

Redactive

What we learnt building our AI product

—zapier

Felicis

▲ ATlassian Ventures

BLACKBIRD



Redactive enables enterprises to build secure
GenAI apps with real time permissioned data






It's been a ride!





December 2023 - Starts work on first product (RTSS).

January 2024 - Product conversations and partnerships.



February 2024 - First implementation of designed architecture.

March 2023 - Testing, refactor, testing, refactor.



April 2023 - Grinding, component assembly, website completion.

May 2024 - Product launch



#1

Build with a Purpose

Spend as much time as possible refining the core product, the core feature that you are trying to create. Really try to understand what the purpose, use case, value of your product is.

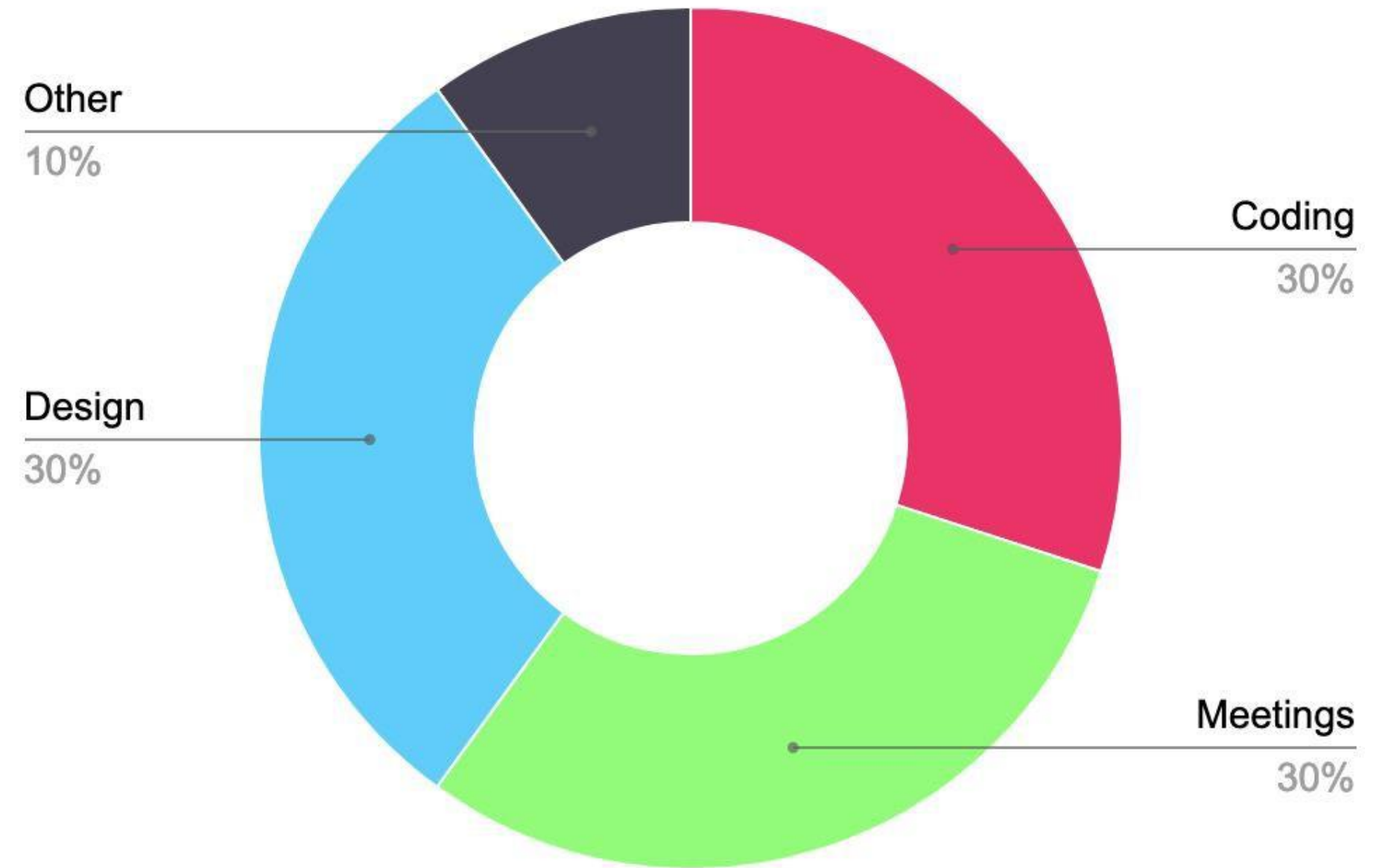
```
82     async def get_user_ids_by_email(self, emails: list[str]) -> list[str]: ...
83
84
85     """
86     Returns a list of provider-specific user IDs from a given list of user `emails`.
87
88     The returned list of user IDs may represent a subset of `emails` if:
89     - an email is not be associated with an account with the provider
90     - the querying user does not have permissions with the provider to search user IDs by email
91     - the provider does not offer an API endpoint to to search user IDs by email, instead returning an empty list
92     """
93
94     async def fetch_chunks(self, source_ref: SourceReference) -> list[Chunk]: ...
95
96     # Enumerate source references for data connector. This should filter out references
97     # that we definitely know we can't index, but may include some references we aren't sure about.
98     async def enumerate_source_references(self, since: datetime | None = None) -> AsyncIterable[list[SourceReference]]:
99         # to stop pyright complaining about incompatible return types in implementing classes.
100         # https://discuss.python.org/t/overloads-of-async-generators-inconsistent-coroutine-wrapping/56665/4
101         yield []
102
103     async def refresh_token(self, refresh_token: str) -> TokenPair: ...
104
105     def is_system_url(self, url: str) -> bool: ...
106
107     async def fetch_chunks_by_url(self, url: str) -> list[Chunk]: ...
108
109
110     class EmbeddingService(Protocol):
111         async def embed_chunks(self, chunks: list[Chunk]) -> list[EmbeddedChunk]: ...
112
113         async def embed_query(self, semantic_query: str) -> Embedding: ...
114
115
116     class DatabaseService(Protocol):
117         def index_chunks(self, chunks: list[EmbeddedChunk]) -> None: ...
118
119         def delete_all_chunks_for_source(self, source: SourceReference) -> None: ...
120
121         def query_chunks(
122             self, query: Query, embedding: Embedding, filters: DatabaseFilters, count: int, skip: int = 0
123         ) -> list[RelevantChunkMetadata]: ...
124
125         def get_stored_source_version(self, source: SourceReference) -> str | None: ...
126
127
128     class SearchService(Protocol):
129         async def query_chunks(self, query_request: QueryRequest) -> QueryResponse: ...
130
131         async def get_chunks_by_url(self, get_request: GetChunksByUrlRequest) -> GetChunksByUrlResponse: ...
132
133
```

#2

Keep Things Simple

Your most valuable resource is going to be time.
You're not going to be able to do everything.

Time Breakdown



#3

Remember the Business Context

Business doesn't get lost in the development cycle. Don't spend all your time fixing bugs that users aren't paying attention to.



What customer problem is this solving?

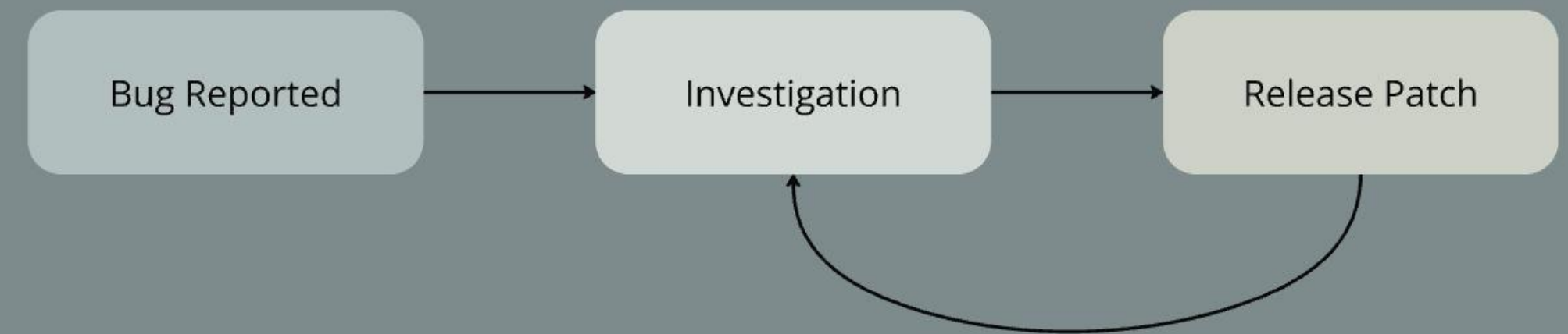
A clear and concise description of how this task will benefit end users.

e.g. AI App Devs want to be able to search for documents in their customer's confluence instances.

What needs to be done?

A clear and concise description of what need to be done.

1. New fetcher to fetch data from Confluence
2. New query filter to filtering workspace





Thanks!



Robert Koch, Software Engineer
@kochie