Retention Promotion Tenure Committee Report

Evaluations of: Matt Might, Assistant Professor, School of Computing, College of Engineering
Evaluation for: Retention
Completed by: Undergraduate Student Advisory Committee
Recommendation:

Retention			Promotion			Tenure		
Yes	**No**	**Abstain**	**Yes**	**No**	**Abstain**	**Yes**	**No**	**Abstain**
5	**0**	**0**						

PART I. Describe the sources and methods used in gathering the data upon which this evaluation is based. Indicate the number and kinds of responses obtained from other students.

UgSAC relied heavily on the information packet made available by the RPT committee. Additionally, UgSAC talked with past students, attended lectures, reviewed course feedback, and talked to the candidate himself.

The methodology here is designed specifically to discern and characterize the candidate's impact on the undergraduate population. In particular, we consider any quality of a candidate that enriches the state of undergraduate education to be relevant information. This includes (among other things) not only teaching ability, but also things like the ability to identify and encourage talented students, and a demonstrated willingness to reach out to underrepresented groups.

In a lot of ways, the hallmark of Might's impact on undergraduate education is his compilers course, which is notoriously difficult. To gage the specifics of this impact, we began by looking at course evaluations and talking to previous students. Might built this course mostly from scratch, and so careful attention was paid to course materials and people's reactions to them. Additionally, we observed lectures given by him, specifically his Stanford Distinguished Lecture (on video) and the Research Buffet presentation.

Might actually has a strong history of identifying and mentoring undergraduates – one of his students (David Darais) began work this year in the PhD program at Harvard, and another (Phillip Mates) interned at Google over Summer '11. We paid close attention to what they had to say about Might and his mentorship.

PART II. Provide a narrative evaluation of the faculty member's teaching performance. Give particular attention to the faculty member's knowledge of the subject and effectiveness in conveying that knowledge to students.

The consensus is that Might delivered well-organized and entertaining class presentations, and that he demonstrated an exceptional command of the subject as he did so. The response to this effect was strong enough that UgSAC is very confident that we can recommend Might on this basis.

As teaching goes, there seemed to be two common complaints. The first is that Might was absent for a significant number of lectures for the compilers course, almost always either for medical reasons, or because he was presenting research elsewhere. The second is that the compilers course was too hard.

To the first point, while it is clear that Might's absence made the course overall more difficult, it is UgSAC's belief that Might compensated more than adequately, and we would like to address this point so that it is not a matter of contention later. The students themselves note that Might in general was very available and open to meeting with students, and especially was prompt and thorough in his responses via email. Considering the circumstances (and the student responses), UgSAC believes that Might performed exceptionally well under the circumstances.

To the second complaint, UgSAC has investigated and believes that the compilers course represents an invaluable contribution to the undergraduate canon. Particularly, the response from the students who made it through the entire class and finished the project are highly enthusiastic: students leave the class having built a compiler for a large subset of Python, and typically cite the difficulty and comprehensive nature of the class has having had a profoundly positive impact on their abilities as a computer scientist. They also typically cite a feeling of having accomplished something very difficult.

More critically, Might UgSAC believes that Might performed due diligence in communicating the difficulty of the class. If Might's performance in communicating this fact is insufficient to ameliorate concerns about the course being too hard, UgSAC finds it hard to believe that it is possible to do so at all. Additionally, based on feedback, UgSAC believes that Might made an admirable effort to pull the class towards this goal, working with the students to fight through the discouragingly hard problems the students had to deal with.

Finally, and less noticeable, UgSAC is impressed with the clarity of Might's teaching goals, and his resilience in pursuing them. Not every teacher has clear goals to develop targeted skills in their students; often, UgSAC feels that professors teach because they have information. Might's dedication, availability, and willingness to reach out to students, however, has demonstrated that he is not only a talented instructor, but an instructor who wants to watch students become successful, which is an invaluable quality for a professor to have. The epitome of this, UgSAC feels, is his central goal for the undergraduate compilers course, which is to provide in-depth experience developing a compiler that they can list on a resume. The result is that the course is not merely a course at a college, but a part of the student's professional experience.

PART III. State the reasons for the Student Advisory Committee recommendation in this case.

It is easy to forget that great members of the CS community are rarely born. More often, it's a combination of talent and excellent mentorship. Unfortunately excellent mentors are very difficult to

find. UgSAC is enthusiastic about Might's case: he should be retained at all costs because he is an excellent leader at pretty much every level of the undergraduate experience.

In the first place, Might seeks out and encourages the development of talented students. Two examples of this fact are David Darais, who joined Harvard's PhD program in Fall '11, and Phillip Mates, who interned at Google in Summer '11. Both students credit Might's mentorship as a critical point of inflection in their development as computer scientists.

In the second place, Might's compilers course is a demanding and important contribution to the development of the students who take it and make it through. At the end of the course, students will have developed a compiler for a significant subset of Python 3, which is not only rewarding, but also something students can mention on a resume.

Also worth mentioning here is that the undergraduate compilers course is largely the way it is because of Might himself. Where most compilers courses teach on Mini-Java, Might was able to reframe the course while still keeping it manageable. Might wrote most of the course material himself, and pushed students to finish this challenging task.

This could easily be lost in the shuffle, so UgSAC would like to be emphatic here: these points about his compilers course are a very important issue. Might could have easily focused on research and just taught the class as it has always been taught. Instead, he re-imagined and re-implemented it from the ground up, and (as you will read in his teaching statement) he did so specifically to give students valuable skills and a project they can list on their resume. Might does these things because he is interested in developing students into productive members of the community.

This makes the case for us. Our unanimous and uncontested approval of the candidate is funded intellectually by our belief that this quality is *crucial* to the teaching success as a professor. For this reason, and all the reasons previously discussed, we are extremely grateful to have Might as a professor, and our whole-hearted recommendation is to do whatever needs to be done to keep him here at Utah.