Where the User meets the Code
Lessons from Interaction Design
@noahcrowley
GitHub, Twitter
Time Series Data

• Anything with a **Timestamp + Value**

• 2018-09-25 2:30pm CDT, 72°F

• 93.180.71.3 - - [17/May/2015:08:05:32 +0000] "GET /downloads/product_1 HTTP/1.1" 304 0 "-" "Debian APT-HTTP/1.3 (0.8.16~exp12ubuntu10.21)"
I’m a Developer Advocate

& member of the Developer Relations team
InfluxData Platform

API-centric
Building an API

- Business Requirements
- Information Architecture
- Users
- Protocols
- REST paradigm
API Design = Interaction Design
Users & Products

Moment of Interaction
Users & API
1980s
Bill Moggridge and Bill Verplank
IxD draws from multiple disciplines

- Information Architecture
- Psychology & Social Science
- Graphic & Industrial Design
- Semiotics
- Ethnography

- Overlap with User Experience (UX) design
IxD is still young

- Many different modalities and methodologies
- Terms, concepts, and definitions still evolving
- Provide a useful set of tools for framing the discussion around designing for the moment of interaction
Personas
Affordances
Cognitive Friction
Personas
Persona = Representation of a User
The practice of Personas

- A “fuzzy” practice with various approaches
- Qualitative vs. Quantitative
- Use of fiction & assumptions
- Iterative
- Allow you to create archetypes and representations that can be shared and discussed
Validate assumptions
Gather new information
You != Your Users
Active Process
Learn -> Synthesize -> Share
Drive Engineering Priorities
Developing Personas

- Conduct user research—gather trends
- Summarize and synthesize
- Conduct further research on groups and patterns you’ve identified
- Create Personas
- Create Scenarios
- Share!
Personas -> Users
Affordances -> Products
Origin & Adoption

- Root in evolutionary psychology—coined by James Gibson in 1977
- All action possibilities afforded to the animal by the environment
- Appropriated by Don Norman for use in Interaction Design in 1988
Physical vs. Virtual Affordances
Affordances in IxD

- Began to diverge from ecological psychology definition
- Man made objects have physical affordances, but they also have affordances of other types
- Our environment includes social structures, culture, and past experience
- Norman would later argue that signifiers are more important than affordances
Sequential & Nested Affordances
Affordances in API Design

- Protocol affordances:
  - UDP vs. TCP
- API Paradigms: REST vs. gRPC
- Information Architecture
- Dealing with False & Hidden affordances
Cognitive Friction
Mismatch between mental model & reality
HTTP 200

The system encountered an error.
Easing Cognitive Friction
Lists all the endpoints

List of the endpoints.

Responses

↑ 200 Returns the links to the top level endpoints.

RESPONSE SCHEMA

```
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>me</td>
<td>string &lt;url&gt; Location of the me endpoint.</td>
</tr>
<tr>
<td>layouts</td>
<td>string &lt;url&gt; Location of the layouts endpoint</td>
</tr>
<tr>
<td>sources</td>
<td>string &lt;url&gt; Location of the sources endpoint</td>
</tr>
<tr>
<td>mappings</td>
<td>string &lt;url&gt; Location of the application mappings endpoint</td>
</tr>
<tr>
<td>dashboards</td>
<td>string &lt;url&gt; Location of the dashboards endpoint</td>
</tr>
<tr>
<td>external</td>
<td>external links provided to client, ex. status feed URL</td>
</tr>
</tbody>
</table>
```

RESPONSE SAMPLES

- 200 Returns the links to the top level endpoints
- default Unexpected internal server error
Documentation & Errors

- Your signifiers
- Swagger Documentation
  - Provides signifiers and allows users to construct a mental model
  - Adds affordances, eases requirements for interacting with API
- Error Codes & Messages
  - Provides feedback at the moment of friction
Layering affordances

- InfluxData Platform
- API built on top of HTTP
- Swagger documentation built on top of API
- Client library that interacts with API (including REPL)
- Give the user a way to explore the space and build their mental model
- Provide immediate feedback when that model is incorrect
The Future
“People who are really serious about software should make their own hardware.”

-Alan Kay

Creative Think Seminar, July 1982
Thanks!
@noahcrowley