Fifth International Conference
Modelling and Development of Intelligent Systems
June 23 - 25, 2017
"Lucian Blaga" University
Sibiu - Romania

Why and how Practice impacts Confidence in introductory object oriented programming Courses

Viggo Holmstedt, Shegaw A. Mengiste

Abstract

Why and how do practice of object oriented programming concepts impact self-confidence to instructors teaching introductory object orientation? Does it also impact confidence from students? In this paper, we seek to answer what motivates instructors to build and maintain a codebase over a long time. An aspect that stimulates curiosity is the characteristic of bimodal distribution of marks, a robust characteristic still in search for a conclusive explanation. We do not investigate that characteristic, but refer a few explanations through the study of practicing instructors and their motives. In this investigation, we adopted the notions of social worlds and boundary objects as analytical lenses. We identified two important social worlds: the social world of instructors of introductory object oriented programming and the social world of students introduced to object oriented programming.

Keywords*: Codebase· object oriented programming·self-confidence·credibility·social worlds·boundary objects*

1 Introduction

For many years educational researchers have been attempting to develop models of instruction that specify causal connection between class room teaching behaviours and student achievement [1]. As Ryoo et al [2] pointed out, what challenges educators the most is often not the lack of teaching materials, but the constraints imposed upon a course. These include (1) type of audience, (2) the limited amount of time available for a prearranged set of course objectives, (3) a rigid curriculum, and (4) the choice of teaching method. More specifically, in the context of teaching computer science courses, existing research shows that there have been a number of problems faced by instructors teaching object-oriented programming courses to undergraduate students. Understanding object-oriented concepts like classes, constructor invocation, overloaded constructors, friend functions are difficult for students [3].

A more sensitive aspect of teaching object-oriented programming is the instructors experienced level of abstraction and pace of change in new technologies. Sometimes instructors may not have sufficient experience in the field to teach the subtle and often abstract issues convincingly. It can also be challenging to absorb, adapt and then accommodate new tools adopted in teaching and instruction. This aspect may have sensitive implications, where few instructors would admit that the problem pertains to themselves. If admitted, though, it might influence self-esteem and infer a possible decrease of student confidence and credibility.

The teaching of object oriented introductory programming have intriguing characteristics, like high dropout rates, as reported by Nilsen & Larsen [4]. Another is the mark distribution. Contrary to many other curricula, introductory classes in object oriented programming exhibit bimodal mark distribution [4]. Bimodality here refers to many high and low marks, with few in the middle. The finding of bimodal distribution led to their [4] interest for, and an implementation of, the Keller [5] principle and the associated PSI (Personalized System Instruction). In the original paper, Keller referred to the phenomenon as "upside-down" [5]. Robins [6] reports that the bimodality has been robust through many decades and populations. He further introduced the notion of learning edge momentum (LEM) to explain the factors that contribute to the bimodal distribution of marks. He argues that LEM arises from the interaction of two factors: primarily, successfully acquiring one concept makes learning other closely related concepts easier, but failing to understand one core concept makes further learning harder [ibid 6pp. 40]. Compliance to governmental wishes for a normal distribution of marks (Gauss, single modal, form) is likely to wear down a teacher's self-confidence in terms of setting marks that seldom meet the expectations, not even over long time.

To improve the learning results in introductory object oriented programming courses, some instructors are motivated to develop and maintain a system (a codebase) as a tool to support their teaching. Developing and maintaining this tool type and similar systems presupposes enhanced knowledge and mastery. In this paper, our focus is in exploring how the knowledge and mastery of object oriented programming concepts and the prolonged use of code-based tools affect the self-confidence and credibility of instructors teaching introductory object oriented programming courses. We argue that the characteristics of our research problem may be common to almost all subjects taught on higher levels of education. However, we also think that the bimodal distribution is special to introductory object-oriented programming, because of the subject's characteristics of needing activities with demanding precision and a high abstraction level. We think that few other curricula share this explicit characteristic.

As such, we aim to understand how specific instructor activities, behaviour and actions affect the learning process. More specifically, we want to investigate how the instructors' long engagement and experience in practicing what he/she teaches contribute to build self-esteem and to increase student acceptance and credibility; and to see if this can be augmented by the instructor's long-term activities of local codebase maintenance. We do not measure the learning results, nor the mark distribution, but claim by abduction that the self-esteem and student confidence and credibility will benefit the teaching activity. We investigated the problem based on data gathered from 10 instructors employed at several higher learning institutions in Norway, who had many years of experience in teaching object-oriented programming. Some of our respondents adopted codebase to support their teaching activities. Besides, we have also used the many years experiences of one of the authors of this paper who have been engaged in creating and using a codebase called Evalanche [1][7].

As analytical framework, we used concepts of social worlds and boundary objects from the literature of symbolic interactionism [8-12]. The social worlds perspective is a well-established and rich sociological tradition to investigate and analyze teaching and learning as a socio-technical process [13, 14] that involves participation of different actors.

The remaining structure of the paper is as follows: Section 2 provides an overview on our conceptual framework. Section 3 introduces the research approach and methods employed. Section 4 presents the empirical findings. The last section, presents the analysis, discussion, and implications of the study.

[1] Evalanche: an LMS supporting teaching and enhancing learning as well as improving self-confidence and credibility among students.

2 Conceptual Framework: Social Worlds & Boundary Objects

The concept of 'social worlds' has its roots in the American sociological tradition of symbolic interactionism [11, 15, 16] which primarily focuses on meaning construction "amongst groups of actors – collectives of various sorts – and on collective action — people doing things together and working with shared objects" [9]. The emphasis on the creation of meanings through interaction has strong parallels with the social construction of technology (SCOT) [17]. The key argument in both symbolic interactionism and social construction of technology is that the development of technological artifacts is an outcome of complex social and political negotiations [13]. Another important concept introduced by the social world's perspective is the notion of arena. An arena is a field of action and interaction among a potentially wide variety of collective actors (be it human, organizational or technical artifacts). In arena, the different social worlds that focus on a given issue and prepared to act in some way are brought together [11, 18]. Gieryn [19] also identified three important properties that are common to all social worlds: the potential for division and segmentation into sub-worlds; intersection with other social worlds; and legitimization through the definition and enforcement of standards and boundaries of a social world.

The concept of boundary objects appears to investigate and analyze boundary-spanning phenomena in organizations. Boundary spanning describes activities that occur at the *boundaries* between different groups with different tasks and activities and with a need to cooperate for a common goal [20]. When there are two or more groups with boundary spanning activities (tasks), establishing meaningful communication and collaboration between those groups inevitably poses significant challenges. There are usually challenges associated with transferring [21, 22] and translating information from one world to another; there are problems of information integrity and translation.

Star and Griesemer [23] argued that certain boundary objects are flexible enough to accommodate different interpretations or interests emanating from various distinct social worlds, yet robust and plastic enough to maintain a common identity across all social contexts, then allowing translation to take place across the boundary. Gal, Yoo, and Boland [24, 25] describe boundary objects as:

> *"… abstract or physical artefacts which reside in the interfaces between organizations or social communities and have the capacity to bridge perceptual and practical differences among diverse communities in order to reach common understandings and effective cooperation" (Gal et al., 2004, pp. 194).*

The notion of boundary objects have been widely used to explore a broad range of settings where boundary-spanning activities are critical. For example, Henderson [26] used engineering drawings and sketches as boundary objects to enable communication across different occupational communities. Pawlowski & Robey [27] used the notion of boundary objects to analyze the knowledge brokering work of IT professionals. Tudhope et al [28] also illustrate prototypes as important boundary objects in software development activities, since they facilitate negotiation and communication practices between users and developers. Various artefacts and representations in software development practices mediate the communication between different groups and the process of synthesizing their diverse interests; e.g. between designers, between designers and managers, between designers and end-users [20]. In those contexts, boundary objects need to provide the basis or common ground, which different groups share and use to shift their "knowledge boundaries" in order to solve their joint heterogeneous problems (ibid, 2006).

In the context of the study, we adopted the notions of social worlds and boundary objects as our analytical lenses. We identified two important social worlds: the social world of instructors and the social world of students. The two social worlds intersect where they are both concerned about the learned outcome. The social world of instructors introduces and executes the novelty aspect [29], while the student world is challenged by the same novelty. The challenge of the novelty lies in the fact that programming concepts become more abstract along the learning path, and that the rules for best

practices become harder to accept, and even believe, for novice programmers. The concerns are divergent in the sense that the instructors world want to teach methodically and systematically, while the student world want to execute learning steps fast, as in finishing a list of exercises. In this divergence of concerns, there is a great need for confidence and credibility to impose and accept each other's aims. A boundary object is therefore one that ensures self-confidence in the instructor, as well as confidence and credibility in the student towards the instructor.

3 Research Approach and Methods

3.1 Research approach

This study employed an engaged scholarship approach [30], which seeks to combine both qualitative and quantitative methods. We believe that this approach offers us both the methodological and analytical leverage in our investigation.

Van de Ven defines engaged scholarship as "a participative form of research for obtaining the different perspectives of key stakeholders (researchers, users, clients, sponsors, and practitioners) in studying complex problems" [ibid30]. This approach, according to Van de Ven [30] aims to move beyond the simplistic notion that research knowledge is generated in the lab and then packaged and diffused into practice. Rather, Van de Ven adopts an interactional view in which professional and research practices contribute to each other's growth through different types of activity. By embracing qualitative as well as quantitative methods and promoting process studies as well as variance studies, the notion of engaged scholarship offers opportunities to transcend the traditional dichotomies of core versus diversity, rigor versus relevance, and positivist versus interpretive. Moreover, Van de Ven [30] identified four different forms of engaged scholarship: *informed basic research, collaborative basic research, design and evaluation research, and action research (ibid).* In this research, we categorized our research as an informed basic research that aims to describe specific aspects of the social phenomena teaching and learning.

3.2 Data Collection Methods

The study has relied on two prominent sources for empirical material. Primarily we interviewed 10 instructors from higher learning institutions in Norway who have been teaching introductory object-oriented courses. The data collection was in Norwegian, and later translated to English. Our secondary source is the experiences of the first author of this paper. From 1999 to 2017 he developed the codebase Evalanche to support his teaching. His experiences contributed a lot in giving interesting insights and findings for the scope and purpose of this paper.

Informal Meetings with Instructors

Informal discussion with colleagues who have been engaged in teaching introductory object-oriented courses triggered this study. As most of the colleagues had many years of teaching as well as relevant practice in teaching object-oriented programming courses, it became interesting to approach some colleagues to discuss their experiences. At first, the discussions took place in informal meetings from 2009 to 2012. The discourse revealed that some of those colleagues practiced object oriented programming, as well as built software to either administrate their teaching, or to support some other aspect of the student's needs. Most of those colleagues claimed that the overall teaching and learning outcomes would likely benefit from it. Then the idea matured to design a research agenda targeting the teaching of introductory object-oriented programming in combination with maintaining code relevant to the teaching and learning activities.

For further exploration of the topic and the problems involved, we conducted telephone interviews with 12 instructors. During an informal introductory interview, we discussed a respondent's relevance of practice to find out whether it would suit the investigation. We then contacted the 12 instructors through email to let each decide on a further participation in the investigation. This approach left us with 10 relevant respondents.

Questionnaire

We prepared a questionnaire with a guide and deployed to the 12 candidate instructors. Before answering the questionnaire, the respondents considered three premises:

- Did the coding activities have a duration equivalence of more than 12 months practice?
- Were the coding activities relevant to teaching undergraduate object oriented programming, administrative activities included?
- Has the coding activities influenced a perception of vocational self-confidence, ability to teach and general role relevant to teaching, including credibility and confidence from students?

Two candidates exited the investigation, while 10 respondents remained. The 10 remaining candidates qualified to answer the questionnaire.

Longitudinal experience

As our second source of data, we also relied on the data collected from the many years of experiences of the first author of this paper in both teaching undergraduate object-oriented programming courses and in developing the codebase Evalanche [7] to support the teaching activities. The values of the experience pertain to observations in several domains, like technology, acceptance from the relevant social worlds, compliance to social rules, user demands on reliable services.

3.3 Data Analysis Technique

Established methods for handling qualitative data inspired our analysis of the empirical data. NVivo, using coding to nodes as a main approach, supported the responses analysis. The coding helped us localize relevant statements, sometimes related across the actual questions. We read carefully through the interviews from the questionnaire to get an insight of the total material. To inspect concrete themes related to our research question, we coded fragments from the responses, and let NVivo help us find relationships and contexts across respondents and questions. The analytical framework, which embraced the notions of social worlds and boundary objects, has been used to inform the NVivo analysis and thereby served as a sensitizing device for what 'to look for' in the data. Based on those analysis techniques, we were able to find relevant statements to present in the findings section.

4 Findings

In this section, we present our findings from the interviews and the personal experiences of one of the authors. Our empirical evidence has two prominent sources. The first source is the questionnaire, with evidence from instructors presented in section 4.1.

The second source is from the first author's experiences of gaining practice, insight and knowledge related to development, maintenance, using and administering of a learning and teaching management system through 18 years, presented in section 4.2.

4.1 First source: Responses from Instructors

System types

From the coding of answers to nodes, we could find out more about what kind of systems the respondents had worked with.

a. A course in study technique as part of the Keller based experiment [4]

b. An ambitious tool for collecting several learning resources [31]. The tool collects, organizes and then presents diverse instructional material from sources of many different formats

c. Administrative tools for salary negotiations and tools for individual work plans

d. A variety of tools for research or professional activities

e. An ambitious tool for the administration of learning and teaching [7]

With these types of systems in mind, we will study relevant evidence from the responses repository.

Effects of creating and maintenance of codebase

Most respondents reported they have gained from the coding activities in terms of both new skills and confidence. A few respondents link their general level of profession to their high level of skills and competence. More typically, one respondent states: *"Doing big development projects is by its nature positive, especially since the development is within technologies I am teaching daily."* Statements like "*Developing big systems means much coding and work that we tell the students about*" sustain the development of personal skills and coding competence.

Most respondents have gained relevant technology experience from the very coding activities. Statements like "*One will meet problems [...] to overcome. By this both skills and competence grows.*" posed in the learning arena, generate a feeling of credibility from students. One statement is "*This [the credibility] is connected to my own professional self-esteem. But also the fact that one can refer to having developed big systems lets one get a natural authority.*" In general, the long work with systems feels important: "*I have experienced that my students have respect for my long practice*". Several respondents report a positive effect of better and more interesting teaching methods due to their practice of maintaining code to complement their teaching.

Most respondents clearly relate augmented confidence to their coding and maintenance activities, as in the statement: *"maintaining the code is important as it gives me hands-on experience with new methods, new tools and new document formats."* This is also important for credibility, further supported by the next evidence: *" I can confidently say [to the students] that I have experience with ... And I can elaborate more".*

Attitudes of respondents towards their role

We learned about the respondents' perception of their role and social mission linked to maintaining code. An important statement is this: "*I mean that the project could be useful to quite many, and that it could contribute to considerable cooperation and sharing of experience.*" There is also an interesting dimension of duty. "*I should be able to develop tools that I need.*" Statements like "*I think it is my duty to be updated within my field*" demonstrate a sense of commitment to the role apart from personal concerns.

Some also indicated their concerns for still more: "*I think it is important, not least the overarching aspects*". Through a personal follow-up of this statement, we learned that the expression "overarching aspects" referred to concerns for the quality of educated system developers. We also found benefits to the educational economics: "*I believe the system has caused an increase in the production of credits*". One of the respondents gave a quite challenging piece of evidence: "*I think that if possible, all programming instructors of higher level should practice* [programming skills] *in the industry from time to time, or at least take an active part in programming projects, as for instance Open Source projects*."

Motivation for developing and maintaining codebase

All respondents had personal interest in mastering several technologies. Still it is interesting to learn that no respondent let this be the sole motive. "*No [it was not my own needs], but it was a good bonus.)*". For the maintenance of administrative tools, we find the interesting statement: "*... I have selected methods and tools relevant to me in my teaching, as well as relevant to the development in the subject I teach.*"

We also found very friendly statements concerning the amount of *quality material made accessible* by one of the tools: "*Those examples and explanations are accessible for the students and to myself when constructing lessons and demonstrations*". The responses indicate that coding instructors generally think their efforts have positive influence. One respondent expressed that "*My own need for doing something good for the students is a rational motive [for maintaining of the code] for me*".

4.2 Second source: Gained insight from development

Developing a boundary Object (the Evalanche System)

The first author of the paper began coding a tool for the administration of learning and teaching, Evalanche [7], in 1999 to gain better teaching skills in Java programming. The ambitions to develop the system were also:

a. to give undergraduate students in college a fast feedback on a great number of exercises;
b. to conquer administrative friction related to the publishing of exercises, collecting answers and computing the results by creating the shortest possible path to the student's work;
c. to support iterative and incremental guidance, where students could involve themselves in a dialog tied to each exercise and each try. The file-set submitted by a student for each try, should also be persisted and easily retrieved anytime.

As boundary object, Evalanche has helped to facilitate skill and knowledge transfer from instructor to student. Since the maintenance was just as much a coding training arena as a tool to support coaching of students, several technologies were exploited to satisfy current needs. The boundary object convey informational objects concerning both social worlds. The handling of students goes on continually. The instructor publishes exercises, the student submits solutions, and the instructor delivers individual guidance and assessment in repeating cycles for each student, each exercise and each try for as long as the student needs. The boundary object emits friendly emails to remind procrastinators. The coder continually extends the boundary object in several interesting ways. A recent example is using IoT, realized by signalling a smart wrist device, to give the instructor an early submission alarm in order to deliver the quickest feedback possible to the student.

Effects of creating and maintaining of codebase

Some instructors of vocational skills may be subject to incredibility caused by the instructor's distance to practical work. Credibility may improve, though, if the instructor can refer to earlier, preferably *recent* practice relevant for the taught topics. The Evalanche codebase maintenance inferred multiple ways and cases to boost the interest and involvement in many existing technical fields, as well as motivation to be exposed to upcoming technologies.

Pattern archaeology, detection and accommodation have been huge resources for tailoring the code. When stuck somewhere in the code, implementing a better pattern for the mechanism in question has often solved the problem. Even *considering* another pattern may be enough to unleash better semantics to help solve a complex problem. Many times bad semantics lead to a state of misunderstanding. Maintaining code infers an appreciation of good semantics. This piece of experience and several more have in turn contributed to the development of specific courses and instruction material in *Software Design Patterns for Object Oriented Systems* [32, 33].

The numerous repeating cycles of counselling for each student, each exercise and each try can improve the student's power of abstract thinking. The boundary object's ability to let the instructor drill students in specific problems improves generalisation and abstract thinking. The iterating and intense training over time add the abstraction ability to the student's problem solving characteristics. The students' ability of abstraction may contribute to reducing marks bimodality [4, 6].

Object oriented problem solving require practical skills as well as a degree of abstraction sometimes scarce among young students. Kunkle [34] states that the choice of a specific introductory language or tool has small effect on the learning results. There is a possibility that a large amount of repetitions stimulates the abstraction abilities needed to understand the solution. The boundary object does exactly that.

A main ambition from the very beginning of the project in 1999 was to remove most of the administrative friction between *emitting exercises* and *finishing their evaluation*. This kind of friction is described in more detail in [7], where there is also a brief discussion of further advantages of

reduced administrative friction. The boundary object allows intense training by quickly connecting instructor, student and submission. Exploiting this characteristic enhances the student learning as well as the credibility and even enthusiasm in the student social world.

Attitudes towards role

A motivation inferred by the teacher role was to let Evalanche boundary object open the shortest possible path between submission and assessment. Another motivation was to engage both social worlds in building a better system, and discussing coding issues affecting their own user experience and specific technical curiosity. The system has registered several instructors. But few have used it over time, similarly to what is stated for System Type b [31], section 4.1. This experience has contributed to knowledge of how to gain peer acceptance for in-house software. Real world experiences from maintaining and *running* multi-user software, have improved the role as instructor and author, as well as helped develop robust attitudes to the social mission aspects of system development.

Among strategies for augmenting acceptance for boundary objects like Evalanche, we suggest to involve one or more applications of artificial intelligence. According to Gartner's Hype Cycle from July 2017, the concept of AI is still ascending towards the top, while ML is just over the top, still 2-5 years before disillusionment. In this perspective, an optimization of the boundary object in terms of AI services would be a mainstream strategy applied at the right time. We elaborate this strategy in section 7, Future research

5 Analysis & Discussion

We have investigated why and how long practice impacts confidence in introductory object oriented programming courses. The instructors under investigation had created boundary objects that inferred self-esteem and perceived student confidence in several ways, positively stimulating learning processes in both social worlds. We kept the intersection between the social worlds of instructors and students of introductory courses of object oriented programming stable, while the boundary objects varied. The 10 respondents had created diverse boundary objects with the common characteristic of augmenting self-confidence and professional experience onto the instructor world, and perceived confidence from the student world. The boundary objects had these characteristics for a variety of reasons. Some added facilities that satisfied the informational requirements or needs of each of the social worlds. Others invoked professional respect by virtue of their size and complexity. We will now enhance the table in section 4 to analyse the system types, and to facilitate a discussion of how specific system types satisfy needs of the social worlds.

Type	
a	This boundary object type is strongly structured within the experiment at UiA [4]. The boundary object was the foundation of a pedagogical experiment targeting bimodal results in introductory programming courses. The report [ibid 4] points out a few weaknesses in the use of the object, and lets us know that the "new course format was a moderate success".
b	This ambitious boundary object concentrates on conveying and assembling learning materials from any thinkable open source. As such that object fits the success criteria of satisfying informational requirements, as claimed by Adam Worrall [35] in his analysis of Star and Griesemer [23] and Meleis [36]. In addition, the construction of the object demanded by its nature a variety of specialized technical insights and colleague cooperation, and thus the boundary object helped transferring the learning results among both colleagues and students.
c	This boundary object type creates distance to the transfer of knowledge between the social worlds of interest here. Instead, the objects engages almost all colleagues from all faculties within one campus, since they affect important parts of the local administration. Therefore,

	their construction infer general professional confidence and a lot of diverse technical experience.
d	For the researcher members of our instructor world, the boundary objects are research tools. They usually satisfy informational requirements either directly or through research reports. The many interesting and very specialized informational needs in that field infers professional respect on restricted and high level arenas.
e	As discussed in section 4.2, a main idea was to support *fast* response to quite *many* submits, and let the student improve the answers iteratively. This may influence what learning edge [1] each student is drawn towards at an early stage.

Characteristics that influence positive perceptions from both social worlds are duration and pedagogical relevance of boundary objects. The commitment to boundary object maintenance is likely to be motivated by the common targets of the two social worlds. The engaged instructor is motivated by the learning outcome, as well as the perception of confidence from the student social world. This assertion strongly conforms to the analysed evidence. Tin et. al [37] found that more than half of the teachers were motivated by students and by teaching itself. A basic assertion is that motivation is good for teaching. We will also assert the positive influence of ownership to your job as a factor that regulates motivation. Hutabarat [38] discusses motivation and performance as important for organization success and outcomes, and states that there is a significant direct effect of job-satisfaction to job-performance. Boundary object maintenance belongs to the factors that enhance the notion of ownership, and thus influences the notion of ownership. The possibility of spending resources on boundary objects rests on confidence and autonomy at work. Teachers in Hutbarat et al. were motivated because principals trusted them and gave them considerable autonomy.

The creation and management of boundary objects is a key process in developing and maintaining coherence across intersecting social worlds [23]. In the intersection of our social worlds, there are some surprisingly difficult knowledge transfer issues. At first, there are syntactic challenges. The transfer of knowledge must happen slowly enough for the student world to accept and incorporate it. After some kairos [39] time shifts, the boundary object shifts state to semantic, and then to pragmatic [29] state. All this happens along the increase in abstraction of the learning challenges. All along the increasing novelty charge [ibid 29], the boundary object must execute its capacity to spawn self-confidence and credibility power and effect within the social worlds. For external codebases this is done through already accumulated reserves of experience, like a few years of industrial experience. For internal codebases it is more likely to be a continuous process with renewable energy and technology as demands to the running systems development activities.

We think our investigation indicates that weakly structured boundary objects of the types we have related, work well in the transfer of knowledge between the two social worlds. In most of their strong structured forms in the intersection of the social worlds, they infer self-confidence and a feeling of student credibility and confidence. From this conclusion, we think that instructors should be encouraged, or at least not discouraged, if they practice relevant coding. In hind sight, we are also convinced that our choice of the social worlds perspective gave us the opportunity to examine the experiences and learning processes that occur in shifting and open constellations of actors. This opens up the broader context as well as the temporal dynamics for analysis [9, 18, 40].

Our analysis drew on the notion of boundary object; a notion that has enjoyed a substantial take-up outside STS [41]. In the original formulation of the notion, the interpretive flexibility was a core characteristic of a boundary object. This would facilitate ‘cooperation without consensus’, i.e. communication and/or coordination could be achieved between different social worlds. However, the worlds would remain different and autonomous. The notion of boundary objects could explain exactly this balance of autonomy with collective action. It is also worthwhile to note that these boundary objects were seen to be the outcomes (rather than causes) of processes where convergence between the

needs of the different worlds had been achieved and maintained [42]. Other studies, such as [29, 43] also recognize boundary objects as 'causes', as instruments to achieve collaboration, and also as means to facilitate knowledge transfer and learning. Our findings also revealed that boundary objects like Evalanche serve as outcomes of the code development process and set the ground to establish collaboration between the student and instructor worlds. It was also an instrument to facilitate knowledge transfer and build self-esteem and confidence for instructors. In our case, the boundary object facilitated learning not merely through mediating or aligning the worlds of students and instructors. The meeting arenas that emerged around the codebase did not just facilitate a 'transfer' of knowledge between different worlds. A more central role was that these meetings between social worlds generated situations where different meanings were exposed. The multiple ways of presentation, examining and processing of concrete boundary objects, let the differences of values, opinions, frames, expectations and preferences become visible. A major role of the boundary object was thus to uncover tensions and to stimulate collaboration. This goes beyond mere mediation [44].

6 Threats to validity

The evidence for this qualitative research is from carefully selected and scarce respondents, as well as one author's reported experiences. In Onwuegbize and Leech [45] we can investigate validity in terms of qualitative research. We learn that interpretivists are encouraged to improve upon frameworks like the Qualitative Legitimation Model [46]. However, our evidence stems from a population being very small by its nature. We therefore claim the results to be interpreted as findings from an exotic and interesting department of the social worlds under investigation. We found strong evidence of positive influence on self-esteem and perceived confidence. But, is it even possible to find negative perceptions of confidence when the respondents are already selected by their confirmation of a perception of vocational self-confidence? One respondent who claimed his self-confidence to originate from his professional grade and situation, rather than from the creation of boundary objects, opens for exactly that possibility. The existence of confirmation bias still threatens this interpretation, since the data set violates a reasonable compliance to rich and thick descriptions, as discussed as early as 1970 by Becker [47].

Onwuegbize and Leech encourages "interpretivists to document how they obtained their data, their interpretations, and their conclusions." The degree to which we have failed to document the data collection and interpretation therefore influences a possible threat to validity.

7 Future research

Our research ambition is to invoke further research and discussions of effects of long term code maintenance for instructors of introductory programming courses, as well as to establish an opinion on the usefulness of such activities. We would like to have repeated the research presented here at a bigger scale, involving a much broader international respondent universe than we could find in Norway at this time.

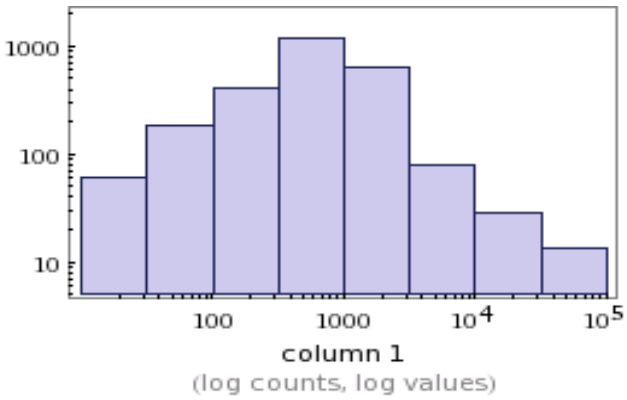
We would also like to investigate the LEM suggested by Robins [1], and link experiments with very fast responses to student submissions. If it is true that the effect of Learning Edge Momentum is particularly strong in introductory programming, then it is of interest to find out if early response could attract students towards the positive edge.

Tools like Evalanche support iterative guidance and assessment to help individual students achieve high levels of marks and enhanced understanding of the exercise. We would like to investigate whether iterative guidance is sustainable in institutional and societal perspectives, since iterative guidance and assessment demand considerable resources from the instructor. The instructor would benefit from predicting exactly which students will benefit from the resources in terms of completing mandatory work, in cases where mandatory work is a prerequisite for admittance to the final exams.

Instructors could then spend more resources on students with the better possibilities. The present situation is that instructors sometimes spend resources randomly and in vain.

Evalanche measures several parameters, as for instance the lag for the student first response on an exercise, as well as the lag for the student's first response to his submission from the instructor. Facilitating short lags is an explicit attribute of Evalanche, for the purpose of enhancing the student learning, motivation and the promotion of positive LEM effects.

To give an idea of response intervals over a long period we created a simple diagram showing the amount of first responses vertically against delay in minutes horizontally. The total amount of first responses is 4060 for the period 2006 to 2011.



The mean of responses is 1599, where the response delay is around 1000 minutes. Such a response frequency combined with low delays is made possible by reducing administrative friction[7]. Still, an interesting question is whether the high response rate is worth the effort. Following this question, it becomes even more interesting to consider ML techniques as an application of AI to provide predictions of the success rate in terms of student mandatory work.

We indicate a simple design of supervised learning of the outcome of response rates in the simple table below. The actual implementation is slightly more involved.

Student	diff(T(submit),T(served)) in minutes	Mandatory Success
1	10	true
2	200	false
3	320	true
4	1020	true
5	23000	false
6	8	false

We indicate with this table that both high lag values and very low lag values for an exercise may combine with false values. Even considering that each student will have measures from several exercises, we would have to decide on using the average experience of delay, the summed total, or put

more weight on the position in time for the measured lags. Anyway, we would hesitate on predicting a settled failure based on this design alone.

A more robust approach would be to access and engage all available student behavioural measures in Evalanche, to build a neural network of 0..1 values for perceptrons, with the outcome of predicitive analytics for mandatory success. In a multi-layered neuron design we would consider adjustable eights, identified by weight limits to predict whether a student with such output is likely to be admitted to the final exams. This design involves learning algorithms on the several layers of neurons. After a learning period supplied with tests we would decide whether it was actually possible to predict the chance of mandatory success based on social world behaviour for this boundary object. If not, it would be open to further research to optimize the boundary object for more precise and targeted measures of student behaviour in order to balance resource usage on individuals in the student social world.

Comparable open source LMS instances

Several interesting open source LMS instances exist as comparable boundary objects among similar social worlds as ours. A few examples of standardized tools are Moodle, ILIAS and Sakai, and less open source projects like Khan Academy and Canvas. They all go far beyond Evalanche in marketing, purpose and appeal. The first coding of ILIAS happened almost simultaneously with Evalanche. ILIAS reached SCORM [48] compliance a few years later. Moodle was released around 2002, and Sakai a little later. Those LMS structure learning contents well. Evalanche do not, but emphasizes fast and personalized instructor feedback.

An instructor from our social world could benefit from participating among the open source developers of one of the standardized LMS tools. The perceived effects of creating and maintaining codebase fragments could even be better, or at least similar, as member of a developer community, compared to working inside the more limited and private Evalanche project. The actual chosen LMS and its policies would, however, limit the freedom to explore several very different technologies, communication variances and presentation forms. There would also still be an issue to have colleagues, and the institution, to accept same LMS.

References:

[1] Robins, A., *Learning edge momentum: a new account of outcomes in CS1.* Computer Science Education, 2010. **20**(1): p. 37-71.

[2] Ryoo, J., F. Fonseca, and D.S. Janzen. *Teaching Object-Oriented Software Engineering through Problem-Based Learning in the Context of Game Design*. in *Software Engineering Education and Training, 2008. CSEET '08. IEEE 21st Conference on*. 2008.

[3] Holland, S., R. Griffiths, and M. Woodman, *Avoiding object misconceptions*, in *Proceedings of the twenty-eighth SIGCSE technical symposium on Computer science education*. 1997, ACM: San Jose, California, USA. p. 131-134.

[4] Nilsen, H. and E.Å. Larsen, *Using the Personalized System of Instruction in an Introductory Programming Course.* NOKOBIT, 2011. **2011**.

[5] Keller, F.S., *Good-bye, teacher.* Journal of Applied behavior analysis, 1968. **1**(1): p. 79.

[6] Robins, A., *Learning Edge Momentum*, in *Encyclopedia of the Sciences of Learning*, N.M. Seel, Editor. 2012, Springer US: Boston, MA. p. 1845-1848.

[7] Holmstedt, V., *Regulering av oppgavefrekvens gjennom automatisering*. Vol. 4/2002. 2002, Tønsberg: Høgskolen i Vestfold.

[8] Clarke, A.E., *AS ORGANIZATIONAL THEORY.* Social organization and social process: Essays in honor of Anselm Strauss, 1991. **119**.

[9] Clarke, A.E. and S.L. Star, *The social worlds framework: A theory/methods package.* The Handbook of Science & Technology Studies, 2008. **3**: p. 113-137.

[10] Mark, G. and S. Poltrock, *Groupware adoption in a distributed organization: transporting and transforming technology through social worlds.* Information and Organization, 2004. **14**(4): p. 297-327.

[11] Strauss, A.L., *A Social Worlds Perspective.* Studies in Symbolic Interaction, 1978. **1**: p. 119-128.

[12] Strauss, A.L., *Social Worlds and Legitimization Processes*. Studies in Symbolic Interaction, ed. N. Denzin. 1982, Greenwich CT: JAI Press.

[13] Garrety, K. and R. Badham, *The Politics of Socio-technical Intervention: An Interactionist View.* Technology Analysis & Strategic Management, 2000. **12**(1): p. 103-118.
[14] Strauss, A.L., *Continual permutations of action Aldine de Gruyter.* New York, 1993.
[15] Shibutani, T., *Reference Groups as Perspectives.* American Journal of Sociology, 1955. **60**(6): p. 562-569.
[16] Unruh, D.R., *The Nature of Social Worlds.* Sociological Perspectives, 1980. **23**(3): p. 271-296.
[17] Bijker, W.E., T.P. Hughes, and T.J. Pinch, *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology*. 1987: MIT Press.
[18] Strauss, A.L., *Scientists and the evolution of policy arenas: The case of AIDS.*, in *stone symposium of the society for the study of symbolic interaction*. 1991: San Fransico, CA.
[19] Gieryn, T.F., *Boundaries of science.* Handbook of science and technology studies, 1995. **393**.
[20] Gkeredakis, E. and K. Samiotis. *The Process of Creating Boundary Objects: The Case of a Knowledge Management IT Artefact*. in *OLKC 2006 Conference at the University of Warwick, Coventry on 20th–22nd March*. 2006. Citeseer.
[21] Von Hippel, E., *"Sticky information" and the locus of problem solving: implications for innovation.* Management science, 1994. **40**(4): p. 429-439.
[22] Szulanski, G., *Exploring internal stickiness: Impediments to the transfer of best practice within the firm.* Strategic management journal, 1996. **17**(S2): p. 27-43.
[23] Star, S.L. and J.R. Griesemer, *Institutional ecology,translations' and boundary objects: Amateurs and professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39.* Social studies of science, 1989. **19**(3): p. 387-420.
[24] Gal, U., Y. Yoo, and R. Boland Jr, *The Dynamics of Boundary Objects, Social Infastructures and Social Identities*, in *Sprouts Working papers on information systems,*. 2004, ase Western Reserve University: Cleveland, Ohio.
[25] Gal, U., Y. Yoo, and R. Boland Jr, *The Dynamics of Boundary Objects, Social Infastructures and Social Identities.* ECIS 2005 Proceedings, 2005: p. 57.
[26] Henderson, K., *Flexible sketches and inflexible data bases: Visual communication, conscription devices, and boundary objects in design engineering.* Science, technology & human values, 1991. **16**(4): p. 448-473.
[27] Pawlowski, S.D. and D. Robey, *Bridging user organizations: Knowledge brokering and the work of information technology professionals.* MIS quarterly, 2004: p. 645-672.
[28] Tudhope, D., P. Beynon-Davies, and H. Mackay, *Prototyping praxis: Constructing computer systems and building belief.* Human-Computer Interaction, 2000. **15**(4): p. 353-383.
[29] Carlile, P.R., *Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries.* Organization science, 2004. **15**(5): p. 555-568.
[30] Van de Ven, A.H., *Engaged scholarship : a guide for organizational and social research*. 2007, Oxford ; New York: Oxford University Press. xii, 330 p.
[31] Stenseth, B. *WXT*. 2009 2009 [cited 2016 2016-03-23]; Publishing resource]. Available from: http://www.ia.hiof.no/wxt/wxtsite/.
[32] Holmstedt, V., *Mønstre og tekniske applikasjoner*. 2007, Bergen: Fagbokforlaget, Norway.
[33] Holmstedt, V. *Objectification*. 2010 [cited 2017 22 March]; List of teaching materials]. Available from: https://sites.google.com/site/mitsoptew/.
[34] Kunkle, W.M., *The impact of different teaching approaches and languages on student learning of introductory programming concepts*. 2010: Drexel University.
[35] Worrall, A., *Boundary Object Theory: Concepts, Propositions, and Limitations* Florida University: Florida University. p. 20.
[36] Meleis, A.I., *A model for evaluation of theories: Description, analysis, and critique of theory.* , in *Theoretical nursing: Development and progress* J.B. Lippincott, Editor. 1991, J. B. Lippincott: Philadelphia. p. 213-245.
[37] Tin, L.G., L.L. Hean, and Y.L. Leng, *What motivates teachers.* New Horizons in Education, 1996. **37**(1): p. 19-27.
[38] Hutabarat, W., *Investigation of teacher job-performance model: Organizational culture, work motivation and job-satisfaction.* Asian Social Science, 2015. **11**(18): p. 295.
[39] Jaques, E., *The form of time*. 1982: Crane, Russak New York.
[40] Clarke, A., *Situational analysis: Grounded theory after the postmodern turn*. 2005: Sage.
[41] Zeiss, R. and P. Groenewegen, *Engaging boundary objects in OMS and STS? Exploring the subtleties of layered engagement.* Organization, 2009. **16**(1): p. 81-100.
[42] Star, S.L., G.C. Bowker, and L.J. Neumann, *Transparency beyond the individual level of scale: Convergence between information artifacts and communities of practice.* Digital library use: Social practice in design and evaluation, 2003: p. 241-269.

[43] Carlile, P.R., *A pragmatic view of knowledge and boundaries: Boundary objects in new product development.* Organization science, 2002. **13**(4): p. 442-455.
[44] Jonsson, K., J. Holmström, and K. Lyytinen, *Turn to the material: Remote diagnostics systems and new forms of boundary-spanning.* Information and Organization, 2009. **19**(4): p. 233-252.
[45] Onwuegbuzie, A.J. and N.L. Leech, *Validity and qualitative research: An oxymoron?* Quality & Quantity, 2007. **41**(2): p. 233-249.
[46] Maxwell, J.A., *Qualitative research design (Vol. 41)*. 1996, Thousand Oaks, CA: Sage Publications.
[47] Becker, H.S., *Sociological work*. 1971: Transaction publishers.
[48] Kazi, S.A. *A conceptual framework for web-based intelligent learning environments using SCORM-2004*. in *Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on*. 2004. IEEE.

VIGGO HOLMSTEDT
University College of Southeast Norway
Department of Economics and Managemen
NORWAY
E-mail: vh@usn.no

SHEGAW A. MENGISTE
University College of Southeast Norway
Department of Economics and Management
NORWAY
E-mail: sme@usn.no