Systems Design North Star

seththompson@, esprehn@, Jan 2017

Goal

The goal of the Systems Design program is to ensure that the **Chromium codebase is architected to support a simple, efficient, and high quality implementation of the web platform** that can fluidly evolve with the demands of the greater web ecosystem. This vision includes:

- Providing high-level design feedback on large architectural changes
- Disseminating implementation knowledge across a large, distributed team
- Researching comparative analyses of other platform implementations
- Coordinating cross-cutting architectural changes
- Facilitating discussion of systems design issues with the right stakeholders
- Building tools and frameworks to make implementation of features easier
- Tracking cross-cutting refactoring efforts or system-wide projects
- Guiding the evolution of the Chrome codebase by divining its future shape

The Systems Design program is not a gatekeeper of Chrome's architecture, but rather a consultant / evangelist / enabler of a healthy codebase and forward-looking engineering practices.

How we plan to get there

Research

Research our engine and others (both web and other systems like iOS/Android) and produce documents for knowledge spreading, as well as documents about ideas for our system to evolve.

- Produce comparative analyses of other platforms / systems
- Study history of Chromium contributions, processes, and codebase changes
- Track metrics and measurements to understand how Chromium evolves and how development velocity shifts over time (see metrics section below)

Identify

Identify systems in the engine (ex. Paint and Layout) that could be improved by changing the way they interact.

- Track new features (intent-to-implement) and major architectural changes
- Review design docs across Blink / Chromium
- Develop an understanding of how Chromium changes over time
- Find overlap between problem spaces, projects, and feature implementations
- Find similar functionality that could be refactored into shared components

Simplify

Simplify our implementation through refactoring projects and design guidance.

- Kick off refactoring projects such as Blink Onion Soup
- Advocate for simplification of Chromium through deduplication, servicification, and frameworkification
- Identify areas of Chromium which are ripe for rethinking or removal

Enable

Enable folks to rapidly evolve the code base and web platform with libraries, documentation, and tracking of projects and backlogs (aka air-traffic control).

- Provide libraries to enable teams to efficiently implement new Blink features
- Implement opinionated frameworks to simplify bootstrapping new Chromium services
- Host meetings, summits, and convergences to bring together the right stakeholders to solve cross-cutting architectural problems
- Maintain a backlog of Systems Design projects, problems, or ideas
- Perform high-level developer outreach and internal team outreach to keep up with an evolving web ecosystem

Example projects / successful outcomes

- Unifying the bindings system between PDFium and Blink
- Rethinking the overlap and shared functionality of the paint system between web and browser
- Building platform frameworks to bootstrap implementations of new features on top of an opinionated set of libraries

How will we measure that we're doing a good job?

- By monitoring the existence of a well-established and well-documented process for guiding the design and implementation of cross-cutting architectural changes, including:
    - A backlog of Systems Design projects, problems, or ideas
    - A regular meeting to review implementation statuses and discuss prioritization of the backlog
    - Systems Design program is able to read and understand 90% of new Chromium design docs
- By approximating the degree to which teams are successful at delivering large architectural changes in a timely manner
    - Qualitatively: if you ask any one person "is it easy to change our implementation of the web" their answer should be yes
    - Quantitatively: explore indirect measures of implementation efficiency and architectural simplicity by observing how often commonly modified directories need to change, how quickly new features get spun up, and how large CLs are for cross-cutting changes
- By surveying Chromium engineers about the helpfulness of the Systems Design program