Vision: Investigating Web API Developer Experience in Relation to Terms of Service and Privacy Policies

Aidah Ichario, Manuel Maarek
School of Mathematical and Computer Sciences
Heriot-Watt University
Edinburgh, United Kingdom
ai58,M.Maarek@hw.ac.uk

***Abstract*—Rapid advancement in Web technologies has seen a shift from the use and implementation of closed applications to open information sharing in the form of Web APIs (Application Programming Interface). Developers make use of Web APIs by integrating them into their Web and mobile applications. These APIs however, come with Terms of Service (ToS) and Privacy Policies drawn up by the API providers that the developers must abide by. This research seeks to identify issues developers face with Web API ToS with regard to privacy. We propose to ground our understanding of ToS and privacy policy issues and their subsequent implications on Developer Experience from content analysis of developer discourse with card sorting. We report on a small scale pilot experiment.**

***Index Terms*—Developer Experience, Web API, Terms of Service, Privacy Policies, Content Analysis, Online Discourse**

1. Introduction

There is often a misconception and take-for-granted attitude when it comes to prioritising user experience in relation to software developers. This is mainly because unlike end-users, developers are tech savvy and are often expected to find their way around the technicalities of things. While it has become paramount to have user friendly and robust systems for end-users, the prevailing attitude towards software developers is that they are experts and *should know better*. For example, in the context of cryptography Application Programming Interface (API), developers often struggle with complex API and poor documentation forcing them to seek for answers from fellow developers on online developer communities like Stack Overflow, answers potentially undermine security [1]. This issue highlights the need for usable secure APIs (Application Programming Interface) [2], [3].

Web APIs, which allow connection and integration of Web services within applications, have seen a widespread adoption and growth over the past years [4]. Web APIs come with Terms of Service (ToS) including privacy policies to govern their use. Many Web API providers when designing ToS reuse existing ToS or worse, copy content from each other [5]. The similarities in the ToS often mislead API users on the compatibility of ToS of the different APIs they wish to integrate into their applications. Integration of APIs with incompatible data and privacy policies often lead to breach alerts and threats of termination of access to the API service by the API providers. In worse cases, API users are forced to drop the API in question and undo all the progress they have made in building their application. Often, leading to starting over with another API whose ToS or privacy policies are compatible. Furthermore, ToS are presented in an all-or-nothing fashion with little or no regard for the possibility of customising terms based on API user's needs. Web API providers are focusing on functionality suited to garner a big API user base. However, this focus could impact the User Experience (UX) which is the Developer Experience (DX) in this case. Moreover, ToS and privacy policies attached to API use are pre-requisite to accessing the API, which can impact DX.

In our research, we aim to contribute to a better understanding of Web API DX in relation to ToS and privacy policies. One may argue that *nobody really reads ToS*, which the NameDrop experiment [6], [7] proved to be the case. However, it is important to shift attention to the underlying reasons why these ToS are not being read and what consequences they pose. Efforts by *ToS;DR*[1] to apply ratings on ToS of different providers are a step in the right direction. And more can be done to understand and resolve issues with ToS and privacy polices of Web APIs that negatively impact DX. For that purpose we aim to answer the following research questions.

- **RQ1** What are the issues with Web API privacy policies?
- **RQ2** What implications do these issues have on Developer Experience?
- **RQ3** What Web API privacy policies best practices do developers identify?

To answer these research questions, we propose to employ grounded theory with content analysis and card sorting focus group in three phases. In this paper we present this methodology (Section 3) to characterise and validate Web API DX issues. We present the results of a pilot experiment (Section 4) we conducted.

2. Background and Related Work

This section reviews relevant literature on Web APIs, on terms of service and privacy policies of Web APIs, on the concept of DX, its importance in software engineering and how it ties into the topic of Web APIs and privacy.

1. *Terms of Service; Didn't Read* https://tosdr.org/

2.1. Web APIs

Web APIs enable programmers to write applications using a variety of publicly accessible Web services [8]. They provide a programmatic, network based access to remote data or functionalities.

Web APIs conforming to the REST architectural principles are characterised by relative simplicity and natural suitability for the Web while relying almost fully on the use of URIs for resource identification and HTTP for message transmission. Based on this simple technology, many Web sites such as Facebook, Google and Twitter offer easy to use public APIs. Thereby, allowing third parties to reuse heterogeneous data from diverse services in data oriented service compositions called *mashups*. For example, a chain of shops could enrich its mobile application with a map API to provide direction to its nearest shop. The same chain could provide an API showing available supplies in each shop. These services could be either free of charge or have a paid access.

These mashups participate in the emergence of service ecosystems where business functions are delivered as a service. As evidenced by data from ProgrammableWeb[2], thousands of public APIs are available and consumed to co-create new capabilities [9]. As the API economy matures, new opportunities emerge. Customers are turning into API and data providers, while citizen developers contribute to building ecosystems around Web APIs. The Web API providers in a bid to secure their economic interests, attach terms of service to their APIs to ensure API users stay within the bounds of these interests [10].

2.2. Terms of Service (ToS) of Web APIs

As the popularity of Web API use and adoption grows, so does its research [8], [11]. It however reveals little work in the area of Web API ToS.

The emergence of the API model of open partnerships and interoperability was born from Service oriented Architecture (SOA) which was built on the concept of business-to-business interactions. This API model re-defined ToS and Service Level Agreements (SLAs) from a one-to-one basis in SOA to a self-serve and instant gratification [5]. These self-serve services offered by Web APIs come with agree before use conditions known as ToS also synonymous with Terms and Conditions. A number of concerns were identified with existing ToS and privacy policies that may impact whether an API can be utilised in the development of new solutions [5]. They highlight issues that may arise from the assumption that agreement to the ToS is presumed based on continuous use. Web API providers often change the ToS and privacy policies with no notification to users of changes.

While Web APIs bring new and improved functionality, they come with incompatibility issues [12] linked to changes in ToS, privacy policies or the API itself. These often require modifications and upgrades on client programs. All the risks associated with these changes are borne by the API user. This is often outlined in the liability section of ToS by the API provider to distance themselves from the responsibility of any fallback from changes in the policy or the API itself. Third-party library updates are having an impact on end-user privacy and security [13].

2.3. Privacy Policies of Web APIs

Privacy policies are "the channel through which internet services communicate to their users the data they collect from them and what it is used for" [14]. These policies confront users with the option of either; a) accepting all the terms and losing control of their data, or b) rejecting the content of the policy and not being allowed access to the service. Web API users through integration of APIs in their applications may require to pull personal data. Web API providers, in a bid to protect their end-user data, define privacy policies API users must implement. API users are either required to embed a customised privacy policy of their application or a predefined policy is provided by the API provider. The API providers that require a customised privacy policy from the API user, often set minimum expectations that are linked to data policies required by law. Privacy policy enforcement however remains largely a human process [15] due to current lack of standards for privacy policies. In addition, policies are not directly connected to the procedures for implementing them. Although OpenAPI[3] offers a common specification language for describing APIs, no equivalent to W3C's P3P[4] has been adopted for Web APIs. Each API provider imposes different standards and requirements for privacy policies, it poses challenges for API users when selecting APIs to integrate in their applications. If the privacy concerns of a combination of APIs differ, it impacts the resulting composite service and its privacy policy for the end user. Such incompatibilities in the privacy policies of different Web APIs rob the API users of the freedom of choice. This limitation leaves them with only APIs whose terms are compatible, often with a high trade off on functionality which gravely impacts the DX. Another setback with the data policies is notifications of breaches in privacy policies. These breach notices come with threats of suspension of services and may therefore cause stress, anxiety and frustrations on the part of the developers. A study [16] suggested that psychological disorders like job burnout and anxiety among developers can be lowered significantly by lowering the number of bad experiences.

2.4. Developer Experience (DX)

The concept of DX [17] was influenced by the concept of UX. DX is defined as: "A means for capturing how developers think and feel about their activities within their working environments, with the assumption that an improvement of the developer experience has positive impacts on software development project outcomes" [17].

Software development is an inherently human-based, intellectual activity [18]. A number of studies [19], [20] indicate that human factors are the most important factors for software development both in terms of performance and quality [21]–[23]. The studies report a strong reliance of software project success on humans, while tools and methods only amplify the productivity of highly skilled

2. https://www.programmableweb.com/

3. https://www.openapis.org/

4. https://www.w3.org/P3P/

and well-coordinated development teams. The usability of API could impact security as recent studies into the usability of non-Web cryptography API have shown [24]–[26]. Recent works are focusing on analysing developers' security conversations [27], [28]. A systematic review [29] of literature on programmer experience published over the last 10 years presents the significance of developer environments and reports that motivation and choice of development tools played a big role in determining programmer experience.

3. Methodology

Our research approach takes on *Grounded Theory* to build a theory of the impact of Web API ToS and privacy policies on DX from developer online discourse. DX, in concept and definition, looks at the social-human aspect of software development that goes beyond the technical and economic dimensions. The psycho-social nature of the research topic called for a systematic approach to the discourse data with the primary focus of drawing meaning on how the identified issues are impacting DX. Grounded theory provides systematic steps that allows structuring and sorting of unstructured online discourse [30].

In this section we present our methodology decomposed into three phases as follows. Firstly, we investigate Web API policy issues through the eyes of the developers by analysing discourse from online communities (Section 3.1 and 3.2). Secondly, we derive from the structuring exposed by our content analysis of discourse, a set of issues for ToS, privacy policies and Web API DX (Section 3.3). Thirdly, we consolidate these findings by evaluating these issues and best practices with Web API users through a focused group *card sorting* activity (Section 3.4).

3.1. Data Collection and Cleaning

To gather discourse, we conducted Google searches for discourse relevant to our research topic. We searched for "Terms and privacy policy issues", "API restrictions", "API developer experience", and "Rant on API policy". The search returned links to Stack Overflow, Hackernoon, TechCrunch, IBM Developer, VentureBeat, DEV, Twitter, Reddit and DevRant. A scan through Stack Overflow did not yield discussion related content but Q&A posts so we decided not to explore it as a source of discourse data. We gathered articles on Web API DX from blogs: Hackernoon, TechCrunch, IBM Developer, VentureBeat, DEV and DevRant. Twitter and Reddit are social media websites where users use #-tags and subreddits `r/` respectively to label discussion boards with topics so that anyone interested in a topic can follow it and contribute. A Twitter search of #API, #APIpolicy, #APIprivacy and #APIissues yielded a large number of tweets with threads of discussion. These were extracted for filtering to find relevant discourse. A search in reddit, generated over 200 results tagged as discussion, rant, question, announcement. These results were filtered by "popular" and "all time", out of which twenty threads with the highest upvotes were selected for the data cleaning process. This brought the total number of discourse threads collected to 100.

We started our data cleaning process by categorising the gathered discourses into discourse files based on sources and topic. A discourse file comprises the downloaded Web pages and corresponding discussion thread. After categorising, we performed a content analysis using Nvivo software. This was done by importing the discourse files to Nvivo, then performing text search queries of keywords "privacy", "terms", "policies". These queries generated word trees linking dominant fragments of sentences to the keywords. This enabled us to filter out the most relevant discourse files for coding and analysis. The filtered files showed the highest saturation of discussion threads being on topics about three Web APIs Facebook, Instagram and Google. Bearing in mind the qualitative nature of this study and the scope of the pilot experiment, we narrowed our focus to online discourse about these three popular Web APIs. This criterion enabled us to filter the discourse files to 15 files. These were scrutinised to identify those specific to issues relating to privacy policies, ToS and DX. This finally resulted in 8 discourse files.

3.2. Discourse Coding

Coding of discourses was done using the three coding methods: open, axial and selective [30].
Open coding is the first step in the coding process which entailed reading each line of text to identify, name, categorise and describe phenomena found in the discourse text. The open coding process generated two abstract parent nodes: issues with ToS, and issues with privacy policies. As the open coding progressed, 12 child nodes corresponding to the abstract parent nodes were created as containers for emerging ideas from the discourse.
Axial coding, done simultaneously with open coding, is the process of identifying relationships between nodes through discourse with strong common themes.
Selective coding involves finding the main theme that drove the narrative around which the research questions revolved. This content analysis of discourse for the main theme in the form of selective coding identified developer experience as the driver of the study. As coding was done, *memo-ing* was done on reflections, ideas and emerging theories from the data concerning DX. One discourse file, representing more than 10% of the data, was coded by two coders. After one iteration which clarified some codes, the coding resulted in a satisfactory κ inter-rater reliability coefficient of 0.8 computed by NVivo. The coding of all 7 remaining files was performed by one coder.

3.3. Content Analysis of Discourse and Deriving Memos

Coding of discourse allowed us to engage with the data to identify how Web API ToS and privacy policies are impacting DX. The coded discourse revealed frustrations on the part of developers resulting from the current nature of ToS and privacy policies. These were coded under parent nodes: "Issues with ToS" and "Issues with privacy policies".

The combination of revelations from the analysed discourse and literature on ToS, privacy policies and DX (see Section 2) are used to design topic cards. Based

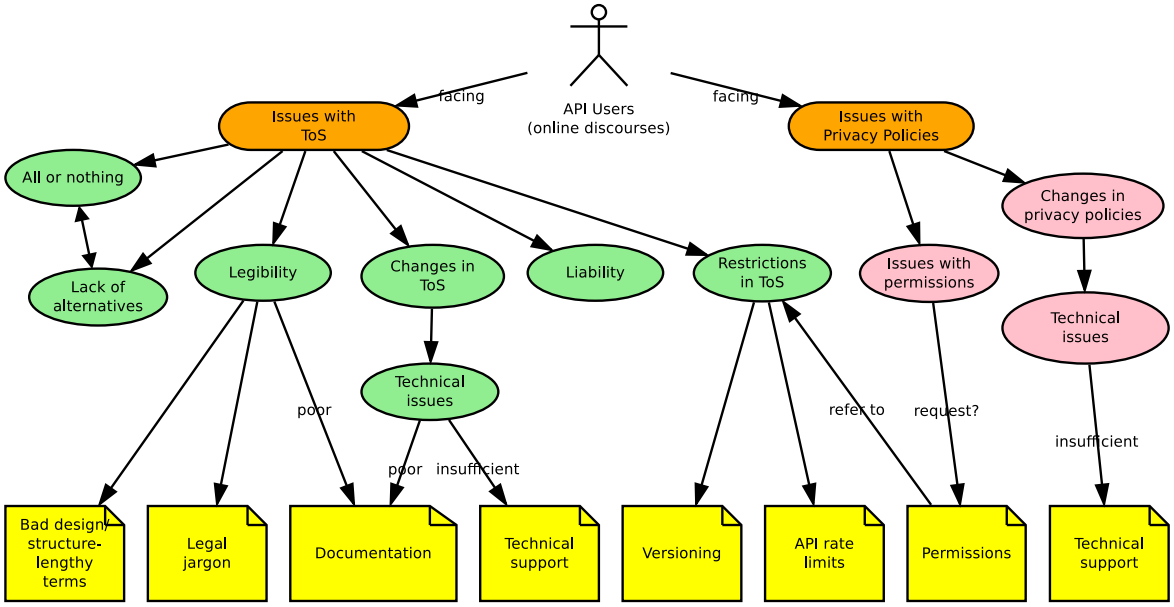


Figure 1: Concept Map of Perceptions Emerging from Content Analysis of Discourse
The concept map of perceptions emerging from content analysis of discourse consists of nodes and parent nodes, and memos identified by the coding process. We derived these memos and facts from the literature to create a set of issues/best practices cards.

on the child nodes, we write explanatory memos for each of the identified issues. The memo-ing process, core element of grounded theory, involves writing notes on ideas, reflections and opinions emerging from the data. These memos express issues and possible solutions in the form of best practices that can be adopted to improve DX with Web APIs.

3.4. Card Sorting Focus Group

All memos are transformed into cards. We use a focus group enhanced with a card sorting activity. This serves to mimic the situational context of online discourse. It provides an added value of observation, audio recording with the interview element incorporated in the structured tasks of the card sorting. Participants recruited among Web API users are presented with the deck of cards and a flipchart with three columns: ToS, privacy policies and DX. The participants are then tasked with the following:

- **Task 1** Sort the cards and categorise them based on the themes provided on the chart.
- **Task 2** Briefly discuss the reason for their sorting (audio recorded, prior signed consent was given).
- **Task 3** Identify what good practices can be adopted by API users before and while implementing Web APIs (best practices for API users).
- **Task 4** Identify which best practices are applicable to the API providers and designers of ToS. Do you agree with all the outlined practices? If not, discuss (audio recorded).

4. Results

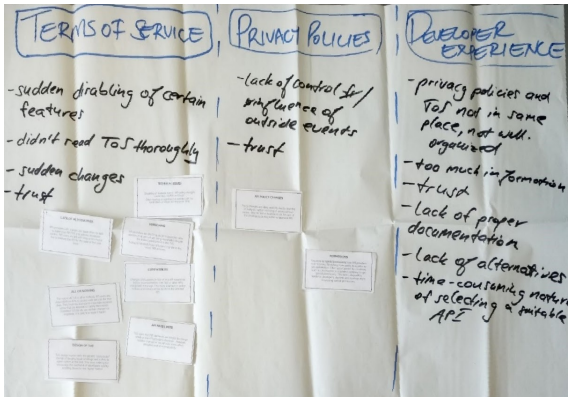
4.1. Analysing Discourses to Identify DX Issues

What emerges from the content analysis of discourse indicates that developers are facing a number of issues while using Web APIs. The most prominent being issues associated with ToS and privacy policies, e.g., legibility issues with ToS that are too lengthy, filled with legal jargon, and poorly documented. An API user in one of the discourses had this to say:

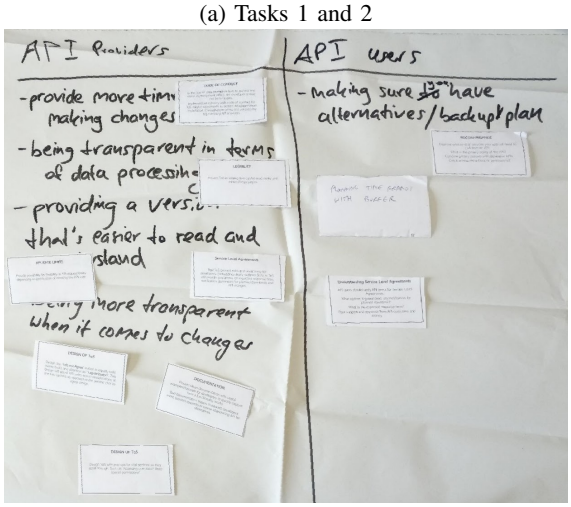
> "These terms of service can be pretty lengthy. It is important to highlight the sections in the terms or privacy policies that will impact them. Especially that they don't pay attention to ToS..."

This prompted a deeper search in the discourse for suggestions made by the API users on what can be done to "make their lives easier". While coding discourses and integrating facts from the literature review, we saw the emergence of additional ideas on how these can be applied to answer **RQ3** and alleviate the identified issues. These ideas were documented using the aid of *memos*. Figure 1 represents the concept map designed from memos of perceptions emerging from content analysis for issues with ToS and privacy policies. The two parent nodes; issues with ToS and issues with privacy policies each have corresponding child nodes. These child nodes contain coded discourse that may answer **RQ1** and **RQ2**.

The content analysis process also revealed the API user as not just a victim of bad practices by the API providers but also as a perpetrator of bad DX. Analysis of the API user stories on developer blogs revealed that some of the ramifications faced by API users were as a result of not reading ToS and privacy policies. Problems of privacy policy breaches, issues with special permissions and compatibility issues are a primary responsibility of the API user. They confessed to only re-visiting terms of use or privacy policies when served with breach notices by the API provider. It is no wonder that in the discourse, one user cited being banned from one API due to incompatibility issues with privacy policies of one application that integrated two or more APIs. It shows that Web API



(a) Tasks 1 and 2



(b) Tasks 3 and 4

Figure 2: Flipcharts Resulting from Card Sorting Activity

users are not checking and comparing ToS and privacy policies before selecting APIs to use in their applications.

As illustrated in the concept map in Figure 1, it emerged that API users are facing issues with privacy policies and ToS. These issues were sub categorised as child nodes under their respective parent nodes; issues with ToS and issues with privacy policies. We designed a deck of cards from the emergent ideas and perceptions from content analysis. For example, the memo on technical support details how changes in privacy policies were reported to often disable some features, causing applications to crash, leading to technical issues. Technical support from the API provider is said to be insufficient with significant downtime and no notification of expected response time. This memo was transformed into a card titled "Technical issues" and another card "Service Level Agreements".

4.2. Card Sorting Focus Group Activity

This style of evaluation was chosen with the aim of exploiting the benefit of a focused group discussion. Participants to the pilot experiment were recruited from postgraduate computer science students API users following ethical approval by the University.

Figure 2a shows the results from Tasks 1 and 2. The participants selected relevant cards from the deck and categorised them either under issues with ToS, privacy policies or DX. This task served as a way to validate the identified issues and validate their coding to the corresponding parent nodes representing the three themes. The outcome of these tasks are to answer **RQ1** and **RQ2**.

During the sorting process, the sample API users also discussed the issues portrayed in the cards. The transcribed audio discussion revealed they all agreed that the current design of ToS is painfully boring, with legal jargon and information scattered in the document. Often they find themselves quickly scrolling down for the *"Click to Agree"* button.

They also pointed out that sometimes they are faced with no alternatives but to use an API with unfavourable terms because it provides access to a large user base. These API providers being the key players in the market, often do little to provide sufficient technical support or decent documentation for the API user.

After mapping how the issues affect DX, Tasks 3 and 4 served the purpose of evaluating the applicability of the best practices presented in the cards. The outcome of these tasks are to answer **RQ3**. The output in Figure 2b shows the focus group evaluation and categorisation of the best practices in terms of what applies to the API users versus the API providers.

5. Discussion

In our approach, the focus group sorting of topic cards derived from content analysis serves to mimic the situational context of online discourse but with the added value of observation incorporated in the structured tasks of card sorting. This gave a fresh perspective on the issues and best practices by questioning and refining their applicability. Although the sample population of participants to this pilot study was small and therefore not diverse nor representative, it serves the purpose to ascertain the feasibility and scope of a full-scale experiment and to validate the methodology. The investigation shines a light on areas of concern in ToS and privacy policies for the Web API providers to look into, so as to provide usable Web APIs with favourable terms. Most importantly, through content analysis, this study confirmed the social impact of digital agreements; the click to *"Agree"* phenomena that comes with software tools. It gives the unadulterated insight into some of the daily struggles of the developer as a user through the *weltanschauung* of the API users in the online discourse.

The use of content analysis of discourse as an approach to the research problem while being appropriate is not without its shortcomings. The nature of content analysis entails studying text in conversations beyond the literal meaning while keeping in mind the situational context. This leaves room for a biased interpretation of the subjects meaning as the researchers view shapes the interpretation of the discourse. This research endeavoured to overcome this limitation of bias by incorporating key elements of discourse into a designed activity where participants were tasked to identify issues in the text that relate to the research problem and code into categories. The card sorting activity allowed participants freedom to "code" cards (blind coding), hence validating findings in relation to the themes representing the research topic. With sorting,

participants selected without bias the most relevant best practices in relation to the identified problems.

6. Conclusion

We presented a methodology combining grounded theory with content analysis of developer discourse, the design of topic cards, and a focus group card sorting. We conducted a pilot study applying this methodology to investigate issues with privacy policies in ToS and their implications on DX. The approach validates the content analysis of discourse with practitioners during card sorting activity. In our study, the card sorting focus group activity was tested on postgraduate students Web API users. We plan to run a full experiment including API providers and API designers to reach a more representative view.

The use of Web APIs in applications development has revolutionised the software engineering field. The benefits of Web APIs are undeniable. However, this pilot study reveals that ToS and privacy policies that come with Web APIs are much less favourable to API users and affect DX. We believe the full experiment could confirm that minimum standards and guidelines are needed for designing ToS and Privacy policies to make DX a priority and reduce the risks and uncertainty for API users. The methodology could also be employed to investigate other issues impacting DX.

References

[1] F. Fischer, K. Böttinger, H. Xiao, C. Stransky, Y. Acar, M. Backes, and S. Fahl, "Stack Overflow Considered Harmful? The Impact of Copy&Paste on Android Application Security," in *IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 121–136.

[2] M. Green and M. Smith, "Developers are Not the Enemy!: The Need for Usable Security APIs," *IEEE Security & Privacy*, vol. 14, no. 5, pp. 40–46, 2016.

[3] N. Patnaik, J. Hallett, and A. Rashid, "Usability Smells: An Analysis of Developers' Struggle With Crypto Libraries," in *Symposium on Usable Privacy and Security (SOUPS)*, 2019.

[4] L. Murphy, M. B. Kery, O. Alliyu, A. Macvean, and B. A. Myers, "API Designers in the Field: Design Practices and Challenges for Creating Usable APIs," in *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2018, pp. 249–258.

[5] M. Vukovic, J. Laredo, and S. Rajagopal, "API Terms and Conditions as a Service," in *2014 IEEE International Conference on Services Computing*, 2014, pp. 386–393.

[6] A. Oeldorf-Hirsch and J. A. Obar, "Overwhelming, Important, Irrelevant: Terms of Service and Privacy Policy Reading among Older Adults," in *ACM International Conference on Social Media and Society (SMSociety)*, 2019, pp. 166–173.

[7] J. A. Obar and A. Oeldorf-Hirsch, "The biggest lie on the Internet: Ignoring the privacy policies and terms of service policies of social networking services," *Information, Communication & Society*, vol. 23, no. 1, pp. 128–147, 2020.

[8] E. Wittern, A. T. Ying, Y. Zheng, J. A. Laredo, J. Dolby, C. C. Young, and A. A. Slominski, "Opportunities in Software Engineering Research for Web API Consumption," in *IEEE/ACM Int. Workshop on API Usage and Evolution (WAPI)*, 2017, pp. 7–10.

[9] K. Huang, Y. Fan, W. Tan, and X. Li, "Service Recommendation in an Evolving Ecosystem: A Link Prediction Approach," in *IEEE International Conference on Web Services*, 2013, pp. 507–514.

[10] M. Maleshkova, C. Pedrinaci, and J. Domingue, "Investigating Web APIs on the World Wide Web," in *IEEE European Conference on Web Services*, 2010, pp. 107–114.

[11] M. P. Robillard, E. Bodden, D. Kawrykow, M. Mezini, and T. Ratchford, "Automated API Property Inference Techniques," *IEEE Transactions on Software Engineering*, vol. 39, no. 5, pp. 613–637, 2013.

[12] J. Li, Y. Xiong, X. Liu, and L. Zhang, "How Does Web Service API Evolution Affect Clients?" in *IEEE International Conference on Web Services*, 2013, pp. 300–307.

[13] E. Derr, S. Bugiel, S. Fahl, Y. Acar, and M. Backes, "Keep me Updated: An Empirical Study of Third-Party Library Updatability on Android," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 2187–2200.

[14] O. Lawani, E. Aïmeur, and K. Dalkir, "Improving Users' Trust Through Friendly Privacy Policies: An Empirical Study," in *Risks and Security of Internet and Systems*, C. Lambrinoudakis and A. Gabillon, Eds. Springer, 2016, pp. 55–70.

[15] C.-M. Karat, J. Karat, C. Brodie, and J. Feng, "Evaluating interfaces for privacy policy rule authoring," in *ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2006, pp. 83–92.

[16] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, "On the Unhappiness of Software Developers," in *ACM International Conference on Evaluation and Assessment in Software Engineering (EASE)*, 2017, pp. 324–333.

[17] F. Fagerholm and J. Munch, "Developer experience: Concept and definition," in *IEEE International Conference on Software and System Process (ICSSP)*, 2012, pp. 73–77.

[18] A. Endres and H. D. Rombach, *A Handbook of Software and Systems Engineering: Empirical Observations, Laws, and Theories*. Pearson Education, 2003.

[19] H. Sackman, W. J. Erikson, and E. E. Grant, "Exploratory experimental studies comparing online and offline programming performance," *Communications of the ACM*, vol. 11, no. 1, pp. 3–11, 1968.

[20] T. DeMarco and T. Lister, *Peopleware: Productive Projects and Teams*. Addison-Wesley, 2013.

[21] A. Mockus, "Organizational volatility and its effects on software defects," in *ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE)*, 2010, pp. 117–126.

[22] N. Nagappan, B. Murphy, and V. Basili, "The influence of organizational structure on software quality," in *ACM/IEEE International Conference on Software Engineering (ICSE)*, 2008, pp. 521–530.

[23] C. Bird, N. Nagappan, H. Gall, B. Murphy, and P. Devanbu, "Putting It All Together: Using Socio-technical Networks to Predict Failures," in *International Symposium on Software Reliability Engineering (ISSRE)*, 2009, pp. 109–119.

[24] L. Lo Iacono and P. L. Gorski, "I Do and I Understand. Not Yet True for Security APIs. So Sad." in *European Workshop on Usable Security (EuroUSEC)*, 2017.

[25] P. L. Gorski, L. Lo Iacono, D. Wermke, C. Stransky, S. Möller, Y. Acar, and S. Fahl, "Developers Deserve Security Warnings, Too: On the Effect of Integrated Security Advice on Cryptographic API Misuse," in *Symposium on Usable Privacy and Security (SOUPS)*, 2018, pp. 265–281.

[26] J. Gao, P. Kong, L. Li, T. F. Bissyandé, and J. Klein, "Negative Results on Mining Crypto-API Usage Rules in Android Apps," in *IEEE/ACM International Conference on Mining Software Repositories (MSR)*, 2019, pp. 388–398.

[27] T. Lopez, H. Sharp, T. Tun, A. Bandara, M. Levine, and B. Nuseibeh, "Talking About Security with Professional Developers," in *IEEE/ACM Joint Int. Workshop on Conducting Empirical Studies in Industry (CESI) and Int. Workshop on Software Engineering Research and Industrial Practice (SER IP)*, 2019, pp. 34–40.

[28] T. Lopez, T. Tun, A. Bandara, L. Mark, B. Nuseibeh, and H. Sharp, "An Anatomy of Security Conversations in Stack Overflow," in *IEEE/ACM International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*, 2019, pp. 31–40.

[29] J. Morales, C. Rusu, F. Botella, and D. Quinones, "Programmer eXperience: A Systematic Literature Review," *IEEE Access*, vol. 7, pp. 71 079–71 094, 2019.

[30] B. G. Glaser and A. L. Strauss, *Discovery of Grounded Theory: Strategies for Qualitative Research*. Routledge, 2017.