

The Journey to STRUDEL: How We Came to Embrace User Experience in Scientific Ecosystems

February 21, 2024

Lavanya Ramakrishnan

 <https://strudel.science>

 <https://ux.lbl.gov>

Lawrence Berkeley National Lab
LRamakrishnan@lbl.gov

Team



**Lavanya
Ramakrishnan**
lramakrishnan@lbl.gov



Dan Gunter
dkgunter@lbl.gov



Sarah Poon
sspoon@lbl.gov



**Rajshree
Deshmukh**
rajshreed@lbl.gov

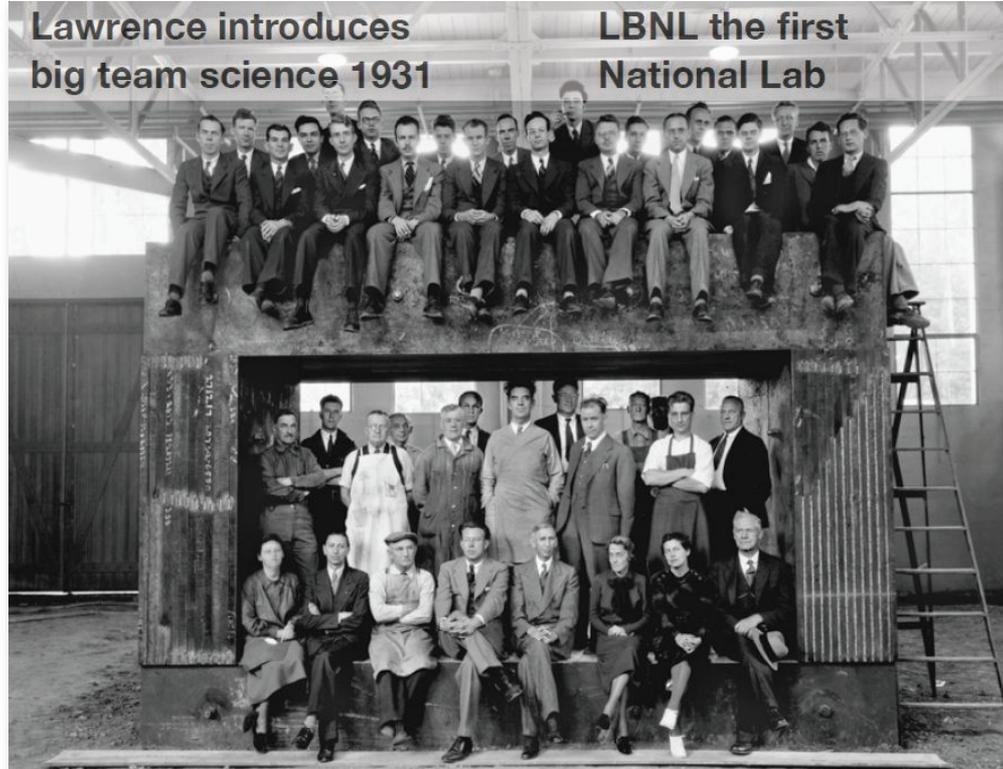


Cody O'Donnell
ctodonnell@lbl.gov

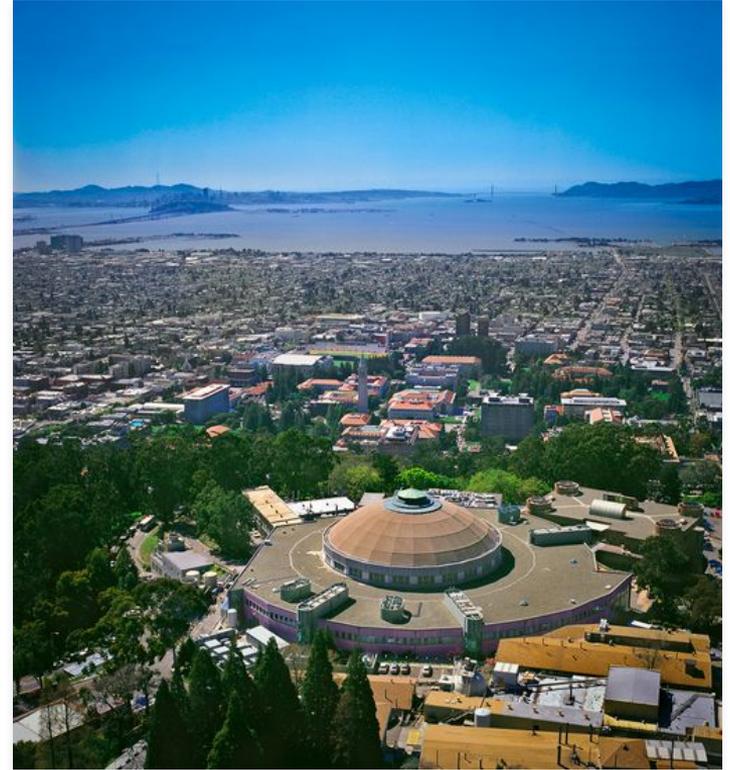


Drew Paine
pained@lbl.gov

Team science is at the core of what we do at Berkeley Lab



from LBNL image archive



Workflows: How do we enable researchers to effectively and efficiently manage their computation and data?



ESnet



Molecular
Foundry



ALS



JGI



NERSC



ESS-DIVE



AmeriFlux
Network



NMDC



KBase

Workflow management

- data abstractions
- HPC and distributed
- resource management
- autonomous pipelines
- reproducibility

Data management

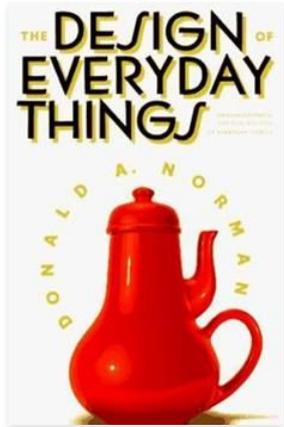
- search through AI-driven metadata extraction
- data change
- provenance

Why user experience (UX) matters for scientific software

How our team views UX for scientific software development

How these experiences lead to STRUDEL as a way to provide open source tools to help teams build more usable scientific software

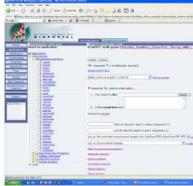
How did I get here ...



~2001

North Carolina Bioportal

- **Features**
 - access to common bioinformatics tools
 - extensible toolkit and infrastructure
 - OGCE and National Middleware Initiative (NMI)
 - leverages emerging international standards
 - remotely accessible or locally deployable
 - packaged and distributed with documentation
- **National reach and community**
 - TeraGrid deployment
 - scheduled for summer 2005
- **Education and training**
 - hands-on workshops across North Carolina
 - clusters, Grids, portals and bioinformatics



~2005



2012

Menti poll — Question 1

Mentimeter

In three or less words, what is your role in a scientific software process or team?

User experience designer	UX research & design	UX strategy	user researcher
Design, outreach, writing	Communication coordinator	research scientist / product owner	Software project manager

👍 👤

Mentimeter

In three or less words, what is your role in a scientific software process or team?

Design & UXR Lead	Ux designer	PI
-------------------	-------------	----

👍 👤

Menti poll — Question 2

Mentimeter

What does user experience mean to you?

User first	designing a tool for end users	Easy and useful	Improving the accessibility and reproducibility of research code and data through user interfaces
Satisfying experience doing the job that needs to be done	focusing on usability	Skating to the puck	understanding a user and building an interface that matches their needs

👍 👤

Mentimeter

What does user experience mean to you?

Focusing on how users will solve problems with your tool	How the user thinks about your product and uses it to their benefit	That's what I worked for... good experiences create a nicer world in general	thinking actively about the design of a tool to make something that fits user needs
--	---	--	---

👍 👤

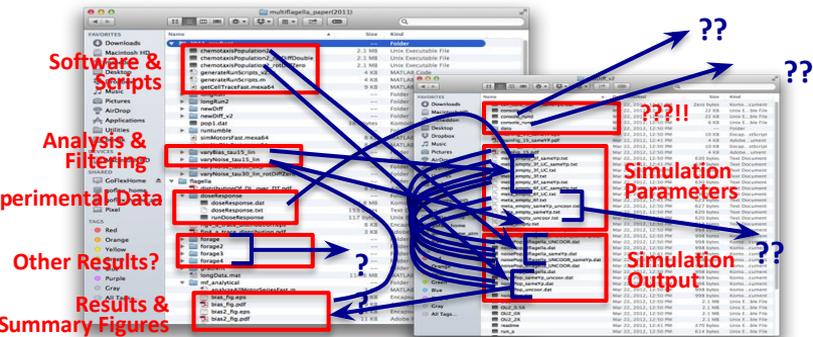
Why is building usable scientific software challenging?

Realities of scientific work

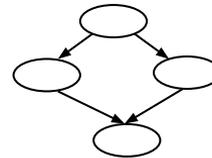
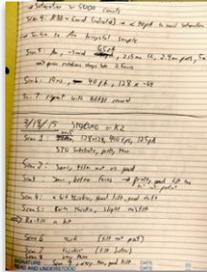
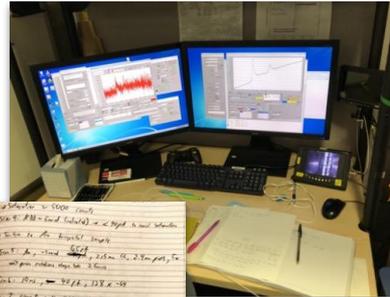
Don't fit into nice graphs

Supporting artifacts and context are not captured

Collaborations have complex software stacks



Courtesy: Paramvir Dehal, KBase team



Courtesy: DESI project

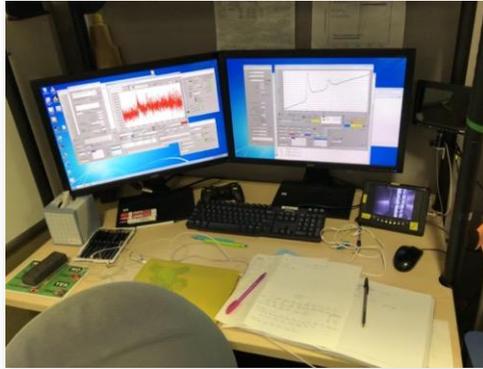
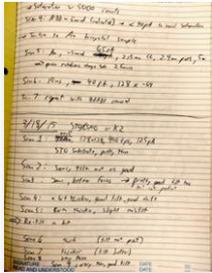


Source: Ameriflux project

New work practices that don't fit into current work process will likely not get adopted.



How we see UX in scientific software development

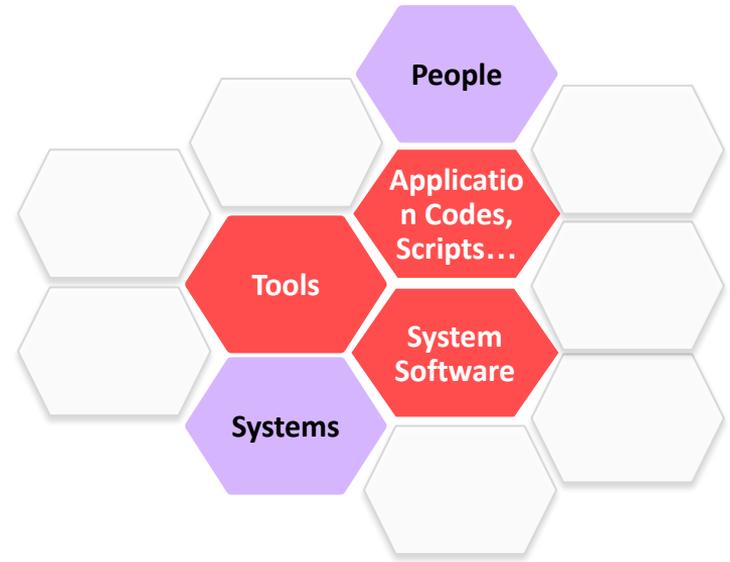
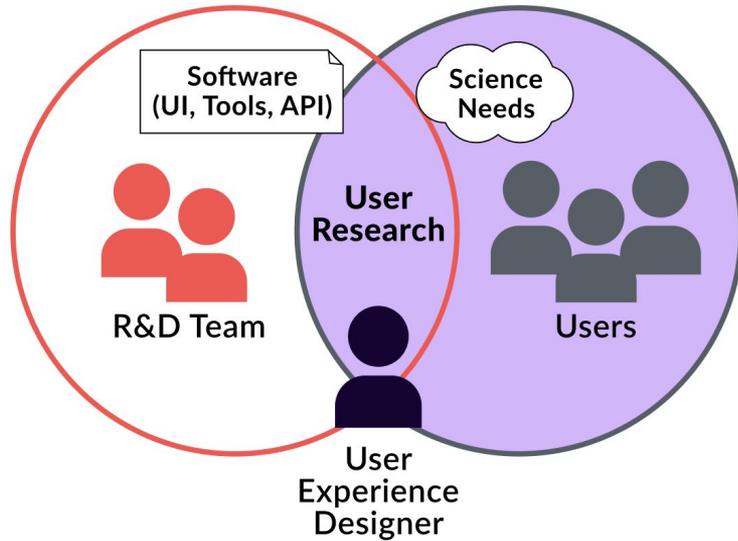


Scientific software project involve art as much as science

Just like pastry making... such as strudel

STRUDEL enables teams to create user-centered software for scientific communities. Plan, design, and build better scientific software projects using STRUDEL Planning Framework and Design System.

Our UX approach to addressing challenges in scientific workflows



User research gives you a **process to verify/validate your “intuition about what the user needs” (hypothesis) and convert into action**

User research processes can significantly improve the research and software outcomes

Discover Explore

- Interviews
- Contextual Inquiry, observations
- Competitive Analysis

Synthesis

- Journey Maps
- Scenarios
- Design Constraints or Considerations

Design

- Wireframes
- Detailed Mockups
- Prototypes

Usability Tests

- Interfaces, APIs

- Increased Productivity For End Users
- Decreased Development Costs and Time
- Increased Adoption
- Better and/or Lesser Documentation and Training
- Fewer Errors/Bugs, Lower Costs

How do we define User Experience (UX)?

User experience (UX) is the **practice** of developing services & products that provide ***consistent, relevant, productive, & joyful*** experiences for users.

Misconception: UX is purely focused on graphical user interfaces.

Best Practice: UX practices are employed to shape *everything* from internal organizational processes to all varieties of user interfaces (UIs) & interactions among systems & users.

Ten Principles for Creating Usable Scientific Systems

1. Solve the right problem first
2. Understand user motivations
3. Understand the context of use
4. Validate and verify what you have heard
5. Test before building; test after building
6. Clean interfaces can't make up for bad design
7. Build for the right user (i.e., computer engineers vs scientists)
8. Understand the user's metrics (usually not performance)
9. Cost/benefit for the science team is different from the development team
10. Be willing to iterate (early and often)



#1 Source: Dula Parkinson



#2 Source: Ameriflux project

Planning, design & stewardship of scientific software often *tumultuous, even chaotic*

Individuals often fulfill roles that are varied, multifaceted

Never enough resources (time, \$\$ people)

Management & planning can be ad hoc responding to emerging scientific demands and needs

UX often an afterthought at best

Uncommon is an industry-like *Product Management* role who stewards vision, user engagement, etc.

STRUDEL builds on our experiences incorporating UX in many scientific projects

Providing UX as consultants, typically design or some usability research



IDAES
Institute for the Design of
Advanced Energy Systems



ESS-DIVE
Deep Insight for Earth Science Data



KBase
PREDICTIVE BIOLOGY

Incorporating UX as key part of our R&D Projects

Deduce



SCIRA



Science
Capsule

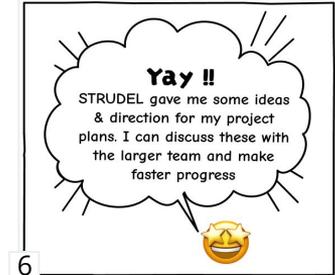
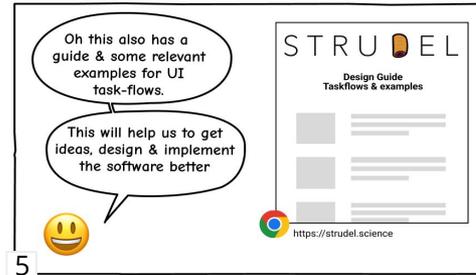
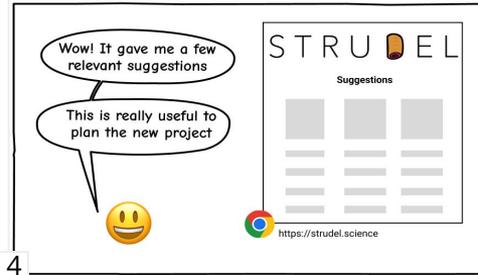
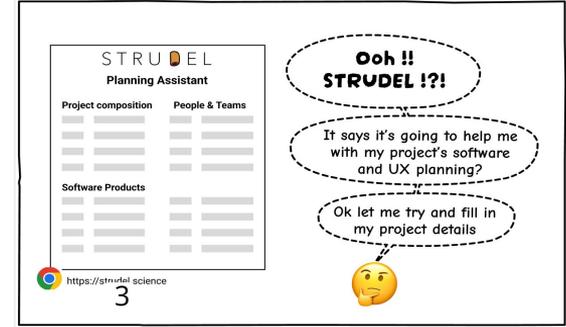
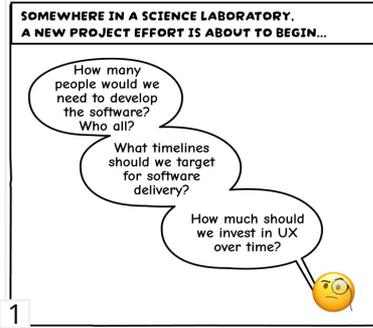


Systematically expanding
& abstracting insights from
this repeated work

STRUDEL

The long-term STRUDEL vision

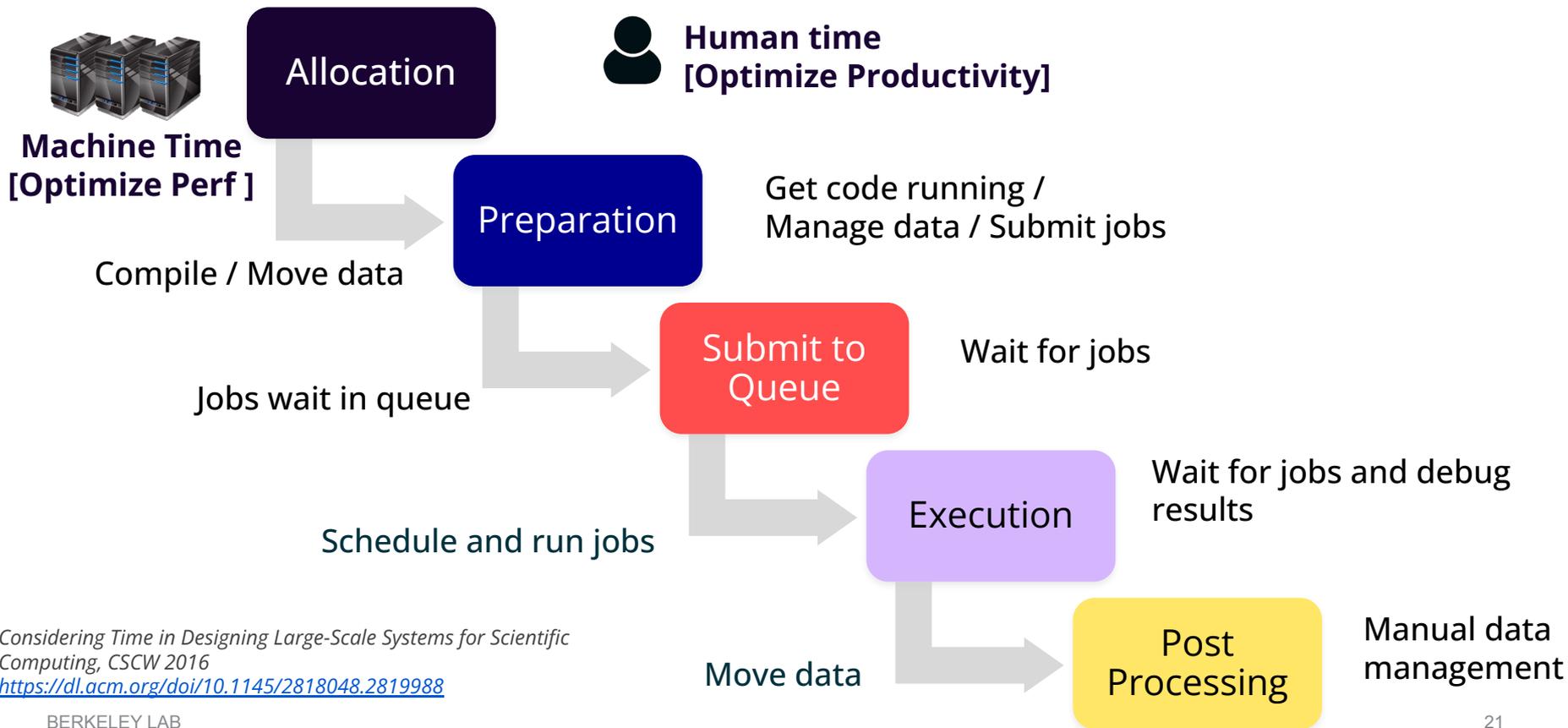
Our aim is to develop products that help scientific software teams simplify adoption of UX approaches to enable more usable, sustainable software.



Our Experiences

- **HPC user's perception of computing time**
- **Challenges adapting Jupyter to HPC environments**
- **Leveraging UX research to improve R&D process**

Time is a key factor in our optimization strategy ...



Considering Time in Designing Large-Scale Systems for Scientific Computing, CSCW 2016
<https://dl.acm.org/doi/10.1145/2818048.2819988>

UX research highlighted how incorporating open source software in HPC environments requires strategic adaptations

Qualitative UX research in 2019-2020 investigated experiences with Jupyter on NERSC HPC systems

UXR surfaced *joyful* and *frustrating* user experiences, showed challenges & opportunities HPC environments face incorporating common open source tools



😊 Streamlined JupyterLab setup makes accessing HPC resources easier & users happier

😄 Adaptations facility provided for pre-configured Jupyter kernels & python environments made for productive experience

📁 Customized JupyterLab file system browser was small but significant improvement for users

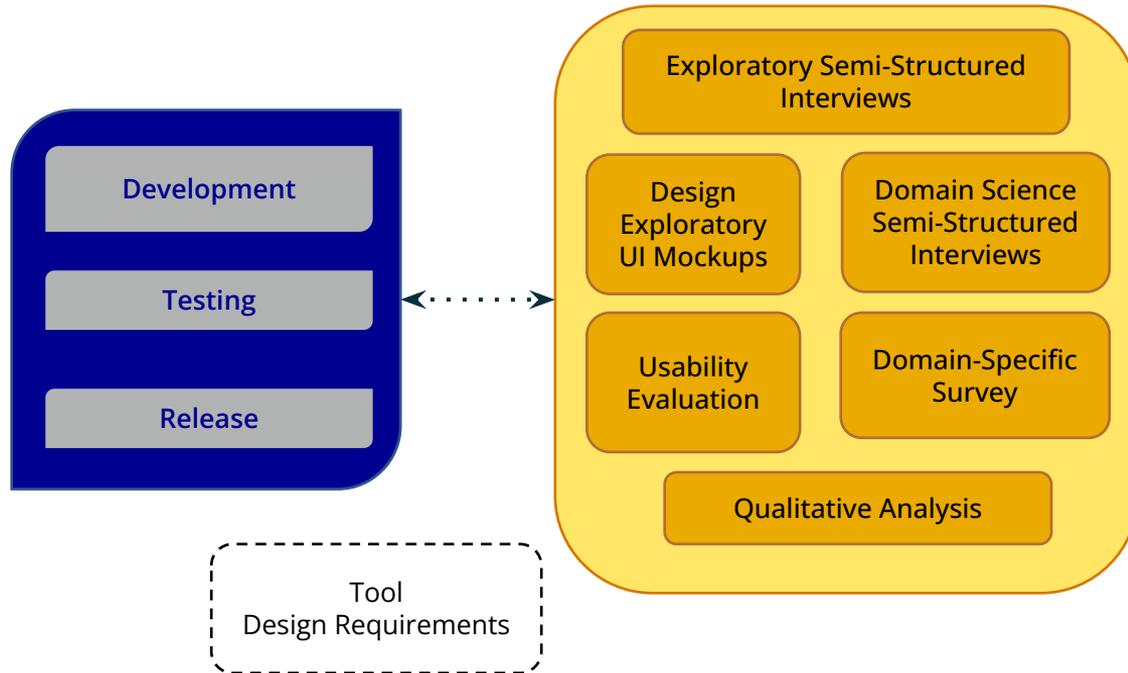
😡 Facility maintenance windows induce frustration

😞 Customization of a shared Jupyter instance is tricky

😞 Real time collaboration not simple or easy to accomplish ←

Follow on R&D work tackled these challenge!

User research methods can weave closely with the R&D process to produce better results for the project and users.



Menti poll — Question 3

Mentimeter

What are your challenges in planning, designing, and developing scientific software?

dev team jumps to building without asking questions	Aligning with project timelines, adjusting to shifting timelines	Conflicting team priorities	having collaborators understand how long it takes to do user research
moving quickly and responsively to user needs	Demands for doing novel research and no direct funding for producing good software	People love to add ad hoc ideas that end up take a lot of time to develop and sometimes discarded	I'm an RSE and when I collaborate with researchers I like to use Notebooks because they are easier for them to inherit and maintain the interface

👍 👤

Mentimeter

What are your challenges in planning, designing, and developing scientific software?

technical debt in design space limits design choices without major refactoring	finding qualified people willing to do the work	Unclear and changing requirements	behavioral patterns change throughout the semester, making decisions difficult and increasing data collection time
Understand the issue			

👍 👤

Menti poll — Question 4

Mentimeter

What are your challenges in developing user interfaces (web and API) for scientific software?

understanding the skill sets needed to develop a tool

User callable libraries are not the typical product for ux

Complex user needs/workflows

Too many opinions from different perspectives

Some stakeholders wait well past design stage to provide feedback so much more expensive to fix

finding qualified people willing to do the work

different behaviors throughout the semester means long data collection time for UXR

Building UI / pipeline in parallel with research: before output data may exist

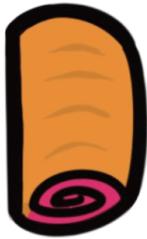


Mentimeter

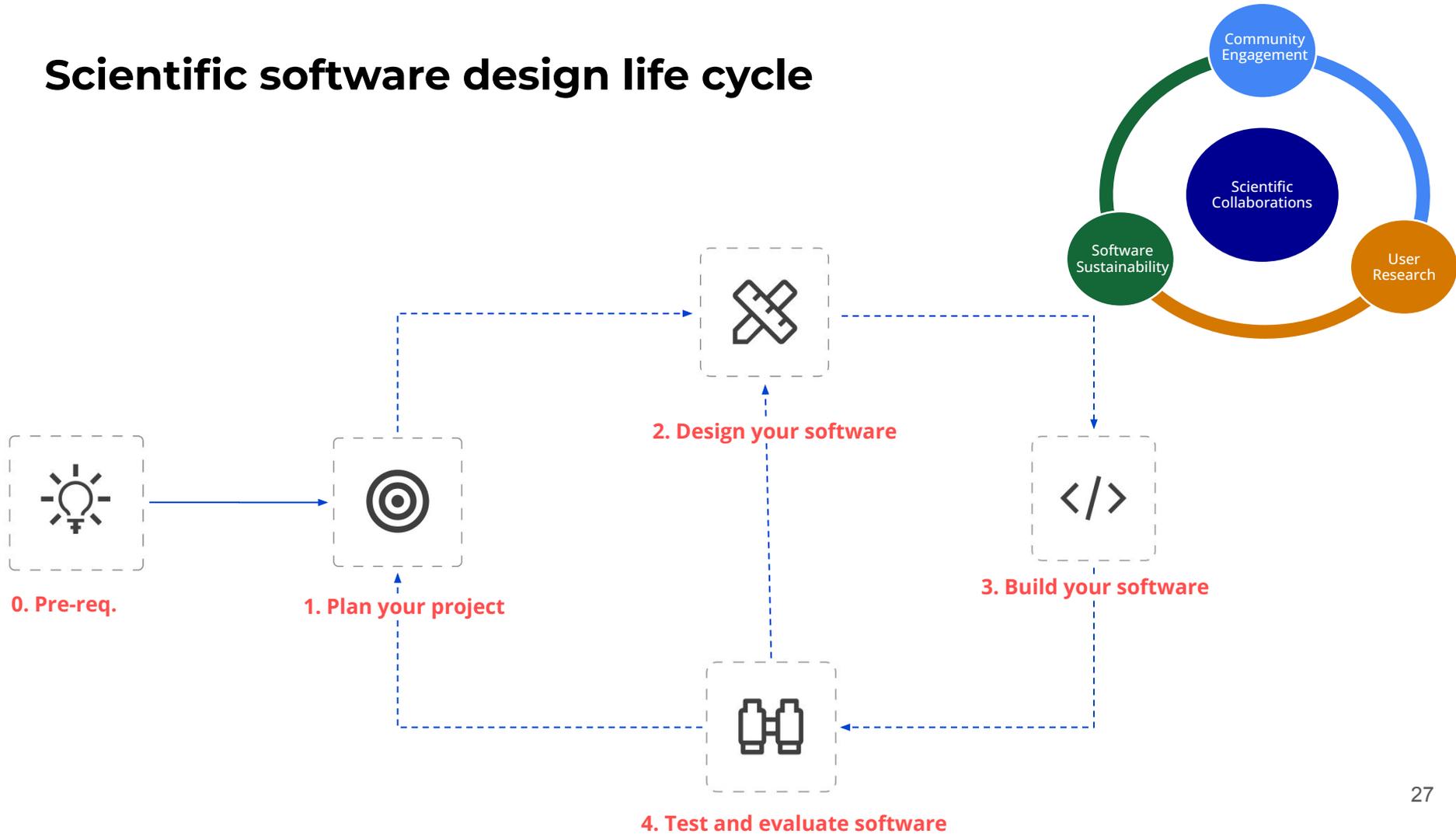
What are your challenges in developing user interfaces (web and API) for scientific software?

little control of underlying software stack at runtime.

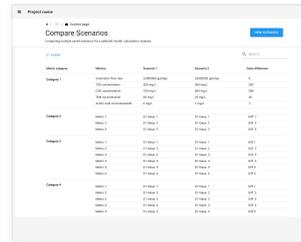
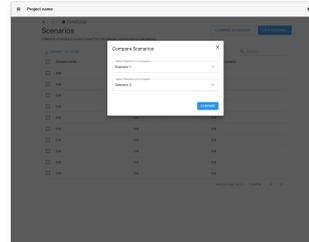
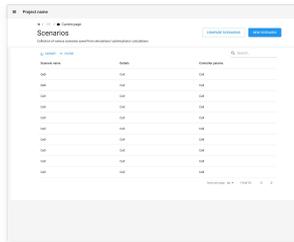
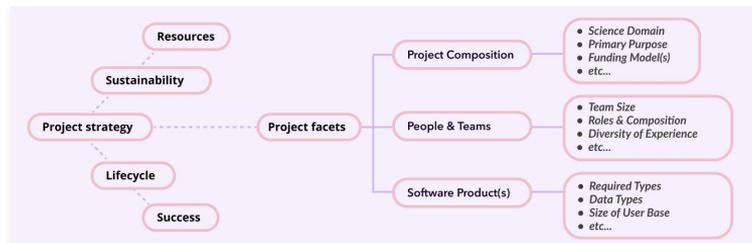


S T R U  E L

Scientific software design life cycle



STRUDEL: Open source project with two key products



Typology of Scientific Software

informing a strategic

Planning Framework

Design System

with

Task Flows

Supporting the scientific software development life cycle



0. Pre-reqs.



1. Plan your project



2. Design your software



3. Build your software



4. Test and evaluate software

Categorizing Patterns in Scientific (Software) Work

Relevant Stages



0. Pre-reqs.



1. Plan
your
project



2. Design
your
software

Today

Typology is a first attempt to categorize questions & concerns we have seen repeatedly across projects, environments, etc.

Tomorrow

Crafting a strategic **Planning Framework** from this categorization & resources to enable better project planning & software design

Primary facets of typology & example application

Project Composition

People & Teams

Software Product(s)

Project X in biology

(selected example dimensions)

- **Science Domain:** Supports a single domain, Biology
- **Primary Purpose:** Data repository/service
- **Funding Model(s):** DOE base funding for data repository
- **Team Size:** Large
- **Roles & Composition:** Domain Scientists, SE, Admins
- **Diversity of Experience:** Students, Postdocs, Early Career, Seniors - Biology, CS, Data Science backgrounds
- **Required Types:** Web App, API
- **Data Types:** Experimental, Observational
- **Size of User Base:** Large
- **Computing Paradigms:** Personal, Cluster HPC

Design system

A design system is a set of reusable components and patterns for designing and building UIs as well as guidelines on when and how to use them.

What is **unique** about the STRUDEL design system?

Designed specifically for scientific UIs.

Enables building UIs applicable across different scientific domains

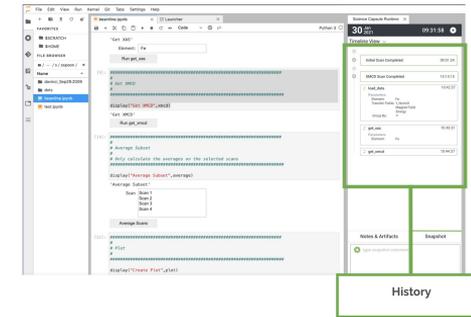
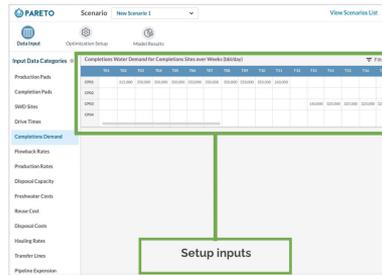
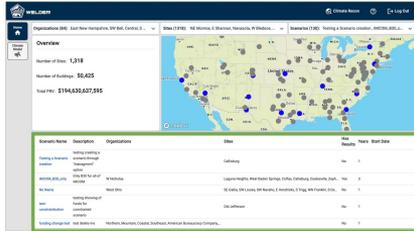
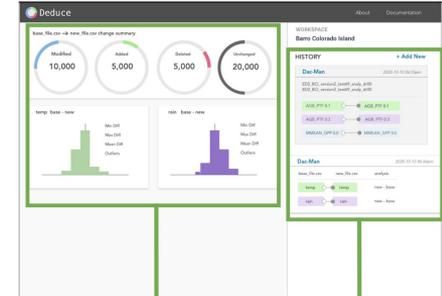
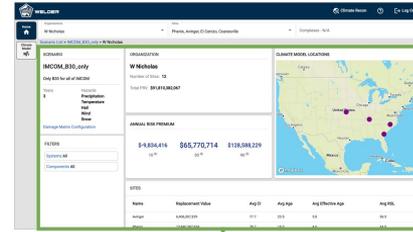
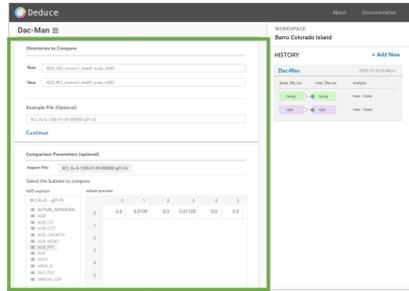
Focuses on the larger flow & function of UI

Gives you a jump start to think about entire UI flow rather than starting from scratch

Designed by experts for experts.

Informed by over a decade of collective UX experience in the sciences and democratizes good UX practices

Identifying Task Flows From Common UI Needs



Scenario Selection

Select Inputs

Dashboard summary of results

History

Task Flows

Task Flow: series of steps represented by screens which helps user to accomplish particular task in the scientific software's user interface

Similar Task Flows exist across various types of scientific software.

Relevant Stages



2. Design
your
software



3. Build
your
software

Analysis

Run Computation

Run Interactive
Computation

Compare Data

Data

Explore Data

Explore Data Repositories

Contribute Data

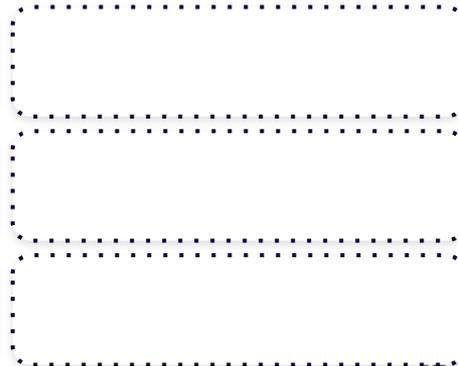
Exploration

Monitor Activity

Track State

Manage Account

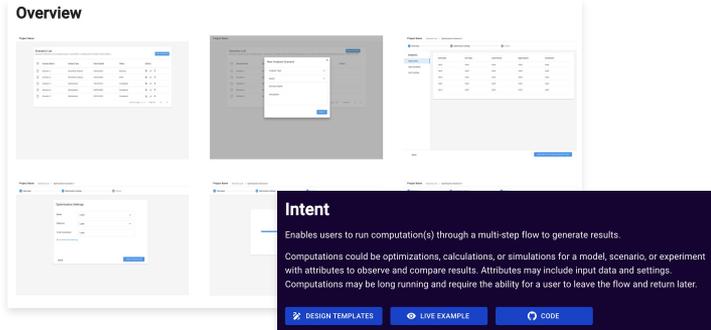
Community Contributions



Task Flow Resources

Design templates & guidelines for the series of steps involved in the Task Flow.

These templates are available as images and as design files on **Figma community** for customizing designs.



Intent

Enables users to run computation(s) through a multi-step flow to generate results.

Computations could be optimizations, calculations, or simulations for a model, scenario, or experiment with attributes to observe and compare results. Attributes may include input data and settings. Computations may be long running and require the ability for a user to leave the flow and return later.

[DESIGN TEMPLATES](#) [LIVE EXAMPLE](#) [CODE](#)

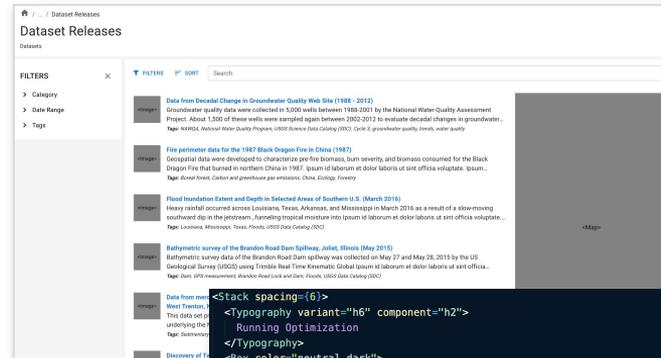
Guidelines for adapting the Task Flow

- Break the task flow into multiple workable steps and use a progress indicator / stepper to help users see the progress and remaining steps in the process to complete.
- Organize information into sections that are easy to digest. This helps improve the readability and searchability.
- Offer guidance, tips, and links to detailed documentation for complex inputs & interactions.
- Pre-fill the forms with sensible default values wherever possible, especially if data inputs require long forms.
- Consider allowing users to upload input data as external files or spreadsheets, especially for computations that require large amounts of input data.
- Make attributes searchable and filterable to make it easy to find attributes of interests.

strudel-kit

Web interactive templates and coded UI library for high level components & task flows from our design system.

Uses **React javascript framework** and is built on top of the popular Material UI (MUI) components library



```
<Stack spacing={6}>
  <Typography variant="h6" component="h2">
    Running Optimization
  </Typography>
  <Box color="neutral.dark">
    <Typography>This could take several minutes.</Typography>
    <Typography>You may leave this page and return later. Your progress will not be affected.</Typography>
  </Box>
  <LinearProgress variant="determinate" value={70} sx={{ height: 10 }} />
  <Typography color="neutral.dark">
    Started 05/24/2023 12:32:33
  </Typography>
</Stack>
```

Using STRUDEL Design System example

Relevant Stages



2. Design
your
software

Science need: A UI to run optimizations on models

1. Select relevant Task Flow from STRUDEL:



Run Computation

Enables users to run computation(s) through a multi-step flow to generate results.

execution

scenario

job

simulation

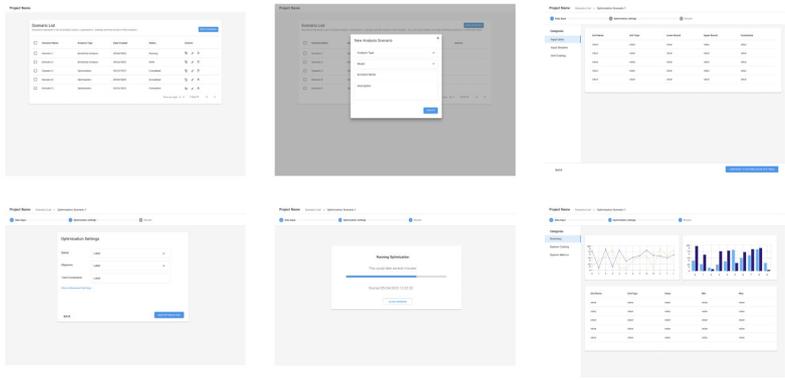
optimization

stepper

wizard

2. See details:

Overview



Intent

Enables users to run computation(s) through a multi-step flow to generate results.

Computations could be optimizations, calculations, or simulations for a model, scenario, or experiment with attributes to observe and compare results. Attributes may include input data and settings. Computations may be long running and require the ability for a user to leave the flow and return later.

[DESIGN TEMPLATES](#)

[LIVE EXAMPLE](#)

[CODE](#)

Guidelines for adapting the Task Flow

- Break the task flow into multiple workable steps and use a progress indicator / stepper to help users see the progress and remaining steps in the process to complete.
- Organize information into sections that are easy to digest. This helps improve the readability and searchability.
- Offer guidance, tips, and links to detailed documentation for complex inputs & interactions.
- Pre-fill the forms with sensible default values wherever possible, especially if data inputs require long forms.
- Consider allowing users to upload input data as external files or spreadsheets, especially for computations that require large amounts of input data.
- Make attributes searchable and filterable to make it easy to find attributes of interests.

Using STRUDEL Design System example

3. Understand selected Task Flow, map to requirements and plan customizations

'Run Computation' Task Flow template screens:

Task Flow Step 1: Initiate new computation from list view of scenarios/ models

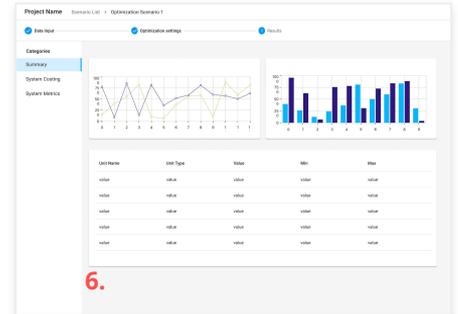
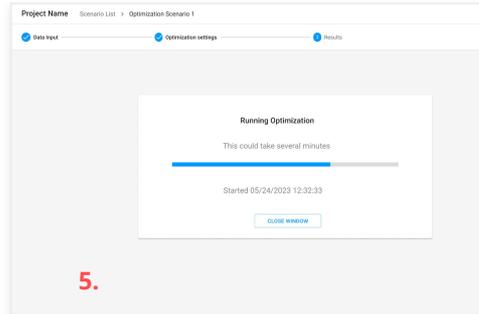
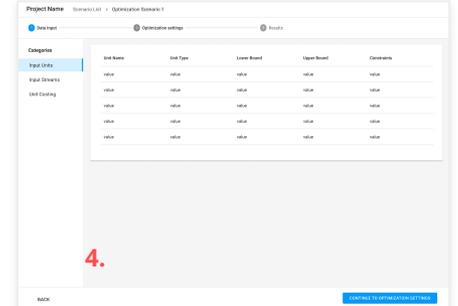
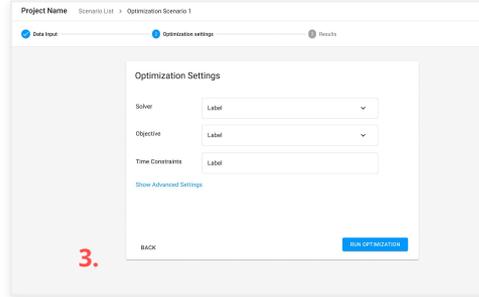
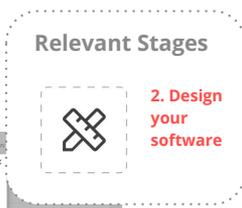
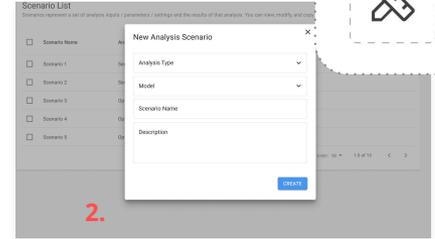
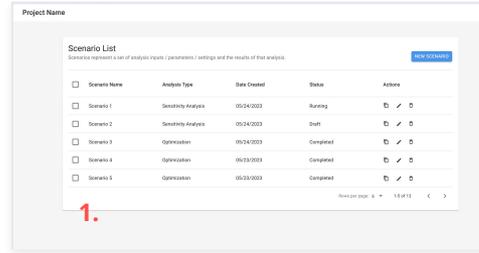
Task Flow Step 2: New optimization/ analysis meta data input form

Task Flow Step 3: Input optimization parameters

Task Flow Step 4: Select optimization settings

Task Flow Step 5: Progress view for long process optimization

Task Flow Step 6: Optimization results view



Using STRUDEL Design System example

Relevant Stages

</>

3. Build
your
software

4. Browse prototype & code implementation guide from STRUDEL-kit UI Library

The screenshot shows a web application interface for an optimization scenario. The top navigation bar includes 'Project name', 'Scenario List', and 'Optimization Scenario 1'. Below this, there are three tabs: 'Data Inputs', 'Optimization Settings', and 'Results'. The 'Data Inputs' tab is active, displaying a table with columns for 'Unit Name', 'Unit Type', 'Constraints', 'Lower Bound', and 'Upper Bound'. The table contains five rows of data, all with 'value' in the 'Unit Type' and 'Constraints' columns, and '0' in the 'Lower Bound' and '1' in the 'Upper Bound' columns. A 'CONTINUE TO OPTIMIZATION SETTINGS' button is located at the bottom right of the table area.

Unit Name	Unit Type	Constraints	Lower Bound	Upper Bound
value	value	value	0	1
value	value	value	0	1
value	value	value	0	1
value	value	value	0	1
value	value	value	0	1

5. Create base app and start integrating your customizations, data and APIs

The screenshot shows the 'User Quickstart' section of the STRUDEL documentation. It includes a 'Prerequisites' section stating that Node.js and NPM must be installed, with a terminal command to check their versions. A 'Get Started' section provides instructions on installing the STRUDEL CLI tool using pip, followed by a terminal command to create a base app. A warning icon indicates that the CLI is currently only available on TestPyPI. The bottom part of the screenshot shows a code snippet for a component, likely a progress indicator, using HTML-like tags for styling and layout.

User Quickstart

Prerequisites

Node.js and NPM must be installed to run the web applications you generate with strudel-cli. To check if you already have Node.js and NPM installed, open a terminal and run:

```
node --version  
npm --version
```

If both commands return a version number, you should be good to go. If not, you can download both tools together here: <https://nodejs.org/en/download>

Get Started

Install the STRUDEL CLI tool:

```
pip install -i https://test.pypi.org/simple/ strudel-cli
```

! strudel-cli is only on TestPyPI for the moment. When it is published to PyPI, you will be able to omit the -i option

Create a base app:

```
strudel create-app my-app
```

```
<Stack spacing={6}>  
<Typography variant="h6" component="h2">  
  Running Optimization  
</Typography>  
<Box color="neutral.dark">  
<Typography>This could take several minutes.</Typography>  
<Typography>You may leave this page and return later. Your progress will not be affected.</Typography>  
</Box>  
<LinearProgress variant="determinate" value={70} sx={{ height: 10 }} />  
<Typography color="neutral.dark">  
  Started 05/24/2023 12:32:33  
</Typography>  
</Stack>
```

Get Involved!

Join the STRUDEL Community

Next events in the STRUDEL series

- **Getting Started with the STRUDEL Design System**
 - Virtual webinar on Zoom
 - Friday March 8th, 11am Pacific
- **STRUDEL Design System Hackathon**
 - In person @ Berkeley Lab
 - Tuesday March 19, All Day
 - *Space is limited!*



Visit our website to learn more & use our products!



<https://strudel.science>



Have comments?

Start a conversation on our [GitHub](https://go.lbl.gov/strudel-discussion)
<https://go.lbl.gov/strudel-discussion>



Join the US-RSE User Experience working group to connect with the larger community of practice!



#wg-ux on the **US-RSE Slack**
<https://go.lbl.gov/usrse-uxwg>



Join our mailing list to keep up to date & contribute to the community!

strudel-community+subscribe@lbl.gov

Thank you!