flow. Consider the example of a novice programmer learning to program a robot. At point A, a simple task such as switching the motors on and off is challenging and rewarding if completed. If more complex tasks are not attempted as skills increase, boredom may occur (point B), but the person can return to the flow channel by attempting more complex tasks (point D). Alternatively, if the beginner is confronted by a challenge which slightly exceeds their current skill level (e.g. a tight deadline by which the robot must be programmed to follow a line), some anxiety may occur (point C), but it can be relieved by working to improve skills. Once the requisite skills are mastered, the person completes the task and returns to the flow channel (point D). In each case, the path to the flow channel is clear, but if a person strays too far from the flow channel, motivation will be undermined.

2 ROBOSUMO MODULE DESIGN

2.1 Module structure

Students work in teams of three over a semester (13 weeks) building an autonomous robot. A competition involving a simple task, “the race to the wall”, takes place midway through the module. The module concludes with a second competition, a robot sumo tournament with cash prizes. Each team receives a kit of low-cost components, including a breadboard, motors, sensors, a microcontroller (with a low-cost USB programmer), resistors, capacitors, diodes and LEDs. Teams purchase (or scavenge) additional parts to complete their design. The kit is designed to facilitate robot building with *real* engineering components, but without specialist equipment. Practical work can therefore be carried out at home or elsewhere, allowing sustained focused work, free from the constraints of college opening hours.

The stated learning outcomes of the module focus on group work and project management. However, the design task is intended to build competence in electronics, programming and mechanical design. Both process and product are therefore assessed. There are four equally weighted assessment components: individual contribution to group process; individual contribution to technical attainment; individual documentation; and robot performance in competition. In earlier years, each team documented their work on a shared wiki. However, to facilitate fairer individual assessment, every student is now required to write an individual blog documenting his or her work on the project.

2.2 Specific measures to facilitate flow

The design of the RoboSumo module is continuously refined to create conditions in which students are more likely to experience flow. Components, equipment and software have been chosen to facilitate independent work outside college labs. Admittedly, pre-selecting some of the parts makes the design task more prescriptive. Nevertheless, we now provide a semi-complete (albeit basic) set of components on day one to propel students into more productive learning from the outset.

3 STUDENT EXPERIENCES

3.1 Methodology

During the second half of the module (between the "race to the wall" and sumo competitions), semi-structured 1-hour interviews were carried out with six student volunteers, each from a different team. A phenomenological approach was used to gain an insight into the experience of participating in the module. The questions spanned a range of topics, but those directly related to flow were:

- Can you describe how you work on this module outside of class time? Whereabouts, alone or with others, in a quiet or noisy environment?
- How long do you work for? How does time pass during these sessions? Do you enjoy the session? Do you become absorbed? Do you feel you make progress, achieve success, experience frustration, feel annoyed, feel irritated, feel happy, find it rewarding, feel a sense achievement?
- What types of things do you do during these sessions? Build circuits, design, test, program?
- What do you do after the session? How does it influence your experience in the next class?
- Do you tell anyone you worked? What do you say to them?

Information gleaned from these interviews was combined with evidence obtained from tutor observation throughout the module and from analysis of student blogs.

3.2 Student experiences

Student 1: This student's team assigned a technical role to each individual member (programming, wiring and mechanical construction). This student was responsible for programming, which he spent time doing alone, but he mentions being unable to test his code without other team members (and presumably the robot) present. He articulates a clear preference for working together with team mates, "*I'm not motivated on my own. I can take my time. Around other people I work more quickly and respond to their demands, and work quicker. I spend more time with the group than on my own. If I run into problems on my own I've no help, it's quiet and I get bored. With the group, the company keeps you interested.*" However, he indicates that working alone becomes easier as confidence grows, "*I can program on my own and show [my work] to the group at next meeting... It gets easier the longer you spend at it.*" He identifies real-time feedback on work as useful: "*The most valuable time for learning is testing during the lab, you see what works and [what] doesn't... You find out whether you're right or wrong about your ideas in the class. You find out by testing, by changing wiring, location and program and testing.*" He describes the team working together with clear goals outside class, but always in a college lab: "*We stayed in for a good few hours the week or two before the race... It's the same work we do in the lab... We knew what we had to do.*" His account of the final preparations is ambiguous: "*We didn't get all the problems fixed but we were there for a number of* *hours, we were frustrated and tired at the end... Time didn't drag but didn't fly in either, we were busy and time does go quickly but it is a long time in there.*"

Student 2: This student's team adopted individual technical roles, but exchanged them as the need arose. The team were not under too much pressure: "*I know some teams stay up all night wiring but we haven't had many situations where we were stressed. I have felt there is a lot of time.*" He says the team did little practical work at home. However, there is evidence that the team's efforts intensified in the final days before the competition: "*Yes, you sacrifice, you put less effort into a lab report if the race is the next day... there was a day we skipped two lectures to prepare, to make sure it would do what we wanted.*" He acknowledges the frustration of working alone when skills are not yet adequate to meet a challenge, "*I suppose it's frustrating if you don't know what the problem is. If you're without a lecturer and you don't know what the problem is, it is difficult to solve... We didn't have sessions alone at the start but were more comfortable alone at the end of semester.*" However, he identifies the benefit of individual work outside class as, "*Less stress, you feel you've done lots.*"

Student 3: This student's team designated individual roles (wiring, programming, mechanical construction) after a few weeks, which he says increased productivity. He sees individual roles as providing a learning opportunity. Each person's role should be one "*where we can improve the most ourselves, so as well as doing a good job, we're improving on something we're not strong on. For example, I do programming, where I'm not the best, and get the opportunity to improve on that.*" However, he emphasises the importance of assigning roles where people are not out of their depth: "...*we made sure they were comfortable, if they weren't comfortable we gave them a role they were more comfortable, like to do more, easier.*" He recounts frustrating aspects of the team's final preparation for the competition, "*We had spent loads of time, 15 hours in two days after 7 o'clock; we were tired and sick of same problems, so decided to cut out all sensors.*" As the competition approached, frustration intensified. Ultimately, in desperation, the design was dramatically simplified: "*The further the day went, the more frustrated we got and that [feeling] didn't leave us for the last day... we were very frustrated. We didn't have a clear head coming in [on the] last day; every problem led to another problem [being] discovered. At the end of it we decided to leave everything out and just do batteries and motors.*" Despite these difficulties, he suggests that valuable lessons were learned: "*They were the two hardest days. We wanted to work hard. If we had started working like that earlier, we would have performed better. Benefits were that we learned that on [the] last day we worked better because we were physically close to each other while doing our separate jobs, so we could talk, reassure, not be isolated... We learned how to be a better team on last day.*"

He spent time doing detailed manufacturing work alone at home, "*I spent long enough that day. The body was such that if it wasn't done precisely, motors and batteries would not fit. [They] had to slot in, [a] separate axle for each motor had to fit in a slot... [I] took my time working on it at home.*" Due to other commitments, he finds it hard to work at the weekend, but identifies clear benefits to working at home: "*I find it good doing work at home. [You] can ring team mates, but you're alone and sometimes it's easier to work alone as you're not distracted, you have your own computer, you're more comfortable. [It's] easier in one sense, more relaxing... without people putting you under pressure.*"

Student 4: This student had a difficult experience due to lack of engagement by the other members of the team. He sees group projects as worthwhile, but questions whether students possess the required group work skills in first year. Citing the example of another university's undergraduate programme, he suggests that the technical complexity of the robot design task is more appropriate to third year. He proposes that the module could be replaced by "*a group-based module that is easy, with evidence that each person is doing their part, without punishing the group as a whole. A lot of knowledge is required for this module.*" Following a series of frustrating technical problems, he describes one long night working alone to complete a robot for the competition: "*I head home on Thursday to the desk in my room. I was going to do this until it was done. Didn't look at the clock, no point as time didn't matter as I wanted to get it done.*" Although it did yield rewards, he would not choose to repeat this period of intense work: "*Once I was done there was a feeling of satisfaction but it was pretty much grin and bear it, it was something I had to do.*"

Student 5: This student took a leading role in assigning technical duties (programming, wiring, mechanical construction) to individual team members. He mentions later that there was some crossover of roles. Programming seems to have been a source of frustration within the team, both due to differing perspectives on who was responsible for what, and because some problems were difficult to solve. He describes the intense preparations for the race as "*Frustrating.*" He mentions that programming is frustrating when you can't get the program to compile – a situation in which many novice programmers struggle to make use of available feedback (e.g. error messages). Once a program compiles and runs, many novices find it is easier to learn from what works and what doesn't. He also regards it as preferable for more than one team member to contribute to this aspect of the design: "*With programming you get one thing sorted and then the other thing breaks, there's not a lot of winning in it unless it compiles, mainly frustrating but, once it's done, enjoyable. [In] a team of three [you] can take a break, let [someone else] take a look.*"

Student 6: This student states that his team did "*90%*" of the work in class, but extra sessions were required as deadlines loomed. He says that one team member constructed a complete working robot over the two-week Easter holiday. He also describes sessions in which he and a different team member worked together on the programming: "*They go on until both of us are happy with it, back and forward with ideas, we both agree on something, put it down and see if it works; there's no arguments. We work in one of the labs here.*" Although frustrating problems were encountered, these sessions seem to have been productive and ultimately satisfying: "*...for the race to wall it was 3 to 4 hours. We were banging our heads against the wall. That's annoying about the module - you have problems but there's a sense of relief when you get them fixed, [it] makes you happy. It's good to watch the robot work because I feel 'I actually made this'. You came together as a group and we built this and it doesn't have a controller and it moves and does everything itself, because of everything that you put into it.*"

Other insights into student experiences:

The use of individual student blogs has delivered a number of benefits. Once a tutor subscribes to a student's blog, immediate email notification can be sent each time a new post is published, providing a useful timestamp. The blog system also facilitates very convenient inclusion of photo and video content, a facility which many students use extensively to document their work. A frequent scenario involves a student publishing a blog post very late at night, which evidently (based on its content) marks the end of a marathon work session. Videos included in many of these blog posts reveal elaborate practical work being carried out in students' homes – in kitchens, bedrooms, workshops and sheds – providing evidence that the module design does facilitate practical work outside the college, creating greater opportunities for flow.

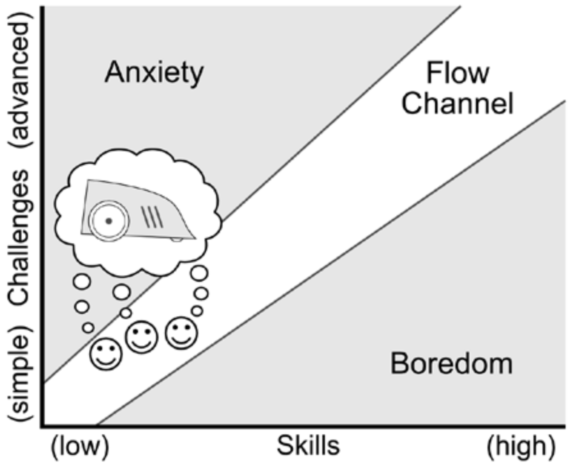

Fig. 2. Initially, all team members are programming novices. Sharing one set of equipment, the team works together to learn the basic steps to program the robot. At this stage, the team performs prescriptive tasks such as flashing an LED or turning motors on and off, which build familiarity and confidence.

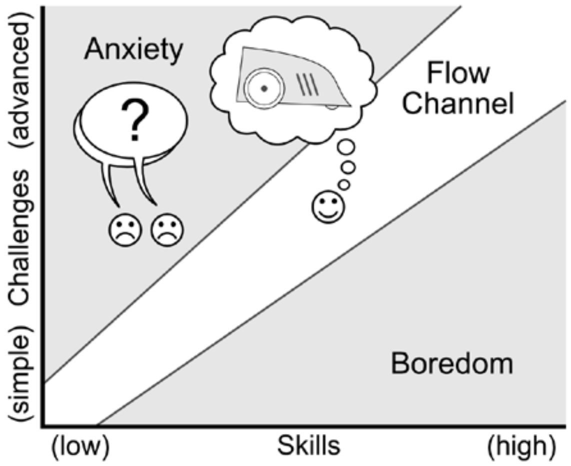

Fig. 3. Often, a team member emerges as the lead programmer and takes sole possession of the programming equipment. With uninterrupted access to the equipment, skill level rapidly increases. Meanwhile, without access to the equipment, other team members are powerless to contribute or to catch up.

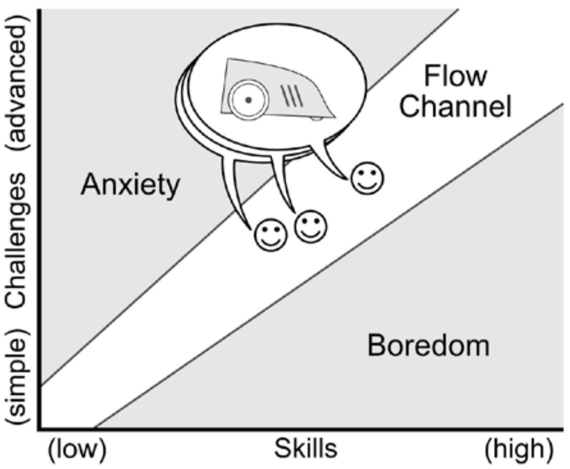

Fig. 4. Although one team member learns faster than the others, effective communication and peer instruction keep all members engaged in the front line of development. The peer instruction process deepens the lead programmer's understanding. Together, the team can critically analyse proposed solutions.

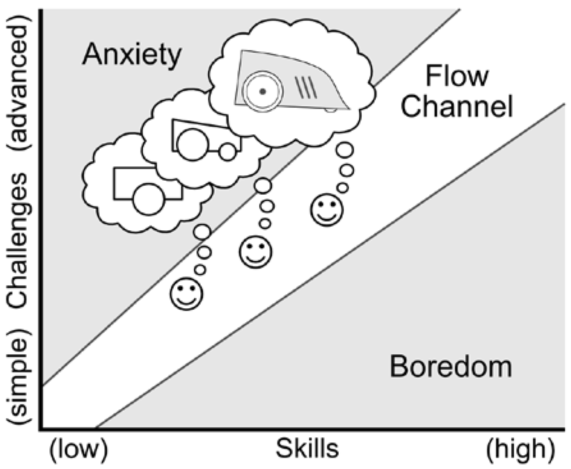

Fig. 5. With a very low-cost development platform, all team members can have their own equipment, which facilitates individual learning. A lead programmer may drive the front line of development, but other team members can apply programming to simpler problems, such as testing subsystems.

These blog posts and videos provide a fascinating window into the work of teams and individuals outside the classroom, and reveal that some students exhibit remarkable *sprezzatura*. For example, a team may present a working robot at a competition without mentioning that any exceptional efforts were required, but a blog post may later provide clues that one team member made such an effort.

4 CONCLUSIONS

To sustain flow, skills and challenges must be in balance. In the RoboSumo module, some students face challenges which exceed their skills, undermining motivation. When a student falls far behind a team mate in a technical area, it is tempting to give up trying. Consider the example of programming the microcontroller. To date, teams have been given a single USB programmer. Initially, few students have experience of microcontroller development, so prescriptive tasks are assigned to build confidence before teams move on to more creative challenges. At this point most students are appropriately challenged and enjoy the team's minor triumphs, as illustrated in Fig. 2. If one team member emerges as the dominant programmer, the team's single USB programmer often remains in his or her possession, which reinforces disparity. As that person's skills increase, other team members become marginalised, as illustrated in Fig. 3. Some students say they do not feel entitled to take the USB programmer between classes because their programming skills are weak. Unfortunately, this makes it almost impossible for them to catch up and become a useful contributor. In these circumstances, the understandable temptation is to abdicate responsibility for that element of the project and focus on other areas.

Some teams avoid this problem through effective communication and peer instruction, ensuring that all team members learn (albeit to different degrees) and remain involved in the *front line* of development (as illustrated in Fig. 4). Peer instruction is rewarded in the module's assessment scheme (based on tutor observation and blog evidence), but not all teams practice it. Next year, every student will receive his or her own programming equipment, so that weaker programmers will be free to practice. If unable to contribute to the front line of development, they can still undertake useful programming tasks of appropriate difficulty (as illustrated in Fig. 5), such as writing simple microcontroller programs to test subsystems.

REFERENCES

[1] Csikszentmihalyi, M, (2008). Flow: the psychology of optimal experience, Harper Perennial, Modern Classics edition.

[2] Csikszentmihalyi, M., (1996). Creativity: Flow and the Psychology of Discovery and Invention, Harper Collins.

[3] Shernoff, D. J., Csikszentmihalyi, M. (2009). Cultivating engaged learners and optimal learning environments. *Handbook of positive psychology in schools*, pp. 131-145.

[4] Michaud, F., Lucas, M., Lachiver, G., Clavet, A., Dirand, J. -M., Boutin, N., Mabilleau, P., Descôteaux, J. (1999). Using ROBUS in Electrical and Computer Engineering Education. In *Proc. American Society for Engineering Education*, Charlotte, June 1999.

[5] Michaud, F., Clavet, A., Lachiver, G., Lucas, M. (2000). Designing toy robots to help autistic children - an open design project for electrical and computer engineering education. In *Proc. American Society for Engineering Education*, June 2000.

[6] Mills, J. E., Treagust, D. F. (2003). Engineering education - Is problem-based or project-based learning the answer? *Australasian Journal of Engineering Education*, *3*, 2-16.