Raising MSR Researchers: An Experience Report on Teaching a Graduate Seminar Course in Mining Software Repositories (MSR)

Ahmed E. Hassan
Software Analysis and Intelligence Lab (SAIL)
School of Computing, Queen's University, Canada
ahmed@cs.queensu.ca

ABSTRACT

This experience report discusses my views on raising MSR researchers through a graduate-level seminar course. A key goal of this report is to kick start a discussion on this topic within our growing community. A discussion for which there is rarely a suitable venue. Yet, it is an essential discussion to have as a community grows, especially given the rapid growth of the MSR community over the past decade.

1. INTRODUCTION

Much of the literature on computer science education focuses primality on undergraduate education. Over the years, the SIGCSE conference series has become a key venue for sharing and discussing teaching philosophies about undergraduate education. SIGCSE led to the birth of many valuable initiatives for improving computer science education at the undergraduate level. The Nifty Assignments repository [8] is one example of such valuable efforts, where educators can share, reuse and extend examples of interesting assignments and projects for undergraduates.

Unfortunately, there exists no specialized venue for disseminating, and discussing experiences and best practices in graduate education for computer science in general and software engineering in particular – with very few publications on this important topic (e.g., [2]). All too often, such discussions and decisions are often relegated to a general research methods graduate course that is taken by all graduate students across all sub-disciplines of computing in their first term of graduate studies.

The design and content of such a course is often too wide in scope. Many of the needed skills to succeed as a theory researcher vary considerably from the needed skills to succeed as a systems or software engineering researcher. Nevertheless, due to the small size of the graduate cohorts and limited resources, it is often the case that students from various sub-disciplines of computing are co-taught the same research methods course.

MSR'16, May 14-15 2016, Austin, TX, USA

ISBN 978-1-4503-4186-8/16/05…$15.00

DOI: http://dx.doi.org/10.1145/2901739.2901780

Recently, at Queen's University we split off our research methods course into two courses: a general research methods course and a systems-oriented research methods course, thanks to strong financial support from NSERC (the Natural Sciences and Engineering Research Council of Canada). This splitting has permitted us to customize the course per sub-disciplines of computing.

While such discipline-focused research methods are a step in the right direction, we should take a more holistic view about graduate education even within our area-focused graduate courses (i.e., one's own grad course). Our primary goal should not be solely focused on teaching students about the latest and greatest advances in our area (e.g., Mining Software Repositories, Testing, or Requirement Engineering). Instead we should keep an eye on raising the next generation of our community. As for that is the generation that will shape and lead our community for years to come.

This paper recounts my experiences and lessons learned from teaching a term-long graduate course on Mining Software Repositories since 2007. The primarily goal of this paper is to kick-start a community-wide discussion about the important topic of raising the next generation of researchers in our community. This paper is not complete by nature nor are the proposed methods empirically validated through large scale studies. Nevertheless, I believe that the paper will be valuable to others throughout our community. Valuable for faculty members who plan to teach such type of grad course. Valuable for students who wish to understand the rationale behind the design of a graduate course.

Paper organization. Section 2 gives a brief overview of my MSR grad course. Section 3 highlights the risks associated with solely covering the latest MSR research. Section 4 discusses the importance of replication as a pedagogical and mentoring instrument. Section 5 discusses critiquing as an essential and powerful tool for top researchers. Section 6 gives a brief summary of the paper and the discussed topics.

2. MY MSR GRAD COURSE

My primary goal of teaching my MSR course is to raise world-class MSR researchers – this goal applies even to students who are not doing their thesis research in MSR. I believe that many of the MSR teachings (e.g., empirically-driven research, extensive use of analytics techniques, and an eye on the practical impact of research) are applicable across many disciplines of computing. Furthermore, the discussed presentation and writing advice applies across many research disciplines.

Each of the previous eight offerings of the course had on average ten graduate students. The students appreciate the course – their average rating of the course is 4.8 out of 5. However, the course rates as one of the heaviest workloads in the department. Nevertheless, the course project and assignments has led to the publication of 27 papers over the past few years: 2007(2), 2008(2), 2009(6), 2010(6), 2011(2), 2012(4), 2013(1), 2014(2), and 2015(2) – the bracketed numbers indicate the number of published papers in that year not the number of published papers by students who took the course on that year since a journal publication might take a few years to get published. Some of the papers (e.g., [1]) brought many MSR thinkings to other disciplines of computing and are well-cited today. Moreover, ten of the previous students that took or audited this course are now faculty members who are today actively doing research in MSR or Software Engineering, in general.

Course Design. The design of the course is influenced by pedagogical thinkings on apprenticeship and situated learning [5]. Hence, the course deliberately avoids being lecture based. Lecturing is an ineffective method for graduate level teaching since lecturing often leads to a superfluous disengaged understanding of the covered material [9]. Nevertheless, lecturing is used in the first two weeks to provide students with essential background material; then the rest of the classes are conducted in a seminar-style setting as follows (classes are three-hours long and are held once a week):

- In the first week students are given an overview of research achievements in MSR. This overview highlights important general challenges across the field.
- The second week students are given a high level introduction of various techniques (e.g., data mining techniques such as Random Forest, validation approaches such as ten fold cross-validation, code analysis techniques such as island parsing, and mining heuristics such as the SZZ approach), and tools (e.g., R) that are commonly used throughout MSR publications.
- Weeks three to 10 are dedicated to paper presentations and discussions.
- Weeks 11 and 12 are for course project presentations (varies based on size of class).

Two to three research papers are covered each class. Each student is assigned to present at least one paper per term. A student is expected to present and guide the discussions around that paper. All students are expected to have read the papers that are covered in a particular week. Each student (who is not presenting on that particular week) is expected to submit a one-page critique of one of the papers that are covered during that week. The critique should give a brief summary of the paper, while highlighting three strong points and three points for improvement. In earlier offerings, students could pick any paper to critique. In recent offerings, students are assigned a paper to ensure that there is enough critiques for each covered paper. The covered papers are used by the instructor (i.e., me) as a backdrop to communicate various MSR teachings, research design, publication strategies, and writing style.

The marking scheme is as follows: 10% for class participation, 20% for paper presentation, 10% for weekly critique, 20% for a hands-on assignment, and 40% for a project. The assignment is done in groups of 3-4 students. The project is done individually. Both the assignment and project require the submission of a 10-page double column report.

This paper primarily focuses on lessons learned about the overall course design. I will not delve into detailed lessons about designing the assignment and project due to space limitations.

3. FOCUSING SOLELY ON THE LATEST RESEARCH IS NOT SUFFICIENT

I discuss here a few key observations and lessons learned about the main designs aspects of a grad course (e.g., the selection of discussed papers and covered topics)

Greater emphasis should be put on the variety of topics and methodologies. The impact of the covered technical topics is relatively minor. All too often when designing a new graduate course, the emphasis is on the technical content in the course, e.g., which technical topics should one cover in the course. For example, a typical graduate course in MSR should cover topics such as mining data from source control or issue tracking systems. Furthermore, the course should also cover topics around code quality. Interestingly the specific topics are always the first thing that one considers when designing a graduate course.

These chosen topics are often based on the instructor's own sub-interest and prior MSR experiences. One often spends a considerable amount of effort in selecting these topics. However, over the years I found that the impact of these choices are minor relative to the other decisions that one must take when designing a graduate course with a long lasting impact on the students. For instance, the specific topics are always evolving as our community evolves with new repositories being uncovered, and new research methodologies gaining wider adoption across our community. For example, developer surveys and manual analysis of the data along with traditional quantitative analysis have become more common in recent years. Hence, one should not fret too much about the specific topics but should put more emphasis on giving students exposure to a variety of topics and a multitude of methodologies that are used to study these topics. The goal of such variety would be to ignite their minds into thinking beyond the status quo for our community.

Covering classic papers is more valuable than covering recent advances. Once the to-be-covered topics are chosen, one must follow up by picking the specific papers to be presented on these topics. Over the years, as MSR continues to grow the complexity of the used approaches by MSR papers has grown tremendously. As researchers, we are often excited about discussing the latest and greatest research. Hence we often seek to cover in our courses some of the latest and most recent advances and papers, instead of covering older classic papers

However, the methodologies in such recent papers are often too complicated for MSR newcomers. Students are often too overwhelmed by the complexity of the methodologies and all too often miss the key messages of the papers. The same holds for selecting a journal publication versus the earlier conference paper which in most cases communicates the main ideas in a much more succinct and clear manner.

Furthermore, the continuous and rapid evolution of the field leads to one switching papers often with many papers not being repeated across offerings. Surprisingly, the repetition of papers is a desirable thing. Imagine one never repeating an undergraduate lecture from year to year. As papers are repeated, one is able to provide a better ped-

agogical experience to students. For example, one is able to reuse their knowledge about what aspects students find hard to understand, or how students often misinterpret findings. Furthermore, one is able to identify ideal spots (e.g., findings, figures, or statements) that can be used to drive students to reach their own conclusions without one lecturing about a specific topic. Such style of learning through cues that force students to actively engage and apply their learning is desirable especially in grad courses [3].

Discussing the historical backdrop of research is as essential as presenting the technical aspects. With the growing popularity of search engines and the digitization of knowledge, students can easily locate papers on various topics of interest. However, there are many important nuances that students (or even young researchers) are not able to discern easily from simply reading papers. Nuances like researchers which are often always leading the pack, researchers who have a knack for picking interesting problems, or researchers who have special skills (e.g., able to communicate complex problems in a very simple fashion) and expertise (e.g., ones with extensive in-depth machine learning or statistical expertise). Furthermore, it is also important to have discussions about family trees of researchers and research teams. Such discussions help students recognize patterns of paper-writing and research-methodologies across our field. Students can use such patterns to structure for their own future research efforts. Finally, it is essential to give students some background about the area (e.g., work that was thought to be a dead-end (or very hard to publish) but has become popular in recent years). Hence for each presented paper, the presenting student is asked to not only discuss the current paper but to give the class a brief overview of the key authors of the paper (e.g., who are they? what were they doing at paper publishing time?, Where are they today? What work have they been doing since the writing of the paper, especially the work related to the presented paper). This historical backdrop is augmented by the instructor. Nevertheless, students are asked to do it so they can learn how to do such historical digging later on in life.

Such a backdrop helps the course achieve three key goals:

- Students feel that they are part of a much bigger community in which they are excited to grow and which they want to grow, instead of being solely focused on the technical aspect of the work.
- Student acquire a historical perspective of the challenges of younger researchers like themselves (helping them cope with their own challenges (e.g., future rejections) and the overall community's historical missteps (helping them protect our community by avoiding the repetition of community missteps later on).
- Students are able to locate links (through people) for other related work that might not be easy to spot by only focusing on the technical relations.

4. REPLICATIONS ARE AN OUTSTANDING PEDAGOGICAL AND MENTORING INSTRUMENT

In earlier offerings, I put great emphasis and effort in the design of the course assignment. Students were divided in groups and were given a curated data set (by me), and were asked to answer specific (never-investigated-before) research questions (e.g., [4, 6]). Each group had to produce a ten-page double column report. Each group had to give a progress update presentation and a final presentation. Groups were also expected to identify and resolve inconsistencies in findings across groups, and groups were given permission to share data after their update presentation.

The assignment's key goal was to offer students a hands-on experience about the challenges that are associated with producing an MSR paper (e.g., data analysis, paper writing, and differences in results). The assignment was done in a group setting so students can mentor each other. The expectation is that some students are good coders, others are strong analysts, and others are great writers. In many ways, the goal was to teach them how to bike, then to remove the training wheels once they are ready to do their final project (which is done individually). An important side effect was to the ability to eventually publish the final reports as novel research – something which we were fortunately successful in doing on a few occasions (e.g., [4, 6]).

The tackling of a novel research questions often lead to students that are confused about the expected final outcome (since that outcome is a moving target as we learned more about the data and the research questions). For this reason, the course assignment was switched to a different approach: replication of prior MSR papers with a twist. The twist can be either new data, new pre/post processing of the data, or new modeling techniques. The replication is an ideal pedagogical instrument since students have a very concrete and clear idea of what the final outcome of the work will be (i.e., produce the same exact paper with the same research questions and methodology – even same figures – but make sure that the results reflect the requested changes in the replication). Perhaps one of the most important benefits of replications is that the methodology has already been chosen and described in the original paper; often this is one of the most challenging aspects of conducting new research, and it is often one of the areas where students struggle the most. This new approach helps achieve the same goals while guiding students in a more structured fashion.

Unexpectedly, some of these replications have uncovered new findings and observations that have been published in the literature (e.g., [7]). I believe a key reason being that students had a very clear template to follow (i.e., the earlier paper), so they were able to put their focus on interesting intricate details that might have been beyond the original research and to uncover publishable observations.

5. CRITIQUING IS AN ESSENTIAL TOOL FOR TOP RESEARCHERS

At first glance one might think that the key goal of a graduate course is to teach students about the latest research in the area. For instance, earlier offerings of the course were concerned that each student read at least one paper per week (hence the request that a student must submit one paper critique per week). However, over the years I came to the realization that another key aspect is to teach students how to critique work (i.e., to recognize good and bad work).

Such critiquing skills are essential for their future research careers where they have access to a much wider set of research works with no experienced person available to point out major problems with it. Furthermore, the critiquing prepares them for when they receive reviews for their own papers later on in their academic career. For example, the critiquing of papers, that are written by others, gives them

an idea of how others might misinterpret their results, writings, or intentions. In recent years, I made critiquing as an essential skill that a student is expected to learn through my course. This is done through a variety of methods as I explain below.

The use of easychair to manage the critiquing process. In earlier offerings, students would privately submit their critique to me via email or some online course management software. Other students did not have access to the critique. Hence, it was not read by anyone other than me. The goal of the critique was to ensure that each student read at least one paper. Later on I used the critique to raise points that students might be too shy to raise themselves during the in-class discussions.

Recently I moved the whole critiquing process to `easychair`. In `easychair`, each presented paper is uploaded as a separate submission. Students are asked to bid for papers which they would like to review for each week. Then a paper assignment is generated (just like in a conference setting). By moving to `easychair`, each student is able to read the critiques of others. Hence they are able to learn from others (and anecdotally I have noticed quite a bit of improvement in critique quality as the term progresses, compared to the old private critiquing setup). Furthermore, the critiques can be discussed in an online setting well-before the class time (giving students a glimpse of the whole paper reviewing process very early in their careers and freeing time in the class for more deeper discussions). Finally, the student that is allocated to present each paper is responsible for summarizing and responding to the critiques of their peer reviewers.

There are interesting side benefits of using `easychair` to manage this process:

- Students receive notification of updates (e.g., new critiques, update critique, new comments) – so they are up-to-date about the concerns of their peers,
- `easychair` manages the access control extremely well since students cannot read the other critiques for their assigned paper until they submit their own first. Furthermore, students can read critiques for other papers that are being covered that week so they are already getting different points of view from their peers.
- Students are able to decide at the end of the term if any papers should be rejected (i.e., dropped) in future offerings (in a sense the whole course is formulated like a term-long Program Committee meeting).

I have also moved to using `easychair` for assignment and project submissions (the submission are treated as reviews for a phantom assignment or project paper). The use of `easychair` for such submissions again helps students in being aware of the work of their peers in a more active and automated fashion.

A bigger focus on best practices for critiquing and the importance of active critiquing. In earlier offerings, the critiquing process was quite ad hoc and passive in nature. Students were not given much guidance on how to go about producing a good (e.g., critical yet constructive and respectful) critique. Nowadays, the course puts a stronger emphasis on critiquing by spending more time on discussing a) what is the ideal structure of a research paper, and b) what are the common building blocks of an MSR study. It is interesting to note that these two aforementioned points are examples that highlight the importance of having such a discussion in an area-focused course instead of relegating such a discussion to a general research methods course (since these discussions vary considerably across communities). For example, I formulated a set of anti-patterns for critiquing. Some notable examples are listed below:

- *Obvious results:* Students are asked to rethink the validity of such a critique had they not read the paper (since quite often things look quite obvious once an elegant and simple solution has been proposed).
- *N+1 systems:* Students are asked to think deeper about the goal of our field – Are we in search for a unifying theory that unifies knowledge across the whole field (and hence a considerable amount of systems must be studied for each paper)? Or are we more case study focused and are concerned about showing that the work can help at least a few systems (that are possibly impacting the lives of many)?
- *Industry vs. open source:* A classic critique is to complain that the studied projects are open source ones or industrial ones. Students are asked to go beyond the specific projects and to think about the overall impact of the results and on the availability/rarity of the studied data.
- *Not novel (e.g., Replication studies):* This is rarely raised once students are halfway through their assignment.
- *Un-addressable critiques:* Students are asked to combine their critique with a realistic way to address it.

Furthermore, students are reminded throughout the course about being respectful and mindful of others. That at the end of the day there is a person (or set of persons) that are behind the work and who have worked hard to make it happen. Hence students are always reminded to reread their critique and think of how they would have felt about it, if it were for their own work. For this reason, I rarely cover papers by my group in the course so students are more comfortable with their critiquing. Finally, I always aim to have one paper in the term that has major flaws in it so the students can learn to judge each work on its own merits and to exercise such critiquing skills without being influenced by the fact that a piece of work is already published.

6. CONCLUSION

This paper presents a set of guidelines and lessons learned about the design of an MSR graduate course. The paper departs from the traditional advice of designing such courses by highlighting the fact that the technical content is rarely the main concerns of the course. Instead other more important concerns (such as mentorship and critiquing) play a much wider role in producing a graduate course that has a long lasting impact on the students. As an experience report, the paper is subjective in nature and its main goal is to push such a discussion about MSR grad course design to the forefront of our community. Such a discussion is essential for the long term success and health of the rapidly growing MSR community.

7. ACKNOWLEDGEMENTS

I am grateful for the students who participated in my course over the years. Their willingness to openly critique the course and provide constructive feedback has played a big role in shaping the course and many of views in this paper. I would like to thank the `easychair` team for permitting our use case of their paper reviewing system, and the msr reviewers.

References

[1] K. Elgazzar, A. E. Hassan, and P. Martin. Clustering WSDL documents to bootstrap the discovery of web services. In *IEEE International Conference on Web Services, ICWS 2010, Miami, Florida, USA, July 5-10, 2010*, pages 147–154. IEEE Computer Society, 2010.

[2] D. M. German. Experiences teaching a graduate course in open source software engineering. In *In: Proceedings of Open Educational Symposium of the 1st International Conference on Open Source Systems*, pages 326–328, 2005.

[3] D. F. Halpern and M. D. Hakel. Applying the science of learning to the university and beyond: Teaching for long-term retention and transfer. *Change: The Magazine of Higher Learning*, 35(4):36–41, 2003.

[4] W. M. Ibrahim, N. Bettenburg, E. Shihab, B. Adams, and A. E. Hassan. Should I contribute to this discussion? In J. Whitehead and T. Zimmermann, editors, *Proceedings of the 7th International Working Conference on Mining Software Repositories, MSR 2010 (Co-located with ICSE), Cape Town, South Africa, May 2-3, 2010, Proceedings*, pages 181–190. IEEE Computer Society, 2010.

[5] J. Lave and E. Wenger. *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, 1988.

[6] H. Malik, I. Chowdhury, H. Tsou, Z. M. Jiang, and A. E. Hassan. Understanding the rationale for updating a function's comment. In *24th IEEE International Conference on Software Maintenance (ICSM 2008), September 28 - October 4, 2008, Beijing, China*, pages 167–176. IEEE Computer Society, 2008.

[7] T. H. D. Nguyen, B. Adams, and A. E. Hassan. Studying the impact of dependency network measures on software quality. In *26th IEEE International Conference on Software Maintenance (ICSM 2010), September 12-18, 2010, Timisoara, Romania*, pages 1–10. IEEE Computer Society, 2010.

[8] Nifty Assignments repository. http://nifty.stanford.edu/, 2016. [Online; accessed 14-March-2016].

[9] C. K. Sara Steen, Chris Bader. Rethinking the graduate seminar. *Teaching Sociology*, 27(2):167–173, 1999.