An Empirical Survey of GitHub Repositories at U.S. Research Universities

Samuel D. Schwartz
Department of Computer Science
University of Oregon
Eugene, Oregon, USA
sam@cs.uoregon.edu

Boyana Norris
Department of Computer Science
University of Oregon
Eugene, Oregon, USA
norris@cs.uoregon.edu

Stephen F. Fickas
Department of Computer Science
University of Oregon
Eugene, Oregon, USA
stephenfickas@gmail.com

***Abstract*—In this work we aim to partially answer the question, "Just how many research software projects are out there?" by searching for open source GitHub projects affiliated with research universities in the United States. We explore this through keyword searches on GitHub itself and by scraping university websites for links to GitHub repositories. We then filter these results by using a large language model to classify GitHub repositories as research software engineering projects or not, finding over 35,000 RSE repositories. We report our results by university. We then analyze these repositories against metrics of popularity, such as stars and repository forks, and find just under 14,000 RSE repositories meet our minimum criteria for projects which have a community. Based on the time since a developer last pushed a change to a RSE repository with a community, we further posit that 3,300 RSE repositories with communities and a link to a research university are at risk of dying, and thus may benefit from sustainability support. Finally, across all RSE projects linked to a research university, we empirically find the top repository languages are Python, C++, and Jupyter Notebook.**

***Index Terms*—Empirical survey, universities, research software engineering, GitHub, repositories**

I. Introduction

Research software engineering (RSE), as a formal academic field of study, is relatively new. Broad questions like, "*Just how many research software projects are out there? Empirically speaking, what do they look like?*" continue to be ripe for answering. In this paper we expand on our earlier work [1], which explored and inventoried open source software projects hosted on GitHub with some nexus to a US Department of Energy national laboratory. In this work we apply similar methodology to uncover RSE repositories, but point to a comparatively larger ecosystem: all major research universities in the United States.

Specifically, the overarching research questions which guide this work are:

RQ1 What are all of the public software projects / repositories on GitHub with some nexus, even a weak one, to the major research universities in the United States?

RQ2 Of these software projects with some sort of nexus to a research university, which ones are RSE projects? More generally, if given a large set of repositories how can we determine which ones are RSE and which are non-RSE?

RQ3 Of the RSE projects / repositories found through answering RQ2, which ones are likely to be popular outside of the project's core developers? That is, which projects are used by the community – either a university-internal community, or external/world-at-large community?

RQ4 Of the RSE projects / repositories used by the community, which are still actively developed or maintained? Are there unmaintained projects with community use? If so, do we have any sense of what skills are needed to maintain them?

In answering these questions through previously established exploratory approaches, we also contribute new findings around the use and efficacy of a modern large language model as an aid to automatically identify Research Software Engineering projects at an empirical scale.

Finally, in order to cover as much ground as possible within existing time constraints, we did limit ourselves to GitHub repositories associated with US research universities for this work. We look forward to examining non US universities, industry research labs, repository sources other than GitHub – and within GitHub non-repository pages (e.g., institute owner pages), and so forth in future papers.

II. Related Work

In examining the literature, there is relatively little directly comparable work in this space [2]. That said, there is some work done that helps orient this paper within the literature. For example, [3] and [4], both of which explore ways to link research papers and their associated software repositories.

However, we can compare to [1], on RSE GitHub repositories in US Department of Energy national laboratories. We rely heavily on the approach used in [1] in this work but reapply it to the research university context. In [1] the authors scraped GitHub and national laboratory websites, among other approaches, to develop an inventory of lab-linked software repositories. Novel to this work, which also incorporates a large language model (ChatGPT) to classify repositories as RSE or not RSE, we compare our findings the results in [5] and [6] which also used ChatGPT to classify highly specialized topics in medicine and GitHub repositories, respectively.

Once we had identified RSE projects, we did analysis of their characteristics. To help ground our work, we can look to papers like [7] and [8] which discusses the relationship between stars and forks in open source GitHub repositories. This is augmented by work like [9] which, among other findings, illustrates that stars and forks are challenged as useful metrics for community existence when projects are distributed across multiple repositories.

III. RQ1: Projects with a nexus to a major US research universities

A. Definitions, Initial Scope, Overarching Approach:

Our goal is to cast a wide net when answering our formal research question, "RQ1: What are all of the public software projects / repositories with some nexus, even a weak one, to the major research universities in the United States?" That said, we do define some scoping parameters. First, we will limit our search to projects ultimately hosted on GitHub. We also define "major research university" as "R1" universities under the Carnegie Classification of Institutions of Higher Education.

We identify universities meeting the R1 classification by relying on the data reported by the universities themselves to the federal government's Integrated Postsecondary Education Data System (IPEDS) run by the National Center for Education Statistics. All institutions of higher education are required to submit information to this system on an annual basis. We used the IPEDS data from the 2022 reporting cycle, the most recent available, to identify 146 R1 universities to evaluate. This data set also contained information about a university's institutional characteristics, such as the website of each university, and self-reported aliases and nicknames (e.g., "MIT" for the Massachusetts Institute of Technology).

To identify software projects associated with these research universities we deployed two approaches. (1) We first scraped the websites of each of the universities, looking for links to GitHub repositories. (2) We then searched GitHub itself for repositories by using keywords associated with each of the universities.

B. Approach 1: Scraping university websites

On December 27, 2023 we deployed a webcrawling spider on all of the 147 R1 universities, using the root domain extracted by the self-reported URL from the 2022 IPEDS data as a starting point. This spider, which obeyed the robots.txt file from each university, scoured the institutional websites using a breadth-first search traversal of links on HTML pages. The spider ran until all linkable HTML pages had been traversed or 200 hours had passed. On each page, the spider scraped any link to a GitHub repository. These links were subsequently cleaned and filtered for repository links of the form `github.com/owner/repository`. Duplicates were consolidated into a set of unique repositories.

1) Results: Of the 146 R1 universities, 116 (70%) allowed scraping of their website. The rest of this analysis focuses on these 116 universities.

TABLE I
Comparison of repositories found in the US Department of Energy National Laboratory from the data obtained in writing [1] and US R1 Universities that permitted website scraping.

Comparison Group:	National Labs	R1 Universities
Number of institutions in group	17	116
Mean number of URLs scraped per institutional website	306k	384k
Minimum number of unique repositories found on a institutional website. *Note: A 0 means there was at least one institution.gov/institution.edu domain which had no links to GitHub repositories at all.*	0	0
Mean number of unique repositories found on institutional websites	94	58
Median number of unique repositories found on institutional websites	35	16
Maximum number of unique repositories found on a institutional website	482	951

Overall, 44,566,662 distinct URLs were traversed, and average of 384,195 per site. A total of 10,056 unique GitHub repositories were pointed to by one or more links across all of the 116 universities that permitted website scraping.

We compare these raw findings with the national laboratory study [1] in Table I.

C. Approach 2: Searching on GitHub

We also used GitHub itself to search for keywords associated with each university to find affiliated repositories, similar to the approach used in [1].

The keywords used were lowercase versions of (1) the official name of the university (e.g., "University of Oregon"); (2) aliases drawn from self-reported IPEDS data (e.g., "UO,") or derived from the domain name (e.g., "UOregon") – which was feasible since GitHub searches are not case sensitive; and (3) the domain name itself (e.g., "uoregon.edu"). To provide consistency, we examined only the 116 universities which allowed web scraping.

GitHub helpfully provides a command line interface which provides the same search results (up to 1000 results) as its website's search function, which sorts the results by GitHub's self-described "best match." We applied the command `$ gh search repos keyword --limit 1000 --json url` to each of the keywords identified for each institution, which resulted in a list of up to 1,000 repositories for each keyword search.

1) Results: Across all 116 universities, 187,728 unique repositories were identified through keyword searches. However, unlike in [1], we were unable to manually examine each repository and make a human determination if the project was a university-affiliated RSE repository or not. There were just too many results given time constraints.

However, we did carry out a cursory examination of several keyword search results. We found that some keywords resulted in many repositories which had little connection to either research or universities. This was particularly true for keywords associated with state or flagship public universities,

whose keywords often were identical to the name of the state in which they were located. For example, when we examined the keyword "Arizona," we found the query produced several search results which were clearly linked to research software projects at either the Arizona State University and the University of Arizona. These results also included repositories related to tourism in Arizona, a gaming engine framework happened to be called Arizona, and other non-RSE projects. That said, we observed that the overwhelming majority RSE projects found by searching with broad keywords like "Arizona" were, indeed, associated with one of the universities in Arizona.

Our next question then, became, "How do we identify which of these 187,728 repositories are RSE projects? For that matter, which of the 10,056 links to GitHub repositories found in Approach 1 are RSE projects? How can we do this in an automated fashion?"

IV. RQ2: Is a given repository an RSE repository?

To determine whether a repository is an RSE repository at scale, we turned to OpenAI's GPT 3.5 Large Language Model (ChatGPT). After exploring several different prompt templates, we settled on a format similar to the approach used by [6]. Namely, we asked ChatGPT to determine whether a repository was an RSE repository by providing a definition for RSE, providing a description of the repository, and, in breaking with [6]'s requested binary yes-no output, we asked here for a probability from 0 to 1 that the repository was RSE. We did this because exploratory work with a handful of repositories suggested that relaxing to a continuous case led to more consistent and easily parsable responses than prompts which asked for a binary yes-no answer.

The definition for "RSE" came from https://us-rse.org/about/what-is-an-rse/. We created a description of each repository by piping the text obtained from the GitHub CLI command `gh repo view owner/repository`, which includes the repository's name, brief description, and the contents of the README document if available. Putting it all together, the prompt followed this template:

> Consider the following definition of a Research Software Engineer:
>
> "We like an inclusive definition of Research Software Engineers to encompass those who regularly use expertise in programming to advance research. This includes researchers who spend a significant amount of time programming, full-time software engineers writing code to solve research problems, and those somewhere in-between. We aspire to apply the skills and practices of software development to research to create more robust, manageable, and sustainable research software."
>
> Consider the following information about a Github repository:
>
> ===
> "View Repo" Data Inserted Here
> ===
>
> Given the above definition and GitHub repository information, is this GitHub repository a research software engineering project? Report your answer as a probability between 0 and 1. Place this number at the end of the message.

To validate this approach, we randomly selected two sets of 385 repositories, which gives a margin of error of $\pm 5\%$ on a 95% confidence interval. One set of 385 randomly selected repositories came from the pool of repositories found by scraping university websites ("scrape set") and one set of 385 from the pool of repositories found by searching GitHub for keywords ("search set"). The first author manually identified each repository in both of these sets as an RSE or non-RSE repository in light of the description above.

In the case of the scrape set, 227 of 385 (59%) were manually identified as RSE repositories. In the case of the search set, only 50 of 385 (13%) were manually identified as RSE repositories.

In both of the sample sets, the last number of each outputted message given the above prompt to ChatGPT was, indeed, a figure between 0 and 1 which corresponded to the likelihood of the repository being an RSE project or not. This number was extracted and paired with each repository.

Consequently, we had rows data, where each row was of the form [repo-name, human-assigned-boolean-value, chat-gpt-assigned-probability].

To convert the ChatGPT assigned probability to a Boolean value, we needed to apply a threshold. E.g., [repo-name, human-assigned-boolean-value, chat-gpt-assigned-probability $>$ threshold].

To determine this threshold, we looked for values which would maximize a sense of agreement across the distribution between the human-labeled data and the ChatGPT labeled data. Specifically, we looked at Cohen's Kappa, a common measure of inter-rater reliability. We also looked at F1 score, as a different measure to balance precision and recall. Across both data sets, Cohen's Kappa and F1 score was maximized (or nearly maximized) when the threshold was set at 0.75. Results summarizing this agreement are in Table II.

These results are not perfect, but they seem to align with the results in the very little existing work around the classification of niche and technical topics using ChatGPT, such as in [6]. In this work the authors explored the empirical identification of malicious repositories that are nominally educational. They found that in "gray cases" ChatGPT couldn't even agree with itself more than 90% of the time. In [5] the authors examined whether ChatGPT gave accurate yes-no answers to niche medical questions and found simple percentage accuracy of around 80%. Given this context, the agreement results we

TABLE II
SUMMARY OF HUMAN AND CHATGPT AGREEMENT WHEN LABELING REPOSITORIES IN TWO DIFFERENT SETS AS RSE OR NON-RSE

Data Set:	Scrape	Search
Simple percentage agreement between human and ChatGPT RSE determination.	77%	90%
Cohen's Kappa (Agreement levels of None, Minimal, Weak, Moderate, Strong, Almost Perfect [10])	0.5 (Weak)	0.6 (Moderate)
F1 Score	0.82	0.66

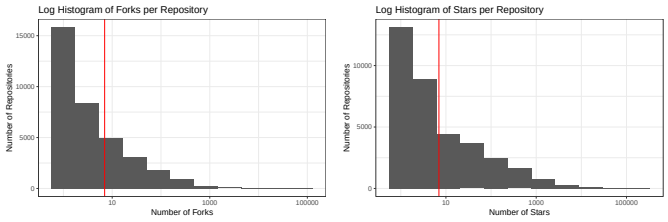

Fig. 1. Histograms of the number of stars and forks per RSE repo. The red line, six in both cases, is a threshold used in [1] to indicate an RSE repository has a community. These histograms seem to follow that same trend.

report here seem within the realm of state-of-the-art when using ChatGPT for classifying concepts which largely revolve around niche technical information and fuzzy definitions, although it is also clear there is ample room for improvement.

A. Results

We applied the above prompt to the combined 193,921 repositories from the union of the "scrape" and "search" data sets.

There were a few errors. In 615 cases (0.317%), GitHub could not find the repository. This was due to the repository being deleted or an error in the repository's name from a link that was scraped. In 19 cases (0.010%), ChatGPT could not assign a probability. This was due to a lack of description or README about the repository. In 579 cases (0.299%), ChatGPT errored out, saying the prompt had an issue. This was always due to the project's README having a non-standard character encoding, a binary file uploaded as the README, or an excessively long length.

In total, there were errors in 1,213 cases (0.626%). All of these repositories which encountered errors during processing were set aside. In the case of 192,708 repositories (99.374%) we were able to assign an RSE probability between 0 and 1. Applying our threshold of 0.75 to be considered RSE, we and ChatGPT ultimately identified 35,361 repositories (18.2%) as RSE repositories under the US-RSE definition. These results are disaggregated by university and shown in Table IV. We also provide summary statistics about the number of repositories and RSE repositories found in Table III

TABLE III
SUMMARY STATISTICS (MINIMUM, 25%, MEDIAN/50%, MEAN, 75%, AND MAXIMUM) OF THE NUMBER OF REPOSITORIES WITH A NEXUS TO A UNIVERSITY, AND THE NUMBER OF RSE REPOSITORIES, FOUND ACROSS ALL UNIVERSITIES.

	All Repositories	RSE Repositories
Min.	131	14
1st Qu.	1093	127
Median	1495	261
Mean	1796	334
3rd Qu.	2353	417
Max.	11856	2117

We also did overlap analysis. That is, which universities had repositories in common? Given our set of 116 universities, this resulted in $\binom{116}{2}$ = 6,670 pairs of universities to examine. Of these 6,670 pairs, 5,142 (77%) had no repositories in common at all, RSE or otherwise. 6,101 of the 6,670 (91%) had no RSE repositories in common. In the vast majority of these 9% of pairs with some RSE repositories in common, most institutions had just one or two. Understanding why overlap is so sparse, and the relationships among those universities which do have overlap, remain as future work.

V. RQ3: POPULARITY OF RSE PROJECTS

Of the RSE projects / repositories found through answering RQ2, which ones are likely to be popular outside of the project's core developers? That is, which projects are used by the community – either a university-internal community, or external/world-at-large community? We tackle this question by first engaging in a histogram analysis of the stars and forks of the 35,361 RSE projects identified in § IV. These visualizations can be found in Figures 2 and 1.

To do this analysis we obtained additional meta information about these projects. We obtained meta-data on each of them by using the command-line GitHub API tool with the command `gh api -H "Accept: application/vnd.github+json" -H "X-GitHub-Api-Version: 2022-11-28" /repos/OWNER/REPO` where `OWNER/REPO` is replaced with each name of the repository listed in the supplemental material. This metadata includes the number of stars and forks for a repository, as well as the timestamp of the last push to the repository.

In light of the threshold values of six stars or six forks or more as reasonable choices for defining an RSE project as one with community, and in line with previous work [1], we filter the 35,361 RSE projects accordingly. This resulted in 13,940 RSE repositories, or 39.4%, as being considered as having a community – a notably higher percentage than the national laboratory case, which had only 25% of it's RSE repositories found to have had a community. Understanding the "why" is important future work, and will likely involve individual case studies or analysis of the number of contributers, as projects in universities may have more student volunteers over time than projects in national laboratories, thus artificially increasing the number of repositories with community sizes in the 6-16 people range. This, however, is speculation.

TABLE IV: All GitHub repositories found by university, through both scraping .edu websites for links to GitHub and by searching for keywords related to the university on GitHub itself. The number of RSE repositories, as determined by ChatGPT provided probabilities with a threshold of at least a 75% likelihood, is also shown.

	Scraping .edu websites			**Searching on GitHub**			**Scraping ∪ Searching**		
University(.edu)	**Total**	**RSE**	**(%)**	**Total**	**RSE**	**(%)**	**Total**	**RSE**	**(%)**
All	**6346**	**4453**	**70.2%**	**187728**	**30971**	**16.5%**	**193921**	**35361**	**18.2%**
arizona	81	42	51.9%	4146	1116	26.9%	4222	1158	27.4%
asu	10	7	70.0%	1371	130	9.5%	1381	137	9.9%
auburn	25	1	4.0%	1036	101	9.8%	1061	102	9.6%
bc	0	0	0.0%	1268	313	24.7%	1268	313	24.7%
binghamton	29	12	41.4%	434	45	10.4%	463	57	12.3%
brandeis	10	2	20.0%	490	103	21.0%	499	105	21.0%
brown	57	48	84.2%	1511	373	24.7%	1568	421	26.9%
bu	63	33	52.4%	3078	518	16.8%	3141	551	17.5%
buffalo	45	34	75.6%	2363	329	13.9%	2408	363	15.1%
case	17	15	88.2%	2391	488	20.4%	2408	503	20.9%
clemson	30	21	70.0%	782	147	18.8%	812	168	20.7%
cmu	132	113	85.6%	2361	746	31.6%	2492	858	34.4%
colorado	6	3	50.0%	11850	2114	17.8%	11856	2117	17.9%
columbia	49	27	55.1%	1871	404	21.6%	1919	431	22.5%
cornell	773	489	63.3%	1595	389	24.4%	2357	876	37.2%
dartmouth	75	45	60.0%	1165	213	18.3%	1239	258	20.8%
drexel	37	22	59.5%	1019	151	14.8%	1056	173	16.4%
du	0	0	0.0%	1616	269	16.7%	1616	269	16.7%
duke	23	18	78.3%	1909	246	12.9%	1932	264	13.7%
emory	28	20	71.4%	847	145	17.1%	874	165	18.9%
fiu	1	1	100.0%	1056	66	6.3%	1057	67	6.3%
fsu	25	19	76.0%	2498	213	8.5%	2523	232	9.2%
gatech	67	59	88.1%	2025	347	17.1%	2092	406	19.4%
georgetown	75	44	58.7%	1643	243	14.8%	1717	286	16.7%
harvard	110	69	62.7%	2470	397	16.1%	2578	466	18.1%
iastate	71	40	56.3%	1558	120	7.7%	1627	160	9.8%
illinois	16	12	75.0%	4333	581	13.4%	4348	592	13.6%
indiana	2	1	50.0%	2335	204	8.7%	2337	205	8.8%
jhu	6	1	16.7%	2747	311	11.3%	2752	312	11.3%
k-state	5	4	80.0%	1138	155	13.6%	1143	159	13.9%
ku	23	13	56.5%	1593	229	14.4%	1616	242	15.0%
louisiana	0	0	0.0%	1889	331	17.5%	1889	331	17.5%
louisville	0	0	0.0%	1259	60	4.8%	1259	60	4.8%
lsu	15	5	33.3%	1031	111	10.8%	1046	116	11.1%
memphis	4	1	25.0%	744	61	8.2%	748	62	8.3%
miami	3	1	33.3%	3312	713	21.5%	3315	714	21.5%
missouri	0	0	0.0%	2542	447	17.6%	2542	447	17.6%
mit	547	436	79.7%	2088	500	24.0%	2628	933	35.5%
msu	27	10	37.0%	3547	584	16.5%	3573	593	16.6%
ncsu	67	31	46.3%	2504	405	16.2%	2571	436	17.0%
nd	46	31	67.4%	1741	459	26.4%	1785	490	27.5%
ndsu	0	0	0.0%	243	42	17.3%	243	42	17.3%
njit	1	1	100.0%	1042	74	7.1%	1043	75	7.2%
northeastern	0	0	0.0%	1505	220	14.6%	1505	220	14.6%
northwestern	200	94	47.0%	1657	272	16.4%	1856	365	19.7%

TABLE IV: All GitHub repositories found by university, through both scraping .edu websites for links to GitHub and by searching for keywords related to the university on GitHub itself. The number of RSE repositories, as determined by ChatGPT provided probabilities with a threshold of at least a 75% likelihood, is also shown.

	Scraping .edu websites			**Searching on GitHub**			**Scraping ∪ Searching**		
University(.edu)	**Total**	**RSE**	**(%)**	**Total**	**RSE**	**(%)**	**Total**	**RSE**	**(%)**
odu	0	0	0.0%	1095	70	6.4%	1095	70	6.4%
ohio	5	3	60.0%	2382	340	14.3%	2387	343	14.4%
okstate	3	1	33.3%	1350	153	11.3%	1353	154	11.4%
olemiss	2	0	0.0%	129	14	10.9%	131	14	10.7%
oregonstate	158	108	68.4%	2084	202	9.7%	2242	310	13.8%
osu	13	7	53.9%	1299	174	13.4%	1312	181	13.8%
ou	2	1	50.0%	2101	349	16.6%	2103	350	16.6%
pitt	67	39	58.2%	1481	188	12.7%	1547	227	14.7%
psu	19	13	68.4%	1982	291	14.7%	2000	303	15.2%
purdue	14	5	35.7%	2554	439	17.2%	2568	444	17.3%
rice	90	76	84.4%	1914	274	14.3%	2003	349	17.4%
rochester	5	3	60.0%	832	132	15.9%	837	135	16.1%
rpi	74	56	75.7%	1209	294	24.3%	1283	350	27.3%
rutgers	61	44	72.1%	2597	339	13.1%	2657	382	14.4%
sc	3	0	0.0%	2159	355	16.4%	2162	355	16.4%
stanford	234	184	78.6%	3819	982	25.7%	4050	1164	28.7%
syracuse	1	1	100.0%	2461	394	16.0%	2462	395	16.0%
tamu	11	8	72.7%	1077	97	9.0%	1088	105	9.7%
tufts	33	18	54.6%	1208	198	16.4%	1241	216	17.4%
tulane	28	17	60.7%	231	26	11.3%	259	43	16.6%
ua	19	16	84.2%	1776	553	31.1%	1793	567	31.6%
uab	1	0	0.0%	1015	59	5.8%	1016	59	5.8%
uah	0	0	0.0%	1015	90	8.9%	1015	90	8.9%
uark	1	1	100.0%	715	88	12.3%	716	89	12.4%
ucdenver	9	5	55.6%	1099	150	13.7%	1108	155	14.0%
uci	16	13	81.3%	2743	1150	41.9%	2758	1163	42.2%
ucla	80	65	81.3%	1132	217	19.2%	1210	281	23.2%
uconn	4	3	75.0%	949	103	10.9%	952	106	11.1%
ucr	10	4	40.0%	1282	174	13.6%	1291	178	13.8%
ucsb	91	57	62.6%	1143	230	20.1%	1230	286	23.3%
ucsd	64	49	76.6%	2278	383	16.8%	2338	430	18.4%
udel	2	1	50.0%	1483	279	18.8%	1485	280	18.9%
ufl	951	870	91.5%	2946	586	19.9%	3897	1456	37.4%
uh	59	45	76.3%	1275	209	16.4%	1334	254	19.0%
uic	8	1	12.5%	1290	137	10.6%	1297	138	10.6%
uiowa	16	8	50.0%	1611	295	18.3%	1626	303	18.6%
umaine	14	8	57.1%	187	32	17.1%	200	40	20.0%
umass	49	29	59.2%	1306	225	17.2%	1354	254	18.8%
umbc	20	1	5.0%	1030	96	9.3%	1043	97	9.3%
umd	582	336	57.7%	1256	229	18.2%	1837	565	30.8%
umich	35	16	45.7%	2567	694	27.0%	2602	710	27.3%
umn	9	1	11.1%	2120	311	14.7%	2129	312	14.7%
umt	1	0	0.0%	1114	108	9.7%	1115	108	9.7%
unc	3	1	33.3%	1085	415	38.3%	1088	416	38.2%
unh	138	120	87.0%	1050	114	10.9%	1188	234	19.7%
unlv	10	7	70.0%	511	50	9.8%	521	57	10.9%

TABLE IV: All GitHub repositories found by university, through both scraping .edu websites for links to GitHub and by searching for keywords related to the university on GitHub itself. The number of RSE repositories, as determined by ChatGPT provided probabilities with a threshold of at least a 75% likelihood, is also shown.

	Scraping .edu websites			**Searching on GitHub**			**Scraping ∪ Searching**		
University(.edu)	**Total**	**RSE**	**(%)**	**Total**	**RSE**	**(%)**	**Total**	**RSE**	**(%)**
unm	11	2	18.2%	1107	207	18.7%	1118	209	18.7%
uoregon	1	0	0.0%	1763	243	13.8%	1764	243	13.8%
upenn	68	45	66.2%	5730	1308	22.8%	5796	1351	23.3%
usc	10	7	70.0%	1299	260	20.0%	1309	267	20.4%
usf	9	8	88.9%	1140	163	14.3%	1149	171	14.9%
usm	0	0	0.0%	1016	55	5.4%	1016	55	5.4%
usu	11	8	72.7%	1107	66	6.0%	1118	74	6.6%
uta	2	0	0.0%	1141	131	11.5%	1143	131	11.5%
utah	1	0	0.0%	2695	526	19.5%	2695	526	19.5%
utdallas	76	60	79.0%	515	101	19.6%	590	161	27.3%
utep	0	0	0.0%	305	36	11.8%	305	36	11.8%
utexas	36	23	63.9%	1570	310	19.8%	1605	333	20.8%
utsa	2	0	0.0%	1027	30	2.9%	1029	30	2.9%
uwm	286	161	56.3%	876	90	10.3%	1161	251	21.6%
vanderbilt	24	18	75.0%	2717	320	11.8%	2740	338	12.3%
vcu	16	5	31.3%	1029	101	9.8%	1045	106	10.1%
virginia	4	1	25.0%	2350	300	12.8%	2354	301	12.8%
vt	8	7	87.5%	1626	486	29.9%	1634	493	30.2%
washington	91	63	69.2%	2262	413	18.3%	2353	476	20.2%
wayne	3	2	66.7%	2088	92	4.4%	2091	94	4.5%
wisc	25	12	48.0%	2392	578	24.2%	2416	590	24.4%
wsu	4	0	0.0%	1334	79	5.9%	1338	79	5.9%
wustl	6	1	16.7%	1197	223	18.6%	1203	224	18.6%
wvu	0	0	0.0%	684	70	10.2%	684	70	10.2%
yale	161	101	62.7%	1225	163	13.3%	1384	264	19.1%

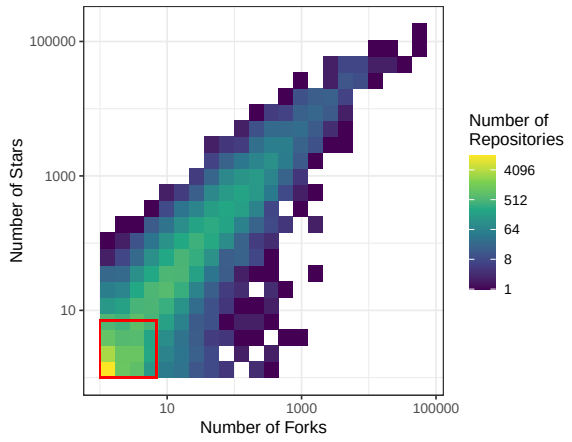

Fig. 2. Matrix view of the number of repositories per star-fork pair. The Pearson correlation coefficent of Stars and Forks is 0.865, which is in line with previous research. [11]

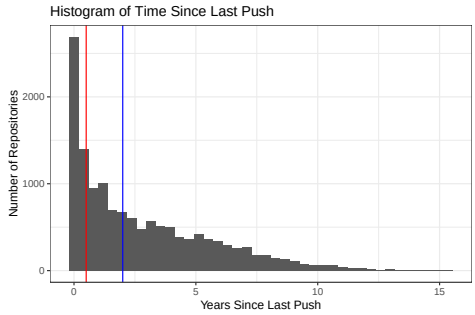

Fig. 3. Histogram of the last time an RSE project with community received a push. The red line indicates the six month mark. The blue line indicates the two year mark. These lines form the arbitrary threshold boundaries we selected for "healthy," "dying," and, "dead" repositories, which also match prior work [1].

VI. RQ4: Which RSE projects are active? Which are on life support? How do they differ?

The metadata pulled from the GitHub command line API in the previous section also contains data on when the last push to a repository occurred. We examined this "last push" timestamp on the 13,940 RSE projects we classify as having a community. We found that 3,805 (27%) repositories had a push in the last six months, 3,346 (24%) repositories had their last push more than six months ago but less than two years ago, and 6,784 (49%) of repositories had their last push more than two years ago. Using the broad buckets from [1], we classify these repositories as healthy (push in last six months), dying (push more than six months ago, less than two years ago), and dead (push more than two years ago). This is visualized in Figure 3.

Of increasing interest to the community is the importance of sustainability support for RSE projects [12]. To determine the crudest sense of what technical skills are needed for sustainability support for these projects, particularly "dying" projects, we examined the most common language used in each project according to the GitHub metadata. We examined all languages which had at least 1% prevalence in a partition. The three most common top languages in RSE repositories, across all categories of "healthy," "dying," and "dead" included Python, Jupyter Notebook, and C++. We show these full rankings in Tables V, VI, and VII.

TABLE V
Top languages of healthy RSE repos:

	Language	RSE Repositories	Percentage
1	Python	1211	33%
2	C++	472	13%
3	Jupyter Notebook	261	7%
4	C	229	6%
5	JavaScript	158	4%
6	Java	142	4%
7	Go	136	4%
8	Rust	121	3%
9	R	99	3%
10	TypeScript	95	3%
11	HTML	84	2%
12	C#	76	2%
13	Shell	72	2%
14	MATLAB	45	1%
15	Julia	43	1%
16	Ruby	43	1%

TABLE VI
Top languages of dying RSE repos:

	Language	RSE Repositories	Percentage
1	Python	1285	40%
2	Jupyter Notebook	351	11%
3	C++	318	10%
4	JavaScript	165	5%
5	C	163	5%
6	Java	120	4%
7	HTML	100	3%
8	MATLAB	73	2%
9	R	70	2%
10	C#	65	2%
11	Shell	53	2%
12	Rust	50	2%
13	TypeScript	48	1%
14	Go	39	1%

The three most common top languages in RSE repositories, across all categories of "healthy," "dying," and "dead" included Python, Jupyter Notebook, and C++. We show these full rankings in Tables V, VI, and VII.

These findings suggest that the skills needed for sustainability support for the vast majority of vulnerable RSE projects is reflective of the skills used in the development of RSE projects generally. Better understanding the attributes of these projects from a code/repository analysis perspective, and how it compares with reports from RSE developers such as in [12] is important future work.

VII. Future Work

There remains ample future work in this area. In §III, for example, different approaches to identifying university affiliated RSE projects can certainly be explored. For example, first identifying GitHub users working at research universities via self-defined indicta of institutional affiliation (e.g., email address), and subsequently attempting to identify if any of these users' repositories are RSE projects. Even in this work, we note that of the 147 R1 universities identified in the IPEDS data only 116 were examined due to website scraping policies. These remaining unexplored 31 universities

TABLE VII
TOP LANGUAGES OF DEAD RSE REPOS:

	Language	RSE Repositories	Percentage
1	Python	2084	32%
2	C++	780	12%
3	Jupyter Notebook	741	11%
4	C	479	7%
5	Java	316	5%
6	JavaScript	305	5%
7	MATLAB	181	3%
8	Matlab	163	3%
9	R	160	2%
10	HTML	156	2%
11	C#	125	2%
12	Shell	118	2%
13	Go	71	1%

certainly warrant further investigation. Better understanding of the interactions between repositories with homes at multiple different universities is also an area of interesting work. Better identification of RSE projects using large language models is certainly needed and a major threat to the validity of our work. Lastly, there are many further analyses of the projects identified as RSE that can be done to help identify characteristics of these repositories compared to other types of open source software projects.

VIII. CONCLUSION

In this paper we applied two different approaches to obtain a set of open source GitHub repositories at all R1 US research universities which allowed website scraping. We then used a GitHub repository's description and README file in a prompt to ChatGPT to identify 35,361 research software engineering repositories of the 193,921 repositories identified as having some possible nexus to a research university. ChatGPT agreed 77%-90% of the time with a human's determination of whether a GitHub repository was an RSE software repository.

Identifying and benchmarking the number of RSE repositories out there associated with US research universities is of key relevance to this community, as it permits comparative work going forward and a pool of specific, identifiable repositories on which other empirical analyses can be carried out.

As an example of one such analysis we found that, of these 35,361 RSE projects, it's plausible that at least 13,940 (39.4%) have some community use outside the core development team. Of these repositories with community, only 3,805 (27%) had received a push in the last six months, suggesting many RSE projects could use sustainability support. Across all RSE projects linked to a research university, we also found the top languages were Python, C++, and Jupyter Notebook.

There are certainly additional analyses we could explore going forward, such as examining productivity metrics (e.g., issues) of RSE repositories. Bottom line, by inventorying RSE repositories at US research universities we have unlocked a rich new horizon for future exploratory work in the empirical analysis of these repositories. We look forward to reporting our exploratory findings in subsequent papers.

REFERENCES

[1] S. D. Schwartz, S. F. Fickas, B. Norris, and A. Dubey, "A survey of open source software repositories in the us department of energys national laboratories (under review)," 2024.

[2] S. D. Schwartz, "A five-year survey of literature in software engineering and repository mining research," in *Research Reports, Department of Computer Science, University of Oregon*, 2023.

[3] D. Garijo, M. Arroyo, E. Gonzalez, C. Treude, and N. Tarocco, "Bidirectional paper-repository tracing in software engineering," in *Proceedings of the 21st International Conference on Mining Software Repositories*, ser. MSR '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 642646. [Online]. Available: https://doi.org/10.1145/3643991.3644876

[4] S. Wattanakriengkrai, B. Chinthanet, H. Hata, R. G. Kula, C. Treude, J. Guo, and K. Matsumoto, "Github repositories with links to academic papers: Public access, traceability, and evolution," *Journal of Systems and Software*, vol. 183, p. 111117, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121221002144

[5] B. Koopman and G. Zuccon, "Dr ChatGPT tell me what I want to hear: How different prompts impact health answer correctness," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 15 012–15 022. [Online]. Available: https://aclanthology.org/2023.emnlp-main.928

[6] M. R. Masud and M. Faloutsos, "Unveiling a hidden risk: Exposing educational but malicious repositories in github," 2024.

[7] H. Borges, A. Hora, and M. T. Valente, "Understanding the factors that impact the popularity of github repositories," in *2016 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2016, pp. 334–344.

[8] M. A. Moid, A. Siraj, M. F. Ali, and A. O. Amoodi, "Predicting stars on open-source github projects," in *2021 Smart Technologies, Communication and Robotics (STCR)*, 2021, pp. 1–9.

[9] M. Biazzini and B. Baudry, ""may the fork be with you": novel metrics to analyze collaboration on github," in *Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics*, ser. WETSoM 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 3743. [Online]. Available: https://doi.org/10.1145/2593868.2593875

[10] M. L. McHugh, "Interrater reliability: the kappa statistic." 2012.

[11] K. Yamamoto, M. Kondo, K. Nishiura, and O. Mizuno, "Which metrics should researchers use to collect repositories: An empirical study," in *2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*, 2020, pp. 458–466.

[12] J. C. Carver, N. Weber, K. Ram, S. Gesing, and D. S. Katz, "A survey of the state of the practice for research software in the united states," in *PeerJ Computer Science 8:e963*, 2022.