Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 96 (2016) 826 – 837

Procedia Computer Science

19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

The PASCA: a Mail Based Randomized Blinded Peer Assessment System for Complex Artifacts

Alexey Neznanov[a], Olga Maksimenkova[a]*

[a]*National Research University Higher School of Economics, Moscow, Russia*

Abstract

Active learning and formative assessment techniques are the cutting edge of the modern education. This paper considers peer assessment automation and touches the topic of high actuality in advancing ICT for active learning. The roots of the study are obtained difficulties in adoption of the currently available peer assessment systems to engineering education introductory courses. The main goal of the paper is to collect software requirements and to build a peer assessment system, which may be easily agreed with standard educational routines (e.g. e-mailing) and which is be able to support complex artifacts interchange during a peer assessment session. The software requirements specification (SRS) for such a system has been created as a result of reviewing educational studies, technical reports and academic publications on common peer assessment processes and existing peer assessment systems. We also introduce the open source Mail-based Randomized Double-Blinded Peer-assessment System for Complex Artifacts called PASCA, which is developed according to this SRS and the results of PASCA adoption to an introductory programming course.

*Keywords:*self-assessment, computer supported collaborative learning, active learning, computer science education, programming education, educational software, blinding, randomization, collaborative learning;

* Corresponding author. Tel.: +7(495)772-95-90 * 22672.
E-mail address: omaksimenkova@hse.ru

doi:10.1016/j.procs.2016.08.261

1. Introduction

Active learning practices become more popular among the educators all over the world these last decades. At the very beginning active learning was introduced as a teaching approach[1], which expected students doing things and thinking about this things. Later, active learning has been expanded by including formative assessment techniques[2]. This techniques are strongly student oriented, involve students in educational planning, and provide feedback, which students may treat as an algorithm or a guideline in their next steps in learning. That is the reason why the questions related to this type of assessment are widely studied and discussed by the active learning practitioners of different fields of knowledge[3,4,5,6]. By now, formative assessment has settled as a powerful and effective education approach.

At the same time, the appearance of close relation between active learning and formative assessment causes the evolution of collaborative and cooperative assessment techniques[7,8]. *Peer assessment* (PA) seems to be one of the most popular and well-studied collaborative formative assessment methods[9,7,10]. Peer assessment means a learning procedure of evaluating where students revise each other's works, evaluate them according to priory formulated criteria and provide feedback.

Of course, the era of total computerization, telecommunication, and distributed systems brought into live lots of educational software for distance and *computer supported collaborative learning* (CSCL). For now, there are plenty *peer assessment systems* (PAS) have been developed along with *learning management systems* (LMS), e-learning and *massive open online courses* (MOOC) platforms. Besides that, the application of PASs, which support users' interaction while peer assessment, have been documented in plenty academic works[3,11,12,13].

Undoubtedly, CSCL seems to be appropriate to a such hi-technology area as *computer science education* (CSE). The trends of global education mentioned above stimulate such main streams in technical education as *Science-Technology-Engineering-Mathematics* (STEM) and *Conceive-Design-Implement-Operate* (CDIO)[14]. This in its turn forced CSE society to start the adoption of active learning and formative assessment methods. The meaningful changes have been done even in basic parts of the *Computer Science Curricular* 2013[15,16].

As a result of this high interest, several works about the practice of using formative assessment have been appeared in the variety of *computer science* (CS) courses[17,18]. But, the experience of automated PA implementation within such courses is not well-documented. Moreover, papers, which deal with PASs development[19] and adoption[20,4] to CSE are rare and leave several gaps.

First, there is no systematically analyzed and compared requirements of different educational domains. Thus, most of PASs[21,22] allow to submit only a set of text fields and do not support any other type of a submission or an attachment, for example an archive or several files. As far as some courses specific to CSE deals with sets of source-code files it is difficult to adopt these systems to anonymized PA without losing the anonymity of students.

Second, existed PASs are quite overspecialized and the flexibility of their preliminary installation is relatively low. It is well-known that PA may be randomized[23], contain a self-assessment step[9], be implemented by groups or by individuals, etc. Furthermore, PAS should have a possibility to be tuned in properly to support several variants of PA tracks.

Third, explicit *software requirements specification* (SRS) for the PAS suitable for the particular educational domain is missed (authors have dedicated the only SRS, which was published online as a result of education project [http://www.it.iitb.ac.in/frg/wiki/images/5/5c/Grp5_MOOCs_SDD_2013_06_26.pdf]).

Forth, PA is a complicated assessment procedure, which needs students to be familiar with[3]. If automated PA is used, the extra instructional problems are risen, because students should meet not only a new assessment procedure, but a new software as well. The pilot PA session is a good answer, but if we take into consideration first-year CS or humanities students, we ought to select PASs with as easy user's interface and use cases as possible. Moreover, for mentioned groups of students it is significant not to spend tons of time on learning extra educational software. Consequently, if PAS is stand-alone and is not implemented as a part of corporate *learning management system* (LMS) it should be integrated with routines, which are familiar to the most of students and teachers, for example, e-mailing.

The general purposes of this paper are: 1) to agree CS educational needs with worldwide PA practice; 2) to overview existing PASs' implementations; 3) to introduce the original Mail-based Randomized Double-Blinded Peer-assessment System for Complex Artifacts called PASCA in conjunction with requirements analysis and the findings of its first adoption.

2. Domain and related works

In this section we provide a reader with a clear context of educational and technological directions, which are combined in this work.

2.1. On peer assessment methodology

PA as a form of formative assessment has rather short but rich history. In different countries and knowledge area educators have held experiments and described studies connected with PA implementation[24], efficiency[25], scaling, etc. Being interested in PAS development in this sub-section we have generalized the works suitable to collect software requirements and to understand a peer assessment processes.

Several review papers, which were published by the leading educational scientists between 1995 and 2015, were taken into consideration. In 1998 Topping reviewed 109 research papers[23], which had been published between 1980 and 1996. He underlined the significance and the necessity of participants' matching and randomization within the PA process. Following this result, we may consider that a flexible PAS *should implement* high quality *methods of randomization*.

A year later, Dochy and the colleagues reviewed quantitative studies on self- and peer assessment[9]. They dedicated a great formative role of PA and singed out the significance of clear predefined assessment criteria. Nearly the same result was given in the meta-analysis of comparative studies by Falchikov and Goldfinch[7], who also emphasized that PA may be used only to provide formative feedback. In case of automatization, it means that a PAS *should support flexible grading rubrics and formative feedback*.

In 2010 Kollar and Fischer[26] introduced the review, which concerned on cognitive facilities of PA, but also contained a part on PA processes modelling. From the results of this work the users' roles in PA process are defined as *an assessee* and *an assessor*. The assessee sends his work to be evaluated, and the assessor evaluates the work received and gives a formative feedback. In PASCA instead of assessee and an assessor we use correspondingly *a submitter* and *a reviewer*. Let us admit that during a participation in a PA a student generally plays both these roles.

Summarizing all above we should say that lots have been done in the field of PA investigation, implementation and even automatization. The presented body of literature make it possible to collect basic software requirements and define basic roles in PA processes.

2.2. Computer supported peer assessment: challenges and solutions

In this sub-section we have reviewed investigations on CSE and on various PASs, which provide full or partial support of PA. Because of our great interest is collecting specific technical requirements and understating details of an adoption process, we also have included the review of existing PA solutions.

Actually, PA adoption to CS courses, especially to computing and programming courses, is quite well-studied. Thus, Isomottonen and Tirronen[11] discussed PA as a part of self-directed learning on functional programming course. The results of PA were used to manage the self-directions, but no specific PAS was adopted during the instructions. On the contrary, Hundhausen, Agrawal, and Agarwal[27] presented the *Online Studio-Based Learning Environment* (OSBLE). This software was used to support the research process and to collect data from several investigations on the PA implementation of programming code review. Furthermore, Hundhausen and colleagues referenced several web-based PASs with different set of functions (e.g. anonymization, feedback). Despite the paper dedicated valuable results to CSE, it did not even give a recommendation on PAS development or adoption as, for example, it had been done by Isomototonen and Tirronen[11]. But, we may define several data objects in terms of Hundhausen and colleagues: *code solution* and *code review*. Nevertheless, PASCA was first adopted to an introductory programming course, it may support PA with artifacts to any course. Here and after we will use more general terms *submission* and *review* for mentioned data objects.

Surprisingly, it is turned out that web-based PASs are probably the most popular in this class of educational software[28]. Lin, Liu, Yuan and the colleagues have published a coherent cycle of articles which is devoted to web-based PAS *NetPeas*[29] and to the educational aspects of its adoption to CS courses[30] and to STEM education[19]. The study of the model underlying the *NetPeas* demonstrated a high formative orientation of the system. This means that the system supports a reaction on review and the uploading of modified work. In spite the fact of such detailed investigations were done and described as for the system development as for its adoption, we have not found much information about its specific characteristics, project solutions, and basic algorithms.

The quite circumstantial description of a web-based PAS's design was given by Anson and Goodman[3]. The primary goal of their development was to design and adopt a system to support peer feedback interchange. The main disadvantage in the context of our paper is that the Anson's and Goodman's system does not support interchange of any additional data objects, like a source code files. But, during the PASCA adoption we can follow the significant procedural piece of advice that students should have a *preliminary practice with PAS*.

Yet another web-based PAS called *MyPeerReview* was introduced by Hyyrynen and the colleagues[22]. This open-source system was developed to be used directly in computer programming courses and the study of its application to web-programming course described in paper by Hamalainen and the colleagues[6]. It should be mentioned that in idem papers the authors discussed several existing systems and concluded their procedural and technique limitations to CSE. There are two limitations of *MyPeerReview* in our case. At first, the system seems to be unsupported nowadays. At second, we are going to keep a set of tools applied to our educational process constant. But we share the idea[27] of post-test *students' feedback on the system collection and analysis*.

The most popular and widely used by several universities PAS called *Aropä* was introduced by Hamer and the colleagues in the middle 2000s[21]. Today *Aropä* is a powerful web-based educational system developed according to PA principles. It supports such features as a review of reviews, double-blinded dialogues between a submitter and a reviewer, and time-on-tasks. However, despite the fact the authors succeed in applying *Aropä* to CS courses[31], the system does not support any types of artifacts in submissions, it is also hardly integrated to the routine business-processes, and expands the set of educational tools. Nevertheless, *Aropä* is probably the first PAS with extended peer review cycle.

As far as we can conclude, the majority of implementations do not support any type of artifacts except text. By this line of reasoning, the most of reviewed PASs appear to reproduce paper-and-pencil form of the appropriate educational activity. Thus, we cannot evaluate special requirements associated with complex artifacts like software application solutions, diagrams, databases and the others, which frequently appear in STEM courses.

3. Software requirements specification for PAS

The features and requirements for a PAS, which were collected from the domain and technical reviews (sections 2.1, 2.2), are generalized in this section. The result of this generalization is given here in a brief SRS form.

3.1. Main concepts of peer assessment process

Below in this paper a *Peer Assessment* (PA) is defined as an assessment procedure organized in form of randomized *Peer Review* (PR) of arbitrary artifacts threated as the results of an assignment with a priory formalized assessment criteria. The process of PA of the results of single assignment is called a *PR session*.

Main roles of PA processes:

1. **Teacher** – any organizer or manager of a PA process with full access to a PA data objects.
2. **Student** – any trainee who participates in a PR session.
3. **Initial Author** – a student who was registered as future Submitter in a PR session.
4. **Submitter** – an Author who creates an artifact and submits it to the PAS.
5. **Reviewer** – a student who writes a review and sends a complete PR form to the PAS.

Main PA data objects:

1. **PA parameters** – a set of formal parameters for a current PR session.
2. **PR form** – a table that specifies fields in a review for some type of an artifact. Ideally, a PR form contains clear fields description and supports the basic validation of fields' values. Also any worthy PR form should contain a text field called *free comment* for immediate informal feedback.
3. **Submission** – a complete artifact, submitted by a Submitter into the PAS as a result of the assignment in the current session.
4. **Review** – a complete PR form, received from a Reviewer.

5. **Feedback** – an additional information from a Student, different from a Submission and a Review.

Some remarks about concepts and terms:

1. In our case of PA, we suppose that the set of Reviewers is equal or less of the set of Submitters in terms of sets theory.
2. We use separate verbs *Submit* and *Send* to distinguish actions on the stage of collecting artifacts (submissions) and in different situations.

3.2. Basic motivation of original mail-based peer assessment system construction

The gaps exposed while reviewing the CSE domain allow us to formulate a number of desired requirements:

1. Support of complex artifacts as submission content (like source code, design documents, complete software projects, etc.).
2. Support of complex review form which should be easily updated and changed.
3. Students should not study and use additional software (we suppose that now everybody can use e-mail).
4. A teacher can tightly integrate PAS with common university IT infrastructure.

3.3. Use cases

The list of main PASCA use cases of the system from the Teacher point of view (by the stages of PR process):

1. Preparation stage:
   - 1.1. Prepare assignment task (description file).
   - 1.2. Prepare PR Form with validation rules and assessment criteria (including rubrics).
   - 1.3. Prepare source list of PR participants (initial authors) and their e-mail addresses.
   - 1.4. Fill in PR parameters and schedule.
   - 1.5. Anonymize participants and build randomization scheme of PR with initial mapping between Submitters and Reviewers.
2. Submissions collecting stage:
   - 2.1. Broadcast task description file to the students.
   - 2.2. Gather Submissions from the students.
   - 2.3. [Optional] Remap reviewers based on missed submissions.
3. Reviews collecting stage:
   - 3.1. Send PR Forms to the students.
   - 3.2. Gather Reviews from the students.
   - 3.3. Calculate final marks and check status of all participants.
4. PR results analysis stage:
   - 4.1. Send PR results to the students.
   - 4.2. Gather additional feedback from the students.
   - 4.3. Build final PA session report.
5. [Permanently available actions]:
   - 5.1. Check current submissions and reviews status.
   - 5.2. Broadcast information letters and feedback.
   - 5.3. Archive mailbox and PR data.

3.4. Specific requirements

Main functional requirements for the use cases listed above (Sub-section 3.3):

1. Importing lists of students with e-mails from external sources.
2. Automated delivering of a task description file converted into PDF.
3. Supporting various e-mail addresses for one student.
4. Basic anonymization of artifacts.
5. Blinding of the participants to support single- and double-blinded review process.

6. Randomization of the reviewers.
7. Independent preparation of PR forms.
8. Integrated validation of PR form before sending review.
9. Automated assessment procedure by PR forms processing after review process.
10. Generating reports about submissions, reviews and final assessment results.

Main non-functional requirements:

1. Using only standard Microsoft Office components.
2. Support any IMAP mailbox as a "*server side*".

4. Current project state

4.1. Technologies and main metrics

The PASCA was designed to use Microsoft Office 2010-2016 or Office 365. Involved components of Office suite are Excel, Outlook, and Word. Optimal mail server is Microsoft Exchange (2010 or higher). But we also can use any IMAP mailbox. For now, project has been fully tested in:

1. Microsoft Windows 8.1 and 10 operation systems.
2. Microsoft Office 2013, 2016 and 365.
3. Mailboxes in Google (http://mail.google.com) and Yahoo! (http://mail.yahoo.com) free mail services.

The main project metrics are relatively low. Without additional third-party components, there is about 75 KB (2362 lines) of the original source code, because most of low-level tasks are implemented and executed by the components of the Microsoft Office Suite and standard Microsoft Windows components.

4.2. Functionality

PASCA focuses on main scenarios of a PA process from a Teacher's point of view. PASCA architecture ensures high level of PA materials (Authors lists, PR forms, reports) reuse. We have used the Excel as an application host (see Fig. 1), a main data storage, and a report builder.

MS Outlook is used for all mailbox management tasks. It means that a user needs to set up an Outlook account and writes its name in PASCA's settings. This solution may be treated as a drawback but the result is very handy, because of useful additional tools available in the Outlook. We provide a user with the option to change mailbox *account* and mailbox *folder*, used for processing e-mail messages.

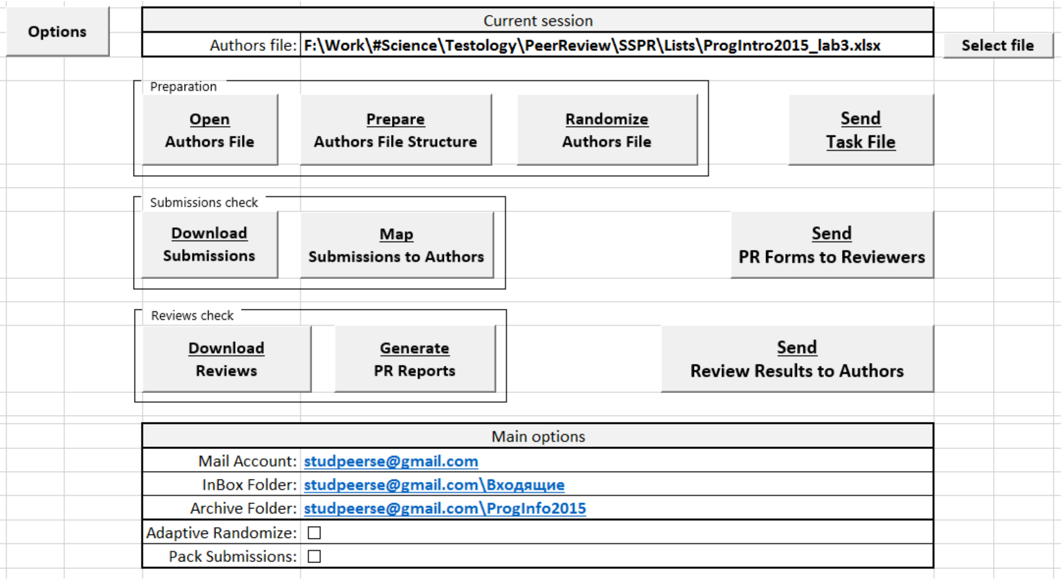
PASCA has two main types of events: facts of an e-mail message *send* and *delivery* with timestamps, generated by an e-mail system. Any student may use *secondary* e-mail address on a submission stage in addition to *primary* e-mail address in an initial Authors list. Primary and secondary e-mail addresses of each participant are checked for compliance on the following stages of PR session. All the auto-generated messages are signed by the *Peer Review Robot*. All the data objects on the moment of the event are represented as attachments in corresponding e-mails.

Each PR session is represented by one excel workbook. The first worksheet of the workbook is an Authors list, the second – PR session parameters (see Fig. 2), the followings – randomization scheme, submissions status, reviews status and various reports (see Fig. 3).

We suppose that a Teacher has already had a source of students' list with e-mail addresses. In our educational practice such a list is available in 99% of study process cases. In a very rare conditions a Teacher should additionally check the correctness of the list.

A separate Excel file embodies a PR form that supports a multi-field review, complex assessment rules, a validation scheme (see Fig 4). Parameters page in PR session workbook contains link to the used PR form and the number of fields in the PR form.

MS Word is used to prepare a task description file and to convert it into PDF format. Also a Teacher can broadcast additional feedback and comments to all Authors at any time.



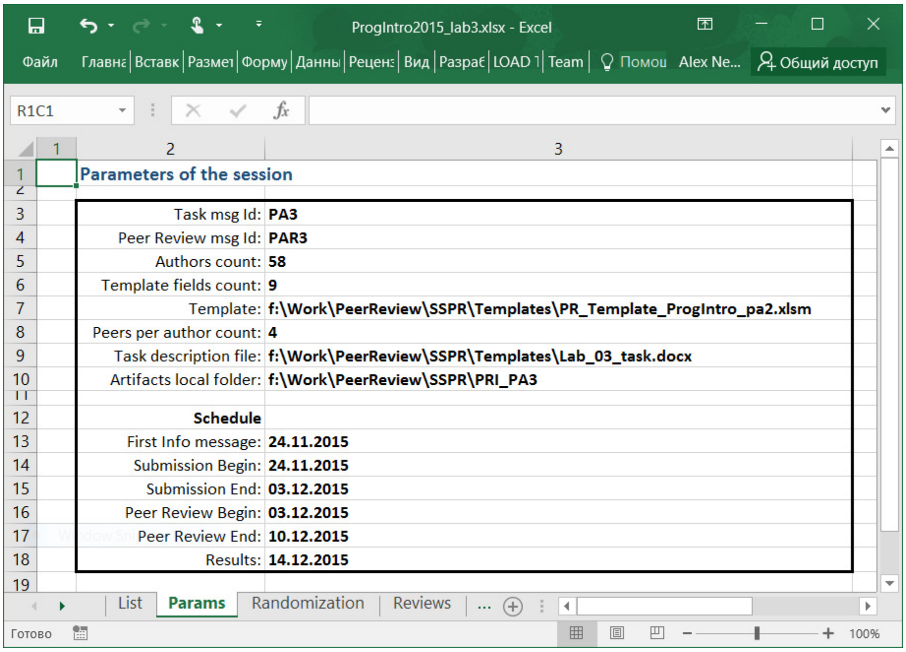
Current session

Authors file: F:\Work\#Science\Testology\PeerReview\SSPR\Lists\ProgIntro2015_lab3.xlsx

Main options

Mail Account:	studpeerse@gmail.com
InBox Folder:	studpeerse@gmail.com\Входящие
Archive Folder:	studpeerse@gmail.com\ProgInfo2015
Adaptive Randomize:	☐
Pack Submissions:	☐

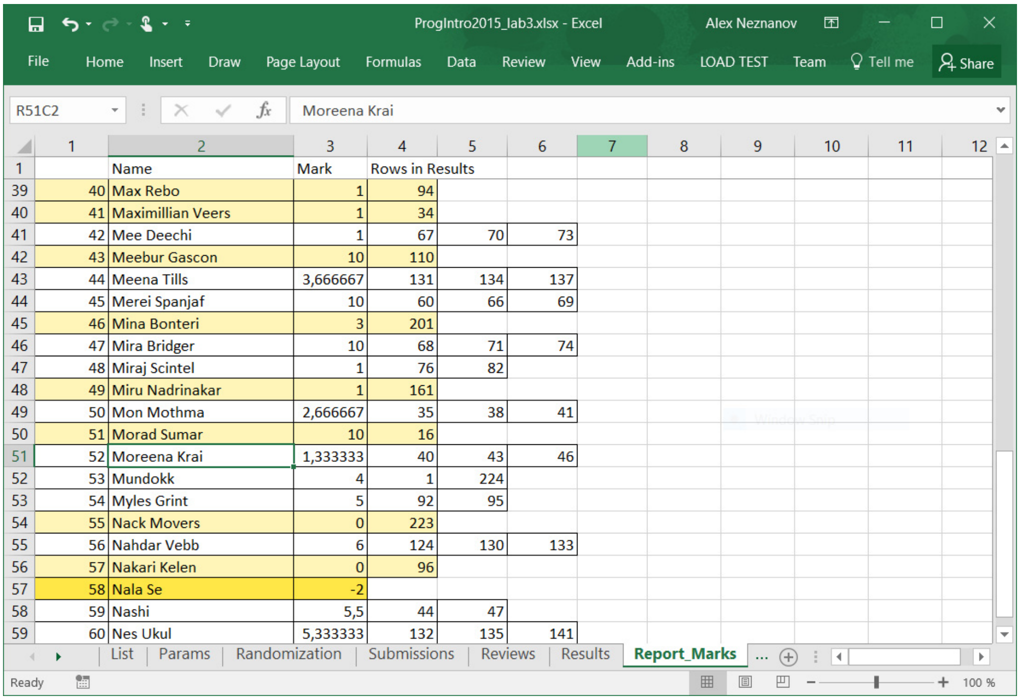
Fig. 1. Screenshot of PASCA main worksheet.



Parameters of the session

Task msg Id:	PA3
Peer Review msg Id:	PAR3
Authors count:	58
Template fields count:	9
Template:	f:\Work\PeerReview\SSPR\Templates\PR_Template_ProgIntro_pa2.xlsm
Peers per author count:	4
Task description file:	f:\Work\PeerReview\SSPR\Templates\Lab_03_task.docx
Artifacts local folder:	f:\Work\PeerReview\SSPR\PRI_PA3
Schedule	
First Info message:	24.11.2015
Submission Begin:	24.11.2015
Submission End:	03.12.2015
Peer Review Begin:	03.12.2015
Peer Review End:	10.12.2015
Results:	14.12.2015

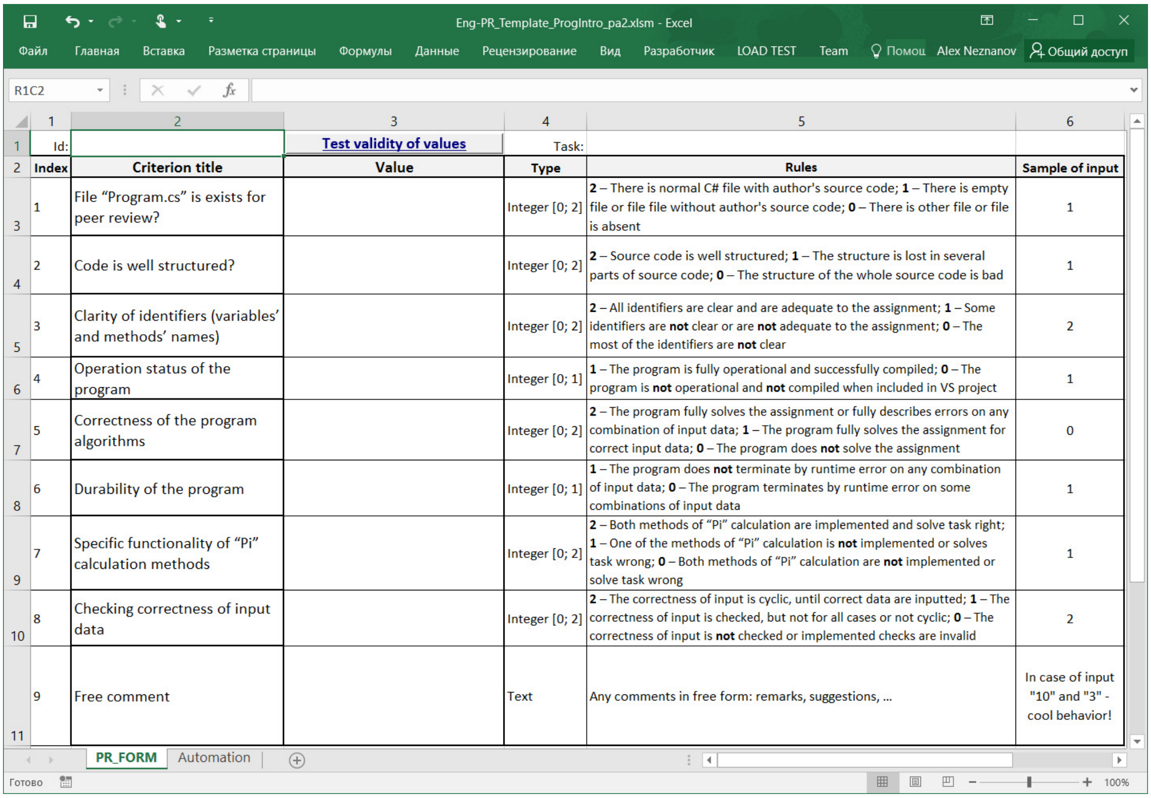
Fig. 2. Screenshot of PASCA PR session parameters page.



R51C2 | Moreena Krai

	1	2	3	4	5	6	7	8	9	10	11	12
1		Name	Mark	Rows in Results								
39	40	Max Rebo	1	94								
40	41	Maximillian Veers	1	34								
41	42	Mee Deechi	1	67	70	73						
42	43	Meebur Gascon	10	110								
43	44	Meena Tills	3,666667	131	134	137						
44	45	Merei Spanjaf	10	60	66	69						
45	46	Mina Bonteri	3	201								
46	47	Mira Bridger	10	68	71	74						
47	48	Miraj Scintel	1	76	82							
48	49	Miru Nadrinakar	1	161								
49	50	Mon Mothma	2,666667	35	38	41						
50	51	Morad Sumar	10	16								
51	52	Moreena Krai	1,333333	40	43	46						
52	53	Mundokk	4	1	224							
53	54	Myles Grint	5	92	95							
54	55	Nack Movers	0	223								
55	56	Nahdar Vebb	6	124	130	133						
56	57	Nakari Kelen	0	96								
57	58	Nala Se	-2									
58	59	Nashi	5,5	44	47							
59	60	Nes Ukul	5,333333	132	135	141						

Fig. 3. Screenshot of PASCA PR session report with review counts and automatically generated marks (anonymized for publcation).



R1C2

	1	2	3	4	5	6
1	Id:		Test validity of values	Task:		
2	Index	Criterion title	Value	Type	Rules	Sample of input
3	1	File “Program.cs” is exists for peer review?		Integer [0; 2]	**2** – There is normal C# file with author's source code; **1** – There is empty file or file file without author's source code; **0** – There is other file or file is absent	1
4	2	Code is well structured?		Integer [0; 2]	**2** – Source code is well structured; **1** – The structure is lost in several parts of source code; **0** – The structure of the whole source code is bad	1
5	3	Clarity of identifiers (variables' and methods' names)		Integer [0; 2]	**2** – All identifiers are clear and are adequate to the assignment; **1** – Some identifiers are **not** clear or are **not** adequate to the assignment; **0** – The most of the identifiers are **not** clear	2
6	4	Operation status of the program		Integer [0; 1]	**1** – The program is fully operational and successfully compiled; **0** – The program is **not** operational and **not** compiled when included in VS project	1
7	5	Correctness of the program algorithms		Integer [0; 2]	**2** – The program fully solves the assignment or fully describes errors on any combination of input data; **1** – The program fully solves the assignment for correct input data; **0** – The program does **not** solve the assignment	0
8	6	Durability of the program		Integer [0; 1]	**1** – The program does **not** terminate by runtime error on any combination of input data; **0** – The program terminates by runtime error on some combinations of input data	1
9	7	Specific functionality of “Pi” calculation methods		Integer [0; 2]	**2** – Both methods of “Pi” calculation are implemented and solve task right; **1** – One of the methods of “Pi” calculation is **not** implemented or solves task wrong; **0** – Both methods of “Pi” calculation are **not** implemented or solve task wrong	1
10	8	Checking correctness of input data		Integer [0; 2]	**2** – The correctness of input is cyclic, until correct data are inputted; **1** – The correctness of input is checked, but not for all cases or not cyclic; **0** – The correctness of input is **not** checked or implemented checks are invalid	2
11	9	Free comment		Text	Any comments in free form: remarks, suggestions, ...	In case of input "10" and "3" - cool behavior!

Fig 4. Screenshot of PASCA PR form template sample.

4.3. Anonymization and randomization

Constantly, each Submitter and Reviewer has a random unique 6-digit ID from the range [100000-999999] and each PR form has a random unique 7-digit ID from the range [1000000-9999999]. Thus, the ranges of Participants IDs and PR forms IDs do not overlap. After building a PR session workbook from an initial Authors list all subsequent actions use those IDs.

Several randomization schemes and algorithms have been tried. Basic non-adaptive algorithm is used for standard randomization of all authors. It is based on classical Richard Durstenfeld permutation algorithm[32] with checking for non-equal submissions assigned for one Reviewer. More interesting adaptive algorithm is used for uniform workload of the Reviewers, which takes into account missing submissions. We continue experiments on different randomization schemes.

5. Adoption

PASCA was first adopted in the Fall semester of 2015-2016 academic year while introductory programming course (C# language was used) at Faculty of Computer Science of National Research University Higher School of Economics. During the instruction, 58 first year software engineering students were engaged into three PR sessions. The sessions were separated from each other with one or two weeks. Each session was consisted of two parts: individual programming assignment and review. The duration of every programming assignment was nearly a week, the review parts lasted from 4 to 5 days depending on programming assignment complexity. PR sessions covered the following topics:

- Serial algorithm.
- C# code documentation and decoration (using commentaries).
- C# operators (conditions, loops).
- C# arrays (regular arrays, jagged arrays).
- Static methods in C# (returning a value, passing parameters by value and by reference).

Despite the simplicity of PASCA, the first session was prepared as a pilot. The goal of this session was to meet students with the system and to adopt them to a new type of assessment. So, the first session was mostly focused on PA process and contained rather easy programming assignment.

5.1. Students' reaction

After three PA sessions 48 students participated in an anonymous post-course survey. The survey consisted of quantitative and qualitative parts. The first one contained general choice questions about the introductory programming course. The list of questions significant to present study is listed below:

- Evaluate using 10-point scale (from 1 to 10, where 1 – "absolutely useless", 10 – "very useful"), how did peer assessment assignments help you to learn new material?
- Which of peer assessments assignments was the most useful to you personally (choice one answer)?
    - (Lab1) Serial algorithm. Data types. Operators. C# code documentation and decoration (using commentaries)
    - (Lab2) Conditions. Loops. Static methods in C#.
    - (Lab3) C# arrays. Passing arrays into methods.

The second one was targeted on gaining feedback on PA support properties, it contained the following open-ended questions:

- Which email client(s) did you use to communicate with the Peer-Review Robot?
- What troubles did you have
    - while receiving assignments from the Peer-Review Robot?
    - while submitting your work to the Peer-Review Robot?
    - during filling in a grading form?
    - during receiving and interpreting the result of your submission?

- What mistakes did you make during
  - reading and interpreting assignments from the Peer-Review Robot?
  - finalizing and submitting your work to the Peer-Review Robot?
  - filling in a grading form?
  - receiving and interpreting the result of your submission?
- Have you got any suggestions about improvements in interaction between Peer-Review Robot and email user?

The results of processing and analysis data gained through quantitative questions, listed above generalized at Fig. 5 and Fig. 6. It is not a surprise that more than 80% of students found Lab2 and Lab3 more valuable for learning than a pilot Lab1, which was marked as the most useful only by 16% of students.

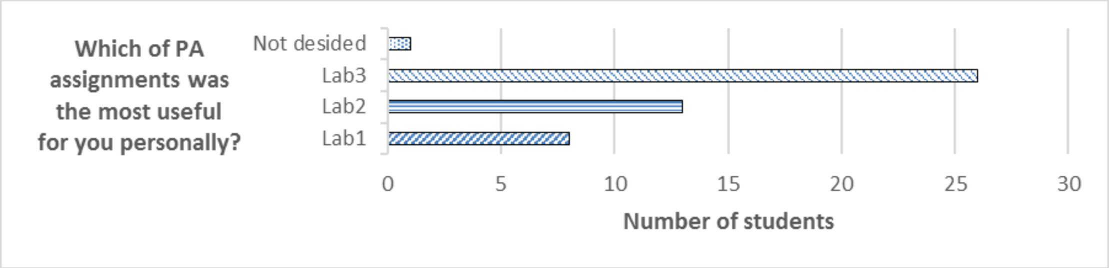


Fig. 5. Peer assessment assignments usefulness rating.

Fig. 6 shows the distribution of scores on PA's utility (*Mean* = 5.67; *SD* = 2.47). Summarizing this quantitative feedback, we may consider that 25% of students rated PA as useless or as having a weak impact on learning. More than a half of students found PA satisfactory and good enough to manage with new material. As expected the minority of students (17%) scored PA with the highest marks.

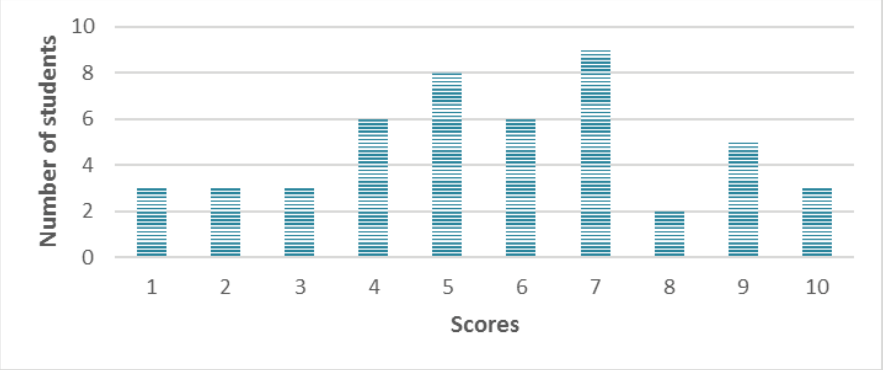


Fig. 6. Peer assessment assignments scores' distribution.

Moreover, the qualitative part of the survey demonstrates the directions of the future work in PASCA's usability improvement.

"Add a possibility of changing the e-mail address" (Student 1)

"It would be better if it is impossible to receive an empty letter" (Student 2)

"Sending all the reviews *in one letter would be more comfortable" (Student 3)*

We have also received interesting suggestions on a PA procedure and its scoring, for example:

"If somebody did not evaluate submissions, I suggest adding him to a blacklist and subtract 10 points from his score" (Student 4)

Surprisingly, it has been turned out that not all of the CS students check corporate e-mail systematically.

"I have forgotten about the third lab, because I use e-mail rare. May be alerts sending to the group's e-mail should be added" (Student 5)

5.2. Problems

Adoption has helped to detect interesting problems:

1. Some students had problems with mailbox management. For example, they used invalid PR session ID while replying to the message with PR form.
2. Students need an additional channel for reminders.
3. PASCA needs special automatization of the deadlines checking. We can use Exchange notification system but have problems with standard IMAP mailboxes.

6. Future development

Because the following academic year is planned this Spring, several critical troubles should be solved as soon as possible. So, the nearest future we are going to focus on following problems:

1. Improved notification system. For now, if PASCA uses external IMAP mailbox than there are no automatic notifications at all due to the unavailability of callback functions on the server side.
2. Blinding in small groups. This problem is linked to additional anonymization of artifacts.
3. Quality of different adaptive randomization schemes.
4. Automatic preliminary checking of artifacts. For example, we can try to compile source code of simple programs as artifacts in SE courses: compile errors will exclude program from reviewing.
5. Review reliability analysis and cheating prevention.

Problems we are not going to focus on:

1. Scaling process on more than 500 students in one PR session. There are very different approaches to PAS as part of high-scale MOOC platforms[4].
2. Calibration of reviewers.

Our investigation argues that there is no ready suitable PAS to meet requirements for PA with complex artifacts and PR forms. But we got the bunch of good ideas and project solutions that leads us to the PASCA implementation.

In the near future we are going to fix some UI glitches and construct web-service for centralized scheduling and management of reminders.

7. Conclusion

This paper presents software requirements for a mail-based peer assessment system and introduces the Mail-based Randomized Double-Blinded Peer-assessment System for Complex Artifacts (PASCA), which is developed according to these requirements. Using this system, participants of educational process (teachers and learners) are not required to master new business process or to use or set up any additional software except of standard email system. Furthermore, PASCA supports blinding and randomization during a peer assessment session.

Currently, PASCA provides all the functionality, which is declared in this work. Moreover, the system was successfully adopted in the introductory programming course for the first-year software engineering bachelor students. The feedback, which was gained during the adoption, dictated the directions of the work for the nearest future. PASCA will be improved by adding notifications, validators, and adaptive randomization algorithms.

Finally, PASCA is claimed to be an open-source project and now is freely available at the repository (http://bitbucket.org/SiberianShaman/pasca).

Acknowledgments

The article was prepared within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE) and supported within the framework of a subsidy by the Russian Academic Excellence Project "5-100".

References

1. Bonwell C, Eison J. *Active Learning: Creating Excitement in the Classroom*. Washington, D.C., 1991.
2. Clarke S. *Active Learning through Formative Assessment*. London: Hodder Education, 2008.
3. Anson R, Goodman J. A Peer Assessment System to Improve Student Team Experiences. *Journal of Education for Business*, 89(1), 2014. pp. 27-34.
4. Kulrarni CWK, Le H, Chia D, Papadopoulos K, Cheng J, Koller D, Klemmer S. Peer and self-assessment in massive online classes. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 20(6), 2013. pp. 1-31.
5. Davies P. Peer assessment: judging the quality of students' work by comments rather than marks. *Innovations in Education and Teaching International*, 43, 2006. pp. 69-82.
6. Hamalainen H, Hyyrynen V, Ikonen J, Porras J. Applying peer-review for programming assignments. *International Journal on Information Technologies & Security*, 1, 2011. pp. 3-17.
7. Falchikov N, Goldfinch J. Student Peer Assessment in Higher Education: A Meta-Analysis Comparing Peer and Teacher Marks. *Review of Educational Research*, 70(3), 2000. pp. 287-322.
8. Van Zundert M, Sluijsmans D, Van Merrienboer J. Effective Peer Assessment Processes: Research Findings and Future Directions. *Learning and Instructions*, 20, 2010. pp. 270-279.
9. Dochy F, Segers M, Sluijsmans D. The Use of Self-, Peer and Co-assessment in Higher Education: a review. *Studies in Higher Education*, 24(3), 1999. pp. 331-350.
10. Nulty D. Peer and Self-assessment in the First Year of University. Assessment & Evaluation in Higher Education, 36(5), 2011. pp. 493-507.
11. Isomottonen V, Tirronen V. "Teaching programming by emphasizing self-direction: How did students react to the active role required of them? *ACM Tras. Comput. Educ.*, 13(2), 2013. pp. 6-21.
12. Carrol J, Jiang H, Borge M. Distributed collaborative homework activities in a problem-based usability engineering course. *Educ. Inf. Technol.*, 20, 2015. pp. 589-617.
13. Clark N, Davies P, Skeers R. Self and peer assessment in software engineering project. Proceedings of the 7th Australasian conference on Computing education, 2005.
14. Wu YT, Anderson O. Technology-enhanced stem (science, technology, engineering, and mathematics) education. *J. Comput. Educ.*, 2(3), 2015. pp. 245-249.
15. Computer Science Curricula 2013. Curriculum Guidelines for Undergraduate Degree Programs in Computer Science, 2013.
16. IEEE Computer Society, Guide to the Software Engineering Body of Knowledge, Version 3.0, P. Bourque and R. Fairley, editors, 2014.
17. Blau H, Moss JEB. FrenchPress Gives Students Automated Feedback on Java Program Flaws. Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education, 2015. pp. 15-20.
18. Anderson P, Nash T, McCauley R. Facilitating Programming Success in Data Science Courses through Gamified Scaffolding and Learn2Mine. Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education, 2015. pp. 99-104.
19. Lin KY, Yu KC, Hsiao HS, Chu YH, Chang YS, Chien YH. Design of an assessment system for collaborative problem solving in STEM education. *J. Comput. Educ.*, 2(3), 2015. pp. 301-322.
20. Kulkarni C, Socher R, Bernsein M, Klemmer S. Scaling short-answer grading by combining peer assessment with algorithmic scoring. Proceedings of the first ACM conference on Learning @ scale conference, Atlanta, 2014. pp. 99-108.
21. Hamer J, Kell C, Spence F. Peer assessment using aropa. Proceedings of the ninth Australian conference on Computing education, Ballarat, Victoria, 2007. pp. 43-54.
22. Hyyrynen V, Hamalainen H, Ikonen J, Porras J. MyPeerReview: An Online Peer-Reviewing System for Programming Courses. Proceedings of the 10th Koli Calling International Conference on Computing Education Research, Koli National Park, 2010. pp. 94-99.
23. Topping K. Peer Assessment Between Students in Colleges and Universities. *Review of Educational Research*, 68(3), 1998. pp. 249-276.
24. Tillema H. Student Involvement in Assessment of their Learning. In: Wyatt-Smith C, Klenowski V, Coberts P, editors. Designing Assessment for Quality Learning, Springer, 2014. pp. 37-52.
25. Lew M, Awis W, Schmidt H. Accuracy of students' self‐assessment and their beliefs about its utility. *Assessment & Evaluation in Higher Education*, 35(2), 2010. pp. 135-156.
26. Kollar I, Fischer F. Peer assessment as collaborative learning: A cognitive perspective. Learning and Instruction, 20, 2010. pp. 344-348.
27. Hundhausen C, Agrawal A, Agarwal P. Talking about Code: Integrating Pedagogical Code Reviews into Early Computing Courses. *ACM Trans. Comput. Educ.*, 13(3), 2013. pp. 14-28.
28. Yeh S, Liu T, Graf S, Wang Y. Exploring the Development of Web-based Peer Assessment System. Proceedings of EdMedia: World Conference on Educational Media and Technology 2008, Vienna, 2008.
29. Liu E, Lin S, Chiu C, Yuan SM. Student participation in computer science courses via the Network Peer Assessment System. *Advanced Research in Computers and Communications in Education*, 2, 1999. pp. 744-747.
30. Lin, Liu E, Yuan S. Web-based peer assessment: feedback for students with various thinking-styles. *Journal of Computer Assisted Learning*, 17, 2001. pp. 420-432.
31. Hamer J, Purchase H, Denny P, Luxton-Reilly A. Quality of Peer Assessment in CS1. 5th International Workshop on Computing Education Research, Berkeley, 2009. pp. 27-36.
32. Durstenfeld R. Algorithm 235: Random permutation. *Communications of the ACM*, 7(7), 1964. p. 420.