

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

February 21, 2024

Lavanya Ramakrishnan

Lawrence Berkeley National Lab
LRamakrishnan@lbl.gov

- 🌐 <https://strudel.science>
- 🌐 <https://ux.lbl.gov>

Funded by Sloan Foundation, Liz Vu & Josh Greenberg Program Officers, Grants #10074 & [#10572](#)



ALFRED P. SLOAN
FOUNDATION



UXSciData
.lbl.gov



BERKELEY LAB
Bringing Science Solutions to the World

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

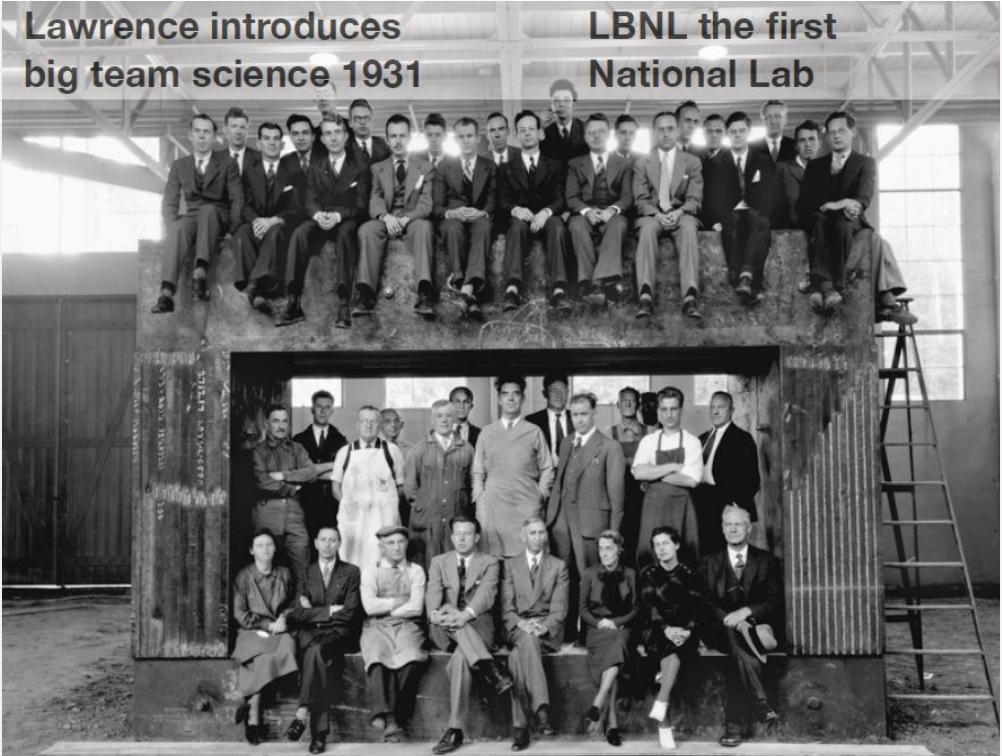


Drew Paine
pained@lbl.gov

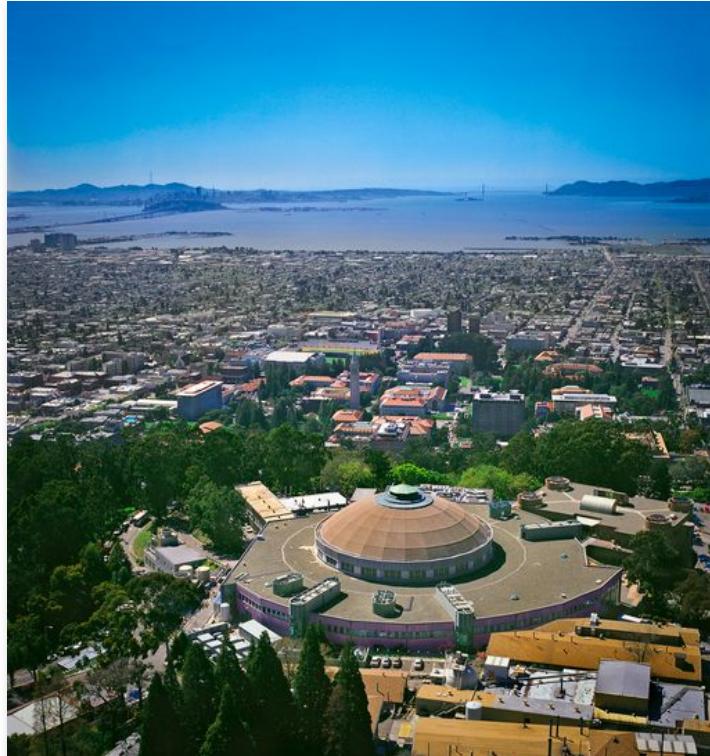
Team science is at the core of what we do at Berkeley Lab

Lawrence introduces
big team science 1931

LBNL the first
National 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

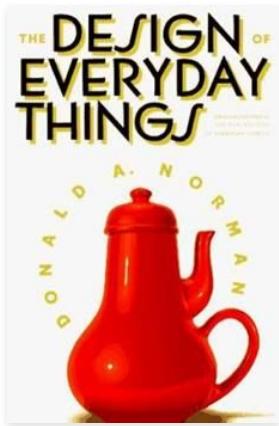
- search through AI-driven metadata extraction
- data change
- provenance

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

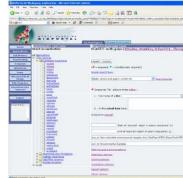
How did I get here ...



~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

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

This image shows a Menti poll interface. At the top, it asks "In three or less words, what is your role in a scientific software process or team?". Below this, there are four categories of roles: "User experience designer", "UX research & design", "UX strategy", and "user researcher" in the first row; and "Design, outreach, writing", "Communication coordinator", "research scientist / product owner", and "Software project manager" in the second row. Each category has a corresponding light gray box for input. At the bottom right of the poll area, there are three small circular icons: a thumbs up, a person, and a refresh symbol.

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

Design & UXR Lead Ux designer PI

This image shows a second Menti poll interface with the same question: "In three or less words, what is your role in a scientific software process or team?". It displays three roles: "Design & UXR Lead", "Ux designer", and "PI". Each role has a corresponding light gray box for input. At the bottom right of the poll area, there are three small circular icons: a thumbs up, a person, and a refresh symbol.

Menti poll — Question 2

What does user experience mean to you?

Mentimeter

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

Like 11

What does user experience mean to you?

Mentimeter

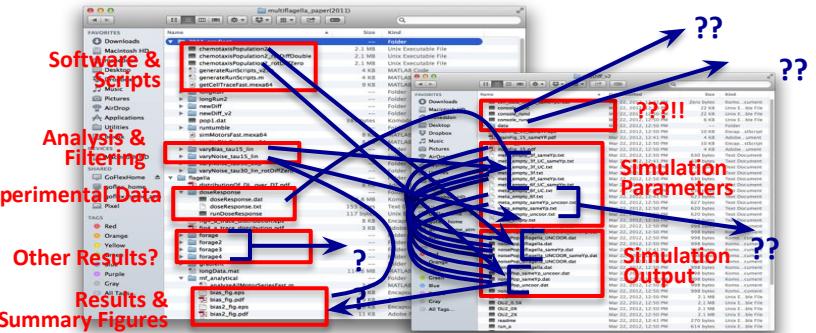
Focusing on how users will solve problems with your tool	How the user thinks about your product and uses it to their benefit	That's what I worked for.. good experiences create a nicer world in general	thinking actively about the design of a tool to make something that fits user needs
--	---	---	---

Like 11

Why is building usable scientific software challenging?

Realities of scientific work

Don't fit into nice graphs



Courtesy: Paramvir Dehal, KBase team

Supporting artifacts and context are not captured

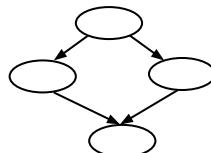


Courtesy: DESI project

Collaborations have complex software stacks



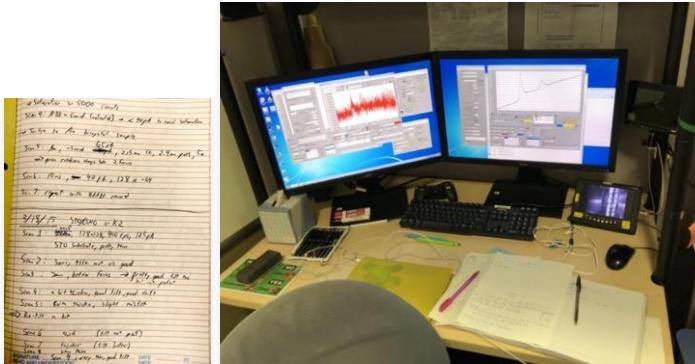
Source: Ameriflux project



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



How we see UX in scientific software development

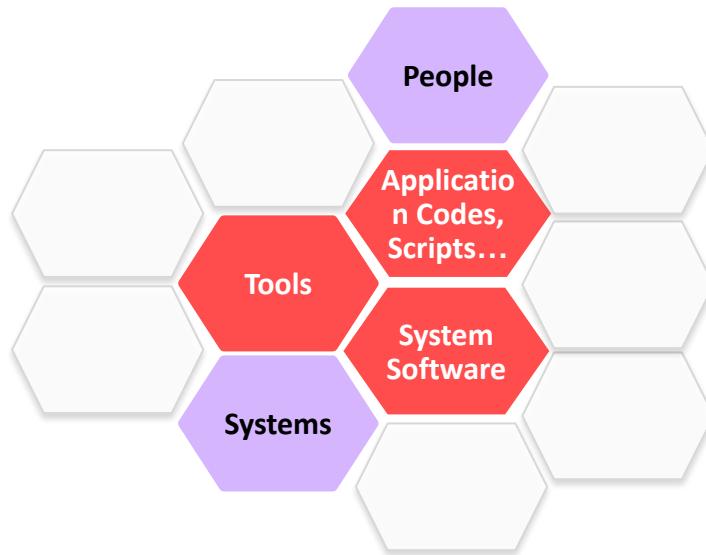
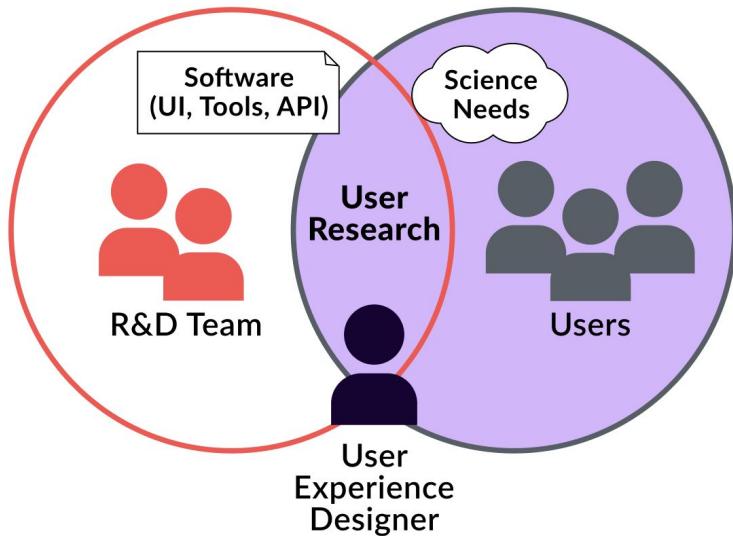


Scientific software project involve art as much as science

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



Incorporating UX as key part of our R&D Projects

Deduce



SCIRA

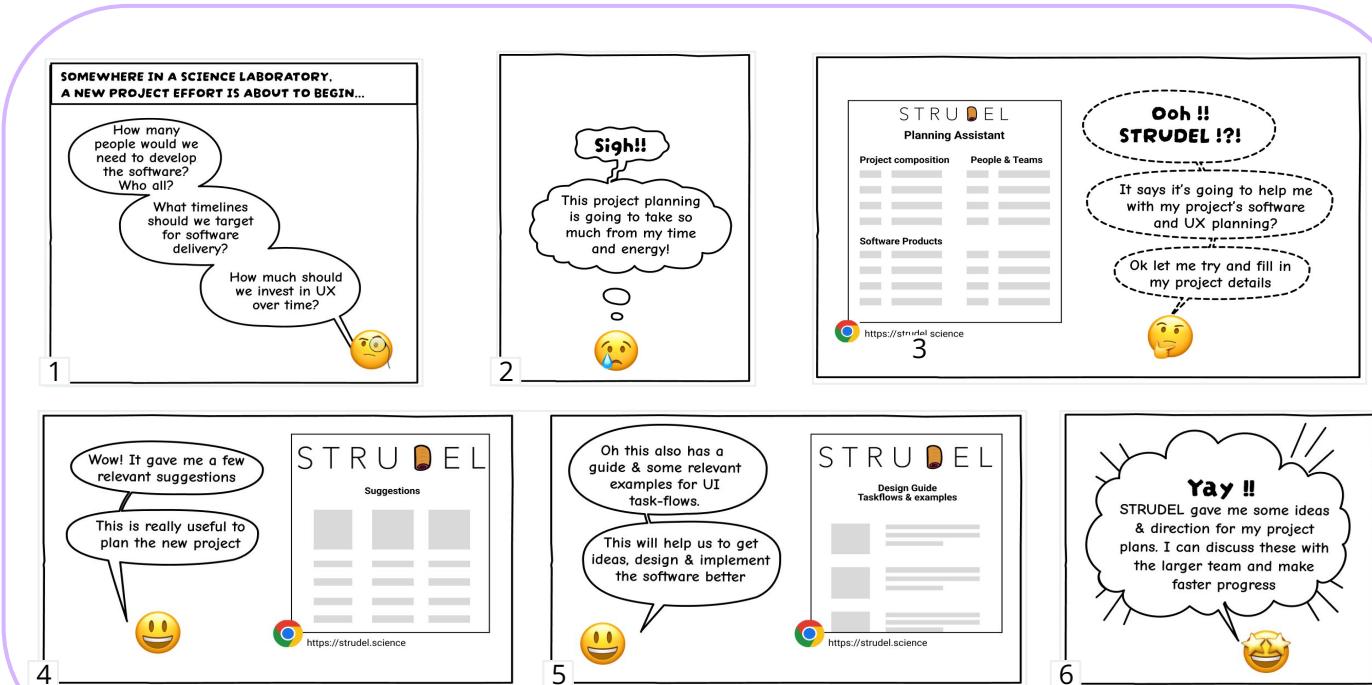


Systematically expanding & abstracting insights from this repeated work

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.



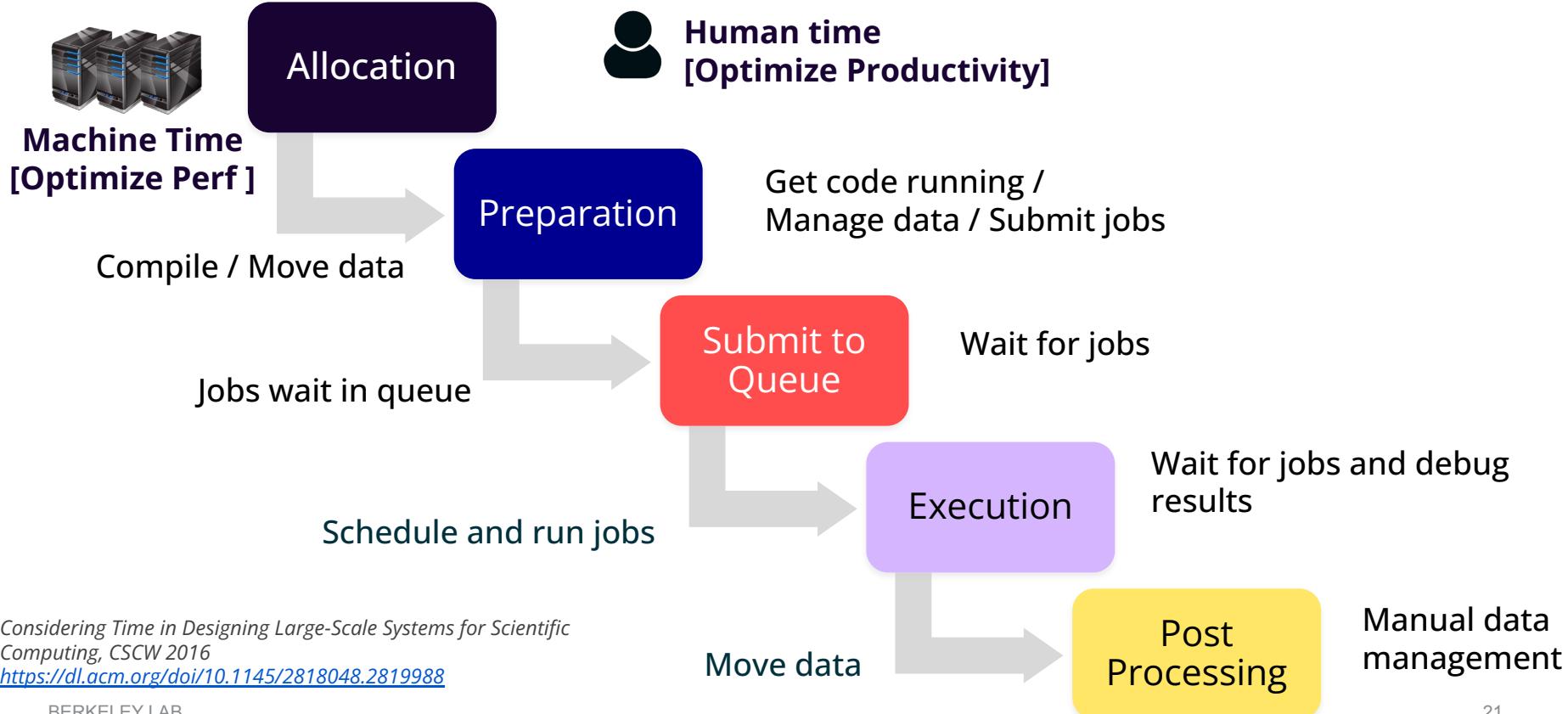
Our Experiences

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

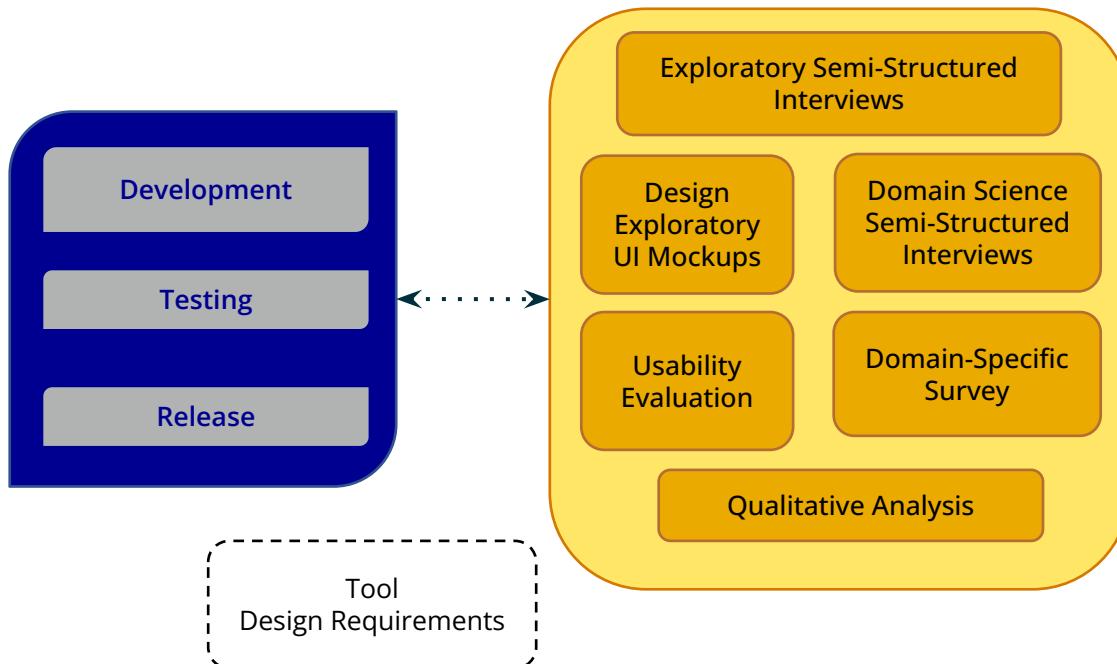
😡 Customization of a shared Jupyter instance is tricky

😩 Real time collaboration not simple ← or easy to accomplish



Follow on R&D work tackled these challenges!

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



Menti poll — Question 3

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



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

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

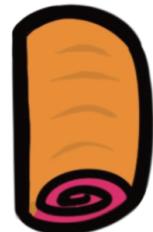
+ 0 -

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

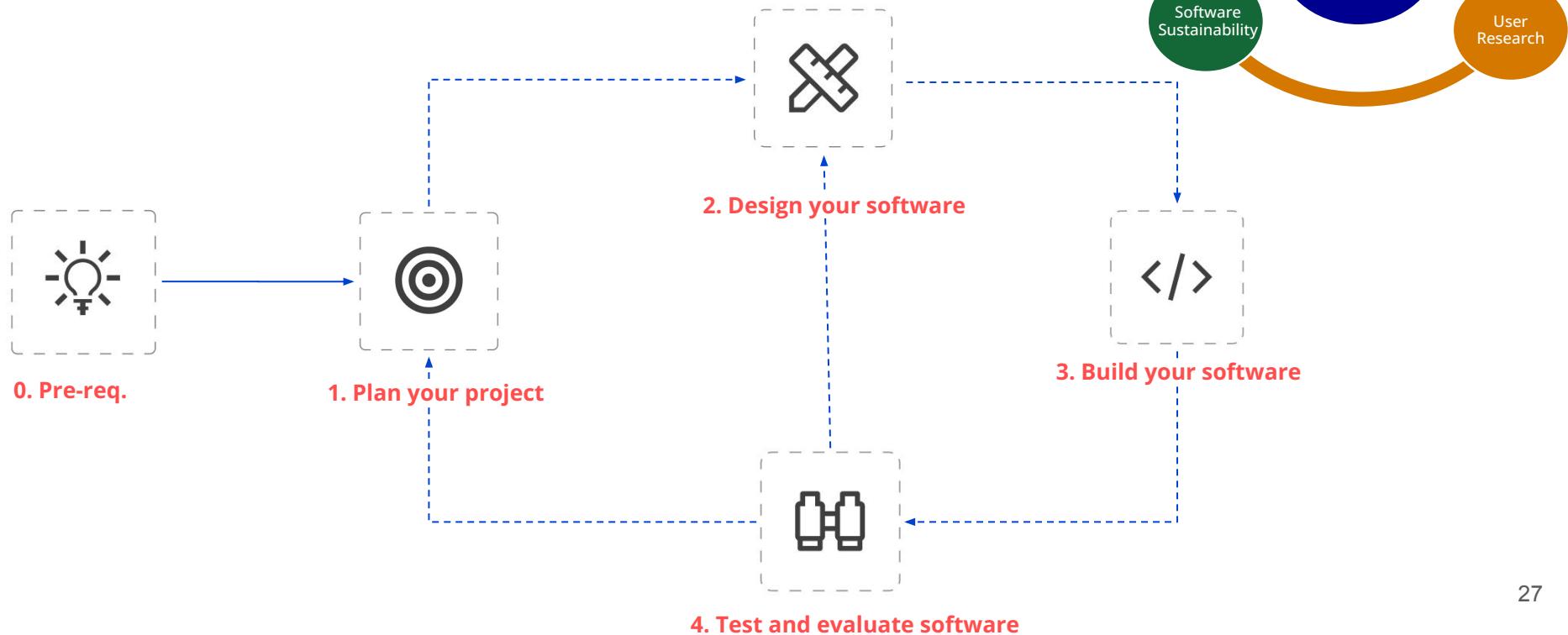
little control of underlying software stack at runtime.

+ 0 -

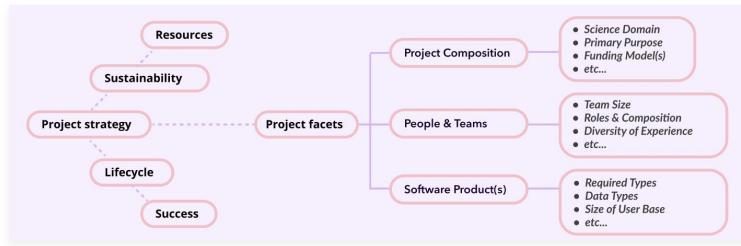
STRU E L



Scientific software design life cycle



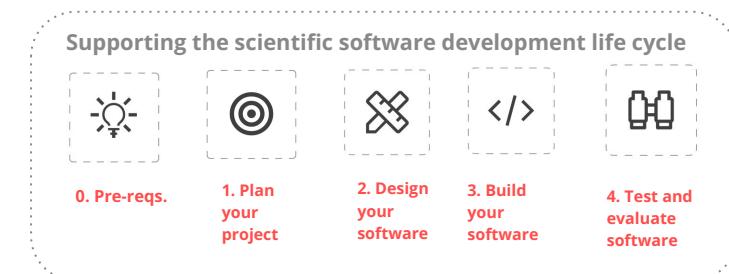
STRUDEL: Open source project with two key products



Three screenshots of the STRUDEL web application interface are shown side-by-side. The left screenshot shows the "Project facets" page with a table of data and a search bar. The middle screenshot shows the "Compare Scenarios" feature with a modal dialog. The right screenshot shows a detailed view of the "Scenarios" table, listing various scenarios across different categories.

Typology of Scientific Software
informing a strategic
Planning Framework

Design System
with
Task Flows



Categorizing Patterns in Scientific (Software) Work

Today

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

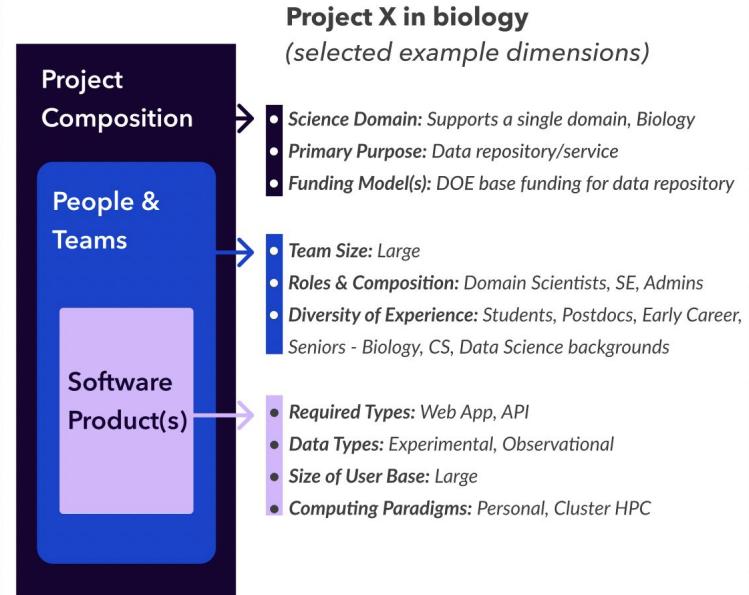
Tomorrow

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

Relevant Stages



Primary facets of typology & example application



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

Scenarios

Name	Date Created	Status	Actions
Case Study Scenario 1	09/14/2012	Optimized	
Case Study Scenario 2	09/16/2012	Draft	

Scenario selection

Deduce

Dac-Man III

Directories to Compare

Example File (Optional): `BCI-AUC-0000-00-000000-gt-12`

Comparison Parameters (optional):
Input File: `BCI-AUC-0000-00-000000-gt-12`

Setup inputs

WELD SITE

Number of Sites: 1,218
Number of Buildings: 55,425
Total PHS: \$194,635,677.55

Scenarios

Description	Organizations	Site	Max Results	Short Desc
Galburg	No	1		
Lagrange, West Baker Springs, Cawthron, Cawthron, Franklin, Daka, No	Yes	2	10 Galburg, Latta, No, Franklin, Cawthron, Franklin, Daka, No	
NO Galburg, Latta, No, Franklin, Cawthron, Franklin, Daka, No	No	1		
NO Jefferson	No	1		
NO Latta, No, Franklin, Cawthron, Franklin, Daka, No	No	1		
NOing change site, not data site	No	1		
Northwest, Mountain, Central, Southern, American, Business, Company,	No	1		

Scenario selection

PARETO

Scenario: New Scenario 1

View Scenario List

Input Data Categories

- Production Pads
- Completion Pads
- SWD Sites
- Drive Times
- Completion Demand
- Flowline Rates
- Production Rates
- Disposable Capacity
- Preheater Costs
- Reserve Cost
- Disposal Costs
- Heating Ratios
- Transfer Units
- Pipeline Expansion

Completion Water Demand for Completion Sites over Weeks (bbl/day)											
	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU
CWS	311,000	309,000	305,000	302,000	299,000	296,000	293,000	290,000	287,000	284,000	281,000
CWS	311,000	309,000	305,000	302,000	299,000	296,000	293,000	290,000	287,000	284,000	281,000

Setup inputs

Scenario Selection

Select Inputs

Barro Colorado Island

HISTORY

Dac-Man

ANIMAL RISK PREVIEW

SITES

Dashboard summary of scenario

PARETO

Scenario: New Scenario 1

View Scenario List

Results Details

93%
7,544,654 bbls
953,628 bbls

Water Deliveries By Destination

REVIEW INPUTS SETTINGS

Dashboard summary of scenario

Dashboard summary of results

Deduce

WORKSPACE: Barro Colorado Island

HISTORY

Dac-Man

Dashboard

History of runs

Timeline

Run Details

History

Notes & Artifacts

Snapshot

History

History



2. Design
your
software



3. Build
your
software

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.

Analysis

Data

Exploration

Community Contributions

Run Computation

Explore Data

Monitor Activity

Run Interactive
Computation

Explore Data Repositories

Track State

Compare Data

Contribute Data

Manage Account

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.

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 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](#)

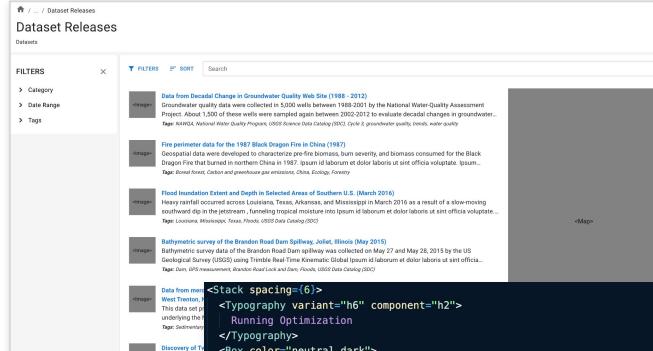
Guidelines for adapting the Task Flow

- 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:



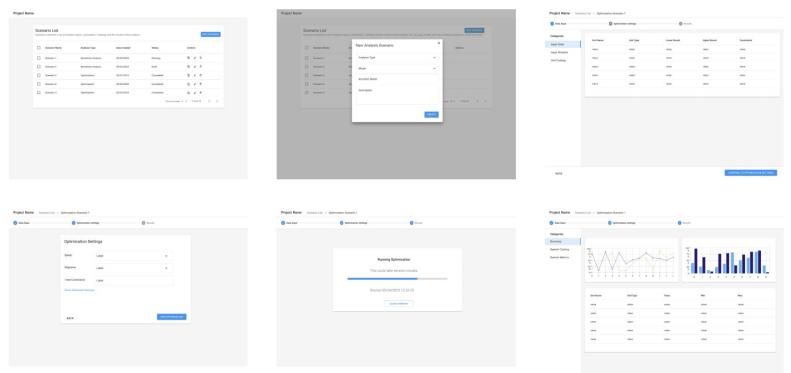
Run Computation

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

The screenshot displays the STRUDEL Design System interface across six numbered steps:

- Step 1:** Scenario List - A table showing five scenarios: Scenario 1 (Sensitivity Analysis, Draft), Scenario 2 (Sensitivity Analysis, Draft), Scenario 3 (Optimization, Compiled), Scenario 4 (Optimization, Compiled), and Scenario 5 (Optimization, Compiled). A 'New Scenario' button is visible.
- Step 2:** New Analysis Scenario dialog - A modal window for creating a new scenario. It includes fields for 'Analysis Type' (set to 'Model'), 'Scenario Name' (Scenario 6), and 'Description'. A 'CREATE' button is at the bottom.
- Step 3:** Optimization Settings form - A form for defining optimization parameters. It includes sections for 'Solver' (Label), 'Objective' (Label), and 'Time Constraints' (Label). A 'RUN OPTIMIZATION' button is at the bottom right.
- Step 4:** Input Data table - A table for input data. It has columns for 'Unit Name', 'Unit Type', 'Lower Bound', 'Upper Bound', and 'Constraints'. Rows show 'value' for all columns.
- Step 5:** Running Optimization progress bar - A progress bar indicating the status of an optimization process. It shows a blue bar at approximately 20% completion with the text 'Running Optimization' and 'This could take several minutes'. A timestamp 'Started 05/24/2023 12:32:33' and a 'CLOSE WINDOW' button are also present.
- Step 6:** Optimization Results view - A dashboard showing various metrics. It includes a summary table for 'System Coding' and 'System Metrics', and two line graphs for 'System Coding' and 'System Metrics'.

A vertical dashed line on the right side of the interface is labeled 'Relevant Stages' and '2. Design your software'.

Using STRUDEL Design System example

Relevant Stages



3. Build
your
software

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

Project name Scenario List > Optimization Scenario 1

Data Inputs Optimization Settings Results

Categories

Input Units

Input Streams

Unit Costing

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

Rows per page: 100 ▾ 1–5 of 5 < >

CONTINUE TO OPTIMIZATION SETTINGS

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

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!