Holiday Extras

2017 / 18



My Year in Industry

John Taylor

Engineer

Abstract

This report will detail my journey as a placement student, all the way from finding and securing my placement through to completing my year. It will cover information about the business itself, the teams, the major projects I’ve worked on and touch on many key experiences that have contributed to an enriching year. I will explain what I’ve learnt throughout the year and how the experience has shaped me as an Engineer and even more so as a person.

Placement Details

**Name:** John Taylor

**Placement Dates:** 3rd July 2017 → 31st June 2018 (Contract extended until 31st August)

**Job Title:** Engineer

Placement Address

Holiday Extras Ltd,

Newingreen,

Kent,

CT21 4JF

Supervisor

Luke Hansell

Dev Lead

luke.hansell@holidayextras.com

Table of Contents

[1 Introduction to my Year in Industry 6](#_Toc519952907)

[1.1 Why Holiday Extras? 6](#_Toc519952908)

[1.2 Journey to Holiday Extras 6](#_Toc519952909)

[1.2.1 Initial Interviews 6](#_Toc519952910)

[1.2.2 Application + Interview with HX 7](#_Toc519952911)

[1.2.3 Offer from HX 7](#_Toc519952912)

[1.3 Holiday Extras: a brief overview 7](#_Toc519952913)

[1.4 My Job Role 8](#_Toc519952914)

[2 Working for Holiday Extras 9](#_Toc519952915)

[2.1 Web Team Structure 9](#_Toc519952916)

[2.2 Pod Structure 9](#_Toc519952917)

[2.3 Agile Process 10](#_Toc519952918)

[2.3.1 Sprint Structure 10](#_Toc519952919)

[2.4 What’s it actually like? 12](#_Toc519952920)

[3 Timeline 13](#_Toc519952921)

[4 Onboarding 14](#_Toc519952922)

[4.1 Weekly 1-2-1s 14](#_Toc519952923)

[4.2 Pod Rotations 14](#_Toc519952924)

[4.2.1 New Products 15](#_Toc519952925)

[4.2.2 API 15](#_Toc519952926)

[4.2.3 Customer Experience 16](#_Toc519952927)

[4.2.4 Finishing off 17](#_Toc519952928)

[5 Customer Experience 18](#_Toc519952929)

[5.1 Background 18](#_Toc519952930)

[5.1.1 People 18](#_Toc519952931)

[5.2 Notable events 18](#_Toc519952932)

[5.2.1 Void of Tom Price 18](#_Toc519952933)

[5.2.2 Welcoming of Jordan Clague 19](#_Toc519952934)

[5.3 What I did 19](#_Toc519952935)

[5.3.1 Booking History 20](#_Toc519952936)

[5.3.2 Reviews 22](#_Toc519952937)

[6 Tech Foundations (Dev stream) 24](#_Toc519952938)

[6.1 Background 24](#_Toc519952939)

[6.1.1 People 24](#_Toc519952940)

[6.1.2 Process 25](#_Toc519952941)

[6.2 What I did 25](#_Toc519952942)

[6.3 Notable events 26](#_Toc519952943)

[6.3.1 Steve’s sudden resignation 26](#_Toc519952944)

[6.4 Thoughts and feelings 26](#_Toc519952945)

[7 Projects 28](#_Toc519952946)

[7.1 Booking History 28](#_Toc519952947)

[7.2 Reviews 28](#_Toc519952948)

[7.3 Microservice Re-homing 28](#_Toc519952949)

[7.3.1 Challenges 29](#_Toc519952950)

[7.3.2 Thoughts and Feelings 30](#_Toc519952951)

[7.4 Dependency Graphing 30](#_Toc519952952)

[8 Other notable events 30](#_Toc519952953)

[8.1 Chauntry Acquisition 30](#_Toc519952954)

[8.2 Purple Parking Acquisition 30](#_Toc519952955)

[8.3 GDPR Enforcement 30](#_Toc519952956)

[8.4 Contract extension 30](#_Toc519952957)

[9 Conclusion 30](#_Toc519952958)

# Introduction to my Year in Industry

During the university selection process, the year in industry programmes really stood out to me, making it easy to say that I was excited for this year. It became one of the most pivotal factors in my decision to turn down King’s College London and come to Kent. The placement scheme sounded so much more fleshed out and enriching here. Having this passion before I had even made it to the year in the industry really fuelled my anticipation for the year.

## Why Holiday Extras?

It’s fair to say that before Holiday Extras came to present at the university I had never heard of them, so what was so enticing? Whilst it’s hard to honestly be excited about selling airport parking; the role, benefits, and emphasis they placed on their placement program really stood out. They also told us they only source students from Kent and that a lot of them had kept graduate positions, meaning that I’d already be surrounded by people of similar experiences.

Their entire presentation conveyed an image of an extremely relaxed and fun workplace with all the vibes of a start-up. They boasted about all the awards they’d received acknowledging them as one of the best places to work in the UK. They also had current placement students give their stories and experiences of their placement, all of which told tales of many learnings and great opportunities at HX.

Whilst all of this was promising I was initially reluctant to settle into a web-based role. I wasn’t completely against the idea but work closer to the back-end had always interested me more. Due to this I initially passed up the opportunity and only came back to apply when applications re-opened. By that time, I had realised that the work environment, benefits, and opportunities were too good to ignore.

## Journey to Holiday Extras

Having come into the placement process with no previous work experience I was at a complete fresh start without even a CV prepared. Luckily, by the time it came to my application at HX I’d already sorted out my CV, sent applications and even had an interview.

### Initial Interviews

The interview was with a company called SquaredUp who are a small start-up close to my home in London. Whilst it was useful to gain experience the process also knocked my confidence a little, they asked some fart-assy questions that put me on the spot more than expected. However, it also showed me that interviews are a two-way process, from those questions alone I knew that company was too ‘hippy’ to be a good fit for me.

### Application + Interview with HX

After putting in an application to HX I received an invitation to an assessment day. This day consisted of a group task, presentation, and individual interview. On the day I remember feeling nervous before arriving, but the relaxed environment helped me to settle in quickly once there.

We started with the group task which required us to plan out the handling of travel disruption caused by snow. We had to plan an email to customers as well design a system for the call centre to use as a means of reporting contact with customers. At first this was a little daunting to be working on with three unknown peers, but I quickly became distracted in trying to make the best impression possible; engaging with the group, conversing about ideas, and turning them into a full solution. When it came time to present our work the calm atmosphere helped me present confidently and I went to lunch pleased with my performance so far.

This quiet confidence in my performance so far really helped me going into the individual interview. In a complete contrast to my experience with SquaredUp this seemed to go a lot smoother. I didn’t feel that many questions really put me on the spot too much and I was able to be open about my initial reluctance to a web role. They had many re-assurances for me that there was enough modern technology in use to keep things interesting. This made me feel that they really understood where I was coming from and saw that my focus was on learning as much as possible. I appreciated what they had to offer in that regard.

I eventually left the day feeling pleased with the impression I’d given as well as being sure that this type of environment was for me. I began to realise how much this role would resonate with me despite my initial feelings.

### Offer from HX

After about a month I received a phone call telling me that HX would like to offer me the position. I accepted the role then and there because I’d spent all of the time since the assessment day feeling sure that it was the right fit for me.

## Holiday Extras: a brief overview

Holiday Extras started its life as the Apple Booking Company (ABC) when it was founded by Gerry Pack in 1983. It came to fruition when Gerry noticed a gap in the Airport Hotels market, they had previously been targeted to business travellers and never towards leisure travellers. Initially this was achieved through a simple booking platform in partnership with travel consultants who latched on to the idea; the new market area provided lucrative and convenient business.

Throughout 35 years of growth and a rebranding as Holiday Extras, the company’s offerings have expanded hugely, and they now specialise in a vast array of travel products. This includes airport parking, lounges, insurance, car hire, resort transfers and more. Now having around 800 employees situated across offices in Kent, Germany and Bulgaria it’s fair to say the company has grown with its success.

During those early years the company strived as a platform for travel agents to make bookings for their own clients, but April 2000 saw the launch of the HX website. Along with it came the ability for direct customers to make bookings. This was a large step at the time as most travel companies were only targeting one market or the other, not both. Nowadays over 90% of bookings come via the website with less and less reliance on the call centre for business.

### Should get something about awards in here somewhere? Might be in Section #2

### Need to add a description of my objectives for the year or what I want to gain, expected outcomes etc (job confirmation, initially few but gained more about communication etc)

## My Job Role

When joining my official job title was as an ‘Engineer’ which kept things open to interpretation. It was difficult to predict exactly what that might encompass initially. However, it soon became clear why it was left open with each team serving different purposes through different platforms. It was clear there would be many opportunities for me to learn across all of these different areas, platforms and codebases. I understood why ‘Web Developer’ didn’t truly fit the bill of this job despite that being the main throughput.

During my time with HX I managed to gain experience across a variety of systems and I ended up truly appreciating the fluidity of my role. Not only did I have the opportunity to work on the front-end of the site, but my manager actually planned my second team to be a challenging back-end focused role. This gave me a taste of both worlds and really helped to push my learning to a higher level, I’m thankful that my own development was such a focal point for the company.

# Working for Holiday Extras

This section will detail the structure and process of HX as well as how I feel about my time spent interacting with it as a whole.

Firstly, to pre-face, despite the fact HX has ~800 employees there are ~150 of us in the actual web team where I’ve spent all of my time. Therefore, the majority of what I speak about will be the structure and process of this web team as a subset of the entire team. This also affects how much I enjoyed my work at HX, the web team is held on a pedestal and treated the best out of the business. We’re given fun events, encouraged to attend subsidised conferences and provided the flexibility do these things. Whilst some roles in the CC only have small windows of time to actually enjoy what makes us such a “fun start-up”. So yes, *I* am the 1%.

## Web Team Structure

The web team is one of the biggest areas of focus within HX since there is a direct link to customers and a huge potential to drive growth and profit. It would be fair to say the work that’s done within the web team has some of the biggest influence on HX’s success.

The entire team is split up into smaller sub-teams known as pods, which were recently renamed to ‘teams’ but this hasn’t caught on yet. Pods were originally focused around a single area of the business, taking on all work that relates to that area. For example, the euro pod would pick up work that would improve the experience for our European customers. This approach allows each pod to work independently in a self-contained manner, with cross-pod communication still available for larger multi-pod projects.

There’s also the concept of ‘pop-up pods’ which would be created to achieve shorter term goals that either, didn’t fit into a single pod or needed multi-disciplined engineers to achieve. A good example of this is the Personally Identifiable Information (PII) pod, which started when we had <40 days to fix something which Google would detract our SEO rankings for.

Within the last few months of my placement the company began making a shift towards ‘mission-based’ pods. The intention of this was that pods would spawn dynamically around missions that the company wanted to achieve. This makes the pods much more reactive and allows fluidity with the assignment of engineers to missions that they suit best.

I personally have remained within a pod that isn’t truly mission based so I’ve not experienced it first-hand but it sounds like a sensible approach.

## Pod Structure

As much as the web team is built of pods each pod contains a concoction of roles that make the pod successful. The exact structure varies per-pod, but most pods contain a few engineers, a software tester, an agile coach and a product owner.

**Engineers**

The engineers within a pod are one of the main workhorses. They spend the majority of their time writing code to actually implement new features or fix issues. They’re also involved within the planning of work by helping use their experience to estimate the difficulty or time investment of certain pieces of work.

**Software Testers**

The testers within a pod are responsible for testing the work implemented by the engineers, they will ensure it works as expected in a variety of environments and scenarios. This is especially important within HX’s complex systems as there are often use cases that engineers might not have considered. Testers also typically offer a huge wealth of knowledge about the business and the complexity of all systems combined.

**Agile Coach (AC)**

The agile coach within a pod typically manages the workflow of the pod on a day-to-day basis. They will ensure that the agile process ([detailed in 2.3](#Agile Process)) is followed and that everyone can achieve what they’ve set out without issue.

The agile coach within a pod manages the workflow of a pod on a day-to-day basis. Ensuring that everyone in the team can work efficiently and effectively together without being affected by blockers on their work.

**Product Owner (PO)**

The product owners are responsible for managing relations with stakeholders, communicating and setting goals for the pod. This ensures the work being done suits the business and stakeholders’ needs.

Each of these roles play a vital part of each pod at HX and when combined they form a strong core structure for the pod to thrive off of.

## Agile Process

Every pod within the web team uses an agile approach which aims to deliver work quickly in an incremental pattern. This has become common within technology where the flaws of older methodologies, like the waterfall model, have been highlighted by the adaptive and reactive nature of modern technology.

### Sprint Structure

At HX this process manifests as bi-monthly sprints which aim to be a self-enclosed two-week section of work.

{{{{{ Sprint Diagram }}}}}

#### Sprint Planning

As demonstrated by the graph above the sprints have a clearly defined start and end. Most pods in the web team use JIRA to keep track of tickets and will store a backlog of all of the tasks they need to get to. This backlog is used when planning out exactly what should be done for the next two-week sprint. Tickets are ‘pulled in’ to a sprint based on the priority given by the PO, engineers and testers will work together to discuss what they think can realistically be accomplished.

#### Development Stage

Once a set of tickets have been agreed for the sprint there is then two-weeks of continuous workflow where every day the pod will meet to discuss progress. At this point the engineers will pick up tickets from the sprint backlog, working towards completing the requirements specified on the ticket.

#### Pull Request Review Stage

When a developer completes a piece of work he will open a pull request (PR) which can then be reviewed by at least one other developer. It will be checked to ensure the code has been written to a high standard and that it looks suitable without introducing avoidable technical debt. If the reviewer(s) deem it acceptable then the code will move on in the process to the testing phase. If not, then the original developer will usually discuss the feedback given and adapt the code as they see fit. This cycle continues until the code is deemed acceptable.

#### Testing Stage

During this phase the tester(s) of the pod will prove that the code works on their own machine in a range of scenarios. They will typically test it across different browsers and environments to prove thoroughly that it works as expected. This helps to cover areas that the engineer wouldn’t have tested and even use cases that they may not have known about. If there are any problems at this stage it will be sent back to the engineer to be investigated and addressed. If the fix involves code, then the process jumps back up to the development stage and has to come back through review. Eventually when the requirements have been met and no issues are found during testing it will be deployed.

#### Deployment Stage

Due to the closeness of testing and deploying this stage will usually be handled by the tester. There are two main steps of deployment, one of which is to deploy to staging and another from staging to production. The staging environment exists as a sandbox, hidden from customers but providing a more realistic testing environment than a local machine. This is another reason that deployments are commonly handled by testers, because they can use the time in staging to give it one final test before it goes live. As soon as a production deployment has been finished the change will be visible to the customers.

#### Mid-Sprint Review

During this iterative deployment approach work will be deployed frequently so some pods find a mid-sprint meeting beneficial. This meeting is used to look back on the progress that has been made so far, evaluate any blockers, issues and whether the sprint is still achievable. In some cases, work will be removed from the sprint to align priorities that need to be completed, or extra tickets could be pulled in if the sprint is going well. The agile coach will play an important balancing game during these meetings to ensure the sprint stays in equilibrium.

{{{ Post-sprint DEMO to stakeholders }}}

#### Post-sprint Retrospective (retro)

At the end of the sprint there will be a retrospective session that will look back at the sprint as a whole, evaluate what went well as a pod and what didn’t go so great. The agile coach will often prepare unique ways to run this meeting, but the goal is ultimately the same; to get everyone talking honestly and sharing their opinions. From the back of these meetings the pod should make adjustments in the future sprints to improve. I personally struggled with the retros in the Customer Experience Pod, every two weeks we’d have the same discussions, and nothing changed for months. They became dull and dreary and I had given up on the process actually making a difference. I haven’t since been in a pod that has actually had retros, so I can’t say I’ve seen a first-hand contrast in another team.

## What’s it actually like?

In many ways Holiday Extras does manage to live up to its ethos and start-up vibe. Even without having worked anywhere else I can still recognise the uniqueness of the workplace. The entire open plan office has an air of friendliness where the atmosphere stays relaxed, and everyone is approachable. This really helped me in my early days when I was adjusting and settling in to the office. The ability to go to anyone in the web team for help in such an easy and inviting environment really sped up my initial learnings. I feel thankful to have had that opportunity when I needed it most.

Even more than the casual discussions around desks that keep the day-to-day entertaining there are plenty of fun events organised. As early as my first week with HX we were outside playing tennis and eating strawberries with cream to celebrate Wimbledon.

<http://take.ms/ESIEt>

These events have continued throughout my time here including recently bush craft, human foosball and a rounders tournament. It’s been really refreshing to attend the events just to get out of the continuous work mode and get to know colleagues better.

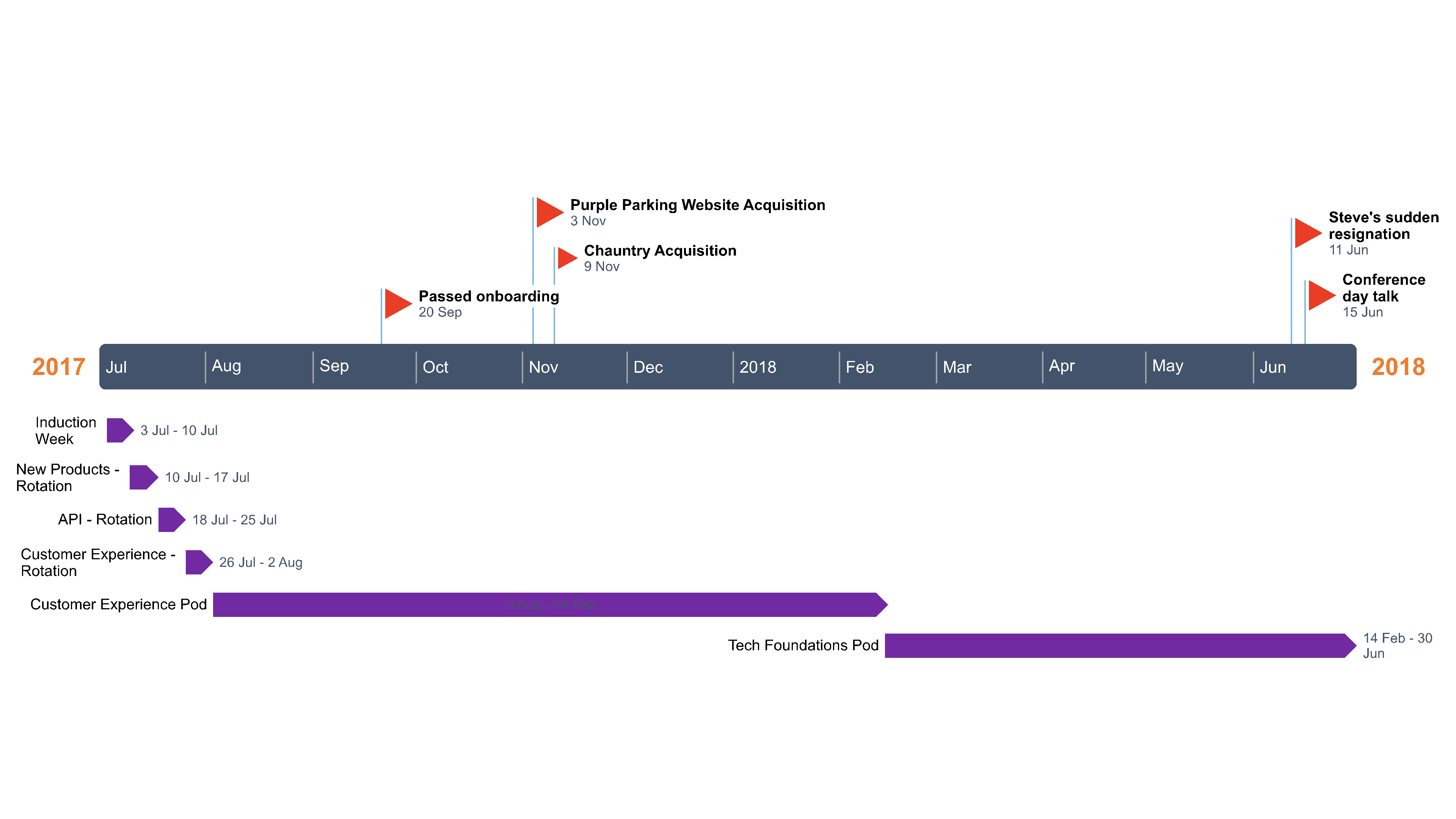
Besides the organised events there are often impromptu goings on and treats hosted for us. For example, we were given mince pies around Christmas, our own Easter eggs at Easter and frequent free ice cream throughout the summer. It’s nice to receive these little boosts throughout the day and further shows the uniqueness of the HX workplace.

We also have the hosted ‘shout outs’ every Thursday afternoon where directors or even the CEO himself will update us on the business’ progress, our success against the budget, any new starters, leavers and more. Another key feature is the calling out cards which are essentially nominations given out for notably displaying the HX values. I can remember being called out as part of the Customer Experience pod and appreciating the recognition and gratitude for my work. It definitely goes a long way to making hard work feel worthwhile when it’s noticed and rewarded.

Another huge element of HX that makes it stand out as a unique culture is the Project Lounge (PL) Fridays. These occur on a fortnightly basis and offer an entire day just for personal development or working on a project to help push the business forward. It’s great to be given this free reign and have the ability to make a difference through creative ideas. I benefitted hugely from these during the start of my placement since I could spend an entire day digging deep into HX’s technology and frameworks. I’ve continued to enjoy PL time throughout the year and have found the time extremely beneficial to my own development.

In summary, I’ve found the environment at HX has really aided my overall development throughout this year. My technical skills have been improved because of how comfortably I can approach people and gain insight easily. Whilst my personal skills have improved tenfold, especially moulded by office chatter and special social events helping me network with colleagues.

# Timeline

This section will give a quick overview of where I’ve spent my time during my journey at HX. Providing context for the next few sections of the report that will detail my experiences during onboarding, time with different pods and other notable events.

# Onboarding

When jo­ining HX there is an initial three-month onboarding period in which a basic set of objectives must be achieved to ensure you’re fit for the company. This period had a high focus on our individual learning as Industrial Placement Students (IPs).

{{{ Talk about my introduction week, laptop setup, time spent shadowing Jamie+Becky, laptop setup, repo setup.. shock of how many projects I’d need etc }}}

Workshops were hosted for us every Tuesday and Thursday which provided us with knowledge about Node, git, frameworks like React and specific platforms at HX. Some of the earliest sessions were overly basic but brimming with HX context. Although I can remember them ramping up in difficulty, with a talk by Mark Fermor about containerisation with Docker being far too complex. Looking back, my start at HX would have been a lot slower had it not been for these classes at the beginning.

## Weekly 1-2-1s

Throughout this period, I also had to meet with my manager on a weekly basis for what HX calls a 1-2-1. At first this made me a little nervous, coming from no work experience I’d never had a manager to report to about my progress. However, after my first 1-2-1 I realised there wasn’t much to fear and that it was actually quite casual. They felt like more of a catch-up with gentle guidance around benefitting more from my onboarding time.

Looking back, I can recognise that these sessions helped me considerably. Before joining HX I struggled to reflect and envision ways I could improve myself as an individual. This is why I came into my year with simplistic objectives (((Describe my pre-year objectives during my intro and link back to section))) about gaining experience in the industry. It was the realisations that I came to in my reflective 1-2-1 sessions that highlighted key areas where I could focus my efforts and improve. I can remember discussing my communicative weaknesses early on and being able to evidence improvements just weeks later. The reactive nature of these meetings proved extremely useful and having them every week enabled me to adapt quickly during this period. This set me leaps and bounds ahead in comparison to when I first joined.

## Pod Rotations

Following my first introductory week I spent three weeks rotating into different pods, gaining insight into what they each did and how they worked. This also offered a good opportunity to get to meet other members of the team, building up a repertoire of familiar faces. As part of this experience our manager had us present to himself and the other IPs at the end of each rotation. This gave us an opportunity to share in each other’s learnings as well as practice our presentation skills.

### New Products

For my first rotation I spent a week with the new products pod. They focus on creating new ventures for the company outside of the traditional core products. This involves a full flow from conceptualisation, creation, market launch and maintenance during the initial growth. Some of their more successful products include rental car hire, resort transfers and FastTrack passes, even if those passes were sold at cost price for the first 6 weeks.

During my time with new products I managed to get my hands stuck in and helped work on some tickets whilst also being exposed to more of the sprint process. Just within my first two days they had their end of sprint demo to their stakeholders, followed the next day by the planning of their upcoming sprint. At first it felt strange to be attending their serious meetings, but they were very accommodating and didn’t expect much from me, it was more to give me the experience and see what they were like. I appreciate the exposure that I gained from this when I look back on it.

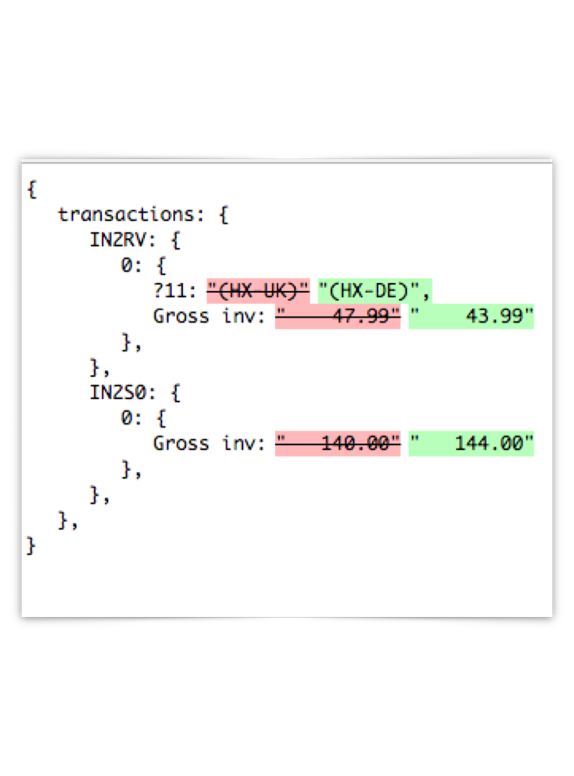
### API

Having gotten to grips with the work I was given in the new products pod and beginning to understand the sprint process I was excited coming into my week with the API pod, especially seeing as it had the potential to be more technical than I’d seen so far.

During the week that I spent with the API pod they were fairly far into a large project called Thunderbird. This project aimed to merge together our UK booking system with its German counterpart so that we had a single central platform, rather than having a difficult to expand system-per-country. The unified result of this project would make it much easier to globalise HX’s business.

To achieve this, we would spend time every day at 2pm comparing diffs between responses from the live systems with the experimental unified system. Over time this would highlight every single difference there is until the responses are always the same. I enjoyed getting involved with this work since it was a big get together as a team, all working to investigate the differences and seeing what needed changing and where. This helped me expand on my group communication at the stages where it was becoming clearer as one of my weakest areas.

Aside from this I also managed to get my hands dirty and picked up some quite technical work within the pod. They already had figured out the diff process for the responses from both systems and asked me to expand this for NOMREP files. I had no idea what the NOMREP files were at first, but George Bates helped get me up to speed explaining that they were essentially transactional logs in an ugly tab-separated format. By the end of the week I had successfully implemented a diff process for these files which produced readable output in the browser with colour highlighting, shown below.



I was really glad to have been able to get stuck into this challenging ticket with the pod due to my concern about whether the placement would be technical enough for me. It had me re-assured that there was still plenty on offer for me at HX despite some pods having a large web focus.

### Customer Experience

{{{ Should I actually explain what CEX is here seeing as they’re my next permanent pod section??? (explain and link to the section where I go deep) }}}

My final rotation was into the customer experience (CEX) pod which felt like a bit of a strange choice in this process, seeing as they would be my first permanent pod anyway. I would have preferred the opportunity to be immersed into another pod’s work to broaden my horizons further. Despite this I still managed to enjoy the week getting to know my final pod and gaining a head start into working with them.

I spent the majority of this rotation pairing up with Tom Price who as an ex-IP was quick to offer me assistance, running me through CEX’s projects as a whole and pairing with me on my first ticket. This really helped me kickstart my week with them and lead to me completing a few tickets with a reasonable level of independence. This boosted my confidence in my technical ability at this early stage, leaving me feeling more comfortable to work on my own even so early on. I’m glad HX offered the opportunity for this autonomy. {{{ Could quote the feedback tom gave me on this? }}}

The one ticket that stands out to me was a re-design of the CARE login page, this is used daily by the call centre (CC) staff and hadn’t been changed for a long time. Tom regretfully admitted to me that the original was actually something he had made in a rush before. The reason this ticket stands out is that it was done during one of CEX’s CC sit-in days, where we would work from desks within the CC, so they could approach us to raise issues as one of the pod’s “customers”. This allowed me to spend time on a few designs of a new login page and actually go and collect opinions from the CC. Initially it was quite intimidating to approach people in the CC but Han had offered to support me and it ended up being fine. I feel that this ticket was extremely beneficial to me as it showed me the importance of the design stage, as well as communicating with the customers at the earliest opportunity. My final product could have been a lot worse had I not sought out their opinions before the actual implementation.

{{{ Can’t find an original gabble screenshot }}}

[[[CEX-1418]]]

### Finishing off

As the onboarding period drew to a close it was time to prepare for my review with my manager, aiming to prove that I had achieved the objectives set out for the three months. A lot of this was done based on feedback collected through the 360 system, which is used to collect opinions from up to ten colleagues. I luckily received a lot of good feedback from the people I’d worked with closely and those who’d seen me during rotations. On top of this, I discussed the huge improvements in my communication over a short course of time, which Luke agreed with also.

After successfully passing onboarding we were welcomed to the team in the week’s shoutout where we were each given a HX hoodie and bottle of champagne. It felt nice to be invited into the team with this announcement as a recognition of a good start.

# Customer Experience

Having already spent a week with CEX during my onboarding rotations my actual joining of this pod didn’t feel hugely ceremonious. Although I was still glad to make a start on settling into my long-term pod, saying goodbye to the learning intensive onboarding period.

## Background

The customer experience pod (CEX) are focussed around fixing customer-raised issues and enhancing their interactions with us, as well as supporting operations in the call centre to improve how well we can serve customers. In a realistic world, the vagueness of what can improve a customer’s experience leaves some odd work filtering down to the pod. The primary focus during my time in the pod was to encourage customers to self-serve rather than relying on the call centre for support.

### People

The pod had a fairly typical structure and process according to the agile process I’ve already described ( {{{ link }}} ), originally consisting of:

* Mark Cridland (Product Manager)
* Han Cork (Agile Coach)
* Becky Cox (50% Tester)
* Jamie Matthews (Tester)
* Mike Holloway (Remote Engineer - UK)
* Adrian Kateliev (Remote Engineer – Bulgaria)
* Kiril Panchev (Remote Engineer – Bulgaria)
* Tom Price (Engineer – UK)

The people within the pod ended up changing far more frequently than it probably should have due to re-structures, resignations and maternity leave. Therefore, I’m not going to detail every change to this and instead highlight only ones that made a pivotal difference in my year.

## Notable events

During my 7 months with the pod there are a few key events that shaped my experience, highlighting my strengths and weaknesses throughout the ups and downs.

### Void of Tom Price

After only one day of officially joining the pod it was announced Tom Price would be moving. They needed his resources in a pop-up pod to fix a data violation that google would punish us for in SEO rankings. This shook some of the confidence I’d built as Tom was one of the people I’d worked closest with thus far. I was a bit worried about now being the only engineer within the pod that works from the office, especially trying to fill in where Tom’s vast knowledge was.

I knew I had one week left with Tom in the pod and had the easiest access to him in the office, therefore I turned his departure in an opportunity to cement myself within the pod. I spent the rest of the week working with him and had arranged a full handover on the Friday, where he ran me through all of the stuff he was in the middle of including handing over PRs directly to me. I tried to gather as much knowledge as possible to carry forth for the pod’s benefit. This settled my nerves a little and left me feeling comfortable as what I was calling the “Budget Tom Price”. I was further re-assured during my following 1-2-1 where Mark and Han had fed back to Luke the value they found in my investment.

### Welcoming of Jordan Clague

Jordan joined the pod just before Christmas and it fell down to me to spend time getting him up to speed. We spent all of the day together, with about two hours dedicated to talking through what the pod does, who is involved and what projects we have. Moving on to helping him setup codebases and getting him through his first ticket.

Looking back, this actually provided a lot of value to me. It helped me see how much I had improved with my communication even in a mentoring scenario I hadn’t really experienced before. I’ve also received good feedback from Jordan about this experience:

*“John was the pod member who walked me though my pod's previous (and current) work when joining the pod and sat with me as I worked on my first ticket and set up numerous systems. At the time, I wouldn't have assumed John was an IP due to how confident / assured John was.”*

***- Jordan***

This feedback amplified the overall experience and proved that my focus on improving my communication was paying off. I appreciated the opportunity to gain experience as a mentor.

## What I did

During my time in the pod I managed to pick up quite varied work as was the nature of CEX’s tickets. I found this useful as a new starter since it gave me a good breadth of knowledge about areas across the business. Having said that, I also managed to find opportunities to cover some areas more thoroughly through projects that I took the lead on.

The broader areas of my work in the pod involved picking up tickets on Render, which was a front-end platform we were the owners of. This means that we had some semblance of responsibility for maintenance, enforcing best practices and playing a role in the bigger decisions surrounding its future. We also consistently worked with HAPI, which is HX’s own API for handling bookings, product lookups, supplier info and other data retrieval to support other platforms. There were also projects further from the core that we worked on including gabble, hammertime and some of our own purpose built microservices.

{{{ Contrast paragraph linking to projects section }}}

### Booking History

One of my earliest projects was the new booking history service, this was handed over to me from Tom Price and ended up being my sole focus for my first two to three months in the pod.

The booking history service aimed to form a new centralised source of booking information. Before it’s conception booking data was siloed in to a few different data stores across platforms, making it hard to retrieve a holistic set of data on bookings. The pre-existing method to look up bookings involved talking directly to CHIPS, the legacy booking platform at the centre of our operations. Besides limitations in the expandability here as CHIPS is maintained externally, it’s also an unnecessary load added to an outdated platform that won’t always handle it gracefully.

With this aim in mind the solution took the form of a microservice using our in-house dockyard architecture, which consisted of tools that aimed to make it simple for engineers to create, deploy and maintain smaller individual codebases. This contrasts hugely with some of HX’s core projects, like HAPI, which were monolithic projects with little separation of responsibilities. The booking history service would actually become one of the earlier projects built on this architecture offering. It was exciting to be building on top of brand new technology to HX at the time.

When Tom handed over to me the service had actually already been started by him, but he hadn’t pushed too far into it. He talked me through it during the handover I arranged which enabled me to pick up exactly where it had been left.

The overall design of the service included listening to events from our in-house purpose-built data pipeline and populate a MySQL database with the information provided. The pipeline provided a way of sending and receiving events, where we would be relying on the “server\_ecommerce” event being sent when a booking is made. We would need to listen for these events and extract data from them, which would be stored into the database.

#### Challenges

Although the overview of this service seemed to be quite simple there were still quite a few challenges that cropped up along the way. Some of these stemmed from my lack of experience but equally there were issues experience engineers struggled to explain.

Initially my biggest challenge was simply getting to grips with NodeJS, the entire microservice implementation at HX and especially the asynchronous nature of this application. All of these are things I hadn’t dealt with in great detail before and had to pick up for the first time during this project. All things considered, I would say this actually ended up going quite well. It only took me a few weeks to be running with this project and have a good grasp about the main concepts employed. The confidence gained here helped me to feel comfortable with being at the forefront of a big project so early on at HX. I’m thankful for the opportunity to have been trusted so early on.

##### Duplicate Events

There came a more serious challenge when we seemed to be receiving duplicate events from the data pipeline. This was detrimental to the service as it had the potential to pollute the data, creating inconsistencies that would be unfavourable when trying to create the single source of truth about bookings. So much so that it actually ended up being a focus for an entire month. I had to spend a lot of time investigating the exact cause of the duplication and struggled to come up with anything. I can remember looping in more experienced engineers, including those who created the data pipeline and the microservice architecture and none of them had an answer for me either.

Eventually my suspicions lead to there being a deeper issue within CHIPS, moreover surrounding the webhooks that CHIPS sends to our systems when a booking is made. This would fall on the external Chauntry team to be fixed, but unfortunately, they were completely occupied with the ongoing thunderbird project. This meant that the problem wouldn’t be fixed at its core for a long while, we had to work our way around this.

My original proposed solution was to implement a cache for the service. This involved caching every booking that we received, as well as checking the cache before ever storing a booking. This *should* have allowed us to eliminate duplicate entries from ever being allowed to enter the database. I discussed this with various engineers around the business before starting it and everyone agreed that it was a sensible solution. However, once this had been implemented and tested it proved to be only partially effective. This puzzled me for a few days, as well as the engineers I spoke to about it. On paper and during local testing this change was undeniably a suitable solution. However, it just didn’t behave correctly in production, we suspected this was to do with the fact the service ran across multiple instances in production causing a race condition. At this point I’d become frustrated with the project, I constantly kicked myself for not being able to provide a perfect solution.

Finally, I concluded that nothing right now could solve the duplicate event issue on the ingested data. Therefore, I decided to fix the issue by re-structuring the database. It would now accept multiple entries for a single booking, but when requested we would **only** provide the most recent information. This didn’t feel very clean, but it was the only true solution on the table in the timeframe, especially with the Chauntry team being completely occupied.

##### Stakeholder Pressure

During the difficulty with the duplicated events there was also pressure from above, our stakeholders were struggling to understand the value of booking history due to the time it was taking. I took this upon myself and spoke to all of the people I knew who were waiting for it to be ready. I got them all to send me the reasoning behind what they wanted and why, providing this to Mark and Han to help them convince the stakeholders. I received feedback from both of them that they really appreciated the initiative I showed, and it helped a lot with pursuing the project to completion. This lifted my spirits and helped me get back on track instead of being in a rut of frustration.

#### Thoughts and feelings

This project came very early in my placement at HX and provided me with a good set of learning opportunities, especially surrounding the microservice architecture at HX. I believe that it really helped me build confidence early on and establish myself within the pod and the wider web team.

I was quite lucky to start with microservices as they are self-contained and don’t come with years of legacy bundled into them, making the initial understanding much easier. This sped up my early progress and enabled me to showcase my abilities to the pod, building their confidence in me owning this project. I don’t think an entire experienced pod would put this in my hands at this stage had I not been able to prove my confidence.

At the same time, this project wasn’t without it’s difficulties. Throughout all of the problems described I noticed that I picked myself apart too much. I can remember frequent drives home during the duplicate issues where I would kick myself for not solving it that day. This frustration wasn’t beneficial to anybody and would only put my mental state in a rut. In future I should be more aware that this negativity doesn’t actually help, instead focusing on things in a different more positive light. Especially considering this was a problem that experienced engineers couldn’t figure out and agreed with my proposals on.

### Reviews

Towards the end of my time in CEX I got involved with a project to make our reviews more transparent and clear. Mainly, our stakeholders wanted us to base our ratings and displayed reviews off of the last two years of data. Previously reviews were historical from the date of the product’s conception, which meant that if it had been changed recently it would be hard for that to be reflected in reviews.

Towards the end of my time in the pod I got involved with investigating issues surrounding reviews on the website. Our stakeholders wanted us to only show reviews that were within the past two years, rather than since the products conception. Sounds simple enough, right?

Unfortunately, this ended up being a complicated issue riddled with challenges caused by legacy tech. I spent a lot of time with Ricardo, who had experience with the original system, digging deeper into where reviews come from and how they’re stored. The information was collected via surveys sent to customers after their trips, it included a basic rating of the product, rating of HX and any comments they might have. All of which was fine as a collection source, however the way the database that stored this had been designed was not fit for our purpose.

Rather than containing a persistent store of all reviews the database stored a calculated result which was deleted and replaced daily by a cron job. The fact that this was an overwriting operation and continuously erased previous data there was no easy way for us to just query for two years of data. I ended up working closely with Adrian on this and we tried to create a second table containing summaries for two years of data. This began to work sort of as expected, but it also exposed some deeper underlying issues with the way reviews were originally created.

Due to these complications I ended up having to leave the pod without finishing the project. This was disappointing, it’s not easy to walk away from a project and not have closure on it. I did manage to reduce the frustration of this by arranging a handover to Jordan who’d joined the pod two months prior. I have since seen him carry the project to completion, which laid the project to rest once and for all in my mind.

Overall, I feel that this project actually went reasonably well. There were issues with the legacy tech and the archaic database design but, it eventually was solved with the help of my investigation and partial solution. I think one of the key results was also how it helped establish me across the wider team rather than just my pod. This was shown by the fact that I became the “go to guy” for a while on this subject, especially when Ricardo, who I originally sought help from, started sending people to me. It was also a final proof for my communication-based objectives at the time, since there were a lot of new faces involved seeking updates, explanations and discussions from/with me.

{{{ Thoughts and feelings of the entire pod experience }}}

* Loved the people, great team, if I was here forever I would want to always stay with them
* Spent too long there but….
* Work wasn’t always exciting and it could be annoying tweaks and fixes

# Tech Foundations (Dev stream)

My second pod move felt long overdue by the time I eventually switched to Tech Foundations, I’d become stale where I was and was extremely excited for the change. Especially being invited closer to the back-end where I had always originally envisioned myself.

During the week that I joined the pod was actually splitting up into three streams. The original Tech Foundation pod grew large and had multiple separate focuses, therefore it was split into Dev Foundations, UI Foundations and Data Foundations. The specific stream I’ve been focused in is Dev Foundations.

## Background

The Dev foundations stream focuses on helping other engineers ship code quickly and with the best processes possible. This is achieved by providing easy access to infrastructure and purpose-built toolsets, especially in the form of the microservice architecture.

The microservice architecture and all of the tools surrounding it are actually the biggest single focus of the pod. Throughout my time in the pod almost everything we’ve been working on has revolved around maintaining, improving and expanding our offering to the team. This included our own in-house self-built CI service, dockyard-deploy scripts, dockyard-tools like dockyard-rpc and many others.

### People

This pod had an atypical structure compared to the rest of HX. We didn’t have an individual Product Manager and Agile Coach, instead we had Steve who acted as a role somewhere in between. Our entire structure was:

* Steve Coppin-Smith (“Pod Lead”)
* Oliver Rumbelow (Engineer)
* Mark Fermor (Site Reliability Engineer)
* Khusro Jaleel (Site Reliability Engineer)

This list is quite a bit shorter than what I laid out as a ‘normal’ pod at HX earlier in this report. Not only did we have Steve as an overall pod lead, we lacked a tester and two members of the team had a unique SRE role.

The SRE role has a deep focus on the actual infrastructure behind what we offer to the rest of the team. They are in charge of technical setup, maintenance and management of our staging and production environments, the processes of deployment to them and some of the steps in between. In all truth, they had a wide-ranging set of responsibilities due to how important the infrastructure was as part of our microservice offering.

The Engineers in the pod (Myself and Oli) were focused on the areas which required larger portions of code written. This mainly covered tools and interfaces between the engineers and the infrastructure we were providing.

I feel that I actually landed in a unique position within the pod. I was really the only engineer that had recently worked on the other side of the coin, consuming all of these tools rather than creating. As well as the fact I maintained a connection with engineers from the rest of the business. This meant that I was surrounded by people full of feedback on the tools we were providing, and I had a good grasp on features that they wanted offered. I found it extremely useful to be able to bring these viewpoints to the pod, applying it to the work I undertook.

### Process

The process within dev foundations was a complete contrast to what I had experienced in CEX and how the majority of the web team operates. We didn’t use jira to keep track of tickets, our only big meetings were bi-weekly demos and there was less dictation from above. Every engineer in the pod functioned with a lot of autonomy, taking control of the entire lifecycle of tasks from identification, design, implementation, deployment and maintenance. As long as what we were doing provided value against our current quarterly OKRs we could decide entirely solo what to work on.

Having come from a pod where stakeholders chose exactly what they wanted done each sprint this was a completely new experience. It initially took me some getting used to, it’s hard to start self-identifying what needs to be improved without an initial grasp on some of this tech. Luckily, Steve seemed aware of this and during the early days he had Oli help me find things to do. Over time I settled in to the approach and actually got quite good at self-organising within the pod, keeping all possible tasks on a todo list in slack.

## What I did

The tooling we offered to engineers was comprised of various different services, scripts and codebases, meaning that I managed to gain experience in a wide spanning set of projects. Especially encouraged by the self-driven nature of the work where I could find and improve any aspect of the process.

Some of the codebases that I worked with most frequently were the ci-service, infrastructure-service and access-service. Although outside of larger projects I’ve managed to delve into many areas, including scripts written to automate annoying tasks. One example of this was a script I wrote to audit exactly which version of the node-toolbox was being used across all 100+ microservices. This helped us know who was far out of date and would come into issues as we needed to release critical updates.

Outside of the smaller improvements and fixes made across all codebases I also had some key areas of focus. The two most notable examples are the access-service and dependency graphing projects that I worked on, both of which are detailed in my projects section.

## Notable events

I felt my time within dev foundations was actually a lot less eventful than CEX, in terms of people or process changes. It contrasted hugely from CEX where It felt that every new sprint came with a new resignation, pod member leaving or completely new faces joining. There was only one event that actually stands out in dev foundations.

### Steve’s sudden resignation

Steve’s resignation came in extremely short notice and was a shock to all but his closest friends. He spoke to us privately on the day of the announcement to explain some of his reasoning before almost disappearing within the same day, taking his last bits of annual leave before he went. I didn’t see this coming even though I’d personally observed some of the areas that frustrated him, including the bastardised agile process that was too far from the original agile manifesto for his liking. He’d done an amazing job of applying a more barebones process to our pod, but the business wouldn’t let this approach spread further.

I was concerned initially about the pod without Steve but he truly had made us highly autonomous in a way that would allow us to continue without him. So far, we’ve clearly managed to prove that we’re capable of this to our stakeholders as we still operate without a pod lead to this day.

This entire experience was surprising since it came out of the blue, but I feel the fact I’ve managed to continue delivering and supporting my teammates is a testament to how the year has shaped me. It shows the level of confidence, self-organisation and drive I’ve developed to be able to continue thriving in an autonomous workflow independently. Steve even provided me feedback upon his departure that reassured me about the value I brought to the pod. This left me in a confident position to carry on without him.

## Thoughts and feelings

Overall, I feel that the dev foundations pod has given me exactly the technical challenge I was looking for. It’s been an incredible experience to work alongside such talented engineers and gain insight into the deeper world of infrastructure. I’ve benefitted hugely in terms of knowledge but also the confidence it’s given me, being able to hold my own even surrounded by some of HXs most experienced technical engineers. I have faith in myself, my work and the rest of my career in this area thanks to the opportunity.

I’ve felt especially lucky to work with Mark and gain insight into his deeply technical world. He’s willingly taken me under his wings to show me about containerisation, kubernetes and cloud architecture. All of which play a hugely important part in our systems and many of those in use in the modern world outside of HX. I’m happy to have had the opportunity to learn so much about modern infrastructure that’s becoming used globally.

{{{ Maybe needs expanding }}}

# Projects

## Booking History

## Reviews

## Microservice Re-homing

Ever since the earliest design of microservices they’ve been imagined as belonging to a specific team, we call these owners. This ownership is significant for who maintains and takes responsibility for each service, whilst also correlating to significance in the architecture running them. This originally meant that from point of conception this owner was set in stone, due to the declarative infrastructure supporting them which wasn’t easy to move around.

However, with the businesses shift towards shorter-term mission-based pods which could come and go dynamically, there would be issues with the long-term ownership model. With this now becoming more necessary I took charge of investigating the possibility of re-homing services.

I spent time plotting out all different dependent areas which stored and relied on information about who the owner was. I narrowed it down to this list:

* Grafana Dashboard
* Github Teams
* Package.json
* Sumologic collectors
* Database instances

I then explored the APIs for each of these areas and managed to build a proof-of-concept script for almost all of them, showing that re-homing would be a definitely possibility. The only thing that seemed more difficult was re-allocating the databases, which I’ve detailed further in the challenges section.

Having shown the pod my initial investigation into this they were happy with my evidence that re-homing would be possible, although it was never going to be easy. I stepped forward and started working on it right away, turning my proof-of-concept into a fully-fledged process. I appreciated that the pod had enough trust to leave this in my hands even though I’d only been with the pod for a few months.

Eventually, I had the entire script built and working as an RPC endpoint in the access service. I spent time testing this before taking it to our demo and proudly showing the stakeholders what I’d achieved. It felt great to be able to tackle real problems for the pod and have a large impact.

I was also tasked with making a workplace post about the project. I had this edited by Steve and he had nothing but praise for the writing, and the post seemed well received by the web team. This was a nice confirmation of my written communication and helped prove that I had improved in these areas. It also boosted my recognition across the web team as now everyone comes to me for all things access-service and service re-homing. It’s nice to receive appreciation for work that I’ve achieved and knowing it’s having the desired impact.

### Challenges

Throughout the project the complications of database re-homing loomed over my head. In the initial design of microservices lead to there being a database instance per pod, which hosted a database per-service within it. The diagram below explains this structure a little further:

infrastructure-service

Customer Experiences’ Database Instance

Dev Foundations’ Database instance

access-service

contact-service

- Service’s database

- Database instance

Key

booking-history-service

This shows that all databases are grouped into an overall database instance that is created per-team. Meaning that to move a database’s ownership it has to be transferred from one instance to another, which is complicated and leaves practically unavoidable downtime during the move.

I spent some time drafting ideas on a way to write a completely down-time free moving mechanism, but it was extremely complex. I arranged one or two meetings with the pod to discuss this around a whiteboard. In these sessions we eventually agreed that it was too much investment at this point. Instead settling with leaving the databases for how they are. There’s no technical limitation that means the databases **have** to be moved, but with a multi-tenancy instance setup there can be negative effects. For example, if one of the services on the instance has an extremely busy database and the load is too much all co-tenant databases could be affected. This is why they were originally grouped by team, it would be the team’s own fault if their other databases were affected. We’d now be leaving it up to luck which databases are co-tenants on each instance. One day we might come back to this if it becomes an issue but it’s functional as is for now.

### Thoughts and Feelings

This project stands out as one of the more impactful things I got to do at HX. It completely changed the previous approach to microservice ownership, shifting the singular responsibility into dynamic responsibility across the web team, leaving pods open to pick up and own services fluidly.

Due to the fact this was one of my earlier pieces of work in dev foundations it also established my place as a vital contributor. Building this trust and respect within the pod has proved useful ever since, being able to confidently speak my mind and know the pod will fully consider what I say. It’s fair to say this could have been much harder without opportunities like this, especially being surrounded by engineers with vastly more experience than me in an extremely technical pod.

I also managed to demonstrate my problem solving and decision-making skills throughout the investigative stages of the project. Especially when I took the initiative to host meetings to discuss the database problem.

Overall, I feel this project was valuable to the business and even more worthwhile for my own development. I got to independently experience a complex issue all the way from investigation, design and final implementation. Carrying a project like this through from start to finish on my own so early in my career reassures my confidence and trust in myself.

## Dependency Graphing

* Infrastructure RPC dashboards
* Recent expedites, with so many services (100+) it’s hard to always know where issues **stem** from
* Evaluation of options, CLI implementation of graphviz, online website api, eventually gave up to d3 demands, looked at vis.js – still looks easier than d3 and just as good. Began to prep data sources, initial version, feedback form pod, added the function breakdown, presented, feedback on legend – added legend and posted to workplace
* Good feedback from stakeholders during demo – Si wood got excited and was talking about having it take a screenshot every week to see our inf grow (needs seedy thing)
  + Workplace post is a good piece to mention, lots of love
  + ALSO the presentation I hosted

# Other notable events

## Chauntry Acquisition

## Purple Parking Acquisition

## Maybe that expedite with Becky? Probably not – too much content now…

## GDPR Enforcement

## Conference Day Presentation

## Contract Extension

# Conclusion

NOTES

* What I did
  + Big projects
    - Access service – rehoming services
      * Source of truth from scratch
      * Scripting to turn that json blob into database queries
      * Database structure design (screenshot could be nice)
      * Workplace post – steve’s title love
    - First project – lighthouse service
    - Infrastructure RPC dashboards
      * Workplace post is a good piece to mention, lots of love
      * ALSO the presentation I hosted
    - Scripting shit – toolbox audit, PII audit, big renovate
      * Really enjoyed this nature of the job, felt very fluid and flexible – just get the job done in the fastest way possible. AUTOMATION

Loose bibliography

<https://www.holidayextras.co.uk/about-us/company-history.html> - HX info