Epicodus Internship Report

Michael Kaiser-Nyman, President

Maureen Dugan, Outreach Coordinator

For more information: maureen@epicodus.com

Participating Companies

Bunker

cloudability

CROWD SUPPLY

CrowdCompass

DAYLIGHT

Factor.io

graphAlchemist

mentor me

Notch8

POP ART

Quick Left

REVELATION

SpendWell™

switchboard

THE CLYMB®

UNCORKED

Summary

After conversations with all of the employers who participated in Epicodus's internship program, we've come up with some suggestions we hope anybody hosting interns will consider:

- Integrate interns as closely as possible with permanent staff.
- Before interns start, run through a clean installation of the software they will be working on. Fix any issues and document the process.
- If interns will be working on an existing product with other developers, start flagging bugs and features appropriate for the interns two or three weeks before they start.
- Informally check with interns as often as once an hour.
- Have interns sit side by side so that they can easily help each other out.
- Designate one or two staff developers who interns can first turn to when they are stuck.
- Hold weekly one-on-one, process-oriented, non-technical checkins with each intern.
- Organize a weekly social event, such as lunch, to bring interns, their supervisor(s), and other staff together.

Setting up an internship program takes work, but the participating employers were universally happy with the outcome, whether it was getting work done on an open source project they had never gotten around to, hiring an intern into an employee at the end, or just the joy of working with excited new developers. We hope our findings and suggestions from this report will make it easier for any company to start a successful, rewarding internship program.

Background

Epicodus is a four month, forty hour per week, in-person class on programming. Students coming into Epicodus have tried programming before, either on their own or through other introductory classes. At Epicodus, they learn full-stack web development for four months, and then we help connect them with local companies to fill junior-level development positions.

In our spring 2014 class, we introduced an internship component for the last month of the course. 17 companies hosted 43 students through our program. Some of them had established internship programs, but most had never hosted interns before. The only requirement from Epicodus was that the students spend at least 30 hours per week coding and have at least one developer staff member mentoring them.

After the end of the program, Maureen and I sat down with every company that had participated to learn how they structured their internship, what their onboarding process looked like, how they provided support to their interns, what they thought went well, and what they want to change next time. We also talked to the students who participated in the internships to find out what worked and didn't work from their perspective. From these interviews, we distilled down some ideas and practices we think every company with or starting an internship program should consider.

Types of internships

The work given to interns broadly fell into three categories.

- *Full integration*: Some companies added their interns directly to their existing development team. For example, CrowdCompass and Factor.io simply assigned tickets to their interns from their main development queue.
- *Carved-off project*: Other companies caved off an independent piece of the their main product off for the interns to work on. For example, Revelation assigned their interns to build a new video uploading tool for their app that had minimal interaction with the rest of their codebase.
- *Siloed project*: Finally, some companies gave their interns a project that had nothing to do with the company's main project (or, for consulting companies, nothing to do with client work). For example, Notch8 had their interns work on an existing app for their non-profit client House of Sound, and Pop Art had their interns add features to an open source file transfer utility.

Students in every category of work were very happy with their internships and the opportunity they had to experience a professional programming environment. That said, generally speaking, **the more closely integrated the students were to the company's staff, the more they felt**

productive, supported, and happy. Companies using the "full integration" strategy naturally assimilated the students into their staff, whereas companies who carved off projects or assigned a siloed project had mixed success. This isn't to say that a carved-off or siloed projects are a bad fit for interns; it simply underscores the need for extra diligence to make sure that interns are integrated into their host companies in other ways, as outlined in the following sections.

Preparing

Regardless of what type of work was given to the interns, every company spent some time preparing the project for the interns to work on. Here are a couple things that worked especially well.

Project installation

Many projects, especially on teams that have not added a new developer for a while, are missing setup documentation or dependency configuration. Several companies proactively ran through setting up their project on a clean machine (or virtual machine) to identify any problems with their setup process and to improve their setup documentation. Huston Hedinger from Graph Alchemist told us: "Setting up the project for new junior developers helped me catch problems and find pain points that I hadn't noticed before. Fixing those issues and improving our documentation made it easy for our interns to get up and running, and also will make it easier for future developers to get going, which is very important for any successful open source project." We'd strongly recommend that companies **run through a clean install, fix any issues, and document the setup process before interns start**.

Flagging issues for interns

Some companies that fully integrated their interns into their teams started marking simple features and bugs and saving them for interns as far out as a month before the interns started. By the first day of the internship, they had several days worth of relatively easy tasks for the interns to get started with. Maciej Skierkowski from Factor.io said: "Three weeks before my interns began, I started flagging things I thought would be good for them to start out with. On the first day of their internship, my interns were able to pick up a couple small bugs, fix them, and push their own code into live production. It gave them an immediate gratification, and made Factor.io confident that they would have a cadence of strong contribution." **Flagging bugs and features a few weeks before interns begin** will help make sure the interns are able to have meaningful, appropriate work as soon as possible.

Onboarding

Most companies' onboarding process involved introducing the interns to the other members of the team they would be working on, showing them around the office, and getting them situated at their workstations. They also went through their development process and tools, such as their ticketing system, the format of their daily standups, and how their sprints are organized.

Depending on the complexity of the project, interns took one to three days to get their development environment set up and ready to use.

Technical support

The companies participating in the internship program developed several strategies to help support their interns when they needed technical help.

Informal checkins

Most companies had a mentor assigned as a point person to work with the interns. Some of these mentors made a point of checking in with their interns multiple times a day to make sure things were going well. A common and surprising theme we heard from employers was that interns didn't ask for help enough. Jonathan So of Revelation Global told us that "The biggest feedback I had for my interns was that they should ask for help quicker and more often." From our conversations with students, we learned that many of them were worried about wasting their mentors' time, to the point where they were overcompensating by not asking for help enough. At Bunker, Nathanael Merrill developed a strategy to deal with this. "A few times I found out that one or both of my interns had been stuck on a problem for several hours," he recounted. "I began just checking in with them every hour or so to see what they were working on and if they needed any help. Usually it only took a minute or two, but often it saved them from spending a long time researching an answer I could tell them off the top of my head." We think Nathanael's developed a good rule of thumb: **when possible, informally check with interns once every hour or so.**

Peer support

Part of how Epicodus structured our internship is that we sent at least two students to every company that participated. Our goal was for the students to provide each other support, so that they could turn to their pair to ask questions before taking the time of an employee at the host company. We got very positive feedback on this approach from the host companies, saying that

it generally worked very well. Often, **companies had interns sit side by side, so that they could easily turn to each other when they needed help.** Lorenzo Ciacci of the Clymb reported that "Sending us two interns was genius. For the first few days, Lauren and Aaron pair programmed on the same computer, working together to wrap their heads around our codebase and get through their first few tickets. As they grew more comfortable, they worked on separate computers but continued to ask each other questions, bounce ideas off each other, and occasionally come together to pair on more difficult problems."

Support from experienced developers

Of course, interns need more support than what they can give each other. Most companies had some kind of chat room where developers asked each other questions, and added interns to the room. On the positive side, these rooms provided a way for the students to get help whenever they needed it. On the negative side, students reported that they often felt intimidated by asking questions in front of all the company's developers. What some companies found effective was to **have one or two "go-to" people who were especially excited about working with the interns and whom the interns could ask questions when they got stuck.** For example, at Daylight Studio, Shawn Mann told us that "We decided early on that Dave McReynolds (partner and head of development) would be the point person for the interns. In addition to regular meetings with the rest of the team, if they needed help or guidance, they could go to him directly. We believe this led to less confusion and a better overall experience for both the interns and our company."

Non-technical support

Before the internship began, most companies and students were primarily concerned about technical aspects of the internship. It turned out, though, that the non-technical aspects were just as important.

One-on-one checkins

All companies had checkins of some sort, whether specifically for technical review or just generally to see how things were going. Some companies went further and instituted specific weekly one-on-one checkins to see how the intern was doing and how the internship could be improved. These companies said their one-on-one checkins were invaluable for helping identify problems early on and make the internship work better. For example, Andrew Beeks at CrowdCompass said that "We had multiple weekly one-on-ones with interns - one with a focus on progress, technical issues, team engagement, Q&A, etc. and the other focused only on non-technical items related to our hiring and management processes. They were tremendously

useful in helping us to identify issues interns were having and, during the first meeting, we discovered that two interns had run out of work to do the previous day and didn't know where to get more. In another instance, interns were able to articulate areas where they lacked any training or context, which in this case was relating to our QA processes. By regularly soliciting feedback from interns we were able to guarantee they had more meaningful work with a greater understanding of our own SDLC while also identifying areas of concern for interns, including their successes, failures, long-term objectives, career aspirations and how the CrowdCompass internship fit into those plans." We'd strongly recommend every company **institute weekly one-on-one, process-oriented, non-technical checkins with each intern.**

Social events

A few companies instituted weekly lunches with their interns and a couple staff. For example, Matt Clark at Notch8 said that "Getting together for lunch and outside of work was key to helping the developers and interns build personal relationships and work better together." We'd suggest taking a page from Matt's book and **have a weekly social event, such as lunch, to bring interns, their supervisor(s), and other staff together.**

Conclusion

Despite the loose structure and lack of guidelines of our first internship program, we heard very positive feedback from almost every student and employer who participated. We've learned, though, that there are a few relatively low-investment practices that can make an internship work very well - and often improve the software and team process of the hosting company. We hope you'll take some of these recommendations and try them out at your own company.

We want to leave you with a final thought from Chris Coogan, Director of Technology at SpendWell Health: "The Epicodus developers that we brought on as interns were productive within 2 to 3 weeks. It always takes some investment to get any new developer up to speed on a team and it turned out to be a very worthwhile investment with the Epicodus students."