ABSTRACT

In this study, the focus is on facilitating the API discovery process for developers and organizations. According to research articles, APIs are discussed in this research as the fundamental areas that are considered significant when developers or organizations explore an API. The areas include usability, documentation, stability, collaborative communities, and the popularity of an API. We created a design and creation research process and listed key aspects of each mentioned area above in a final artifact, where the idea is to allow people to look at the artifact and use it in a possible API exploration.

In conclusion, the artifact covers usability, documentation, stability, and collaborative communities. There are leading questions in the artifact to determine whether or not an API is suitable for a given purpose.

Keywords API, API characteristics, Lookup and Select, Pattern language methodology, Partner API, Popular API.
Acknowledgments

We would like to thank our research partner BeFa AB that has been ideal in their cooperation and our mentor Kristin Hane for the guidance you have given us. Hence, making this research possible. In addition, we would like to acknowledge the countless support from our families and friends. It has been wonderful to have all your support, and to be reminded to never give up when there's something you're passionate about together.
Table of Contents

1 INTRODUCTION .............................................................................................................. 5
  1.1 BACKGROUND ........................................................................................................ 5
  1.2 PROBLEM DEFINITION ...................................................................................... 8
  1.3 AIM ....................................................................................................................... 8
  1.4 RESEARCH OBJECTIVES ................................................................................. 8
  1.5 LIMITATIONS ....................................................................................................... 9
  1.6 OUTLINE ............................................................................................................... 9

2 LITERATURE REVIEW ............................................................................................... 10
  2.1 API TYPES ........................................................................................................ 10
    2.1.1 Public API ....................................................................................................... 10
    2.1.2 Private API ..................................................................................................... 10
    2.1.3 Partner API .................................................................................................... 10
  2.2 LOOKUP AND SELECT ..................................................................................... 11
  2.3 POPULARITY ....................................................................................................... 12
  2.4 CHARACTERISTICS .......................................................................................... 13

3 RESEARCH METHODOLOGY ...................................................................................... 15
  3.1 RESEARCH STRATEGY ...................................................................................... 15
  3.2 DATA COLLECTION ............................................................................................ 16
  3.3 SELECTION ......................................................................................................... 16
  3.4 METHOD OF ANALYSIS .................................................................................. 17
  3.5 ETHICAL CONSIDERATIONS ............................................................................ 18

4 RESULTS AND ANALYSIS ......................................................................................... 20
  4.1 DOCUMENTATION RESULTS ......................................................................... 20
  4.2 USABILITY RESULTS .......................................................................................... 20
  4.3 STABILITY RESULTS ........................................................................................ 21
  4.4 COLLABORATIVE COMMUNITY RESULTS ................................................. 21
  4.5 FINAL ARTIFACT RESULT ................................................................................ 22

5 DISCUSSION AND CONCLUSION ............................................................................ 24
  5.1 DOCUMENTATION CRITERIA ........................................................................... 24
  5.2 USABILITY CRITERIA ......................................................................................... 24
  5.3 STABILITY CRITERIA ........................................................................................ 25
  5.4 COLLABORATIVE COMMUNITY CRITERIA ................................................... 25
  5.5 A SOLUTION TO THE SELECTION TASK ..................................................... 26

6 METHOD CRITIQUE AND FUTURE STUDIES ......................................................... 27
  6.1 FUTURE STUDIES .............................................................................................. 28
  6.2 AUTHORS’ THOUGHTS ....................................................................................... 28
Abbreviations and definitions

term  definition

API    Application Programming Interface

SDLC  System Development Life Cycle

IS     Information System

UI     User interface

IDE    Integrated development environment
1 Introduction

This section contains a clear background as well as a problem definition, the aim, research questions, limitations, and the structure of this research.

1.1 Background

An Application Programming Interface (API) is a middleware that enables two applications to communicate. Additionally, one can allow a product and a service to interact with another application and synchronize data. Subsequently, there are numerous benefits associated with developers applying an API rather than producing a brand new piece of software whenever the need for an additional feature arises. The conundrum of what approach a developer should take on emerges. Another essential point is that when implementing a new API, the aim is to simplify the development process and reduce the frequent tasks intended to transmit information between computer components (Chen et al. 2017). Similarly, Hora and Valente (2015) state that adopting an API should increase productivity. While this is the case, the problem is that APIs change over time, and developers are interested in upgrading their applications to obtain assistance from the most recent APIs (Chikkala, 2017).

Furthermore, such tasks face numerous challenges associated with API popularity and API migration (Hora & Valente, 2015). Based on the findings of McDonnell, Ray, and Kim (2013), it is arguable that challenges related to API popularity are such that APIs rapidly evolve, causing numerous libraries to provide similar features (McDonnell et al., 2013). Consequently, developers have trouble deciding which API to employ. Equally important are the challenges developers face due to API migration, recognizing that APIs are backward-incompatible and that such factors contribute to client applications failing. Thus, the need to uncover an alternate API to prevent technical errors one could encounter when updating software (Hora & Valente, 2015).

Additionally, choosing an appropriate API is particularly important when making software. Consequently, if one selects the API and successfully integrates it into its technology, access is granted to several services. These services could be hardware platforms, operating systems, and other software modules on a computer. Although there is an assortment of APIs commonly applied on platforms, multiple API alternatives may be available that hold similar characteristics. Moreover, certain APIs are constructed with the intent of the design to be cross-platform-oriented software. By contrast, other APIs are only available in one particular programming language.

Given that there is a myriad of APIs to select from, developers have formed unique strategies to uncover APIs that hold essential requirements. Consequently, based on the findings of Chen et al. (2011), knowledge gathered by the research community is subtle, implicit, and not well understood. This is when attempting to establish a set of criteria for selecting and-/or switching an API.
Even though studies on popular APIs and what type of characteristics they hold versus what Hora & Valente (2015) refer to as "ordinary APIs" exist. Research on how developers select APIs with specific prerequisites is limited.

However, Mileva, Dallmeier, & Zeller (2010) explore this subject to some degree by mining hundreds of open-source projects, including their external dependencies in their research. As a result, analysis of the obtained data against APIs is performed based on popularity to alert developers when to consider switching to a different API. Mileva et al. (2010) state that there is no suitable way to assess the quality and success of an API. They, therefore, consider an API's popularity to be an indicator of its success and that a lack of popularity or a decline in popularity indicates a lack of success. To leverage wisdom in this matter, they mine a large number of projects. Benefits of API popularity data and usage recommendations are dealt with, as well as an approach to finding API usage trends is presented. The approach in this scenario is a prototype tool that has the ability to offer usage recommendations. The popularity analysis is straightforward according to Mileva et al. (2010) with the clarification that they estimate how many times each month a single API element is utilized across the 200 open-source projects they mine. And is the basis of their conclusion which is that they create a tool prototype that analyzes the collected data and illustrates the usage trends of API pieces. This information might be quite useful for API manufacturers in identifying the weak points in their products. They can also make useful suggestions to API users based on prior usage statistics. The broad and diversified range of projects they assess, as well as their positive evaluation outcomes, ensure that the recommendations provided by their tool prototype are valid. This leads to the final conclusion, that API element usage trends are one technique for demonstrating API users’ historical choices and predicting their future.

Hora & Valente (2015) also look into API Popularity but with a different stance. They argue that two challenges developers meet are due to APIs evolving and therefore changing. The side effect in correlation to such is the requirement to update the applications resulting in challenges involving API Popularity, and API migration. To tackle those obstacles, they recommend the apiwave tool. The tool is intended to monitor API popularity and migration of major open-source frameworks/libraries to assist API clients and developers, and is proposed for filling the gap between literature and real-world tools in the context of API evolution. The aim expressed is to understand how APIs develop over time and make such large volumes of data easily available with the focus on two aspects: API popularity and migration.

Outcomes that eventually are proposed in regards to handling API migration are either with the support of modified IDEs or by mining source code. The conclusion on the other aspect, API popularity is inspired by the source mentioned in the paragraph above, which is related to the research conducted by Mileva et al. (2010). However, Hora & Valente (2015) fail to express how the aim to fill the gap between literature and real-world tools is achieved even though they present the output of the apiwave tool and why it could be useful in the real world.
What the research conducted by Mileva et al. (2010) partly lacks is a universal approach applicable to various projects and not limited to Java projects only.

In addition, the prototype tool utilized while conducting the study is not provided and therefore potential users are left with only information and no help in the matter of being recommended an API to use.

Hora & Valente (2015) do on the other hand not fail to present a universal approach on the subject of finding an API, the apiwave tool presented is expressed as a tool that only keeps track of API popularity and migration, which does in fact only manage GitHub Java projects. In other words, the apiwave tool is significantly limited and therefore cannot be used as a universal approach in the overall software development field, which means that the gap between literature and real-world tools isn’t fully met.

While having valid points that one should take into consideration during the selection of an API, the issue is that an appropriate approach is not really expressed, and the limitations outwin the advantages. However, the findings of these studies are valuable when producing an artifact that does have an overall approach to finding an API when working with additional resources.

A substantial amount of research has been undertaken on the subject of APIs, which also has resulted in several challenges. Specifically, Bondel et al. (2021) mention concern about the user perspective of APIs and subsequently goes even further by narrowing down the topic to branch it out into the categories of public, partner, and group APIs, which are further explained under the section 2.1 below. With intent to provide more in-depth research on a matter in the field is portrayed as, on the whole, comprised of the lack of literature on management literature with the design perspective as objective. While the essence of the research is no different, Doucette (2008) goes into further detail concerning qualities developers should bear in mind when developing a proper API that correlates with the aspect of usability. In addition, the researcher mentions how up-to-date literature provides general guidelines one should adhere to when developing an API. Aiming to answer the question, what makes an API usable?

The relationship between the attribute usability and APIs is that evolves, updates, and changes become empathized. Consequently, the user cannot operate the API or at least impact performance in an ineffective manner (Chen et al., 2011; Hora & Valente, 2015; Sturm et al., 2017); both cause immense inconvenience for developers, and the quest for an API is forced in the event of the API becoming non-operative. As a result, many studies concentrate on selecting an API conducted with the aims and findings from several viewpoints.
1.2 Problem Definition

The search for a suitable API to be utilized by developers in an organization is a task that does not have a clear approach when exploring the myriad of analogous APIs that exists in addition to what matters one should take into account while browsing for such. This is an issue that is not only time-consuming and costly given that there is no universal routine to the task of finding an API the impromptu process often employed by development teams results in no API being selected.

1.3 Aim

This research aims to produce an artifact that companies and their developers can use as a guide to find suitable partner APIs by asking questions deemed highly appropriate for such tasks, to simplify the selection process and make the task of selecting an API more efficient.

This research will explore a combination of research and analyses conducted prior. The challenge associated with API implementation in an organization is initially to classify essential elements while the developer is concurrently on the lookup phase defined by Marchionini (2006). Chikkala (2017) describes this phase as the selection process for an API that ought to align with client requirements.

The characteristics explored of popular APIs in this research address the challenges corporations meet when deciding on a suitable API. The attributes of usability, documentation, community, changeability, and stability concerning APIs will be central throughout the investigation. The rationale behind such is that these characteristics are deemed to be some of the most vital ones in this matter (Chen et al., 2011; Doucette, 2008; Hora & Lima, 2020; Meng, Steinhardt, and Schubert, 2019). One may reason that these elements determine if developers find an appropriate API or not.

The emphasis of this research is to establish and define an artifact intended for companies and developers. The aim in relation to the artifact is to hold the characteristics that are found to be the most important ones to bear in mind while in the process of finding a suitable API. In addition, the characteristics will hold questions linked to the search for an API. The questions will be fundamental and separated into a specific characteristic.

1.4 Research Objectives

A series of questions derived from the lack of research on the topic described in the problem statement. As a result, the aspiration is to answer the following questions which also will be essential to the design and creation process of the artifact.

Main question: What are the main characteristics of popular partner APIs?

Subquestion 1: What are the most common impediments when seeking an API?

Subquestion 2: What approach ought to be adopted to find an API?
1.5 Limitations

This research will concentrate on producing an optimal approach in the quest for partner APIs that assess the following established elements of an API:

✓ Documentation
✓ Stability
✓ Usability
✓ Collaborative community

The evaluation is limited to previous studies and investigations and is exploratory since new data about this topic emerges continuously. Additional limitations include the stages of developers’ looking for and selecting a mainstream partner API in the development community, consequently excluding the integration phase.

This research is limited to Partner APIs since the subjects of this research are companies and APIs primary.

1.6 Outline

This chapter starts with the background where the current situation is portrayed thereafter API types are described followed with the problem definition, purpose and goal, research objectives, aim and limitations described.

The literature review chapter bring up the topics partner API, lookup and select, popularity and characteristics of APIs in relation to previous research conducted on such topics.

The methodological chapter presents the research process and its procedures.

This chapter relates to the findings on the API characteristics, documentation, usability, stability, collaborative community and lastly API popularity.

This last chapter provides critique regarding this research and the authors own speculations and thoughts.

Figure 1: (Created by the authors, 2022)
2 Literature Review

This section provides an overview of the characteristics of a public API, a private API, and a partner API to provide distinction. Moreover, this section compares different approaches to API exploration, what determines the popularity of an API, and finally a review of an API's properties.

2.1 API Types

2.1.1 Public API

A public API, referred to as open API, is an application programming interface publicly accessible to software developers. Public APIs are available on the Internet and shared freely, allowing a network that the owner can access to give users universal access. Concerning Public APIs, they will also be constructed in many ways depending on when employed. Still, the main objective of every public API is to be used and accessible to as many clients as possible (Bondel, Matthes, and Landgraf 2021).

2.1.2 Private API

A private API allows developers within an organization to access portions of their back-end data and application features. Emerging applications that derive from developers are, if approved, made public by the assumption that the API interface is not available to everyone unless they interact directly with the API manufacturer (Bondel et al., 2021).

2.1.3 Partner API

Partner APIs that allow business-to-business communication are restricted. That aside, a selected group of external developers or API consumers are permitted accessibility with the understanding that they possess required certificates and permits. As a result, partner APIs comprise stronger authentication, authorization, and security mechanisms. (Bondel et al., 2021).

While there has been significant research about partner APIs, few researchers have considered pattern language. According to Bondel et al. (2021), the primary focus area, the technical aspects of APIs concerning API management of public, partner, and open APIs on an organizational level, is not a matter most view. However, instead, they are concerned about the API consumer.

The concentration on technical aspects concerning API management does mean a lack of the non-technical aspect this research will supply. The key findings relevant to this research are how Bondel et al. (2021) work with the Design and Creation methodology and link the pattern language. The pattern language provides an overall view of API management, including API evolution, API representation, and API quality, which further helps create an artifact. Furthermore, characteristics of a partner API that are relevant to the developer when selecting such gives us an insight relevant to the process of defining an artifact.
2.2 Lookup and Select

Choosing an API or switching between APIs in software engineering is, according to Chen et al. (2011), more than a programming practice. The researchers conducted a qualitative study to target people who have experienced choosing/switching APIs. A categorization was made regarding what API users consider when choosing an API into three categories – Looking, Doing, and Planning. The looking process involves discovering APIs. Doing involves learning and testing APIs before committing. The planning process considers the effects of a commitment if an API is chosen.

The ‘looking’ process for APIs was divided into different key aspects for further investigation on findability, documentation, and whether any active communities are associated with each API. As for the ‘doing’ process, which involves experimenting with the API, the key factors were learnability, environment setup, examples of what it offers, dummy programs, and level of abstraction. Commitment to the API is a vital part of the planning process, which entails developers updating APIs, fixing bugs, and being active in API communities.

The planning process also included support for multiple environments and openness to change so that API users could offer recommendations. Furthermore, Chen et al. (2011) conducted an exploration of APIs with a considerable influence on the previously mentioned three categories. This, since they cover how to find an API, how to manage the API found, and how to test the API found. When comparing a study written by Chikkala (2017), the categories looking, doing, and planning are mentioned similarly as in the research conducted by Chen et al. (2011).

Further research revealed that API exploration occurs through API marketplaces, informal search on the internet, and component identification using prior experience, e.g., whether a team member has explored APIs before. The process of evaluating an API follows the process of finding an API. Based on the ideas of Chikkala (2017), the study results argue that evaluation methods are decision support systems, simulation, method engineering, strategic contracting, and procurement. Lastly, the project team usually does the step in exploring APIs; no straightforward approach is described after locating and evaluating an API since these criteria are defined as crucial when finding an API.

Based on a study conducted by Hora & Lima (2020), the difficulties related to finding an API are attributed to the plight of distinguishing between popular APIs and ordinary APIs. Their study aimed to clarify the differences between familiar APIs and popular APIs to facilitate further research on API exploration and API development.

Results showed that the popular APIs had more code comments than the regular ones, but this can, according to Hora & Lima (2020), also result in them becoming too large and complex. Furthermore, the results showed that the most popular APIs have more changes and are more unstable than the regular APIs, but they may not necessarily have more contributors. In addition, they found that the popular APIs are often used together with other APIs and packages, but the difference is exceedingly insignificant compared to the ordinary APIs.
This study does not provide an example of a popular API or a standard API. Having this information is necessary to make a broader analysis of the difference between these two. As well as to justify the inclusion of the difference between popular APIs and the ones that are viewed as ordinary in the final artifact.

According to Yu et al. (2016), software libraries are complex, and traditional search methods do not focus on looking for APIs, which makes the process of exploring APIs inconvenient and time-consuming. To facilitate the exploration of an API, Yu et al. (2016) defined a search-based recommendation algorithm based on API methods to enable API exploration. The algorithm can recommend appropriate API methods based on user input written in natural language. It takes only 0.7 seconds for the algorithm created by Yu et al. (2016) to find relevant APIs based on API descriptions, as illustrated in their study. This method is used to find appropriate APIs based on user input written in natural language. However, it does not cover areas such as documentation, stability, usability, and community collaboration, which are the focus areas of the final artifact.

Nevertheless, (Hora & Valente, 2015) makes a valid point when migrating an API. Since APIs constantly evolve and are likely to change and a side effect is that developers need to upgrade their applications, often the reasons to benefit from newer or improved APIs. However, these activities involve challenges because of API popularity as well as migration.

2.3 Popularity

In the journal article by Hora & Lima (2020),” What are the characteristics of popular APIs? A large-scale study on Java, Android, and 165 libraries. ” The emphasis is on the characteristics of popular APIs. Moving on to the research,” how do developers choose APIs?” Although APIs are a common subject, Chen et al. (2011) present a comprehensive insight into how developers select an API.

Chikkala (2017) is one of the few researchers that have concentrated on selecting an API and highlights the lack of such studies. Only focus on the developers’ viewpoint, the human behavior associated with the selection process, and what criteria developers currently consider when selecting an API. Consequently, Chikkala (2017) does not provide a methodology for developers but instead emphasizes measures useful for academics working to establish an artifact demonstrating how to select an API. Thus, studies have touched on what characteristics an API holds and its significance; there are gaps in the area. Moreover, the focus on partner APIs linked to exploring available solutions that a developer can incorporate into its software system has not been identified.

Even though external libraries are an essential part of a project's source code, Mileva et al. (2010) state that no direct feedback methods exist to display an API's popularity. Their analysis focused on each API's ‘weak spots,’ i.e., the elements that are unpopular or have declined in popularity. Mileva et al. (2010) conclude that analyzing a library’s weak spots can benefit its users since software developers who use external libraries can benefit from knowing which are the ‘weak spots in the library’.
When there is a decrease in the use of an API element, it is an indication that most users prefer not to use it. It is also useful for library developers to analyze weaknesses. Mileva et al. (2010) write that they must understand how the end-user will use their APIs to develop a better product. Once they have identified the ‘weak spots’ in their API, they can analyze why it is popular. As with the product results, such data can help API manufacturers to improve their products overall based on consumer preferences.

Mileva et al. (2010) define popularity by computing the number of times per month a specific API element is used throughout the two hundred open-source projects they have mined. They argue that API elements usage trends are a means for displaying the preferences of the API users from the past and anticipating its future. The objective is to give recommendations on the rising popularity of a particular API element and be able to provide reasons behind such a suggestion. This research is necessary because popular APIs and their characteristics relate to each other (Hora & Lima, 2020). The gap, however, regarding API popularity that no one has seemed to fill is how developers manually, by selection criteria, go on choosing an API that holds acceptable characteristics and is widespread.

2.4 Characteristics

The characteristics an API holds are crucial when looking into such, and nonetheless the fact that APIs are consistently evolving. This makes it more important since a developer and their team need to know when the time has come for a change of API (Chikkala, 2017; McDonnell et al., 2013; Mileva et al., 2010). Meaning that characteristics of an API do not become outdated once an organization settles for an API. While using qualitative content analysis in the empirical study conducted by Chikkala (2017) offers valuable insights on the topic of characteristics linked to APIs. The developer perspective is crucial in this area; the research states that prior research does not consider the industry practices. While this is an essential point, the fact is that there is value in such studies as well. Hence, there is a need to improve the practices of selecting an API based on characteristics by taking relevant research and evaluating relevant factors and not excluding one because of the other.

Hora & Lima (2020) research characteristics of popular APIs based on the number of client systems, and the association does not imply causation. However, both (Hora & Lima, 2020, p.12) and (Chen et al., 2011, p.4) as well as (Chikkala 2017, p.10) do mention characteristics concerning APIs but from different points of view and research approaches; all necessary and relevant when designing and creating an artifact for developers.

The findings on the characteristic documentation by these researchers all underline the importance of a well-documented API to provide aid for developers. The documentation quality needs to be considered within the frame of important characteristics an API holds. (Chen et al., 2011; Chikkala, 2017; Hora & Lima, 2020).
Equally important is the characteristic stability; developers selecting an API look into how stable an API is and only consider it if perceived as stable. Subsequently, the level of stability has a role in ‘How popular an API is in the developer field’ (Chen et al., 2011, p.5; Chikkala, 2017). Both support the characteristic referred to as usability, the importance of this quality is described under the heading Literature Review found above. To elaborate, developers cannot find an API usable if no documentation is found and it is not stable even though the characteristics of stability and usability are not in a direct link to each other.
3 Research Methodology

This section begins with an account of the choice of method and further presents the research’s data collection, selection, and ethical considerations. Finally, the method analysis for this thesis is to be delivered.

3.1 Research Strategy

For the aim to be achieved outlined in section 1.4 above, this research access to information from a large set of research articles related to API characteristics and components is needed. The strategy used is based on a design and creation research method as the artifact focuses on streamlining API implementation where the emphasis is on developers and organizations.

While the objective could have been approached by conducting a case study, the decision to conduct a design and creation research strategy was ultimately the preferred strategy for producing an artifact. This is due to case studies being an approach that does not produce such (Oates, 2006, p.20).

The design and creation research method describes many opportunities for new ways to use the technology. This methodology is applied when developing mobile computing, tools for remote studies, strategies for developing web-based systems, new user interfaces, new search tools, and new digital art forms. Based on this description of this method, it can lead to an artifact focusing on either instrument for remote learning or search tools that facilitates the implementation and findings of APIs.

This research method produces and applies knowledge to create practical artifacts. The objective was to guide the artifact user through a well-instructed path. The development methodology in this research involves the waterfall model and can be characterized as the research process system development lifecycle (Oates, 2006). This is because all analysis is completed before moving on to the design phase, which in the SDLC model is phase two (Oates, 2006, p.39, p.113). Lastly, the research by nature ended up being explorative, because the subject is relatively new and continuously evolving.

The evaluation criteria to perceive the results as successful was based on the following criteria:

- Functionality
- Accuracy
- Reliability
- Fit with organization

With the characteristics suitable to the research objectives in section 1.4.
3.2 Data collection

The data collected in this research was extracted from several sources, including scientific articles, based on information concerning application programming interfaces. The search engines adapted for collecting relevant data were Google Scholar and Diva-portal. Google Scholar specializes in scholarly literature and academic resources. Diva-portal holds research publications and student theses from fifty different universities. Moreover, when searching for relevant articles following keywords: API, API characteristics, API types, API uses, API implementation, and API maintenance were employed.

Additionally, to collect essential knowledge in preparation for shaping an artifact that will guide developers through the API research, this data collection methodology was built on the internet and documents. This is the method used for generating data to enhance the validity of this research where methodology triangulation is employed, which involves using more than one kind of method to enhance the validity and credibility of a study's findings. This research data collection methodology was built on Oates (2006) description of how to conduct a document and the internet.

3.3 Selection

The selection in this research derived from partner APIs where the focus was on companies and developers within an organization that requires an application programming interface. The developers and organizations were selected at random, with the only requirement of having practiced any API exploration method. These actors will have the greatest need for an alternative approach when applying for a new API. In this case, the manufactured artifact aims to guide the user to an alternative API for the organization. (Bondel et al., 2021)

Picking this study concerning organizations and developers within organizations is partially grounded in a study written by Myers & Stylos (2016). The central theme refers to a significant problem with APIs. The problem portrayed is linked to APIs and how developers use them at various levels. They spend an exceptionally long time learning new APIs, and APIs are frequently misused, resulting in bugs and sometimes security issues. The frequent problem is that APIs must supply the needed functionality for one’s organization, but even if they do, the design of the API could make them unusable.

According to Myers & Stylos (2016), creating an API requires making hundreds of design decisions at several levels, affecting the usability. Usability measure how simple an API is to learn, how effective developers are with it, how well it prevents errors, how simple it is to use, how dependable it is, and how well it matches the mental models of its users.

Furthermore, this is a critical part of why selecting the correct API from the beginning is essential. So that companies and developers within the organization avoid wasting time on an API that may have poor documentation. Moreover, design can affect the use of the specific API.
The search for information about the best possible method for maintaining APIs would also have been interesting to include in the selection. However, since there is not enough time, this was not investigated.

Given the lack of research in this area, numerous studies and literature were selected relevant to API exploration. All relevant information about API use, API exploration, and the properties of APIs was assembled to expand knowledge in API exploration.

3.4 Method of analysis

The method analysis was conducted throughout the SDLC methodology described in section 3.1 above, where the artifact was produced after a detailed analysis was accomplished. The artifact that was developed through the design and creation research strategy would, at its final stage, be a model, which Oates (2006) defines as something that combines constructs to represent a situation. This model is used to aid in the subject of problem understanding as well as solution development. The process includes the five steps: awareness, suggestion, development, evaluation, and conclusion.

The awareness step is like the other steps intertwined in the research iteratively and answered in different sections. Awareness, which is defined by Oates (2006) as the phase where recognition and articulation of a problem are conducted, is seen in the background sections 1.1 above and 1.2. Moving on to the suggestion step, it is essential to offer a tentative idea of how the problem might be addressed. This can be found in chapter 6 below. The development step is conducted and displayed in section 4 below, and includes the tentative design idea and its implementation. The fifth step, evaluation, is also conducted in chapter 4, and holds the examination of the final artifact, which also contains an assessment of the final artifact’s worth. In the last step, the conclusion is conducted and provided in section 4 below, where the results from the design process are consolidated and written. The identification of knowledge gained together with any loose ends that might exist in addition to unexpected results or anomalous results that cannot be explained yet and might be the topic of further research (Oates, 2006).

The reasoning for conducting a design and creation research process directly connects to the problem definition. It has to do with new knowledge emerging for developers and the research community to make use of. Thus, the level of this research will not conduct the implementation of the final artifact where testing is performed due to this research being a bachelor's degree thesis.
For the artifact to be considered successful, there were several criteria that must be met. These are outlined in section 3.1 above. The objective is “proof of concept” only and, therefore, not evaluated if the artifact, in the end, works in a real-life context (Oates, 2006). The purpose of the research has more to do with the design solution includes certain properties to behave in a particular way. Which, in this case, is to help developers decide what API to select when searching.

Lastly, the approach selected for this research is an IT research approach, which is defined as the artifact to be a tool to make people more efficient or effective.

Using the Design and Creation methodology, the production of an artifact is the additional objective with a pattern language methodology where API management activities such as a focus on API providers where collaboration with other stakeholders occurs. All to support established organizations to initiate and manage public, partner, or group API initiatives. Each pattern of the pattern language addresses a more specific problem and the corresponding solution within the pattern language framework. Concerning the purpose of a pattern language, Bondel et al. (2021) argue that it distributes knowledge in respective domains and offers academics knowledge about evolution (Bondel et al., 2021, p. 3).

The practical approaches that form the basis for the creation of the artifact following the Design and Creation methodology, are based on previous studies and documents of API’s, found on the internet. These methods were chosen based on Oates (2006), where the use of multiple data collection methods increases validity through method triangulation described in section 3.2. Through a review of previous documentation and the use of the internet to find previous research on APIs, a broader comparison could be drawn in the areas of usability, documentation, collaborative communities, and stability, which are described and compared in section 2.

By using the method of triangulation, and also the comparison between the sources regarding API properties outlined in section 2, this study was able to gain a deeper insight into each API area. This was captured in four separate tables containing the most important aspects for each API area, which are presented in section 4. With these tables, an artifact was able to be created that covered each topic of this study’s research, both using the named methods above to create the tables, but also the aspects within each table.

3.5 Ethical considerations

Because this is a design and creation research that aims to produce artifacts, corporations and organizations can enforce in their work that the ethical aspects of the ease of access and copying applied to honest writing will be used. To prevent any wrongdoing referencing for the data found and used in research is conducted with the APA referencing method. Equally important, Oates (2006) mentions that the Internet causes more issues for an ethical researcher. Data retrieved from the Internet requires that information be incorporated in this research: whether the people are study subjects or authors of works.
Furthermore, Oates (2006) suggests using an ethical evaluation guide that states questions one should consider when deciding whether to use a source of interest as an instrument to check that a research report of interest is ethical.

In addition, this research follows The European Code of Conduct for Research Integrity (2019) expresses “Good Research Practices” with the following context:

✓ Research Environment
✓ Training, Supervision, and Mentoring
✓ Research Procedures
✓ Safeguards
✓ Data Practices and Management
✓ Collaborative Working
✓ Publication and Dissemination
✓ Reviewing, Evaluating, and Edit

Concerning the list paragraph above, among many ethical standpoints, the following but not limited to, are considered. The research results and interpretation were published open, honest, transparent, and accurate manner. Additionally, the research complies with codes and regulations relevant to the research discipline. The ethical aspect of data practice and management follows transparency concerning how to gain access to or use regarding the data and research materials conducted throughout the research. All collaborators in the research take accountability for the integrity of the research.

All authors are entirely responsible for the content of a publication unless otherwise indicated. Dedication to the research community is demonstrated by refereeing, reviewing, and evaluating. Revision and evaluation of submissions for publication, funding, appointment, promotion, or reward transparently and justifiably. (The European Code of Conduct for Research Integrity, 2019).

The way the research practices are accomplished is by the authors having support regarding training, supervision, and mentoring from the research department. This also provides a research environment by delivering information from libraries as well as a portal with published peer-reviewed work. The research procedures are followed by approved research methodologies known in the research community.

A dilemma encountered was about the data practice and management topic where the research conducted by Chen et al. (2011) did not supply the year of publication. However, this was solved by contacting the authors via email, where they provided approval regarding using their paper and what year the study had been conducted.
4 Results and analysis

This section describes relevant topics that have emerged during the research process and discuss patterns as well as connections discovered. The first four subparts represent topics in relation to a characteristic mentioned by one or more authors in the research collected during the data collection process. Lastly, in section five, a further analyzed discussion and conclusion will be investigated and portrayed.

4.1 Documentation Results

Table 1 shows important aspects for API documentation to consider when exploring an API. Those aspects of documentation include understandability, learning resources, finding artifacts, and completeness. This table outlines the most important criteria for determining the quality of an API's documentation, as found in the studies written by Chikkala, 2017, Chen et al., 2011, and Hora & Lima 2020. These criteria for documentation are explained in more detail under section 5.1.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understandability</td>
<td>(Chikkala, 2017), (Chen et al., 2011), (Hora &amp; Lima, 2020)</td>
</tr>
<tr>
<td>Learning resources</td>
<td>(Chikkala, 2017), (Hora &amp; Lima, 2020)</td>
</tr>
<tr>
<td>Finding artifacts</td>
<td>(Chikkala, 2017), (Chen et al., 2011), (Hora &amp; Lima, 2020)</td>
</tr>
<tr>
<td>Completeness</td>
<td>(Chikkala, 2017), (Chen et al., 2011), (Hora &amp; Lima, 2020)</td>
</tr>
</tbody>
</table>

*Table 1: Documentation characteristics*

4.2 Usability Results

Table 2: Usability characteristics on page 21 displays the main factors to consider regarding the usability of an API, where the data collection showed that the most important factors were intuitiveness, appeal, complexity, freshness, and navigation. The purpose of this table is to determine the usability of an API, based upon the mentioned criteria in the table for a potential measurement of API usability. The criteria for usability are explained in more detail under section 5.2.
Table 2: Usability characteristics

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intuitiveness</td>
<td>(Chen et al., 2011), (Doucette, 2008), (Chikkala, 2017), (Myers &amp; Stylos, 2016)</td>
</tr>
<tr>
<td>Appeal</td>
<td>(Chikkala, 2017), (Myers &amp; Stylos, 2016)</td>
</tr>
<tr>
<td>Complexity</td>
<td>(Doucette, 2008), (Chikkala, 2017), (Yu et al., 2016), (Myers &amp; Stylos, 2016)</td>
</tr>
<tr>
<td>Freshness</td>
<td>(Doucette, 2008), (Hora &amp; Lima, 2020), (Chikkala, 2017)</td>
</tr>
<tr>
<td>Navigation</td>
<td>(Chen et al., 2011), (Doucette, 2008), (Chikkala, 2017), (Myers &amp; Stylos, 2016)</td>
</tr>
</tbody>
</table>

Table 3: Stability characteristics

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customization</td>
<td>(Chikkala, 2017) (Hora &amp; Lima, 2020) (McDonnell et al., 2013)</td>
</tr>
<tr>
<td>Interoperability</td>
<td>(Chen et al., 2011) (Hora &amp; Lima, 2020) (Hora &amp; Valente, 2015)</td>
</tr>
<tr>
<td>Developer control</td>
<td>(Chen et al., 2011) (Chikkala, 2017)</td>
</tr>
</tbody>
</table>

4.3 Stability Results

Table 3 below shows the main criteria for evaluating the stability of an API, showing that customization, interoperability, and developer control are considered to be important characteristics. This table was created to emphasize which areas to examine when looking at the stability of an API, as these three criteria shown in the table were considered as the basis for the stability of an API. The criteria for stability are explained in more detail under section 5.3.

4.4 Collaborative Community Results

The last table, Table 4: Collaborative Community characteristics displayed on page 22 displays the results from the research regarding collaborative communities were arranged, resulting in online support, size and popularity, and rate of activity as important criteria. The purpose of these criteria is to investigate whether APIs found during API exploration have a collaborative community, to partly measure its popularity. The criteria regarding collaborative communities are explained in more detail under section 5.4.
Criteria | Authors |
--- | --- |
Online Support | (Chen et al., 2011), (Chikkala, 2017), (Hora & Lima, 2020) |
Size and popularity | (Chen et al., 2011), (Chikkala, 2017), (Hora & Lima, 2020) |
Rate of activity | (Chen et al., 2011), (Chikkala, 2017), (Hora & Lima, 2020) |

Table 4: Collaborative Community characteristics

4.5 Final artifact result

The final result is based on the tables presented in sections 4.1, 4.2, 4.3, and 4.4, where each criteria is based on previous research regarding important aspects to consider during API exploration. The purpose of this artifact is to support the API exploration process within API usability, stability, documentation, and collaborative communities. The idea behind the artifact is to provide concrete guidance options for each area presented, where during the API exploration one can rely on the questions presented in Figure 3, which are based on previous API exploration research that covers key aspects for each area presented in this research.

The recommended approach when using this artifact is to ensure that the majority of the questions presented in Figure 3, can be answered with certainty that the information retrieved is correct. Moreover, the user should keep in mind that the content of this artifact is restricted to documentation, usability, stability, and collaborative communities, and that the user may determine on its own what is most essential for the found API.
Figure 3: Final artifact with questions included
5 Discussion and conclusion

This section discusses criteria and results, as well as a discussion of a possible approach for the API selection process.

5.1 Documentation criteria

When looking into API characteristics, documentation is essential to include in API development, as the documentation contains instructions on how to use the API. A part derived from the documentation criteria is understandability, which implies that documentation for APIs must be understandable, concise, and include information that does not mislead the user which is detailedly described in section 5.1 above. The documentation also provides an additional criteria defined as learning resources, which are relevant to tutorials and online resources, and the importance of having source code available for problem-solving using the API, as mentioned further described in section 2.4. When locating APIs, documentation should be intuitively organized so that the information you need is easy to identify as quickly and efficiently as possible. Completeness is also a factor if the information on an API’s topic is detailed and not misleading in any direction (Chen et al., 2011; Chikkala, 2017; Hora & Lima, 2020).

Based on the results of the composite of research explored, the topics the majority bring up when talking about documentation and questions that arise are presented in Figure 3, which contains clear and concise questions to evaluate API documentation.

5.2 Usability criteria

An essential aspect of the API is its design, which includes the criteria that the architecture of the API is not too misleading in any way and is deemed usable. Another criteria within the usability topic are the intuitiveness criteria, which also is an essential aspect of API usability. This implies the significance of having an exemplary user interface, which as a result allows the user to perform the desired actions efficiently and effectively without causing unnecessary distractions (Chen et al., 2011; Doucette, 2008; Chikkala, 2017; Myers & Stylos, 2016). Among the areas of appeal within usability includes the importance of good visual features and appealing visuals (Chikkala, 2017; Myers & Stylos, 2016).

The usability criteria also contain topic complexity, which involves information overload considering the steps required to complete a task or ease of use. A factor related to usability is that one should avoid to provide excessive information about a particular topic which directly influences how long it takes to solve a problem (Doucette, 2008; Chikkala, 2017).
The sub-part of the usability criteria freshness, is also linked to the question if the UI is following the latest trends or if they still are based on old and outdated designs. (Doucette, 2008; Chikkala, 2017; Hora & Lima, 2020).

Lastly, usability considers the aspect of user navigation, which involves a user’s ability to navigate (or explore) the software and find the information they are looking for. (Chen et al., 2011; Doucette, 2008; Chikkala, 2017; Myers & Stylos, 2016)

Just like the documentation section above, the basis of the final questions one should ask themselves when selecting an API is based on the results of the multiple types of research explored. The topics the majority bring up when talking about usability and questions that arise are presented in Figure 3, containing clear and concise questions aimed at assessing API usability.

5.3 Stability criteria

An API's stability is an essential factor that includes customization, interoperability, and developer control. These areas include the ability of an API to adapt according to user needs, which is an essential factor regarding customization in an API’s stability (Chikkala, 2017; Hora & Lima, 2020; McDonnell et al., 2013). The API's interoperability allows the API to be modified to work with other APIs (Chen et al., 2011; Hora & Lima, 2020; Hora & Valente, 2015). Lastly, the stability of an API also includes developer control, which is how much structure the software imposes on users, or how much freedom is given to developers (Chen et al., 2011; Chikkala, 2017).

The final questions regarding the criteria defined as API stability are based on the same development process that has been applied during the development of questions in the areas of documentation and stability, presented under sections 5.1, and 5.2. Based on the results of the multiple types of research explored, the topics the majority bring up when talking about API stability and the questions that arise can be seen in Figure 3, containing clear and concise questions for evaluating API stability.

5.4 Collaborative community criteria

The collaborative community criteria include partially online support for each API, such as blogs or websites for developers in terms of collaborative communities. Collaboration communities also consider the size and popularity and level of activity of each community. (Chen et al., 2011; Chikkala 2017; Hora & Lima, 2020)

Lastly, the questions for the collaborative community part are developed in the same manner as the API areas listed above under sections 5.1, 5.2, and 5.3, which are based on the results of the multiple types of research explored. The topics the majority bring up when talking about collaborative community and questions that arise are outlined in Figure 3, containing clear and concise questions for assessment of an API's collaborative community.
5.5 A solution to the selection task

A possible solution to start the selection process of an API is to look into how popular the API is. This is an excellent first step due to a high amount of usage of an API most likely embraces that the API is something favorable in the developer community. This can be confirmed by looking at sections 2.3 and 2.4 above, which mention two possible solutions.

One can be that the developer looks in too how many times an API is called. The second solution is based on the number of client systems to get an idea of how popular an API is. Therefore excludes an API that does not fall within such boundaries and then look at the characteristic features mentioned in Figure 3, above.

The main research objective defined in section 1.4 above is partially answered in section 2.4 above and is fully answered in section 4 above where each characteristic is laid out and described in detail. The main characteristics are as a result the following, Documentation Results, Usability Results, Stability Results and lastly Collaborative Community Results. We, therefore, deem to have answered the main question successfully.

The following sub-question 1 found in the same section as the main question defined as “What are the most common impediments when seeking an API?” is answered in the sections 1.1 and 2.2 above where the obstacles mentioned in section 1.1 are described vaguely to be related to API popularity and the fact that they rapidly evolve. In addition that there are numerous libraries that provide similar features. A more detailed description and the answer is found in section 2.2 Lookup and Select, where these obstacles are further described and other parts of the seeking task in relation to such as well. Some of the obstacles defined are in relation to the difficulty of distinguishing between popular APIs and ordinary APIs. Another obstacle has to do with traditional search methods not focusing on the looking phase for APIs, which makes the process of exploring APIs inconvenient and time-consuming to name a few.

The last sub question "What approach ought to be adopted to find an API?", can be answered by referring to the final artifact. The artifact is the result of comparing API areas, API exploration approaches, and analyzing what the most common barriers for previous approaches to API exploration are. In an API exploration for areas of documentation, usability, stability, and collaborative community, it is recommended that one follows the questions presented in Figure 3, for each area, and confirm they can be answered. Lastly, when choosing this approach, it's recommended to start from sections 5.1, 5.2, 5.3, and 5.4, where the questions are clearly described to gain clarification for the questions presented in Figure 3.

When it comes to the aim found in section 1.3 above, we deem it to be successfully answered concerning organization previously having problems classifying API elements when implementing and searching for APIs. We completed research based on research articles and analyses that resulted in clear elements for the areas of usability, documentation, stability, and collaborative community regarding API properties which are demonstrated in sections 5.1, 5.2, 5.3, and 5.4.
6 Method critique and future studies

In this section, a methodological critique of the choice of method is discussed, as well as the recommended starting point for further research, and lastly, the author's thoughts regarding this research.

When conducting research and analysis on previous API exploration, this research produces an artifact supported by research articles to facilitate the search for developers regarding APIs. As a result, the selected researches were solely based on previous research on API properties.

For data collection on APIs, documents and the internet were chosen. This was determined partly to explore different types of APIs, API properties, and existing problems around API exploration. Additionally, to compare the literature found with each other to get an artifact with as high accuracy as possible. The alternative data collection methods in a design and creation research were excluded. This due to lack of time and limited knowledge about APIs, which necessitated a deeper understanding of API areas to strengthen the results.

To make the artifact as dependable as possible, credible sources were chosen and evaluated on the Scopus portal, where authors can see the number of times research has been cited, underlying the level of credibility a study has. The credibility was assessed by comparing the sources around API areas to find similar explanations that support API properties’ which were also measured by examining the number of mentions of each source, where the number of mentions had an impact on the decision of whether the source was credible used in this research are public. They serve to describe the characteristics of an API and demonstrate why it is important to examine an API’s documentation, usability, stability, and collaborative community. This research includes sources related to API exploration and API properties for developers and organizations.

The accuracy of the artifact cannot be determined because it has not been tested in any way. Therefore, its credibility cannot be indicated. Thus, it is not recommended to rely solely on this artifact. Since this research is based on Information Systems research, it aims to develop a final artifact that will make it easier for developers and organizations. This is to discover and select APIs more efficiently and should ordinarily be assessed in a real-life scenario that has not been achieved due to lack of time.

When evaluating the research with the problem, theory, and models presented in the introduction and literature review chapter, the final result does correlate with the focus illustrated in the introduction and throughout the research.
However, section 3.3 above called, selection, mentions partner APIs in the first paragraph and is thought of throughout the research but fails to be expressed, which is something that would have been mentioned more if the research had been redone today. However, the purpose and goal are something that we believe are well-thought-out and would not be altered even if we were going to conduct this research again.

Lastly, the research methodology has been evaluated, and although it has been conducted in a thorough and well-analyzed manner, due to the limited time, it might have been more appropriate to use another research methodology. This to achieve a higher level of accuracy since we could for example when using a case study interacted with an organization and achieve the in real life expectancy of an IS research.

6.1 Future studies

It should be noted that the subject has good development potential for future research in several API-related areas that are not covered by this research. As this research is based on usability, documentation, stability, and collaborative community related to popular partner APIs, there is room to research the omitted parts mentioned in section 1.3 above.

6.2 Authors’ thoughts

APIs are an essential topic to investigate because, in many cases, an API is used for development purposes. Another necessary part of keeping in mind is to be aware of the features that an API includes when an API becomes a cornerstone of a fully developed application. Our thinking revolves around what actual consequences could result from implementing an API into an application without familiarizing oneself with its properties and what it contains.
References


