Preparing Software Engineering Apprentices for Industry

Matthew Barr
University of Glasgow
Matthew.Barr@glasgow.ac.uk

Derek Somerville
University of Glasgow
Derek.Somerville@glasgow.ac.uk

CCS CONCEPTS

• Social and professional topics → Computing education; **Computing education programs**; **Software engineering education**.

KEYWORDS

software engineering, apprenticeships, work-based learning

ACM Reference Format:
Matthew Barr and Derek Somerville. 2020. Preparing Software Engineering Apprentices for Industry. In *Proceedings of the 2020 International Computing Education Research Conference (ICER '20), August 10–12, 2020, Virtual Event, New Zealand.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3372782.3408116

The Graduate Apprenticeship in Software Engineering is a four-year, work-based undergraduate degree programme. Apprentices spend most of their time in the workplace but attend university for two eight-week blocks of intensive tuition in the first year. This 'front-loading' of teaching ensures apprentices are productive as early as possible. The programme was developed in consultation with industry to ensure content met with employer expectations [1]. As part of the consultation, the tasks that employers expected apprentices to be able to undertake during their first year were identified. These included testing, code review, and bug fixing. Sometimes referred to a "legitimate peripheral participation", being able to undertake such tasks allows apprentices to engage in a community of practice around software engineering [2], making meaningful contributions whilst posing minimal risk to the business. An analogy for this participation is that of the ironing that an apprentice tailor might undertake: doing the ironing is not only of value to the business, but teaches the apprentice a useful foundational skill. Thus, the material covered in the initial eight-week block was designed to prepare apprentices to tackle the "ironing" when they returned to the workplace. The courses comprised 'Foundations of Professional Software Engineering', covering topics such as testing and development methodologies, and 'How to Learn a New Language', a language-agnostic introductory programming course.

With the taught portion of the first year complete, and students back in the workplace, both employers and apprentices may reflect on the effectiveness of these courses as preparation for work. In the case of employers, interviews with mentors and line managers are now underway, while apprentices have been asked to keep a daily workplace diary of their activities. Initial analysis of these data indicates that, in many cases, apprentices exceeded expectations in terms of the breadth of the tasks they are able to complete. Many apprentices are engaged in testing and bug fixing, as expected, although one employer indicated that their apprentice was merely implementing bug fixes, as specified by a more experienced developer. Another employer expressed doubt that their apprentice possessed the problem solving ability to fix bugs. These views, however, were in the minority.

Code review has generally not been among apprentices' "ironing" tasks, with employers suggesting that apprentices were not ready to review teammates' code effectively. However, apprentices are contributing code and developing significant new features in addition to the expected bug fixing. The degree of apprentice autonomy has also impressed employers. Only one employer suggested that their apprentice needed "hand holding". The remaining employers have stated that apprentices are "exceeding" expectations or otherwise "impressed". Some employers assigned apprentices to side projects, indicating an initial reluctance to let inexperienced developers work on the main code base. However, most apprentices were working on 'real' applications within two months of their first block of university teaching. In general, this work has involved implementing smaller features, fixing bugs, and writing the tests required to provide coverage of the newly-written code. As one apprentice noted in their workplace diary:

> Making unit tests to test the logic of a web application. Working with stakeholders to solve a support ticket that came in regarding an issue with a user's data not being up-to-date.

Another apprentice's diary illustrates the breath of their workplace activities:

> Worked on the JIRA ticket assigned to me. Created a new component in Angular and introduced a dynamic grid view that an internal team have developed. Also created a service to retrieve historical data, which will then be displayed in this view. Tests were all written in a test driven format for this component and service.

Employers also reported that apprentices were working well within their teams, suggesting that the professional software engineering material covered in the initial teaching block provided adequate preparation. With apprentices making additions and fixes to live code bases, the language-agnostic programming course also appears to have been effective.

REFERENCES

[1] Matthew Barr and Jack Parkinson. 2019. Developing a Work-based Software Engineering Degree in Collaboration with Industry. In *Proceedings of the 1st UK Ireland Computing Education Research Conference (UKICER).* Association for Computing Machinery, 1–7. https://doi.org/10.1145/3351287.3351292
[2] Erin J. Zaffini. 2017. Communities of Practice and Legitimate Peripheral Participation: A Literature Review:. *Update: Applications of Research in Music Education* (Nov 2017). https://doi.org/10.1177/8755123317743977