CSEE&T 2017 Hall of Fame Nomination: Studio-Based Master of Software Engineering Program at Carnegie Mellon University

I. Summary

Since its founding in 1989, the Studio-based Master of Software Engineering (MSE) Program at Carnegie Mellon University (CMU) has been a trailblazer in advancing the practice of masters-level professional software engineering education. From its inception, the program was designed around a comprehensive development project, the *Studio*. The *Studio* provides students with a team-based, mentored, multi-semester engagement with external clients, allowing them to "learn by doing" as they apply skills and techniques derived from complementary core and elective courses. Going far beyond the common "capstone" experience of many other programs at the time, the *Studio* concept was unique because of the amount of time set aside for practice-based learning (almost a third of the entire curriculum), its approach to active mentoring (engaging seasoned software engineering professionals), and its focus on reflective practices (with explicit attention given to retrospective analysis of positive and negative experiences). This formulation of a software engineering professional degree program has had far-reaching and lasting impacts on software engineering education.[1]

II. History

The CMU MSE Program was created in 1989 as a joint initiative of the Software Engineering Institute and the Computer Science Department. The goal of the program was to produce the future technical leaders in industrial software engineering through a curriculum focused on the practical yet disciplined application of foundational principles, skills, and techniques. At the time of its founding, the field of professional software engineering education was just beginning to emerge, and the MSE attempted to take a leadership role in identifying both curricula and educational practices that would help to transform the practice of software engineering.

From the outset, the founders of the program (Nico Haberman, Angel Jordan, James Tomayko, and Norman Gibbs, among others) recognized that software engineering was not a "spectator sport" [1][2]. Rather, in order to achieve its goals, the program would need to find ways in which students could put into practice the ideas and skills that they were being taught in the classroom. Following Herb Simon's studies showing that learning is directly correlated to "time on task" [3], the program design included three equally important components: (1) core courses to impart foundational principles and relevant skills, (2) elective courses to allow students to pursue an aspect of software engineering in depth, and (3) team-based development centered on a real customer and reflective practice. Each of these components was allotted approximately one-third of the total curriculum. The *Studio* formed the centerpiece of this triad, providing the main forum where students would explore techniques in a practical setting and take primary responsibility for end-to-end software development.

For more than 25 years, and despite many changes in computing and software technology, the rise of new methods of development, and the general maturing of the field of software engineering, the *Studio*-based concept for professional education has remained intact. Indeed, for most students, it continues to be the most important and distinctive influence on their growth as software engineering professionals. The Program has been remarkably successful. Within CMU, it has grown from a small program of 10 students into a suite of professional degree programs with more than 80 students annually. Externally, it has led to the creation of numerous other programs, all attempting to duplicate the basic pedagogical principles of the MSE. These programs have been established in India, Portugal, and South Korea, among others.

III. The Studio-Based MSE

The *Studio*-based MSE has four distinctive aspects. Together they brought to life a new way of envisioning software engineering education.

1) **The use of a project as the centerpiece of a professional software engineering master's degree.** The *Studio* becomes the locus of demonstration for student accomplishment and maturity. Accomplishment is demonstrated through the successful application of best software engineering practices throughout the full software life cycle. Maturity is demonstrated as students take responsibility for making all key decisions, justifying these decisions, and explicitly learning from their successes and failures.
2) **The use of experienced mentors to guide the project experience.** The *Studio* concept requires that seasoned professional software engineers guide students throughout the duration of the project. Rather than directing student teams, the role of a mentor is to ask probing questions, challenge students to justify their decisions,

[1]Nominated by Grace A. Lewis, Principal Researcher, Carnegie Mellon Software Engineering Institute, glewis@sei.cmu.edu

2377-570X/17 $31.00 © 2017 IEEE
DOI 10.1109/CSEET.2017.10

and help students learn from their mistakes. A typical *Studio* normally involves two mentors per team (of 4-6 students). Mentors not only attend regular team meetings, they meet one-on-one with students on a weekly basis. The process of mentoring has been codified, and all mentors must go through an "apprenticeship" period to learn the art of mentoring.

3) **The use of realistic projects.** By *realistic* we mean: (a) solving problems of practical significance, (b) working with an external client, c) involving non-trivial engineering decisions, (d) dedicating a significant period of time — typically the full 16 months of the program, and (e) performing in a team-based setting. Projects are solicited from industry, and are selected by a committee based on factors that include (a) educational value, (b) engagement of the client, and (c) appropriate scope.
4) **The use of reflection as an explicit learning goal.** Strongly influenced by Schön [4], students are expected not only to make decisions, but also to justify them, and to reflect on their success and failure. Decisions can range from management practices, to ways in which to interact with clients, to design and implementation choices and rationale. Reflection is judged at several milestones — primarily at mid- and end-of-semester presentations to the entire faculty. Additionally, each student is required to write a final reflection paper that revisits their role and decisions in the project in an effort to distill take-away lessons from their experience.

IV. Impact

The MSE has had a significant impact on the way in which software engineering professional education is carried out. More broadly, it has influenced the practices of delivering professional master's degrees in other disciplines. Within CMU, the program has successfully grown from a small "boutique" offering to a set of degree tracks that make it one of the largest master's programs in the School of Computer Science. Additionally, it has served as the inspiration for other master's level professional programs, such as the Master in Human-Computer Interaction, which adopted its *Studio*-based approach.[2]. External to CMU, as noted above, the program has been widely duplicated. In particular, three significant collaborations led to the creation of MSE "clones" in other countries: India, Portugal, and South Korea. These programs transitioned the educational materials and pedagogical principles (with minor adaptations) to their local setting, attempting to create as close a duplicate of the program as possible [5]. In addition, the MSE was recently acknowledged through a Software Engineering Distinguished Education Award given by the IEEE Technical Council on Software Engineering (TCSE) in 2017.

[2]https://www.hcii.cmu.edu/academics/mhci/capstone-project

V. Resources

The *Studio*-based MSE has developed a number of resources that have enabled it to be understood and used by others, some of which are available at https://tinyurl.com/y9u7aouy:

- Multiple papers on the MSE Program and the *Studio* have been published in journals and conference proceedings [6][7][8][9][10][11][12][13].
- Training materials for running an effective *Studio* project are codified in the document "Practice Based Studio: An Operational Guide for Faculty and Mentors" that is provided to all new mentors, and updated based on the own program's reflection process.
- The best student final reflection papers are provided to mentors and new students as guidance, such as *People Issues in an MSE Studio Group of Individuals*, *Requirements Elicitation and Analysis for Research Software Development Projects*, *Predicting and Driving Software Construction via Upfront Architectural Design*, *The Product vs. Process Dilemma in the MSE Studio Context*, and *MSE Studio Project: The Viewpoint of a UC Student* [14].

References

[1] N. E. Gibbs, M. A. Ardis, A. N. Habermann, and J. E. Tomayko, "The Carnegie Mellon University Master of Software Engineering Degree Program," in *SEI Conference*, 1990, pp. 152–154.

[2] G. A. Ford and M. A. Ardis, "1991 SEI Report on Graduate Software Engineering Education," Carnegie Mellon University, Software Engineering Institute, Tech. Rep., 1991.

[3] J. R. Anderson, L. M. Reder, H. A. Simon, K. A. Ericsson, and R. Glaser, "Radical Constructivism and Cognitive Psychology," *Brookings Papers on Education Policy*, no. 1, pp. 227–278, 1998.

[4] D. A. Schön, *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, 1984, vol. 5126.

[5] S. Kang, I.-Y. Ko, J. Baik, H. Choi, and D. Lee, "KAIST-CMU MSE program - The Past and the Future," in *Software Engineering Education and Training (CSEE&T), 2010 23rd IEEE Conference on*. IEEE, 2010, pp. 49–56.

[6] D. Garlan, M. Shaw, C. Okasaki, C. Scott, and R. Swonger, "Experience with a Course on Architectures for Software Systems," *Software Engineering Education*, pp. 23–43, 1992.

[7] J. E. Tomayko, "Carnegie Mellon's Software Development Studio: A Five Year Retrospective," in *Software Engineering Education, 1996. Proceedings., Ninth Conference on*. IEEE, 1996, pp. 119–129.

[8] D. Garlan, J. E. Tomayko, and D. Gluch, "Agents of Change: Educating Future Leaders in Software Engineering," *IEEE Computer*, 1997.

[9] M. Shaw, J. Herbsleb, and I. Ozkaya, "Deciding What to Design: Closing a Gap in Software Engineering Education," in *Proceedings of the 27th international Conference on Software Engineering*. ACM, 2005, pp. 607–608.

[10] D. Root, M. Rosso-Llopart, and G. Taran, "Proposal Based Studio Projects: How to Avoid Producing "Cookie Cutter" Software Engineers," in *Software Engineering Education and Training, 2008. CSEET'08. IEEE 21st Conference on*. IEEE, 2008, pp. 145–151.

[11] G. Taran and J. Hartje, "Software Engineering Leadership: A Student Initiative to Promote Change Agents in Industry," in *Software Engineering Education and Training (CSEE&T), 2010 23rd IEEE Conference on*. IEEE, 2010, pp. 173–176.

[12] M. Bass, "Software Engineering Education in the New World: What Needs to Change?" in *Software Engineering Education and Training (CSEET), 2016 IEEE 29th International Conference on*. IEEE, 2016, pp. 213–221.

[13] A. J. Lattanze, "Practice Based Studio," in *Software Engineering Education and Training (CSEET), 2016 IEEE 29th International Conference on*. IEEE, 2016, pp. 1–7.

[14] A. Damasceno, "MSE Studio Project: The Viewpoint of a UC Student," in *Software Engineering Education and Training (CSEE&T), 2011 24th IEEE-CS Conference on*. IEEE, 2011, pp. 502–506.