
Examining Transition and Integration of Graduate Apprentice Software Engineers through Onboarding

Omar Tufayl (2027205)

April 23, 2020

ABSTRACT

This research study attempts to understand the experiences of software engineering graduate apprentices through the framework of onboarding processes as a means to effectively transition from academia to industry. After conducting interviews with apprentices and their employers, investigating their experiences with preboarding, socialisation activities, support networks, understanding codebases, software tools and work methodologies, access to information and knowledge bases, and onboarding evaluation, apprentices generally held positive opinions on their onboarding experiences, with suggestions for areas of improvement.

1. INTRODUCTION

There is a gap between the skills and competencies of Computing Science and Software Engineering graduate students and the work-ready demands of enterprise and industry. [1] Academia has primarily taught theoretical underpinnings, assuming graduates and employers will facilitate the connection between knowledge and practice in the real world. This is no longer a sustainable model as evidenced by the growing discontent from employers seeking to engage with a workforce that is more relevantly suited to the requirements of today's workplace realities. [2]

Universities are therefore increasingly partnering with industry to create paths for students to acquire the deeper principles of computing on campus while in parallel practically applying and contextualising that knowledge in the workplace through Graduate Apprenticeship (GA) schemes, internships, placements and coursework based on simulating real-world project dynamics. [3][4] These initiatives attempt to enhance practical abilities by addressing the needs and challenges of real world software engineering activities individually and collectively as part of larger teams, thereby allowing students to gain a greater depth of knowledge and practice, becoming increasingly competent and holistic practitioners.

While the imperative to successfully bridge the gap between theory and practice has been understood and accepted by universities, students and employers, and various initiatives have been undertaken to address this issue, a closer inspection into the techniques and processes implemented by industry to successfully integrate software engineering students into the workplace, specifically the use of onboarding processes as transitioning mechanisms, will help to further close the current gap between theoretical understandings and practical abilities, highlighting observations to continually improve the current educational paradigm, making

the transition from academia to industry more productively effective. [5][6]

1.1 Software Engineering Apprenticeship

The Software Engineering Graduate Apprenticeship programme (GA) has been developed by The University of Glasgow to address the knowledge and skills gap between theory and practice, alternating time apprentices spend on campus learning computing science and software engineering, with time in the workplace covering work-based learning in partnership with employers. This combination allows apprentices to experience both aspects of learning and allows the university to align its teaching with industry needs. This study attempts to understand the experience of software engineering graduate apprentices enrolled in this course along with their employers through the framework of onboarding processes, determining it as an effective means of transition from academia to industry.

2. BACKGROUND

Current research concerning onboarding primarily focuses on general practices across the industry. The needs and requirements of students engaged in graduate apprentice programmes that closely combine theory and practice in complementing cycles of activity, do overlap with professionals but are also unique in that apprentices have little to no prior industry experience from which to draw upon and comparatively better understand work environments and their professional requirements.

2.1 Onboarding Advantages

Effective onboarding processes have provably shown to afford multiple benefits for integrating new employees into organisations. From improved attraction and retention of software engineers in a competitive talent acquisition landscape, providing an accelerated experience for orientation and initiation in a new work environment, delivering deeper comprehension with codebases and software tools, to measurable productivity and financial returns on investment for organisations, including reduced time and effort for human resource and talent managers, better overall new employee integration leading to stronger collaborative bonds among work colleagues. The advantages of onboarding within the software engineering industry are well-suited to address the transition of apprentices from academia to industry. [7][8]

2.1.1 Employee Recruitment

Successfully executed onboarding practices in the context of broader recruitment strategies are proven to positively in-

cline prospective employees towards companies, creating the basis for future engagement with company goals and culture. [9] In a competitive software engineering acquisition talent market, the result of an onboarding process specifically oriented towards apprentices will further enable employers to create appropriately suited integration processes, accordingly positioning themselves with advantage to potential employees.

2.1.2 *Employee Satisfaction and Retention*

Employee satisfaction levels can be directly influenced by onboarding activities, and consequently an onboarding process designed and implemented cognisant of this can better align employees personal goals with organisational aims and objectives, establishing a superior employee-organisation fit [10], ensuring short and long-term retention and avoidance of increasing employee acquisition costs. [11]

2.1.3 *Organisational Performance*

Effective onboarding has a critical effect on the time to and sustained employee productivity and performance, directly correlating with organisational ability to achieve aims and objectives. [12] An onboarding process informed and designed through consultation with employees and employers allows companies to achieve a faster time to productivity, enhancing overall performance.

2.1.4 *Organisational Success*

The combined favourable outcomes of increased employee recruitment, satisfaction and retention contributing to faster routes to higher organisational performance, all promote the likelihood of achieving success as defined by the organisation through lower human resource turnover costs, higher retention rates and faster and more productive output per employee. [13]

2.2 **State of the Art**

The existing literature regards onboarding for professional software engineers is extensive with studies ranging from the value employees perceive gaining from the process [9], and it's importance in attracting, retaining and incentivising new software engineers [14], to facilitating faster comprehension of an existing codebase [15] and the effects of socialisation activities on work commitment and performance. [16] Further operational and competitive advantages are demonstrated by organisations utilising onboarding, creating more cohesive cultures, reducing high financial costs associated with employee hiring and turnover, and achieving accelerated overall performance through new employees.

A number of studies have also compared various strategies to better equip new employees towards enhanced productivity, specifically in the software engineering field [6][13][17]. These practices seek making the transition to a new role a positive experience for both the employee and the organisation as a whole. Studies have been conducted on understanding onboarding as a tool in improving initial code comprehension [18] and exploring strategies to assist software engineers in efficiently locating resources to support development [19]. Further research has examined how early socialisation affects employees in acquiring required domain knowledge [20][21] and the impact of onboarding on employee motivation and commitment towards organisational mission statements. [10]

The currently available research however does not in the main specifically address the unique needs and requirements of student software engineers advancing into industry roles, particularly addressing their lack of prior experience entering the workplace and the effects, or none, this has on their integration into organisations. Our hypothesis is that students, specifically apprentices, have individual and collective requirements during onboarding that are particular to their career level and professional standing. In general, apprentices transitioning to the workplace are doing so for the first time in a full-time role, and in particular face challenges seasoned professionals do not such as learning soft-skills, namely different communication styles and protocols, project management and the often uncertain and changing nature of real-world project dynamics. Additional challenges in the workplace include understanding organisational structures and in particular hierarchies of authority which can be explicit or implicit, particularly in the form of flat organisations, which can add complexity to initially overwhelming environments. Further, dynamic interactions with clients and customers is often in contrast to a learning environment where assignments are clearly and precisely specified with defined evaluation criteria.

3. **METHODOLOGY**

This study conducted interviews with apprentices and employers to understand experiences of onboarding processes from both perspectives for the purpose of gaining insight into best effective practices, challenges faced integrating apprentices into the workplace, and views on how to improve the alignment between academic activities with industry needs. Specific areas of consideration are preboarding, orientation and initiation, formal and informal socialisation, support networks and learning, codebases, software tools and work methodologies, information and knowledge bases and overall onboarding evaluation.

3.1 **Objectives**

The key objectives were to collect and aggregate qualitative data, analyse and understand underlying themes of interest to make observations intended to act as reflections on data gathered.

- To collect and aggregate the results of inquires into onboarding practices at companies with software engineering operations through detailed interviews with apprentices and employers.
- To analyse and understand current practices and compare the results to better understand apprentices experiences, identifying what is currently working and where there is room for improvement through discussion and reference to the state of the art.

3.2 **Participants**

To gain participants for the study, apprentices and employers were contacted through the Software Engineering Graduate Apprentice course coordinators at The University of Glasgow.

3.2.1 *Apprentices*

Four software engineering apprentices took part in the study, two at small to medium sized organisations and the

remaining two at a larger organisation. They are referred to here as A1, A2, A3 and A4.

3.2.2 Employers

Three employers took part in the study, all involved in developing software of various scales, from web based products to internal software tools for operations and administration management. They are referred to here as E1, E2 and E3.

3.2.3 Interruption to Research Activities

During the study period, a viral outbreak, named COVID-19, was declared a Public Health Emergency of International Concern by the World Health Organisation on 30th January, 2020. [22] This prevented face-to-face interactions and caused a reprioritisation in activities for potential participants, resulting in lower than expected numbers of apprentices and employers being interviewed.

3.3 Areas of Concern

The interview questions consisted of six broad areas of concern, including preboarding, induction and orientation, formal and informal socialisation, support networks and continuous learning, codebases, tools and work methodologies, information and knowledge bases and evaluation and feedback mechanisms.

3.3.1 Preboarding, Induction and Orientation

Activities conducted before an employee actually commences work and intended towards initiating the employee into the company are collectively termed as preboarding, induction and orientation. These include the first points of contact such as job posts, recruitment fairs, branded marketing material presenting the company's aims, objectives and culture, initial in-person and digital communications, and the first welcome to the work environment. Also included in these are routine activities such as human resource requirements, acquiring tools necessary for job roles and an introduction towards the company and specific job expectations and methodologies.

3.3.2 Formal and Informal Socialisation

Meeting people in an organisation is integral to employees feelings of community and belonging, which are fundamental in creating a cohesive and collectively aligned organisation. [9][11] With high costs to employers associated with employee attraction, selection and retention, one of the main purposes of onboarding is to maximise the level of new employee engagement, quickly and productively, in particular through activities that socialise the employee effectively within the organisation, transforming the employee from an outsider to an insider. [16] Socialisation activities for newcomers can be understood as informal, those comprising casual interactions, and formal, where interactions are planned with specific intent and purpose to fulfil a particular organisational outcome.

3.3.3 Support Networks and Continuous Learning

Through socialisation activities new employees are able to build networks of support within the company for assistance and instruction when needed, drawing on more experienced colleagues knowledge when faced with uncertainty. This provides a means of guidance which increases the newcomers familiarity with work practices and reduces time to actual

productivity. [23] Alongside support networks, regular one-to-one mentoring and skills training programs, new employees can continually learn and advance in their capabilities, creating a sense of the organisation being invested in their personal development, thereby establishing a greater commitment and intrinsic motivation towards achieving overall company aims and objectives.

3.3.4 Codebases, Tools and Work Methodologies

Software engineers can encounter difficulties in trying to comprehend code written by others, in turn slowing development time and as a result increasing the cost to maintain codebases for the organisation as a whole. [24] When and how new software engineers become acquainted with codebases, and the tools and work methodologies used can have an important impact on their integration within their teams and their eventual effectiveness and ability to make meaningful and impactful contributions, leading to feelings of accomplishment and purpose, both critical in maintaining productive momentum for individuals and teams.

3.3.5 Information and Knowledge Bases

Identifying and accessing relevant information and knowledge bases reduces the need for apprentices to reach out to other team members and provides a source of autonomy in being able to solve problems, alleviating feelings of being a burden on others and creating a sense of individual competence. Companies that provide and continually enhance such resources are able to leverage company-wide knowledge and make it accessible to everyone within the company, thereby operating at the collective knowledge level of all employees.

3.3.6 Evaluation and Feedback Mechanisms

Giving employees ample opportunities to provide feedback within the organisation creates a healthy environment open to adapt and evolve in the face of changing internal and external realities. Properly facilitated feedback mechanisms are essential in allowing newcomers to provide a valuable outsiders perspective on conventional company policies and practices. Alongside this, newcomers need to be included in the design and delivery of the onboarding process to ensure its regular improvement and relevance to employees actual and real needs.

4. RESULTS

Comprehensive question sets for both apprentices and employers addressed the key areas of enquiry and are divided into those directed towards apprentices (A) and employers (E) respectively, querying specific onboarding practices from each perspective.

4.1 Defining Onboarding

Employer interviews began with asking what onboarding meant to them, when the process began and what its duration was and its overall importance. They were also questioned on whether the process was personalised for each individual, or if there was a standardised procedure for everyone.

What does onboarding mean to you?

All employers agreed that onboarding is a means in bringing newcomers to the point where they integrate enough with the company, understanding its products, technology, processes and culture to be able to fully participate in business

directed activities in their particular role. “Getting someone to the point where they are familiar enough with the company, its products, its processes, its culture if you like, to play a part” (E1). For another employer giving the apprentices “the best experience” (E2) was the overall purpose of onboarding so that they could “integrate into the organisation, that they understand their role, and what they’ve been asked to do and that also our managers and the teams understand why they’re here” (E2). “I think it’s a mixture of the cultural aspect, as well as the technical aspect, and just the general administration side of things” (E3).

When does apprentice onboarding begin?

The interview stage is where all employers stated onboarding began for their organisations. “It begins before they get to the company, at interview stage, we always have as part of the interview process a real task that we set them that is actually related to what we do and that’s a software development task that gives us an insight about how they go about solving a particular issue with our platform that we use” (E1). Another employer said “company wide, it begins at the recruitment stage, so as soon as that offer is made that’s when we start looking at the onboarding experience” (E2). This was echoed by a third employer: “when we offer them the job, which is prior to them starting, it could even arguably be as early as the interview by setting expectations” (E3). Once interviews had been conducted, making sure new employees have a point of contact within their eventual teams was a priority for employers. Administrative tasks such as explaining remuneration, holidays, benefits, preparation for the first day at work and gaining references are all be done once job offers are made.

How long is the onboarding period?

The onboarding period ranged from a few weeks to several months, the actual duration depending on the apprentice being onboarded, and also on the size of the organisation. “Depends entirely on the individual, what their skills are when they arrive” said one employer (E1), adding “you’re observing people, you’re watching their reactions, you’re not doing it in a formal prescriptive way, you’re picking up on vibes”. In smaller organisations apprentices were able to frequently interact with all members which aided their accelerated ability to contribute quickly to their role responsibilities. In larger organisations, apprentices had more formalised onboarding procedures and these usually could last for three months. “Although most of it will be in the first month, all that heavy duty stuff, their welcome day, understanding a bit more about the business, there are two to three months where they have constant check-ins with their manager” (E2). In both cases, managers, mentors and team colleagues did not expect apprentices to contribute immediately and allowed an extended time to familiarise themselves with their environments and work responsibilities. Onboarding was also seen as part of a larger process, transitioning into longer term training: “the intensity of the onboarding kind of wanes off, it never disappears because it goes off into a staff internal training process” (E3).

How important is onboarding for your organisation?

Attracting new software engineering talent and retaining that talent were the two primary reasons employers stated being the essential factors driving the prioritisation of on-

boarding within their organisations. “It’s all about the recruitment but also the retention of staff as well, and onboarding is a huge part of that” stated an employer (E2). Employers cited typically losing staff within the first three months, adding significant emphasis to being able to integrate new employees into the organisation quickly and efficiently by aligning the needs of the organisation with those of newcomers. Being cognisant of apprentices inexperience, employers stressed the need to give them the confidence and skills to adapt and grow within their new professional roles, believing the initial onboarding experience establishes the foundation of the interaction between employee and the organisation. Onboarding “sets the tone of the whole relationship with your organisation” (E3).

Who is involved in the onboarding process?

Smaller organisations involved all members in the onboarding process, with apprentices being able to interact with everyone, in varying levels of engagement depending on their job role. “It’s spread, there are people who are less involved, there are some personalities that are more geared up for it, maybe more patient and they don’t mind going through the basic stuff” (E3). In larger organisations the main points of contact during onboarding were the immediate team and human resource personnel, along with initial interactions with higher levels of management.

What tools do you use to design and manage your onboarding process?

Simple checklists, and intuition were the guiding technologies for smaller teams, referring to resource constraints and an inclination to adapt to the apprentices needs as and when required. “I don’t have the time to do that, as well as product development and business development with everything else I’m doing, it’s all done intuitively” (E1). Some organisations utilised software applications specifically designed to create and perform onboarding, including progress indicators to allow apprentices and their managers to understand how they were progressing and where interventions or support might be appropriate. “We’ve got our learning academy tool, that’s where our onboarding stuff sits as well. We make sure people work through that because it’s a good tool for reminding them what they have to do” (E2). In both cases, onboarding is viewed as evolving through feedback from newcomers to continually improve the process. “We’ve actually got two checklists, the first thing we tell people when they start is to make notes of things they’d wish they’d known, and then we feed that back so whoever was the last onboarded updates the onboarding doc for the next person” (E3).

How personalised is the onboarding process or is it standard for everyone?

In all responses, employers mentioned their onboarding process having both personalised and standard elements. “It’s very personalised, we’re dealing with one individual at a time in a small team so they get a lot of personal support and personal help” (E1). In larger organisations there is a combination of personalisation and standard procedures, with all apprentices provided with basic induction, with more personalised training and onboarding being given by immediate team members, depending on their specific area of function within the organisation. “So they have that structured pre-

sentation on day one, they had their welcome day, and the rest of that time was spent with their individual teams” (E2).

4.2 Preboarding

4.2.1 Apprentices

Between conducting their interviews and beginning the first day of work, apprentices described exchanging several emails with human resource personnel at their respective companies, conducting background checks, gathering of personal and financial details, signing of contracts and welcoming them to the company. One apprentice was invited to a lunch introduction with the rest of the team before the actual working start date. Other than information through official routes of company culture and objectives, apprentices stated they had researched the company on their own by visiting their website and conducted searches on social media to find out more about the people and teams they would be joining.

How did you come to know about the aims, culture and history of your organisation before starting?

Official induction days specifically designed for the apprentices, include business and technology presentations to share more information about overall objectives alongside various projects currently underway. “We had a meeting with the CEO and the CIO and they told us why they were doing the GA [Graduate Apprentice program] and what they hope to achieve, so it was a lot of bringing people like us in to develop new ideas. That was quite useful to see where they’re going” (A1). Apprentices also conducted research on social media and websites, understanding aims, culture and history of their respective employers. “It was just reading stuff on the Internet, and getting my own idea, so I had little bits here and there, but after the presentation, it all came together, and I was like I know where I’m working now” (A4). One apprentice said “I blitzed their website” (A3), another mentioning doing searches on Twitter. In combination, knowing more about the companies eased apprentices early apprehensions before going into their new environments.

4.2.2 Employers

How are administrative tasks carried out?

The majority of administrative tasks are carried out between the interview and the actual first working day, with the remaining being finalised within the first few days, at most weeks. Most of the initial requirements such as details for legal, payroll, policies and contracts are signed and sent back to the employers before the first working day via email. “Yes it’s before they come in” (E1). “Payroll, all that stuff, policies are all sent to them, their contract signed and sent back to us before they even start” (E2). In larger organisations, further administrative work is carried out on specific platforms designed to automate the process and provide standardised and consistent experiences for all apprentices. These can be completed within an agreed time frame, interspersed with work activities during the first few weeks.

How is the organisations aims, culture, history introduced to apprentices?

“We talk informally, but the weekly conference is where I tend to talk, give them a bunch of information about what

we’re doing” (E1) one employer said. Employers spoke about the importance of giving apprentices a sense of responsibility and an appreciation of the impact and influence of their role and how they contributed to overall success of the business. Apprentices were also involved in activities surrounding Scottish Apprenticeship week, with presentations to their CEO, and upper-level management, giving them visibility that is not usually associated with larger organisations, recognising their important role in the overall national economy. “We get our graduate apprentices to meet with our CIO once every month and they meet with our CEO at the start of their program, so that they can get that visibility on the business that they might be lacking” (E2).

4.3 Orientation and Initiation

4.3.1 Apprentices

What was your first day like?

Apprentices described their first day using words such as ‘intense’, ‘intimidating’, ‘uncertain’ and spoke of feeling nervous about being able to live up to perceived expectations from employers. “Intense, it was intense, it’s quite intimidating, a lot of uncertainty” (A1). The term “imposter syndrome” (A1) was used, “just feels like you’ll get caught out one day, like I shouldn’t actually be here” but in the whole these fears were eased by employers welcoming apprentices through gestures such as greeting them at reception and accompanying them as they were introduced to wider team members. One apprentice experienced getting lost while trying to find the company offices, “I got horrifically lost and didn’t know how to get in the door but I got in eventually” (A3), while another spoke of their manager not being present on their first day leading to some confusion within their teams as to how to engage the apprentice. “My manager wasn’t in that day, at reception you’re suppose to call for your line manager but she wasn’t in, so I had to contact [another] manager who wasn’t... didn’t remember that I was coming in that day, so she came down and said ‘oh, I forgot’, it all seemed a bit rushed” (A4).

Gender imbalances and a sense of software engineering being a predominantly male occupation were reinforced during the initial stages of interview. “I was the only female during the interview, and it feels like there’s a different dynamic a little bit” said an apprentice (A1), adding “you’re very aware that you don’t want to say too much and come across as quite bossy and kind of pushy as a girl, so there’s always a slight awareness in the back of your head”. These concerns were removed once apprentices integrated into their teams where they found a balanced gender diversity. This was referred by female apprentices as a major factor in their continued commitment to the graduate apprentice programme, leading to an increased sense of belonging and engagement within the software engineering profession. They found their working environments “a lot more relaxing” (A3).

How was your workstation setup?

Workstations were mostly setup by apprentices with minimal guidance, “I kind of got left to it a little bit” (A1), allowing them to get involved in figuring out how to properly configure their work environments, with appropriate software and tools, leading to a sense of accomplishment in their first few days. In some organisations, individual work is carried

out on the apprentices machine, with team centred activities conducted on already configured shared computers.

Were your job role expectations explained to you?

The majority of apprentices stated experiencing their greatest uncertainty in the first few weeks where their responsibilities and job role expectations were unclear while facing increasing amounts of new information such as the use of version control, unknown programming languages and paradigms, complex mechanics of the devops process, and contending with fully understanding agile methodologies while additionally attempting to comprehend new technical terms and domain specific vocabularies used in the software engineering workplace. Understanding and emphatic support and guidance from mentors and wider team members was mentioned as essential in managing this challenging period for apprentices, leading to a renewed appreciation for their own potential and commitment to organisational aims and objectives.

Support from their mentor was crucial at this time to reassure the apprentice of fears they had: “my mentor did talk to me about it, ‘I know this is going to be tough but you can do it and if you can’t, tell me immediately’, so it did feel there was a support there when required and constant checking in” (A3). Another apprentice also spoke about the support they received: “my manager was great because the first day he sat with me and said ‘we’ll go through what we can get planned for you, what we can get you sitting down and looking at” (A1). One apprentice indicated struggling more than most with regards to job role expectations, citing joining at a time when their team were occupied migrating and upgrading a project, leading to an insufficient amount of time available to assist and support the apprentice. This improved once the team initiated work on a new project, involving the apprentice increasingly in development work, leading to greater enthusiasm towards their work, and hope this would further improve as their knowledge and skills advanced.

Can you describe the first task you did?

First tasks for all apprentices were designed mindful of their inexperience, involving them in activities that were simple enough to accomplish but which also provided an insight into broader software engineering systems for development and delivery. “It was mostly looking at the JSON which is where we get the database linking into the program” (A3). These initial tasks ranged from changing a configuration file and updating text on a website to formatting a data exchange document and understanding basic server management. “The first tasks I did were to update text on the website and that introduced me to the whole development flow” (A2). While the tasks varied, common responses from apprentices were that undertaking these tasks allowed them to see continuous integration and deployment pipelines in action, and becoming familiar with the methods and tools needed to accomplish work. These tasks were often guided by mentors allowing apprentices to learn new skills in a supervised and collaborative manner. “That was fairly simple and I was pairing with my mentor at the time, so that was good, she just kind of step me though it” (A1).

How were you feeling after your first week?

After the first week, apprentices described being overwhelmed with self-doubt despite being supported by managers and team members reiterating they were new to software development, and that they should not expect to understand everything at once. They were also assured there would be an initial period of customising to various tools and practices in becoming proficient and making meaningful contributions to their work. “After the first week I felt really lost, everything was new, I didn’t even know what Git was, so to go to a full on development environment, was like a shock to the system” said an apprentice (A2), explaining the support they received was helpful in this period: “but the good thing is we have a developer that started before me and so all the questions that I had, he would help me”.

Being given more freedom and autonomy to self-manage, and prioritise their work than they had previously experienced gave apprentices a sense of independence, but also caused insecurity initially in not being explicitly told what to do with their time. “You’re left to prioritise your own work and you’re given that independence, it’s a great thing but I think at the start I was, I feel like someone should be telling me, you should be doing x, y and z” (A1). These feelings gradually dissipated over the subsequent weeks with apprentices gaining in self-confidence and being able to participate fully in daily standup and kanban meetings.

Overall apprentices felt invigorated and excited to continue in their new roles after their first week. “I was more excited than anything. I was excited to get on with a new thing, a brand new thing, for me to do, a new job. Obviously the pay is decent and to just get on with it, that’s what I was excited for, I was excited for just everything” (A4).

4.3.2 Employers

What is the employees first day like?

Employers aim to allow the apprentice to experience mainly a mixture of social and technical aspects during their first day, with a lunch or coffee with team members to allow for informal conversation and a rapport to be gradually built in getting to know one another and also for an introduction to the software products currently in development. “It’s partly social we tend to go out for lunch just to get to know the person a bit better, we also give them time to work through products, just to get a real feel. That’s done with a mentor saying you know, do this, learn that, go there, that process can take quite a few weeks” (E1).

This also allows apprentices to begin understanding their role better within their teams, “it was a bit of an understanding about what their team does and where they sit within that team” (E2). Apprentices also began understanding the culture of the company, the physical layouts of offices, platforms used for software delivery, security and infrastructure, along with particular software applications used, and acquainting themselves with the various technologies, particularly those requiring further study to become familiar with. A typical first day with one employer was where the apprentice would “start an hour after everyone else, we get the time to get all of our emergencies out of the way, and then setting up their accounts, getting them onboarded with version control, email, calendar, all of that stuff, even if they’re not so familiar with the operating system, can be even as basic as that, getting their desk setup, bit of show and tell around

the office, introduction to folk so they have a bit of time with everyone including the remote guys” (E3).

Would you describe your onboarding as structured, semi-structured, or unstructured?

Onboarding in the organisations interviewed was unstructured for those that were smaller in company size, with employers saying they adopt that approach preferring to adjust and respond to the apprentices individual and particular needs as the onboarding progressed rather than have a standard procedure that was not dynamically adapted to any situation. “There’s an element of does that make sense to you, what do you want to learn about now, we can talk about the customer or we can talk about this or we can talk about that, what do you want to do, otherwise it just becomes monotonous” (E3). Those with semi-structured processes, mixed elements of standard procedures applied to all apprentices with those that adjusted to each apprentices requirements. Structured delivery would be conducted by company colleagues outwith the apprentices immediate team and unstructured onboarding would be predominantly undertaken by mentors and buddies.

How is the first day made as satisfying and effective as possible?

An understanding and affinity for the psychological state of apprentices, taking into consideration their lack of previous work experience, came through when employers spoke about treating apprentices well, putting them at ease by being aware they will most probably be feeling overwhelmed and nervous, and accompanying them so that they are not left on their own while simultaneously balancing that with giving the apprentice a sense of independence and freedom to explore their new surroundings. “I think being conscious of the fact that they’ll be overwhelmed, they’re always overwhelmed, they’re always nervous, they’re never quite themselves, so it’s just putting people at ease” (E3).

4.4 Software Development

4.4.1 Apprentices

How were you introduced to the codebase?

There were a range of responses here, from being given guided codebase walkthroughs to initially exploring the codebase without any assistance, asking for clarification as and when needed. The initial period when apprentices first accessed the codebase also varied, with some having exposure very early on in the first week, to others being shown the codebase weeks, and in one case, months into their apprenticeship. “My mentor gave me the codebase and said it’s there, you can go through it if you want, and then she started off giving me little changes to try. I know there are areas that I don’t play with because that’ll mess with the entire program but I appreciate that they are there and how they work” (A3). Most recollected this as being a challenging experience, having had very little to no understanding of real-world codebases and the tools used to manage them. Encouraging prompts from team members to use particular features within integrated development environments to inspect the code made it easier to find relevant functionality and to gain an overall clearer comprehension of the codebase. “I was still a little bit intimidated because I thought

this is going to be huge and I’m not going to know and then working with the support pair and them giving me a little advice on what bits are easier to follow, I found helpful, seeing people do something” (A1).

Shadowing team members, watching them work on the codebase, was a particularly helpful experience for apprentices, allowing them to gain exposure to the codebase while a more experienced developer explained their work, making it easier to follow. Coursework for university focused on understanding codebases also allowed apprentices to ask questions in the workplace that further clarified concepts. Code walkthroughs allowed apprentices to use version control tools, understand the software architecture and different services within the organisation, with attention being directed towards simpler areas of the codebase that the apprentice could start to experiment with. “I didn’t even pull because I was so lost, so they pulled for me and explained this is how you change branch, this is how you pull and push, all that” (A2).

How did you get introduced to team workflows?

Team workflows were significantly influenced by the size of the team, with smaller teams having more informal lines of communication contrasting with larger teams following more structured practices. Agile methodologies were used throughout to varying degrees of customisation suited to the organisation. “We do retrospectives, we have two or four week sprints, that’s pretty much it as far as Agile goes” (A2). Apprentices spoke of being included in daily scrums and standups, participating in pair programming and kanban sessions and discussing backlogs, user story cards and previous sprints during retrospectives. At first these new ways of working were daunting, “I think at the start, I barely uttered any words, I would just kind of listen, even the retrospectives I remember at the start, they’ll do a lot of anonymous voting for what we can discuss and things and I didn’t even want to participate in that” (A1) but as their confidence and experience grew they found themselves supported by team members to gradually participate in technical discussions.

Teams varied in their support to participate, with some including apprentices in their first weeks and others delaying inclusion till several months later whereby the apprentice had gained experience and familiarity with software tools and development in general. Instant messenger applications are used considerably to facilitate inter-team communication, alongside the use of cloud based collaborative software applications to share documents, schedules, storyboards, activities and for prioritising and making work visible throughout the team and organisation. “We have a shared Discord server, that’s where we also communicate as well as on Google, we’ve got a shared document, with all the activities, who’s working on what, what’s missing” (A3).

Is your source code reviewed?

Apprentices said their source code was reviewed by senior developers before being merged into production, “I pull from master everyday just to get the latest updates and whenever I finish something I push to my own branch which normally the senior developer checks” (A3), while one apprentice had not begun any software development at this stage, another participated in pair programming for all development, citing that as an organisational practice to encourage constant

code review by the navigator while the driver controls the keyboard writing code. “We don’t do review afterwards, because they say that if we’re doing pair programming, then there should be constant review happening” (A1).

Do you have style guides for the code you write?

Code writing conventions were generally communicated verbally, with some instances of documentation on internal wikis detailing information on how to write code in a standardised manner shared by everyone in the team. “They do have a Confluence page with all that information up, but mostly to be honest it’s through talking with the other devs” (A1). Experienced team members preference for minimal comments, allowing tests to describe functionality and shared naming conventions were communicated to apprentices, alongside practically demonstrating writing code in a clear and accessible manner.

4.4.2 Employers

How are apprentices introduced to the codebase?

Employers all stated they aim for apprentices to be introduced to their codebase gradually and by a senior member of the team as a means of familiarising them: “it’s the mentor and their manager that introduces them to the system that they’re working on, the codebase they’re working with and what their job actually is in terms of what they do” (E1). Attention was given to explaining the overall product or service, its core functionality, the organisational purpose it serves and the technologies used to build it, being given increasing responsibility to make changes to it as they progress in their knowledge and skills. “Gradually being given more and more responsibility for the programming of activities in our product and that’s an ongoing situation, more for their skills development” (E2). An emphasis was placed on minimising feelings of being overwhelmed, by initial interactions with the codebase designed to be as small and simple as possible, such as fixing minor bugs, while still allowing the apprentice to get practical experience learning the software development environment and greater ecosystem of interacting software components. Interactions with the codebase are progressively developed over time to match and challenge the apprentices increasing abilities. “It’s tiny, tiny things that you can feel like you’re winning, nobody wants to come in and hit like a big massive issue, fixing a bug in spaghetti, so it’s very much all the effort is in learning the environment, the tools, all of that, and then you fix a typo and someone reviews it and you feel good because you’ve completed a lot of things and that goes on for as long as it takes, then when they start to feel comfortable, when they start asking can I get something harder, then you know” (E3).

Do you have pair programming as part of onboarding?

Apprentices at all the companies interviewed had some form of collaborative work experience built into their onboarding processes, with varying degrees of sophistication depending on the size of the organisation and the resources available. Where teams were larger, apprentices were embedded in multi-disciplinary teams, with roles such as scrum master, developer, system architect, product owner and business design lead, all supporting the apprentice with multiple perspectives on software system design and construc-

tion. Smaller teams replicated this with the apprentice being in close proximity to all members of the whole organisation and able to overhear conversations taking place outwith their direct role responsibilities, informing their technical and strategic understanding of the entire business.

Pair programming, with a more experienced member of the team leading the writing of code and verbally commenting on their thoughts and actions, while the apprentice observes and listens and is able to ask questions, take place throughout all the workplaces. “They’ll actually be doing something with somebody sitting with them talking them through it but they’re actually the person with hands on the keyboard” (E1). This also included working with remote developers through screen sharing. “It’s definitely the informal, if we think it would be good for one of the new guys to get exposed to something with someone a bit further up, and we do it even with senior levels, ‘watch me’, so it’s not necessarily the traditional pair programming” (E3). Employers encouraged pair programming to demonstrate agreed upon programming practices, such as clean and clear code, easing them into learning through imitation by recognising work patterns and code writing conventions, gradually progressing them to becoming contributing members of their teams.

4.5 Support Networks

4.5.1 Apprentices

Have you been assigned a mentor/buddy?

Mentors were described by apprentices as essential in helping them understand and navigate their roles as software engineers. Guidance from a more experienced team member that the apprentice could reach out to and be given advice greatly accelerated learning and social comfort in new working environments. Apprentices appreciated being coached by mentors, and given reasonable expectations for their work performance considering they were still learning software development. “My mentor was really good at saying we don’t expect you to know things yet so don’t be sitting there stressing” (A1) commented one apprentice. Buddies, complementary and in addition to mentors, provided apprentices with more direct and immediately accessible peer support for everyday matters, with a distinction made between both in terms of when to reach out to them. “First I would talk to the buddy and then the mentor if it’s something more difficult” (A2) an apprentice said.

Mentor relationships were not always positively structured with one apprentice (A4) stating that they did have a mentor but were in different teams which made regular contact and support difficult: “it’s easier now, but at the start and coming up to this project it’s not been as plain sailing as we probably would have hoped it’d have been, because she was showing me stuff to do with their team, which was specific to their team but it wasn’t specific to my team”. In general, apprentices said they felt supported by everyone in their team, with one apprentice saying they had two mentors and “both help me out with coding and how stuff works” (A3) with the support and socialising extending beyond work as “we’d grab lunch together and chat”.

What are your steps in solving problems?

Various problem solving strategies are used including reaching out to other team members during agile ceremonies such

as huddles which one apprentice explained as “if there’s something me or the other person is stuck with, we can just go to the tech huddle and just say we’re having issues with this, does anyone have any ideas?” (A1). Checking console logs and version control commits were other approaches, with an apprentice saying “I check the commits, go online, try and solve it myself, and then reach out” (A3). Another saying that they do not like to reach out until they have thoroughly tried to solve the problem themselves: “I don’t like reaching out until I’ve tried solving the problem myself. The team is very supportive and encourage me to ask them questions, they’ve said you’re here to learn so ask us anything” (A2). Information resources on company intranets to locate team specific documentation are also useful, “they put a bunch of resources for us, [documenting] how we do things” (A1) said one apprentice. Simply reaching out to the person sitting next to you, even if they are not in the same team was helpful in leading to a solution, “the guy that sits next to me, everything that I ask him isn’t always like, he can’t always solve it, so he’ll tell me the next best person to ask” (A4).

4.5.2 Employers

Do you have a mentor and/or buddy system in place?

Mentor and buddy systems are in place in all the organisations studied, with varying degrees of implementation. An employer at a smaller organisations said “there’s a couple of weeks where they spend quite some time together, just getting to know each other and talking about how to go about the work we do” (E1), adding that it would be done “on a needs basis, they’re not forced to do it”.

Larger organisations had more structured approaches to mentor systems with distinct roles, where both mentors and buddies are asked to nominate themselves for the role, as an employer explained “you can’t have that mentor relationship if you’re not wanting to do it” (E2). While all employers in the graduate apprenticeship programme are trained by The University of Glasgow, these mentors and buddies were also formally trained through a Scottish Mentoring Network Training partner. A more informal approach at larger organisations is taken towards training buddies, as these roles are more focused on day to day interactions and less about professional development. Things buddies may do are “making the apprentice feel comfortable” (E2) or information and knowledge sharing that would assist the apprentice since “they’re not coming out of university with a four-year degree and have had experience in internships and placements and stuff. They were coming brand new into software delivery and software engineering and a lot of them don’t have the basic skills” (E2). Essentially “making sure these coaches and buddies have that time to explain things to them and sit with them and work with them” (E2).

Sensitivity to the apprentices academic background with regards to pairing them with a suitable mentor was mentioned by one employer saying “the guy who we normally pair with, he didn’t actually go to Uni’ for Computing Science, he went for something else, so we thought it would be helpful if [the apprentice] had an academic mentor within the company with a CS degree” (E3).

4.6 Knowledge and Training

4.6.1 Apprentices

How would you describe the learning curve required for your job?

Apprentices all said they faced challenges when entering the workplace related to a lack of practical software development experience and an unfamiliarity with technical and business terminology and working practices. “The learning curve is huge” (A1) and could at times with advanced technologies be “brutal” said one apprentice. The teaching apprentices received during their university sessions while in some cases did not make apparent sense immediately, conceptualised later on when connected with working activities. “A lot of the stuff you would get taught in class, ‘oh when would you actually use that, like in a real life situation’, and then coming into work and you actually see it getting used and you’re like ‘that’s actually handy’” (A1). Also, resources sent out by university course lecturers via online learning platforms and email were “really helpful” (A1) along with team members sharing links and suggestions to find out more about an unfamiliar technology or topic.

Topics that were not covered during university teaching were encountered in the workplace, such as functional programming, with the apprentice saying it did not apply to other apprentices experiences and so would not have made sense to include in the courses, but it was an area where they had to self-learn. Another area was web technologies which had not been taught in university until after the apprentice had already learnt it in the workplace: “the company I was working in has a web based product and we didn’t do web technologies in Uni’ till the second block by which time I had already learnt those. It would have been helpful for me to have had that earlier” (A2). Other courses such as “software engineering practices were helpful in orienting myself around terminology and Agile ceremonies” (A2). One apprentice did feel there was not enough emphasis in teaching for working in smaller organisations, with the focus predominantly on “big companies, with teams of five or more people” (A3).

Lack of practical experience with programming at work made one apprentice feel other apprentices with more development focused teams “progressed a lot more than I have” (A4). Participating and getting involved in standups and meetings “seemed like, this is useless because I don’t understand what’s going on” (A1) but was actually beneficial as “hearing terms over and over in similar discussions you kind of start recognising patterns to make other things make a bit more sense. I’ve got a better understanding just from hearing people” (A1).

Transparency and open communication with team members assisted with knowledge and skill acquisition, “I say to anyone that I’m pairing with, if I don’t understand something just explain it to me like I’m five, like I’m brand new again, because sometimes you start to feel I’m starting to get it a little bit and then they’ll just introduce something new, and I’m totally floored by it” (A1). Unfamiliar technical or business terminology was also an issue and asking for clarification supported apprentices growing vocabulary, “I need to say to them this is jargon to me, I’m not going to understand that” (A1) and team members will clarify and explain those terms to increase the apprentices knowledge, “it starts to make a clearer picture, it kind of starts to make more and more sense, everyday it gets a little bit more easier”.

4.6.2 Employers

How do you aim to integrate new employees into the organisation so that they can contribute quickly and efficiently?

Being mindful of gradually leading the apprentice to become a contributing member of their organisation, an employer said “you can’t expect them to be as useful as somebody who’s been here for two or three years” (E1) and that a personalised, intuitive approach was better suited to properly meet both the apprentices and organisations needs. Onboarding and integrating is “not done in a step by step way, it’s not a checklist, and a lot of it is to some extent an intuitive thing” (E1). Adding “we all know each other very well, we’re in very close proximity, so we work closely together, we do pick up things quite quickly, if things are not right. We have a team conference call once a week and that’s when, and obviously we get together as well, but that’s when we can sort out issues and chat about what we’re doing, raise any other issues and that’s when we get a feel for where people are”.

“Giving apprentices as many tools as quickly as possible and letting them absorb that information” (E2) was the approach of one employer. Accelerated absorption of new information was encouraged through engagement with other team members through standups, kanbans, pair programming and other interactive activities. This overloading technique allowed apprentices to then make the connections from theoretical concepts while at university to practical experiential knowledge gained in the workplace. This approach, alternating between the workplace and the learning environment strikes a balance to help get apprentices up to speed quickly, and it shows in their productivity according to the employer: “getting that real eight weeks of solid just education piece and learning was massive, massively different to them in terms of how productive they are now” (E2). Recognising there are gaps in the apprentices knowledge and skills, one employer said filling in those gaps early was a key priority by getting the apprentice learning and using the main technologies in their software development stack and then progressing the apprentice from there. The key was giving the apprentice self-confidence to “make them feel I can go out and do this, the next job that comes up, they’re less reliant on you” (E3).

Is there a resource that apprentices can easily access to help solve problems and answer questions before asking someone?

Various resources are made available to apprentices to access information and knowledge on their own, before reaching out to other team members, such as references for “general information and broader wider learning, documentation, along with how do you learn a new language and how do you go about keeping a codebase tidy” (E1). Access to online learning platforms such as PluralSight and company intranets were made available to apprentices. “On the intranet they go in and record their holidays and absences and put expenses in and all that stuff. There are structure charts if you want to find somebody in the organisation and find their phone number, you would search for them in a structure chart, there should be a picture of them” (E2). This resource also made available policy information, along with a daily newsletter to update employees on company-wide

initiatives and relevant news. “We’ve also got a communications email that comes out daily” (E2).

Smaller companies made information available in various resources the apprentice could access, combining information available on company specific networks and also on external cloud based services. Part of onboarding was informing apprentices on how to find this information, “knowing where to look for stuff and knowing when to ask ” (E3). The employer added they recognised there was room for improvement there and that they were always open to being asked if anything was unclear. “It’s fair to say we’re not great at it, the goal is to have it, but you know there’s always legacy stuff and it’s always the last thing you want to update, but we have a very open door policy to asking questions. Everyone is on Slack so you don’t have to interrupt, you can always just fire it off and someone will pick it up later” (E3).

4.7 Socialisation and Culture

4.7.1 Apprentices

What socialisation activities have helped you integrate into the organisation?

Apprentices spoke about being able to easily talk to team members as and when needed and having lunch together to help build working bonds. “The team are all really friendly, during lunch we’ll just sit and talk, it’s quite a close team” (A1). Face-to-face communication was sometimes preferred over digital, “if there’s an issue, rather than sending emails, they’ll just go over to one another”. This ease of communication was not expected in cases, “it’s quite easy to socialise with them, didn’t expect that actually, thought they’d all be sitting on their own” (A1). Emphasis was placed on the importance of maintaining good relations with team members, especially in small organisations by one apprentice as “you need to be tight-knit” (A3). The role of humour and lighthearted interactions was also highlighted, “we do joke about a bit” (A3). Making an effort to participate in team nights out, “I’ve been to all the nights out” (A4), was seen as a way to help encourage and facilitate working relationships, as the same apprentice said “I think it’s helped, because instead of just walking past someone, you walk past with a smile, it’s made me more pals”.

Do you feel heard within the organisation?

The majority of apprentices said they felt visible and heard in terms of being depended upon and able to interact and relay feedback to peers and managers, within their organisations and teams. Apprentices in the same organisation, but in separate team environments led to differing responses here, with one apprentice saying “I find they are actively wanting this to work so they’re quite open to seeing what they can do to change. Same with the university, it’s really good at taking feedback as well, it’s been good” (A1) while another apprentice said that they did not feel heard, “at first I think it was because I was an apprentice, and because I don’t think the team had anything for me to do” (A4).

4.7.2 Employers

How is communication promoted within the company?

As an introduction and to promote team cohesion, one employer encouraged apprentices to meet everyone in the company, “I actively prod everyone, I’ll just say everyone should have an hour face time with each other” (E3), and the topics of discussion most often were not directly work related, “literally chat about everything, from code to dogs whatever you want to talk about”. These conversations helped in building stronger connections between team members, “it was kind of linking bits of the company together so that they had that relationship” (E3). Overall, these informal exchanges had a direct benefit on the business for the employer in terms of productivity, “everybody needs to understand each others strengths and weaknesses, it’s the same as any team right, you have to know” (E3).

How much of onboarding is face-to-face vs digital?

For most companies, onboarding before the first working day was mainly digital, with a switch over to face-to-face once the apprentice had started being involved in the work environment collaborating with other team members. Employers said “before they start in the business it’s all pretty digital” (E1) and “from the first day forward, it’s all face-to-face” (E3).

4.8 Onboarding Evaluation

4.8.1 Apprentices

What do you think can be improved for other apprentices being onboarded here?

More information on the teams before they eventually joined them was a suggestion to improve the overall onboarding experience. “I think maybe a brief explanation of the team you’re going into, like what they do” (A1). “I think giving them a little more prep to say this is the team you’re going into, this is maybe the kind of service that they work with, would have been good. I mean even just to calm your nerves a bit” (A1). “I think making sure you’re getting put into a team where there’s a lot more code to see, so a couple of the guys it’s like I think they only deal with a little bit of SQL and stuff. I think we get an advantage, getting to see people programming in Java and using design patterns, as opposed to just doing queries” (A1). This was echoed by another apprentice saying to “make sure there’s an actively developing team the apprentice developer is going into” (A4). Together with team information and apprentice-team fit, advice on technologies being used within those teams was proposed as a way of allowing apprentices to plan in advance and feel prepared. “Maybe telling you what languages” (A1) were being used, so that apprentices could “at least have a wee look and get yourself a little bit familiar with what you’re going to be looking at” (A1).

Having codebase introductions be more structured was another suggestion by an apprentice, saying “the introduction to the codebase could have been more structured, as it was, I was told where to access it and then I had to explore it myself”. While they felt more confident in navigating the codebase now, the first contact was not pleasant according to them. “I understand it now but that initial period was very disorientating, trying to understand the big picture of everything on my own” (A2). The apprentice said in order to

avoid newcomers having to experience that, they were creating information resources others could access and which they also found helpful themselves as references: “there’s a wiki I’m writing now for future employees to understand the issues that I faced, I’ve found that helpful in documenting what I know as a personal teaching aid” (A2).

4.8.2 Employers

Can the onboarding here be improved?

The partnership between companies and the university, both being platforms for the development of the apprentices, was an area where there was room for improvement according to one employer. “It’s working really well, I think the two sides support each other” (E1) but that there was need for more of an understanding of the different working environments apprentices find themselves, in particular those for smaller organisations and how they require teaching that is more suited to their particular opportunities and challenges. “It’s that level of [them] understanding the environment [the apprentice is] working in. The size [of the organisation] and the culture and the reality of the difference between a startup and a bank. There’s a huge variety of working environments for software developers” (E1). A particular emphasis in the curriculum towards soft-skills was also recommended: “time management, project management, soft skills and fitting into company culture” (E1), were all very important skills for apprentices to learn as they were “coming straight out of school” (E1) with no previous work experience with which to relate in their new environments.

More training for managers with regards to dealing with young people was a priority in improving the onboarding for one employer, “I think the thing we’re missing is more training for managers on how to deal with young people” (E2). Soft skills were again mentioned as an area that could be further enhanced “to teach managers how to deal with young people, what their mindset looks like, how they will be different from somebody coming into the business with years of experience with them” (E2). Nurturing a professional affinity between managers and apprentices was essential as “a sixteen year old coming straight out of school in their first job having no idea what to expect, no idea what to wear, are probably too shy to introduce themselves, so will hang about waiting for some face they know or might recognise or somebody to approach them. So its just about understanding the mindset of a young person” (E2). “That could really help us in the onboarding process”.

For smaller organisations, there were resource constraints in being able to improve the onboarding process with an employer citing an example where the apprentice was unsure about certain company policies. “I didn’t like the fact that was [a thing] and it shouldn’t have been a concern for her. That’s an area we need to do better on, how we do that I think, it’s probably to do with the size we’re at. I think when we’re a bit bigger we’ll essentially have someone who is HR, so we’re kind of in this gap” (E3).

What are your metrics to determine the success of onboarding apprentices?

Relying on intuition and getting a sense of whether the apprentice was fitting into team culture and was able to be productive were key metrics for smaller companies. “I do measure pretty regularly what’s going on, what they’re com-

pleting, what they're working on", said one employer adding "it's just little things which make you aware that an employee is committed to what they're doing or is taking liberties. It's a character thing" (E1). Having a conversation with other managers to discuss the apprentices performance would be initiated with a "small chat, basically how are they getting on, 'I think they're doing well', 'I think they're doing badly', 'why?'. So it's all that gut feel type, and it's probably to do with the speed at which they are progressing, with which they're closing the issues" (E3). How inquisitive the apprentice is also factored into assessing how well they were managing in their role: "how many questions are they asking, if they're silent, that's bad, if they're asking a question every minute that's also bad because they're not going off learning themselves" (E3).

More formalised methods of assessing the onboarding process were in place within a larger organisation such as monitoring retention statistics and conducting evaluation surveys with apprentices. The evaluation survey feedback "was mostly positive, there were a couple of tiny things that people were like you could do this better, which is great because you don't want everything to be perfect all the time. We want something to strive towards" (E2). Apprentices were given the opportunity to complete the survey anonymously, encouraging sharing honest thoughts and feelings on their experiences.

Does onboarding support apprentice satisfaction and retention?

Creating a satisfying and effective onboarding experience was beneficial for both apprentices and their employers, especially for companies who are in a competitive software engineering acquisition market. "We're competing with financial institutions so we're competing with all these big software houses and we struggle because we can't pay the same salaries that these big software houses can, so we have to think about what other ways can we entice staff to come work for us" (E2). Recruitment and retention were primary incentives for an efficient onboarding process, with one employer saying onboarding "is all about the recruitment but also the retention of staff as well". Conscious of the unique needs apprentices have coming into the workplace for the first time, an employer said "they're straight out of school so it's about giving them the tools they need to survive, and retain them past the end of that apprenticeship program" (E2).

Being empathic to apprentices was a key factor in creating onboarding processes that were satisfying and effective for them while simultaneously achieving organisational aims and objectives. "I think empathy is absolutely critical, you can't maintain a cohesive team if you can't recognise, like if somebody is not performing 100%, and you pull them in and go 'you're not performing well', I mean chances are they've got something going on in their personal life. So you have to be able to detect that, there's no reason why a normally motivated person has decided to just be not very good today. If you don't have empathy to your personality, or even if you just haven't learnt it, you're going to treat that problem with an absolute sledge-hammer. Really it's more like, 'do you want to take the afternoon off?'" (E3).

5. DISCUSSION

In understanding the results, thematic analysis techniques

[25][26] have been applied to obtain the essential characteristics of participants responses in relation to comprehending apprentices onboarding experiences and in light of research literature. As such, we identify, analyse and interpret patterns of meanings within the qualitative data, consolidating these into conceptual clusters for discussion. Further observations are presented as reflections for consideration in future interventions within onboarding processes for software engineering apprentices and students.

Defining Onboarding

In asking employers how they define onboarding, the primary themes that emerge are transitioning the apprentice from a state of unawareness of the organisation, its people, products, technology, processes and culture, to a state of experiential knowledge enabling them to participate in work activities, able to contribute to organisational aims and objectives.[14] Employers further understood onboarding to mean that the apprentice, and those around them, should understand as part of the onboarding process, their purpose within the organisation, and that they should be able to fulfil that purpose effectively to their own and the organisations satisfaction. The initiation of onboarding was consistently as early as recruitment, where expectations would be set, thereafter the interview, administrative tasks would be conducted before the apprentices first day. [11] Duration of onboarding varied from weeks, to months, dependent on the apprentice being onboarded and in view of their relative inexperience in software engineering. The importance of onboarding was seen in being a method of attracting, recruiting and retaining software engineering talent, and a means of establishing the foundation of engagement between the apprentice and the organisation. In smaller organisations, all members were involved to some degree in the onboarding of the apprentice, whereas in larger organisations, human resource personnel and the immediate team participated in onboarding. The method of performing onboarding was through a combination of intuition, adapting to the apprentices needs, and by the use of checklists and software that allowed standardised and qualitative measures to inform progress. [27] In general, onboarding consisted of personalisation predominantly for smaller organisations and a combination for larger organisations.

- Onboarding should be a mechanism by which the apprentice understands their purpose within the organisation and is enabled to best achieve that.

Preboarding

Apprentices came to know about the aims, culture and history of their organisations through official induction days, presentations and personal research through social media and the internet, pursuing the organisations website to inform them before the interview and their first day. Involving apprentices towards the background, aims and objectives of the organisation can have an effect on their engagement and connection during onboarding and after. [9] Administrative tasks were predominantly carried about between the interview and the first day through digital means, with any remaining work completed face-to-face within the first few weeks. Completing administrative tasks efficiently presents an easy process improvement that can offer important returns in the onboarding experience for apprentices. [13]

- Engage apprentices towards the aims and objectives of the organisation, involving them in the overall vision to create an emotional connection that cultivates commitment.

Orientation and Initiation

The first day at work was overall an uncertain and intimidating experience for apprentices, grappling with perceived notions of employer expectations and nervousness in being able to fulfill them adequately. Most organisations welcomed apprentices making an effort to put them at ease. For one apprentice this was not the case, but in general all apprentices felt welcomed on their first day. There was an awareness of gender diversity by female apprentices, which was relaxed once they joined their teams. Making an effort to put apprentices at ease on their first day impacts their perception regarding the organisations interest in them, their growth and success. [9]

Setting up workstations and software tools was mostly left to the apprentice to complete, and this resulted in an early sense of accomplishment. [28] Job role expectations took time to be fully understood, due to professional inexperience, with support from mentors during this period being described as crucial to acclimatise properly to the work environment and its responsibilities. [16] First tasks were deliberately simple, mindful of the apprentices inexperience and designed to expose them to the complete software development cycle. Assistance from mentors was essential for apprentices to successfully navigate unfamiliar work practices and domain knowledge. [29] Feelings of being overwhelmed, autonomy, excitement and a recognition of the need to self-manage themselves were experienced by apprentices after their first week. Employers described their onboarding programmes as a combination of unstructured and semi-structured, with the apprentice's needs informing the unstructured component of the process. Being sensitive to apprentices psychological states was essential to make the first day as satisfying for the apprentice as possible by employers. [30]

- Apprentices should be met by a member of their team on their first day, who is enthusiastic to introduce them to the organisation and initiate them towards meaningful work.

Software Development

Codebases were introduced to apprentices through guided walkthroughs with assistance from mentors and in combination with self-exploration. Employers aim for these walkthroughs to be gradual, with increasing amount of control and responsibility given to apprentices over time. [24] Methods of producing work individually and in teams is performed using various agile methodologies including scrum, standups, sprints, backlogs, anonymous voting, kanbans, pair programming, retrospectives and the use of user stories to understand, prioritise and accomplish work, [31] with in-person collaboration along with digital communication used in parallel to facilitate this. Code quality is maintained through review by a more experienced member of the team either after the code is written or while it is being written during pair-programming sessions. Pair programming is seen by employers as a way for apprentices to learn through imitation. Adherence to standardised conventions of code

writing is promoted through agreed upon documented standards, but mostly is enforced through verbal commenting during or after the code is written.

- Apprentices should be encouraged to participate early on in activities that enable work, such as agile ceremonies.

Support Networks

Supporting apprentices through mentor and buddy systems all contributed towards accelerated understanding and competence for apprentices in their new working environments. Mentoring can have a significantly positive impact on the apprentice alleviating feelings of isolation, reducing learning curves, allowing them to focus on appropriate activities and reduce time to productivity. [23] Problem solving strategies for apprentices included starting with self-reliance, progressing to searching internal and then online documentation, and as a last resort, reaching out to buddies and mentors. Supporting apprentices to be empowered by making documentation and knowledge resources easily available can avoid challenges in finding the relevant information to complete tasks. [19]

- To develop a productive relationship between mentor and apprentice, both should ideally be in the same team, with the mentor receiving appropriate training.

Knowledge and Training

Unfamiliarity with professional experience in software engineering resulted in apprentices describing the initial learning curve in the workplace as intense, but where they were able to connect theoretical concepts to practical implementations. Some topics faced in the workplace were not covered in their academic learning, where support from team members and self-learning filled in areas of weakness. For some apprentices lack of programming experience made integration into their teams difficult, however through a combination of transparency and open communication with team members, knowledge and practical skill gaps were addressed quickly. Employers were conscious of apprentices inexperience and adapted onboarding processes to accommodate them through tools, training and support. [8] Additional resources are also made available to allow apprentices to be able to manage problems themselves before reaching out to others. Consequently, documentation need not benefit already existing members of the team for it to be useful for new employees, who have different information needs, especially if they are still members of higher learning environments and are making their first steps in a professional role. [24] Without continual investment in employees knowledge and skills companies risk losing their human resource competitive advantage, and also face a reduction in employee morale and motivation. To promote early and efficient exploration of work environments and codebases, documentation in the form of knowledge bases and information resources are vital to assist newcomers in accomplishing their first tasks, often designed to introduce them to the code and tools used while minimising the risks to the rest of the organisation from any potential negative or adverse results. [24]

- Giving apprentices access to organisational information and knowledge resources allows them to feel a sense of autonomy and competence in solving their own problems.

Socialisation and Culture

Through formal and informal socialisation activities apprentices are able to foster positive working relationships with colleagues which transfer over to increases in productivity and satisfaction at work. Employers encourage apprentices to meet other team members to build more cohesive teams. Without this critical emotional and personal identification bond, employees are at risk of leaving the organisation due to a sense of detachment, isolation and associated stress due to intrinsic human needs of belonging, positive interaction and acceptance not being adequately fulfilled. [9][11] The more accommodated employees become, the greater their commitment to organisational aims and objectives and therefore by extension their effort in pursuing those shared values and goals. [16] Communication being a significant factor in assessing the ability of employees to proactively voice concerns, freely speak their minds in creative engagements and challenge the existing status-quo, in-turn help employees feel heard and understood and are particularly important in creating healthy and successful engineering cultures. [32].

- Due to initial feelings of intimidation in a new environment, apprentices may need particularly encouraged to speak up early if they experience problems in the workplace.

Onboarding Evaluation

A number of suggestions were made by apprentices to continually improve the onboarding processes at their companies, these included: more information on the teams they would be joining, inclusion within teams that were actively doing development work and were available to support the apprentice, and more structured introductions to codebases. Employers cited a desire for more collaboration between the university and smaller organisations to shape the curriculum towards being suitable for a variety of software engineering workplace environments. Other areas employers felt their onboarding processes could be improved were more training for managers to understand the mindsets of younger people. Metrics used by employers to determine the success of onboarding were intuitively sensing how the apprentice was progressing, monitoring retention rates and assessing feedback from evaluation surveys. The major advantage employers seek from onboarding programmes are in recruitment and retention of apprentices, asserting through the delivery of well designed onboarding they are able to gain competitively in attracting and retaining software engineering talent. Empathy was seen as a core factor in being able to create effective onboarding processes, especially for apprentices who were new to the workplace and need more support than experienced professionals to become productive members of their organisations.

- High-frequency communication channels between the apprentice, university and employer during the early phase of onboarding should be encouraged to elicit problem identification and management.

5.1 Study Limitations

A number of limitations are inherent in this research study. These include the data gathered being from a small sample size, within only one learning institution, conducted within a narrow time frame of apprentice experience and with no

follow-up to make more informed assessments. There are no interviews with the Graduate Apprenticeship course coordinators which prevents achieving a balanced opinion to better understand apprentice and employer responses. This study did not have access to university assessment data, exam and attendance results, which would have provided another perspective on the data gathered.

6. CONCLUSIONS

This research study attempts to understand the experience of software engineering graduate apprentices and their employers through the framework of onboarding processes as a means of transition from academia to industry. Participants in the study included software engineering apprentices and their employers, who were interviewed to determine their experiences of onboarding. Overall apprentices stated having positive opinions in the way they were welcomed and integrated into their organisations with further suggestions on how to improve the onboarding processes at their organisations. There were instances in which the apprentice was not properly integrated into organisational activities initially, and further study is required to determine the root cause of those situations. Apprentices and employers are enthusiastic to further improve the process by which apprentices are integrated into the workplace and see onboarding as a critical element in making the transition from academia to industry more effective.

Future Work

To improve on this study, future work will include broadening the pool of participants from apprentices, to computing science and software engineering students and graduates, and moving from one institution to multiple institutes of higher learning. Regular assessments will be undertaken to evaluate experiences more accurately over a longer time frame. Implementing intervention strategies, particularly where participants suggested improvements to onboarding, will be proposed and monitored to measure their effectiveness. A request for more data sources from both the university and employers will allow for several qualitative and quantitative perspectives in addition to interview data, enabling a richer understanding of current onboarding processes. Shadowing within the workplace will provide a fine-grained understanding of workplace realities and generate a more accurate data corpus on which to create a model of onboarding usable by employers.

Acknowledgments

This research would not have been possible without the counsel, guidance and support of Tim Storer, Quintin Cutts, Matthew Barr, Anna Doyle and the participation of the software engineering apprentices and their employers, all to whom are extended sincere gratitude and appreciation.

7. REFERENCES

- [1] Nurul Ezza Asyikin Mohamed Almi, Najwa Abdul Rahman, Durkadavi Purusothaman, Shahida Sulaiman. Software engineering education: The gap between industry's requirements and graduates' readiness. 2011.

- [2] Joseph Maguire, Steve Draper, Quintin Cutts. What Do We Do When We Teach Software Engineering? *UKICER: Proceedings of the 1st UK & Ireland Computing Education Research Conference*, 2019.
- [3] Matthew Barr, Jack Parkinson. Developing a Work-based Software Engineering Degree in Collaboration with Industry. 2019.
- [4] Ray Dawson. Twenty dirty tricks to train software engineers. *ICSE '00: Proceedings of the 22nd international conference on Software engineering*, pages 209–218, 2000.
- [5] Cooper-Thomas, Anderson, Cash. Investigating organizational socialization: a fresh look at newcomer adjustment strategies. *Personnel Review*, 41(1):41–55, 2012.
- [6] Maggie Johnson, Max Senges. Learning to be a programmer in a complex organization: A case study on practice-based learning during the onboarding process at Google. *Journal of Workplace Learning*, 22(3):180–194, 2010.
- [7] Shanika Karunasekera, Kunal Bedse. Preparing Software Engineering Graduates for an Industry Career. 2007.
- [8] MAUT-LS Professional Issues Committee. Report on onboarding and orientation for librarians and other library staff. 2014.
- [9] Errin Staunton. *An exploratory study of employees' perspectives on the value of onboarding programs*. PhD thesis, Capella University, 2017.
- [10] Amanda M Meyer and Lynn K Bartels. The impact of onboarding levels on perceived utility, organizational commitment, organizational support, andc. *Journal of Organizational Psychology*, 17(5), 2017.
- [11] Kaylee L Pike. New employee onboarding programs and person-organization fit: An examination of socialization tactics. 2014.
- [12] Hannah McKelvey and Jacqueline Frank. Improving onboarding with employee experience journey mapping: A fresh take on a traditional ux technique. *Weave: Journal of Library User Experience*, 1(9), 2018.
- [13] Alice Snell. Researching onboarding best practice: Using research to connect onboarding processes with employee satisfaction. *Strategic HR Review*, 2006.
- [14] Igor Steinmacher, Marco Aurélio Gerosa, and David Redmiles. Attracting, onboarding, and retaining newcomer developers in open source software projects. In *Workshop on Global Software Development in a CSCW Perspective*, 2014.
- [15] Rebecca Yates, Norah Power, and Jim Buckley. Characterizing the transfer of program comprehension in onboarding: an information-push perspective. *Empirical Software Engineering*, 25(1):940–995, 2020.
- [16] Talya N Bauer and Stephen G Green. Effect of newcomer involvement in work-related activities: a longitudinal study of socialization. *Journal of applied psychology*, 79(2):211, 1994.
- [17] Jolie O Graybill, Maria Taasil Hudson Carpenter, Jerome Offord, Mary Piorun, and Gary Shaffer. Employee onboarding: Identification of best practices in acrl libraries. *Library Management*, 2013.
- [18] Rebecca Yates. Studying onboarding to improve program comprehension tool support. In *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*, pages 257–258. IEEE, 2010.
- [19] Jianguo Wang. Supporting developer-onboarding with enhanced resource finding and visual exploration. 2012.
- [20] Howard J Klein and Natasha A Weaver. The effectiveness of an organizational-level orientation training program in the socialization of new hires. *Personnel Psychology*, 53(1):47–66, 2000.
- [21] Howard J Klein, Jinyan Fan, and Kristopher J Preacher. The effects of early socialization experiences on content mastery and outcomes: A mediational approach. *Journal of Vocational Behavior*, 68(1):96–115, 2006.
- [22] World Health Organisation. Statement on the second meeting of the international health regulations (2005) emergency committee regarding the outbreak of novel coronavirus (2019-ncov).
- [23] Fabian Fagerholm, Alejandro Sanchez Guinea, Jay Borenstein, and Jürgen Münch. Onboarding in open source projects. *IEEE Software*, 31(6):54–61, 2014.
- [24] Rebecca Yolande Yates. Onboarding in software engineering. 2014.
- [25] Virginia Braun and Victoria Clarke. Thematic analysis. 2012.
- [26] Moira Maguire and Brid Delahunt. Doing a thematic analysis: A practical, step-by-step guide for learning and teaching scholars. *AISHE-J: The All Ireland Journal of Teaching and Learning in Higher Education*, 9(3), 2017.
- [27] Daniel M Cable, Francesca Gino, and Bradley R Staats. Reinventing employee onboarding. *MIT Sloan Management Review*, 54(3):23, 2013.
- [28] Adriaan Labuschagne and Reid Holmes. Do onboarding programs work? In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 381–385. IEEE, 2015.
- [29] Cheri Ostroff and Steve WJ Kozlowski. The role of mentoring in the information gathering processes of newcomers during early organizational socialization. *Journal of Vocational behavior*, 42(2):170–183, 1993.
- [30] Talya N Bauer and Berrin Erdogan. Organizational socialization: The effective onboarding of new employees. 2011.
- [31] Jamie A Gruman, Alan M Saks, and David I Zweig. Organizational socialization tactics and newcomer proactive behaviors: An integrative study. *Journal of vocational behavior*, 69(1):90–104, 2006.
- [32] Raphael Pham. Improving the software testing skills of novices during onboarding through social transparency. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pages 803–806, 2014.