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
Need Thesis

**Stakeholders + Domain**
Person P [in setting S] wants to achieve goal G but obstacles O_{1-N} get in the way.

**Core tension**
Any solution also has to:
- satisfy constraints X_{1-N},
- minimize costs Y_{1-N},
- and avoid obstacles Z_{1-N}.

**Axioms**
As designers, we bring the following principles and constraints A_{1-N}.

Approach Thesis

Our approach, ____________, has characteristics C_{1-N} that help stakeholders achieve their goal G while avoiding obstacles O_{1-N}.

Evidence

- **Need**
  - How do you know?

- **Evidence**
  - How do existing approaches fail?

- **Evidence**
  - What differentiates your approach from previous solutions that failed?

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

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

and security vulnerabilities [11, 9, 7]. Therefore, it is crucial to design APIs that meet 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

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 meet 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 [5, 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

<table>
<thead>
<tr>
<th>practical problem</th>
<th>design arguments</th>
<th>system model</th>
<th>findings</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SETTING]</td>
<td>[SETTING]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SETTING:</td>
<td>SETTING:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHALLENGES:</td>
<td>CHALLENGES:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSEQUENCE:</td>
<td>CONSEQUENCE:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>To do [what we want to do]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>we introduce [what we've built]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[which allows]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## research

<table>
<thead>
<tr>
<th>class of problems</th>
<th>conceptual approach</th>
<th>technical approach</th>
<th>takeaways</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETTING:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CHALLENGES:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSEQUENCES:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 these 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 meet 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 the 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 [5, 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 security vulnerabilities [11, 9, 7]. Therefore, it is crucial to design APIs that meet 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 underinvestigated 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 [5, 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
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.