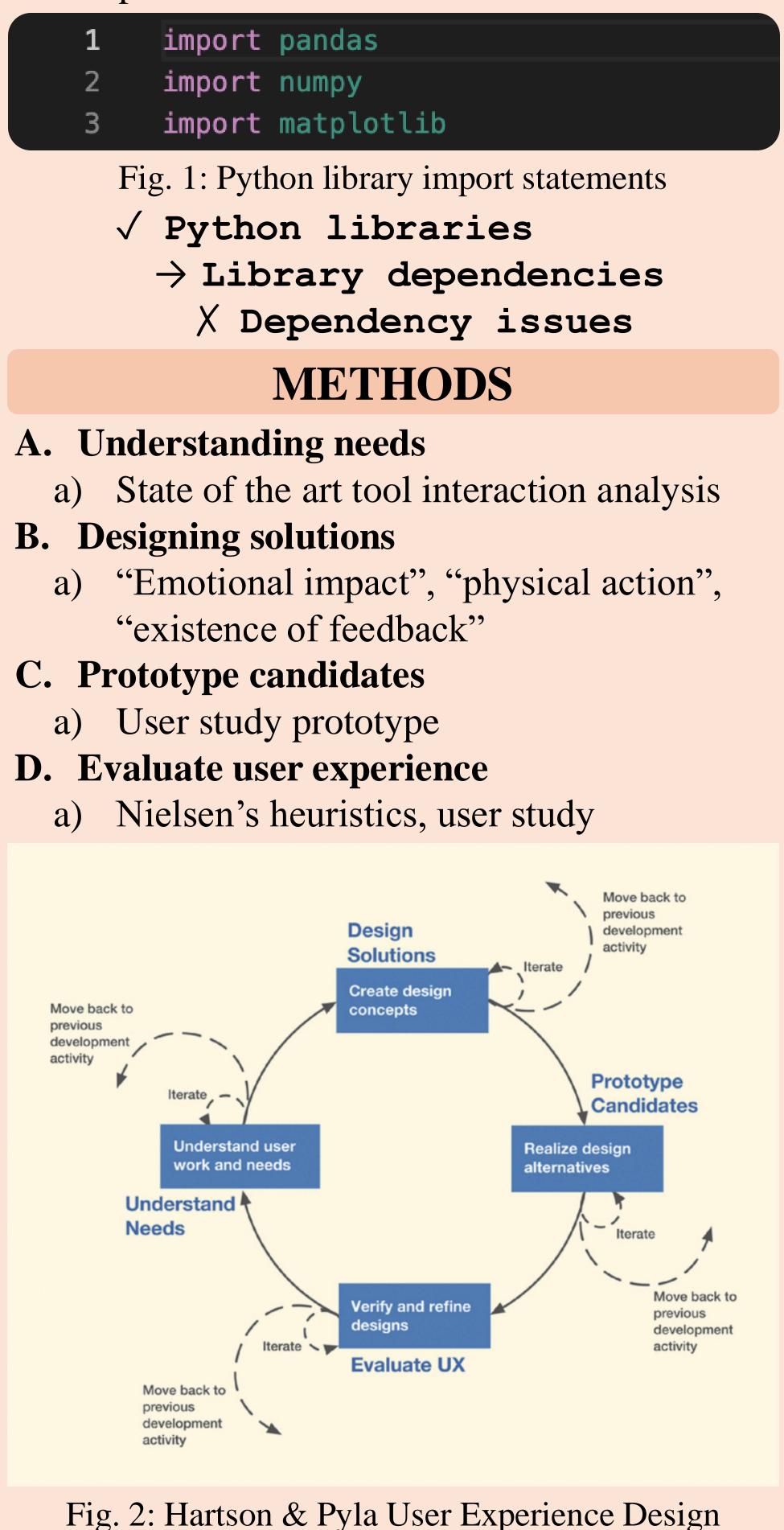


Multicultural Academic Opportunities Program

## **INTRODUCTION**

Timely recognition of and actionability toward resolving Python library dependency issues is crucial in supporting a productive software engineering workflow. Considering not all Python developers have machine learning (ML) knowledge, direct interaction with ML models utilized for backend development is not an ideal user experience. Thus, usability engineering plays a critical role in enabling Python developers with actionable information.



Lifecycle

# **User Interface Development using CustomTkinter for Automating Python Dependency Recognition**

Travis Chan, Dibyendu Brinto Bose, and Dr. Chris Brown College of Engineering Department of Computer Science, Virginia Tech, Blacksburg, VA

#### RESULTS Library Recommendation App Exit Load File & Generate Recommendations Save Recommendations ome Centrality Analysis Community Impact Prediction Welcome to AutoPyDep AutoPyDep is an automated AI tool helping software engineers detect and resolve dependency issues Instructions for Use Click on the "Load File & Generate Recommendations" button to upload your text file containing library names and versions. Navigate through the tabs (Centrality Analysis, Community Impact, Predictions) to view detailed insights and recommendations for each dependency Click on the "Save Recommendations" button to save the generated recommendations for future reference **Key Features** Presents calculated node centrality and level of centrality concern for each dependency Presents the number of community libraries and a list of all impacted libraries for each dependency along with the impact graph.

Centrality Analysis Community Impact Predictions

1. Upload Your File

2. View Recommendations

Save Your Recommendations

Node Centrality Centrality Concern level Community Impacted libraries

Key Terminology The importance of a node within the graph, often measured by the number of connections it has and its position within the network. Indicates whether a node's centrality suggests a high impact on the network, which may require the node to be updated or maintained A group of nodes within the graph that are more densely connected to each other than to other nodes, often representing related libraries or dependencies Libraries that may be affected by updates or changes in a particular node due to their connections within the graph

	Library Recommendation App			
Load File & Generate Recommendations	Save Recommendations	Exit		
	Home Centrality Analysis Community Impact Predictions			
Library	Node Centrality (x10e9)	Centrality Concern Level		
pandas_2.1.4	0.014	No high centrality concern		
NumPy_1.26.4	20.311	No high centrality concern		
matplotlib_1.5.1	1.136	No high centrality concern		
Pillow_2.6.1	322.705	No high centrality concern		
Scikit-learn_1.1.3	0.582	No high centrality concern		
	Relevant Terminology			
Node Centrality	The importance of a node within the graph, often measured by the number of	connections it has and its position within the network		
Centrality Concern level	Indicates whether a node's centrality suggests a high impact on the network, w	which may require the node to be updated or mai		

Presents the predicted next release date and reasons for update (categorically) for each dependency

Commands ē pip install pandas==2.1.4 panda D install numpy==1.26.4 NumPy

D p install matplotlib==1.5.1 matplotlib D pip install pillow==2.6.1 Pillow ē pip install scikit-learn==1.1.3 Scikit-learn

	Library Reco	mmendation App	
Load File & Generate Recomm	nendations Save Reco	ommendations	Exit
	Home Centrality Analysis	Community Impact Predictions	
Library	Version	Community	Impacted Librarie
pandas_2.1.4	2.1.4	1 libraries	matplotlib_1.5.1
NumPy_1.26.4	1.26.4	122 libraries	Requests_0.7.3 Dash_2.10.2 Requests 2.30.0
matplotlib_1.5.1	1.5.1	104 libraries	NumPy_1.26.3 NumPy_1.22.1 Requests 0.7.3
Pillow_2.6.1	2.6.1	11 libraries	TensorFlow_2.8.4 TensorFlow_2.2.3 TensorFlow_2.3.3
Scikit-learn_1.1.3	1.1.3	1 libraries	NumPy_1.24.0
Community Impacted libraries	A group of nodes within the graph that are more Libraries that may be affecte	erminology densely connected to each other than to oth d by updates or changes in a particular node	
Load File & Generate Recom		ommendation App	Exit
	Home Centrality Analysis	Community Impact Predictions	
Library		ext Release Date	Predicted Node Categories
pandas_2.1.4		ly 2024	Bug Fix
NumPy_1.26.4	Ju	ly 2024	Performance Improvement

matplotlib\_1.5.1

Pillow\_2.6.1

Scikit-learn\_1.1.3

July 2024

August 2024

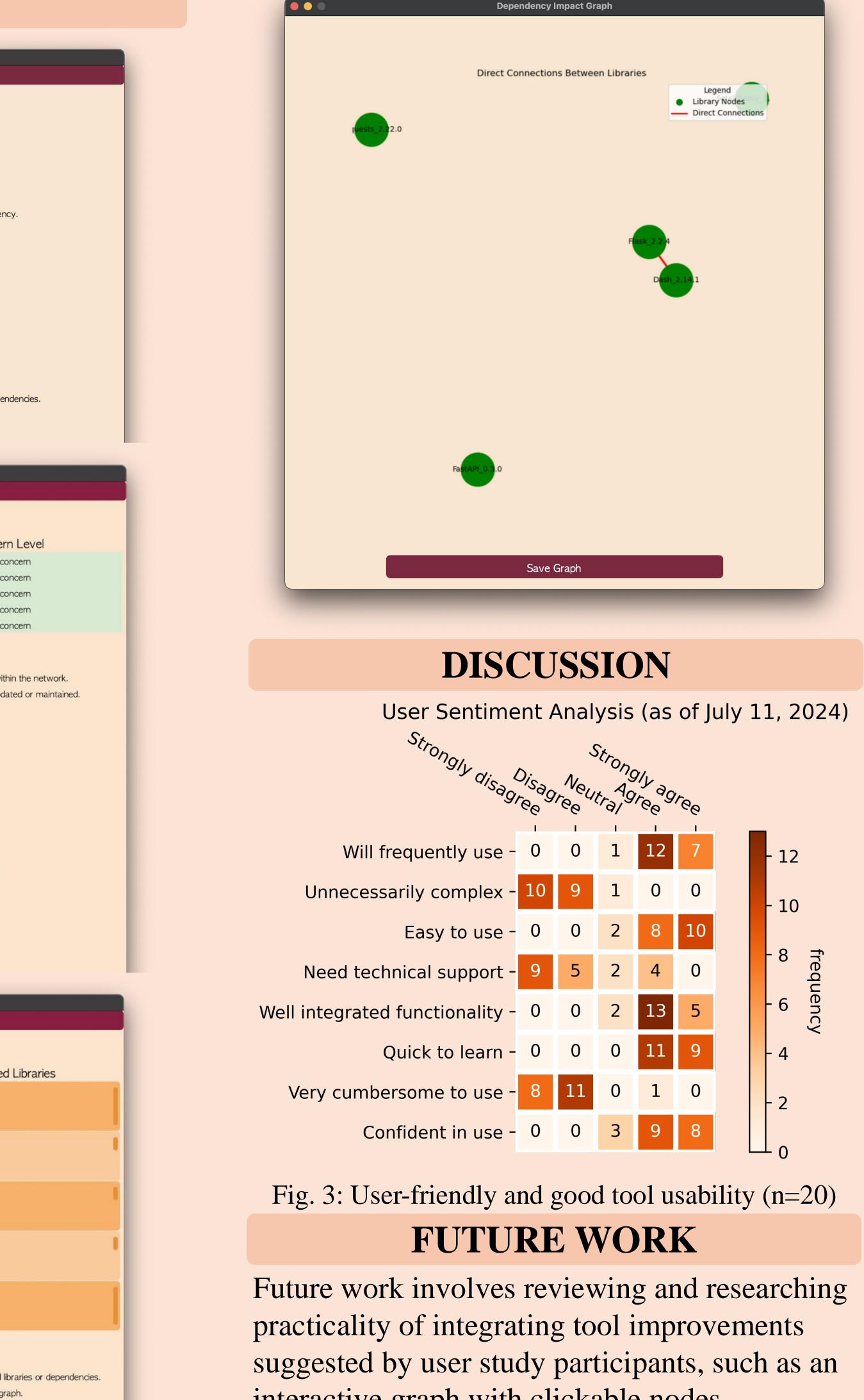
July 2024

Bug Fix, Performance Improvemen

Security Bug Fix, New Feature







interactive graph with clickable nodes.

## ACKNOWLEDGEMENTS

Thank you to Code World: No Blanket, specifically Dr. Chris Brown and Dibyendu Brinto Bose, and the Multicultural Academic **Opportunities Program.** 



Multicultural Academic Opportunities Program

### **INTRODUCTION**

Timely recognition of and actionability toward resolving Python library dependency issues is crucial in supporting a productive software engineering workflow. Considering not all Python developers have machine learning (ML) knowledge, direct interaction with ML models utilized for backend development is not an ideal user experience. Thus, usability engineering plays a critical role in enabling Python developers with actionable information.

## METHODS

### A. Understanding needs

a) Analysis of user work and needs reveals a variety of interaction forms for dependency management tools

#### **B.** Designing solutions

- "Emotional impact" a)
- "Physical action" **b**)
- "Existence of feedback" **c**)

### **C.** Prototype candidates

a) User study prototype

### **D.** Evaluate user experience

- a) Nielsen's heuristics
- b) User study to inform tool's impact on software development workflow
  - a. 20 participants (70% M, 30% F)
    - 1 to 10+ years of Python experience

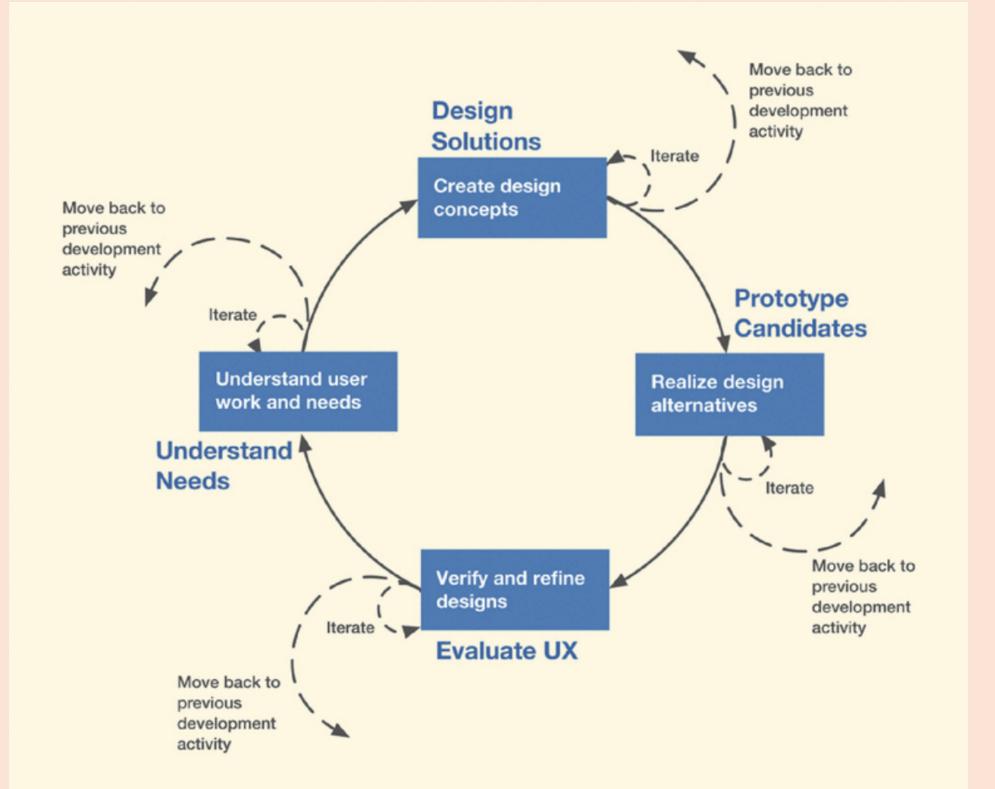


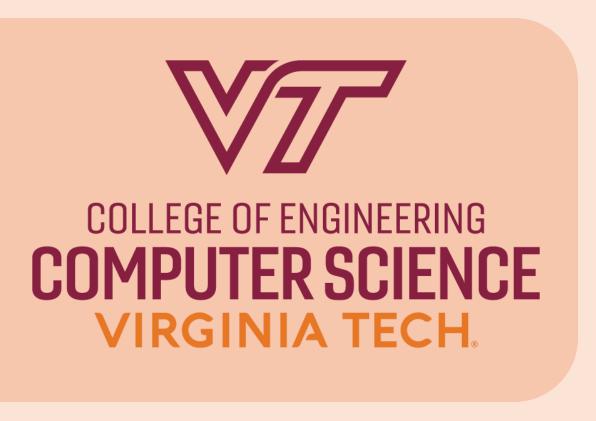
Fig 1. Hartson & Pyla UX Design Lifecycle

# **User Interface Development using CustomTkinter for Automating Python Dependency Recognition**

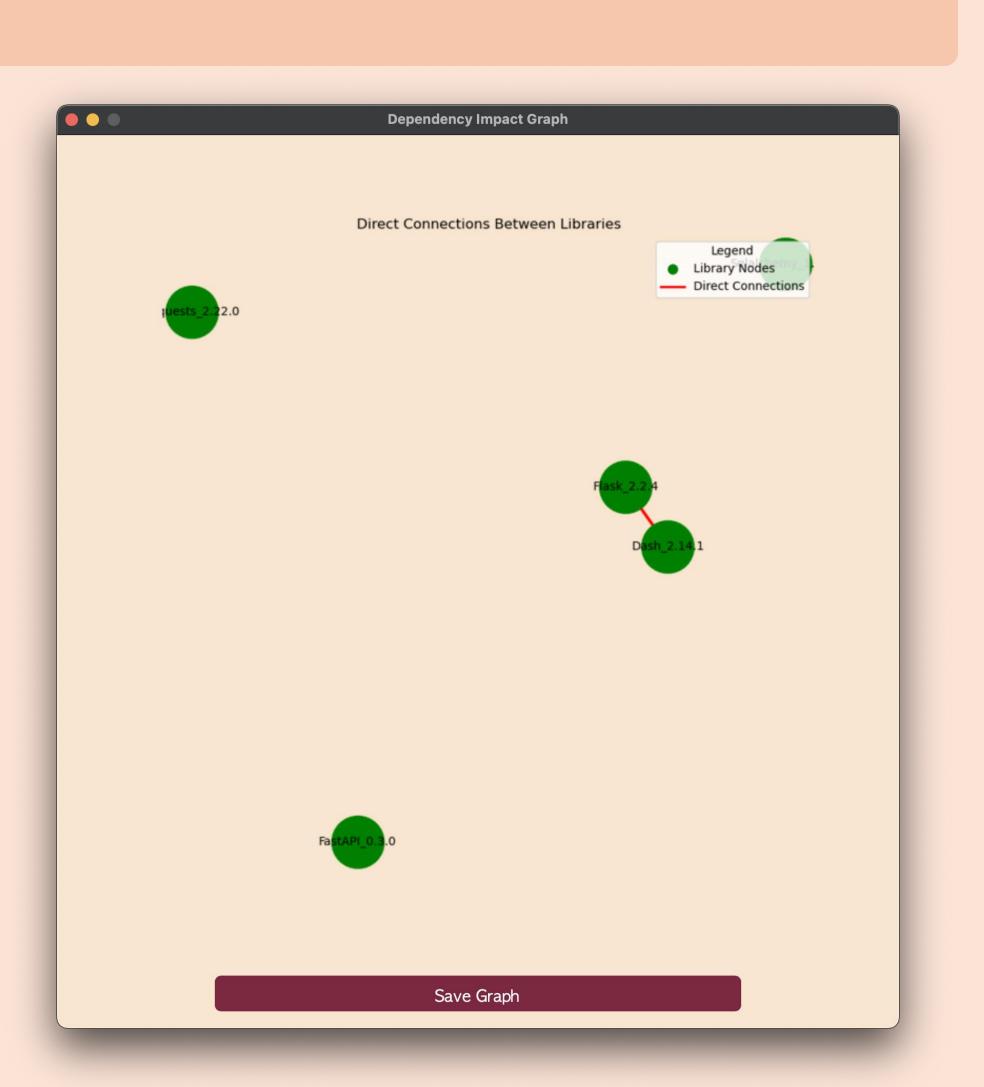
Travis Chan, Dibyendu Brinto Bose, and Dr. Chris Brown College of Engineering Department of Computer Science, Virginia Tech, Blacksburg, VA







## RESULTS



## DISCUSSION

User Sentiment Analysis (as of July 11, 2024)

Strongly	~	S	Stron.	<b>•</b> •		
Strongly disag	Disag ree	Neu	traj	lly ag	ree	
Will frequently use -		0	' 1	12	י 7	
Unnecessarily complex -	10	9	1	0	0	
Easy to use -	0	0	2	8	10	
Need technical support -	9	5	2	4	0	
Well integrated functionality -	0	0	2	13	5	
Quick to learn -	0	0	0	11	9	
Very cumbersome to use -	8	11	0	1	0	
Confident in use -	0	0	3	9	8	

Fig 2. User sentiment analysis heat map **FUTURE WORK** 

Future work involves reviewing and researching practicality of integrating tool improvements suggested by user study participants.

## ACKNOWLEDGEMENTS

Thank you to Code World: No Blanket, specifically Dr. Chris Brown and Dibyendu Brinto Bose, and the Multicultural Academic **Opportunities Program.** 



