



Developing Senior Skills: A Diary-Based Exploration by a Junior Software Engineer

Amaru Korpimäki

Haaga-Helia University of Applied Sciences

Bachelor of Business Administration in Information Technology

Thesis

2024

Abstract

| |
|---|
| Author(s) Amaru Korpimäki |
| Degree Bachelor of Business Administration in Information Technology |
| Title of the Report/Thesis Developing Senior Skills: A Diary-Based Exploration by a Junior Engineer |
| Number of Pages and Appendices 54 + 0 |
| <p>This diary-based thesis explores the professional development of a Junior Software Engineer at a Finnish procurement analytics company, during an eight-week period from September 18th, 2023, to December 1st, 2023. The thesis documents the daily tasks and experiences encountered by the junior developer, while also investigating the essential skills required for novice software engineers to transition into more senior roles.</p> <p>By examining the daily activities and challenges faced by the Junior Software Engineer, the thesis aims to identify the key competencies that differentiate junior and senior software engineers and to analyze the professional growth obtained after the reporting period.</p> <p>The research employed a diary-based methodology to document the Junior Software Engineer's experiences over the eight-week timeframe. Through daily entries, the thesis captures the practical aspects of the junior developer's role while also reflecting on the skills and knowledge required for success in more senior positions. The analysis of these entries allows for the identification of key skill gaps and areas for development that junior software engineers should focus on to progress in their careers.</p> <p>The diary entries revealed that effective communication and strong technical skills are crucial for junior software engineers aiming for more senior roles. Daily experiences highlighted the importance of clear communication in collaborating with senior developers and other stakeholders. Furthermore, the entries emphasized the need for a robust technical skillset to tackle complex problems and contribute to software projects. By focusing on refining these core competencies, junior software engineers can improve their possibilities to move to more senior positions.</p> |
| Keywords Software development, professional development, frontend development, communication, collaboration. |

Table of Contents

| | | |
|-------|---|----|
| 1 | Introduction | 1 |
| 1.1 | About the Company..... | 1 |
| 1.2 | My main responsibilities | 1 |
| 1.3 | Technical and not technical skills/tools at work..... | 2 |
| 1.4 | Objectives and limits of professional development | 3 |
| 1.5 | Reporting period..... | 4 |
| 1.6 | Key professional concepts..... | 4 |
| 2 | Framework | 8 |
| 2.1 | Analysis of your current work | 8 |
| 2.1.1 | Prior experience in software development | 8 |
| 2.1.2 | Insights and Analytics team | 8 |
| 2.2 | Interaction situations | 10 |
| 2.3 | Reoccurring themes for analysis | 11 |
| 3 | Diary entries | 13 |
| 3.1 | Observation week 01..... | 13 |
| 3.1.1 | Weekly analysis | 15 |
| 3.2 | Observation week 02..... | 17 |
| 3.2.1 | Weekly analysis | 20 |
| 3.3 | Observation week 03..... | 22 |
| 3.3.1 | Weekly analysis | 25 |
| 3.4 | Observation week 04..... | 27 |
| 3.4.1 | Weekly analysis | 30 |
| 3.5 | Observation week 05..... | 32 |
| 3.5.1 | Weekly analysis | 35 |
| 3.6 | Observation week 06..... | 37 |
| 3.6.1 | Weekly analysis | 40 |
| 3.7 | Observation week 07..... | 41 |
| 3.7.1 | Weekly analysis | 44 |
| 3.8 | Observation week 08..... | 46 |
| 3.8.1 | Weekly analysis | 49 |
| 4 | Discussion..... | 52 |
| | References | 55 |

1 Introduction

In this report we will analyze the work experience of a Junior Software Engineer working at Sievo Oy (also referred as Sievo). This work will be conducted as dairy thesis where we will share 8 weeks of daily reports exploring the tasks, context, and challenges that the reporter experiences working at a Finnish company specialized in procurement analytics. We will analyze the professional growth of the reporter and attempt to share insights on what are the main skills/factors that help novice Software Engineers to become more senior in their roles.

1.1 About the Company

Sievo Oy, founded in 2003 and headquartered in Helsinki, Finland, is a leading provider of procurement analytics and spend management solutions. With a core commitment to honesty, caring, and exceeding expectations, Sievo serves diverse organizations, helping them gain insights into spending, optimize procurement, and make data-driven decisions. Their comprehensive suite of tools streamlines spend management and fosters strategic growth.

Through strategic partnerships, Sievo continues to innovate, enhancing offerings to empower businesses worldwide in a dynamic marketplace. With over two decades of expertise, Sievo has been dedicated to revolutionizing the way organizations optimize their procurement processes and elevate their procurement strategies.

1.2 My main responsibilities

I joined Sievo Oy in May 2023 as a Junior Web Developer for a trainee position. Later the same year, on August 28, 2023, I was offered a full-time position as a Junior Software Engineer. The entries of this thesis are based on the work done during my time as a Junior Software Engineer.

At Sievo I perform as a Junior Software Engineer within the Insights and Analytics team (also referred to as Insights team). This Team is responsible for the Insights product which attempts to bring actionable pieces of procurement information (insights) to customers for them to be able to act. Some examples of this include the capabilities of negotiating better payment terms with suppliers, and choosing to change suppliers due to cost disadvantages, just to name a few. My responsibilities are mainly engineering tasks related to our product's front-end (the part that users interact with), this includes development and maintenance of the team's product. In simple words, this involves all the activities related to feature development, planning, testing, refactoring, coding, and reviewing own and other engineers' work.

Besides my technical responsibilities, I participate in all our team's meetings. These meetings include a variety of activities that foster team communication, the development of new features, team transparency, and the improvement of our team functioning. We will get back in detail about these activities in chapter 2.

1.3 Technical and not technical skills/tools at work

In this section, we'll describe the skills and tools required to perform in my role as a front-end Software Engineer. While the core responsibilities are technical, success also requires several soft skills. These are particularly important in our team environment where regular interaction with business stakeholders and designers is crucial.

To elaborate, within the Insights team, we can categorize roles based on their primary focus: back-end engineers primarily support front-end engineers, who in turn collaborate with business stakeholders (product managers and/or designers).

This approach is what we will call client-provider dynamic which explains that specific team areas serve the needs of other areas within the team. A diagram that describes this dynamic is available below.

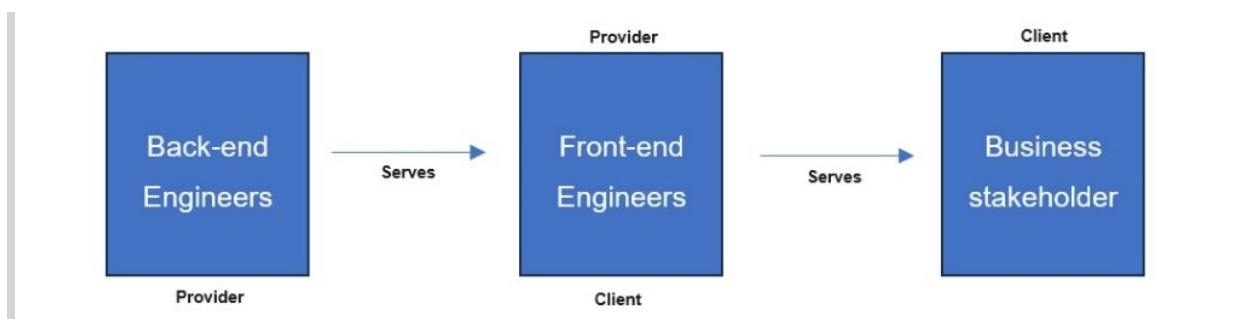


Figure 1. Team's client-provider dynamic

Our team's client-provider dynamic highlights the need for front-end specialized engineers to communicate frequently with the team's business stakeholders to gather business requirements and later translate those into technical specifications. Here are some of the soft skills that are required:

- Effective communication.
- Ability to filter technicalities when communicating.
- Ability to translate technical problems to business stakeholders.
- Ability to transform business requirement in technical specifications.
- Ability to explain to business stakeholders the limitations of our technical solutions.

As we mentioned above, the engineering work is primarily technical. In Table 1., we outlined some of the most important skills to perform in my role as front-end focused Software Engineer.

Table 1. Brief description of technical skills/tools

| Skill/Tool | Explanation |
|-----------------------------|---|
| Technical Skills | |
| JavaScript | A programming language used for adding interactivity to websites. |
| HTML | Hypertext Markup Language, the standard markup language for creating web pages. |
| CSS | Cascading Style Sheets, used for styling and layout of web pages. |
| Jest | A JavaScript testing framework for writing and running unit tests. |
| React | A JavaScript library for building user interfaces developed. |
| React-testing-library | A testing library for testing React components. |
| Styled-components | A library for styling components with CSS and JavaScript capabilities. |
| Git & Azure DevOps Services | Version control and project management tools for software development. |
| React-Query | A library for managing and caching API data in React applications. |
| React-Router | A library for handling routing in React applications. |
| TypeScript | A statically typed superset of JavaScript, enhancing code quality and development experience. |
| Jira | A project management tool for tracking and managing tasks, issues, and projects. |
| Figma | A collaborative design tool for creating user interfaces and prototypes. |
| Teams / Slack / Email | Communication tools for team collaboration and messaging. |

1.4 Objectives and limits of professional development

This thesis work has two main objectives. First, we will describe and analyze the professional development gained throughout the eight weeks documented in this diary thesis. To illustrate the weekly progress, we will frequently revisit three key themes of professional development: balanc-

ing efficiency in feature development with time management, communicating effectively with stakeholders, and maintaining code quality and collaboration. We will describe these three reoccurring themes in detail in chapter 2.

Our second objective diverges from the traditional diary thesis style. We aim to identify the skills and behaviors that launch novice software engineers into more senior roles. By understanding these, we can empower us and other novice engineers to progress in their careers.

1.5 Reporting period

The Diary entries are planned to describe the work performed from September 18, 2023, to December 01, 2023.

1.6 Key professional concepts

This section provides a list of the most relevant concepts that will be used in this thesis work.

Table 2. Key professional concepts

| Concept | Description |
|---------------|---|
| Frontend (FE) | The frontend of a website, termed the client side, involves user interactions like text, images, and navigation menus. It uses HTML, CSS, and JavaScript, prioritizing responsiveness, and performance. Ensuring seamless display on all devices is crucial for user satisfaction.(Hewlett Packard Enterprise s.a.) |
| Backend (BE) | The backend, known as the server side, manages data storage and client-side functionality invisibly to users. Backend developers handle tasks like API creation and non-user interface components, indirectly accessed through the front end by users.(GeeksforGeeks 2023) |
| NPM | Node.js package manager, originated in 2009 to facilitate code module sharing among JavaScript developers. The npm |

| | |
|--|--|
| | <p>Registry is a vast repository containing open-source packages for various JavaScript applications, including Node.js, web and mobile apps, robotics, routers, and more, supporting the JavaScript community's diverse requirements.(NPM s.a.)</p> |
| Azure DevOps | <p>Azure DevOps fosters a cooperative environment and a series of procedures that unite developers, project managers, and contributors for software development. It enables organizations to enhance product development speed, outpacing traditional software development methods.(Microsoft 2022)</p> |
| User Interface | <p>User Interface (UI) encompasses elements that enable user-machine interactions, spanning keyboards, joysticks, screens, or software components. In computer software, UI examples include command-line prompts, webpages, input forms, or any application's front-end.(MDN web docs s.a.a)</p> |
| User Experience | <p>UX, short for User experience, involves examining how users interact with a system to enhance user-friendliness. Its primary objective is to ensure that a system is user-friendly and intuitive.(MDN web docs s.a.b)</p> |
| TanStack Query (We will use its former name as it fits better our context (React Query)) | <p>TanStack Query, formerly known as React Query, is frequently hailed as the absent data-fetching tool for web apps. In technical words, it simplifies the processes of fetching, caching, synchronizing, and updating server state, streamlining these tasks in your web applications.(TanStack Query Docs s.a.)</p> |

| | |
|--------------------------------------|---|
| API | In web development, an API typically consists of code elements like methods, properties, events, and URLs that developers employ within their applications to engage with a user's web browser, the user's device software/hardware, or external websites and services.(MDN web docs s.a.b) |
| Privileged Identity Management (PIM) | PIM is a service that allows organizations to manage important resources in the cloud in the case of Microsoft Azure, PIM is part of the Microsoft Entra ID service.(Microsoft 2023) |
| Azure Application Insights | Azure Application Insights is a service that allows organizations to understand performance and usage of websites. (Microsoft Learn s.a.) |
| Postman | Postman is an API platform that eases the API management and lifecycle. It enables user/organizations to create better APIs in less time.(Postman API Platform s.a.) |
| Microsoft Azure | Also referred as Azure is Microsoft's public cloud (Microsoft Learn s.a.) |
| Azure portal | Web-based solution that unifies Azure resources and allow organizations and user to create and manage them(Microsoft Learn s.a.) |
| Slack | Messaging application for business that aims to bring employees together and streamline collaboration(Slack s.a.) |

| | |
|---|---|
| Version control | Also referred as source control, is all the practices that relate to tracking and managing code changes. (Atlassian s.a.) |
| Continuous integration and continuous deployment (CI/CD) pipeline | Also referred to as a pipeline, it contemplates the series of steps that has to be performed to deliver a software.(Red Hat 2022) |

2 Framework

This chapter sets the starting point of the reporter's skill set and provides further context on the Insights team. This information aims to help the reader further understand the daily entries provided in the next chapter and establish a reference point to compare after the reporting period.

2.1 Analysis of your current work

2.1.1 Prior experience in software development

I developed my knowledge in software engineering through work and studies before starting my role at Sievo. I started my studies at Haaga-Helia University of Applied Sciences in 2021 where I have participated in many theoretical and practical courses that have helped me acquire software development skills.

Before joining Sievo, in 2022, I worked as a Software Developer Trainee at the Finnish postal company Posti Group Oyj (later referred as Posti), where I was part of their Design System team. The work experience at Posti enhanced my skills in JavaScript, React, styled-components, TypeScript, and source control with Git and GitHub.

Prior to starting my current role at Sievo, I worked as a Junior Web Developer. During this traineeship period that lasted from May 2023 till August 2023, I worked in the Insights team improving my skills in Front-end development. I learned React Query, React Router version 6, Azure DevOps platform for source control, and I expanded my knowledge in testing React code with Jest, TypeScript, React, styled-components.

After the Traineeship period I was offered a full-time position as a Junior Software Engineer in the same team (Insights team).

2.1.2 Insights and Analytics team

The Insights team is comprised of Engineers, Designers, and Project Managers that cover all the needs of the Insights product development. The Engineers of the team are subdivided into categories of specialization resulting in Back-end Software Engineers, Front-end Software Engineers, and

Analytics Engineers. In a similar way, Product Managers are subdivided into technical (Engineering Manager) and non-technical (Project Manager).

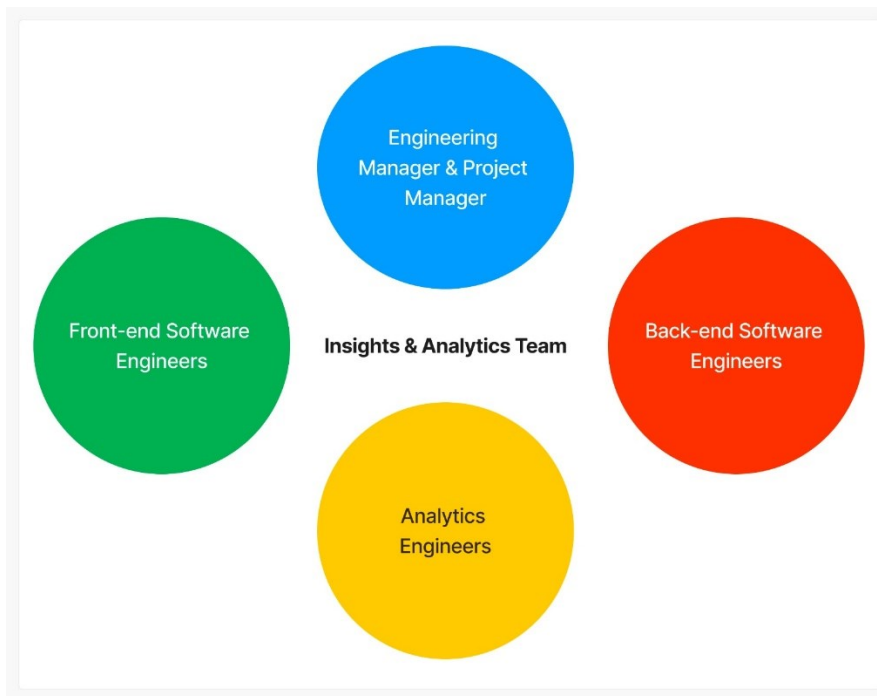


Figure 2. Insights team diagram

My area of responsibility in the Insights team is Front-end development. As Front-end specialized Software Engineer our main concerns relate to how users interact with the application, also known as user interface and user experience.

Performing as a Junior Software Engineer focused on Front-end requires specialized skills. Within the team's context we use the following technologies for our front-end development. Our main building blocks are Reactjs and TypeScript, these two technologies allow our team to build components that are strongly typed and reusable. We also utilize two main JavaScript libraries to manage data flows (React Query) and to manage routing (React Router version 6). To style our components and give the user a pleasant user experience we use the JavaScript library styled-components, which allow us to write JavaScript like code within CSS specifications. Lastly, we use a JS library called Jest for testing purposes.

In the same sense, we need to use other tools/technologies to perform front-end related tasks. Our team uses QlikSense to manage visualizations and data aggregation, and Azure DevOps as version control platform to handle all repositories and pipelines.

The overall description of my previous experience and required technical skills to perform as a front-end focused Junior Software Engineer set a comparison point for future analysis. We will attempt to share insights on areas of development later in the discussion section.

2.2 Interaction situations

As described before, the Insights team categorize our engineers and project managers into specific areas of expertise that represent the responsibilities that each have over our product development. In this sense, performing as a Front-end focused Junior Software Engineer the main interactions with stakeholders within the teams are as follows:

- **Back-end Software Engineers:** Mainly to agree on API contracts, discussed documentation, and further develop our product connecting front-end with back-end via APIs.
- **Analytics Engineers:** Main interactions will support on chart/visualization creation using QlikSense or bug findings in our application that might concern their area.
- **Project Manager:** Mainly for acceptance of features, feature planning, and reporting of feature development in our demo sessions.
- **Designer:** The interactions aim to clarify new features' designs or request changes based on feasibility of development.
- **Engineering Manager:** We interact with the Engineering Manger mostly have 2 in feedback sessions, technical discussions, and daily meetings.
- **Insights Front-end Engineers:** Within the Insights team there is a Senior Software Engineer that shares the same responsibilities and acts as my mentor. In that sense, I will interact with him in planning, reviewing, testing, and developing our frontend application further. That includes mentoring, requests for guidance on feature development and other support interactions.

It is a common practice in software development teams to have reoccurring activities that aim to align the development of features, ensure the product quality, improve processes, among other benefits. Our team's activities are the following:

- **Refinement sessions:** In our refinement sessions, which occur every Friday, we generally discuss the feasibility of implementation of new features brought up from our business stakeholders (Project manager (PM), Team's Designer, or external stakeholder). We also might use this session to share ideas on how to implement features between Engineers.
- **Demo sessions:** In our demo sessions, which occur once every second week, we demonstrate the features that we have developed during the two weeks prior to the meeting. The

objective of this meeting is to sync our business stakeholders with the progress of the products development.

- **Retrospective sessions:** In our retrospective sessions, which occur on the last Thursday of the month, we analyze our team's processes and attempt to improve them. Generally, this meeting would result in small actions points to start improving the team's functionality as soon as possible.
- **Daily sessions:** In our daily session all team members would report to the team their task progress, communicating what was done during the previous workday and what is the plan for the current workday.
- **One on one sessions:** This session occurs once a month and it works as a feedback session between my Engineer Manager and me. Other team members have their own separate one on one session.

Moreover, my responsibilities might also lead me to participate in other company-wide sessions either representing my team's or as an engineer in general to improve or be informed of our company's progress. Some examples of this meeting are:

- **Frontend UX guild meetings:** These meetings occur on a weekly basis and their main objective is to gather engineers with front-end responsibilities and designers to comment on company-wide problems in the areas of front-end development and user experience.
- **Product cast:** This meeting occurs every 3rd Wednesday of the month and its main objective is to inform all employees of the development of the company.

2.3 Reoccurring themes for analysis

As previously mentioned during chapter 1, one of our main objectives is to analyze our professional development. For this purpose, we have chosen three main axis that will be use frequently to evaluate our weekly work. Utilizing these three main themes we aim to ease the evaluation of our own progress. These themes have been selected because they touch on both, areas of our responsibilities and the professional growth of a novice Software Engineer.

The first theme of this thesis is **Efficiency in Feature Development vs. Time Management**. This topic aim to target the velocity in which I will be developing features in the upcoming weeks. According to Kiisinen (2022), velocity of development is a good measure for teams' performance but can became a bad indicator when manager or product owners use to compare between engineers or teams. Because of that we will be using velocity only applied to our own work without comparing how others are performing and aiming only for self-improvement and career development.

The second theme is **Effective Communication**. Using this theme as an axis to explore our professional growth, we aim to delve into one of the crucial areas of software development. To illustrate this idea, Forsgren et al. (2021) mentioned that software development is a collaborative endeavor that relies on effective and extensive communication within the team and with other teams and roles. As described in the previous chapter, our team exhibits the client-provider dynamic, and engineers holding front-end responsibilities are subject to continuous interactions with business stakeholders as their clients, but the interactions are not limited only to business stakeholders. In this sense effective communication plays an important role in our work.

Maintaining Code Quality and Collaboration is the last theme for evaluating our progress. As we previously described, some of our responsibilities include reviewing, testing, and ensuring the quality of our own code and that of other engineers. Using this theme, we aim to track our development in this particular area.

3 Diary entries

3.1 Observation week 01

Monday, September 18, 2023

Today I worked in the development of the “Discard insight feedback modal”. This was the first iteration on the integration with backend. I have updated the hooks and implemented interfaces (for type safety) as well as enums. I already saw potential areas of improvement on my code, for example I’m using multiple useState when I could combine them into one more reusable that can handle multiple properties in one object. Due to this I consider that this will be version one of my code and I will come back later to do code improvements.

During my day I also worked on our secondary repository. This repository is not under active development and only needs maintenance from time to time but as our team owns the responsibilities over it. My task was to update the Qlik host used in this product but when I proceeded to do it, I noticed that our CI/CD pipeline was broken. The issue was caused by inconsistency in the npm registry of the company as some of the library versions were removed from azure DevOps Artifacts manually. I solve this issue by removing yarn.lock and updating the dependencies to posteriorly update the node version of the build pipeline from version 16 to version 18.

The task in our secondary repository was originally to update the Qlik host that the application used but due to internal rollback of the product from another team we did not need to proceed with the update. Anyways I investigated the repository, and I noticed the issue, so I updated the Jira ticket to target the new breaking issue.

Tuesday, September 19, 2023

I continued developing the backend integration for the feature I'm working on today. The integration is nearly completed. The application already fetches data from the database to the backend (BE), which is then received by the frontend (FE).

During development, I noticed some issues with the BE functionality related to the (previously defined) discard reason feature. After researching the endpoints using Postman, I found that the BE would not allow changes in the database to the discard reason unless the status was included. However, it also would not update discard reasons if the request included the same status.

To solve this, I contacted a backend engineer of my team who developed this part. I explained the problem and we agreed they would fix it later in the day. I then communicated these findings to the team in our main Slack channel for transparency and explained my next steps.

For the rest of the day, I focused on refining code I wrote previously. I refactored the main Feedback modal component from using five separate states to set and update data for different fields into a single useState object with just one handler, reducing the code size by six times. I also created a separate component for a confirmation dialog that was previously coded within the Feedback Modal component, improving code readability and reusability.

Wednesday, September 20, 2023

I continued refactoring my code for the discard reason feedback modal. I spent a good part of the morning improving the code's readability and integration with the BE API. Around the middle of the day, the BE was ready for testing the integration.

After testing, I realized my solution required restructuring because the information for the request changed dynamically based on the route. Therefore, I discussed this issue with the other frontend (FE) software engineer on my team. He suggested a different approach where I could cache the information before making the request.

I spent the rest of the day implementing this suggestion. I modified the hooks I previously created into a new version that uses react query to cache changes in the insight data before submitting the feedback to the backend.

Thursday, September 21, 2023

I used the first half of the day analyzing and implementing the new approach that was suggested from a more senior colleague. The approach for the discard reason functionality in integration with BE, was the following: update the queries that React-Query provides in our application first and then when the user has interacted, and we are sure that the interaction finished then send the request to the BE.

With this approach we achieved certain things that are critical for a system that it is computationally heavy, we achieved a fast and flawless UI/UX allowing the users to see immediate changes based on their interactions, and we reduced the computational load of sending many requests to the BE.

In the second half of my day, I received a message from the same engineer saying that in the approach suggested above we did not consider a proper error handling so that said approach need to change and that the complexity of the task increased significantly. He recommended me to communicate the complexity rise to the Engineering Manager and Project Manager of the team.

Friday, September 22, 2023

Friday was a busy day filled with meetings. We started with our weekly refinement session (1.5 hours) where the team discusses issues and features requiring attention for product development. This was followed by a 1-hour introductory session with our new Project Manager (PM), where we, the front-end engineers, showcased the front-end aspects of our product. We also had a slightly longer daily meeting (around 30 minutes).

To welcome our new PM, we had a team lunch that lasted about an hour.

After a day packed with meetings and team activities, I finally had a few hours to focus on the discard reason feedback feature. A senior colleague suggested we tackle this feature and the new approach together, through pair coding. This was an incredibly valuable experience, a total game-changer in my opinion. I gained valuable insights into the thought process of a Senior Software Engineer, which will be useful in tackling future challenging features.

3.1.1 Weekly analysis

In this section we will analyze the first week of diary entries basing our analysis in the 3 main themes for this thesis work (better described in chapter 2) and having in mind the overall objectives of this thesis, meaning, professional growth and attempt to discover what does it take for a novice Software Engineer to move into more senior positions.

First, we can identify and classify the events that happened over the last week. To have a clear understanding, we will structure last week's events as a table to separate them into sections based on each theme for this thesis and later further analyze them in the scope of our thesis objective.

Table 3. First week theme review

| Themes | Events/Actions in the previous week |
|---|--|
| Efficiency in Feature Development vs. Time Management | <ul style="list-style-type: none"> - Development of features equivalent to one story point. - Feedback modal two-thirds completed (2 story points task). |
| Effective Communication | <ul style="list-style-type: none"> - Refinement session. - Communication with Backend Engineer to fix bugs. - Active communication with more Senior Software Engineer (mentorship). |

| | |
|--|--|
| Maintaining Code Quality and Collaboration | <ul style="list-style-type: none"> - Reported issue in Backend related feature. - Reported and fixed issues related to our secondary repository. - Pair coding with more Senior Engineer. |
|--|--|

Let's analyze my feature development velocity last week. I completed one 1-point story and 2/3 of a 2-point story, demonstrating strong progress. Our team uses the Fibonacci sequence to estimate story points (as illustrated in Table 4). This is a common practice in agile teams (ProductPlan).

Table 4. Story points assessment

| Story points | Time needed |
|--------------|--|
| 0.5 | Few hours |
| 1 | 1-3 days |
| 2 | A week |
| 3 | 1-2 weeks |
| 5 | 3+ weeks |
| 8 | A month |
| 13 | Too big (must be divided into smaller pieces). |

Based on the diary entries and Table 4, our development velocity appears to be exceeding expectations. While a single 1-point story can take up to 3 days, I completed nearly 3 story points worth of work, which typically takes 1-2 weeks.

According to Li, Ko and Zhu (2015), strong communication skills are essential for engineers, as they hold the most in-depth knowledge of the product. This is because engineers need to collaborate effectively with various stakeholders. My experience aligns with this concept. Through practice and by sharing my findings with other stakeholders, I've further developed my communication skills. This collaboration has proven valuable in refining the product and preventing future bugs. Additionally, taking ownership of my work allows the team to gather valuable customer feedback, which ultimately leads to product improvement.

According to Li, Ko, and Zhu (2015), the ability to learn is the most critical skill for software engineers due to the ever-evolving nature of the field. This concept resonated deeply with me during my first-ever pair programming session last week with a senior front-end engineer on my team. The experience provided invaluable insights into my colleague's thought process and approach to problem-solving, which I believe will significantly enhance my own skills.

Finally, I see room for improvement in considering the bigger picture during development. While I refactored my feature, I could have analyzed it in the context of the entire application. This broader perspective would have allowed me to design the feature with more flexibility and reusability, ultimately benefiting other developers.

3.2 Observation week 02

Monday, September 25, 2023

Objective for Monday was to implement the functionality of the feedback modal for the discard reason collection using useContext hook from Reactjs.

My day started having daily meeting where I notified my team on the progress made in the feature during last Friday and what I was going to continue doing today.

I started trying to implement the context for the discard reason, the reason behind this is to avoid prop drilling. After trying for most of the first half of the day and try with two different methods without much success I ended up asking for help to my colleague. He gave me a couple of hints and ideas on how to improve my code.

Thanks to those hints I was able to implement the context for the feature and I also got inspired to use react hook form (JS library to manage and validate forms) to handle the feedback modal. I spent couple of hours trying to implement the form using react hook form library without success so I will need to continue tomorrow developing that. I will need to communicate with my manager to understand the expectations on time as I have been working on this feature for quite a long time.

Tuesday, September 26, 2023

The main goal for Tuesday was to get the Feedback modal for collecting discard reasons working using React Hook Form. I started the day by exploring the library documentation for an hour before our daily meeting to understand its integration with Sievo's component library. The biggest challenge I encountered was utilizing the library with Sievo's components, as they don't expose element references. This prevents the library from accessing HTML element values.

During the daily stand-up, I explained the feature's status and the reason behind the extended development time. I emphasized focusing on code quality and flexibility to accommodate future features built upon this one.

After the daily meeting, I continued working on the feature. I successfully implemented it, and the feedback modal now returns a correctly formatted object, ready to be sent to the database through our API. I also had a brief pair-programming session to resolve an error that emerged during development.

Finally, I created a pull request (PR) for the feature and notified the team that it's now ready for a technical review. Tomorrow, I'll address any issues raised in the PR review and begin analyzing the next feature.

Wednesday, September 27, 2023

My main goal for the day was to get the PR I created the day before approved. I arrived at the office about an hour before the daily meeting and started fixing the merge conflicts in my pull request. There were only three conflicting files, and I easily resolved them. Before committing my changes, I ran our application to ensure all functionality remained intact. Unfortunately, I found an issue that crashed the application.

At the daily meeting, I reported my progress from the previous day and outlined my plans for the current day. I also informed the team about the error I encountered that significantly impacted the functionality. I expressed my intent to debug and hopefully resolve it that day to proceed with the technical review.

The error presented an interesting challenge. Even after commenting out the added code and clearing the cache (to eliminate cached errors), the issue persisted. In an effort to understand the problem, I explored unfamiliar parts of the codebase, but unfortunately, remained unsuccessful.

After spending a significant portion of the morning attempting to fix the error independently, I reached out to the Senior Frontend Engineer on my team for assistance. We called and spent over an hour exploring the error together. He was surprised by the issue and could not pinpoint the cause.

My colleague and I decided to split up the troubleshooting effort, working on separate leads and communicating findings via Slack messages. I will continue working on resolving this issue tomorrow.

Thursday, September 28, 2023

My goal for the day was to find the source of the bug that is making the application break. I might not achieve that goal because the day will be full of meetings covering little less than 3 hours of the day between daily meeting, team demo session, company coffee break, and team retrospective session. That means a lot of contexts switching and will probably result in less technical performance.

The first things of the day were daily meeting plus team's demo session. I communicated during the daily that I might not be able to fix the bug before the demo session that meaning I wouldn't be able to show the feature itself, and that I will continue debugging today as yesterday's debugging session with the Senior Software Engineer of my team did not produce results.

Right before the demo I found a certain user flow that worked, and I noticed that even removing all the merge changes the bug was still present. That discovery led me to the conclusion that the error must be in the code I produced and not in the merging.

I demoed the user flow that was bug free in the demo session and I got positive feedback about it beside some comment related to possibility of SQL injections in the text input of my form which were quickly discarded by more Senior Engineers.

In the afternoon I have the retrospective meeting where we discuss how to improve the processes in our team. We clean the board of action points that were put in place by our old PM, and we move forward with the new PM's suggestions. The biggest takeaway from the retrospective meeting was that we need to improve the feedback processes in our team. We generated two actions for everybody in the team, first give two entries of feedback about personal work balance and give and request feedback from three colleagues.

I spent the rest of the day debugging my feature with a new approach. This new approach was, create a new branch and add my changes step by step to see where the app fails. I will continue debugging tomorrow afternoon as I will have a morning exam tomorrow.

Friday, September 29, 2023

Friday's main goal was to find and solve the bug in my feature. I needed to skip daily and refinement session due to exams at Haaga-Helia. Nevertheless, I reported my progress debugging and what I was going to do during Friday by text in our team channel and watched the recording of our refinement session later during the day.

I arrived at work at noon to attend a meeting where we discussed a new feature that our PM is pushing to the current milestone. I participated actively and took notes of all the important details of the features, insights provided by more experienced employees to learn from them and speed up my growth as an employee in general.

I spent the rest of the day debugging my feature, I succeeded in finding the error and fixing the bug using the approach I mentioned in Thursday's diary entry. I felt good because I have been dealing with impostor syndrome because of the amount of time that this feature has taken me to develop.

I finished my day solving merge conflicts, cleaning the code, adding the updates to the UI suggested by PM and Designer. So far, I think I won the week. I dealt with this big bug mostly by myself and I manage to well balance work and studies.

3.2.1 Weekly analysis

In this section we will analyze the diary entries from the past week. We will review the events that occurred, and we will review them under the scope of the themes for this thesis having in mind the main objective of this thesis work.

Table 5. Second week theme review

| Themes | Events/Actions in the previous week |
|---|---|
| Efficiency in Feature Development vs. Time Management | - No improvements in this regard due to development issues and bugs. |
| Effective Communication | - Participated in new feature discussions. - Report of bug status |
| Maintaining Code Quality and Collaboration | - Collaborated with Senior Engineer from my team to solve bug in the code base. |

In table 5 we can find the events of last week separated by the three main themes of professional growth for this thesis. As we can see, last week's communication with stakeholders had a central part.

First, the basic nature of a developers / engineer's work. The role of professional developers is a communication role because there is constant interaction between stakeholders to understand the requirements and needs that need to be integrated into the result of the engineer's work (Martin 2011).

Li et al. (2015) pointed out the need in software teams to share context between stakeholders, mentioning that great software engineers have the skills to share said context as part of their effective communication with other stakeholders as developing a software involved many participants everyone should be able to share their knowledge to achieve success. Based on that we can argue that there might be a difference in the level of communication between novice Software Engineers and more Senior Software Engineers.

I believe that analyzing the interactions that more senior engineers have in meetings and understanding how they provide context to other stakeholders could increase the chances to move into a more senior positions.

While last week's story point output and feature development were lower due to encountering several bugs, I gained valuable experience in debugging. I successfully resolved the most critical issues, and I identified areas where I could benefit from guidance on tackling problems independently.

As Martin (2011) points out, debugging is often seen as separate from coding, but for companies, debugging time is equally valuable. This shows a flaw in our evaluation based on velocity of development but at the same time shows the importance of debugging as development activity. Moreover, debugging is a core skill required by all software engineers (Goodliffe 2014a, chapter 10). In this sense, we can see the value of last week's experiences for our professional development in getting more experienced in the debugging process.

As I mentioned before communication is a skill that senior software engineers use more often regarding sharing context with other stakeholders. Communication can also be used by experienced developers when mentoring junior developers. As a developer gain more experience, they take roles of leadership over less experience developers so the ability to mentor the less experience engineers becomes a key skill to share feedback, share knowledge and growing a positive and productive environment at work. (Mongare s.a.)

Last week's experiences relate to this as I received mentorship from a Senior Software Engineer. Through pair programming I got to see how a Senior Software Engineer approaches problems and splits tasks into smaller pieces to reduce complexity. This allowed me to analyze my own approach to solving problems and to improve my skills as a Software Engineer.

As a summary, skills like sharing context through effective communication with stakeholders, debugging, knowledge transfer, and mentorship are key parts of what constitute seniority. Even if these skills and behaviors were not performed by me, gathering information, and analyzing the behaviors of Senior Software Engineers can help us growth as we gain more experience in the field.

3.3 Observation week 03

Monday, October 02, 2023

The main goal for Monday was to clean up my code related to the discard reason modal feedback and get the feature ready for being reviewed.

I started the day reviewing the changes I implemented last Friday and checking that the application ran properly locally. I proceed to remove many lines of code that were leftovers from previous iterations in the most important parts of my feature, for example, the custom hooks that update the queries and ensure data integrity between backend and frontend.

I continued my day having daily meeting in which I explained the status of my task and communicated that the bug from last week was finally solved.

After the daily meeting I proceed with the cleanup of my code until I run into another issue while testing my feature. The error was related to the fix that I implemented on Friday because I did not consider back then that the information flows not only between statuses but also between filtering states e.g. Data can move from a query with key ['insights', 'future'] → ['insights', 'past'] which is what I consider in the previous fix but I forgot that there's also the possibility for data to move ['insights', 'future', 'marked'] → ['insights', 'past', 'marked'].

Because I did not consider the filter part of the query key some of the interactions with the application were not working as expected.

To summarize, I managed to solve this issue by invalidating the target query which means all the queries that are like ['insights', 'past'] will be re-fetched and in that sense data integrity would be achieved.

I ended the day having my code ready for review and with the mentioned bug fixed.

Tuesday, October 03, 2023

The main goal for Tuesday was to fix the issues that came up from the pull request (PR) review and participate in a new feature discussion session with backend (BE) Engineers.

There were about 20 comments on my pull request, they were mainly related to removing leftovers (unused code), naming conventions (for code readability), and performance heavy usage of function from react hook form library.

I used the following strategy to solve the comments. I first read all of them and try to understand them, then I picked the easiest first, for example, removing leftover code (comments, unused ESLint exceptions, etc.) after the easy comments were solved I moved to the comments that did not need any clarifications but were more challenging, for example, remove low performance functions from react hook form library and replace them with better performance-wise functions from the same library. Lastly, I proceeded to add my explanation, questions and/or clarifications to the rest of the comments (not more than 5).

In addition to solving comments, the other main event of the day was a discussion about our next feature with Backend Engineers. This discussion resulted in a strategy for handling the coming feature, which will require strong collaboration between backend and frontend engineers.

I ended my day without answers in some of my threads (conversation with multiple comments) so I will need to get to them again tomorrow.

Wednesday, October 04, 2023

The main goal for Wednesday was to resolve the rest of the comments on my pull request (PR) and participate in the Frontend UX guild meeting.

I started the day getting some answers to my questions and clarifications on the comments that were not resolved on the previous day. Most of the comments required further investigation to implement the suggestions on my code. I spent many hours finding solutions for the issues described in the comments and trying different approaches into my code (using what I found in practice).

I also participated in the “Frontend & UX guild” meeting. This meeting is an instance where engineers and designers collaborate to find solutions or further improve the company processes or code base in different areas, for example, internal tools, companies’ design system, etc. Wednesday’s topic was Typography and how it is used across the company. My main interventions were suggesting that we use in the application 3 different categories of text headline (for titles), labels (for input labels, category labels, etc.), and body (for any of the text under a headline or a label) and showcase where we use certain typography elements in the Insights application.

At the end of the day, I resolved most of the comments from my PR and I got my feature in our dev environment for our Designer and Project Manager to start testing.

Thursday, October 05, 2023

Thursday goals were, participate in team meeting about pilot release of our product and make improvements in the discard modal feedback feature based on the comments from business side (PM & Designer)

In the first part of the day, I had the preparation for the pilot release of our product. The main objective of this meeting was to sync the team in why, what, when and how we will conduct the pilot of our product so that the whole team is aware of the expectations in terms of time, order of priorities, and what to expect before, during and after the pilot release.

During our refinement meeting, I was asked by our product manager to take summary notes. The most challenging part of the meeting was almost at the end when our Project Manager (PM) asked us if we could be ready with all the new changes before November 8th or November 30th, those being the two tentative dates for the release. The answer from all Engineers was the same, "it is hard to estimate without further thinking." So, we set next Thursday as the day for decision making in this regard, giving us engineers one week to contemplate the feasibility of the plan.

The second part of my day was mainly focused on fixing comments and implementing new parts of my feature that were suggested after what we call in our team "business testing" which means testing done by Project Manager (PM) and Designer.

In the next day I will need to ask for code review for these new parts and repeat the business testing to see if my feature complies.

Friday, October 06, 2023

On Friday, my primary focus was refining my code after the last testing iteration and leaving it ready for approval and participating in one-on-one meeting with our team's Engineering Manager (EM).

I spent the first part of my day at Haaga-Helia due to graded activities so I could not participate in the daily and refinement session. Nevertheless, I reported my progress on our team channel.

I arrived from the university to the office to attend the "one on one" meeting. The meeting went well, the team's EM said to me that he was happy with my performance and collaboration with the team. He also shared my peer feedback scores which were 9/10 in performance, 9.5/10 in collaboration, and 10/10 in values.

After the meeting, I fixed the latest issue where one of my coworker's latest changes caused the undo discard functionality to break. I solved that issue and I left for Monday validating and assessing the solution.

3.3.1 Weekly analysis

This section aims to analyze the Diary entries from last week considering the main themes of professional development for this thesis work and our main objectives.

In the following table you can find the three themes of the thesis and the events that relate to them.

Table 6. Third week theme review

| Themes | Events/Actions in the previous week |
|---|---|
| Efficiency in Feature Development vs. Time Management | <ul style="list-style-type: none"> - No improvements in this regard due to development issues and bugs. |
| Effective Communication | <ul style="list-style-type: none"> - 1:1 meeting with Engineering Manager. - Participated in many meeting with stakeholders on new features and team goal planning. |
| Maintaining Code Quality and Collaboration | <ul style="list-style-type: none"> - Debugging and implementing changes on feature. |

For this week's analysis we will focus on **collaboration and communication** as the main topic to analyze, how do Senior and great Engineers stand out from their peers in terms of communication and collaboration. How can we relate to this based on last week's diary entries?

Software engineers, contrary to popular belief, need strong communication skills. In fact, experienced engineers often spend more time communicating than coding as their careers progress (Flood 2015).

As Flood (2015) highlights, software engineers, contrary to popular belief, need strong communication skills. This aligns perfectly with my experiences from last week. I spent a significant amount of time in meetings, answering technical questions from my managers. It was not just communication, but also translating complex technical details into simpler language for everyone involved.

Moreover, I spent a part of my time last week clarifying the decision I took in my code to more Senior Engineers in my team, arguing why the decisions I made were the correct ones in our context.

As GoodLiffe 2014b refers to in chapter 36 “Speak up!” of his book “**Becoming a Better Programmer**” communication is at the very center of team camaraderie and it is a key feature of well-functioning teams. Nevertheless, it is important to consider when to have meetings or communication interactions as they can break the working flow or state of high concentration when working on something.

Another of the communication and collaboration events of the previous week was the one-on-one meeting I had with the team’s Engineering Manager (Technical Manager). The goal of this meeting is to evaluate my progress and performance as a part of the team.

During this meeting I receive feedback related to my performance, values, and collaboration.

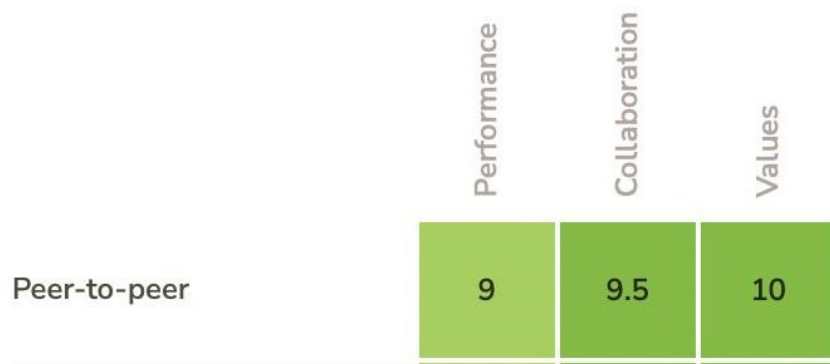


Figure 3. Peer feedback results

Overall, the feedback from my peers not only remarks the importance of collaboration within the team but also highlights my own skills that could start driving me towards more senior positions.

Lastly, we have widely cover collaboration and communication in relation to more business-oriented processes so let us take a moment to describe how communication plays an important role in process related to code reviewing and further improve code from peers’ feedback.

As concluded by Sadowski, Söderberg, Church, Sipko & Bacchelli (2018) in their study that aim to investigate the code review process at Google, peer code review provides multiple benefits being one of the important ways in which engineers teach each other, share best practices, and overall learn from each other.

Regardless of the area in which communication and collaboration is used, it is at the epicenter of my team’s work. Further improving communication, feedback processes, and code reviews can potentially benefit our company in the sense that Engineers from multiple teams could share knowledge not only related to the code base but also related to teams’ processes, and in that way improve from the other engineers’ experiences and whole organizations.

3.4 Observation week 04

Monday, October 09, 2023

The main objective for today was to get my feature fixed and ready for deployment.

I started the day with the daily meeting in which I explained to my team that there was only one more known issue to solve in my feature and that I would be working on that today.

After the daily, I continued working on my feature. The main issue to solve was that the modal did not appear as the most front element in the page having some elements accessible when the modal was open. Having the modal as the most front component is important for users to know that the main interaction is with the discard reason feedback modal.

Before I started working on this fix, I had the impression that it must be related to the place where the anchor for the modal is located (the anchor refers to a placeholder that points where the modal will be attached in the HTML document tree).

To solve this issue, I had to do a couple of things. First, incorporate another layer of the application into my local environment, I am referring to the "Frontend Bundle." This service patches all Sievo's micro-frontends into a single product. Second, find a way to anchor the modal into a higher parent or same level in the hierarchy of the HTML document tree so the modal renders with the proper layout.

After about an hour of researching my solution, I came to the conclusion that for this fix to work I would need to change the anchor of the modal programmatically, meaning, if frontend bundle was present (like in our remote environment) I will use the id from a higher element otherwise (like in local environment) I will use the anchor that I already had.

The later approach worked, and I finished my day pushing the results to my branch for being reviewed by other Engineers.

Tuesday, October 10, 2023

On Tuesday, my primary goal was to deal with comments on my pull request and test the feedback collection feature.

The day started by going through the comments on my pull request. I focused on fixing typos in the code. It might sound small, but it is important for maintaining good code quality. I also paid attention to suggestions for making the code more reusable. I refactored the code to turn a frequently

used function into a utility function. This cleaned up the code and made it easier for others to collaborate with it in the future.

I also spent some time cleaning up any leftover code from previous versions. These leftovers can clutter the project and create confusion, so removing them was necessary to keep everything organized.

In addition to improving the code, I refined the types for certain PUT requests. This ensures that the code remains consistent to prevent potential issues in the future.

I stayed in touch with my team throughout the day. I had discussions with the project manager and designer, making sure the feature aligned with our project's goals and design requirements. We also conducted testing in our development environment.

Typically, a QA/Test engineer handles testing, but we do not have one, so engineers took on the testing responsibility. I dedicated a significant part of my day to testing the feature in our development environment. The goal was to find and fix any bugs early in the process, preventing them from affecting the final product. This proactive approach saves time and resources.

By the end of the day, I had made significant progress on my pull request, collaborated effectively with my team, and carried out thorough testing. It was a day of demanding work, focusing on the details, all to ensure we deliver high-quality software to our users.

Wednesday, October 11, 2023

Wednesday's main objectives were to resolve bugs discovered during the testing conducted yesterday and prepare the feature for another round of review and testing.

The day began with addressing an issue pointed out by our Project Manager. The problem lay in the discard feature, which required a specific functionality adjustment. After discarding an insight from the list, the subsequent insight needed to be selected as the previously discarded one was removed. This modification aimed at enhancing the user experience and making the feature more intuitive.

In addition to this, I worked on fixing merge conflicts and addressing issues related to the new formatting of the insight status. This task involved ensuring that the formatting of the status changed from "completed" to "done" when the queries were updated. This change was vital for maintaining consistency and clarity in our system.

Throughout the day, I meticulously addressed these issues, attempting to improve the feature. It was crucial to make the necessary adjustments promptly, considering that the feature was in a phase of active development.

By the end of the day, I had resolved the identified issues, enhancing the functionality of the discard feature and ensuring the consistency of the status formatting. The feature was now ready for another round of review and testing, bringing us one step closer to delivering a polished and reliable product to our users.

Thursday, October 12, 2023

Thursday's primary objective was to carry out another round of testing and debugging on my feature, addressing any potential bugs that might arise during this process.

The day began with an issue that caught my attention. When a user discarded an insight and immediately moved to the discard tab before the discard alert was dismissed, an unexpected problem surfaced. Trying to select the insight in the discarded list would break the user interface.

To resolve this, I chose a solution involving sending a request on view change (meaning when the user would change from viewing one status list to another with different status). This update ensured that the database would refresh accordingly, and when the user attempted to access the insight, the user interface remained intact. This adjustment was crucial in maintaining a smooth user experience.

In addition to debugging, I had the opportunity to showcase my features during our team's demo session. This provided a platform for my peers to explore and provide feedback on the features. The feedback I received was constructive and valuable, further enriching the development process.

Throughout the day, I carefully worked on addressing the issue I found, ensuring that the feature operated properly. The demo session not only allowed me to receive input from my colleagues but also served as an opportunity to present the progress made.

By the end of the day, I had resolved the identified issue, enhancing the feature's performance and user experience. The feedback from the demo session provided valuable insights for future improvements. The feature was now in a better state, ready for further testing and refinement, and moving slowly closer to its final form.

Friday, October 13, 2023

Today's primary goal was to prepare the feedback collection feature for approval and release. Additionally, I needed to participate in our team's refinement session, focusing on various key topics.

The day began with integrating code improvements into my work. This step aimed to improve the quality and functionality of the feature. In collaboration with a Senior Engineer, I tackled a cache management issue, ensuring the feature operated smoothly. Additionally, I addressed merge conflicts that arose during the integration process. These tasks were crucial for the overall development and readiness of the feature.

Later in the day, I participated in our team's refinement session. We discussed several vital aspects, including the start date for the pilot release with customers. This phase involved planning the release strategy, aligning with our objectives, and ensuring a smooth customer experience.

We also examined our new landing page plan of action, outlining the steps needed to bring it to development. Clear communication and collaboration were essential to ensure that the development process proceeded seamlessly.

Furthermore, the team evaluated the current situation of an ongoing experiment in our team, which involved implementing feature flags. This experiment was important for our future strategies and decisions, and discussions were essential to ensure its success.

Towards the end of the day, I successfully completed my feature, and it was approved for deployment. However, the deployment itself was scheduled for Monday.

3.4.1 Weekly analysis

In this section, we will analyze diary entries from the past week. Our focus will be on how these entries correspond to the core themes and overall objectives of this thesis project. The table below outlines the three main themes for the thesis and the relevant events from last week that connect to them.

Table 7. Fourth week theme review

| Themes | Events/Actions in the previous week |
|---|---|
| Efficiency in Feature Development vs. Time Management | - 3 story points completed after feature was ready for release. |

| | |
|--|---|
| Effective Communication | <ul style="list-style-type: none"> - Demo meeting. - Refinement session |
| Maintaining Code Quality and Collaboration | <ul style="list-style-type: none"> - Continuous reviewing and testing together with fixes and code improvements. |

Our primary focus last week in software development was maintaining both code quality and collaboration. These elements are essential for project success. In the fast-paced world of software engineering, well-written code serves as the foundation for a successful project. Throughout the week, we directly observed how code quality affects every aspect of software development. As we review our work, we will explore the key principles that emphasize the importance of code quality and its impact on the overall success of the project.

One of the key factors of maintaining the code quality in a software project is modern code review. Modern Code Review (MCR) is a widely adopted practice in both commercial and open-source software projects, aimed at ensuring software quality and detecting potential issues in code changes before they are integrated into the codebase. Unlike the formal and synchronous process of software inspection, MCR offers a more convenient and asynchronous environment for developers to review and discuss code changes, making it adaptable for geographically distributed teams. Various MCR platforms, including Gerrit, ReviewBoard, and Phabricator, support this practice. Effective MCR relies on best practices such as collaborative and timely review discussions, code updates, and knowledge transfer to enhance code quality, team awareness, and shared code ownership. However, it can be challenging to uphold these standards due to the multifaceted nature of code review, which involves technical, personal, and social aspects. (Chouchen & al. 2021.)

The daily entries exemplify the practical application of Modern Code Review (MCR) within the realm of software development. These entries report the thorough attention to detail and the iterative process that underlies MCR. In particular, the entries highlight the importance of fixing code issues, engaging in collaborative review discussions, addressing feedback, and conducting thorough testing—all essential components of MCR. They illustrate how MCR helps ensure software quality, prevent potential bugs from reaching the codebase, and foster a culture of shared code ownership and team awareness.

As Winters, Manshreck & Wright (2020) mentioned in their book. There is no magical day when you suddenly always know exactly what to do in every situation. Engineers, whether new to a team or in senior leadership, should continuously embrace the idea that there is always more to learn.

The misconception of equating 'seniority' with 'knowing everything' must be dismissed. Openly asking questions and acknowledging gaps in knowledge encourages a culture of growth and learning. On the receiving end, offering patience and kindness when answering questions fosters an environment where everyone feels safe to seek help. This collaboration accelerates productivity, benefiting the entire team.

It is important to understand as mentioned in previous chapters that continuous improvement is at the very heart of software development, in that sense, modern code review as a process for learning and knowledge share favors the development of individuals as engineers.

The previous analysis does not provide a definitive conclusion regarding the specific skills required to attain the status of a Senior Software Engineer. Seniority in software development is a multifaceted concept that encompasses a wide array of skills and expertise.

Nonetheless, it is evident that the continuous review processes undertaken in the past week have offered us valuable experience in an important aspect of software development. Regular code reviews not only improve code quality but also provide opportunities for learning, growth, and collaboration. These experiences contribute to our journey towards higher levels of seniority. It empowers us to refine our skills, adapt to the industry standards, and prepare for more senior roles.

3.5 Observation week 05

Monday, October 16, 2023

Today's main objectives were deploying completed feature to production (equivalent to three story points), attend my growth discussion with my Engineering Manager, and start a new task.

The day started with the deployment of the "collecting discard reason" feature. I had spent several weeks working on it and getting it up and running was a significant accomplishment. It represented the culmination of my efforts to improve system functionality and enhance the user experience. The successful deployment was proof of my hard work.

Later in the day, I had a conversation with my engineering manager about my career growth. We discussed my progress as a Junior Software Engineer and how we would assess my journey towards a mid-level position. My manager provided some valuable feedback, mentioning that he believed I already displayed the qualities required for a mid-level role. However, he emphasized the need to consult with other engineers who have worked closely with me to make a well-informed evaluation.

The growth discussion was a key point in my career, highlighting my professional development and the potential for advancement within the organization. The positive feedback served as motivation to continue working hard and attempt for more accomplishments.

To move forward with the development, I chose a new task involving the addition of test ids to monitor how much our users used specific features. This task presented another opportunity to contribute to the team's goals and gain insights into user behavior.

Tuesday, October 17, 2023

The primary objective for today was to integrate as many test IDs as possible into our codebase and make significant headway on a new task. Knowing I had a day off on Thursday, I was prepared to put in extra hours today if needed.

I began by working on the test ids integration into our components, as per my task from Monday. I made good progress, but I eventually reached a point where I had to pause. Not all the components that required these test ids were available or under development, so I could not keep implementing this.

With the test id task temporarily on hold, I shifted my focus to a new assignment. I created a pull request for this new task, which involved adding action buttons to inform users about upcoming features in our product. These buttons were intended to enhance the user experience and keep them updated about exciting upcoming changes.

During the day I also started another pull request for a different task. This one involved removing read/unread action buttons and statuses from our product's user interface and logic. This change was significant, restructuring the user experience and simplifying the interface.

I put in long hours of work throughout the day (approx. 12 hours), considering my upcoming day off on Thursday. It was crucial to make the most of my time and ensure that my tasks were moving in the right direction.

Wednesday, October 18, 2023

My main goal for the day was to continue the task I had started the previous day, identifying, and addressing any errors, testing the functionality developed the day before, and preparing the features for my upcoming day off on Thursday.

The day began with an important discussion with the designer. We had encountered an issue where the disabled buttons were not displaying correctly. We worked together to identify and resolve this issue. I will also be creating a report for our design system team (React Common Components team) to inform them about this issue. The problem stemmed from global styling influences originating from a common analytics solution used throughout the organization. This collaboration ensured that the disabled buttons would display correctly, maintaining a consistent user experience.

Next, I successfully completed the task of removing read/unread actions and status. This task was later technically approved through a pull request. The task improved the product's functionality, simplifying the user experience.

I also took part in a testing session for our team's latest experiment involving Feature Flags. We discussed how we could implement them as the technical side of the company was showing interest in their potential use. This session was important in exploring new possibilities for our product and features.

Later, I participated in a Frontend & UX guild meeting. During the meeting, we discussed the implementation of new typography components into our design system. I shared my prior experience with Posti's design system, which was received positively. The goal was to simplify Sievo's typography, aligning with the teams' efforts to create a more simplified design system.

By the end of the day, I had successfully completed the task related to removing read/unread functionality and implemented new action buttons as disabled buttons to provide users with a sneak peek at what's coming soon. This progress moved us closer to our goals and ensured that the product continued to evolve in the right direction.

In summary, Wednesday was a day of productive collaboration and problem-solving. I tackled technical challenges, participated in testing sessions, and actively contributed to discussions on improving our design system and user experience. My efforts were aimed at ensuring a smooth workweek, even with the day off approaching on Thursday.

Thursday, October 19, 2023 (Day off)

Friday, October 20, 2023

The main objective for the day was to put into action the features I had been developing throughout the week – removing read/unread actions and statuses task and implementing action buttons task. Additionally, I needed to choose my next task and begin planning its implementation.

The day started with some learning. I learned about the process of deploying our features to our development environment. This was important for deploying features to be tested specifically by the product side of our team, which includes managers and designers. This is particularly important to ensure that the requirement of features is aligned with the expectations of our business stakeholders.

After this learning phase, I moved on to deploying both features to the development environment separately. This step allowed our manager and designer to evaluate and give their approval for the changes. Their feedback was essential to ensure that the features aligned with our project's goals.

Once I received their approvals, I proceeded to the final stage of the development by making deploying the changes to our production environment.

Later in the day, I moved to reviewing my colleague's work. I identified a problem that had occurred after their deployment. After a thorough investigation, I discovered that the issue was related to an incorrect import of a dependency. I immediately created a pull request with a solution and, upon approval, merged it to resolve the issue. This collaboration and problem-solving were crucial to keeping our product running smoothly.

At the end of the day, I started planning my next task. This new task involved giving users the ability to assign insights to themselves, ensuring that only the assignee could make modifications (for example change statuses).

3.5.1 Weekly analysis

Over the course of the past week, our team's primary attention was directed towards product improvements. A significant portion of our collective efforts were dedicated to the creation and deployment of new features designed to improve our software. These new elements included the removal of read/unread actions and statuses, the introduction of action buttons, and the resolution of any related issues/bugs.

Beyond feature development, the week brought opportunities for learning and implementing deployment procedures, coupled with collaborative efforts with our team members for testing, gathering feedback, and securing the necessary approvals from stakeholders. It was a week characterized by hands-on development work and the persistent commitment to the continuous refinement and improvement of our product.

As Li, Ko & Begel (2019) discussed on their research to find the key qualities that set exceptional Software Engineers apart from their peers. Their comprehensive research involved surveying

1,926 engineers, ultimately pinpointing the top five distinguishing characteristics that make a software engineer truly great. Contrary to what we have discussed in previous weeks emphasizing the importance of soft skills in software engineering, their findings underscored the significance of technical prowess in an engineer's skill set. Notably, their research revealed that attributes such as keen attention to detail and the ability to tackle complexity were ranked highest among technical skills.

Based on the findings from this study, let's examine the events from last week. During this time, I successfully developed two separate features, which enhanced the overall product. Simultaneously, I managed a demanding workload that required frequent context switching, increasing the complexity of my tasks.

While a single week's perspective might be too limited to draw definitive conclusions about junior software engineers transitioning to senior roles, it's worth noting some positive developments. Last week's feature development was remarkably faster compared to previous weeks, potentially suggesting an improvement in my technical skills. As Figure 3 illustrates, my weekly story point completion reached 2, down from a previous maximum of 3. Figure 4 also shows an interesting pattern in the amount of work in progress each week.

Two key observations emerge from combining the data in Figures 3 and 4. First, it takes four weeks to complete a single feature, with three tasks remaining in progress from week 1 to week 4. Second, week 5 is unique: all tasks were both started and completed within the same week. This suggests an improvement in efficiency and possibly technical skills.

Looking at Figure 3, it's also worth noting the upward trend in overall task completion. This gradual increase over time highlights a positive trajectory in terms of productivity and skill development.

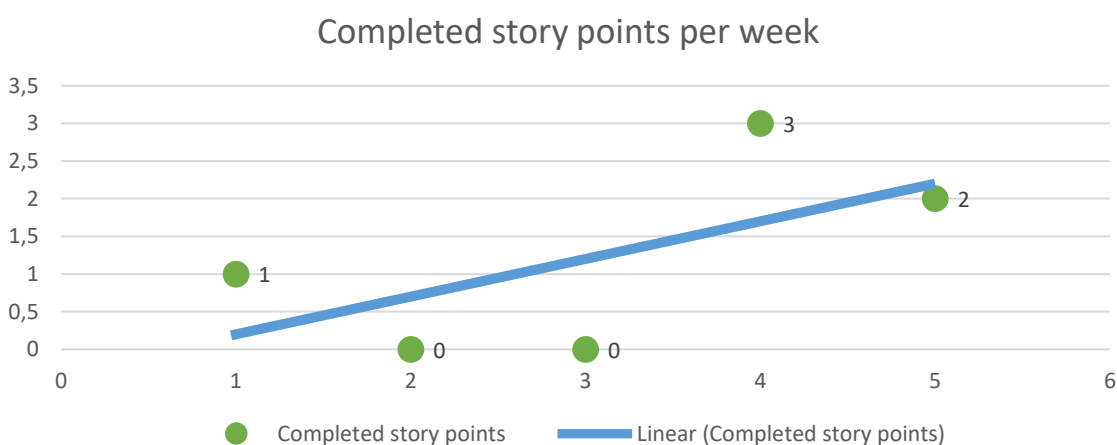


Figure 4. Completed story points per week and trend line.

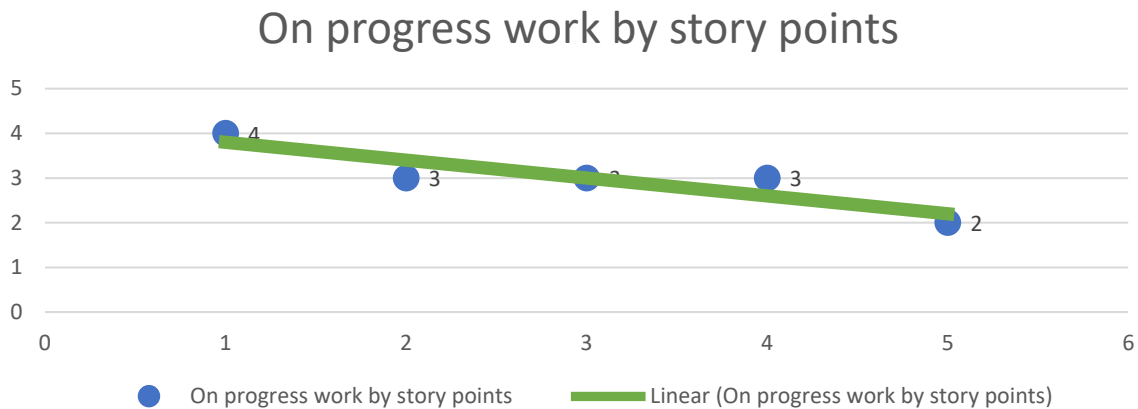


Figure 5. On progress work by story points per week.

In summary, our primary focus last week involved improving our software through the development of new features and refinements. This effort aligns with the emphasis placed on technical skills within software engineering, as highlighted by Li, Ko, and Begel in their 2019 study. My ability to complete features more quickly this week suggests a possible improvement in this area. While a single week's data may not be sufficient to draw definitive conclusions regarding the transition of junior engineers to senior roles, it does indicate a positive trend in terms of task efficiency. These observations underscore our ongoing commitment to continuous improvement and the critical role that technical expertise plays in the field of software engineering.

3.6 Observation week 06

Monday, November 13, 2023

I've been absent from work for the past two weeks due to unforeseen circumstances out of my control and because of that there is a gap between in the entries of this thesis work.

On Monday, our primary objective was to start a new task. The task required from us to develop a monitor and alerting system for our frontend development. The day started with a comprehensive investigation into potential solutions within the Azure environment for monitoring our application. Through this investigation, I discovered that Azure has a service that is used for monitoring applications called Azure Application Insights. I saw its potential use to become a solution because our architecture relies on Microsoft Azure services.

Aside from this investigative work, I collaborated with my colleagues. This involved approving, reviewing, and rigorously testing high-priority features earmarked in our internal pilot.

The day was marked by a dual focus – the exploration of a new monitoring system and hands-on collaboration to ensure the progress of key features within our software development pipeline.

Tuesday, November 14, 2023

On Tuesday, November 14th, my primary focus was to continue with the efforts to set up a monitoring and alerting system for our frontend development. I started by creating an Application Insights resource in Azure to begin implementing a potential solution.

Additionally, I spent time finding a way to store the necessary connection strings and keys from Azure Application Insights in our repository securely. For this I read Microsoft Azure documentation online and explored the company's documentation. This task aims to ensure the integration process is secure.

Simultaneously, I continued being involved in the day-to-day collaboration with my colleagues. I actively participated in reviewing and testing high priority features essential for our internal pilot.

The day was characterized by hands-on work, working towards improving our monitoring capabilities and contributing to the smooth progress of critical features in our software development pipeline.

Wednesday, November 15, 2023

On Wednesday, my main objective was the continued implementation of the monitoring and alerting system for our application.

An important part of the day was dedicated to resolving challenges encountered connecting our repository and the Azure Application Insights monitoring system. After several attempts, I successfully identified the required libraries and configurations to facilitate the transmission of monitoring events, leaving only the actual connection to be established.

Alongside, I investigated the secure storage of connection strings and keys from Azure Application Insights within our repository to ensure reliable integration.

Additionally, I actively engaged in the code review process, assessing multiple pull requests from my colleagues, addressing high-priority business requirements, e.g. the addition of new navigation links to our application.

This collaborative effort was necessary to progress on a critical feature essential for our ongoing pilot initiatives. The day was marked by a combination of problem-solving, investigation, and collaborative contributions. This teamwork ensured we advanced our monitoring capabilities and met key business requirements.

Thursday, November 16, 2023

On Thursday, November 16th, my primary goal was to establish a connection between Azure Application Insights and our frontend repository.

Following the challenges encountered in previous attempts, I persistently worked on connecting the monitoring system. However, I faced difficulties that required me to seek assistance on our Slack engineering channel. There, I learned that our company already has a monitoring system in place that oversees all frontend applications.

Considering this information, I shifted my focus to investigating the utilization of the existing monitoring system. I explored how to implement alerts for specific events, such as application downtime or uncaught exceptions in the browser.

Simultaneously, I continued to contribute to the team's rapid development efforts by providing support through reviews, testing, and quality assurance. The day captured a combination of research and collaborative efforts to enhance our monitoring capabilities and facilitate the ongoing development of features crucial for our pilot initiatives.

Friday, November 17, 2023

On Friday, November 17th, my primary goal centered around investigating the implementation of the existing Azure Application Insights monitoring and alerting system for all the solutions within the company.

To achieve this, I researched into the logs generated by the Azure Application Insights solution, which is globally implemented for all the frontend services across the organization. During this research, I identified several queries within the logs that can be used to extract information related to downtime and specific exceptions, potentially serving as the foundation for creating alerts.

Simultaneously, I remained actively engaged with my team, contributing to the review and acceptance process for the latest features developed for our application's frontend. This collaborative effort aimed to ensure that our application was ready for the upcoming piloting phase.

The day was a mix of investigative work and ongoing team collaboration to enhance our monitoring capabilities and advance the development of critical features.

3.6.1 Weekly analysis

Our primary focus last week was on researching and testing a monitoring and alerting system for our frontend application using Microsoft Azure. We collaborated closely with fellow Engineers to expedite the review, testing, and acceptance processes for high priority features crucial to the upcoming release of the internal pilot of our application.

To better understand last week's work, let's consider the current stage of our product development. Our product, in development since January 2023, has a scheduled pilot launch with our first customer on November 30th. Before this external launch, however, an internal pilot is crucial to gather feedback from our company's most experienced personnel. This internal testing ensures the product's readiness for customer use.

Since last week marked the final week before the internal pilot, the team's primary focus was on releasing the final features. While I wasn't assigned top priorities, I played a vital role by reviewing code quality and testing features within our development environment.

Within this context, it is important to highlight that the work of software development involves research and continuous learning skills. As Li & al. (2019) showed in a study that explored what are the skills that separate the great engineers from ordinary ones. For their study they interviewed 59 experienced engineers across 13 divisions at Microsoft collecting insights to determine certain qualities that make engineers great. Li & al. showed that as the software engineering world and the technology industry is subject to rapid changes and constant upgrades, great Software Engineers have good skills in continuous learning to keep up with the rapid changes of the industry maintaining themselves up to date.

Software development teams often encounter needs outside their immediate area of expertise. This necessitates research, investigation, and a trial-and-error process to find solutions best suited to the specific context and organization of the team. While last week's progress on the monitoring and alerting system may not have been ideal, these initial steps provided valuable learning experiences. This newfound knowledge will benefit you as you progress towards more senior software engineering roles.

As I mentioned in the previous week's analysis, code review plays a crucial role in the software development process as it ensures software quality. Bavota & Russo (2015) found in their study of three large open-source systems that unreviewed commits were over twice as likely to introduce

bugs as reviewed commits. Additionally, they found that code committed after review had substantially higher readability compared to unreviewed code.

IT professionals continually enhance their skills over their careers, with senior software engineers exhibiting the highest technical competencies. Non-technical competencies, however, become more prominent at the management level. Additionally, the significance of non-technical skills tends to increase as IT professionals progress in their careers, surpassing the importance of technical skills over time(Wurzel Gonçalves, Bacchelli, Çalikli & Serebrenik 2023).

Code review seems to be a tool to foster a combination of technical expertise and communication skills that is necessary for Junior Software Engineers to proceed into more senior roles. As said before, Software Engineers seem to rely on technical skills strongly at the beginning but as their careers progress, their communication role takes more importance compared to technical expertise.

Due to this we think that last week's experiences might enhance our capabilities and give valuable insights towards finding what are the set of skills needed for novice Software Engineers to move into more senior roles.

3.7 Observation week 07

Monday, November 20, 2023

The primary goal for Monday was to investigate and begin the implementation of alerts based on queries derived from the logs in Azure Application Insights.

I started the day by exploring how to configure alerts in the Azure portal using the queries identified during the log analysis on Friday. The objective was to gain a practical understanding of setting up alerts for effective monitoring and timely responses to potential issues.

Following this, I created the initial version of the alerting system, focusing on tracking exceptions in our frontend application. However, the results were not as anticipated, and it became evident that the alert was not functioning correctly. This led me to do further investigation about the alert configuration to ensure that notifications to our designated Slack channel functioned properly when triggered.

To address the challenges encountered during the initial implementation, I continued to explore and adjust the alerting system's configuration settings. This iterative process is essential for fine-tuning the alerting mechanism to align with the specific requirements of our application.

In addition to the development work, I dedicated time to reviewing several pull requests related to UX fixes. Collaborating with fellow developers in code reviews not only ensures the integration of high-quality code into the codebase but also fosters a shared understanding of best practices and design principles within the team.

Tuesday, November 21, 2023

Tuesday's main objectives were to address issues with the alerting system's configuration and actively contribute to setting up a new pilot customer for our application.

The day began with a thorough examination of the alerting system to understand why alerts were not triggering notifications on the designated Slack channel. Thanks to this process I found out a couple of important facts: our team was the first to implement an alerting system for the frontend application, there was a crucial element missing in the alert configuration – a payload specifying the Slack channel information. Even if this discovery clarified the issue, further research was necessary, given the absence of similar examples within the company.

To address the issue in the alerting system, I needed to research more to find what would be the correct payload for the alert.

During the day, I also actively participated in a setup workshop for a new pilot customer. The meeting involved a comprehensive walkthrough of the steps needed to establish a new customer's access to our application (Insights product). This included configuring their data in the databases, running the data through the calculation engines, and configuring our frontend to restrict access until the pilot phase.

Wednesday, November 22, 2023

The primary focus for Wednesday was to apply the insights gained from the alerting system research and configure our alerts correctly, ensuring they notify our designated alert channel.

Continuing from the progress made on Monday and Tuesday, I concentrated on understanding how to integrate a custom payload into the alert request. This effort resulted in the successful implementation of the custom payload in Azure alerts, sending the alerts to our designated Slack channel for our development environments.

To keep the team informed, I communicated that they shouldn't be concerned about alerts notifying our Slack channel, as testing was needed to find the most appropriate threshold for alert triggering.

During the testing phase, I observed that the alerts were triggering notifications to the Slack channel every 15 minutes. Recognizing this as unexpected behavior, I dedicated time to debugging the alerting system. After a few hours of investigation, I identified that the threshold for the alerts had been incorrectly set. So, I corrected the threshold, resolving the issue and ensuring that the alerts were now functioning properly.

Thursday, November 23, 2023

The primary goals for Thursday were to present a demonstration of the solutions developed over the past two weeks in our biweekly demo sessions, providing the team with insights into the capabilities of the monitoring and alerting system and to develop the monitoring and alerting system in our production environment.

The day began with an extensive demo of the monitoring and alerting system for my team. I explained its functionality, guided the team on its usage, and clarified the types of alerts they could expect in various scenarios, such as downtimes.

A significant portion of the day was dedicated to implementing the monitoring and alerting system for our production environment. This involved obtaining special permissions, requiring a request through Azure Privileged Identity Management (PIM) to create alerts in the production Application Insights resource.

To keep the updated, I communicated the implementation of the alerting system in the production environment. I explained that the alert production channel would notify some test alerts as part of my ongoing testing to ensure the system's functionality met expectations.

Simultaneously, I documented the implementation of the monitoring and alerting system. The documentation summarized how to use the monitoring solution to explore logs and retrieving data related to our service. It also detailed the process of setting alerts based on specific queries within the monitoring solution and provided instructions on configuring custom payloads to notify Slack channels. The documentation encapsulated key findings from the research and implementation phases of the monitoring and alerting system task.

Friday, November 24, 2023

Today's main goal was to start a new task. The selected task related to User Experience (UX) fixes, specifically, to modify the summary text of our insights.

I began the day by successfully updating the summary text for our insights. This involved fixing issues, creating a pull request, conducting testing, and obtaining acceptance from the management team.

Additionally, I participated in a team refinement session where we discussed a new business requirement. This business requirement involved displaying the top insights in a widget style component on a new page within Sievo's service. The session resulted in a practical plan for approaching this requirement. While the frontend work is set to start in a couple of weeks, the session drew a frame for a coordinated strategy, ensuring alignment between the frontend, analytics, and backend areas.

Simultaneously, I started a new task related to fixing a bug in the summary count of insights when using filters. The task involves the following behavior: Applying filters should only decrease the total number of insights. However, in some cases, we've observed an increase in the count of insights after applying filters.

This task will require careful investigation and correction to ensure the accuracy of insights counts during filtering.

3.7.1 Weekly analysis

In this section, we will analyze the events of the previous week. As in previous weeks, our analysis will attempt to explore what are the skills needed for a Junior Software Engineer to move into more senior roles. To facilitate this analysis, we will focus on the main events of the last week and organize these events into the three main themes of our professional development as can be found in Table 8.

Table 8. Seventh week theme review

| Themes | Events/Actions in the previous week |
|---|--|
| Efficiency in Feature Development vs. Time Management | - 2.5 story points completed during approx. a week |
| Effective Communication | - Demo meeting. - Refinement session |
| Maintaining Code Quality and Collaboration | - Review and quality assurance of colleague's tasks and pull requests. |

Table 8. provides an overview of last week's events organized by each main theme of this thesis. Nevertheless, our analysis needs to explore these events in more depth to gain insights on what are the skills that can be gained from them, as well as, how these skills can make the difference in a career growth perspective going from a junior to a more senior position as a Software Engineer.

During the past week, my primary tasks centered on developing a monitoring and alerting system for the Insights frontend. To begin, I conducted a thorough review of the current monitoring practices within the company. This involved discussions with key stakeholders who possessed valuable insights into existing processes. Additionally, I researched potential solutions on the internet to broaden my perspective. Combining insights gathered from Sievo's practices, stakeholder discussions, and external research, I proceeded to implement a monitoring and alerting solution for our team's frontend.

While this implementation was not without challenges, overcoming these obstacles demonstrated my problem-solving skills and determination to ensure the successful implementation of the solution.

These tasks highlighted the important role of soft skills in software engineering. Problem-solving was crucial in understanding and developing the monitoring and alerting system, while effective communication played a key role in engaging with stakeholders. Creativity was essential in exploring and integrating solutions from research. Overcoming errors during implementation emphasized the importance of resilience. In the field of software engineering, possessing soft skills like creativity, and effective communication is essential for addressing complex challenges and achieving comprehensive solutions aligned with both technical and business objectives.

Software development involves numerous tasks that require problem-solving abilities. Developers must plan strategies to address challenges and generate creative and innovative solutions. Hence, among the various skills required, high analytical problem-solving skills and creativity are necessary for software developers. These skills are essential to the complex and intellectual nature of software development, which relies on cognitive processing abilities. (Graziotin, Wang & Abrahamsson 2014.)

Wynekoop & Walz (2000) have explored the challenges associated with recruiting, developing, and retaining top-performing IT professionals. In their study, outstanding developers were described as individuals with traits such as self-confidence, high motivation, and creative problem-solving skills. These developers also exhibited exceptional leadership, organizational capabilities, analytical and technical skills, along with effective communication skills and the ability to collaborate well within teams.

Soft skills play a significant role in the demand for software developers. Problem-solving skills, identified as one of the most desirable soft skills, were mentioned in 69% of advertisements for IT positions. Communication skills topped the list with a demand of 92%, while creative skills were among the least in demand at 9%. (Ahmed, Fernando Capretz & Campbell 2012.) However, as revealed by Wynekoop and Waltz (2000), managers characterize exceptional developers as individuals who not only excel in technical expertise but also demonstrate a blend of soft skills, including creativity. Agile teams require a broader skill set that extends beyond technical proficiency, encompassing communication, team building, flexibility, creativity, leadership, and stress management abilities. (Ćirić, Stančetić, Gračanin & Ćurčić 2020.)

In summary, the tasks from last week focused on developing a monitoring and alerting system, highlighting the critical relationship between technical competence and soft skills. Addressing implementation challenges stresses the importance of problem-solving and resilience, essential attributes for advancing professional growth. As the demand for skilled IT professionals rises, understanding traits outlined by Wynekoop and Waltz (2000) becomes essential for the progression from junior to senior roles.

The diverse landscape of software development requires skills like problem-solving, communication, and creativity, all crucial for assuming senior-level responsibilities. Soft skills, often overlooked, have a decisive impact on the effectiveness of software developers, particularly in the flexible setting of agile teams (Ćirić & al., 2020). In the domain of software engineering, recognizing, and fostering these skills is fundamental for navigating the transition from a junior to a more senior position, ensuring both individual and organizational success.

3.8 Observation week 08

Monday, November 27, 2023

Monday's main goal was to find the source and resolve an issue with the summary count of insights.

The issue with the displayed count was that the summary count of insights, initially representing the "total" amount of insights for a specific type, increased when applying certain filters, such as the category "product". This issue made our user experience less user friendly.

The day began with an attempt to replicate the bug by exploring every possible combination of filters and insight types. Surprisingly, I could not reproduce the issue. Seeking further clarity, I decided to visit the specific environment where the problem was observed and successfully replicated it there.

After reproducing the bug, I contacted an Analytics Engineer (AE) from my team. The AE suggested creating a list in Microsoft Excel to document both cases (environment with bug and without it) to make the debugging process easier. Additionally, the AE offered to assist in checking the integrity of the data in our analytics solution, Qlik Sense.

I proceeded to create the list for comparison, and to my surprise, only 7 insights matched between the 2 environments using the same filter configuration. I shared the list and my findings with the AE which led him to take the task on its own to further investigate the data side of the insights.

This means that I will need to check with my managers what are the next priorities regarding feature development as our product will start the piloting phase in the coming weeks.

Tuesday, November 28, 2023

Tuesday's primary goal was to start a new task related to displaying the discard reason on discarded insights, marking the conclusion of the discard reason epic initiated months ago, almost at the beginning of these diary entries.

The day started by planning and start implementing the discard reason display task. This involved identifying the areas for adding new components, clarifying acceptance criteria, and reviewing designs.

However, I had to temporarily shift my focus due to a high-priority request from our business stakeholders. The mentioned task requirement was to change the position of certain elements in the UI of our product. I quickly started the development of this feature, after a couple of hours the feature was ready and approved to be deployed.

After addressing the urgent task, I returned to the original discard reason display task. With all necessary clarifications obtained, I began laying out the components in the codebase.

Wednesday, November 29, 2023

The main goal for today was to continue the development of the discard reason display.

The day began with the daily meeting. During the meeting engineers were requested to assist in setting up the Insights product for a new customer environment. I volunteered for the task.

Setting the Insights product for a new customer environment involves extensive preparations on the production databases and following a series of meticulous steps.

Although I initially focused on continuing the development of the discard reason display, the need to prioritize setting up the Insights product for the new customer environment emerged as a higher-priority task. Consequently, I changed my focus to this critical assignment.

Most of my day was dedicated to setting up insights for the new customer. This involved logging into various databases to add insights, configuring our analytics solution, running pipelines and calculation engines, data migrations, and fine-tuning UI components to ensure the customer would not see our application until the start of the pilot. All the previously mentioned steps were done in collaboration with Analytics Engineers.

By the end of the day, I successfully completed the setup for the new customer. I could not progress so much with the task that was my original goal for the day, but I completed a high priority task bringing our product to more customers.

Thursday, November 30, 2023

The primary goals for Thursday were to improve the documentation for enabling Insights for new customers, making it more accessible to non-expert users, and to make progress on the discard reason display task.

I started the day with a comprehensive review of the documentation for enabling the Insights product for new customers environments. Based on my own experience setting up our application for a new customer, I identified gaps in knowledge and made necessary modifications to improve clarity. Once the revisions were complete, I asked feedback from my teammates, receiving positive responses on the improved comprehensibility of the documentation.

Around noon, I participated in our retrospective meeting. This session served as an opportunity for the team to collectively analyze and discuss potential improvements to our team's functioning.

After the retrospective meeting, I continued the implementation of the discard reason display. The focus was primarily on refining the rendering logic and date formatting to improve the overall user experience.

Friday, December 01, 2023

Friday's primary objective was to finalize all requirements for the display discard reason feature, ensuring readiness for a code review on Monday.

I dedicated the morning to final evaluations at the university.

In the afternoon, I successfully completed all the technical requirements for the discard reason display feature. This included addressing various scenarios, such as rendering two selectable discard options, only one option, all selectable plus free text reason, free text reason option only, and scenarios with no discard reason. Each scenario was implemented following the design specifications provided by our team's designer.

In addition to the technical achievements, I had a one-to-one meeting with my team's Engineering Manager (EM). During this meeting, my team's Engineering Manager shared the current plans for our application and provided positive feedback, acknowledging my proactive contribution as a volunteer for setting up the Insights product for a new customer environment. The quick completion of that task was also recognized.

3.8.1 Weekly analysis

In this section we will analyze last week's events. To begin, let's examine the events and gather them around the 3 main themes of this thesis work having in mind the main goal of find insights on what skills are needed from Junior Software Engineers to move into more Senior roles.

Table 9. Eighth week theme review

| Themes | Events/Actions in the previous week |
|---|--|
| Efficiency in Feature Development vs. Time Management | <ul style="list-style-type: none"> - Handling rapid switch of tasks and deliver them. |
| Effective Communication | <ul style="list-style-type: none"> - Retrospective meeting - Taking high priority tasks covering the needs of business stakeholders. |
| Maintaining Code Quality and Collaboration | <ul style="list-style-type: none"> - Effectively collaborate with analytics engineers to find out source of issues. |

In the past week, the efforts were divided and rapidly changing from one task to another due to urgent requirements from business stakeholders. The rapid context switching, and the urgency of the features forced me to gather all the knowledge I had of our application and manage different levels of complexity within the tasks. In practice, I worked on 3 different tasks related to debugging, enabling our service to new customers, and displaying discarded reasons.

Last week's experiences emphasized the importance of handling different levels of complexity while developing software and the ability to change contexts smoothly from different tasks depending on stakeholders' requirements. In addition, effective communication and collaboration played a key role in connecting the pieces of a team to solve complex problems.

Software engineers are required to manage varying levels of complexity based on the nature of their assignments, representing a fundamental skill in their professional toolkit (Sedelmaier & Landes 2014). As part of their education, software engineering students are actively instructed in complex problem-solving skills to enhance their employability and equip them for the multifaceted challenges prevalent in the field (Schefer-Wenzl & Miladinovic 2019).

The demand for proficient handling of complexity in software engineering is emphasized by the escalating challenges faced by software companies within today's dynamic business environment (Rikkila, Abrahamsson & Wang 2012). Rapid technological advancements, evolving user expectations, and the intricate interplay of software systems contribute to an environment where the ability to navigate complexity is a critical asset. As such, the importance of cultivating and refining these skills becomes evident not only for individual software engineers but also for the overall resilience and adaptability of the software industry.

It is also worth mentioning that the rapid change in the industry landscape and the rapid evolution of business needs forces Software Engineers to not only handle multiple levels of complexity but also being able to change tasks depending on business needs.

Frequent context-switching in software development poses a significant challenge, potentially resulting in distraction, sub-standard work, and increased stress (Vasilescu & al. 2016). This issue is particularly pronounced in high-momentum tasks like software development, where maintaining a state of "flow" is crucial for optimal performance (Zahra, Noaen, Zowghi, Far & Barker 2018). The dynamic nature of software development situational contexts adds an additional layer of complexity to this challenge (Clarke & Connor 2015).

Nevertheless, effective mitigation strategies can be employed, and motivation emerges as a pivotal factor in overcoming these obstacles. Various motivators, such as competitive pay, acknowledgment of contributions, and opportunities for achievement, play a vital role in sustaining focus and productivity in software development (Baddoo, Hall & Jagielska 2006). Recognizing and addressing the impact of context-switching on motivation is essential for fostering an environment favorable to sustained concentration and high-quality work in the field of software development.

In summary, there seems to be a conflicting part of Software Engineering where from one side, Engineers need to prepare to solve problems in all different levels of complexity, for this purposes,

aspiring Software Engineers spend time in getting skilled at solving complex problems. In the other side, due to the changing nature of the technology industry, Software Engineers are subject to the stress and disruption on their workflows cause by changing the context of the current tasks to perform activities like meetings, higher priority tasks, etc.

Even though last week's entries don't provide a clear skill set that Junior Software Engineers need to have to advance to more senior roles, they do share insights on what are the certain behaviors that Software Engineers should avoid or should learn to cope with (like context switching) and shows that complex solving problem is a necessary skill that must be develop by software engineers from early stages in their careers.

4 Discussion

In this chapter we discuss the progress in professional growth that we have made over the reporting period. At the same time, we will discuss what are the skills that could help Junior Software Engineers to move into more senior roles.

Throughout the course of the reporting period, we have explored what might be the skills that define seniority. These skills can be categorized into communication skills and technical skills.

By interacting with business stakeholders almost daily and learning from Senior Engineers we have gained the understanding that Senior Software Engineers are proficient communicators. This skill does not only extend to the ability to translate technical requirements into a common language to interact with non-technical stakeholders but also to the ability to transform business requirements into technical specifications. Moreover, we have experienced that an important part to what Senior Software Engineers do in their communication is to translate the “why” are technical requirements needed in certain ways. To put an example, let’s imagine we are engineers trying to request time to make some improvements in performance in certain API responses. We could go to our manager with a lot of technical requirements and why this is technically relevant, in the other hand, a more Senior Engineer acknowledges the fact that business stakeholders might not relate to technicalities, instead a more senior approach would be mentioning how users can benefit, or how our company can benefit from those improvements.

Despite the importance of communication in the role of Senior Software Engineers, code cannot be written just by communicating. Technical skills play a central role in engineering software. By understanding features in their context Senior Engineers can anticipate errors and refine their code better and in less time, bringing value to the organizations more effectively.

There are also skills that cannot be categorized as only technical or communication skills but rather a combination of both. Code review is an activity where Senior Software Engineers show their capabilities in technical skills; by finding improvement areas, suggesting changes, and overall ensuring the quality of the code delivered, and show their skills in communicating; by encouraging less experienced engineers to improve, adapt to best practices, and grow. This communication and technical combination can also be seen in mentoring activities as pair coding, where Senior Engineers and Junior Engineers code together going through the codebase and sharing their thought process.

Regarding professional development, throughout these 8 weeks we have gained experience in many areas of the software engineering field. We have increased our proficiency in areas like programming with React, source control with Git, styling using styled-components, type safety using TypeScript, managing data queries with React Query, and other technologies.

Many novice Software Engineers struggle from the so called “impostor syndrome” which is related to the feeling of not having the enough knowledge to perform one’s work. We have seen improvements in this area as we have developed many features, not only has our programming proficiency increased but also our self confidence to resolve problems independently and proactively.

Having more experience on the field has also allowed us to interact more smoothly with our team and justify our technical decisions to other Engineers. This shows that not only technical and communication skills are required but also to trust that those skills will carry us far.

Besides the mentioned, one of the main issues I experienced at the beginning of the reporting period was the inability to see how the pieces of our software product were interconnected. I think that one of the biggest improvements in my professional development has happened in this area. After the reporting period I was able to better understand features in their context, thinking in their dependencies and where were the source of the issues I have been solving. I have also developed certain strategies to attempt to break problems into smaller pieces and in this way facilitate the development of features.

During these weeks I have also gained some knowledge into how I see my future as Engineer. Before I started the reporting period I had as a dream job to be a “Software Architect”, this vision of my future has only been strengthened during this period and now I am one step closer to reach that goal.

Through our work we have not only contributed to our self-development as professionals but also to our organization and team’s success. During the reporting period we briefly described that our product was starting its first customer pilot aiming to test our solution with real customer data and receive feedback from them to make later improvements. I not only participated in the development of features that were included in the pilot release, but also set up the insight’s product in customer’s environments. This experience is also valuable for my own knowledge because I collaborated in the product from its early stages gaining understanding in architectural decisions and different strategies to develop software.

Moving forward, I plan to continue using the work analysis approach implemented in this diary thesis. I strongly believe that consistent progress reporting and self-analysis will enhance my understanding of professional development and allow me to effectively communicate my contributions to

the team and organization during discussions about salary, promotions, networking, and recruitment for different positions.

At the beginning of the reporting period, I was recently hired into the company, and I wanted to put all my effort into getting promoted, which not only was the inspiration for this thesis but also the driving force to work hard and keep growing as a professional. I'm gladly able to say as I finish this thesis work that I no longer hold the title of Junior Software Engineer, but I have now been promoted to a mid-level position as a Software Engineer.

References

Ahmed, F., Fernando Capretz, L. & Campbell, P. 2012. Evaluating the Demand for Soft Skills in Software Development.

Azure Application Insights - Connectors | Microsoft Learn s.a. URL: <https://learn.microsoft.com/en-us/connectors/applicationinsights/>. Accessed: 1 May 2024.

Baddoo, N., Hall, T. & Jagielska, D. 2006. Software Developer Motivation in a High Maturity Company: a Case Study. URL: https://d1wqtxts1xzle7.cloudfront.net/47403327/spip.26520160721-10288-1pn5khz-libre.pdf?1469108892=&response-content-disposition=inline%3B+file-name%3DSoftware_developer_motivation_in_a_high.pdf&Expires=1704038609&Signature=YyJyGyeBbYv~XttTD4I0njvBVJhPsu2uhxHdwUFbep6~Lt1lxTOS9YoeEdyrAGW9SrcRxHYje-jXa6WJ7ZS35e9utjuRCUpUpQp-dRoF00N8mdjEUu11ing4MK2BVTb5dMYm1oK5VQ~VzgtPpf5u4dwDclKBBI3VpvNKdHYg22fDd-6BCw4bmjeFH3gxqKtx3pv-AFVmRXtzbukx36y3LbY9oHDqc-sibg6VJeCaECzyEAjhdaGF53nkXFIO4nat0ZKw7jW6P3-hxeD4K58NMYWCXrJASILE2WD4BUO-EYMr37cctalTliekJsKRyvSUCOWfy~xgksLI0eQDhaB-Nc~MYw__&Key-Pair-Id=APKAJ-LOHF5GGSLRBV4ZA. Accessed: 31 December 2023.

Bavota, G. & Russo, B. 2015. Four Eyes Are Better Than Two: On the Impact of Code Reviews on Software Quality. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7332454>. Accessed: 24 February 2024.

Chouchen, M., Ouni, A., Kula, R.G., Wang, D., Thongtanunam, P., Mkaouer, M.W. & Matsumoto, K. 2021. Anti-patterns in Modern Code Review: Symptoms and Prevalence. Institute of Electrical and Electronics Engineers Inc. URL: <https://doi.org/10.1109/SANER50967.2021.00060>.

Ćirić, D., Stančetić, M., Gračanin, D. & Ćurčić, J. 2020. The Importance of Non-technical Skills in Agile Project Teams. Springer Nature. URL: https://doi.org/10.1007/978-3-030-43616-2_34.

Clarke, P. & Connor, R. V. 2015. Changing situational contexts present a constant challenge to software developers. 22nd European Conference on Systems, Software and Services Process Improvement (EuroSPI 2015), Springer-Verlag. URL: <https://core.ac.uk/download/pdf/147608827.pdf>. Accessed: 31 December 2023.

Flood, D. 2015. Communication is Key to Software Engineering | LinkedIn. URL: <https://www.linkedin.com/pulse/communication-key-software-engineering-donnie-flood/>. Accessed: 8 October 2023.

GeeksforGeeks 2023. Frontend vs Backend - GeeksforGeeks. URL: <https://www.geeksforgeeks.org/frontend-vs-backend/>. Accessed: 23 September 2023.

Goodliffe, P. 2014a. Becoming a Better Programmer. O'Reilly Media. Sebastopol. URL: <https://learning.oreilly.com/library/view/becoming-a-better/9781491905562/ch10.html#idp1161968>. Accessed: 1 October 2023.

Goodliffe, P. 2014b. 36. Speak Up! | Becoming a Better Programmer. O'Reilly Media, Inc. Sebastopol. URL: <https://learning.oreilly.com/library/view/becoming-a-better/9781491905562/ch36.html#idp3782128>. Accessed: 8 October 2023.

Graziotin, D., Wang, X. & Abrahamsson, P. 2014. Happy software developers solve problems better: Psychological measurements in empirical software engineering. PeerJ, 2014, 1. URL: <https://doi.org/10.7717/peerj.289>.

Hewlett Packard Enterprise s.a. What is a Front-End Application? | Glossary | HPE. URL: <https://www.hpe.com/us/en/what-is/front-end-applications.html>. Accessed: 23 September 2023.

How does Azure work? - Cloud Adoption Framework | Microsoft Learn s.a. URL: <https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/get-started/what-is-azure>. Accessed: 1 May 2024.

Li, P.L., Ko, A.J. & Begel, Andrew 2019. What distinguishes great software engineers? Empirical Software Engineering. URL: <https://doi.org/10.1007/s10664-019-09773-y>. Accessed: 22 October 2023.

Li, P.L., Ko, A.J. & Zhu, J. 2015. What makes a great software engineer? IEEE Computer Society. URL: <https://doi.org/10.1109/ICSE.2015.335>. Accessed: 24 September 2023.

Martin, R.C. 2011. Clean Coder, The: A Code of Conduct for Professional Programmers. URL: <https://learning.oreilly.com/library/view/clean-coder-the/9780132542913/ch01.xhtml>. Accessed: 24 September 2023.

MDN web docs s.a.a. UI - MDN Web Docs Glossary: Definitions of Web-related terms | MDN. URL: <https://developer.mozilla.org/en-US/docs/Glossary/UI>. Accessed: 23 September 2023.

MDN web docs s.a.b. API - MDN Web Docs Glossary: Definitions of Web-related terms | MDN. URL: <https://developer.mozilla.org/en-US/docs/Glossary/API>. Accessed: 23 September 2023.

Microsoft 2022. What is Azure DevOps? - Azure DevOps | Microsoft Learn. URL: <https://learn.microsoft.com/en-us/azure/devops/user-guide/what-is-azure-devops?view=azure-devops>. Accessed: 23 September 2023.

Mongare, S. s.a. The Most Important Skills for a Senior Software Engineer - DEV Community. URL: <https://dev.to/msnmongare/the-most-important-skills-for-a-senior-software-engineer-32m3>. Accessed: 1 October 2023.

NPM, Inc. s.a. npm About. URL: <https://www.npmjs.com/about>. Accessed: 23 September 2023.

Postman API Platform | Sign Up for Free s.a. URL: <https://www.postman.com/>. Accessed: 1 May 2024.

ProductPlan s.a. Fibonacci Agile Estimation | Definition and Overview. URL: <https://www.productplan.com/glossary/fibonacci-agile-estimation/>. Accessed: 24 September 2023.

Rikkila, J., Abrahamsson, P. & Wang, X. 2012. The Implications of a Complexity Perspective for Software Engineering Practice and Research. *Journal of Computer Engineering & Information Technology*. URL: <https://doi.org/10.4172/jceit.1000e103>. Accessed: 31 December 2023.

Sadowski, C., Söderberg, E., Church, L., Sipko, M. & Bacchelli, A. 2018. Modern Code Review: A Case Study at Google. URL: <https://doi.org/10.1145/3183519.3183525>. Accessed: 8 October 2023.

Schefer-Wenzl, S. & Miladinovic, I. 2019. Developing Complex Problem-Solving Skills: An Engineering Perspective. URL: <https://doi.org/10.3991/ijac.v12i3.11067>. Accessed: 31 December 2023.

Sedelmaier, Y. & Landes, D. 2014. Software Engineering Body of Skills (SWEBOS). URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6826125>. Accessed: 31 December 2023.

TanStack Query Docs s.a. Overview | TanStack Query Docs. URL: <https://tanstack.com/query/v4/docs/react/overview>. Accessed: 23 September 2023.

Vasilescu, B., Blincoe, K., Xuan, Q., Casalnuovo, C., Damian, D., Devanbu, P. & Filkov, V. 2016. The Sky Is Not the Limit: Multitasking Across GitHub Projects. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7886974>. Accessed: 31 December 2023.

What is a CI/CD pipeline? 2022. URL: <https://www.redhat.com/en/topics/devops/what-cicd-pipeline>. Accessed: 1 May 2024.

What is Privileged Identity Management? - Microsoft Entra ID Governance | Microsoft Learn 2023. URL: <https://learn.microsoft.com/en-us/entra/id-governance/privileged-identity-management/pim-configure>. Accessed: 1 May 2024.

What is Slack? | Slack s.a. URL: <https://slack.com/intl/en-gb/help/articles/115004071768-What-is-Slack->. Accessed: 1 May 2024.

What is the Azure portal? - Azure portal | Microsoft Learn s.a. URL: <https://learn.microsoft.com/en-us/azure/azure-portal/azure-portal-overview>. Accessed: 1 May 2024.

What is version control | Atlassian Git Tutorial s.a. URL: <https://www.atlassian.com/git/tutorials/what-is-version-control>. Accessed: 1 May 2024.

Winters Titus, Manshreck Tom & Wright Hyrum 2020. 3. Knowledge Sharing | Software Engineering at Google. URL: https://learning.oreilly.com/library/view/software-engineering-at/9781492082781/ch03.html#psychological_safety_in_large_groups. Accessed: 15 October 2023.

Wurzel Gonçalves, P., Bacchelli, A., Çalikli, G. & Serebrenik, A. 2023. Competencies for Code Review. *Proc. ACM Hum.-Comput. Interact.*, 7, CSCW1. URL: <https://doi.org/10.5281/zenodo.7401313>. Accessed: 24 February 2024.

Wynekoop, J.L. & Walz, D.B. 2000. Investigating traits of top performing software developers. *Information Technology & People*, 13, 3, pp. 186–195. URL: <https://doi.org/10.1108/09593840010377626>.

Zahra, H., Noaen, M., Zowghi, D., Far, B.H. & Barker, K. 2018. Two Sides of the Same Coin: Software Developers' Perceptions of Task Switching and Task Interruption. URL: <https://arxiv.org/pdf/1805.05504.pdf>. Accessed: 31 December 2023.