Harvard John A. Paulson
School of Engineering
and Applied Sciences

PL/HCI Seminar (252R/279R)

How to Present a Paper in 279r

Teaching by Demonstration

Resources

1. Design Arguments handout
2. “8-pack” handout (to be posted)
3. Your peers’ questions in the forum

Design Argument #1

Evidence

How do you know?

Need

Need Thesis

Stakeholders + Domain Person **P** [in setting **S**]

Core tension wants to achieve goal **G** but obstacles $\mathbf{O}_{1\text{-}N}$ get in the way.

Evidence

How do existing approaches fail?

Any solution also has to:
satisfy constraints $\mathbf{X}_{1\text{-}N}$,
minimize costs $\mathbf{Y}_{1\text{-}N}$,
and avoid obstacles $\mathbf{Z}_{1\text{-}N}$.

Axioms As designers, we bring the following principles and constraints $\mathbf{A}_{1\text{-}N}$.

Approach Thesis

Our approach, ______________,
has characteristics $\mathbf{C}_{1\text{-}N}$
that help stakeholders achieve their goal **G** while avoiding obstacles $\mathbf{O}_{1\text{-}N}$

Evidence

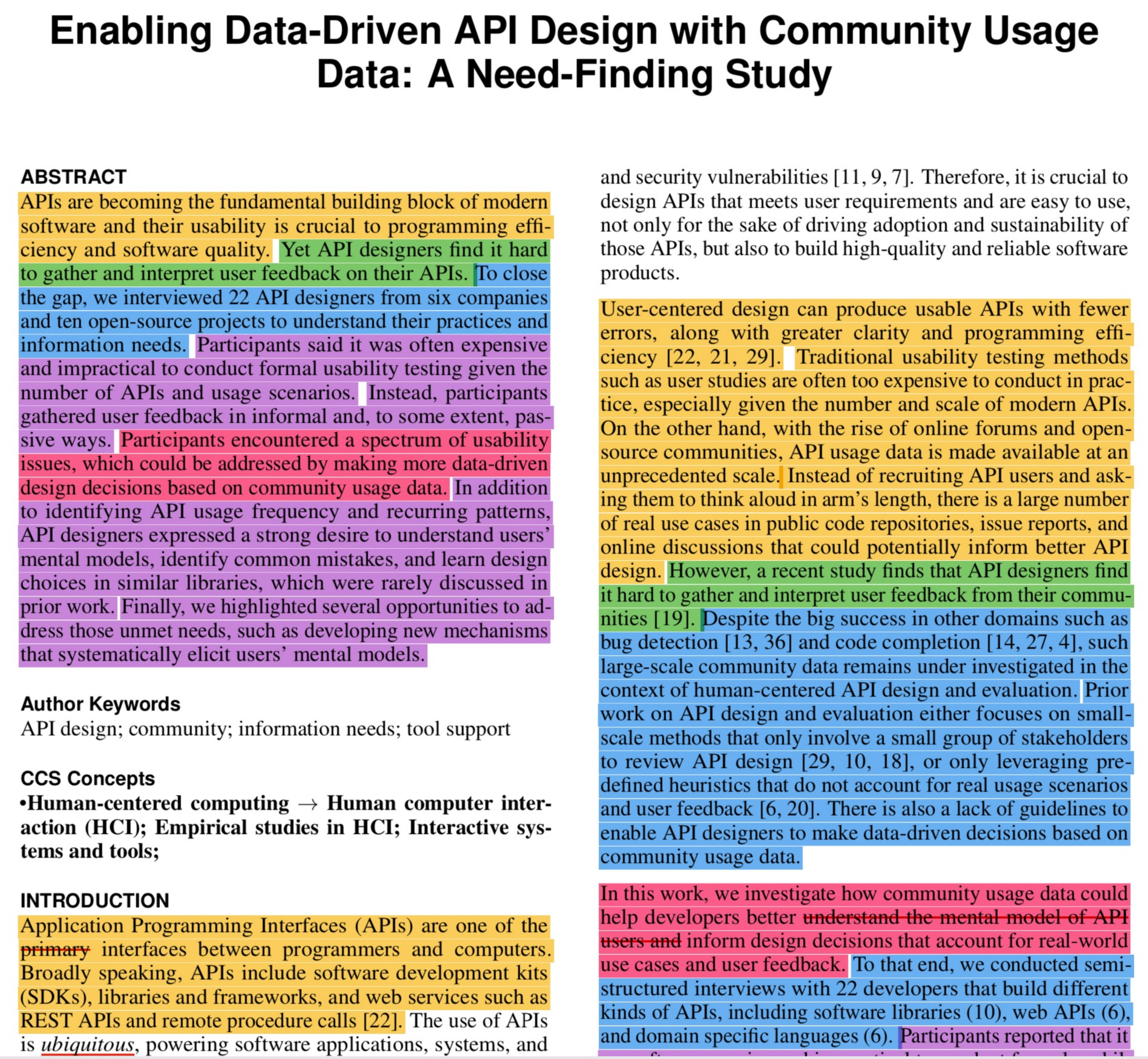
What characteristics have you borrowed from solutions that succeeded in analogous settings?

What differentiates your approach from previous solutions that failed?

How have stakeholders responded to/been able to use your approach?

Steps

1. Annotate the paper with respect to the design arguments



Enabling Data-Driven API Design with Community Usage Data: A Need-Finding Study

ABSTRACT

APIs are becoming the fundamental building block of modern software and their usability is crucial to programming efficiency and software quality. Yet API designers find it hard to gather and interpret user feedback on their APIs. To close the gap, we interviewed 22 API designers from six companies and ten open-source projects to understand their practices and information needs. Participants said it was often expensive and impractical to conduct formal usability testing given the number of APIs and usage scenarios. Instead, participants gathered user feedback in informal and, to some extent, passive ways. Participants encountered a spectrum of usability issues, which could be addressed by making more data-driven design decisions based on community usage data. In addition to identifying API usage frequency and recurring patterns, API designers expressed a strong desire to understand users' mental models, identify common mistakes, and learn design choices in similar libraries, which were rarely discussed in prior work. Finally, we highlighted several opportunities to address those unmet needs, such as developing new mechanisms that systematically elicit users' mental models.

Author Keywords

API design; community; information needs; tool support

CCS Concepts

•**Human-centered computing → Human computer interaction (HCI); Empirical studies in HCI; Interactive systems and tools;**

INTRODUCTION

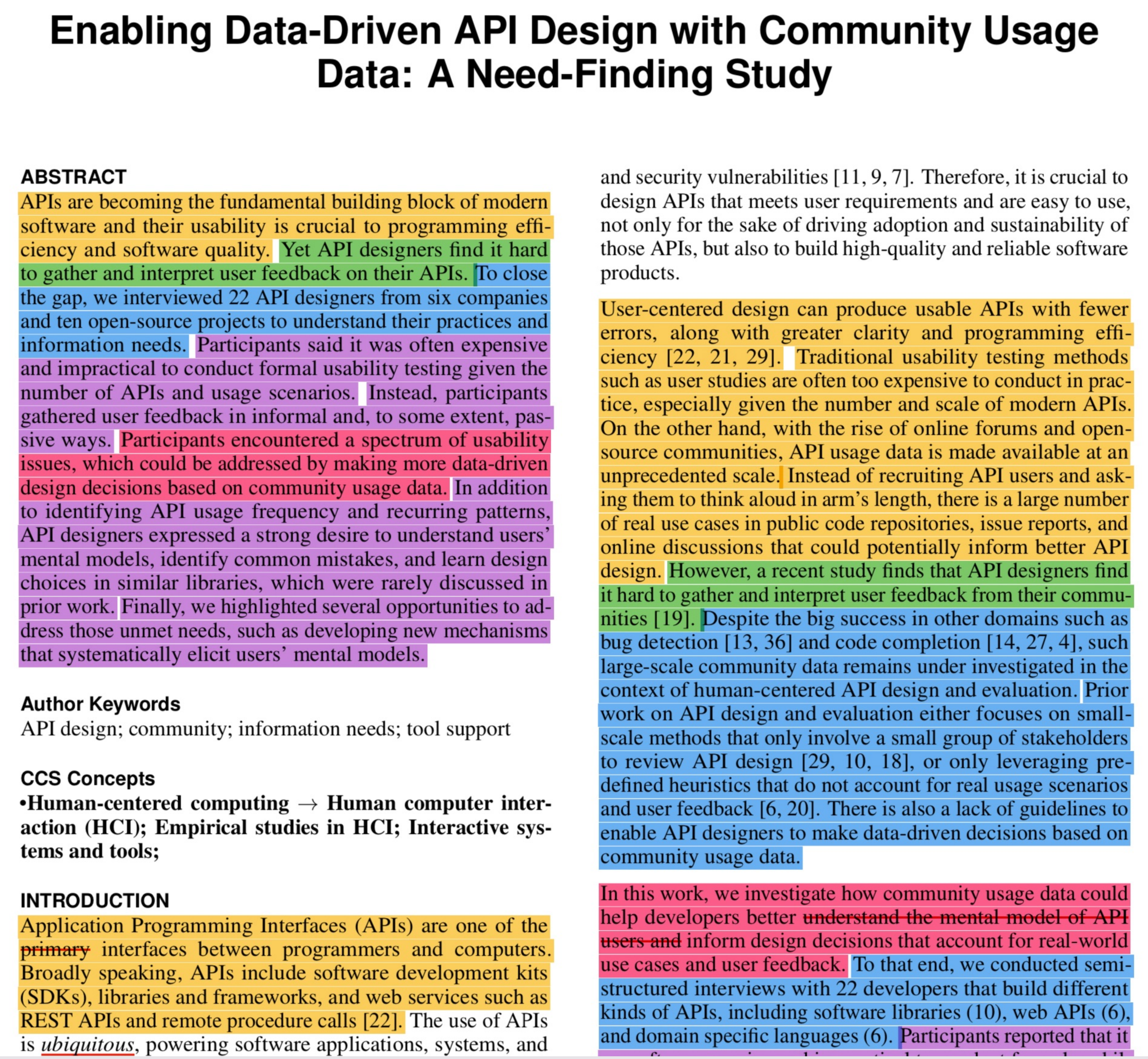
Application Programming Interfaces (APIs) are one of the ~~primary~~ interfaces between programmers and computers. Broadly speaking, APIs include software development kits (SDKs), libraries and frameworks, and web services such as REST APIs and remote procedure calls [22]. The use of APIs is *ubiquitous*, powering software applications, systems, and security vulnerabilities [11, 9, 7]. Therefore, it is crucial to design APIs that meets user requirements and are easy to use, not only for the sake of driving adoption and sustainability of those APIs, but also to build high-quality and reliable software products.

User-centered design can produce usable APIs with fewer errors, along with greater clarity and programming efficiency [22, 21, 29]. Traditional usability testing methods such as user studies are often too expensive to conduct in practice, especially given the number and scale of modern APIs. On the other hand, with the rise of online forums and open-source communities, API usage data is made available at an unprecedented scale. Instead of recruiting API users and asking them to think aloud in arm's length, there is a large number of real use cases in public code repositories, issue reports, and online discussions that could potentially inform better API design. However, a recent study finds that API designers find it hard to gather and interpret user feedback from their communities [19]. Despite the big success in other domains such as bug detection [13, 36] and code completion [14, 27, 4], such large-scale community data remains under investigated in the context of human-centered API design and evaluation. Prior work on API design and evaluation either focuses on small-scale methods that only involve a small group of stakeholders to review API design [29, 10, 18], or only leveraging predefined heuristics that do not account for real usage scenarios and user feedback [6, 20]. There is also a lack of guidelines to enable API designers to make data-driven decisions based on community usage data.

In this work, we investigate how community usage data could help developers better ~~understand the mental model of API users and~~ inform design decisions that account for real-world use cases and user feedback. To that end, we conducted semi-structured interviews with 22 developers that build different kinds of APIs, including software libraries (10), web APIs (6), and domain specific languages (6). Participants reported that it

Steps

1. Annotate the paper with respect to the design arguments
2. Try to fill out the 8-pack



Enabling Data-Driven API Design with Community Usage Data: A Need-Finding Study

ABSTRACT

APIs are becoming the fundamental building block of modern software and their usability is crucial to programming efficiency and software quality. Yet API designers find it hard to gather and interpret user feedback on their APIs. To close the gap, we interviewed 22 API designers from six companies and ten open-source projects to understand their practices and information needs. Participants said it was often expensive and impractical to conduct formal usability testing given the number of APIs and usage scenarios. Instead, participants gathered user feedback in informal and, to some extent, passive ways. Participants encountered a spectrum of usability issues, which could be addressed by making more data-driven design decisions based on community usage data. In addition to identifying API usage frequency and recurring patterns, API designers expressed a strong desire to understand users' mental models, identify common mistakes, and learn design choices in similar libraries, which were rarely discussed in prior work. Finally, we highlighted several opportunities to address those unmet needs, such as developing new mechanisms that systematically elicit users' mental models.

Author Keywords

API design; community; information needs; tool support

CCS Concepts

•**Human-centered computing** → **Human computer interaction (HCI); Empirical studies in HCI; Interactive systems and tools;**

INTRODUCTION

Application Programming Interfaces (APIs) are one of the ~~primary~~ interfaces between programmers and computers. Broadly speaking, APIs include software development kits (SDKs), libraries and frameworks, and web services such as REST APIs and remote procedure calls [22]. The use of APIs is *ubiquitous*, powering software applications, systems, and

and security vulnerabilities [11, 9, 7]. Therefore, it is crucial to design APIs that meets user requirements and are easy to use, not only for the sake of driving adoption and sustainability of those APIs, but also to build high-quality and reliable software products.

User-centered design can produce usable APIs with fewer errors, along with greater clarity and programming efficiency [22, 21, 29]. Traditional usability testing methods such as user studies are often too expensive to conduct in practice, especially given the number and scale of modern APIs. On the other hand, with the rise of online forums and open-source communities, API usage data is made available at an unprecedented scale. Instead of recruiting API users and asking them to think aloud in arm's length, there is a large number of real use cases in public code repositories, issue reports, and online discussions that could potentially inform better API design. However, a recent study finds that API designers find it hard to gather and interpret user feedback from their communities [19]. Despite the big success in other domains such as bug detection [13, 36] and code completion [14, 27, 4], such large-scale community data remains under investigated in the context of human-centered API design and evaluation. Prior work on API design and evaluation either focuses on small-scale methods that only involve a small group of stakeholders to review API design [29, 10, 18], or only leveraging predefined heuristics that do not account for real usage scenarios and user feedback [6, 20]. There is also a lack of guidelines to enable API designers to make data-driven decisions based on community usage data.

In this work, we investigate how community usage data could help developers better ~~understand the mental model of API users and~~ inform design decisions that account for real-world use cases and user feedback. To that end, we conducted semi-structured interviews with 22 developers that build different kinds of APIs, including software libraries (10), web APIs (6), and domain specific languages (6). Participants reported that it

8-pack

practical

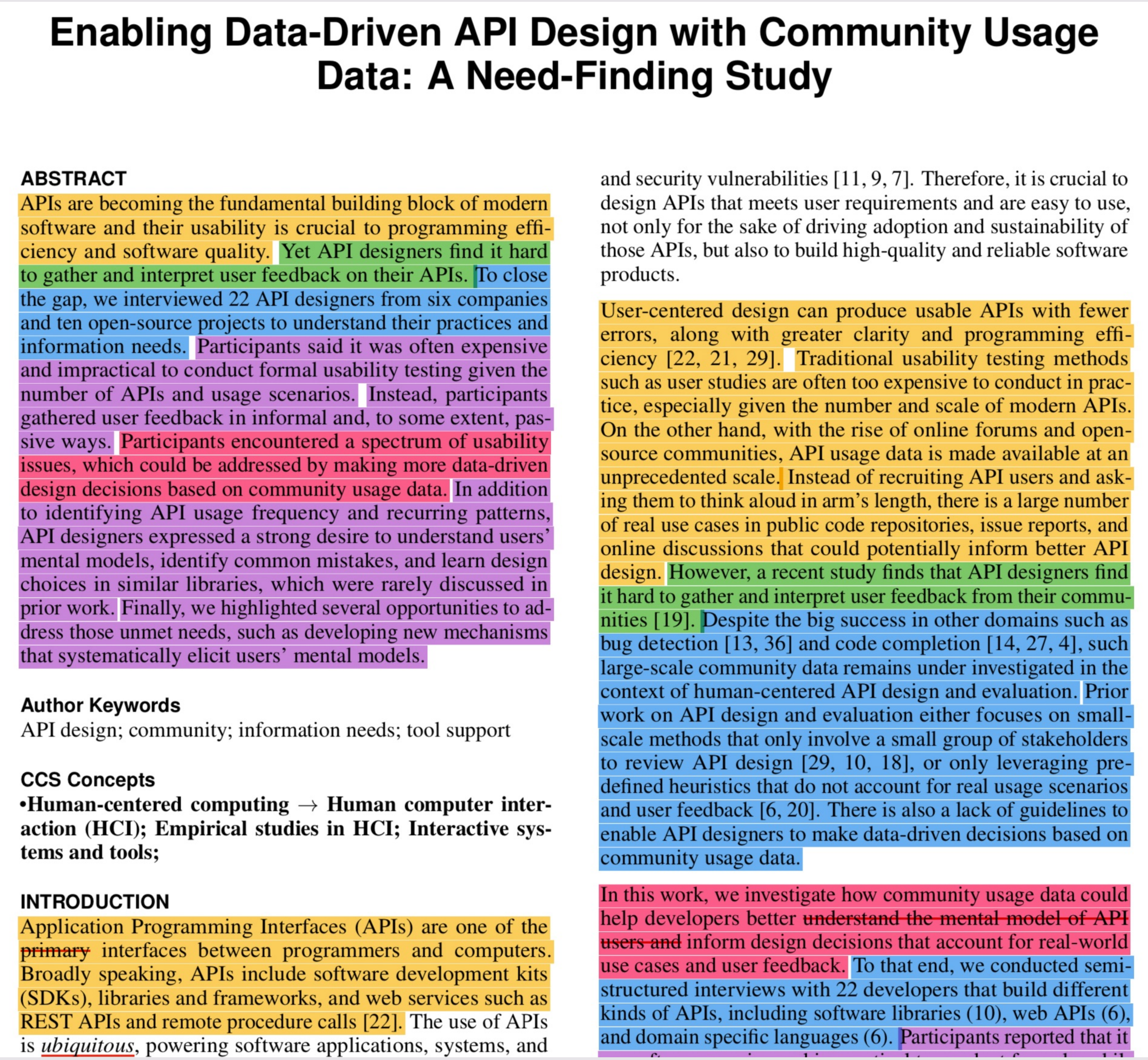
practical problem		**design arguments**	**system model**	**findings**
[**SETTING**] **SETTING:** **CHALLENGES:** **CONSEQUENCE:**	[**SETTING**] **SETTING:** **CHALLENGES:** **CONSEQUENCE:**	To do [what we want to do] we introduce [what we've built] [which allows]		

research

class of problems	**conceptual approach**	**technical approach**	**takeaways**
SETTING: **CHALLENGES:** **CONSEQUENCES:**			

Steps

1. Annotate the paper with respect to the design arguments
2. Try to fill out the 8-pack
3. Go over questions in forum looking for common & excellent questions to try to answer in advance



Enabling Data-Driven API Design with Community Usage Data: A Need-Finding Study

ABSTRACT

APIs are becoming the fundamental building block of modern software and their usability is crucial to programming efficiency and software quality. Yet API designers find it hard to gather and interpret user feedback on their APIs. To close the gap, we interviewed 22 API designers from six companies and ten open-source projects to understand their practices and information needs. Participants said it was often expensive and impractical to conduct formal usability testing given the number of APIs and usage scenarios. Instead, participants gathered user feedback in informal and, to some extent, passive ways. Participants encountered a spectrum of usability issues, which could be addressed by making more data-driven design decisions based on community usage data. In addition to identifying API usage frequency and recurring patterns, API designers expressed a strong desire to understand users' mental models, identify common mistakes, and learn design choices in similar libraries, which were rarely discussed in prior work. Finally, we highlighted several opportunities to address those unmet needs, such as developing new mechanisms that systematically elicit users' mental models.

Author Keywords

API design; community; information needs; tool support

CCS Concepts

•Human-centered computing → Human computer interaction (HCI); Empirical studies in HCI; Interactive systems and tools;

INTRODUCTION

Application Programming Interfaces (APIs) are one of the ~~primary~~ interfaces between programmers and computers. Broadly speaking, APIs include software development kits (SDKs), libraries and frameworks, and web services such as REST APIs and remote procedure calls [22]. The use of APIs is *ubiquitous*, powering software applications, systems, and

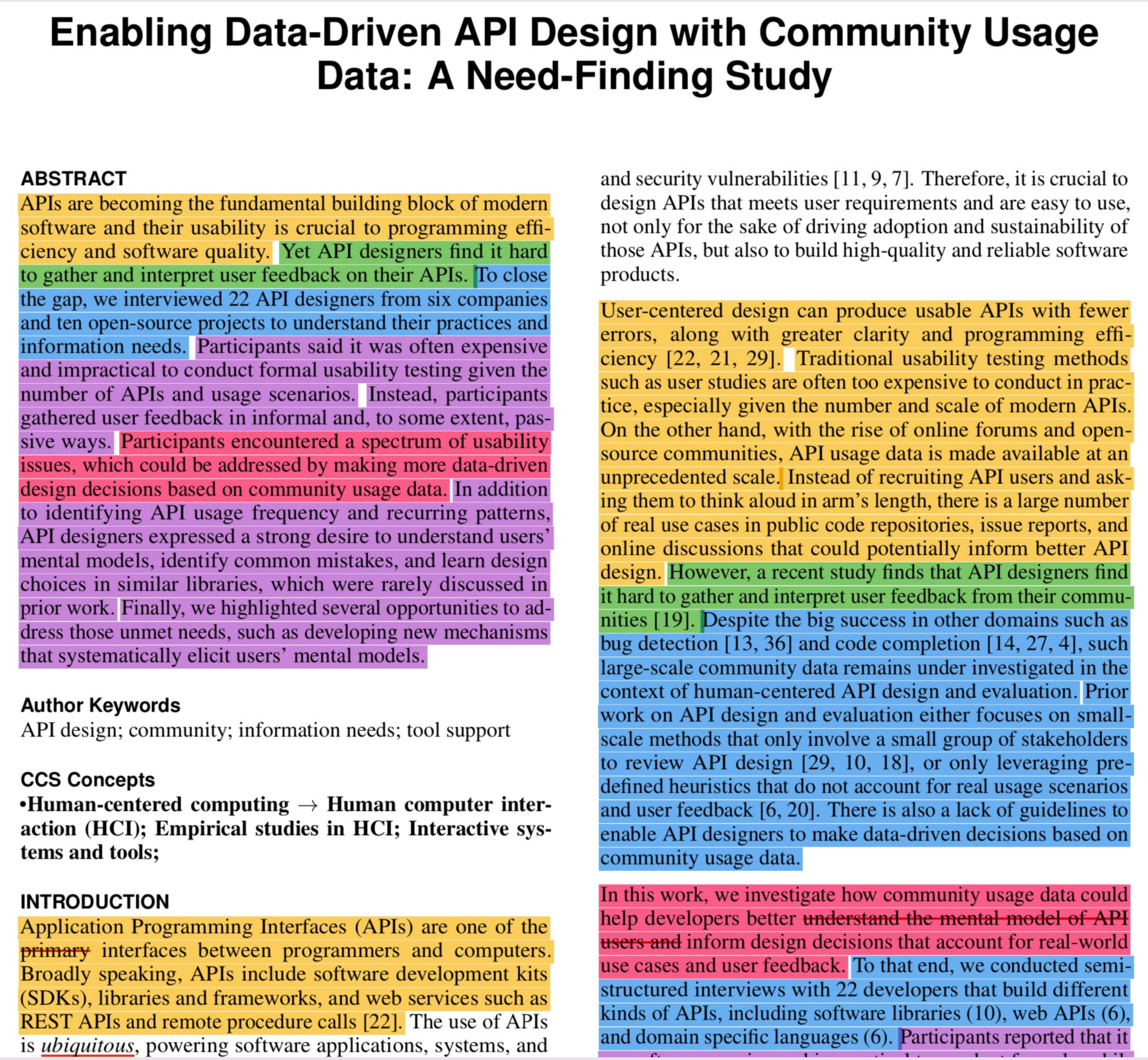
and security vulnerabilities [11, 9, 7]. Therefore, it is crucial to design APIs that meets user requirements and are easy to use, not only for the sake of driving adoption and sustainability of those APIs, but also to build high-quality and reliable software products.

User-centered design can produce usable APIs with fewer errors, along with greater clarity and programming efficiency [22, 21, 29]. Traditional usability testing methods such as user studies are often too expensive to conduct in practice, especially given the number and scale of modern APIs. On the other hand, with the rise of online forums and open-source communities, API usage data is made available at an unprecedented scale. Instead of recruiting API users and asking them to think aloud in arm's length, there is a large number of real use cases in public code repositories, issue reports, and online discussions that could potentially inform better API design. However, a recent study finds that API designers find it hard to gather and interpret user feedback from their communities [19]. Despite the big success in other domains such as bug detection [13, 36] and code completion [14, 27, 4], such large-scale community data remains under investigated in the context of human-centered API design and evaluation. Prior work on API design and evaluation either focuses on small-scale methods that only involve a small group of stakeholders to review API design [29, 10, 18], or only leveraging pre-defined heuristics that do not account for real usage scenarios and user feedback [6, 20]. There is also a lack of guidelines to enable API designers to make data-driven decisions based on community usage data.

In this work, we investigate how community usage data could help developers better ~~understand the mental model of API users and~~ inform design decisions that account for real-world use cases and user feedback. To that end, we conducted semi-structured interviews with 22 developers that build different kinds of APIs, including software libraries (10), web APIs (6), and domain specific languages (6). Participants reported that it

Steps

1. Annotate the paper with respect to the design arguments
2. Try to fill out the 8-pack
3. Go over questions in forum looking for common & excellent questions to try to answer in advance
4. Present design arguments, 8-pack, answers to questions, and a discussion question in a set of slides



Enabling Data-Driven API Design with Community Usage Data: A Need-Finding Study

ABSTRACT

APIs are becoming the fundamental building block of modern software and their usability is crucial to programming efficiency and software quality. Yet API designers find it hard to gather and interpret user feedback on their APIs. To close the gap, we interviewed 22 API designers from six companies and ten open-source projects to understand their practices and information needs. Participants said it was often expensive and impractical to conduct formal usability testing given the number of APIs and usage scenarios. Instead, participants gathered user feedback in informal and, to some extent, passive ways. Participants encountered a spectrum of usability issues, which could be addressed by making more data-driven design decisions based on community usage data. In addition to identifying API usage frequency and recurring patterns, API designers expressed a strong desire to understand users' mental models, identify common mistakes, and learn design choices in similar libraries, which were rarely discussed in prior work. Finally, we highlighted several opportunities to address those unmet needs, such as developing new mechanisms that systematically elicit users' mental models.

Author Keywords

API design; community; information needs; tool support

CCS Concepts

•Human-centered computing → Human computer interaction (HCI); Empirical studies in HCI; Interactive systems and tools;

INTRODUCTION

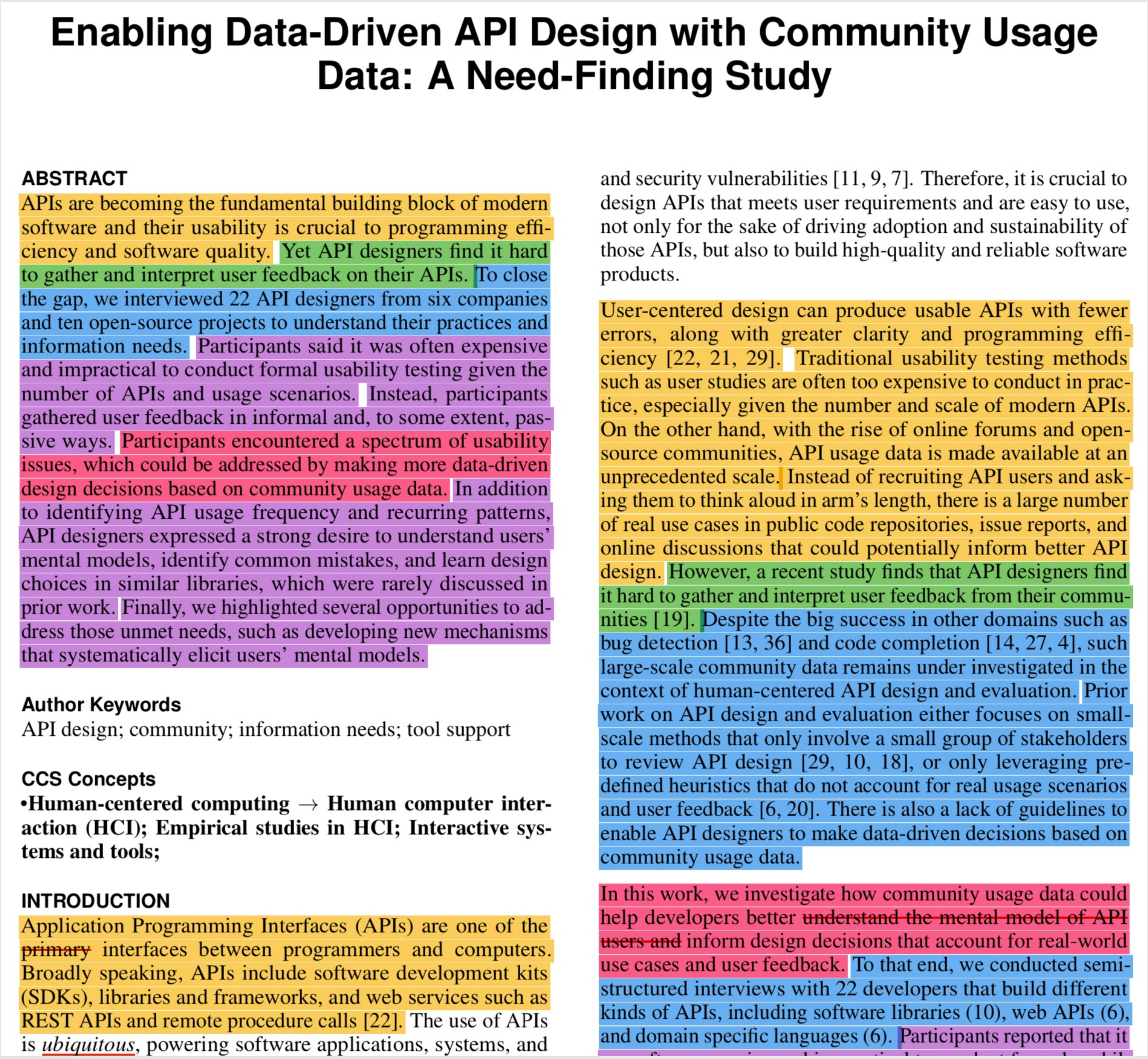
Application Programming Interfaces (APIs) are one of the ~~primary~~ interfaces between programmers and computers. Broadly speaking, APIs include software development kits (SDKs), libraries and frameworks, and web services such as REST APIs and remote procedure calls [22]. The use of APIs is *ubiquitous*, powering software applications, systems, and and security vulnerabilities [11, 9, 7]. Therefore, it is crucial to design APIs that meets user requirements and are easy to use, not only for the sake of driving adoption and sustainability of those APIs, but also to build high-quality and reliable software products.

User-centered design can produce usable APIs with fewer errors, along with greater clarity and programming efficiency [22, 21, 29]. Traditional usability testing methods such as user studies are often too expensive to conduct in practice, especially given the number and scale of modern APIs. On the other hand, with the rise of online forums and open-source communities, API usage data is made available at an unprecedented scale. Instead of recruiting API users and asking them to think aloud in arm's length, there is a large number of real use cases in public code repositories, issue reports, and online discussions that could potentially inform better API design. However, a recent study finds that API designers find it hard to gather and interpret user feedback from their communities [19]. Despite the big success in other domains such as bug detection [13, 36] and code completion [14, 27, 4], such large-scale community data remains under investigated in the context of human-centered API design and evaluation. Prior work on API design and evaluation either focuses on small-scale methods that only involve a small group of stakeholders to review API design [29, 10, 18], or only leveraging pre-defined heuristics that do not account for real usage scenarios and user feedback [6, 20]. There is also a lack of guidelines to enable API designers to make data-driven decisions based on community usage data.

In this work, we investigate how community usage data could help developers better ~~understand the mental model of API users and~~ inform design decisions that account for real-world use cases and user feedback. To that end, we conducted semi-structured interviews with 22 developers that build different kinds of APIs, including software libraries (10), web APIs (6), and domain specific languages (6). Participants reported that it

Steps

1. Annotate the paper with respect to the design arguments
2. Try to fill out the 8-pack
3. Go over questions in forum looking for common & excellent questions to try to answer in advance
4. Present design arguments, 8-pack, answers to questions, and a discussion question in a set of slides
5. Remind yourself of the time: 15-20 min presenting; 15-10 min discussion.



Enabling Data-Driven API Design with Community Usage Data: A Need-Finding Study

ABSTRACT

APIs are becoming the fundamental building block of modern software and their usability is crucial to programming efficiency and software quality. Yet API designers find it hard to gather and interpret user feedback on their APIs. To close the gap, we interviewed 22 API designers from six companies and ten open-source projects to understand their practices and information needs. Participants said it was often expensive and impractical to conduct formal usability testing given the number of APIs and usage scenarios. Instead, participants gathered user feedback in informal and, to some extent, passive ways. Participants encountered a spectrum of usability issues, which could be addressed by making more data-driven design decisions based on community usage data. In addition to identifying API usage frequency and recurring patterns, API designers expressed a strong desire to understand users' mental models, identify common mistakes, and learn design choices in similar libraries, which were rarely discussed in prior work. Finally, we highlighted several opportunities to address those unmet needs, such as developing new mechanisms that systematically elicit users' mental models.

Author Keywords

API design; community; information needs; tool support

CCS Concepts

•**Human-centered computing → Human computer interaction (HCI); Empirical studies in HCI; Interactive systems and tools;**

INTRODUCTION

Application Programming Interfaces (APIs) are one of the ~~primary~~ interfaces between programmers and computers. Broadly speaking, APIs include software development kits (SDKs), libraries and frameworks, and web services such as REST APIs and remote procedure calls [22]. The use of APIs is *ubiquitous*, powering software applications, systems, and and security vulnerabilities [11, 9, 7]. Therefore, it is crucial to design APIs that meets user requirements and are easy to use, not only for the sake of driving adoption and sustainability of those APIs, but also to build high-quality and reliable software products.

User-centered design can produce usable APIs with fewer errors, along with greater clarity and programming efficiency [22, 21, 29]. Traditional usability testing methods such as user studies are often too expensive to conduct in practice, especially given the number and scale of modern APIs. On the other hand, with the rise of online forums and open-source communities, API usage data is made available at an unprecedented scale. Instead of recruiting API users and asking them to think aloud in arm's length, there is a large number of real use cases in public code repositories, issue reports, and online discussions that could potentially inform better API design. However, a recent study finds that API designers find it hard to gather and interpret user feedback from their communities [19]. Despite the big success in other domains such as bug detection [13, 36] and code completion [14, 27, 4], such large-scale community data remains under investigated in the context of human-centered API design and evaluation. Prior work on API design and evaluation either focuses on small-scale methods that only involve a small group of stakeholders to review API design [29, 10, 18], or only leveraging predefined heuristics that do not account for real usage scenarios and user feedback [6, 20]. There is also a lack of guidelines to enable API designers to make data-driven decisions based on community usage data.

In this work, we investigate how community usage data could help developers better ~~understand the mental model of API users and~~ inform design decisions that account for real-world use cases and user feedback. To that end, we conducted semi-structured interviews with 22 developers that build different kinds of APIs, including software libraries (10), web APIs (6), and domain specific languages (6). Participants reported that it