ARCHITECTURE DESIGN DOCUMENTATION

Rohan Padhye and **Michael Hilton**

ADMINISTRIVIA

HW2 deadline TODAY Oct 7th

Midterm Tuesday Oct 12th

NO RECITATION Oct 13/15

October 14th No CLASS – (Mid semester Break)

LEARNING GOALS

Articulate the various purposes of a design document.

Use design documentation to ensure that the correct thing is being implemented.

Write useful, clear, high-quality design documentation.

DESIGN DOCS

COMMON PARTS/TEMPLATES

Overview/feature description: what problem is being solved?

High-level requirements, both functional and quality

Background/key terms

Goals/non goals

Design alternatives, tradeoffs, assumptions

Decision

Other considerations/elements of design

EXAMPLES: SOURCEGRAPH RFCS

https://about.sourcegraph.com/handbook/communication/rfcs

Sourcegraph

Great code search helps you
write, reference, and fix, faster.

Over 800,000 developers use Sourcegraph to:

- Find anything in multiple repositories, fast
- Navigate with definitions and references
- Make large-scale code changes
- Integrate code with other services

Learn more about Sourcegraph →

WHEN TO USE AN RFC:

Sourcegraph

You want to frame a problem and propose a solution.

You want thoughtful feedback from team members on our globally-distributed remote team.

You want to surface an idea, tension, or feedback.

You want to define a project or design brief to drive project collaboration.

You need to surface and communicate around a highly cross-functional decision with our formal decision-making process.

institute for SOFTWARE RESEARCH | **Carnegie Mellon University** School of Computer Science

DON'T USE AN RFC WHEN:

Sourcegraph

You want to discuss personal or sensitive topics one-on-one with another team member.

You want to [make a decision](#) to change something where you are the decider. In the vast majority of cases, creating an RFC to explain yourself will be overkill. RFCs should only be used if a decision explicitly requires one of the bullets in the previous page.

institute for SOFTWARE RESEARCH

Carnegie Mellon University
School of Computer Science

RFC LABELS

WIP: The author is still drafting the RFC and it's not ready for review.

Review: The Review label is used when the RFC is ready for comments and feedback.

Approved: When the RFC is for the purpose of making a decision, the Approved label indicates that the decision has been made.

Implemented: When the RFC is for the purpose of making a decision, the Implemented label indicates that the RFC's proposal has been implemented.

Closed: When the RFC is for the purpose of collaboration or discussion but not necessarily to make a decision or propose a specific outcome that will eventually become Implemented, the Closed label indicates that the RFC is no longer an active collaborative artifact.

Abandoned: When the RFC is for the purpose of making a decision, and there are no plans to move forward with the RFC's proposal, the Abandoned label indicates that the RFC has been purposefully set aside.

OBSERVE SOURCEGRAPH DESIGN DOCS

Docs are publicly available:

https://drive.google.com/drive/folders/1zP3FxdDlcSQGC1qvM9lHZRaHH4I9Jwwa

Some Examples:

RFC37: https://docs.google.com/document/d/1CXDUqE6qUKrduT8TFgWxWC58IUrrK4eiL08c3fKzHyc/edit

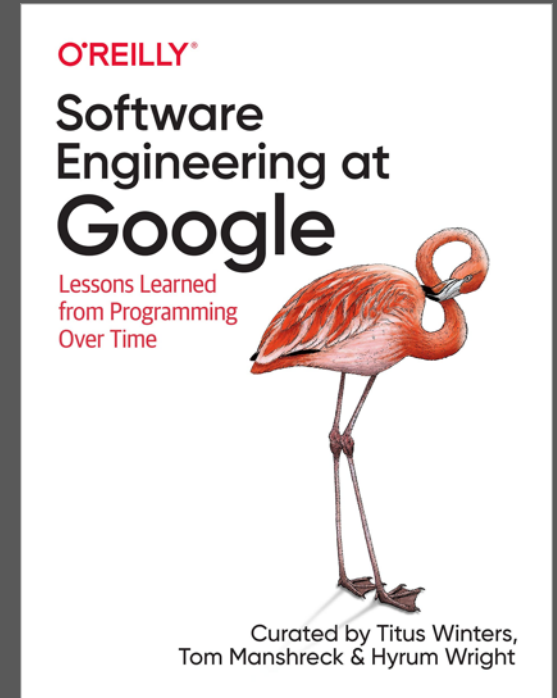RFC443: https://docs.google.com/document/d/1xXlN6eXBGLWA1px5rgfG6cLUJ-23h1-02KEJv_QpnwE/edit

RFC411: https://docs.google.com/document/d/1kyx3PIee2EYbAKeKLT92CPqDnOZLU9v7WE9G_zJy7_o/edit

RFC 390: https://docs.google.com/document/d/1KjV9pNkwCwqzV5ugW6Bn0eQDZw2L8IVuWaaf7IvVYgQ/edit

RFC27: https://docs.google.com/document/d/1ym5c8G5JcrFf5s0QXJqQKBcZRzjVAZZsxWdcUJ7Ukfw/edit#heading=h.sxjamvl1uyzk

DESIGN DOCUMENTS

Best design docs suggest design goals, and cover alternative designs, documenting the strengths and weaknesses of each.

BEST DESIGN DOC PRACTICES

Code review before there is code
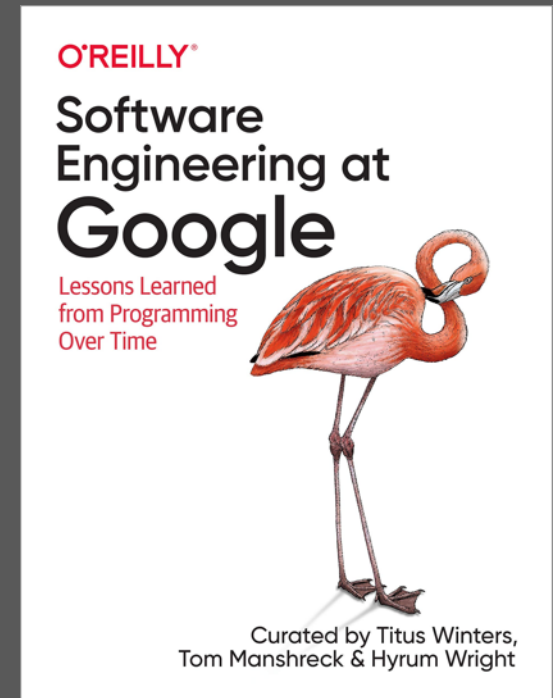
Collaborative (Google Docs)

Ensure various concerns are covered, such as: security implications, internationalization, storage requirements, and privacy concerns.

A good design doc should cover:

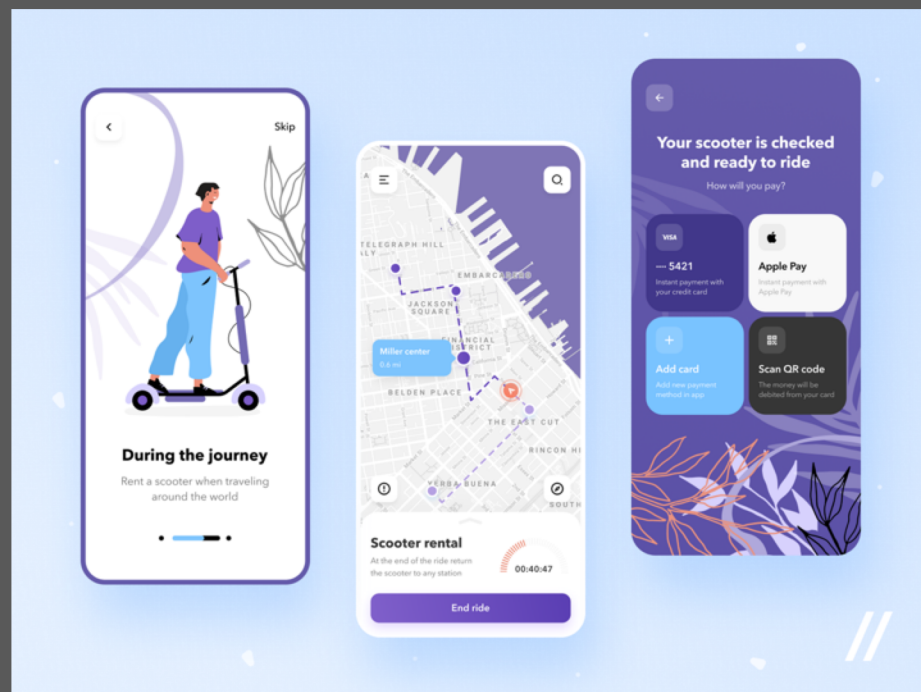> Goals of the design

> Implementation strategy

> propose key design decisions with an emphasis on their individual tradeoffs

O'REILLY®

Software Engineering at Google

Lessons Learned from Programming Over Time

Curated by Titus Winters, Tom Manshreck & Hyrum Wright

institute for SOFTWARE RESEARCH | Carnegie Mellon University School of Computer Science

EXERCISE: BRAINSTORM FEATURE IDEAS

Assume an MVP of a scooter app

What features might we propose?

EXERCISE:

Class is divided in 4 quadrants. Work on one design doc with the other students in your quadrant.

Work in groups of 2 or 3.

Put your AndrewID in "contributors"

REVIEW

We will now review the design docs as a class.

institute for SOFTWARE RESEARCH | **Carnegie Mellon University** School of Computer Science