Improving Student Knowledge Retention For Android Mini-Bootcamps

A Ogborn

MRes Science and Technology

Bournemouth University

November 2019

Abstract

Bootcamp learning is growing in popularity across a range of software development fields - bootcamps teach a specific skill set intensely over a short period of time. Studies have determined that knowledge retention is an issue with this learning format, and retention is poor when tested thirty days post-bootcamp. Whilst skills development during bootcamps is rapid, bootcamps lack the opportunity for the proper application of these skills to enable effective knowledge retention.

Android mini-bootcamps are an understudied subset of the bootcamp format, where the Android app development learning period is restricted to one to two days. This study determined that knowledge retention improved, when students engaged with a specific task, informed by the feature driven software development methodology. This study's findings have clear implications for bootcamp providers interested in the longevity of their skills enhancement delivered within their bootcamp format. A post-bootcamp 'phase' is needed, with a targeted task to apply skills/knowledge gained during the actual bootcamp. Furthermore, this study found that those with existing programming experience demonstrated poorer knowledge retention.

This study provides evidence of the knowledge retention issue with Android mini-bootcamps and how the knowledge retention issue can be addressed. The study also recommends investigating links between post-bootcamp tasks and their impact on retaining specific skills/knowledge.

Contents

List of figures

List of tables

Abbreviations and specialist terms

A Level – Advanced level

API – Application Program Interface

BU – Bournemouth University

GCSE – General Certificate in Secondary Education

IDE – Integrated Development Environment

FDD – Feature driven development

MOOC – Massive Open Online Content

SD – Standard Deviation

SQL – Structured Query Language

UI – User Interface

UX – User Experience

1.0 Introduction

Bootcamp learning is becoming increasing prevalent within the software development industry, offering an alternative format to traditional learning programmes enabling the student to gain a narrowed, targeted skillset (Thayer & Ko 2017)). This study focuses on the specific discipline of Android app development, taught within the smaller mini-bootcamp format. A mini-bootcamp is an intense period of development across a period of one to two days, where no pre-requisite programming knowledge is required, nor promises given of employment post graduation by the bootcamp provider (International Telecommunication Union, 2016; London App Brewery, 2018a). Notable benefits and motivations for attending a bootcamp are numerous. The bootcamp learning format enables a confident and effective skill or knowledge enhancement. Skills are developed rapidly over a short space of time and act as good precursors to longer training programmes (Heskin et al. 2015; Thayer & Ko 2017; Weis et al. 2018; Waguespack et al. 2018). The International Telecommunication Union's (2016) report indicates a demand for Android bootcamps, due to growth in demand for Android skills development. It is useful to understand bootcamps' effectiveness in terms of the level of knowledge retained post-bootcamp. This will build on what is already known about the effectiveness of the skills enhancement gained during the bootcamp.

Despite the benefits of bootcamp learning, studies have found problems. Thayer & Ko (2017, p.245) indicate that they are a *"new and understudied way of training"*. Wilson (2018) suggests that skillsets deteriorate over time post-bootcamp, therefore skills need to be re-used to reduce the rate of deterioration. Similarly, Waguespack et al. (2018) indicates that whilst Android developers become effective during a bootcamp, students are not able to use skills reliably post-bootcamp. Furthermore, several studies within medical fields found significant knowledge retention issues, these were indicated with objective knowledge assessments after a period of time post-bootcamp (Moulton et al. 2006; Sonnadara et al. 2012). These medical bootcamps, which covered technical surgical skills, adopted a similar format to software development bootcamps and provide a useful insight into the knowledge retention issue associated with bootcamp learning. Knowledge retention is an issue which has received much criticism within bootcamp learning

Consequently while software development bootcamps are growing, is an increasing awareness of the knowledge retention issue associated with bootcamp learning and Android app development mini-bootcamps which have not previously been investigated. This study sets out to determine if knowledge retention could be improved in Android mini-bootcamps, with a simple intervention: Participants were asked to complete an additional coding task informed by Feature Driven Development (FDD) methodology (Hunt 2006) over a period of thirty days post-completion of a model Android mini-bootcamp. At the end of the thirty day period, an objective knowledge questionnaire (similar to the format used by Moulton et al. (2006) and Sonnadara et al. (2012)) was used to test improved knowledge retention,compared to a group that did not complete the task. Findings will have direct implications for both bootcamp providers and make valid contributions to the growing sphere of interest surrounding bootcamp learning within academia.

1.1 Hypothesis statement

The following problem and hypothesis statements set out the purpose of the study.

Problem: There is a knowledge retention issue associated with Android app development mini-bootcamp style learning within thirty days of bootcamp graduation.

Hypothesis: Completing an FDD assessed task over a period of thirty days post Android mini-bootcamp will yield improved knowledge retention, when compared to those that do not complete the FDD assessed task.

1.2 Aim and objectives

Aim: To determine if an FDD based assessed task completed post-bootcamp can improve knowledge retention for Android app development mini-bootcamps.

Objectives:

1. In order to identify commonalities across Android app development bootcamp curricula, complete an analysis of different Android bootcamps currently offered in the marketplace to produce a skills mapping
2. Based on the analysis and skills mapping completed in Objective 1 design and deliver a market representative model mini-bootcamp to a cohort of students (group A (control group & group B (experiemental group))
3. Informed by the analysis and skills mapping completed in Objective 1 design an assessed task for Group B to complete for a period of 30 days post-bootcamp
4. Informed by the analysis and skills mapping completed in Objective 1, design a knowledge based assessment questionairre for Groups A and B to complete 30 days post-bootcamp

5. Evaluate group B's performance in the FDD assess task and compare their performance to Group As in the knowledge questionnaire completed in Objective 4

2.0 Literature review

Thayer & Ko (2017) argue that the bootcamp learning format needs to be studied due to its growth in popularity. Only certain aspects of bootcamps have been studied previously e.g. logistics, demographics and employment success rates. There are varying motivations for students wanting to attend a bootcamp, including: skills enhancement, enhanced job prospects, pre-course skills preparation or fostering a general interest (Heskin et al. 2015; International Telecommunication Union, 2016; Weis et al. 2018). Bootcamps are generally exempt from robust validation and course design processes, normally associated with qualifications such as GCSEs, A Levels, Degree Programmes etc. (Waguespack et al. 2018). Typically a bootcamp student could be awarded a 'certificate of completion', with only an un-accredited endorsement by the bootcamp provider themselves (London App Brewery 2018a). Despite bootcamps' lack of accreditation, which comes under criticism from governments, employers and citizens, they are considered an effective learning format for a rapid targeted skills enhancement which is a contributing factor to their growth in popularity (Waguespack et al. 2018). Thus, bootcamp benefits are attractive to potential students and the providers themselves have free reign to design their own curriculum and produce their own certification.

Bootcamp learning is popular in two technical fields: software development and medicine and parallels can be drawn between both. Whilst this research targets bootcamp learning within the sphere of software development which is an under researched area in terms of bootcamp learning, much can be learned from literature concerning medical bootcamps and this is reviewed and cross referenced to draw out lessons affecting both fields. Empirical evidence in both areas will be examined to gain a more thorough insight into the bootcamp knowledge retention issue.

There have been several medical bootcamp learning studies which produce important conclusions regarding bootcamp issues, such as skills enhancement and knowledge retention. Weis et al. (2018) and Ceresnak et al. (2017) both comment on the achieved effectiveness of specialist skills enhancement and gained confidence across surgical skills and pediatric care bootcamps. In contrast, International Telecommunication Union, (2016) and Waguespack et al. (2018) draw on learning from software development bootcamps, which found bootcamp students become effective with a specialist skillset relevant to their respective industry. Generally, it can be seen that bootcamps within both fields focus on the development of highly specific technical skillsets. Fundamentally, however, these are a small selection of a wider group of studies and reports which have found clear benefits of bootcamp learning, which strengthens reasons why the bootcamp learning format is growing in prevalence across the globe. In many cases students demonstrate an enhanced skillset upon bootcamp completion. However, as numbers of bootcamps grow, greater knowledge is needed beyond what can be seen upon the point of bootcamp completion.

Through drawing parallels between different bootcamps across different disciplines, there are clear gaps in knowledge of bootcamp style learning. Due to the wide scope of 'bootcamp' learning taking many different formats, it was firstly important to define what a bootcamp is, and which particular bootcamp format this study focused on.

2.2 Defining a bootcamp

The International Telecommunication Union (2016) outlines four different types of software development bootcamp:

1. Ready to work bootcamp (12-24 weeks, entry requirements (e.g. programming skills tests and/or tasks)
2. Bootcamp+ (1-2 years, entry requirements)
3. Mini-bootcamp (1-2 days, little/no entry requirements)
4. Early education model (workshop, hackathon, coding skills in school)

For the purpose of this study, the mini-bootcamp format was investigated, typically where a bootcamp will last a period of one to two days. Within this context, it is important that issues associated with this format of bootcamps were investigated. However, to ensure sufficient depth and breadth of understanding, this research drew on information from other bootcamp formats such as the very popular 'ready to work bootcamp', which is largely targeted at those wanting to gain an entire work-ready skillset in a short space of time. The main difference between the 'ready to work bootcamp' and the mini-bootcamp are the duration and promises of employment (International Telecommunication Union 2016; London App Brewery 2018a). Regardless of the type of bootcamp, all bootcamp formats maintain the same blueprint features which include:

1. The bootcamp provider self determines the curriculum
2. Intense delivery over a short [but varied] period
3. generally no formal regulation by external bodies

Thayer & Ko (2017) indicate that bootcamps can be delivered in part time, full time, online, residential and face-to-face classroom formats. To ensure sufficient focus within this research only face-to-face formats were investigated. It could be argued that a parallel set of issues arise in both face to face and online formats. Online, distance and remote style learning introduce a complex set of variables, which would need to be investigated separately. Thus, the research, the design and problem were solely focused on solving issues within the remit of face-to-face classroom bootcamp formats. Further narrowing of the focus was also required, because bootcamps focus on a rapid skills enhancement of a specific skillset. Therefore a specific discipline and topic within face-to-face mini-bootcamp learning needed to be determined.

2.3 Focusing on Android app development

Thayer & Ko (2017) suggest that each bootcamp forms its own unique 'community of practice', which is explained by Wenger-Trayener (2015), as a common passion or concern between all students, such as a passion for a particular technology (e.g. Android app development) or desire to enter a certain profession (e.g. software developer or surgical medicine). Moreover, due to bootcamp time constraints, Waguespack et al. (2018) and Thayer & Ko (2017) indicate that bootcamps typically focus on a single technology/topic within a discipline (e.g. Javascript web development, IOS or Android app development). Thus, it is important to ensure that there is a clear subject focus when investigating mini-bootcamps. Mini-bootcamps focusing on the discipline of Android mobile app development form the boundary and focus discipline of this thesis. Any community of practice formed by students attending a mini-bootcamp will all have aligned interests in developing Android apps (Thayer & Ko 2017). Android app development has been targeted in particular, due to the global increase in the number of apps being downloaded and used. 178.1 billion were downloaded in 2017, with a forcasted to hit 258.2 in 2022 (Statista 2018). Of this growing demand, over 70% since 2017 has been dominated by the Android market (Statcounter 2018). The demand for Android app development skills is high and arguably means an increased demand for Android app development courses, such as Android app development mini-bootcamps (Forbes 2018). Whilst there is increasing demand for Android app development bootcamps, it is important to understand that entry requirements for mini-bootcamps and other bootcamps differ. It could be assumed by some, that the intense nature of bootcamps requires a certain level of pre-existing knowledge to access the curriculum material.

2.4 Bootcamp entry requirements and pre-requisites

Typically, there are no explicit pre-requisites or entry requirements for entry onto a mini-bootcamp, as indicated in International Telecommunication Union's (2016) report. This can be seen with UK based London App Brewery's two day Android Development Bootcamp, which appears open to all students who pay to secure a place (London App Brewery 2018a). London App Brewery (2018a) states: "Our weekend course assumes no prior programming knowledge and takes you through app development step-by-step. It could be assumed that students attend a mini-bootcamp in order to gain a specific skillset that they do not already possess, given the targeted and specific nature of all bootcamp learning. However, pre-requisites could filter out and deter applicants from attending mini-bootcamps, meaning a reduction of students and revenue for the respective bootcamp provider. This diverse range of student backgrounds can be seen within testimonials from London App Brewery's (2018a) Android mini-bootcamp students:

"well-run course with all the information provided to get started with mobile-coding" – Doctor

"Having always wanted to develop apps, but never 'getting round to it"; "Like a lot of people I have an idea for an app every other day." – Entrepreneurs

"As a complete beginner to the coding (let alone Swift), I now feel like I have a good grip of the basic building blocks." – Portfolio manager

However, whilst mini-bootcamps generally teach programming basics and fundamentals as part of their curriculum, there is no evidence of mini-bootcamps turning away prospective students if they do possess a technical skillset (e.g. programming learnt through work or other study). Ultimately, it would be down to the prospective student to determine if that bootcamp is appropriate for them. The lack of pre-requisites could have an impact on the level at which skills can be both gained and retained. It is assumed that 'general' programming skills would need to be taught in addition to any other specific skills.

In contrast, mini-bootcamps are smaller than the 'ready to work' bootcamps, which are 10 weeks or more in length. A 'ready to work' bootcamp provider promotes enhanced employment prospects upon post-bootcamp completion, for example, the Makers Academy's bootcamp promotes an employment guarantee and advertises employment statistics (International Telecommunication Union 2016; Makers Academy 2018). These types of bootcamp state a set of pre-requisite competencies or attributes which arguably will improve their employability statistics – a key marketing tool for an expensive course (circa £8000 for 12 week course), particularly when compared to the price of a Masters Degree. The Makers Academy's bootcamp requires students to have completed a range of online courses, engaged with a 4 week pre-course programme and complete an interview style paired programming task (Makers Academy, 2018). This task could typically involve applicants jointly coding a program, which is then assessed by interviews against a list of techniques or competencies. In comparison to mini-bootcamps, which do not make promises of employment, but solely advertise that the student will be able to make and deploy Android apps upon graduation (London App Brewery 2018a). Thus, further contributing to the notion that students have to self-determine if a mini-bootcamp is appropriate for their wants and/or needs. If the student wants or needs Android app development skills and does not already possess a programming skillset, they may consider a two day mini-bootcamp. If students possess those skills already, why would they want to attend?

2.5 Bootcamp issues

Waguespack et al. (2018) challenge the learning of software development skills in a bootcamp format claiming that bootcamps are single minded, focusing too narrowly on skills development vs. the important theorizing and problem solving skills required of a good software developer. Their paper shows bootcamps' time limited nature, results show a narrow focus on aims which largely focusing on repeated exposure to the same technology and tasks in order to gain a particular skill set. This did however; enable the student to become more effective at developing software. Waguespack et al. (2018) argues that problem solving and theorizing are required in bootcamps for students to be able to develop software reliably over time. Therefore by omitting problem solving and theorizing opportunities from bootcamps, the ability for the student to develop software reliably greatly diminishes. Additionally, Thayer & Ko (2017) suggest that it is important to apply skills through developing a portfolio or mini project post-bootcamp. Many students do not have time to complete such a project during the bootcamp itself raising the question "could a portfolio creation or similar task provide the framework for the problem solving and theorizing that Waguespack et al. (2018) argue be vital?"

Wilson (2018) supports the views of Heskin et al. (2015) regarding the effectiveness of bootcamp learning, both indicating that bootcamps provide a good opportunity to upskill students quickly at the start of a longer training process. However, Wilson (2018) argues that skills still need to be kept 'fresh' to ensure skillsets are developed and maintained. Both Wilson (2018) and Thayer & Ko (2017) point towards the need for an extended focus and development of skill application and development, post-bootcamp. Thus, skillsets could quickly become stale, if not applied and developed post-bootcamp. If skillsets become stale (e.g. due to lack of use), they will have deteriorated and would be less effective when the skills are recalled for a purpose (e.g. task). Wilson (2018) states that students need to stretch their current skills – indicating that skills within the bootcamp need to be used and consistently applied. Contrasting this with Waguespack et al's. (2018) paper, skills could be stretched through continued theorizing and problem solving post-bootcamp. It is therefore important to investigate how this could be achieved. Furthermore, Wilson (2018) claims that the programming language Java cannot be learnt over a short duration in order to be able to code at a competent level. This is mere speculation, as there is little research to indicate that Java cannot be taught to enable competency over a short period of time – positive testimonials from London App Brewery's (2018a) students indicate the contrary. Besides, mini-bootcamps do not advocate that students will graduate as 'competent' programmers, just that they will develop skills/knowledge. Ultimately, Java is the predominant language used to develop Android apps (Android, 2018), and it is important to take this into consideration when developing a solution to this problem. A period post-bootcamp could enable the longer period of time that Wilson (2018) claims is needed to effectively hone skills. For example, post-bootcamp challenges/activities, could be used as a tool to continue enhancement of skills over an appropriate period of time.

Waguespack et al. (2018, pg. 55) state that bootcamps lack credibility and deliver *"mixed up results"* due to their un-accredited nature. Accreditation is desired by a range of stakeholders including governments, industry and consumers. Conversely, whilst Waguespack et al. (2018) challenge bootcamps' lack of accreditation, they still argue that the repetitive nature does generate students who are effective with technology, which arguably is desired by a range of stakeholders, particularly industry. Fundamentally, whilst there are benefits and drawbacks to the intensive nature of bootcamps, Waguespack et al. (2018) importantly conclude that the full software development cycle needs to be enaged with, from the introduction to technology, to implementation and evaluation. They allude to the fact that this is not achievable within the time-limited nature of bootcamps and their repetitive 'drilling' style tasks. The time needs to be extended somehow, to enable the completion of the cycle and a change of task.

The intense time limited nature is essentially what defines a bootcamp and differentiates it from more traditional courses. A medical bootcamp study by Moulton et al. (2006) found that the massed practice (imparting skills intensely over a short period of time) approach that bootcamps typically adopt, had an explicit impact on knowledge retention. The study found that simply delivering four weeks worth of learning into one or two days en masse had a negative impact on knowledge retention. The study was robust in its conclusions, it compared two like-for-like groups studying the same content across two different formats. One group studied the content over four weeks and the other experienced the same in a condensed mini-bootcamp style format. Those students that experienced the distributed practice version (where content was taught over the longer period of four weeks), experienced better knowledge retention post-bootcamp. Thus, it could be argued that mini-bootcamps that solely focus on learning over a two day course incur a knowledge retention issue and arguably need to be extended in some way. Knowledge retention within this study was assessed through objective, test style questions at several points up to eighteen months after the mini-bootcamp. These objective-based tests explicitly linked to the objectives of the bootcamp. With knowledge retention becoming a recurring direct or indirect theme, it is important to review it in more detail.

2.6 Knowledge retention with bootcamp learning

The knowledge retention issue raised in Moulton's (2006) study is poignant. There are no explicit investigations into mini-bootcamp knowledge retention within the discipline of Android app development, however this study draws on inquiries which have been made within technical medical fields (e.g. Weis et al. 2018; Sonnadara et al. 2012). These findings were useful to draw on to gain a general understanding of knowledge retention within the general context of bootcamp learning. It is then considered how these issues will affect Android app development mini-bootcamps. It is seen that parallels do exist between the two fields - bootcamps across both software development and medical disciplines typically feature a single technical focus deliverd in the mini-bootcamp format.

Weis et al. (2018) are other authors who advocate the benefits of skills enhancement and confidence gained through bootcamp learning. However, their study found a skills digression over time and thus challenged bootcamps' durability. Weis et al. (2018) further argue that proficiency can be maintained if skills are followed up with regular and deliberate applied practice. There is a clear parallel here between medicine and software development bootcamps, when comparing this to Wilson's (2018) study concerning software development bootcamps, who also argues that skills need to be consistently applied post-bootcamp. Could consistent application of these techniques post-bootcamp improve knowledge retention? It can be seen, that whilst skills are developed quickly and effectively during the bootcamp, students need opportunities to apply the techniques they have learnt. The findings within Moulton's (2006) study explicitly found that practice distributed over a longer period of time, yielded better knowledge retention. Thus, mini-bootcamps currently possess a knowledge retention flaw if delivered as independent one-two day taught courses, where learning ceases at the end of the second day.

Bootcamps are extremely effective at developing a certain skillset, although it depends on how the knowledge is inculcated (Sonnadara et al. 2012). Waguespack et al's. (2012) paper goes further to argue that knowledge retention is very good when bootcamps are delivered in a residential format. There is only one example of a residential style software development bootcamp in the UK, which is promoted as "immersive", promoting *more* time on task and contact time with tutors (Wegotcoders 2018). Whilst this approach may overcome some of the challenges posed by bootcamps, including knowledge retention, features of residential bootcamps would not be easily or practically applied to mini-bootcamps. A residential solution to mini-bootcamps' knowledge retention issue would increase the price significantly and the residential component could prove a barrier to many students who will not want to relocate (Thayer & Ko 2017). However, it could be drawn from this, that *more* time on task and contact with tutors could improve knowledge retention, as indicated by Sonnadara et al's. (2012) study. This could be incorporated into a mini-bootcamp by targeting the 'post-bootcamp' component. When considering the issue of knowledge retention within a software development discipline, it is important to consider a software development methodology, there could be links between how software is made and how knowledge is retained (Charvat, 2003).

2.7 Feature driven development (FDD) methodology

Typically, common practices for creating software which are adopted in industry follow variations of the Agile, or Waterfall methods (Charvat 2003; Hunt 2006). FDD is a type of applied agile methodoglogy, it can be considered a version of the agile principles in practice, as adopted by software developers. Firstly, it is important to consider what agile development means from a non-software development perspective in order to determine a general understanding of its principles. Once FDD's principles are understood, they can then be applied to the context of Android mini-bootcamps. According to Davis (2013), the agile 'process' generally follows these steps:

1. A stakeholder (usually customer or internal manager) outlines a digestible, manageable goal, or vision. Essentially what *they* want to be made/done
2. A list of objective requirements is drawn up to meet this goal
    a. These are written very simply and descriptively, so they are easy to understand, at the desired level of granularity depending on the context
3. This list of requirements is reviewed
4. Each requirement is developed and tested to make sure it works
5. The original stakeholder is consulted to see if they are happy with the completed task and then Step 1 starts again and the next software requirement is completed.

Overall, this Agile approach could be applied to a range of processes inside or outside the sphere of software development.

Developing mobile apps at any scale, whether this involves a single app developed by a sole developer or large corporate projects involving multi-disciplinary teams in the modern environment, both require the developer to be able to adapt and work in a changing, complex environment (Hunt 2006). Whether the student of the Android app bootcamp intends to enter the software industry or not, it could be argued that a modern development methodology should inform an aspect of the bootcamp learning process. Hunt (2006, p. 161) continues by indicating that customers' wants and needs change, these wants and needs normally result in a growing set of features, thus, in order to react to this, developers could adopt a *"feature-centric process"*. FDD is a specific software development methodology which focuses on the development of features that specifically meet customer requirements (Ashmore & Runyan, 2015). FDD is a type of agile development methodology which encompasses a range of processes enabling software to be developed quickly and incrementally, based on a set of requirements (Puri 2009).

FDD is a short and iterative process, which enables the student to develop tangible Android features in progressive increments that increase in complexity (Goyal 2008). For example, once a developer develops one feature, they are able to progress onto the next, and so on. In the context of mini-bootcamps, this is an important point to consider, due to the significant time limited nature of the mini-bootcamp model. Furthermore, if a post-bootcamp period exists as a phase within a bootcamp's curriculum plan, this too will also be time limited in nature. Studies indicate that knowledge retention fades dramatically within thirty days post-bootcamp (Moulton, et al. 2006; Sonnadara et al. 2012). Longer, drawn out methodologies such as the waterfall methodology are time consuming, due to the heavy documentation process in the analysis and design stage of development (Charvat 2003).

The feature driven approach enables a required set of features to be developed within a piece of software, such as an Android app; the developers go through the entire software development process (development, testing, evaluation etc.) iteratively as each feature is developed (Charvat 2003; Goyal 2008). In the context of an Android mini-bootcamp, certain app features could be taught and learnt quickly if following this model. Furthermore, it is understood that all bootcamps are very narrow in their focus, targeting the learning of specific technical skillsets quickly (Waguespack et al. 2018). This is similar to the FDD methodology, which focuses on meeting specific technical requirement, quickly and effectively, then progressing onto the next requirement once completed. Thus, FDD methodology, is an appropriate methodology when considering how knowledge is retained with bootcamp learning, as there are clear parallels between how software is developed using FDD methodology and the nature of bootcamp learning. Overall, this needs to be considered when designing methodology to satisfy this study's hypothesis.

2.8 Bootcamp Learning Summary

Bootcamp learning has been praised for providing an effective format to learn a targeted skillset (Waguespack et al. 2018). This has been evident across both software development and medical fields. Overall, a range of issues with bootcamp learning have been identified, the vast majority focus on issues arising post-bootcamp (e.g. Heskin et al. 2015; Thayer & Ko. 2017; Wilson 2018). Knowledge retention has been raised independently, and other authors raise other problems which are arguably interlinked with knowledge retention.

Waguespack et al. (2018) criticise the lack of theorizing, problem solving and evaluation opportunities, however they do not provide a concrete process or approach to address these in a bootcamp format. There is no indication as to how students should engage in problem solving and specifically which skills should be used to solve said problems. Wilson (2018) indicates that skillsets need to be stretched and kept fresh, in line with Waguespack et al. (2018), he does not offer a concrete approach to stretch and maintain a 'fresh' skillset. Whilst Wilson's critique is interesting, it is merely an open assertion with no concrete solution.

Moulton et al. (2006) indicate that knowledge retention is improved when practice is 'distributed', i.e. spread over a longer period of time. Which arguably goes agains the crux of any type of bootcamp – its short, intense nature. However, rather than extending the face-to-face, taught component of the bootcamp, Moulton et al. indicates that learning experiences benefit from being 'spread' over a longer period of time, indicating that learning is a process, rather than an event. This can be compared to views from Ceresnak et al. (2017) who argue that bootcamps provide an effective pre-cursor to subsequent learning, but adversely affect progress and knowledge retention if delivered as stand alone independent methods of learning. Ultimately bootcamps are physically 'stand alone', however Ceresnak et al. (2017) raise an important point that the 'one off' bootcamp type of learning episode is not as effective in the longer term.

Finally, Moulton et al. (2006) and Sonnadara et al. (2012) found effective ways of determining knowledge post-bootcamp, which is fundamentally important in this study which aims to address knowledge retention. In order to address knowledge retention, the study needs to be able to reliably check knowledge retention after any solution or intervention has been put in place.

Ceresnak et al. (2017) used a series of objective based questions to determine the level of knowledge retention - this approach proved very reliable. The questions within the questionnaire/assessment were targeted at specific skills taught within the respective bootcamp. This is important because it explicitly measures the skills/knowledge that students gained within the bootcamp, whilst not measuring retention of other 'unrelated' skills. Furthermore, Moulton, et al. (2006) adopted a similar approach, adopting a set of structured assessments intermittently over a period of time, targeting specific knowledge and skills. In some cases the use of targeting questioning itself actually improved knowledge retention. This indicates a proven approach to determining knowledge retention and is an important consideration to inform the methodology of the study. Finally, Sonnadara, et al. (2012) further re-enforces the objective approach using a knowledge retention questionnaire, by indicating a time period of thirty days in order to judge whether knowledge has been retained. Whilst there is a confirmation that their studies indicated poor knowledge retention, both Sonnadara et al. (2012) and Moulton, et al. (2006) do not suggest how this thirty day period could be utilised in a structured manner in order to improve the knowledge retention questionnaire score.

Finally, FDD methodology has parallels with multiple aspects of Android mini-bootcamp learning. It is an iterative, short and successful process which could be considered when tackling the knowledge retention issues associated with the much criticised, failing post-bootcamp phase of Android mini-bootcamps (Charvat 2003; Goyal 2008).

3.0 Bootcamp learning model(s)

Following the literature review, two models were developed to illustrate both the learning process of a current Android mini-bootcamp. Figure 3.1 (below) mapped out the bootcamp learning process behind an existing bootcamp in the marketplace, from there it was injected with analysis from the literature review in order to determine a revised model (Figure 3.2), incorporating a proposed solution to the knowledge retention issue. This solution to the knowledge retention issue acted as a blueprint for the methodology of the study. London App Brewery's (2018a) Android mini-bootcamp was mapped into a simple bootcamp learning process model, this indicated the stages and steps within the bootcamp delivery. Figure 3.1 reflects the short, intense nature of a mini-bootcamp, starting with a student securing a place through providing payment, 2 days of intense face to face teaching, followed by optional post-bootcamp chat support.

3.1 Exemplar current mini-bootcamp model

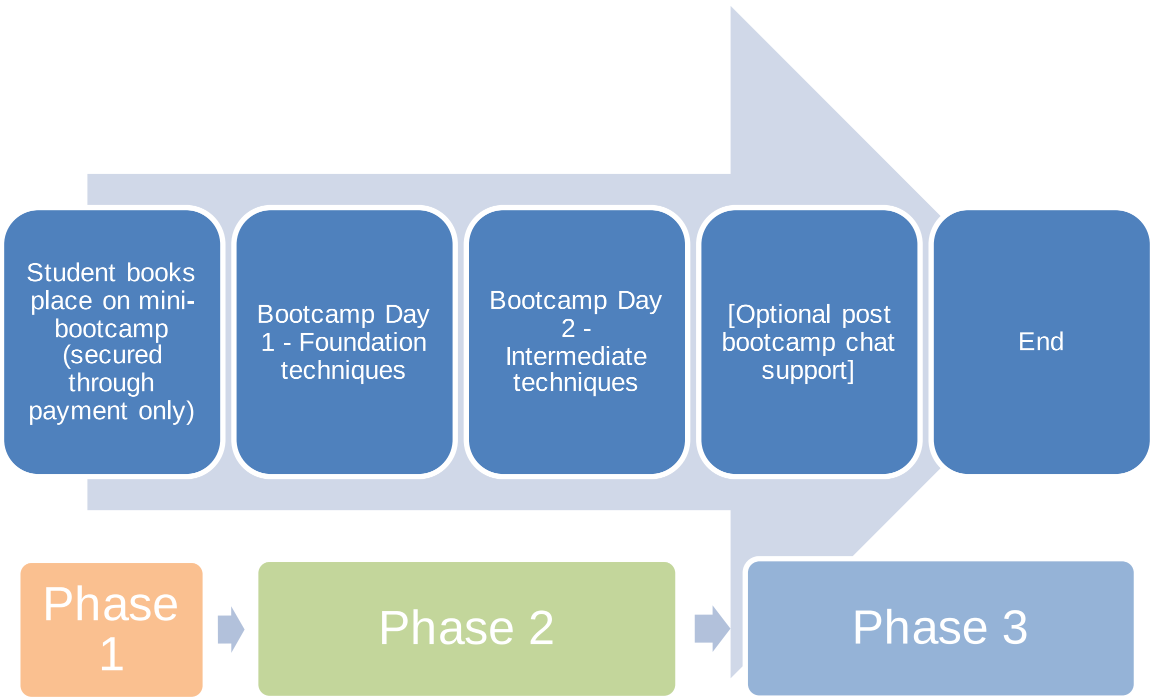

Figure 3. 1: Exemplar current mini-bootcamp model

From this it can be seen that there are essentially three 'phases' that students progress through when they engage with mini-bootcamp learning.

1. **Phase 1**: Pre-bootcamp. This involves the student selecting a mini-bootcamp, booking a place through payment. There are no other requirements. However, there is an assumed understanding that the student wants to gain skills/knowledge delivered during the bootcamp and that face-to-face attendance is required.
2. **Phase 2:** During bootcamp. This involves 2 days of face to face teaching at the bootcamps location, in a classroom setting.
3. **Phase 3**: Post-bootcamp. The bootcamp has finished, there is no expectation that the learning process continues, it can and usually does end as soon as the student enters this phase.

3.2 Extended bootcamp model - phases

An extended bootcamp model (see Figure 3.2) was developed and argued for an optimum mini-bootcamp learning process, incorporating critical and suggested viewpoints from authors in the literature review. This model effectively provided an extension of Figure 3.1, arguing for a more robust 'third phase' post-bootcamp, in an attempt to improve knowledge retention. Phase three within Figure 3.2 incorporated an assessed task informed by FDD methodology, where students were required to apply and develop the skills they gained post-bootcamp by coding features within Android (apps), this task was then assessed by the mini-bootcamp teacher. Students then needed to complete an objective based knowledge assessment to determine if skills and knowledge had been retained. This third phase formed the crux of the methodological approach which is addressed in the next chapter.

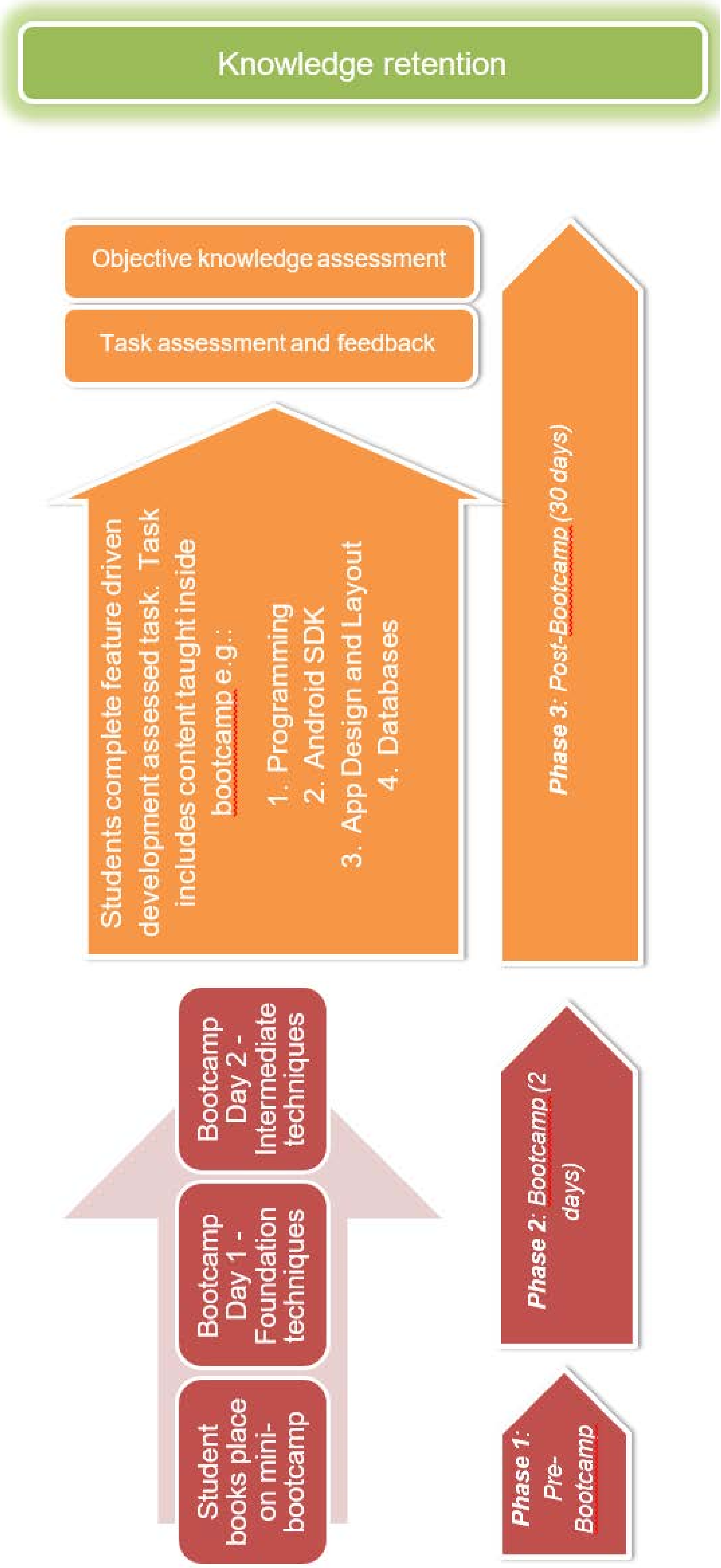

Figure 3. 2: Extended bootcamp model

The pre-bootcamp (Phase 1) is where students choose the mini-bootcamp they wish to attend and book a place through providing payment. A mini-bootcamp typically assumes no prior knowledge of programming - there are no knowledge pre-requisites before joining the bootcamp (International Telecommunication Union, 2016; London App Brewery 2018). A student chooses to attend a bootcamp knowing that the bootcamp is designed with this in mind. Unlike the 'Ready to work' bootcamp model, where students have high expectations of gaining employment upon graduation, a mini-bootcamp student is likely to be attending the bootcamp to gain a specific skill set (Android app development in this case), with no other promises made, other than the intended learning outcomes of the course (International Telecommunication Union 2016; Thayer & Ko 2017). Mini-bootcamp providers are open to students of any background, the learning outcomes do not change, the student will have acknowledged and accepted the format and content of the one-two days programme, including the understanding that no prior programming knowledge is required. Mini-bootcamps adopt a zero pre-requisites approach when recruiting students, although the format does not categorically stop them from requiring pre-requisites. However, this approach appears to be the crux of the mini-bootcamp model, any changes to this and it would no longer become a mini-bootcamp. From a knowledge retention perspective, literature has not indicated this pre-bootcamp phase to be an an area of concern. Therefore the pre-bootcamp 'Phase 1' is not targetted within the methodology in this study. Whether this may impact knowledge retention is unknown, this could be an area for further research.

3.21 'During Bootcamp' - Phase 2

The format of a mini-bootcamp, such as the Android app development bootcamp offered by The London App Brewery, is an intensive time limited period across a period of one-two days (London App Brewery 2018a). During these two days students study intensively, normally from 09.00 – 17.00, each section of the day is progressive in terms of skill development and content, in The London App Brewery's example – foundation skills development progresses towards intermediate skills. The mini-bootcamp format has been praised by authors for its ability to rapidly develop a skillset in a short space of time, thus achieving its fundamental purpose (Heskin et al. 2015; Waguespack et al. 2018). The fundamental challenge with bootcamp learning appears to be explicitly retaining the knowledge and skills gained within these one to two days. It appears as though the mini-bootcamp format is highly effective at achieving its objectives during the one-two days it operates. Thus, Phase 2 of this model was not interfered with. The majority of the criticisms and challenges point to retaining the knowledge post-bootcamp.

3.22 Post-bootcamp period (30 days) - Phase 3

The literature review revealed the need for a compulsory post-bootcamp Phase 3, an important phase which is needed to apply the skills/knowledge gained during the face-to-face bootcamp within Phase 2. Model 1 lacks this, where any sort of post-bootcamp activity is optional - ultimately students could complete the bootcamp and simply not apply/recall any of the skills/knowledge they have gained, which indicates a knowledge retention isssue. A study by Sonnadara et al. (2012) was used to determine a suitable post-bootcamp timeframe in order to target knowledge retention. Within Sonnadara et al's. (2012) study (based on orthapaedic surgical skills bootcamps), three groups of students were tested thirty days after completing a mini-bootcamp. Moreover, the period of thirty days before assessing knowledge retention also appeared within Moulton's (2006) study. Therefore, the thirty day time period has been tested as an approprate time frame to determine if knowledge retention had been affected, it can clearly be seen that thirty days post-bootcamp, knowledge retention had degraded. Furthermore, Moulton et al. (2006) argues for practice to be distributed over a period of time, a thirty day period would enable practice beyond the mini-bootcamp, stretching the practice from a one to two day bootcamp to thirty one to thirty two days, in a more distributed fashion. During this post-bootcamp period, students engaged with a task, informed by Feature Driven Development (FDD) methodology, which was used as a vehicle to distribute practice and target improved knowledge retention.

Finally, it is important to recognise time constraints within this MRes study, after mapping out the project using a Gantt chart prior to emarking on the study, a period of no longer than thirty days was feasible.

4.0 Methodology

This chapters reviews appropriate research methods to inform the research design. Qualitative and quantitative research methods are analysed.

4.1 Quantitative vs. qualitative

Quantitative methods enable the gathering, interpretation and analysis of numerical data, whereas quanlitative methods enable the gathering, interpreation and analysis of narrative data (Teddlie & Tashakkori 2009). The research question itself initially indicates that a quantitative approach is needed. The overall aim of the study: *To determine if an FDD based assessed task completed post-bootcamp can improve knowledge retention for Android app development mini-bootcamps;* indicates that the researcher needed to determine *if* the FDD task had an impact on knowledge retention, then the extent of the impact of the FDD task needed to be determined. This required research methods which returned a numerical dataset in order to calculate a percentage value, rather that data which provided a descriptive account (Crowther & Lancaster 2005). However, if data is needed to provide more of a narrative, or where numerical data is not available, then qualitiative methods may be appropriately considered.

In order to understand the content of app development bootcamp curricula, it was appropriate to consult secondary sources, particularly where time constraints apply, more so given the time parameters of this MRes study. Hypothetically, this study could run comparatively across a plethora of bootcamp providers, however, that posed practical and logistical challenges. Thus, where overall thematic analysis is needed, within a cost and time constrained setting, secondary qualititative research was considered an appropriate approach. Largan & Morris (2019) outline the qualitative secondary research to be a systematic approach to gathering data that already exists, such as data already available on the internet. This is different to primary research where the data is collected first hand. However, one of the clear issues with this approach is the data's credibility, dependability, transferability and confirmability, thus it was important the researcher ensured that the data is authentic and representative (Largan & Morris 2019).

In order to determine if the FDD task had an impact on knowledge retention, a group of participants were compared to an alternative group that did not complete the FDD task. Ultimately this required a type of controlled experiment, where one variable is changed (independent variable) and compared to one that remains unchanged (dependent variable) and enables an elimination of factors which may affect the outcome (Crowther & Lancaster 2005; Cohen et al. 2011). In order to ensure, validity and reliability of the data, a controlled experiment could determine an accurate cause and effect of the FDD task; a controlled experiment enabled the researcher to assess any variables that had an impact. However, this was a challenging issue with experiments matching human variables of each control group, which needs to be done randomly to reduce errors (Cohen et al. 2011).

Questionnaires enabled the collection of survey information, typically whilst the researcher was not present and often provide numerical data (Cohen et al. 2011). This was particularly useful when gathering data on participants' knowledge retention 30 days post-bootcamp. The participants are no longer physically in the same room as the researcher, questionnaires could be completed remotely and are easily accessible via electronic means. However, as Cohen et al. (2011) continues, they were time consuming to prepare.

5.0 Research Design

This section outlines a timeline containing key components and stages of the study, from initial literature review to a proposed solution to the hypothesis.

Step 1	• Literature review
Step 2	• Hypothesis generation and aims/objectives
Step 3	• Outline methodology: • Conduct skills/curriculum analysis of existing bootcamps • Design model bootcamp • Design FDD task • Design knowledge retention questionnaire
Step 4	• Deliver model face to face bootcamp
Step 5	• 30 Day period starts • Participant Group B completes FDD task over 30 days
Step 6	• End of 30 days, Group A and B complete knowledge retention questionnaire
Step 7	• Data analysis - FDD Task & knowledge retention questionnaire
Step 8	• Evaluation of study and objectives

Table 5. 1: Stepped research project timeline

Essentially, this section contains justification for the following components of the Research Design and solution:

1. A model mini-bootcamp (Android app development) – covering skills which formed the basis of the FDD assessed task
    a. This is run twice for 2 groups of 5 students (Group A and Group B)
2. FDD assessed task
    a. Group A did not complete the FDD based assessed task (Control group)
    b. Group B did complete the FDD based assessed task (Experimental group)
3. The FDD informed assessed task

4. Objective skills/knowledge questionnaire

5.10 Bootcamp curricula analysis and designing a model bootcamp

In order to improve reliability and validity of findings within this study, a representatitve example of an Android app development mini-bootcamp was used. Essentially a model experimental face-to-face bootcamp within the phase 2 component of the study, this was based on a secondary analysis of current bootcamp curricula. As seen in Moulton et al's. (2006) study, two different test groups experienced this bootcamp, these are known as Group A (those that did not complete the FDD task – the control group) and Group B (those that do – the experimental group), each group was joined together to experience the same model bootcamp. It was important that both groups were taught at the same time, within the same model bootcamp to ensure an identical experience. Although an 'identical' bootcamp could have theoretically been taught on two separate occasions, they would have been slightly different in their delivery due to the dynamic nature of face to face teaching (difficulty replicating a live teaching experience), thus introducing bias and affecting reliability between the two groups. Both groups A and B contained no more than five people, inline with London App Brewery's (2018a) stipulation that mini-bootcamps should contain a student – teacher ratio of 10:1. Group A and B's demographic make up was determined by a first come first served basis, to model the *'pay for a place'* model that mini-bootcamps adopt. This was in line with the *lack* of pre-requisites for mini-bootcamps (International Telecommunication Union 2016; London App Brewery 2018a). The length of this bootcamp was two days, delivered in a face to face format, in line with what was currently been delivered in the market at the time of writing (London App Brewery 2018a).

To ensure transferability and reliability of the study's findings, the curriculum of the experimental bootcamp was informed by an analysis of current Android app development bootcamp providers. This enables Android bootcamp providers to draw relevance to the findings rather than basing this on a single current mini-bootcamp, as bootcamps are all uniquely designed and delivered, findings would largely only be relevant to that one provider. Essentially, this analysis informed the researcher of a list of skills/techniques taught within the bootcamp, which subsequently informed him of both the requirements of the FDD assessed task and the knowledge retention questionnaire at the end of the thirty days. This linear relationship of skills/knowledge ensured a clear determination of whether the specific knowledge/skills gained within the bootcamp had been retained. A general high level mapping of skills and techniques drawn from a variety of Android bootcamps identified a range of common features, on which the bootcamp, FDD assessed task and questionnaire were based. This can be seen in Appendix A.

To work within the time constaints of this MRes study, a secondary internet based research analysis of Android app development bootcamp provider was completed. Bootcamp curricula that informed the analysis met the following criteria:

1. The bootcamp curriculum is readily available to view on the bootcamp provider's website
2. The training is in a bootcamp format, where the course indicates the term 'bootcamp'

This search returned five bootcamp curricula that could be analysed. Time constraints of the study did not allow for further investigation into those providers that did not provide a full overview of their bootcamp curriculum. Furthermore, completing the study with bootcamp providers directly – involving their students as participants, again would prove prohibitively time consuming and potentially unacheiveable if bootcamp providers did not agree to participate. Secondary internet research could be conducted by the researcher within the time constraints of the study.

5.11 Bootcamp presenter

In order to run a model bootcamp a suitable presenter was required. The presenter needed to meet the following criteria in order for the bootcamp to be delivered as a reprensentative example of an actual bootcamp:

Essential criteria:

1. Physically available to deliver the bootcamp over the 2 days specified
2. Technically competent in Android app development e.g.:
    a. Experience with Java progrmaming language used to develop Android apps
    b. Experience with SQLite used to develop Android apps
    c. Experience with XML used to develop Android apps
3. Teaching experience or knowledge of what is required to teach a face to face audience

Desirable criteria:

1. Prior experience delivering software development bootcamp(s) (ideally Android app development related)
2. Teaching qualification(s)

3. Teaching experience with post-18 age group

Outcome:

The essential criterion: *“Physically available to deliver the bootcamp over the 2 days specified”* proved challenging to meet. There were multiple presenters that met the majority of the essential and desirable criteria, however, they were not available to commit to two days worth of bootcamp delivery. Therefore the researcher adopted the role of presenter, this is due to him meeting the following criteria, as seen in Table 5.2 below:

Essential criteria	**Criteria met**
1. Physically available to deliver the bootcamp over the 2 days specified	✓
2. Technically competent in Android App development e.g.:	✓
a. Experience with Java programming language used to develop Android apps	✓
b. Experience with SQLite used to develop Android apps	✓
c. Experience with XML used to develop Android apps	✓
3. Teaching experience or knowledge of what is required to teach a face to face audience	✓
Desirable criteria:	**Criteria met**
1. Prior experience delivering software development bootcamp(s) (ideally Android app development related)	✓
2. Teaching qualification(s)	✓
3. Teaching experience with Post-18 age group	✓

Table 5. 2: Bootcamp presenter criteria

5.2 FDD Assessed Task

Group B completed a task during the thirty days post-bootcamp, which essentially took the form of a list of app features that they were required to code, requiring students to apply a certain set of techniques (based on FDD methodology). Each feature required coding using certain skill(s)/knowledge, the task was informed by the analysis in Appendix A, can be seen in Appendix B (Ashmore & Runyan, 2015). The FDD approach provided a clear focus on which skills/knowledge were demonstrated in each feature within the task, these were mapped against the list of skills/techniques taught within the bootcamp (Android, 2018b; Usability, 2018). To enable knowledge retention to be improved, the skills taught in the bootcamp were explicitly targeted within this post-bootcamp task. This task essentially aimed to align the agile FDD methodological software development approach, which had very specific sets of technical requirements for a piece of software with the need to target and develop the specific skills taught within the bootcamp. This study argues that if the student practices coding certain features within an app in FDD's short iterative steps, it will systematically target the skills taught within the bootcamp, and thus impact knowledge retention (Charvat, 2003; Goyal, 2008). Furthermore, using FDD methodology to inform this task is highly appropriate, given the software development nature of Android app development – the task will simulate an experience of a customer setting out requirements for an app (Charvat 2003).

Group B was a randomly selected group of five from the ten attendees of the bootcamp, this was done by assigning a number between 1-10 to each participant and randomly selected using Python's random library within the language's native integrated development environment. The random selection completed at the end of the bootcamp ensured that both the teacher and the participants were unaware of who would complete the FDD assessed task – thus reducing bias, where the teacher may inadvertently *'know'* who Group B are and teach/interact with them in a different way. Participants were not allowed to 'swap' with others who were potentially keener to complete the task. This ensured that this task could be assessed by the bootcamp author and would be able to judge whether certain skills/competencies had been demonstrated. This task essentially required them to demonstrate a portfolio of skills, where they applied and demonstrated a breadth of skills learnt within a bootcamp – producing a piece of work incorporating the skills acquired in the bootcamp (Taylor & Smith 2015).

Goodliffe (2015, p.221) suggests that engaging with a task that produces a type of portfolio of demonstrable skills can be considered a way to prove a "*current working set of knowledge*". Moreover, this aligns with Wilson's (2018) viewpoint, that skills should be consistently applied and stretched to ensure they stay fresh. Arguably, the creation of a post-bootcamp task enables skills and knowledge to not only stay current, but also be used to stretch the skills gained within the bootcamp. Ultimately, this could contribute to better knowledge retention than if they had not engaged with the task. Importantly, International Telecommunication Union (2016) indicates that skills gained within bootcamps detract when students venture off into other online learning mediums, such as MOOCs. Online learning programmes and MOOCs may focus on a different set of skills and learning outcomes to the bootcamp, which could mean that skills gained in the bootcamp are not applied and stretched, thus exacerbating the knowledge retention issue as skills are learnt and potentially lost. Thus, the requirements and content within the assessed task needs to explicitly ensure that the skills within the bootcamp are applied and stretched (Wilson, 2018). Thus each task, or required skill needs to relate to the skills/knowledge gained in the bootcamp.

The FDD assessed task was explicitly informed and mapped against the skills/knowledge covered within the bootcamp. A mapping of these skills was required to ensure skills were covered accurately. To ensure that the bootcamps' specific skills and knowledge are retained, feature driven development (FDD) methodology informed the foundation requirements for the FDD assessed task. Essentially, students were expected to code various features within an app, these features explicitly required the use of skills/knowledge gained during the bootcamp. Each feature required coding using certain skill(s)/knowledge, the task informed by the analysis in Appendix A, can be seen in Appendix B (Ashmore & Runyan, 2015). FDD provided a clear focus on which skills/knowledge were to be demonstrated in each feature within the task, these were mapped against the list of skills/techniques taught within the bootcamp (Android 2018b; Usability 2018).

Designing of the FDD assessed task required the following:

1. List of skills/knowledge taught on the bootcamp
2. List of app features required in the task (e.g. Login feature)
3. List of skills/knowledge required for each app feature

5.3 Assessing the FDD task

Brown (2017) indicates that incorporating formative assessment within student learning (regardless of the format) increases motivation and engagement with the learning. This aims to ensure that students engage with the task itself and what they have learnt in the bootcamp by applying and stretching their skills during the post-bootcamp phase, with this there would be a potential lack of motivation to engage with any learning post-bootcamp (Wilson 2018). Formative assessment is developmental and forward looking, it improves student learning, and thus potentially knowledge retention (Wiliam 2011). Incorporating an assessment was desired to motivate students to engage with the task, and hopefully prevent 'opting out'. The engagement with this task is argued to improve knowledge retention. Targeted, specific feedback was provided to improve learning, and provide a motive for engaging with the task (Brookhart 2017). To ensure feedback is specific, the assessment was designed around a checklist of features, derived from the portfolio task itself. The FDD assessed task was assessed upon the completion of the thirty day period. Considering the task is bespoke, a simple approach to assessment was taken. Each feature, which is explicitly mapped to the skills/knowledge taught within the bootcamp, was assessed on a yes/no achieved basis, based on the simple *evidence of* descriptors outlined in Appendix D.

5.4 Determining knowledge retention – objective knowledge retention questionairre

Moulton et al. (2006) determined knowledge retention in their study through objective based outcome assessments explicitly linked to knowledge they gained within the bootcamp. Similarly, Ceresnak et al. (2017) used a knowledge-based examination to determine knowledge before and after the bootcamp. Thus, a set of objective based questions linked to the aims and objectives of the bootcamp informed a questionnaire which both Groups A and B completed. This questionnaire was used to determine knowledge retention thirty days post completion of the bootcamp, as used by Moulton et al. (2006). The questionnaire formed a type of 'test' to determine knowledge retention. Questions were derived from skills/attributes within each group of the taxonomy (Appendix A). The questionnaire can be found in Appendix C.

As highlighted in the literature review, prior programming experience is not a requirement or pre-requisite for an Android mini-bootcamp. However, it was useful to capture this data to contribute to the analysis of findings and evaluation component of this thesis. Thus this question was included:

Q. Did you have any programming experience prior to attending the course? If yes, what?

This did not form part of the knowledge retention assessment score (discussed in the subsequent assessment strategy section). Furthermore, as assumptions were made regarding student motivations (i.e. it is assumed that students did not have existing Android app development skills), it was also be useful to determine if this assumption is true and to identify any correlations that can be drawn with prior experience (generic programming or subject specific (Android in this case) and performance in the knowledge retention questionnaire. Therefore, this question was also included:

Q. Did you have any Android development experience prior to attending the course? If yes, what?

5.5 Assessment strategy

Skills and knowledge delivered within the model mini-bootcamp, informed the FDD task and subsequently informed the knowledge retention questionnaire, this mapping can be seen in Appendix D. It is clear that a transparent and clear mapping of skills filtered through each stage of this method, so that knowledge retention was specifically targeted in order to determine the level of retention for each participant. Within this in mind, both the FDD task and knowledge retention questionnaire adopted a simple 'correct/incorrect' (knowledge retention questionnaire) or 'demonstrated/not demonstrated' (FDD task) approach for determining if the skill has been demonstrated with only one mark/point allocated to each correct answer.

Questions within the FDD were based on the curricula analysis (highlighted in the blue column of Appendix D), descriptors were carefully drawn up to assess knowledge of a particular skill or element of knowledge. They were assessed on a 'demonstrated/not demonstrated' basis. Any non-responses did not attribute any points/marks. Only one mark/point was allocated to each correct answer.

1. Participant's FDD task *raw* score = X
2. Total possible FDD task *raw* score = 22
3. Average percentages (only participants' percentage scores are quoted in this study, not participants' raw score) = X/Y.
4. E.g. if participant A, achieved an FDD *raw* score of 10 and the FDD task contained 22 descriptors to be assessed against:
5. 10/22 = participant A's knowledge retention score would be 45%

The FDD questionnaire could be completed over any days within the 30 day post-bootcamp period.

The expectations of the FDD task for participants was solely that the participants engage with it – performance in this task was not the main focus or crux of the study. Participants were not solely aiming to maximise performance in the task, much as perhaps a university student would, e.g. seeking to meet criteria to improve their grade. However, it is useful to determine a level of performance, in order to do this a numerical value (grade) was determined. This supported the study's analysis and aim to provide insight into any correlations between the FDD task performance and results from the knowledge retention questionnaire. There were a total of twenty two skill areas that the FDD task targeted, each area was based on a simple descriptor for example, to determine if *App Design and Layout* skills have been demonstrated, the following were either assessed as *Yes* or *No*:

1. Evidence of appropriate use of Layout Attributes
2. Evidence of Logcat testing

The assessor of the task used evidence within the students' submission, if they had demonstrated evidence (as appropriately and sensibly determined by the assessor) then at some point during the project the skill had been applied/used. This is in line with demands from Thayer & Ko (2017), Wilson, (2018) and Weis et al. (2018) that skills are addressed and applied post-bootcamp. How these are applied and the extent to which these are applied is irrelevant in this study. Fundamentally, the FDD task itself (Appendix B) provided the important opportunity for students to apply a specific skill. This study also recognised the limitations behind this subjective and loose approach to assessment. If findings deem that further investigation into the performance of students in the FDD task is needed, then a more robust assessment criteria would need to be determined. However, this process met the needs of the study – focusing on improving knowledge retention.

Finally, a similar approach has been taken to determining student performance within the knowledge retention questionnaire. Within this, twenty seven questions were used, mapped against the content taught within the bootcamp. For example, using the above example with skills/knowledge within *App Design and Layout*. The following questions are asked, assessing various skills/knowledge within this skill/knowledge area:

1. *Which best matches the definition of dpi?*

2. *Which of the following are examples of manipulating XML code of an activity?*
3. *User interface elements: buttons, images and edit text are part of which class?*
4. *Name 3 attributes of an edittext user interface element?*
5. *If a programmer wants to align an image to the centre of the screen they could access which of the following to do this.*
6. *Identify 2 important considerations when including images within your app.*

As with the FDD task - the assessor of the knowledge retention questionnaire, used their specialist knowledge where short 'open' answers were given to determine if there was appropriate evidence. However, the majority of the questions required limited responses. If students responded accurately, it was deemed that they have retained knowledge within that skill area. Any non-responses did not attribute any points/marks.

Questions within the knowledge retention questionnaire (highlighted in the red column of Appendix D), were based on the curriculum analysis, questions were carefully drawn up to assess knowledge of a particular skills or knowledge. They were assessed on a correct/incorrect basis, with only one mark/point allocated to each correct answer.

1. Participants knowledge retention questionnaire *raw* score = X
2. Total possible knowledge retention questionnaire *raw* score = 27
3. Average percentages (only participants' percentage scores are quoted in this study, not participants' raw score) = X/27.
4. E.g. if participant A, achieved a knowledge retention questionnaire *raw* score of 12 and the questionnaire contained 27 questions
5. 12/27 = participant A's knowledge retention score would be 44%

The knowledge retention questionnaire was completed on day 30 of the 30 day post-bootcamp period.

5.6 Statistical significance

Due to the low number of participants in the study, generating a fairly small amount of data with which to analyse and draw conclusions, the statistical significance of the dataset needed to be determined. A statistical significance test was completed to determine how likely the data supported a null hypothesis (i.e. that there is no difference between the 2 population test groups in the study). A T- Test was completed and a P value generated to determine the statistical significance of the data. (University of Washington 2002; Yale 2019). This was the most appropriate test, due to the direct comparison between 2 different populations within the study:

Group A (those that did not completed the FDD based assessed task) – Standard deviation of X

Group B (those that did complete the FDD based assessed task) – Standard deviation of Y

The 'standard' level of significance 0.05 is used. If a P value of between 0.01< 0.05 (but not closed to either) is produced then the dataset is determined to be statistically significant (Rumsey 2011).

6.0 Ethical considerations

Bournemouth University's (BU) ethical guidelines were adhered to when conducting the research (Bournemouth University 2018). This included a check list of considerations. This checklist highlighted a few areas for further consideration. Storage of data relating to the participants (name and email address) will be kept in accordance with BU's policy. A participant information sheet made all aspects regarding storage of data including participant withdrawal clear.

The researcher works for a university (other than BU) and contacted potential participants (staff and students) through generic, high level email communications via University systems. In order to do this, appropriate ethical guidance was sought from the university.

To ensure a reliable, valid and credible collection of data when conducting the assessment of the FDD task and analysing questionnaire results, participants were asked not to discuss the FDD task during the thirty day period post-bootcamp.

7.0 Analysis and discussion of findings

This chapter presents the research findings and offers analyses into their impact on the study. Beginning with an analysis of bootcamp providers and their curricula, then leads to a discussion of the experimental model bootcamp and its delivery. Finally, the FDD task and knowledge retention questionnaire results are reviewed.

7.1 Bootcamp curriculum analysis

The analysis of different Android app development bootcamps (Appendix A) revealed the following about curriculum content, the process for completing this analysis is given in outline in section 4.10:

1. A relatively consistent commonality of general themes:
    a. Android Studio IDE
    b. Development of programming skills (Java)
    c. App design and layout (UI/UX)
    d. Databases
    e. Use of Android classes
    f. Use of APIs
    g. App development process or life cycle
2. At a granular level, there are subtle variances between different uses of Android features:
    a. Database technology use: Epicodus, (2018) use a firebase database and teach using SQL and NoSQL, whereas Turn to Tech (2018) use an AWS powered database and teach SQL lite.
    b. Web APIs used: Turn to Tech's (2018) (2018) bootcamp features social networking APIs, whereas Eleven Fifty (2018) introduces Google Maps

From this analysis, the model bootcamp incorporated a set of skills and techniques under the general themes as seen above. Bootcamps by Turn to Tech (2018) and Epicodus (2018) align these key themes in their curriculum in a progressive format. These are where each theme, or set of skills lays the foundations, and enables access to the subsequent set of skills. For example, the first element within Turn to Tech's (2018) bootcamp, provides an introduction to programming skills with Java, following by UI (User Interface) layout and fundamentals of Android app development.

Similarly, initial components of Epicodus's (2018) bootcamp include installing and gaining familiarity with the Android Studio IDE and emulator, followed by interface layouts and design with an overview of XML files. This all takes place before classes and interactions with a database are introduced. Therefore, as knowledge and skills progress within a bootcamp in a 'scaffolded' format, it would be useful to illustrate these in a type of progression taxonomy. This taxonomy will indicate the progress of different skills within their respective themes. Each level of the taxonomy contains knowledge which can be explicitly and independently assessed post-bootcamp. This approach would assist determining exactly which levels or themes of skills/knowledge have been retained after the thirty day period.

7.2 'Model' bootcamp attendance

The methodology outlined a target number of ten participants for the study, as modelled on an existing bootcamp in the marketplace at present, ensuring the teacher:student ratio does not exceed 10:1 (London App Brewery 2018a). A total of ten had agreed to participate and were due to attend the model two day bootcamp. However, in line with ethical considerations, two participants voluntarily withdrew from the study the day prior to the bootcamp attendance, leaving eight participants to engage with the study. This enabled a teacher:student ratio of 8:1 and in line within the existing parameters of London App Brewery's (2018a) current practice. The even number of eight, however, does allow the study to equally split when randomly determining those to complete the FDD based assessed task, thus enabling the study to draw a clear parallel contrast between participants when comparing results of the objective knowledge questionnaire (exactly 50% of participants would have completed the FDD based assessed task and 50% not).

7.3 Statistical significance

Using guidance outlined by University of Washington (2002) and Yale (2019), the statistical significance of the dataset was determined. Fundamentally a T – test is used to compare the deviation between the two populations. The standard level of significance 0.05 (alpha) is used in the test. A P-Value was then produced to determine if it lies between 0.01 and 0.05 – a statistically valid score.

Group A (those that did not complete the FDD based assessed task) – SD 8

Group B (those that did complete the FDD based assessed task) – SD 13

T Score = 3

Degree of freedom = 6

P Value = 0.024

Overall the dataset is determined to be statistically significant lying at a near-medium point between the statistcally significant 0.01 – 0.05 (Rumsey, 2011). Thus, indicating that a reasonable level of support for the hypothesis statement, rejecting the null hypthoesis. However a P score closer to 0.01 would have deemed the score to be more statistically significant (Rumsey, 2011). To improve the statistical significance and bring the P score to 0.01 or 0.00 in order to strengethen the rejection of the null hypothesis, the model bootcamp would need to run for two more iterations (including this study's bootcamp, this requires 3 in total). This was calculated based on a total of 30 participants (15 total acting as a control group and 15 total as part of the experimental group), adhering to the student teacher ratio of 10:1 per bootcamp. Time restrictions within this MRes study meant that extra bootcamps to enable a more statistically significant dataset could not take place. A further 3 bootcamps would enable at least 30 participants (15 Group A; 15 Group B). This assumes that further bootcamps would need to include a full complement of 10 participants in each bootcamp.

7.4 Objective knowledge retention questionnaire results

Averages and relevant statistics were drawn from findings in Appendix E and have yielded interesting results. The questionnaire contained 27 questions directly mapped to the skills/knowledge analysis, which directly informed the curriculum of the bootcamp. The percentage scores were calculated as follows:

6. Participants knowledge retention questionnaire *raw* score = X
7. Total possible knowledge retention questionnaire *raw* score = 27
8. Average percentages (only their percentage scores are quoted in this study, not their raw score) = X/27.
9. E.g. Participant A, achieved a knowledge retention questionnaire *raw* score of 14.
10. 14/27 = 52% (2DP)

7.41 Overall headline statistics

a) Average score of all participants: 61%
b) Range of all participants: 41% - 93%
c) Average of participants with programming experience: 58%
d) Average of participants without programming experience: 63%
e) Average of participants who did not complete the FDD based assessed task (Group A): 49%
f) Average of participants who completed the FDD based assessed task (Group B): 73%
g) Group A performance range: 41% - 59%
h) Group B performance range: 67% - 93%
i) *All* participants in Group B exceeded those in Group A

Firstly, the data explicitly indicates that those participants who completed the FDD assessed task scored much better in the objective knowledge questionnaire, with averages of 73% vs. 49%. This indicates that the FDD assessed task had a positive impact on the participant's knowledge retention, thus providing evidence to support agreement to the original hypothesis below.

Hypothesis*: Completing an FDD assessed task over a period of thirty days post Android mini-bootcamp will yield improved knowledge retention, when compared to those that don't complete the FDD assessed task*.

This is further reinforced by Group B's minimum range score (67%) exceeding Group A's maximum range score (59%) (as seen in Figure 7.1 below). Therefore the performance of all participants who completed the FDD based assessed task exceeded that of the opposite test group. Thus in line with Weis et al's (2018) belief that practice (i.e. app development) needs to be applied post-bootcamp in order to maintain proficiency. The data shows that Group B have arguably demonstrated a maintained proficency, in order to achieve a better results in the objective knowledge questionnaire.

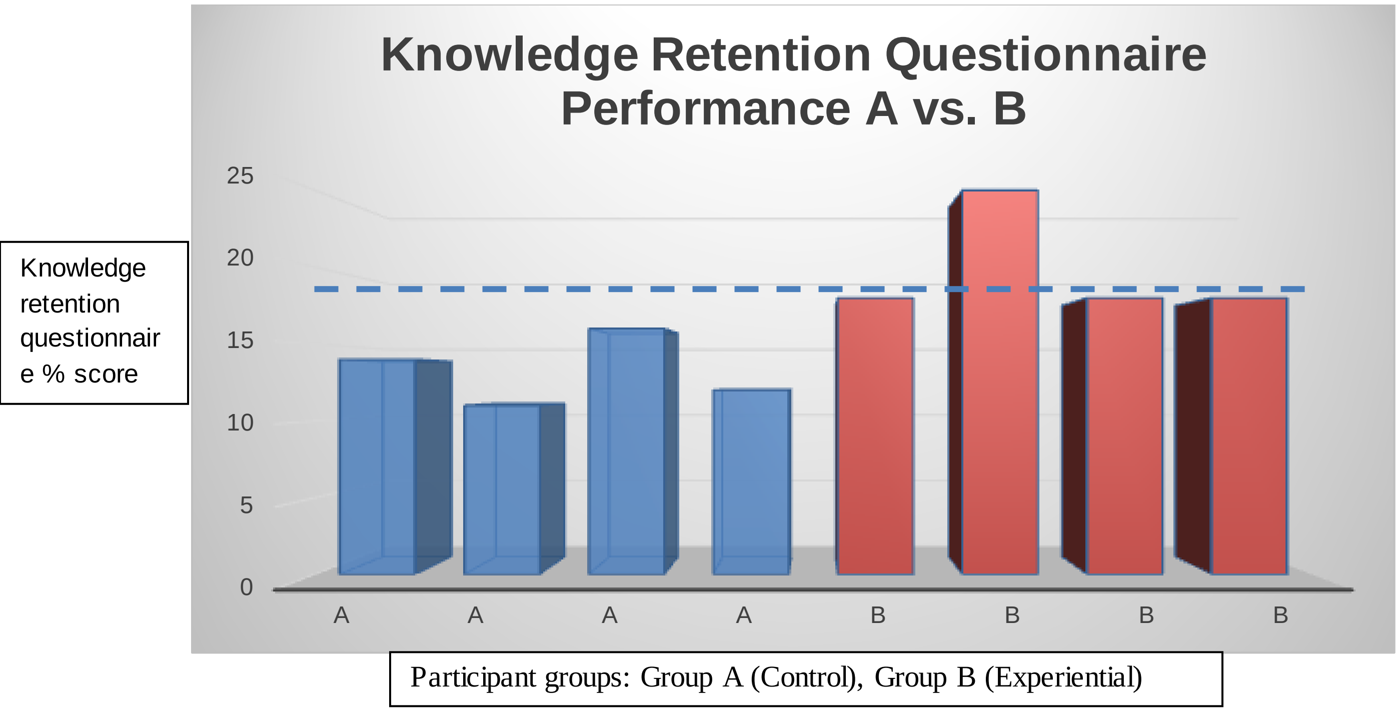

Figure 7. 1: Knowledge retention questionnaire performance group A vs. group B

This is further supported by Wilson (2018) who indicated that learning Java (predominant programming language used to develop Android apps) in a bootcamp format would lead to knowledge retention issues, unless there was an opportunity to extend training to enable skillset(s) to remain 'fresh'. Clearly, skills and knowledge have been kept 'fresh' in order for Group B to outperform Group A in the knowledge retention questionnaire. However, whilst this is reassuring for the hypothesis, there is no data insight to indicate the frequency of engagement with the FDD task over the thirty day period. The researcher received tasks close to the deadline, this may have acted as a revision exercise for the knowlede assessment, rather than the repeated application of skills, that both Wilson (2018) and Weis et al. (2018) advocated.

The range between all participant scores is extensive: 41%-93%. The poorest knowledge retention performance was with participant B (group A) and the strongest knowledge retention performance was with participant F (group B). Furthermore participant B also indicated prior programming experience, which could be argued gave them an advantage from a knowledge retention perspective. Overall, this indicated that the FDD based assessed task has, in some cases successfully targeted specific retention of the skills/knowledge gained in the bootcamp.

Mini-bootcamps typically do not stipulate pre-requisite skillsets, unlike the longer 'ready to work bootcamps' which span over a period of twelve to twenty four weeks (International Telecommunication Union, 2016). Results from this study indicate that those with prior programming knowledge did demonstrate a higher knowledge retention score in the questionnaire. Conversely, they underperformed when compared directly to their non-programming peers (58% vs. 63%). Thus, from knowledge retention perspective, there should be no new motive for mini-bootcamp providers to introduce pre-requisite skills requirements for mini-bootcamps. For other types of bootcamps e.g. 'ready to work bootcamps', these results could indicate that further research is needed if these providers continued to set pre-requisite requirements/tests due to perceived knowledge retention issues. It is understood 'ready to work' bootcamps are keen to ensure accelerated performance during their bootcamp, along with the need to access higher level skills more quickly, thus requiring a greater level of foundation knowledge (Academy 2018). This is largely due to their marketed promises of employment post-graduation, thus having higher standards. However, it must be noted that the deviation between the questionnaire scores of 63% vs. 58% is fairly minimal (5%) and does not take into account participant ability. Further data would need to be collected to effectively determine if prior programming experience has a largely unfavourable impact of knowledge retention.

7.5 FDD assessed task

Group B was comprised of 50% of the entire population that attended the model bootcamp experiment. They were randomly selected and submitted their project for assessment at the end of the 30 days. The percentage scores were calculated as follows:

1. Participants FDD task *raw* score = X
2. Total possible FDD task *raw* score = 23
3. Average percentages (only their percentage scores are quoted in this study, not their raw score) = X/27.
4. E.g. Participant E, achieved an FDD task *raw* score of 13.
5. 13/23 = 59% (2DP)

The following statistics were highlights from the findings:

a) A range of 0 – 73% across participants (NB. one participant scored 0% in the knowledge task)
b) An average score of 40%

Overall, the scores were fairly low, with an average of only 40%. However, drawing on data from the objective knowledge questionnaire, Group B's knowledge retention assessment proved to be greater than Group A's (average of 49% (Group A) vs. 73% (Group B)). The study's hypothesis proposes that participants who simply needed to engage with the FDD assessed task would have a positive impact on knowledge retention, which appears to have proved true. However the individual performance with the FDD assessed task, whilst an insightful datapoint cannot independently prove the hypothesis. Whilst Group B demonstrated better knowledge retention than Group A, it is unknown if Group A engaged with an application or revision of the skills/knowledge that they gained during the bootcamp. Anything that they did or did not do may or may not have impacted the knowledge retention score in the objective based knowledge questionnaire. However, regardless of Group B's performance in the FDD based assessed task, all participants in Group B outperformed all participants in Group A. It can be seen that the highest performer in the FDD based assessed task (participant F) also achieved the highest knowledge retention score in the objective knowledge questionnaire. Whilst this is seen as a correlation at the highest end of the spectrum, all other participants in group B scored equally (67%) in the knowledge questionnaire and yet scored drastically differently in the FDD based assessed task (0%, 27% and 59% respectively). The link between the FDD task performance and knowledge retention is inconclusive, due to the limited dataset. Perhaps in this case, completing the task acted as a revision exercise, enabling better performance in the knowledge questionnaire.

8.0 Conclusions

The study's overall aim was to determine if an FDD task completed post-bootcamp would improve knowledge retention. While recognising the limitations of the analysis, the aim has broadly been achieved. An evaluation against each objective, addressing strengths and areas for development are addressed in this chapter. A set of clear conclusions will be laid out later in this chapter.

8.1 Knowledge retention with Android mini-bootcamps is an issue

This study indicates that knowledge retention is an issue with mini-bootcamps, an average knowledge retention score for all participants of 61%, proves that there is a knowledge 'leak' and skills/knowledge explicitly gained within the bootcamp itself were not fully retained.

This directly supports findings by (Moulton et al. 2006) who was one of the first authors highlighting the knowledge retention associated with the bootcamp learning format. This provided a new insight into Android mini-bootcamps, with implications for both bootcamp provided and academia. Furthermore, the use of a knowledge retention questionnaires as a knowledge assessment, re-enforces work by Moulton et al. (2006) and Sonnadara et al. (2012) who adopted assessment of bootcamp skills to determine a level of knowledge retention thirty days post-bootcamp. However, research is needed to robustly confirm this as an independent knowledge retention issue. The knowledge retention questionnaire could have been completed on day one of the thirty day period, and then subsequently on day thirty. This would more accurately indicate if there is a knowledge 'leak' (retention issue), if so, the extent of this leak could be determined. For example, scores in the questionnaire on day one may be the same as day thirty, in which case, 100% of the knowledge they had has been retained, even if they scored less than 100% on each occasion.

8.2 FDD task has positive impact on knowledge retention

The experimental Group B indicated an improved knowledge retention score in the knowledge questionnaire when compared to those who did not complete the FDD task. Group B scoring an average of 73% vs. Group A's 49%. Thus, engaging with the FDD task over a period of thirty days post-bootcamp improves understanding, indicated with a higher knowledge retention questionnaire score. However, to improve the statistical significance of these findings, data from a further two bootcamps would be needed to generate a more robust data set.

It is not categorically clear whether the improved knowledge retention questionnaire scores were due to the FDD task enabling a better 'understanding' of the content or improved knowledge retention. It could be hypothesised that an enhanced understanding was gained, simply due to the fact Group B engaged with a learning through 'doing' task vs. memorising knowledge to be recalled at a later date. This task may have generated an enhanced understanding of the knowledge gained during the bootcamp and a secondary benefit meant that Group B had a better knowledge retention score the Group A in the knowledge retention questionnaire. Understanding may have improved, but the ability to recall facts (which is what the knowledge retention questionnaire tests) may have not. Therefore, the question is posed as to whether retaining the knowledge and understanding are explicitly linked and how they can be independently improved within an Android app development mini-bootcamp format

8.3 Impact of prior programming experience

Findings regarding the impact of prior programming are interesting, however relatively inconclusive due to the statistical significance of the data, for example where one participant may have over stated their prior programming experience. It is however interesting, and should be investigated further.

Findings within this study indicated that those with prior programming experience achieved poorer knowledge retention scores than those without. Although, as discussed in section 7.2, those with prior programming knowledge may not have necessarily suffered a knowledge retention issue (they may have the same score if assessed on day 1/30 compared with day 30/30), however, they may have a better *understanding* of content. Mini-bootcamps differ in approach to other types of bootcamp by not stipulating any entry requirements prior to gaining a place on the mini-bootcamp. The findings generally support mini-bootcamps' current approach of an 'open door' policy regardless of student background, whilst have clear implications for existing 'ready to work' bootcamp providers that insist on pre-requisite programming knowledge – they may need to evaluate a reasoning for this. Further research and investigation is needed to determine the true impact and specifically determine if prior programming knowledge impacts knowledge retention and specifically which sub-set has an impact within the broad field of 'programming'.

8.4 Further research

The study acted as a type of pilot offering fresh insight into Android bootcamp learning. Laying grounds for a much more focused investigation, along with emerging implications for existing bootcamp providers. Further data is needed through a wider population in order to draw correlations between prior experiences and knowledge retention; further data would also enable insight into which, if any, specific skill/knowledge areas suffer worse knowledge retention. Which pre-existing skills impacted knowledge retention? Were there any particular skills that students commonly struggled to retain?

Furthermore, analysis in section 7.2 revealed that further study is needed to determine whether knowledge retention or understanding (gained through actively 'doing') impacted the knowledge retention questionnaire at the end of the 30 day period. Further steps would be needed to investigate this. For this, the study could be repeated with a larger group of participants. Which compares the impact of a different type of post-bootcamp task comparing which could compare the FDD task (an 'active' type of learning activity that potentially improves understanding), to a group that completes a revision type task (which does not require application and understanding of the techniques, but targets memorising what has been learnt in the bootcamp, not understanding). The study could thus contain the following control and experimental groups as follows:

- Test group A – control group (not engaging with any task post-bootcamp)
- Test group B – complete the FDD task from this study
- Test group C – complete a revision task, which targets memorising knowledge gained in the bootcamp, not active application of the knowledge

This may aim to determine which test group yields better scores within the knowledge retention questionnaire.

9.0 Evaluation and limitations of the study

Overall, the study produced a set of interesting results, not just to be measured against the study's aims and objectives, but also providing implications for future bootcamp practice and further research. This study recognises that there are some limitations which have a bearing on the conclusion, these will be woven into the evaluative discussion.

Firstly, the study's aim and objectives are methodically reviewed against the study's findings. Achievement of the overall aim will be concluded in the conclusion chapter.

Aim: To determine if an FDD based assessed task completed post-bootcamp can improve knowledge retention for Android app development mini-bootcamps

Objectives:

1. In order to identify commonalities across Android app development bootcamp curricula, complete an analysis of different Android bootcamps currently offered in the marketplace to produce a skills/ mapping

Evaluation of objective 1:

Appendix A provided a clear review of content and curricula taught across a five different Android bootcamps. The process of methodically reviewing the Android app development bootcamps currently offered in the marketplace enabled the study to target a valid set of skills. The skills mapping, which involved researching existing bootcamp curricula to map and identify technical curriculum content to key themes/topics, provided a foundation for designing method to achieve Objectives 2 -4. Fundamentally, ensuring knowledge assessments are expliclty linked to a defined set of competencies aligns the study's approach to that by Moulton et al. (2006) and Sonnadara et al. (2012). Also, completing the skills/knowledge analysis, was acheivable for the researcher in the study, so it could be repeated if conducting a similar study in future. It may be difficult for a researcher to acquire access to the students of an existing bootcamp, an existing provider would need to provide access to their students. Furthermore, targeting students of one bootcamp could limit the transferability of the findings, due to a bootcamps' naturally uniqueness to their own curricula. On the other hand, an experimental model bootcamp incorporating analysis of curricula from a range of bootcamps contributed to a more applicable and transferable dataset. Overall, this objective was achieved. However, due to the fast paced and dynamic nature of technology (e.g. Android) and the freedom bootcamps have to change their curricula, from one bootcamp to the next, it would be important that a fresh analysis is completed for any bootcamp based study in the future.

2. Based on the analysis and skills mapping completed in Objective 1 design and deliver a market representative model mini-bootcamp to a cohort of students (group A (control group & group B (experimental group))

The skills mapping, which enabled a highly informed design and delivery of a model mini-bootcamp. This was delivered by the researcher, who is a qualified teacher and technological expert. Due to the pace of the bootcamp, it would be difficult for someone without specialist knowledge to achieve the learning objcetives of the botocamp, thus if running a model bootcamp in future, the deliverer would need both technical skillsets and experience in delivery (e.g. teaching, lecturing etc.). Running the face-to-face bootcamp was also feasable within the time restrictions of this MRes study, data would need to be collected from a range (5 in the case of the study) of bootcamp providers to replicate the broad analysis of skills achieved in this study. The bootcamp was delivered covering all of the skills requirements in Appendix A and provided a good 'model' teaching of the skills, which would be tested in the subsequent knowledge questionnaire.

One important note that became apparent to the researcher, was the onerous nature of the study's requirements on participants, in terms of both time and expectations. Which may have led to two participants opting out of the study very close the delivery of the bootcamp, thus negatively impacting the data set. Whilst experiencing the model mini-bootcamp is attractive to a participant in terms of value, which, should the participant wish to attend a 'similar' mini-bootcamp, they would most likely have to pay a market rate (circa. £400), they still have to commit to two entire days of attendance and may then be selected to complete the FDD based assessed task along with the knowledge questionnaire. This is onerous in terms of time, and participants may realise this and opt out at any stage due to this reason. In future, a 'model' bootcamp needs to ensure it can more reliably attract the full ten particpants required for each bootcamp, to ensure a tight replication of model bootcamp learning (London App Brewery 2018a). This could be achieved through particpant incentives, ideally awarded upon completion of all requirements of the study (bootcamp attendance, completion of FDD based assessed task (if chosen) and knowledge questionnaire).

3. Informed by the analysis and skills mapping completed in Objective 1 design an assessed task for Group B to complete for a period of 30 days post-bootcamp

The FDD task (Appendix B) and the model bootcamp, was designed based on the analysis of current Android bootcamps in the market at the time of the study. The FDD task specifically targeted application of skills/knowledge gained within the bootcamp (Appendix A). Although performance was varied, it did mean that Group B's knowledge retention scores were higher than Group As, indicating a positive impact on understanding of content within the bootcamp. Each skill was then assessed, in a binary yes/no fashion. It would be beneficial to extend the analysis of this data; to understand knowledge retention further in order to determine which skills/knowledge were addressed more confidently than others. A wide range of 0% – 73% meant that some participants struggled or did not apply themselves to the task. This larger leaves insight into performance into the FDD task inconclusive, a larger number of participants would be needed to correlate between the FDD task at a topical level with the knowledge retention questionnaire.

4. Informed by the analysis and skills mapping completed in Objective 1, design a knowledge based assessment questionairre for Groups A and B to complete 30 days post-bootcamp
5. Evaluate group B's performance in the FDD assess task and compare their performance to Group A's in the knowledge questionnaire completed in Objective 4

Knowledge retention was determined by a knowledge questionnaire comprising of 27 questions, explicitly informed by the Android bootcamp skills analysis (Appendix D). Each question targeted specific skills taught within the model mini-bootcamp. This questionnaire was an appropriate method to determine knowledge retention, as modelled in Sonnadara et al's. (2012) study. Firstly, it enabled knowledge retention of the entire population to be tested, highlighting that there is a knowledge retention issue associated with Android mini-bootcamps. This aligns with studies from Moulton et al. (2006) and Sonnadara et al. (2012) who also found the bootcamp format to be detrimental when assessing knowledge retention thirty days post-bootcamp completion. Secondly, it provided a strong datapoint to compare knowledge retention between the two test groups (A and B). With Group B demonstrating a higher knowledge retention score across the board when compared with Group A, thus confirming the original hypothesis.

Fortunately, despite low participant numbers the data was determined to be statistically valid, returning a P score of 0.024. However, this lies as a median value between the desirable score of 0.01 and relatively statistically insignificant 0.05 (University of Washington 2002; Yale 2019). Thus any conclusions within this study are made with this in mind, that statistical significance could be improved. Whilst these general conclusions are determined:

1. There is an knowledge retention issue associated with Android mini-bootcamps, all participants scoring 61% on average
2. A FDD based assessed task positively impacts knowledge retention

As each participant's performance in each question (and thus each skill/knowledge area) within the knowledge questionnaire varies, it is inconclusive if certain skills/knowledge were retained better than others. A larger dataset, across a greater number of bootcamps would be needed to determine this. This level granularity would provide important insight into whether knowledge retention is limited to certain skills/concepts. If they are, bootcamp providers could test adjustments to their programmes and post-bootcamp tasks could be targeted appropriately to develop better retention of specific skills.

Within the knowledge questionnaire, additional questions were added with an aim to determine if any relevant past experience would impact the knowledge retention scores. These questions were:

Did you have any programming experience prior to attending the course?

If yes, what?

Did you have any Android development experience prior to attending the course?

If yes, what?

Overall, 0% of participants possessed prior Android app development experience prior to attend the course. This suggests that mini-bootcamp students are motivated to attend a mini-bootcamp, because they desire to improve skills of a specific skill set that they do not already possess. Furthermore, results indicated that 38% of participants possessed prior programming experience, one participant for example indicated "some python and visual basic". Overall, those with prior programming experience performed on average, worse than their non-programming counterparts (58% vs. 63%). Thus indicating that prior programming experience does not have an impact on knowledge retention of another type of 'programming' related learning (Android app development) when assessed post-bootcamp. This is an important finding for existing mini-bootcamp providers, such as that provided by London App Brewery (2018a), who do not stipulate entry requirements, because entry requirements in this case, would have no impact on knowledge retention of the skills they teach. Thus motivations behind any introduction of prior programming entry requirements would need to be logically justified by other reasoning. At present, entry requirements could serve to deter people booking a space onto a mini-bootcamp. Finally, if the study was expanded or conducted again with a larger number of participants, it would be useful to include further identifiers to help correlate if knowledge retention is improved with pre-determined factors. Such factors could include:

1. Student 'ability' – profiled through either previous academic achievements and/or ability assessment
2. Motivations – why is that student attending the bootcamp?

It is not known if prior programming experience directly related to the content being taught on a bootcamp would impact knowledge retention. E.g. in the case of this project, if participants possessed SQL or Java programming experience, what impact would this have? This would need to be analysed further with a larger dataset comparing participants that have related experience.

The knowledge questionnaire is designed to determine the level of knowledge retention across the entirety of the skills/knowledge covered in the original skills mapping (Appendix D). However, due to the small number of participants it is difficult to draw meaningful correlations between specific skills/knowledge. Each of the participants demonstrated varied abilities to retain each skill covered within the bootcamp. It would be insightful to be able to determine if certain skills suffered better or poorer knowledge retention. In this case, research could be done into the effectiveness of an even more targeted FDD based assessed task, and how this could impact knowledge retention of individual skills/knowledge.

9.1 Recommendations for further research

The field of bootcamp learning within any facet of software development is understudied, any future academic work will make a valid contribution to this field. Studies by Moulton et al. (2006) and Sonnadara et al. (2012) have shown that issues exist, this study contributes to this. Whilst this study highlights the knowledge retention issue, a more robust data set could be achieved to ascertain stronger correlations and conclusions. This study has proven its methodology to be successful in achieving broadly its objectives. This study could be extended over a longer period of time, running another two bootcamps, but crucially using this study as a blueprint, not exceeding ten participants per bootcamp and ensuring up to data skills mapping and analysis of current market trends is important. Refreshing the secondary research that informs the curriculum and skills mapping ensures model experimental bootcamps are current in terms of content will enable findings to be more applicable to bootcamp providers, of whom typically all deliver a bespoke curriculum under a range of topical themes. Furthermore, one of the objectives was to evaluate the knowledge retention of two test groups, analysing the impact a post-bootcamp FDD base task had on knowledge retention. This provided an insight into how targeted tasks could positively impact knowledge retention. However, a larger number of test groups could reveal if particular skills/knowledge suffered worse retention rates. Only an overall knowledge retention score was able to be concluded in this study due to the limited test group. Finally, prior experience needs further investigation, it can be seen that prior programming experience did not have a positive impact on knowledge retention. Prior technical experience or knowledge is a stipulation adopted by a myriad of bootcamp provides in the marketplace. Thus, it would be important to further research prior experiences which affect knowledge retention.

9.2 Benefits of the study

Overall, this study highlights some interesting findings for bootcamp providers, bootcamp students and education researchers.

Bootcamp providers – The mini-bootcamp format presents a range of benefits, however those benefits could be outweighed by a degradation of knowledge and skills post-bootcamp. Investment is needed into nurturing students' skills and knowledge after their skills enhancement gained within the bootcamp programme itself.

Bootcamp students – Bootcamps are a strong choice for students seeking an enhanced skillset during a short period of time. However, students need to consider how they build, retain and develop these skills post-bootcamp. Essentially, findings from this study indicate that attending the bootcamp event is the beginning of a learning journey, learning needs to continue otherwise knowledge and skillsets quickly degrade over time. The first thirty days post-bootcamp need to be used to apply and develop the skills gained during the bootcamp itself.

Education researchers – This study has placed much needed focus onto bootcamp learning, it is understudied, particularly within software development fields. Generally, further research is needed into this learning format, due to its growing prevalence, largely related to the demand for technical skills. Specifically, the period 30 days post-bootcamp has found that knowledge retention could deteriorate, but, measures can be put in place to reduce this degradation, such as an FDD task. Furthermore, knowledge retention needs to be more accurately determined within this thirty day period, capturing knowledge levels at different points during the thirty day post-bootcamp period would provide a more in-depth insight into the impact of the FDD task. It could be that a greater understanding was achieved, rather than more knowledge retained.

References

Makers Academy, 2018. Become a World Class Developer. Available at: https://makers.tech/ [Accessed 22 October 2018].

Android, 2018b. User Interface & Navigation. Available at: https://developer.android.com/guide/topics/ui/ [Accessed 02/11 11 2018b].

Android, 2018. Use Java 8 language features. Available at: https://developer.android.com/studio/write/java8-support [Accessed 23 October 2018].

Ashmore, S. & Runyan, K., 2015. Introduction to Agile Methods. 1 ed. Indiana: Addison-Wesley.

Bournemouth University, 2018. 8B – Research Ethics Code of Practice:

Available at: https://intranetsp.bournemouth.ac.uk/documentsrep/8B-research-ethics-code-of-practice.pdf [Accessed 09 09 2018].

Brookhart, S. M., 2017. How to Give Effective Feedback to Your Students, Second Edition. 2nd ed. Alexandria: ASCD.

Brown, G. T. L., 2017. Assessment of Student Achievement. 1 ed. New York: Routledge.

Ceresnak, S., Axelrod, D.M., Sacks, L.D., Motonaga., K.S., Johnson, E.R. and Krawczeski, C.D. 2017. Advances in Pediatric Cardiology Boot Camp: Boot Camp Training Promotes Fellowship Readiness and Enables Retention of Knowledge. Pediatric Cardiology, 38(3), pp. 631-640.

Epicodus, 2018. Android. Available at: https://www.epicodus.com/android/ [Accessed 13 11 2018].

Eleven Fifty, 2018. Android (Java) 151. Available at: https://www.elevenfifty.org/courses/android-151/ [Accessed 13 11 2018].

Forbes, 2018. Forbes CommunityVoice Connecting expert communities to the Forbes audience.What is This? Available at: https://www.forbes.com/sites/forbestechcouncil/2017/12/21/13-top-tech-skills-in-high-demand-for-2018/#697d97901e5c [Accessed 23 October 2018].

General Assembly, 2018. Android Development Immersive. Available at: https://generalassemb.ly/education/android-development-immersive [Accessed 13 11 2018].

Goodliffe, P., 2015. Becoming a Better Programmer: A Handbook for People Who Care About Code. 1 ed. USA: O Reilly.

Goyal, S., 2008. Feature Driven Development. Available at: http://csis.pace.edu/~marchese/CS616/Agile/FDD/fdd.pdf [Accessed 02/11 11 2018].

Heskin, L., Mansour, E., Lane, B., Kavanagh, D., Dicker, P., Ryan, D., Gildea-Byrne, K., Pawilikowska, T., Tierney, S. and Traynor, O. 2015. The impact of a surgical boot camp on early acquisition of technical and nontechnical skills by novice surgical trainees. The American Journal of Surgery, 210(3), pp. 570-577.

Hunt, J., 2006. Agile Software Construction. 1 ed. London: Springer.

International Telecommunication Union., 2016. Coding Bootcamps: A strategy for youth employment, Switzerland: ITU.

London App Brewery., 2018a. Android App Development Course. Available at: https://www.londonappbrewery.com/event/android-app-development-course/ [Accessed 22 October 2018].

London App Brewery, 2018b. Android App Development Curriculum. Available at: https://www.londonappbrewery.com/wp-content/uploads/2016/11/Android-App-Development-Curriculum.pdf [Accessed 13 11 2018].

Moulton, C., Dubrowski, A., Macrae, H., Graham, B., Grober, E. and Reznick, R. 2006. Teaching surgical skills: what kind of practice makes perfect?: a randomized, controlled trial.. Ann Surg, 244(3), pp. 400-409.

Puri, C. P., 2009. Agile Management: Feature Driven Development. 1 ed. Delhi: Global India Productions Pvt Ltd.

Rumsey, D. J., 2011. Statistics For Dummies. 2nd ed. s.l.:Wiley.

Sonnadara, R. R., Garbedian, S., Safir, O., Nousiainen, M., Alman, B., Ferguson, P., Kraemer, W. and Reznick, R. 2012. Simulation-Based Surgical Education: Orthopaedic Boot

Camp II: Examining the retention rates of an intensive surgical skills course. Surgery, 151(6), pp. 803-807.

Statcounter, 2018. Mobile Operating System Market Share Worldwide. Available at: http://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-201709-201809 [Accessed 23 October 2018].

Statista, 2018. Number of mobile app downloads worldwide in 2017, 2018 and 2022 (in billions). Available at: https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/ [Accessed 23 October 2018].

Taylor, K. & Smith, B. E., 2015. Getting a Web Development Job For Dummies. 1 ed. New Jersey: Wiley.

Turn to Tech, 2018. Part-Time 8-Week Android Bootcamp. Available at: http://turntotech.io/8-week-android-course/ [Accessed 13 11 2018].

Thayer, K. & Ko, A. K., 2017. Barriers Faced by Coding Bootcamp Students. Tacoma, Washington, ACM, pp. 245-253.

University of Washington, 2002. T tables. Available at: https://faculty.washington.edu/heagerty/Books/Biostatistics/TABLES/t-Tables/ [Accessed 17 April 2019].

Usability, 2018. User Interface Elements. Available at: https://www.usability.gov/how-to-and-tools/methods/user-interface-elements.html [Accessed 02/11 11 2018].

Waguespack, L. J., Babb, J. S. & Yates, D. J., 2018. Triangulating Coding Bootcamps in IS Education: Bootleg Education or Disruptive Innovation?. Information Systems Education Journal, 16(6), pp. 48-58.

Wegotcoders, 2018. Immersive Courses. Available at: https://www.wegotcoders.com/courses/immersive [Accessed 23 October 2018].

Weis, J., Farr, D., Abdelfattah, K.R., Hoog, D. and Scott, D.J. 2018. A proficiency-based surgical boot camp May not provide trainees with a durable foundation in fundamental surgical skills. Texas, The Association for Surgical Education.

Wenger-Trayener, E. & Wenger-Trayener, B., 2015. Communities of Practice a Brief Introduction.. Available at: http://delivery.acm.org/10.1145/3110000/3106176/p245-thayer.pdf?ip=86.7.196.57&id=3106176&acc=OA&key=4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35%2E60F1425A4B422422&__acm__=1539687094_d01c871b430e3b33d68f911844646b0c [Accessed 23 October 2018].

Wiliam, D., 2011. Embedded Formative Assessment. 1 ed. Bloomington: Solution Tree Press.

Wilson, G. A., 2018. Could a coding bootcamp experience prepare you for industry?. IT Professional, 20(2), pp. 83 - 87.

Yale, 2019. Tests of Significance. Available at: http://www.stat.yale.edu/Courses/1997-98/101/sigtest.htm [Accessed 17 April 2019].

Appendices

Appendix A: Android curriculum analysis

A high level analysis of curriculum content from current Android App development providers

Bootcamp provider	(London App Brewery 2018b)	(Eleven Fifty, 2018)	(Epicodus, 2018)	(General Assembly, 2018)	(Turn to Tech, 2018)
	Framework: Android	Framework: Android	Framework: Android	Framework: Android	Framework: Android
	Language: Java	Language: Java	Language: Java	Language: Java	Language: Java
	Android studio IDE	Android studio IDE	Android studio IDE	Android studio IDE	Android studio IDE
	Programming: OOP Concepts; Control and data structures	Programming: OOP Concepts; Control and data structures	Programming: OOP Concepts; control and data structures (although not explicitly indicated)	Programming	Computer Science fundamentals, Programming: OOP Concepts; Control and data structures
	Software design (Model-View-Controller) design pattern				
	Cloud data storage: asynchronous API calls	Data models and persistent data	Firebase database: Data Persistence, SQL and NoSQL		Cloud data storage: AWS, SQLite,
	App Design: XML, UI Layout	App Design: Building a UI, screen design,	App Design: UI	App Design: UI, material design, XML	App design: UI and UX

		constraints			
	Android classes and behaviours: e.g. Listeners, activities, intents, adapters	Android classes and behaviours: e.g. activities, intents, adapters, listview, array adapter	Android classes and behaviours: e.g. array adapters, listviews, intents	Android classes [not explicitly mentioned on website]	Android classes: e.g. fragments, intents, gestures, list view
	APIs	Google Maps API	APIs	APIs	APIs
		Preferences			Shared preferences
		Android app life cycle		Iterative design and development sprints	Android app lifecycle

Appendix B: FDD student task

The FDD assessed task completed by experiment Group B in the study

Project Brief	Description
App name:	Visit and Connect in [*Insert place here e.g. Bath*]
Platform:	Android
App project timeline:	30 days
Purpose of app:	This app is for people with a sightseeing interest in [*Insert place here e.g. Bath*]. Visitors to [*Insert place here e.g. Bath*] will be able to gain information on key attractions to arrange and inform visits.
Commercial/private use:	Private – do not deploy to Play Store.
How app will work:	Users firstly login to their app using a username and password. They will be able to access information on key sights in [*Insert place here e.g. Bath*] (e.g. Bath Pump Room); this will include images and links to key websites (e.g. tourist board). This information will be spread across a number of pages within the app. Users will be able to navigate between these pages and interact with each page via UI elements. Users will also be able to search for address information about key locations within [*Insert place here e.g. Bath*] (e.g. Bath Pump Room) via an integrated database.
Target market:	People with an interest in visiting [*Insert place here e.g. Bath*] for sightseeing purposes.
App features	• Login screen – User name and password • The app must contain multiple screens (3-5) using at least 2 different types of activity (e.g. empty activity) containing information about key attractions • The user needs to be able to interact with each screen via UI elements (e.g. buttons, text fields etc.) • You must be able to navigate between screens (e.g. using intents) • Information and media (e.g. images) on key sights in [*Insert place here e.g. Bath*] • GEO location information (e.g. Google Maps) about [*Insert place here e.g. Bath*]

	• The app must be user friendly and readable (e.g. use of fonts, layout, UI design) • The app must be consistent in style and appearance • The app must contain an SQLite Database storing information about key attractions • Ability for user to search database to return information about key attractions (e.g. queries) • The database needs to output data (E.g. via Logcat or on screen to the user). • The app must run on an android device or emulator • Programming – you must ensure that all of the programming concepts are used: Appropriate data types, selection (if, else, else if etc.,), loops (while, for), methods and classes. • App must be developed within Android Studio
Existing apps performing related function(s):	Travel apps: • Trip advisor • Culture trip • Google trips • Timeout
Requirements and parameters	Develop your app inside one 'project over' a period of 30 days. Once you have completed your app email the files to xxxxxxxxxx by 5th April latest.

Appendix C: Knowledge retention questionnaire

The knowledge retention questionnaire completed by all participants.

1. **Did you have any programming experience prior to attending the course? If yes, what?**
2. **Did you have any Android development experience prior to attending the course? If yes, what?**
3. **What is your name?**
4. **What type of data should the number 1.5 be stored as in an Android app?**

Mark only one oval.

Integer
String
Double
Boolean

5. **What data can represent the 2 values: true or false?**
6. **An Object is best described as a:**

Mark only one oval.

An instance of a class where it's sole purpose is to always remain static/unchanged
An object in an instance of a class which can be represented as a collection of data structures
The central feature of object oriented programming which is contains variables that can't be changed once set

7. **What is a method?**
8. **Which data type is useful for storing multiple values (data)?**
9. **The image and button 'views' have an onClick method**

Mark only one oval.

True
False

10. **Select the correct syntax for a while loop**

Mark only one oval.

while {x > 6} (System.out.print ("Hello world"));
while (x > 5) {System.out.print ("Hello world");}
while (x > 5) {System.out.print "Hello world";}

11. **Look at these statements (import android.os.Bundle; import android.util.Log; import**

android.view.view;) - what are they examples of?

Mark only one oval.

imported libraries

imported widgets

imported technologies

imported xml

12. **Apps in Android Studio are developed in:**

Mark only one oval.

html

Java

SAS

Python

Javascript

13. **What are layout files written in?**

Mark only one oval.

html

hypertext markup language

Java

xml

14. **Using a 'Constraint' when developing an app can be described as:**

Mark only one oval.

Another word for 'parent'

The opposite of a 'margin'

A views relative position to another UI element

15. **Describe what this line of code will do:**
ImageView.animate().alpha(0).setDuration(2000).rotation(200).translationXBy(-1500)

16. **A Google maps feature can be created and enabled in an 'Empty Activity' type**

Mark only one oval.

True

False

17. **Select the correct syntax for creating a variable to store an Image View**

Mark only one oval.

ImageView ImageView = (ImageView) findViewById (R.id.imagename);

ImageView ImageView = ImageView findViewById (R.id.imagename);

ImageView ImageView = findViewById (id.imagename);

None of the above

18. **Which line of code will pop up with the message: You've won the prize!**

Mark only one oval.
Toast.makeText(this, "You've won the prize!", Toast.LeNGTH_SHORT).show();
Log.i("Message", "You've won the prize!");
19. **Which Android class enables the app to move from one activity to the next (Intent)**
Mark only one oval.
Intent
View
Widget
os

20. **What best describes an Android 'Activity'?**
Mark only one oval.
A type of view, similar to a text view or image view
An app can contain only one of these and it's the interface
A screen within an applications interface
21. **Which best matches the definition of dpi**
Mark only one oval.
The total number of pixels of a screen
A ratio of pixels to the screens actual display size
Density independent pixel is normally the same regardless of the device
22. **Which of the following are examples of manipulating XML code for an activity:**
Mark only one oval.
android: layout_height = "fill_parent" >
EditText EditText = (EditText) findViewById(R.id.edittextEnterName);
MyDB.execSQL("CREATE TABLE IF NOT EXISTS table2 (street VARCHAR, name
VARCHAR)")
23. **User interface elements: buttons, images and edit text are part of which class?**
Mark only one oval.
Components
Views
UI elements
os.bundle
24. **Name 3 attributes of an edittext user interface element**
25. **If a programmer wants to align an image to the centre of the screen they could access which of the following to do this:**
Mark only one oval.
Attributes within the design view
The XML code

Both within the attributes section of a pages design view and the XML code

26. **Identify 2 important considerations when including images within your app**

27. **What will this statement do: 'openOrCreateDatabase'?**

28. **Why are index values important in SQLite databases?**

Mark only one oval.

They can delete the relevant data

They enable the user to reference the relevant data

They aren't actually important

29. **What does this line of code do when executed?: SELECT * FROM table**

30. **What does this line of code do when executed?: SELECT * FROM table5 WHERE name =='John' AND age > 40**

Appendix D: Skills mapping

Mapping of curriculum content against questions from the FDD task and knowledge retention questionnaire.

Bootcamp curriculum			FDD Task (22) *Evidence of…*	**Knowledge retention questionnaire (27)**				
Programming (Java)	Data		Appropriate use of data types	**What type of data should the number 1.5 be stored as in an Android app?**	Integer	String	Double	Boolean
	Selection		Appropriate use of selection					
	Loops (for and while)		Appropriate use of repetition	**What data can represent the 2 values: true or false?**	Short answer			
	Classes		Appropriate use of classes	**An Object is best described as a:**	An instance of a class where it's sole purpose is to always remain static and unchange	An object in an instance of a class which can be represented as a collection of data structures	The central feature of object oriented programming which is contains variables that can't be changed once set	
	Methods		Appropriate use of methods	**What is a method?**	Short answer			
	Development Environments			**Which data type is useful for storing multiple values (data)?**	Short answer			
	OOP Concepts		Evidence of a OOP principles	**The image and button views have an onClick**	TRUE	FALSE		

				method (True or False)				
				Select the correct syntax for a while loop	while {x > 6} (System.out.print ("Hello world"));	while (x > 5) {System.out.print ("Hello world");}	while (x > 5) {System.out.print "Hello world";}	
				Look at these statements (import android.os.Bundle; import android.util.Log; import android.view.view;) - what are they examples of?	imported libraries	imported widgets	imported technologies	imported xml
Android SDK	Android Studio			**Apps in Android Studio are developed in:**	html	Java	SAS	Python
	Intents		Evidence of intents used to move between screens	**What are layout files written in?**	html	hypertext markup language	Java	xml
	Activities		Appropriate use of activities	**Using a 'Constraint' when developing an app can be described as:**	Another word for 'parent'	The opposite of a 'margin'	A views relative position to another UI element	
	Classes	e.g. Toast, View, SQLite etc.		**Describe what this line of code will do to an imageview: ImageView.animate().alpha(0).setDuration(2000).rotation(200).translationXBy(-1500)**	Short answer			

	APIs		Evidence of API use					
	Libraries		Appropriate use of libraries	**A Google maps feature can be created and enabled in an 'Empty Activity' type**	TRUE	False		
				Select the correct syntax for creating a variable to store an Image View (see page 11)	ImageView ImageView = (ImageView) findViewById (R.id.imagename);	ImageView ImageView = ImageView findViewById (R.id.imagename);	ImageView ImageView = findViewById (id.imagename);	
				Which line of code will pop up with the message: You've won the prize!	Toast.makeText(this, "You've won the prize!", Toast.LeNGTH_SHORT).show();	Log.i("Message", "You've won the prize!");		
				Which Android class enables the app to move from one activity to the next	Intent	View	Widget	OS
				What best describes an Android 'Activity'?	A type of view, similar to a text view or image view	An app can contain only one of these and it's the interface	A screen within an applications interface	
App Design and Layout	Android studio	Creating a project						
		Project file manipulation						

		Android Studio Emulator	Evidence of a working app	**Which best matches the definition of dpi**	The total number of pixels of a screen	A ratio of pixels to the screens actual display size	Density independent pixel is normally the same regardless of the device	
		Android Studio Layout	Appropriate use of layout attribrutes	**Which of the following are examples of manipulating XML code of an activity**	android: layout_height = "fill_parent" >	EditText EditText = (EditText) findViewById(R.id.edittextEnterName);	MyDB.execSQL("CREATE TABLE IF NOT EXISTS table2 (street VARCHAR, name VARCHAR)")	
		Terminal, Logcat & Build	Evidence of logcat testing					
		XML Files	XML file manipulation					
	Building a UI	Creating UI Elements	Appropriate use of UI attributes	**User interface elements: buttons, images and edit text are part of which class?**	Components	Views	UI elements	os.bundle
		UI Element Attributes and Manipulation	Appropriate manipulation of UI attributes	**Name 3 attributes of an edittext user interface element**	Short answer			
		Interactive UI Elements	Appropriate use of interactive UI elements	**If a programmer wants to align an image to the centre of the screen they could access which of the following to do this:**	Attributes within the design view	Within the XML code	Both within the attributes section of a pages design view and the xml code	

	Images	Image properties	Appropriate image attribute manipulation	**Identify 2 important considerations when including images within your app**	Short answer			
		Associate classes						
		Image considerations	Appropriate image considerations					
Databases	SQLite Databases	Creating an SQLite database	Evidence of SQLite database	**What will this statement do: 'openOrCreateDatabase'?**				
				Why are index values important in SQLite databases?	They can delete the relevant data	They enable the user to reference the relevant data	They aren't actually important	
		Inserting Data	Data within SQLite databse	**What does this line of code do when executed? SELECT * FROM table**	Short answer			
		Queries	Appropriate use of database queries	**What does this line of code do when executed? SELECT * FROM table5 WHERE name =='John' AND age > 40**	Short answer			

Appendix E – Data summary

A data collection summary of the study's findings from both the FDD task and knowledge retention questionnaire.

Overall headline statistics

Number of confirmed participants: 10

Number of participants confirmed attendance at bootcamp: 8

Number of responses to questionnaire: 8

Number of participants completing task: 4

Number participants with Android development experience: 0 (0%)

Number of participants with prior programming experience: 3 (38%)

Knowledge retention questionnaire and FDD task results

Total questions: 27

Participant	Programing experience	Project	Group	FDD task (%) – see blue column of Appendix D	Knowledge questionnaire total (%)– see red column of Appendix D
A	No		A	N/A	52
B	Yes		A	N/A	41
C	No		A	N/A	59
D	No		A	N/A	44
E	Yes	Yes	B	59	67
F	No	Yes	B	73	93

G	Yes	Yes (no submission)	B	0	67
H	No	Yes	B	27	67

Average score of all participants: 61%

Range of all participants: 41% - 93%

Average of participants with programming experience: 58%

Average of participants without programming experience: 63%

Average of participants who completed the FDD based assessed task: 73%

Average of participants who didn't complete the FDD based assessed task: 49%