

# A Quantitative Analysis of a Virtual Programming Lab

Jan Vanvinkenroye\*, Christoph Grüninger†, Claus-Justus Heine‡, and Thomas Richter\*

\* *Technische Informations- und Kommunikationsdienste*

*University of Stuttgart, Allmandring 3a, 70550 Stuttgart, Germany*

*{jan.vanvinkenroye, richter}@izus.uni-stuttgart.de*

† *Department of Hydromechanics and Modelling of Hydrosystems*

*University of Stuttgart, Pfaffenwaldring 61, 70569 Stuttgart, Germany*

*christoph.grueninger@iws.uni-stuttgart.de*

‡ *Institute for Applied Analysis and Numerical Simulation*

*University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany*

*claus-justus.heine@ians.uni-stuttgart.de*

**Abstract**—In a recent publication, we described the design and implementation of a web-based programming lab (ViPLab) targeted at undergraduate Engineering and Mathematics courses. While the former article covered the underlying technology and briefly described early experiences, this work provides a quantitative analysis of the user feedback, experience and learning success. To this end, we implemented a survey with one learning group using the web-based tools and a control group working with a traditional setup based on editor and compiler. The survey shows that web-based installations are as efficient as classical tools, while Windows users prefer the web-based chain over the editor/compiler installation on Linux.

**Keywords**—learning success; education; online learning; online programming; virtual laboratories;

## I. INTRODUCTION

In [1], we described the technological design and early experiences of the web-based programming lab ViPLab at the University of Stuttgart. This tool addresses the needs of engineering and natural sciences Bachelor courses by providing a steeper access to programming tools than the more traditional compiler/editor or integrated development environments (IDE) installations: In the experience of the authors, students typically spend too much time to install such tools and get acquainted with them, just to be able to perform their homework assignments. However, due to the change to the Bachelor/Master program at German universities, only little time remains available for such activities as especially undergraduate courses had to be streamlined to fit into the relatively tight Bachelor study plan. ViPLab addresses these needs by providing an entirely web-based solution that integrates into the eLearning system of the university. It allows students to perform simple programming exercises online with a web-browser within the Learning Management framework offered by the university for all its courses. More details on the motivation and the technological background of ViPLab can be found in our earlier work [1].

ViPLab is now for more than six semesters in use at the University of Stuttgart, enough time to collect some

feedback for further development and to allow an estimate of its efficiency and effectiveness. It is therefore paramount to analyze and to systematize students' and lecturers' feedback, and to perform a quantitative analysis to evaluate our development. For example, lecturers receive the impression that ViPLab is better suited for students who are less trained in the use of computers. In contrast, tech-savvy students are impeded by the smaller functional range of ViPLab compared to a fully featured development environment. It was the purpose of the study described in this work to ascertain whether these qualitative impressions are correct or not.

### A. Related Work

The idea to provide a web-based access for programming exercises is not new, and earlier work in this area exists. For example, the VPL module for moodle [2] provides a similar programming lab that depends, however, specifically on the moodle architecture and cannot be integrated into other Learning Management Systems (LMS), quite unlike the SCORM [3] based ViPLab that depends on an established eLearning standard offered by most LMS's. Professional solutions from commercial vendors like MapleTA [4], [5] are also available on the market, but are typically constraint to fill-in exercises in Mathematics, using the Maple or Matlab syntax, and do not allow free programming.

However, while most of these tools claim to be useful and effective, and even likely be so in practical applications, a truly quantitative and systematic study on their usefulness is rarely found. It is the purpose of this article to fill this gap by providing the results a systematic study performed at the University of Stuttgart in 2013.

### B. Goals of the Survey

The usefulness of learning tools like ViPLab is defined by and can be estimated by the learning success of students using it, of course. Learning success can be measured by an achievement test, which will be described in more detail

in section II. The goal here is naturally not only to gain scientific insight, but also to hopefully make a good case for our learning tools and attract lecturers. Additionally, we were also interested in the students' assessment of our web-based solution compared to a traditional solution; to this end, we designed a survey that measures the students' satisfaction with either tool. Thus, the key questions we wanted to find answers on were: Is there a difference in students' achievement and satisfaction depending on either using ViPLab or a locally installed programming environment? Which impact has the student's tech savvy-ness?

This work is structured as follows: In the next section, we describe the test setup and the implementation at the University of Stuttgart. The test tools to answer the above questions are formulated and specified in the following section. Results are then shown and discussed in two following sections. We conclude with an outlook on future work.

## II. IMPLEMENTATION

Sixty bachelor students of Mathematics from the University of Stuttgart took part in the study; the experiment was part of the block seminar which is mandatory for students in their third semester. The learning goal of this course is to gain some basic understanding of the C++ language, to enable students to develop small-scale C++ programs where the focus lies primarily on mathematical applications and numerical mathematics. The course covers C++ control structures, data structures, and simple I/O operations. In specific, the goal of the course does not include how to setup larger scale projects, how to use makefiles or IDEs like Eclipse or Visual Studio. While the latter learning goals are surely useful for software engineers, they are less important for mathematicians and the tight course design did not allow to include them. ViPLab was only used for the first two days to teach basic structures.

Students were split randomly into a test group using the web-based ViPLab, and a control group using a compiler and editor on a Linux system. The test setup was in both cases prepared by the same lecturer, and students were asked to come to the PC pool of the university to perform their assignments there.

### A. Course Design

The course started with a lecture, which was identical for both the test and the control group. Following the lecture, students of both the test and the control group performed hands-on exercises in the PC pool, using either the traditional or the ViPLab-based approach. Despite the tools available for the students, the computer setup was identical, and based on Debian Linux. Exercises were performed in the same room of the computer pool and were conducted by the same lecturer.

The test group worked, as described, with ViPLab which was started directly from the Learning Management System

(LMS) of the university, which happens to be ILIAS in Stuttgart. Those in the control group received code archives distributed by the lecturer, including incomplete code fragments, and worked with the restricted Geany [6] editor. The Geany editor shares some similarities with integrated development environments, but provides a much simpler user front-end. Figure 1 shows both user interfaces with a loaded exercises and a compiler error and provides some impressions on the tool chains used. The setup of the control group closely resembles the traditional setup, except that students did not have to install the compiler and editor themselves and worked with university machines in the PC pool, not their own computer systems at home. Thus, any difficulties and problems setting up the development environment were not covered.

Students of both test and control group were asked to answer a survey at the end of the course. The design of this survey is described in more detail in the following.

## III. TEST TOOLS AND TEST DESIGN

The primary purpose of the survey was to compare the effectiveness of the learning tools towards the formulated learning goal. As already mentioned above, we were also interested in the students' satisfaction with the tools provided.

This leads to the following questions: Is there a difference in student achievement and satisfaction depending on either using ViPLab or a locally installed programming environment? Which environment do they prefer to work with, and is there an impact on the students' tech savvy-ness on their preference? To measure the latter, we use the established test INCOBI-R [7], [8]. For the former, we took questions from Gruber [9] which evaluate student satisfaction.

The INCOBI-R test evaluates three scales: The practical computer knowledge (PRAECOWI), the theoretical computer knowledge (TECOWI), and a scale concerning the assurance in using computers (COMA). These three aspects of declarative knowledge, procedural knowledge, and emotional assurance with regard to use computers represent *Computer Literacy*. From these scales, the test compiles a score that estimates to which extend the students have previous knowledge, which helps or impede to learn programming. In addition, we asked further for the primary operating system the students are acquainted with.

The TECOWI and PRACOWI scales provide scores from zero for the lowest value to one for the highest value, COMA from one for a low level of assurance to three for a high level. The satisfaction can range from one for unsatisfied students to five, completely satisfied. The maximum score in the achievement test is three.

In the overall test design, the independent variable is the choice of the learning environment, i.e. ViPLab versus the traditional setup, and the dependent variables are the satisfaction with the used tools and the score in the achievement

	Control Group	Treatment Group
N	35	22
Score	1.88 (0.86)	2.04 (0.65)
TECOWI	0.50 (0.22)	0.62 (0.21)
PRACOWI	0.59 (0.22)	0.66 (0.19)
COMA	2.52 (1.03)	2.18 (0.76)
Satisfaction	2.76 (1.03)	3.12 (0.61)

Table I  
EVALUATION RESULTS OF THE ACHIEVEMENT TEST, AND STUDENT SATISFACTION SCORES ACCORDING TO INCOBI-R. THE STANDARD DEVIATION IS GIVEN IN BRACKETS.

test. We use the previous knowledge with three variables TAECOWI, PRACOWI, and COMA as control variables.

#### IV. RESULTS

Table I shows the results of the achievement test score (second row) and the student satisfaction with regard to the used tool, and the averaged dimensions from the INCOBI-R (TAECOWI, PRACOWI, and COMA). The left column lists the scores of the control group, the right of the treatment group. The standard deviation is listed in brackets.

Except for the score in the achievement test, the results are not normally distributed. Thus we use the *Wilcoxon signed rank test*, cf. [10] to test our hypothesis: The null hypothesis  $H_0$  is here that there is no difference between control and treatment group. The alternative hypothesis  $H_1$  is that there is a difference. The value of the treatment group for satisfaction is higher compared to the control group, but the level of significance of 0.05 misses the asymptotic significance of 0.073 of the signed rank test. Thus, there is *no statistically significant difference* of user satisfaction between treatment group and control group.

The second question to be answered is whether the students' tech savvy-ness has any impact on the test results. For this, we compared the single variables with the *Spearman rank correlation coefficient*. Here we get only one statistically significant result, namely a negative correlation of the satisfaction of the control group with the choice of Windows as primary operating system (Spearman  $\rho = -0.569$ , significance of 0.0002). The assumption that there is no difference in the satisfaction of the two groups depending of the primary used operating system is the null hypothesis  $H_0$ . The alternative hypothesis  $H_1$  is that there is such a difference. Using a variable for primary Windows users in the Wilcoxon signed rank test, we can refuse the null hypothesis  $H_0$  by a level of significance of 0.05 with the asymptotic significance of  $> 0.0001$ .

#### V. DISCUSSION

We have shown that the use of a web-based tool to learn elementary programming skills is not rated significantly different than a pre-installed traditional compiler/editor framework installed on a PC pool at the university. Neither does

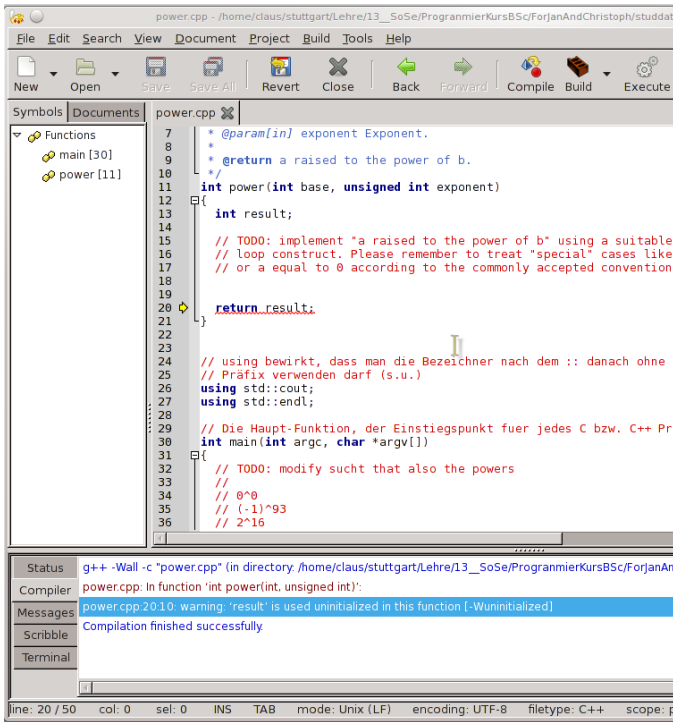
the choice of the tool have any significant impact on the learning success. This is in so far plausible as students use a quite similar graphical input mask to edit the source code and receive feedback from the compiler. Note that the task of *installing* these tools, i. e. the compiler and editor, at the PC at home was not covered or included in this survey. That is, *ViPLab can provide as good results as a well-prepared exercise in a PC pool*.

User satisfaction does, however, depend on the primary operating system students are acquainted with. Apparently, Windows users seem to dislike the Unix-style installation of editor and compiler more than users with other or little background in computing. This supports the impression of the lecturers that ViPLab has a strong point for supporting rather diverse user groups consisting of both experiences and inexperienced computer users.

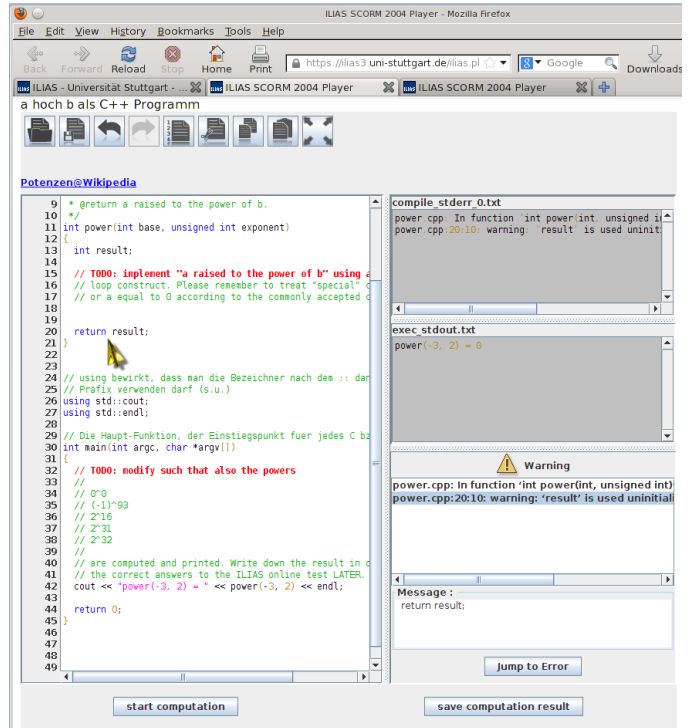
#### VI. CONCLUSION AND OUTLOOK

Even though there is no statistical significance of the learning success on the tools used, we want to emphasize that this is actually a positive result already: On one hand, the described experiment allows a quantification of using ViPLab compared to a well prepared C++ course. On the other hand, in this setting ViPLab cannot show its strengths as described in our earlier work [1]: The lecturer does neither need to install the needed software nor does he need to explain its use: that is, we have not tried to evaluate here the amount of time to be spend by a lecturer to actually *prepare and setup* a good hands-on course on programming lab. We neither tried to evaluate in how far students satisfaction depends on whether they do their homework at their machine at home compared to whether they would have to go to a PC pool, or would need to install compiler and editor themselves. This is left to a future study. The problem here is, however, that students may consider it unfair to be assigned to one group or another, i. e. it might turn out hard to ensure comparable conditions to make the test statistically relevant.

Another difference of practical relevance is that ViPLab allows students to use Matlab without having to acquire a license – the computation runs server side and depends solely on the campus license of the university. This becomes even more relevant for numerical applications that are closer to research where setup times of the traditional installation will grow significantly. The application we have in mind here are Bachelor thesis' based on the DuMuX simulation framework: A bachelor thesis has to be completed in a couple of weeks rather than months, yet the setup and installation times and the time to get acquainted with the DuMuX tool chain can become quite lengthy. ViPLab with a DuMuX back-end allows students to delve into the applications right away and focus on the scientific work rather the installation work. We have at this time no experimental



(a) Editor Geany



(b) ViPLab

Figure 1. Screenshot of (a) the editor Geany used by the control group and (a) ViPLab used by the treatment group

setup to quantitatively compare these uses but the advantages are quite evident.

Currently, we believe that the main contribution of ViPLab for the improvement of undergraduate studies is that it saves the lecturer valuable time to prepare the exercises, and thus to focus more on the quality rather than the technicalities of the homework, but we have to prove this assumption by further systematical examinations. The planned feature to automatically evaluate homework – which got recently funded – might be an additional factor. We also follow our plans to deploy ViPLab in electronic exams where it would allow evaluation of programming skills right on the machine, without depending on unrealistic pen-and-paper exercises.

## REFERENCES

- [1] Th. Richter, S. Rudlof, B. Adjibadji, H. Bernlöhr, C. Grüninger, C.-D. Munz, A. Stock, C. Rohde, and R. Helmig, “ViPLab – a virtual programming laboratory for mathematics and engineering.” *Interact. Techn. Smart Edu.*, vol. 9, no. 4, 2012.
- [2] J. C. Rodriguez-del-Pino, “VPL – Virtual Programming Lab for Moodle,” <http://vpl.dis.ulpgc.es/>, accessed 2011.
- [3] A. D. Learning, “SCORM Documentation Suite,” <http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/2004%204th%20Edition/Documentation.aspx>.
- [4] Maplesoft, “MapleTA 7: Online Testing and Assessment,” <http://www.maplesoft.com/products/mapleta/>, accessed 2011.
- [5] A. Heck, “Assessment with Maple T.A.: Creation of Test Items,” [http://www.adeptscience.co.uk/products/mathsim/mapleta/MapleTA\\_whitepaper.pdf](http://www.adeptscience.co.uk/products/mathsim/mapleta/MapleTA_whitepaper.pdf), technical Report, accessed 2011.
- [6] “Geany website,” <http://www.geany.org/>, accessed: 2013-07-29.
- [7] T. Richter, J. Naumann, and N. Groeben, “Das Inventar zur Computerbildung (INCOB): Ein Instrument zur Erfassung von Computer Literacy und computerbezogenen Einstellungen bei Studierenden der Geistes- und Sozialwissenschaften.” *Psychologie in Erziehung und Unterricht*, vol. 48, no. 1, pp. 1–13, 2001.
- [8] T. Richter, J. Naumann, and H. Horz, “Eine revidierte Fassung des Inventars zur Computerbildung (INCOBI-R).” *Zeitschrift für pädagogische Psychologie*, vol. 24, no. 1, pp. 23–37, 2010.
- [9] T. Gruber, S. Fuß, R. Voss, and M. Gläser-Zikuda, “Examining student satisfaction with higher education services: Using a new measurement tool,” *International Journal of Public Sector Management*, vol. 23, no. 2, pp. 105–123, 2010.
- [10] J. Bortz and N. Döring, *Forschungsmethoden und Evaluation: für Human- und Sozialwissenschaftler*, 4th ed. Berlin, Heidelberg: Springer, 2006.