Information & Computer Sciences
University of Hawaii
Honolulu, HI USA 96822

April 22, 2019

Chris Hundhausen
Editor in Chief
ACM Transactions on Computing Education

Dear Dr. Hundhausen:

Thank you very much for your consideration of my paper, "Evaluating athletic software engineering education", TOCE-2018-0134.

I have completed a revision of the paper. This cover letter provides a copy of all 45 reviewer comments (in italics) and how their comments were addressed in the revision. I have numbered each comment below as [R-N], where "R" is the reviewer number (1, 2, or 3) and N is a number (01, 02, 03 etc) indicating the comment number.

Reviewer 1

[1-01] The very first section should be titled "Abstract", and can be shortened.

Fixed. I have rewritten and shortened the abstract. Note that the lack of a title is due to the IEEE LaTeX template (acmsmall).

[1-02] The structure is confusing, the authors put the introduction and literature review together, making the problem statement abrupt. There is not enough evidence or background introduced to deduce the knowledge gap.

Fixed. I have added a Related Work section with additional background material.

[1-03] The rationale for naming the new pedagogy as "ASE" is unclear, I think there should be more background on the traditional problematic pedagogy, and the athletic strategy, making it more reasonable and pervasive to adopt this new pedagogy.

Fixed. As noted in the paper, the rationale for naming it "athletic software engineering" is because the pedagogy was inspired from two sports: CrossFit and outrigger canoe paddling.

The Related Work section now goes into more detail on alternative pedagogies and how they relate to ASE.

[1-04] The paragraphs that describes the survey and the design of ASE should be put together in a separate section called "Method".

Fixed.

[1-05] The authors should provide more information about the survey in the method section. How

was it designed? Is it quantitative or qualitative or both? Has it been pilot studied to validate?

Fixed. See Section 4.1. It is a mixed methods design, and there was no pilot study in this study.

[1-06] Section 3.2, first paragraph: There should be more information on the students' background: Did all of them have experience of learning from traditional pedagogy? If most of the students were new to SE courses and had only experienced 1 section of ASE pedagogy, the data is biased.

Fixed. Section 4.1 now contains more information on student backgrounds. Yes, all of them had at least two semesters of experience learning from the traditional pedagogy used to teach computer programming. I do not believe that the data is biased as a result of students being new to SE courses. The study is gathering evidence regarding the student experience of learning material in a software engineering course; specifically content involving the acquisition of tech stack competency. I do not believe that this material is so different from the material covered in other technology-oriented courses such as programming or database systems as to make the comparison biased.

[1-07] Section 3.2, third paragraph: If this section is talking about the effectiveness of ASE, I think this paragraph is not appropriate to be put here, since this paragraph is not relevant to the effectiveness of ASE.

Fixed. The problematic paragraph is moved to a later section.

[1-08] A major limitation, as also mentioned by the authors, is that they lack the objective data to prove the effectiveness of this pedagogy. They only data they collected is the students' self reported data. This data can only explain how students accept this pedagogy, but cannot prove the effectiveness.

I agree that this is a limitation, but do not believe that any single study can "prove" the effectiveness of a pedagogy. Getting anywhere close to "proof" requires many replications under many situations. My hope is that this study will provide sufficient detail and contains sufficiently provocative initial findings that others will be motivated to replicate ASE in order to discover for themselves whether the approach provides value in their settings, and if not, what factors in their circumstances might have led to different results. This is discussed in the Conclusions and Future Directions section.

[1-09] Section 3.4: the authors mentioned that there might be side effects of ASE that have negative impact on women and underrepresented group, however, the authors didn't provide demographic information about the students, how many female students reported the negative or positive impacts? Is there perception difference among race and gender? Without this data, the conclusion of side effect is not convincing.

Fixed. This was stated as a conjecture without any additional analysis because the enrollment of women was so low that gender-based analysis could not be performed. Even if the enrollment was higher, the anonymous nature of the surveys would prevent meaningful demographic analysis. I still feel it is very important to raise the issue. I've added text to this Section 4.1 to clarify why no demographic analysis was performed.

(1-10] Section 4: While the authors said it it not clear to choose a control pedagogy, the author

did ask the students to compare it with traditional pedagogy in the survey. So why didn't set the control group with traditional pedagogy?

If I understand correctly, the reviewer is asking why I did not teach two sections, one using ASE and one covering approximately the same material but taught in a more traditional manner (which is specified in the survey as "longer assignments and problems, no time limit, no instructor solution videos"].

While that would have yielded a more "controlled" design, I feel a moral obligation to provide the best possible educational experience to my students. When 80-90% of them consistently indicate a preference for a given approach and that they believe they learn more and better using it, it feels uncomfortable for me to deny them their preference for the sake of a study. I believe that the current findings are scientifically valid and interesting and provide a basis for others to build upon with more controlled designs if they so choose. I point this out as a direction for future research in the final section.

Reviewer 2

[2-01] However, though I understand the pressure may encourage the students to be more focuses under certain circumstances, it also brings risks to the programming education per se. As indicated in the paper, the time-constrained tasks are used in the early stage of the course on low-level concepts, which means the students are likely to have little or less experience of programming.

Fixed. I have added demographic details to Section 4.1, which indicates (among other things) that all of the students had at least two semesters of previous programming experience.

[2-02] The early stage is key to establish confidence and basic understanding of programming for some or most undergrad students. In this case, Im not sure if "speed" is necessarily a more importance factor than "deliberation" in this stage of education.

As indicated in Section 4.4, the data shows that 84% of students felt that this approach improved their confidence in programming. So, if "establishing confidence" is a goal, then the students report that ASE is an effective pedagogy toward that end.

[2-03] As claimed by a student and quoted in the paper, "Actually having people rush to finish first was discouraging for me. I feel that CS in general has a very competitive culture." If used for basic CS courses, I'm concerned about the design of the approach might prevent programming education in general from being more inclusive and accessible to a wide range of students.

I agree that this is a risk factor. The paper notes this as a potential issue, and encourages its exploration as direction for future research.

[2-04] Also, though I agree that the idea of "programming under pressure" may somehow help students prepared for future technical job interviews or other programming occasions such as Hackathones events and likewise competitions. However, I would argue that the scenarios are quite different, as job interviews and certain competitions are designed for a selective process, where basic CS courses are meant to deliver knowledge and skills to people with no or less prior programming background. A good understanding or mastery of these fundamental concepts may take time and be essential to their future success in this field. Im not sure if a "speed-first" pass or

fail setting would encourage the students, who are relatively new to this field, to be more focused on "quantity" over "quality".

Fixed. Unfortunately, the initial version of the paper did not clearly convey what the students the experiencing the course realize in practice: programming quickly is only possible with a "good understanding or mastery of the fundamental concepts". It is exactly this "mastery" or "fluency" that is reiterated to them over and over as the goal of the athletic portion of the course, and which is the key to their completing in the assessments within the time constraints.

I have added additional explanation in hopes of avoiding this confusion for subsequent readers.

[2-05] In general, I believe software engineering is more than just teaching students how to code, but also a way that people think and approach problems in various fields.

Fixed. I agree with this comment. That said, Software Engineering is much too broad to expect that all topics and perspectives can be covered in a single course. I have added references in the Related Work section regarding the trade-offs involved in specifying curriculum for a one semester introduction to software engineering.

[2-06] One of the major contributions claimed in this paper is that the presented "athletic approach, can be used to improve the learning of programming among undergrad students by focusing on speed of applying mechanics for low-level concepts and leave more time for more "creative activities. However, the authors would need to better address the possible negative effects associated with the new approach in the fundamental software engineering education. To be more specifically, the authors would need to better explain the validity of the new educational approach and how to address the side effects associated with the approach.

Fixed. As noted in the paper, 88% of students prefer this approach over the pedagogies used for their prior programming classes. Of those, 42% also indicated that it improved their learning, 20% indicated that it provided useful incentives and/or improved their self-discipline, and 15% indicated that it helped them get used to pressure and/or "real world" situations. Out of the remaining 12%, 7% advocated a mixed approach. So, in sum, only 5% of the students experienced a cumulative "negative" effect such that they would prefer not to use this pedagogy.

What this study shows is that a strong majority of students over a series of six semesters find this approach appealing and prefer it despite the demands and pressure that it places on them. While I agree with the reviewer that it would be interesting and useful to try to better understand the "possible negative effects" as part of future research, I believe the current study stands on its own merits as a contribution to the community.

The paper now communicates more clearly that "speed" is a by-product of competency and fluency with the tools. Just as in other athletic endeavors, being "sloppy" or sacrificing quality in use of mechanics does not lead to speed.

Reviewer 3

[3-01] I think the novelty can be articulated more clearly. It is certainly here, but I had to read the description of AthSe several times. For example, I recently surveyed undergraduate SE courses at many large state universities and found several have "bootcamps" for the first 2-4 weeks to

prepare students with using Git and whatever language/framework is required. This seems similar to the "mechanics" portion of AthSE. Additionally, the separation of "mechanics" from "creative" activities seems like a typical approach based on Bloom's Taxonomy. Is it different? Learning the syntax of a language and how to use an API seem like the "mechanics". Applying those to create a project or learning best practices would be the "creative" activities.

My takeaway of what is novel in this, but could be more clear: a well-structured pedagogy based heavily on repeated practice assignments involving only the mechanical actions with specified time requirements, inspired by athletic workouts, to overcome the challenges with teaching students full-stacks.

Fixed. Thank you for reading the description several times! I am sorry it took repeated readings. I have restructured the introduction to make the novelty more clear, and added text to the Summary and Future Directions section to clarify.

[3-02] Lack of related work. The basis of AthSE does not seem to be well situated in the related work. The section describing other pedagogies is ripe for adding references about their description as well as studies that illustrate the pros/cons of them, but this is largely missing. The paper itself even acknowledges that teaching SE has been experimented with for over 40 years, but there are only 15 references in the entire paper. Moreover, is AthSE an applied form of Active Learning? I think it would be helpful to hear how it is similar and how it is different. Since AthSE relies so heavily on repetition, I would have expected to see references to studies on repetition and the spacing effect [daily workouts, not binge workouts!]. This has been studied a considerable amount by educational psychologists. The separation of mechanics from creative application is begging for a comparison to Bloom's Taxonomy.

Fixed. I have improved the related work section.

[3-03] Unclear main result. How were students able to compare this pedagogy to "traditional" software engineering pedagogies? It seems they only took one software engineering course, so it is unclear how students were able to make the comparison in Section 3.2. Are they just speculating? Are they comparing it to non-SE courses? If so, the claim in the paper is extremely misleading: "88% of students preferred it to a more traditional approach to software engineering". They don't know what a more traditional approach to software engineering is, or at least they have never experienced it!

Fixed. The students are not asked to compare athletic software engineering to a traditional approach to software engineering education, which of course they have not experienced. Instead the "traditional approach" is defined in the survey as "longer assignments and problems, no time limit, no instructor solution videos". Thus, students are comparing it to their prior computer science programming courses, of which they have had at least two and potentially several more prior to this course. The paper now makes it clear what is meant by "traditional approach" in the evaluation.

That said, many students have told me over the years that they feel this approach could be useful in other computer science courses and wished other instructors would adopt it.

[3-04] I really wish the results were presented in tables/figures when they were first referenced, rather than having to flip to an appendix.

I flirted with this organization but felt the Appendix-based approach was more coherent.

[3-05] Possible future work: It would be helpful to have a guide for adopting this in one's SE course! Including what variations may need to be made to suit other instructors, such as scaling it to large classes.

Yes, that's an excellent idea, but I think it's outside the scope of this paper. If there is interest, I would be amenable to creating an online website, such as athletic-software-engineering.org, in which the pedagogy, sample materials, results from its use by other instructors, and other guides would be made publicly available.

[3-06] Suggestion: Split the Introduction into Intro and Background/Related Work. This will give it a more clear organization.

Fixed.

[3-07] There are multiple claims about final project quality, but no evidence. I value the authors' opinion of this, but appropriate language needs to be used.

I searched for all references to final project and project quality, and ensured that in all cases, the paper communicates this as a future direction or that our views are purely anecdotal and require further research to verify.

[3-08] Athletic Software Engineering has been published on before, but there are no citations when it is first introduced in the paper. What is different about this study from prior studies?

Fixed in the Related Work section.

[3-09] Did you study how students space out their WODs? My students do work in binges rather than spreading them out, but I think the WODs would greatly benefit from the spacing effect.

I did not, but this is an interesting idea.

[3-10] I think this could be great for MOOCs.

Agreed. Implementing this at scale is an interesting issue. I have added this as a direction for future research.

[3-11] "not clear what to choose as the "control" pedagogy." Throughout the paper you have made comparisons to a "traditional" pedagogy, so why not compare to that? Either the textbook-oriented approach or a project-based approach would provide nice evidence. You could also empirically sample how universities teach SE and compare to whatever is most popular. This info is often publicly available.

It's an interesting idea, but I am not sure how to obtain an "apples-to-apples" comparison with the data I collected. I believe the current paper and its results are cohesive and can stand alone as a useful publication. Your suggestion could serve as the basis for a different paper.

[3-12] Section 3.5, the improvements to AthSE, does not really have much content. Is there really nothing of substance that could be improved? This feels tacked on.

Fixed.

[3-13] The paper mentions the pedagogy evolved over time. I think it would be interesting to hear more on what worked and what didn't to cause the changes.

Fixed. The changes were mostly content-related: finding assignments that fit the methodology better.

[3-14] The authors acknowledge that the textbook approach to teaching SE is known to be weak and that project-based is more modern, but repeatedly compare to an example of teaching black-box/whitebox concepts. This seems like a straw man. I'd much rather see comparisons being made to the modern project-oriented teaching method.

Fixed. Related work section now contains more details.

[3-15] "they will have biologically habituated to this form of stress". Citation needed.

Fixed.

[3-16] Table 1: I think this over-exaggerates a "very basic tech stack for web application development." I agree with the overall point, but this is more in line with commercial software products that I have worked on. If you have actual evidence on this, it would make your point stronger. I recently saw an ICSE paper on projects' number of dependencies over time, but can not find it right now. Alternatively, a sampling of popular GitHub projects could provide evidence for this.

Fixed. I've added a reference to stackshare.io, which provides details on modern tech stacks.

[3-17] Superfluous adjectives/adverbs: "very" is used a lot. What is the difference between a "very basic" web app and just a basic web app? Other words that could be dropped since their meaning is vague and was not actually measured: "powerful", "precisely", "quickly".

Fixed.

[3-18] Are students able to move beyond the memorization/mechanics phase and apply what they learn about the tools in new ways?

Yes, that happens in the second part of the course when they build a new application from scratch.

[3-19] Suggestion: drop the CrossFit terminology for more general exercise terms. I don't think there is anything specific to CrossFit in your approach (except maybe the timing?), and it may make it more relatable to students who exercise but don't do CrossFit.

This is a very reasonable idea for future development, but I feel compelled to accurately report on the technique in the paper including the actual names used.

[3-20] Page 9, line 5. You have WOD data on 91%. What happened to the other 9%?

Fixed. The remaining cards were lost or illegible.

[3-21] Page 5, "Section 3 describes the survey results and their limitations." Limitations is Section 4.

Fixed.

[3-22] Were the WODs graded? I worry about anything that doesn't have a grade attached since my students often won't do it. But perhaps the final WOD is enough as an incentive.

Yes, the practice WODs are turned in and graded.

[3-23] Was there a dichotomy where the already strong students do well with little to no practice and the weak students struggle regardless of practicing?

I don't know; the collected data does not allow identification of "already strong students".

[3-24] Were there gender differences in the results?

I don't know. The number of women in each section was always so low [1 to 2] that it's impossible to factor out individual variation.

[3-25] Why does AthSE feel competitive to students, since the goal completion time is the same for everyone, not relative, and there isn't a scoreboard?

I don't know. Two thirds of the students reported that it produced competitive feelings, while one third did not.

[3-26] Definition of tech stack and full stack. I have never heard of developers including tools like code editors or linters into the definition of a tech stack. I always considered full stack, front end, and back end to only include the tools necessary to run the system [not work on it]. My guess is this is a fuzzy term without a de facto definition. If you have a reference for this then that would clarify it.

Fixed. stackshare.io is a public repository of tech stacks.

[3-27] Page 10, line 9. "course had both Athletic and Project-based concepts". Isn't the project portion the second half of AthSE?

Fixed.

[3-28] Page 8, line 44. "soon after the Athletic Software Engineering portion of the course concluded". Do you mean when the "mechanics" portion concluded?

Fixed.

[3-29] Page 13, line 39. "pessimistic". Should this be "optimistic"?

Fixed.

Sincerely,

com.apple.idms.appleid.prd.55696458584863755 54d663831536b505a597 34a2b673d3d

Digitally signed by com.apple.idms.appleid.prd.5569645858486375 554d663831536b505a59734a2b673d3d
DN: cn=com.apple.idms.appleid.prd.5569645858486 375554d663831536b505a59734a2b673d3d
Date: 2019.04.22 09:18:19 -10'00'

Philip Johnson