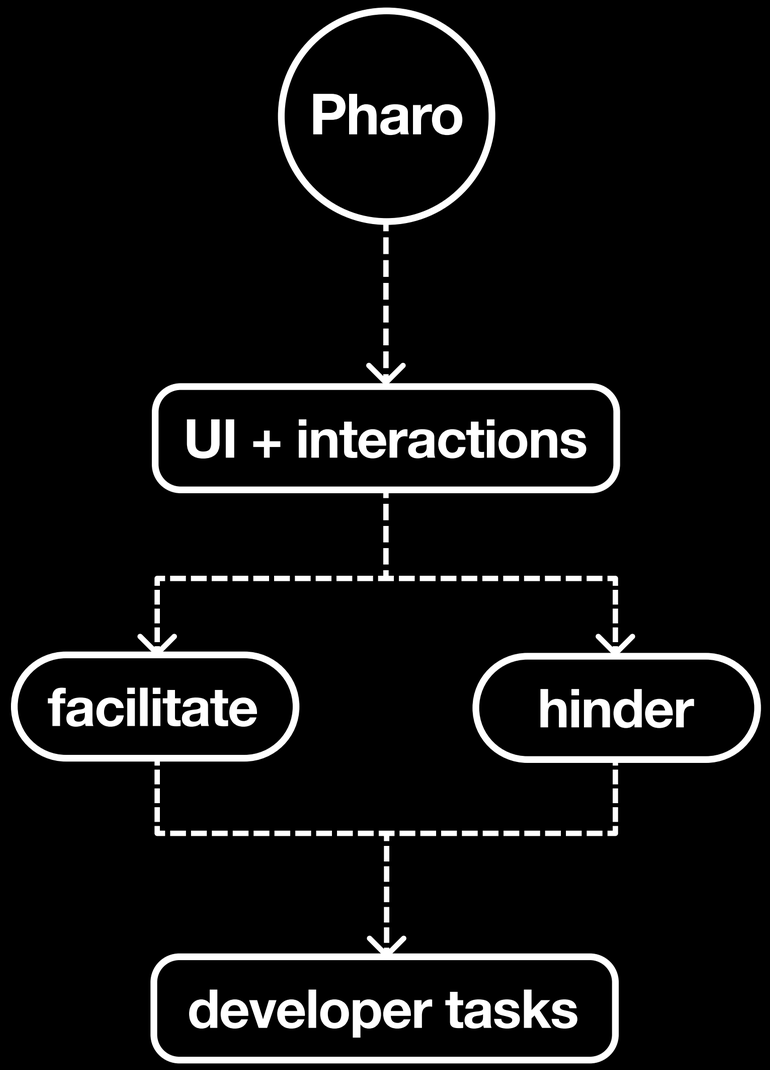
Pharo Usability Study

Thematic Analysis

Santiago Viana

Context

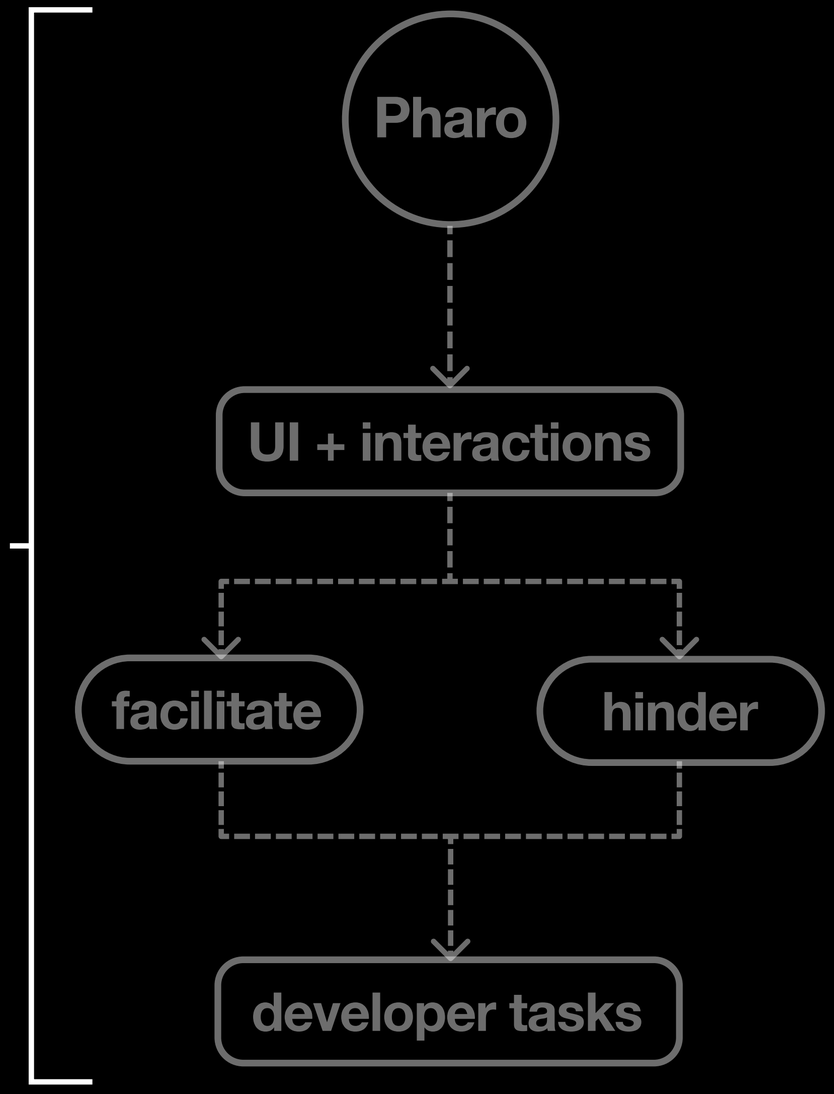
- Publication



Analysis

Interviews

8 Participants *(1-2 hrs. each)*



Methods

1

Observed the Videos

Familiarizing myself with the Data

2

Generated Transcriptions

Fixing mistakes along the way

3

Created the Codes

Double checking with the research team

4

Decide what to Code

Using Thematic analysis tools (Taguette)

5

Grouping Codes into Themes

(Themes are at a provisional stage)

Methods

1 **Observed the Videos**

Familiarizing myself with the Data

2 **Generated Transcriptions**

Fixing mistakes along the way

3 **Created the Codes**

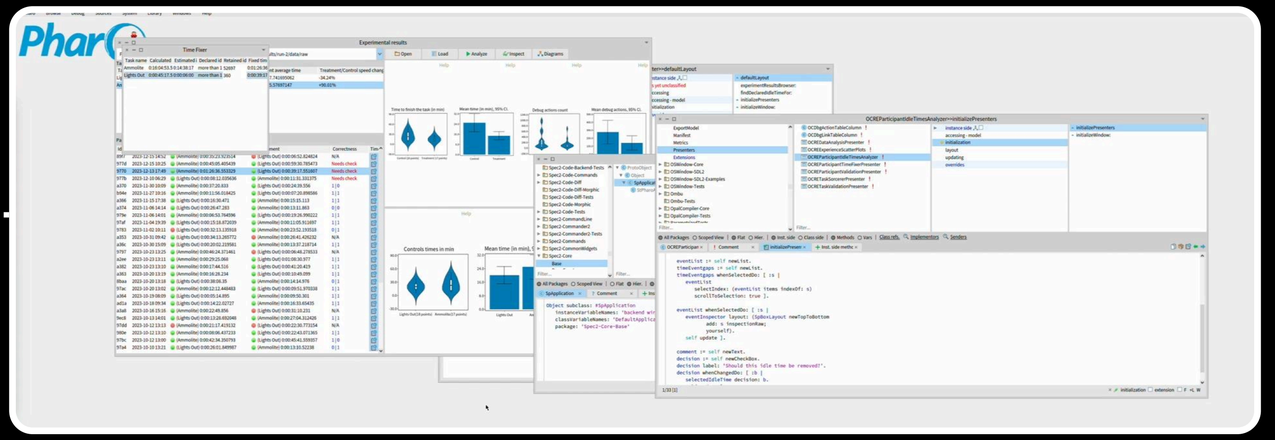
Double checking with the research team

4 **Decide what to Code**

Using Thematic analysis tools (Taguette)

5 **Grouping Codes into Themes**

(Themes are at a provisional stage)



UI + Interactions

Methods

1 **Observed the Videos**

Familiarizing myself with the Data

2 **Generated Transcriptions**

Fixing mistakes along the way

3 **Created the Codes**

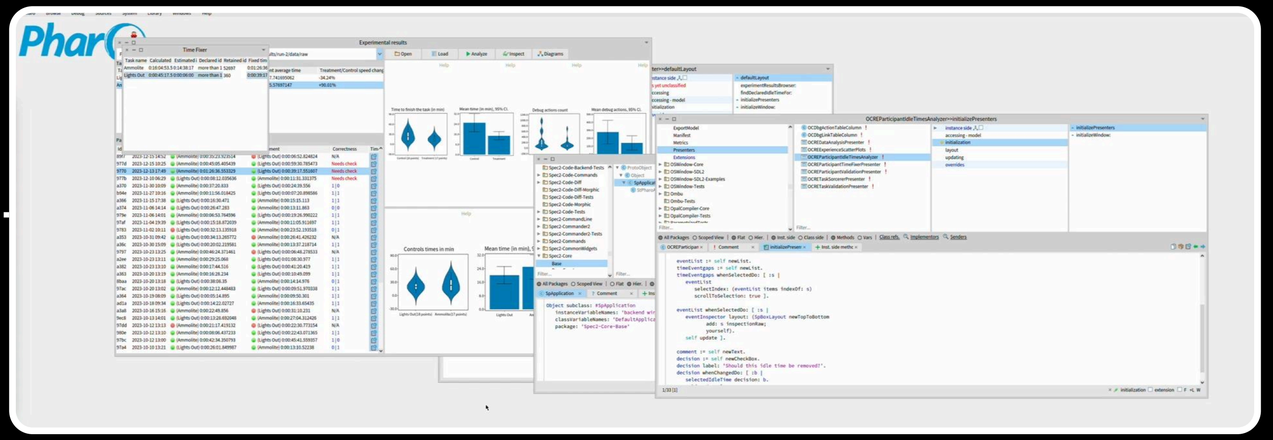
Double checking with the research team

4 **Decide what to Code**

Using Thematic analysis tools (Taguette)

5 **Grouping Codes into Themes**

(Themes are at a provisional stage)



Tasks

Methods

1 **Observed the Videos**

Familiarizing myself with the Data

2 **Generated Transcriptions**

Fixing mistakes along the way

3 **Created the Codes**

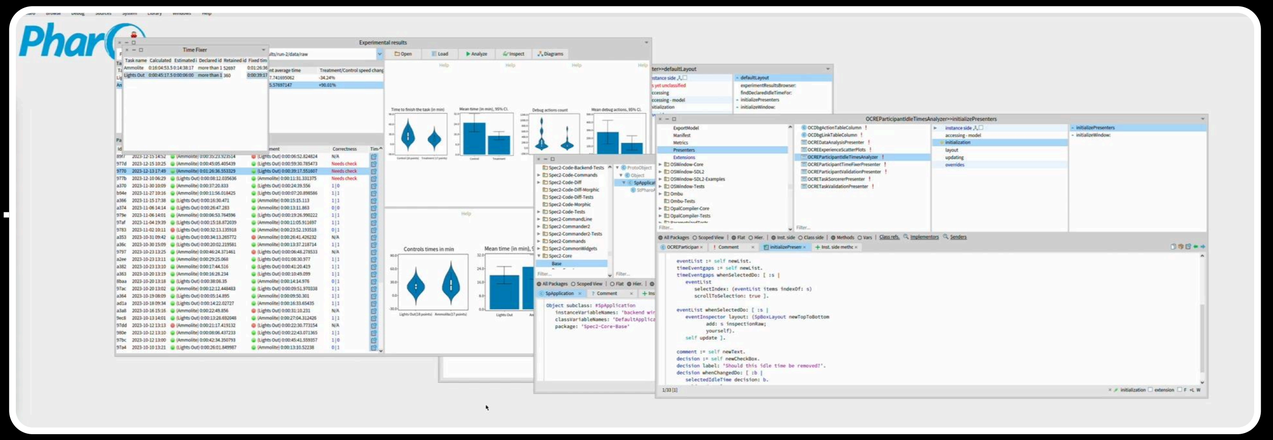
Double checking with the research team

4 **Decide what to Code**

Using Thematic analysis tools (Taguette)

5 **Grouping Codes into Themes**

(Themes are at a provisional stage)



~~Tasks~~

Methods

1 **Observed the Videos**

Familiarizing myself with the Data

2 **Generated Transcriptions**

Fixing mistakes along the way

3 **Created the Codes**

Double checking with the research team

4 **Decide what to Code**

Using Thematic analysis tools (Taguette)

5 **Grouping Codes into Themes**

(Themes are at a provisional stage)

[10:05.840 --> 10:07.280] Vamos a ver que es lo que está haciendo.

[10:07.440 --> 10:08.740] Y porque no está stageando.

[10:10.960 --> 10:11.760] Porque eventualmente.

[10:11.760 --> 10:13.120] Lo que vamos a hacer es entender el problema.

[10:13.360 --> 10:14.060] Escribir un test.

[10:14.780 --> 10:16.140] O ver si podemos escribir un test.

[10:17.260 --> 10:18.060] Hasta ahora.

[10:21.960 --> 10:22.360] Perdón.

[10:22.680 --> 10:24.740] No tengo test del stageando.

[10:24.860 --> 10:26.480] Lo estaba haciendo de manera.

[10:27.760 --> 10:28.160] Orgánica.

[10:29.540 --> 10:30.120] Pero bueno.

[10:30.140 --> 10:31.100] Como lo estoy haciendo por un video.

[10:31.160 --> 10:32.300] Ahora me estoy obligado a hacer test.

[10:33.800 --> 10:34.240] Bueno.

[10:34.400 --> 10:36.760] Nos frenamos en el R29.

[10:40.160 --> 10:40.720] Y ahora.

[10:40.940 --> 10:41.980] Esto debería decir.

[10:42.120 --> 10:43.680] El R29 debería decir true.

[10:44.420 --> 10:45.620] Y de hecho lo dice.

[10:47.120 --> 10:48.380] Vamos a hacer una cosa.

Methods

1 **Observed the Videos**

Familiarizing myself with the Data

2 **Generated Transcriptions**

Fixing mistakes along the way

3 **Created the Codes**

Double checking with the research team

4 **Decide what to Code**

Using Thematic analysis tools (Taguette)

5 **Grouping Codes into Themes**

(Themes are at a provisional stage)

Main Themes

Understanding Information Flow

(Discoverability, Needs, Doubts, Memory/History)

Error Correction & Prevention

(Error Identification - Accuracy - Doubts - Tasks - Memory/History)

Efficient Window Management

(Discoverability - Customization - Layout - External References - Useful Functionalities - Efficiency - Saturation)

Shortcut Learning & Searching

(Discoverability - Shortcuts - Shortcut Standardization - Error Identification - Hesitation)

Main Themes

Understanding Information Flow

(Discoverability, Needs, Doubts, Memory/History)

“Here you can see well that you couldn't see before. Because the location of records. It was hiding a little bit. The thing. ***I knew it because I know the code. But the debugger didn't show it to me.”***

“Navigating Pharo browser codes is extremely complex.”

Error Correction & Prevention

(Error Identification - Accuracy - Doubts - Tasks - Memory/History)

Efficient Window Management

(Discoverability - Customization - Layout - External References - Useful Functionalities - Efficiency - Saturation)

Shortcut Learning & Searching

(Discoverability - Shortcuts - Shortcut Standardization - Error Identification - Hesitation)

Main Themes

Understanding Information Flow

(Discoverability, Needs, Doubts, Memory/History)

Error Correction & Prevention

(Error Identification - Accuracy - Doubts - Tasks - Memory/History)

*“**I don't have a way to go back** to where I was debugging just now. Because it forgot. I need something with memory. **For when I make a mistake. Since I'm going fast I make mistakes.**”*

Efficient Window Management

(Discoverability - Customization - Layout - External References - Useful Functionalities - Efficiency - Saturation)

Shortcut Learning & Searching

(Discoverability - Shortcuts - Shortcut Standardization - Error Identification - Hesitation)

Main Themes

Understanding Information Flow

(Discoverability, Needs, Doubts, Memory/History)

Error Correction & Prevention

(Error Identification - Accuracy - Doubts - Tasks - Memory/History)

Efficient Window Management

(Discoverability - Customization - Layout - External References - Useful Functionalities - Efficiency - Saturation)

“This window goes to the background. You have to keep looking. There it is. ***This is how I set it up this time because I needed to see a lot of things at the same time.”***

Shortcut Learning & Searching

(Discoverability - Shortcuts - Shortcut Standardization - Error Identification - Hesitation)

Main Themes

Understanding Information Flow
(Discoverability, Needs, Doubts, Memory/History)

Error Correction & Prevention
(Error Identification - Accuracy - Doubts - Tasks - Memory/History)

Efficient Window Management
(Discoverability - Customization - Layout - External References - Useful Functionalities - Efficiency - Saturation)

Shortcut Learning & Searching
(Discoverability - Shortcuts - Shortcut Standardization - Error Identification - Hesitation)

*“**Maybe there is a shortcut. No, there's no shortcut for that.** Well, but this is the instruction we want.”*

Shortcut Learning

- *Willingness to learn*
- *Uncertainty*

"***I have used these shortcuts since I programmed in Smalltalk**. Especially the debug it. But it was before Pharo. **So, I don't remember how when I... how I learned that.***"

Shortcut Learning

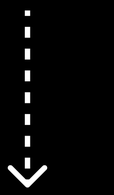
Background

"***I have used these shortcuts since I programmed in Smalltalk***. *Especially the debug it. But it was before Pharo.* ***So, I don't remember how when I... how I learned that.***"

+

*“**Maybe there is a shortcut. No, there's no shortcut for that.** Well, but this is the instruction we want.”*

Background



Solutions

Dev Characterization

- Understand UI/Interactions
- 3 archetypes

1 Specialist

2 Explorer

3 Stack Organizer

Dev Characterization

- Understand UI/Interactions
- 3 archetypes

1 Specialist

2 Explorer

3 Stack Organizer

(Criteria)

Shortcut Use

Interface Interaction

Layout Customization

Tasks and Tools

Dev Characterization

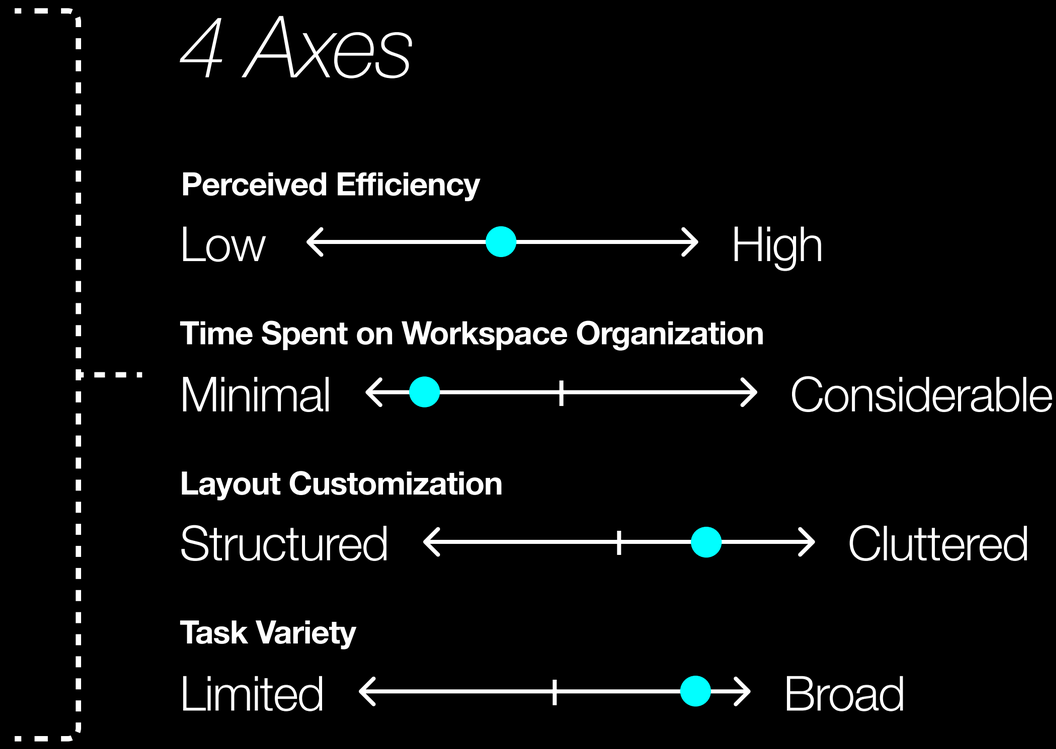
- Understand UI/Interactions
- 3 archetypes

1 Specialist

2 Explorer

3 Stack Organizer

4 Axes



1. Specialist

Perceived Efficiency

Low ←●—|—→ High

Time Spent on Workspace Organization

Minimal ←—|—●→ Considerable

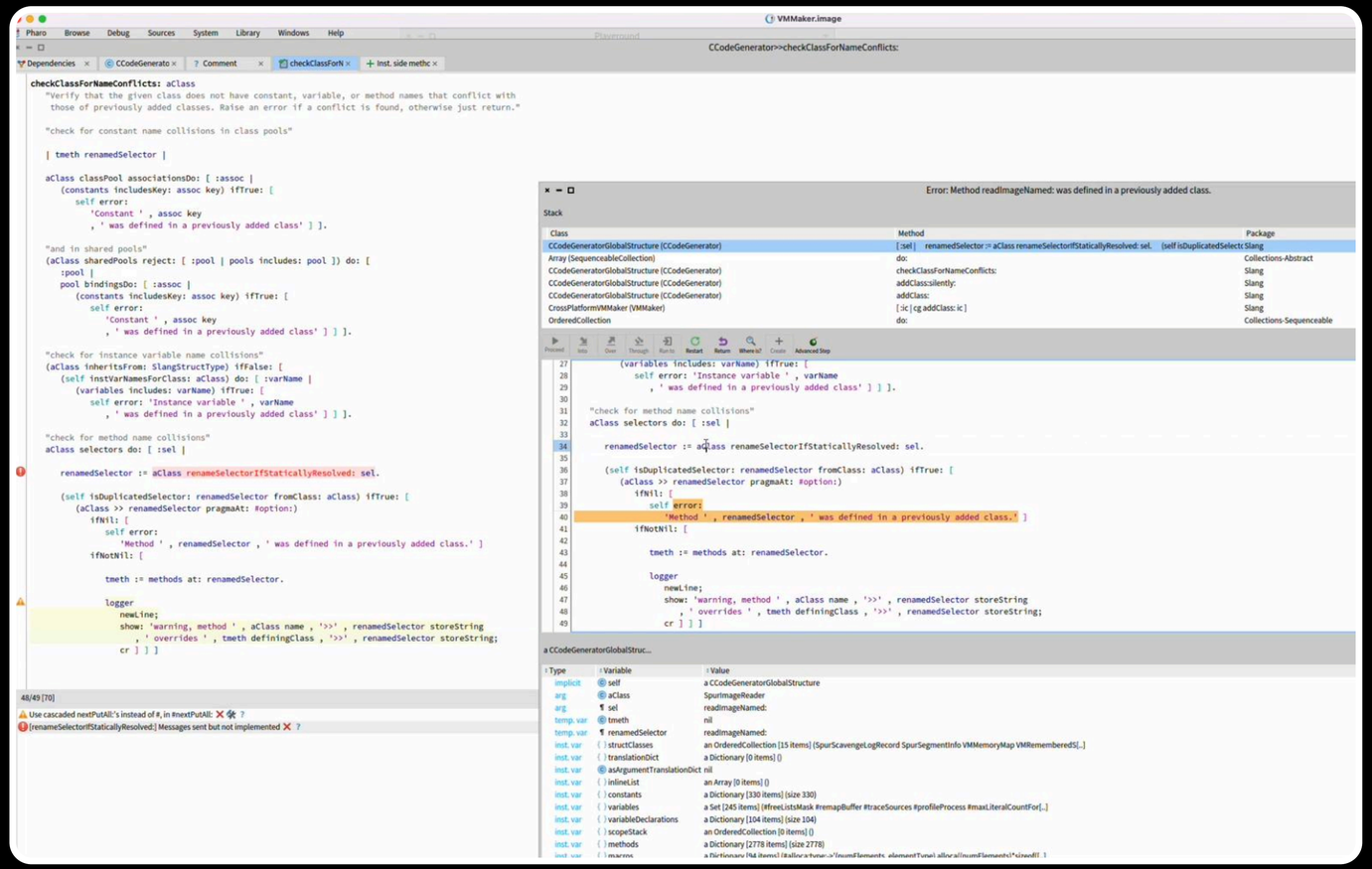
Layout Customization

Structured ←●—|—→ Cluttered

Task Variety

Limited ←●—|—→ Broad

“This is the way I'm organizing it this time because I needed to see a lot of things at the same time.”



Interface Interaction:

- Carefully organizes their workspace and avoids having too many windows open in the first place.
- Prefers to maximize individual windows size to display as much information as possible without overloading the visual space.

Shortcut Use:

- Dives into code manually. Wants to be able to maximize windows with more accuracy and navigate code faster.

2. Explorer

Perceived Efficiency

Low ⟷ High

Time Spent on Workspace Organization

Minimal ⟷ Considerable

Layout Customization

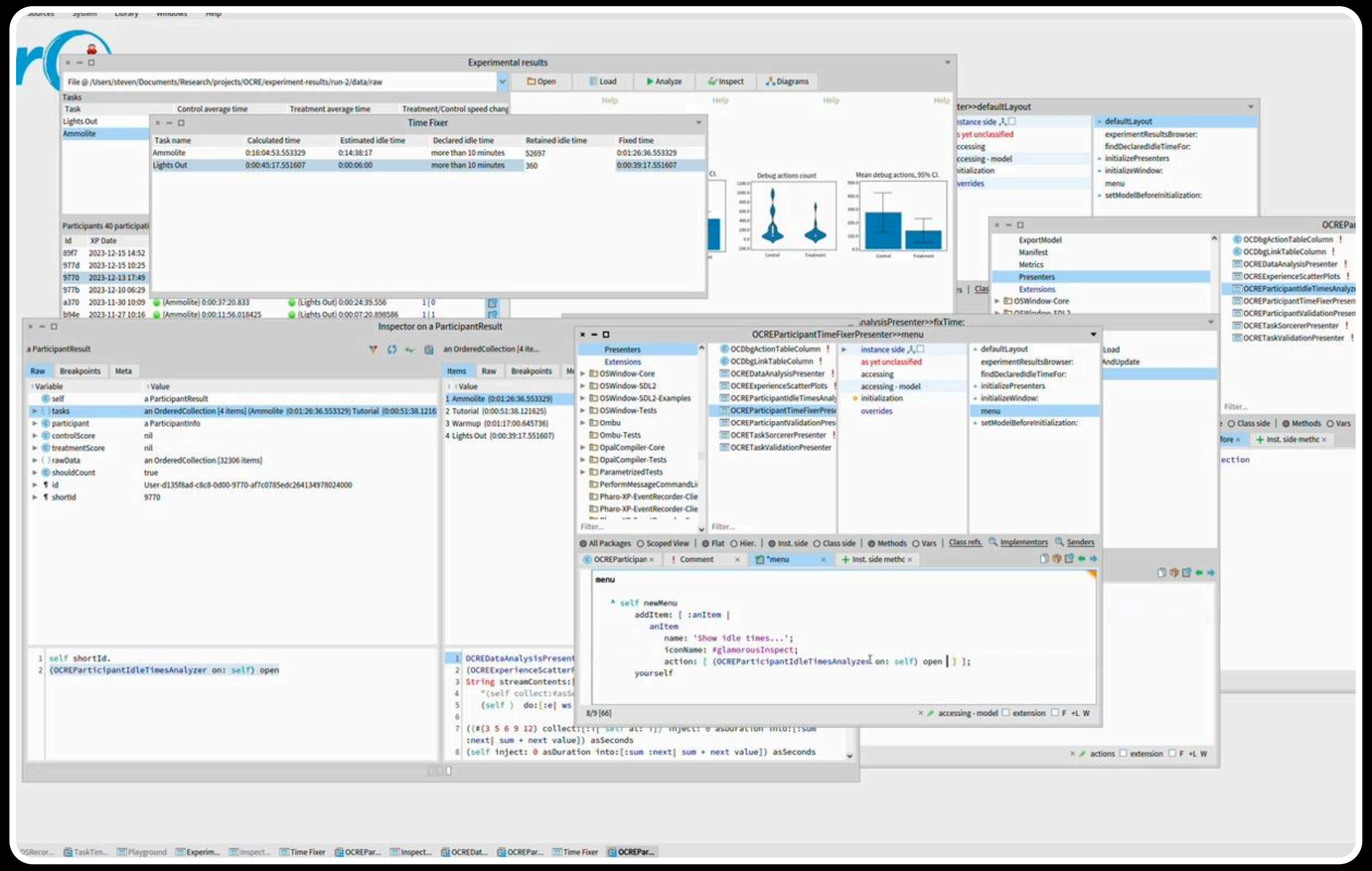
Structured ⟷ Cluttered

Task Variety

Limited ⟷ Broad

“I don't give importance to the code browser. It's going to take 20% of my time. But I prefer to open it. Use it and close it.”

"So there's too much window. And... I cannot understand what I'm seeing. So I'm closing everything."



Interface Interaction:

- They open windows as needed and close them once they’re no longer useful or when too much information on the screen feels overwhelming.
- Dislikes having to arrange and rearrange windows for task specific layouts, won’t do so unless necessary.

Shortcut Use:

- Uses shortcuts to open and close windows, as well as to copy and paste code. Wishes there were shortcuts to speed up arranging windows.

3. Stack Organizer

Perceived Efficiency

Low ⟷ High

Time Spent on Workspace Organization

Minimal ⟷ Considerable

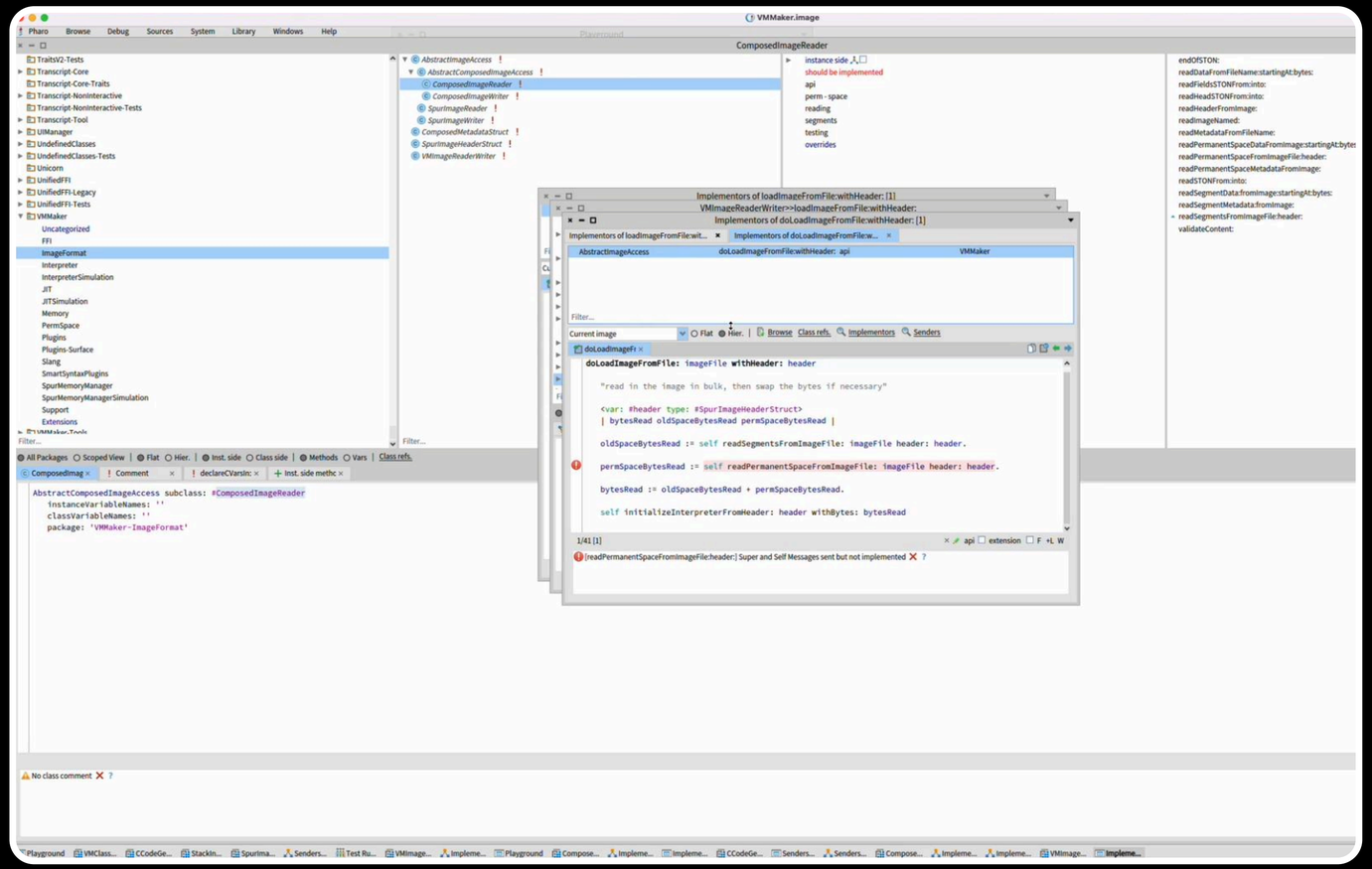
Layout Customization

Structured ⟷ Cluttered

Task Variety

Limited ⟷ Broad

“I use the windows as a stack. Because it's the most comfortable to use with the keyboard, with the current design. That is. I open windows. I close windows. And they open in a stack.”



Interface Interaction:

- Maintains a basic layout with background windows, adding windows according to the flow of each task.
- Sets up stacks of windows related to each task, using a “mind map” to remember their layout and structure.

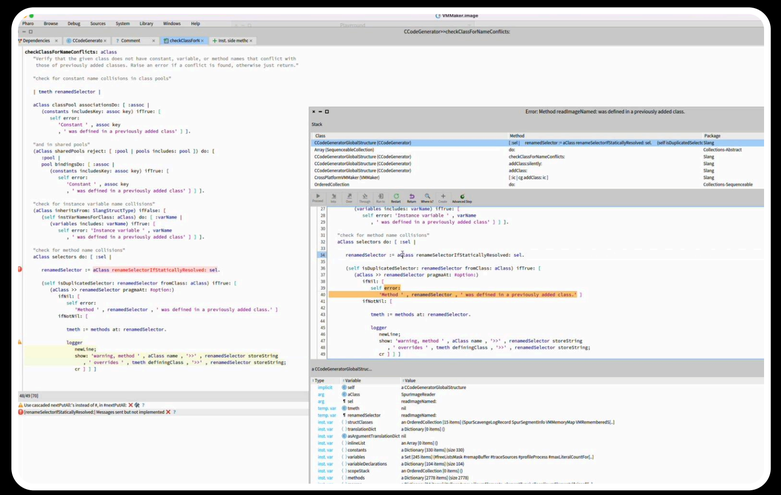
Shortcut Use:

- Uses shortcuts to optimize navigation and quick space organization.

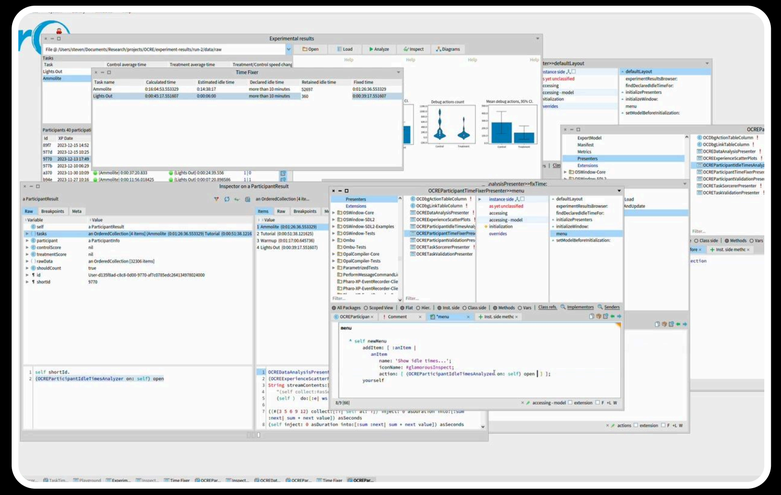
How does Pharo’s UI Impact these Developers?

How does Pharo’s UI Impact these Developers?

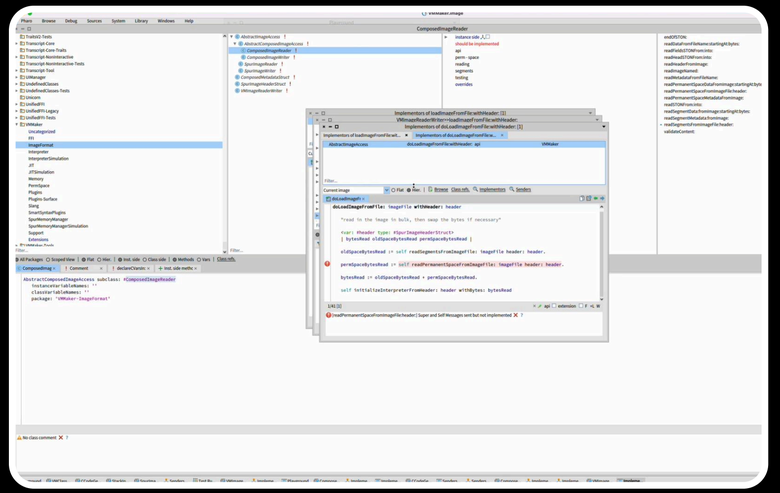
Specialist:



Explorer:

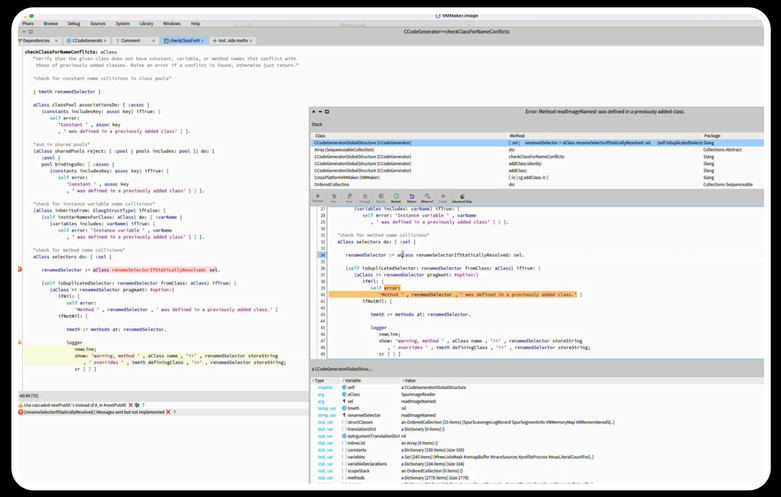


Stack Organizer:

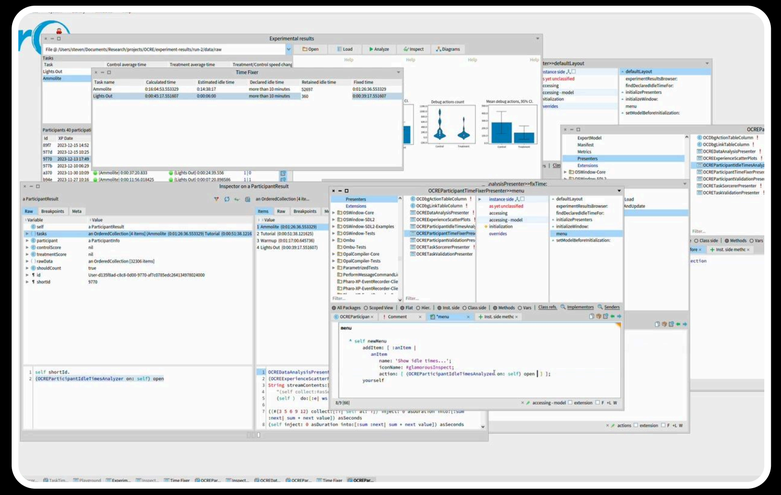


How does Pharo’s UI Impact these Developers?

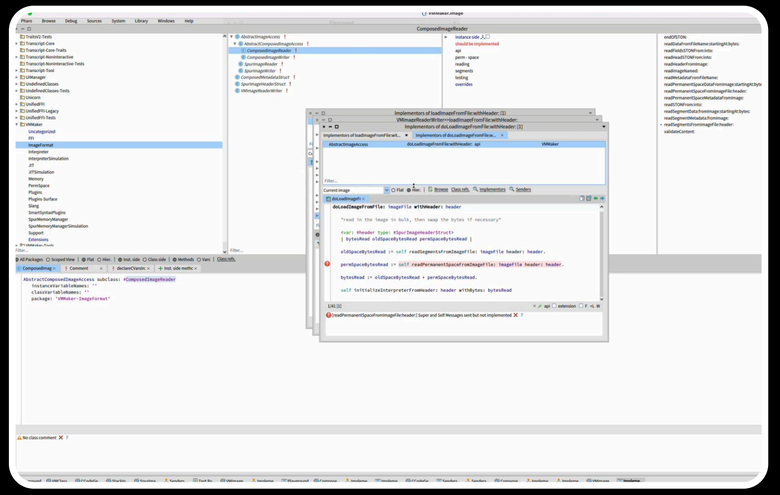
Specialist:



Explorer:



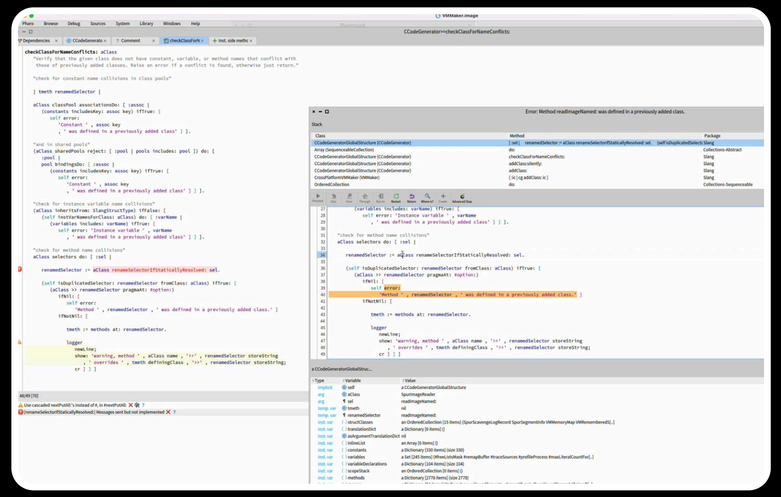
Stack Organizer:



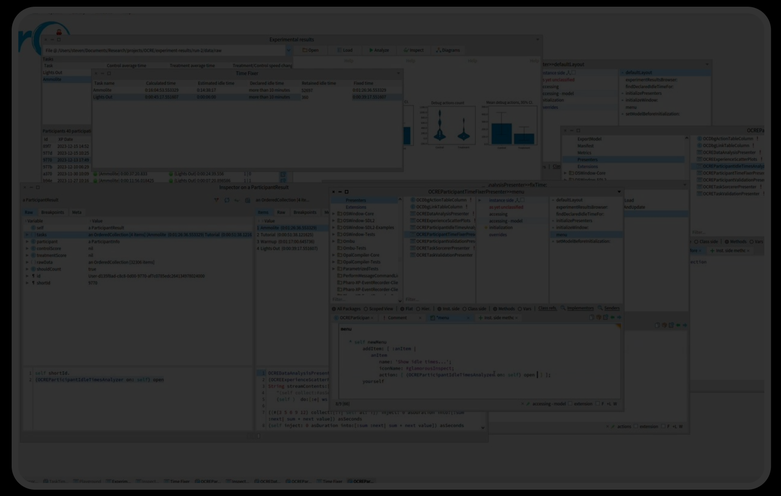
Floating Windows/Tools

How does Pharo’s UI Impact these Developers?

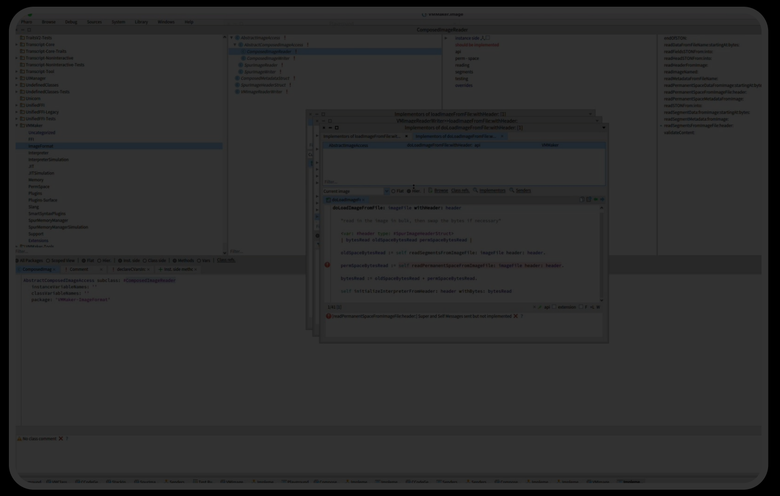
Specialist:



Explorer:



Stack Organizer:



Benefits

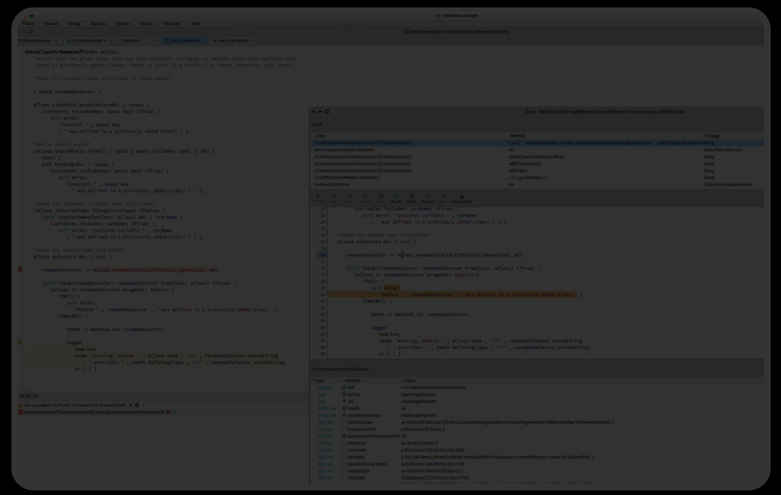
- Layout Customization allows the developer to maintain a clear structure that suits the task at hand.
- Windows help the developer compare and check different information simultaneously.

Drawbacks

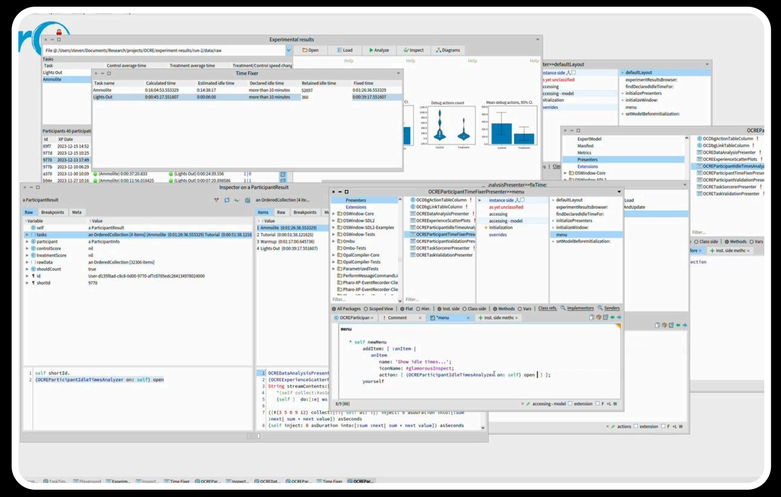
- Spends additional time manually resizing and organizing each window, which can slow down their workflow.
- Their preference for maximizing windows limits how much information they can see at once.
- The absence of shortcut options for layout functions makes it harder to work efficiently.

How does Pharo’s UI Impact these Developers?

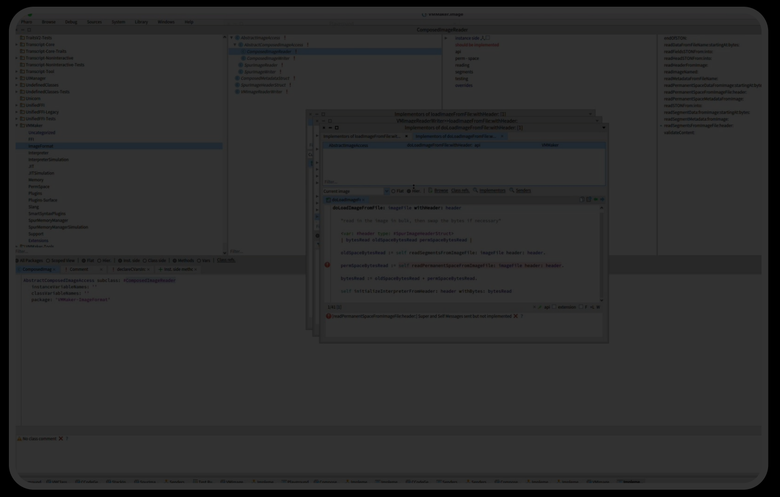
Specialist:



Explorer:



Stack Organizer:



Benefits

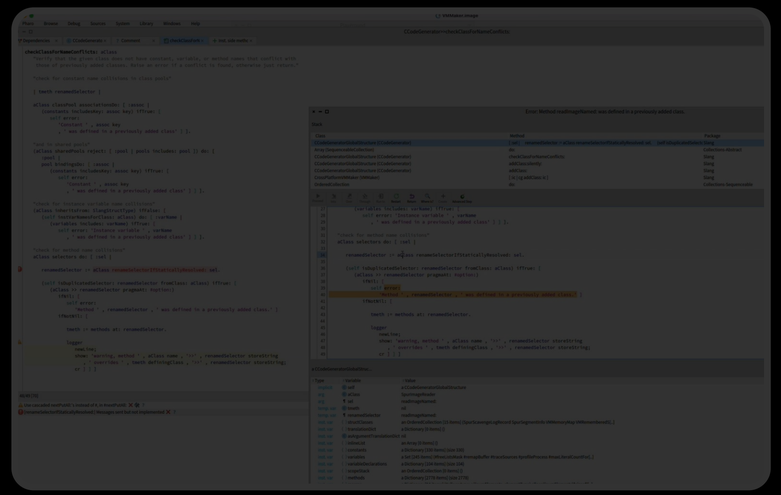
- Customizable layout allows the developer to work on complex projects and tasks that need tool diversity.
- Floating windows help users quickly access different tools and layers to complete tasks.

Drawbacks

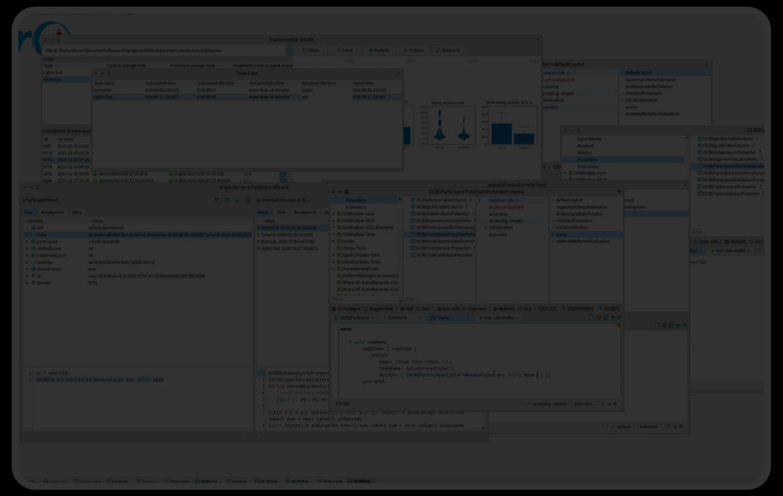
- Once too many windows have been opened, the user’s efficiency goes down, forcing them to eventually close everything.
- Inconsistencies in shortcuts disrupt their flow.
- Lack of structure causes developers to lose track of the tasks and tools’ layouts.

How does Pharo’s UI Impact these Developers?

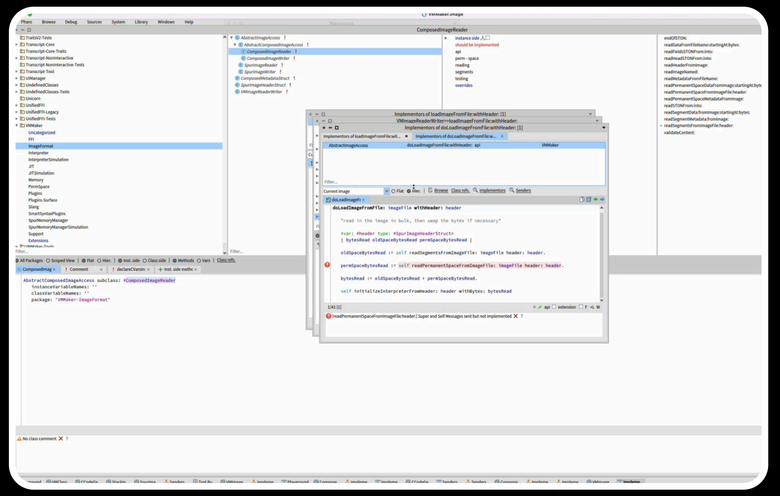
Specialist:



Explorer:



Stack Organizer:



Benefits

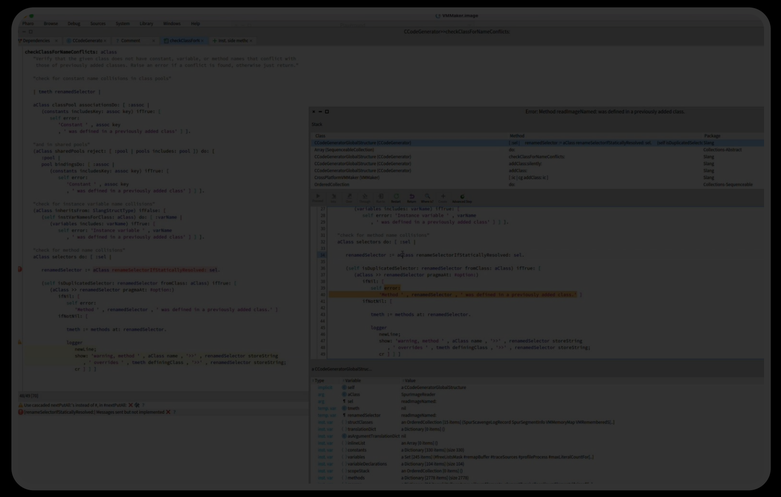
- Developers can work on broader tasks without losing their structure.
- Stack system makes navigating more efficient and reduces mouse use.

Drawbacks

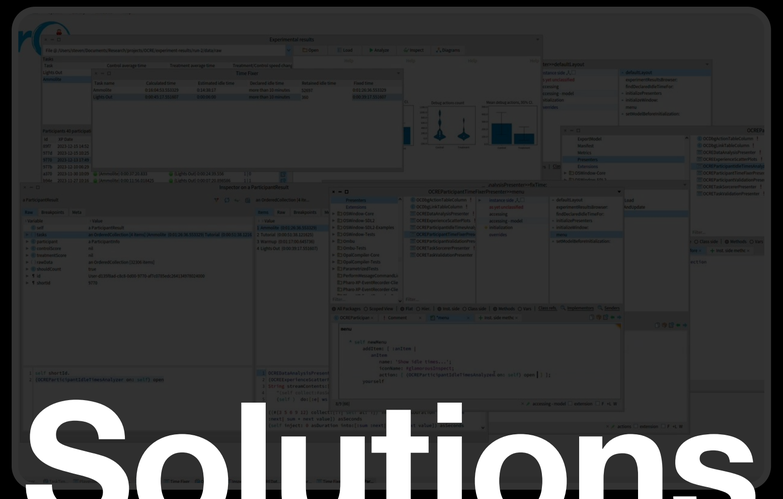
- This layout requires the developer having a mind map/tracking the windows being used.
- Lack of accuracy hinders the developers’ workflow.
- Tool layout and window size preferences aren’t saved, hindering the developers’ speed.

How does Pharo’s UI Impact these Developers?

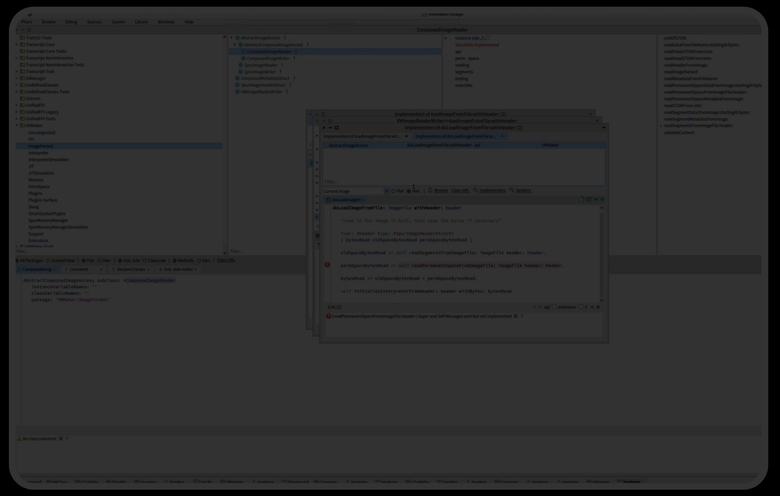
Specialist:



Explorer:



Stack Organizer:



Solutions

Benefits

- Developers can work on broader tasks without losing their structure.
- Stack system makes navigating more efficient and reduces mouse use.

Drawbacks

- This layout requires the developer having a mind map/tracking the windows being used.
- Lack of accuracy hinders the developers’ workflow.
- Tool layout and window size preferences aren’t saved, hindering the developers’ speed.

Solutions

1

Grouping + Tabs

To avoid Visual Clutter

2

Clashing Windows

Faster Window Arrangement

3

Customizable Layout Presets

For dedicated tasks

4

Window Memory

No need to rearrange window layouts

5

Undo Action Shortcut

To correct mistakes

Solutions

1 **Grouping + Tabs**

To avoid Visual Clutter

2 **Automatic Window Clashing**

Faster Window Arrangement

3 **Customizable Layout Presets**

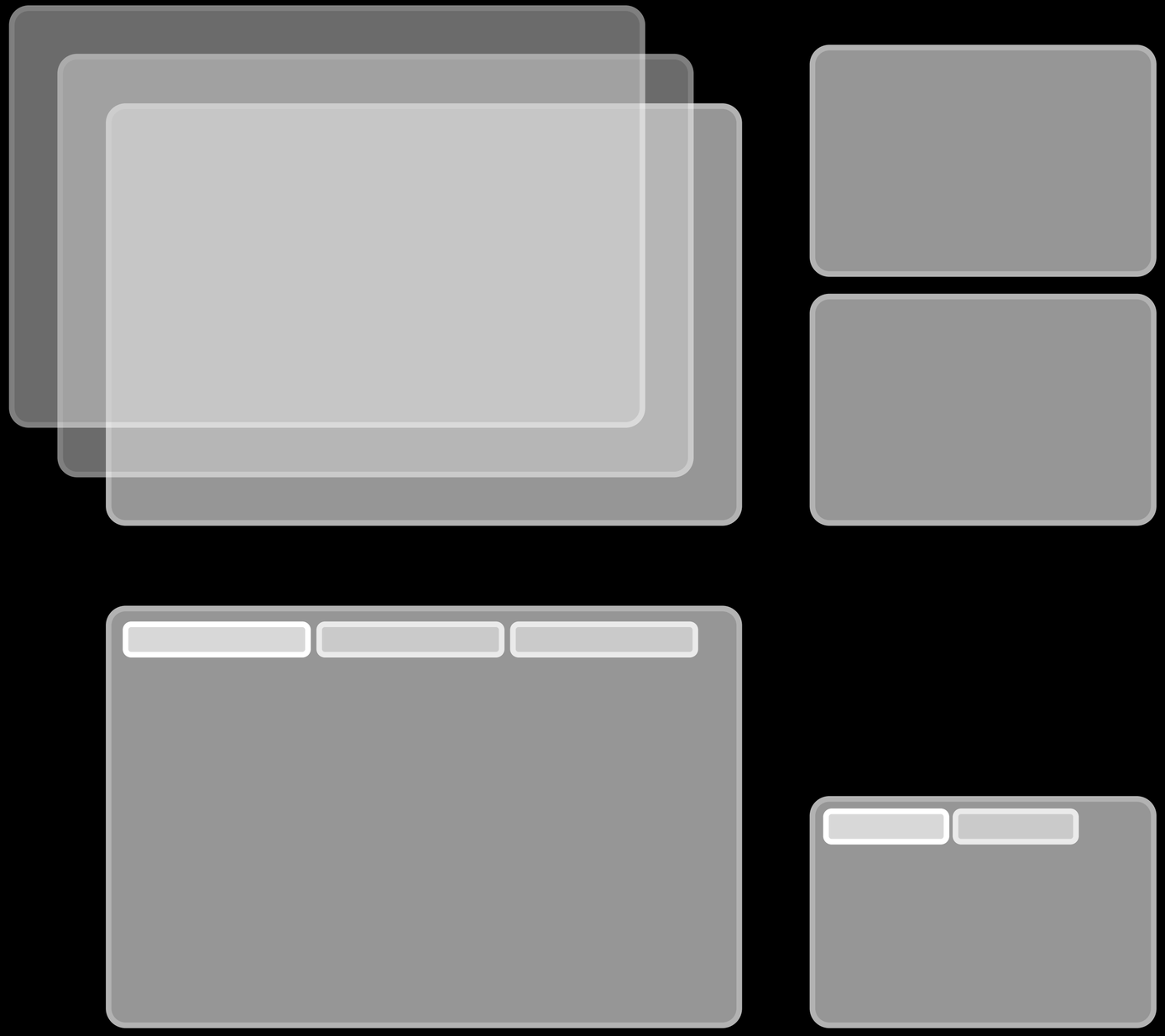
For dedicated tasks

4 **Window Memory**

No need to rearrange window layouts

5 **Undo Action Shortcut**

To correct mistakes



Debuggers

Playgrounds

Solutions

1 **Grouping + Tabs**

To avoid Visual Clutter

2 **Automatic Window Clashing**

Faster Window Arrangement

3 **Customizable Layout Presets**

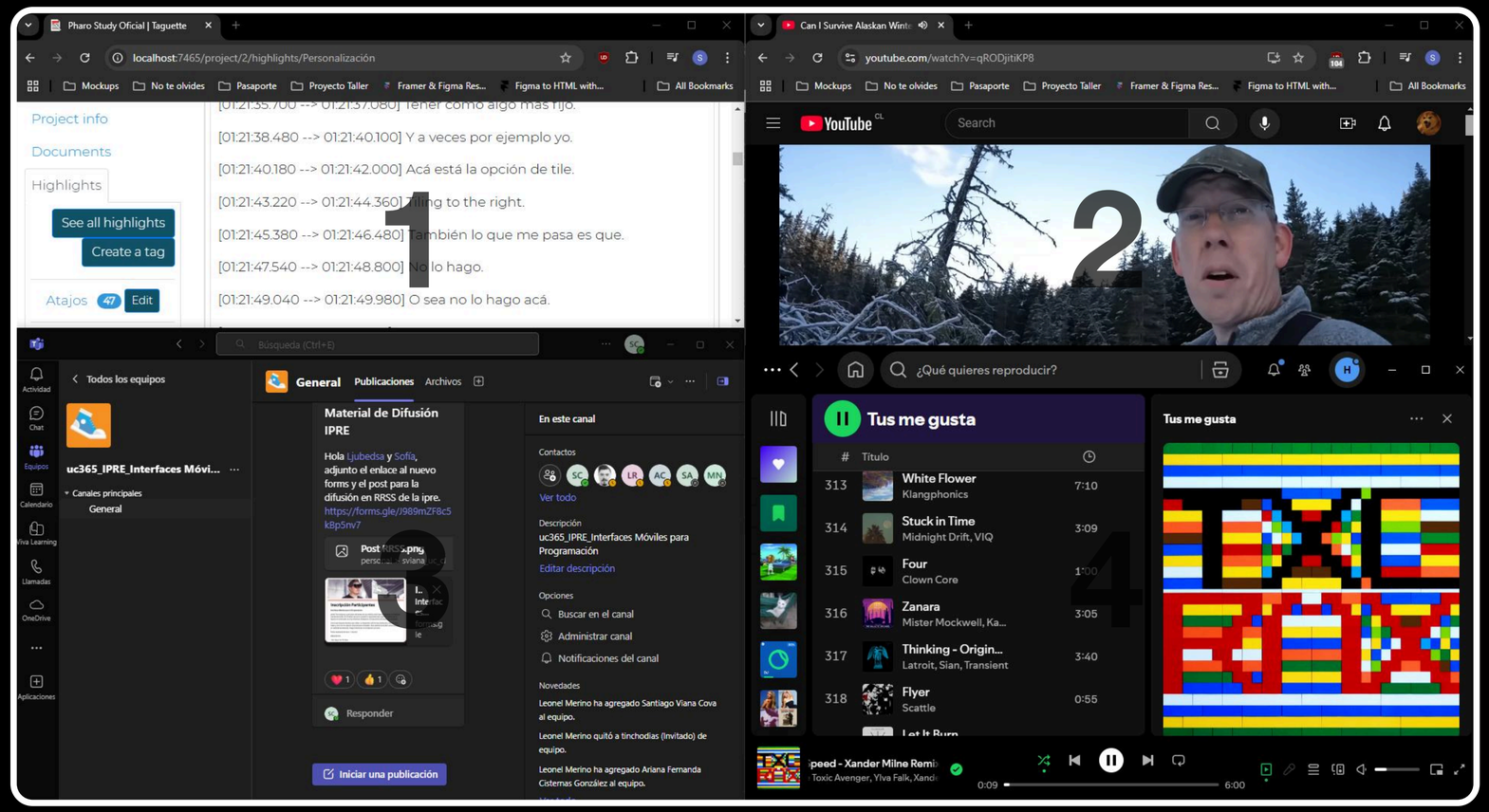
For dedicated tasks

4 **Window Memory**

No need to rearrange window layouts

5 **Undo Action Shortcut**

To correct mistakes



Solutions

1 **Grouping + Tabs**

To avoid Visual Clutter

2 **Automatic Window Clashing**

Faster Window Arrangement

3 **Customizable Layout Presets**

For specific tasks

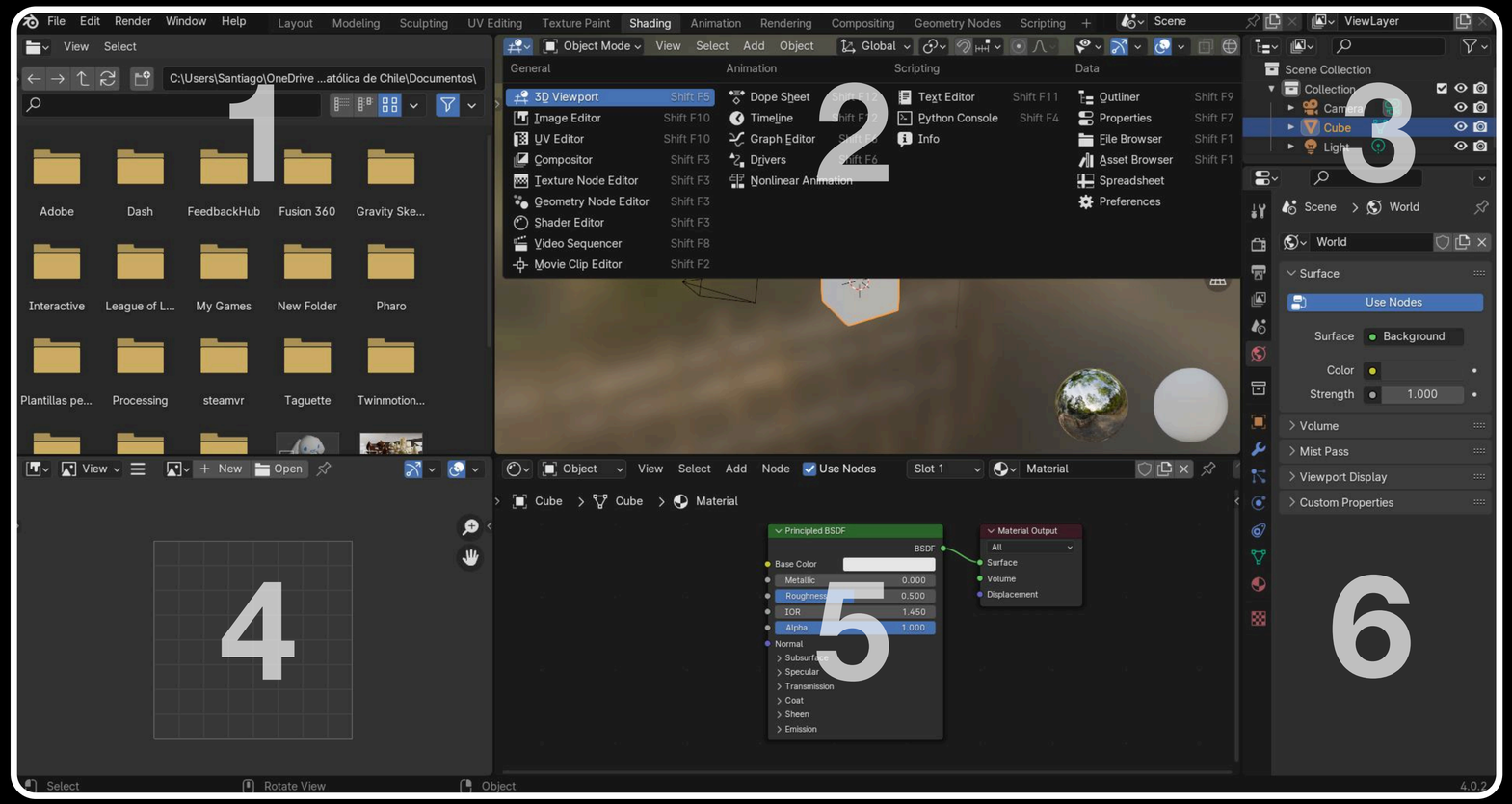
4 **Window Memory**

No need to rearrange window layouts

5 **Undo Action Shortcut**

To correct mistakes

Presets



Solutions

1 **Grouping + Tabs**

To avoid Visual Clutter

2 **Automatic Window Clashing**

Faster Window Arrangement

3 **Customizable Layout Presets**

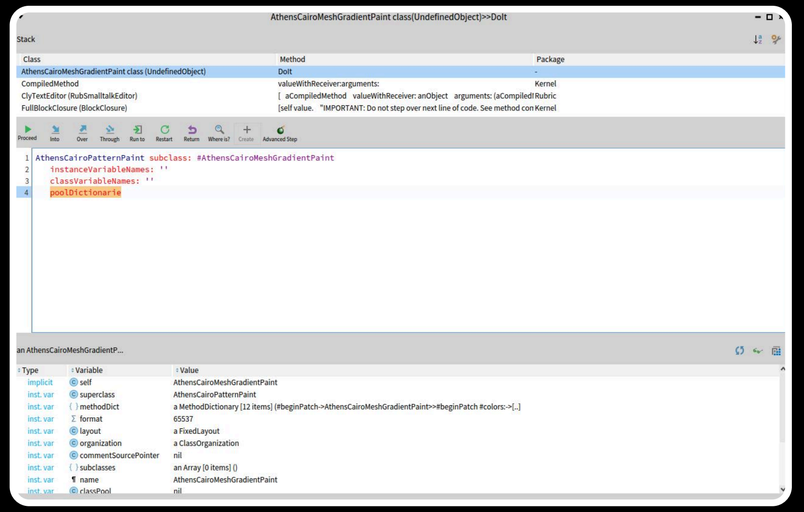
For dedicated tasks

4 **Window Memory**

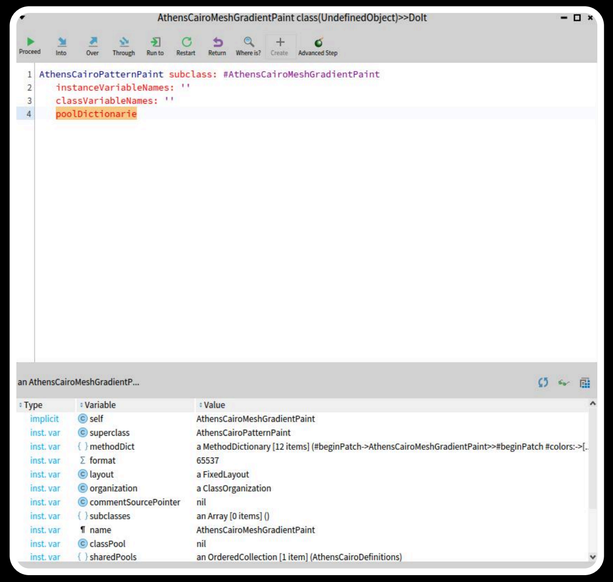
No need to rearrange window layouts

5 **Undo Action Shortcut**

To correct mistakes



Default Debugger



Modified Debugger

Solutions

1 **Grouping + Tabs**

To avoid Visual Clutter

2 **Automatic Window Clashing**

Faster Window Arrangement

3 **Customizable Layout Presets**

For dedicated tasks

4 **Window Memory**

No need to rearrange window layouts

5 **Undo Action Shortcut**

To correct mistakes