UCD Application Developer Survey 2016

Prepared by: Christopher Thielen, Lead Application Developer DSS IT

March 23, 2016

EXECUTIVE SUMMARY

Objective

The federated nature of UC Davis creates challenges in collaboration, resource sharing, and mentorship for developers on campus. This survey was designed to better understand the current environment from a developer's perspective.

Response Summary

Teams

(Developers Per Project, Team Size, Application Count, Uses Designers)

- Projects tend to have 2-6 developers though a single developer per project is not atypical.
- Almost 2/3s of developers support 6+ applications.
- Designers are rarely used.

Technology Stack

(Language Usage, Database Usage, Front-end Javascript Development, Cloud Hosting)

- Java, C#, PHP, and Python top the list with a notable amount of legacy ColdFusion.
- Oracle, MySQL, and SQL Server make up most database usage. NoSQL and other newer forms are not widely used.
- Web front-ends consisting of jQuery or a more sophisticated framework are used by about 2/3s of respondents.
- Cloud hosting is not widely used.

Software Lifecycle

(VCS Usage, Test Usage, Monitoring, Security Scanners)

- Git is very popular.
- Testing is largely not performed.
- Around 3/4s of respondents use some form of application monitoring.
- Security scanners are not widely used.

Community

(Campus Data Source Usage, Open Source Practices, Existing Collaboration, Community Engagement, Sense of Connectedness, Interest in Community)

- Public open source participation is low though a majority of respondents are interested.
- Most teams do not collaborate with other teams often.
- Around 2/3s of respondents engage in some form of community (mailing lists, etc.)
- Sense of connectedness with other developers is very poor.
- Interest in building community is strong.

Team Size vs. Software Lifecycle Practices

- Larger team size correlates with most software lifecycle practices.

Conclusions

(Opinions of survey report author.)

UC Davis commits an estimated $27 million in salary investment to software development on campus but the practices currently utilized do not adequately realize that investment. Development is largely performed along the same divisions of the University's federated structure, resulting in severe strategy separation, redundant effort, and a hindered developer community. Opportunities for collaboration are not widely grasped and the resulting duplicative effort can be found in areas as small as specific software routines up to entire development products.

It should be noted the organizational separation is likely responsible for the positive effect of allowing greater tool choice and autonomy within the development community. It has been theorized that having many disparate teams enables those teams fit for rapid technology adoption to do so without hinderance, allowing the University a certain degree of agility.

Sophisticated development practices found in the modern Software Development Lifecycle (structured design, testing, monitoring, formal security policies, code retirement) are not widely adopted, likely due to inadequate team sizes. Current resource allocations demand an average of at least six applications per developer, suggesting applications receive insufficient coverage.

There are many proposals to address this problem:

- Consolidate development resources where appropriate (college-level, IT service center, or similar). This will enable specialization of talent (test engineers, front-end designers, development operations) to better target the needs of the Software Development Lifecycle.
- Greatly increase the quantity and affordability of shared services: the campus virtualization cluster as well as its dynamic analysis security scanner are beginnings toward this end but additions could be made such as a turnkey continuous integration service, easily-updatable starter templates for application projects (such as the "UCD Arch" bootstrap provided by the College of A&ES), and so on.
- Provide training certificates to mitigate the lack of available developer mentorship in small teams.
- Support community efforts to share source code, provide specific examples of popular subroutines, and maintain a directory of public subject-matter experts willing to answer questions.
- Create a specific on-boarding experience for new developers to quickly understand the resources and structures here at the University.

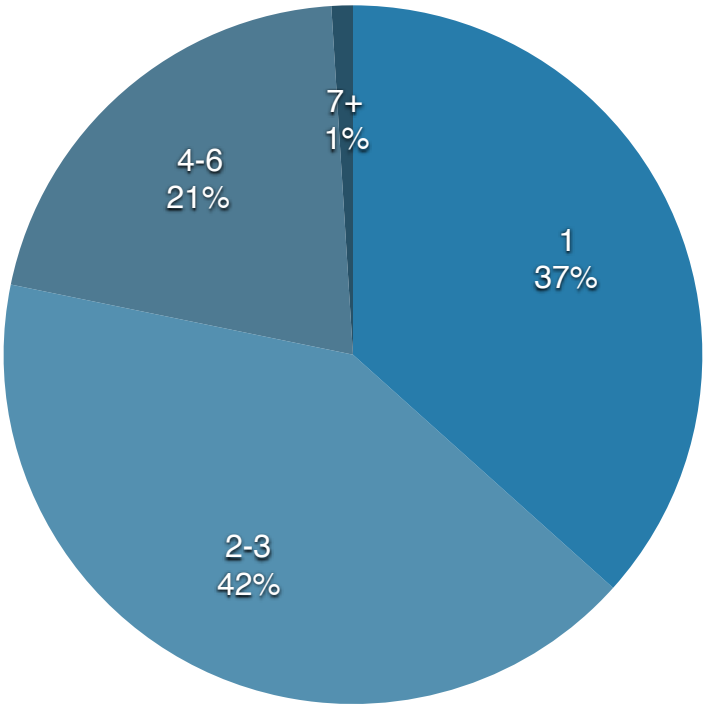
It is the intention of the Application Developer SIG to produce these survey results annually and track any changes to determine growth in these areas.

RESPONSES

Teams

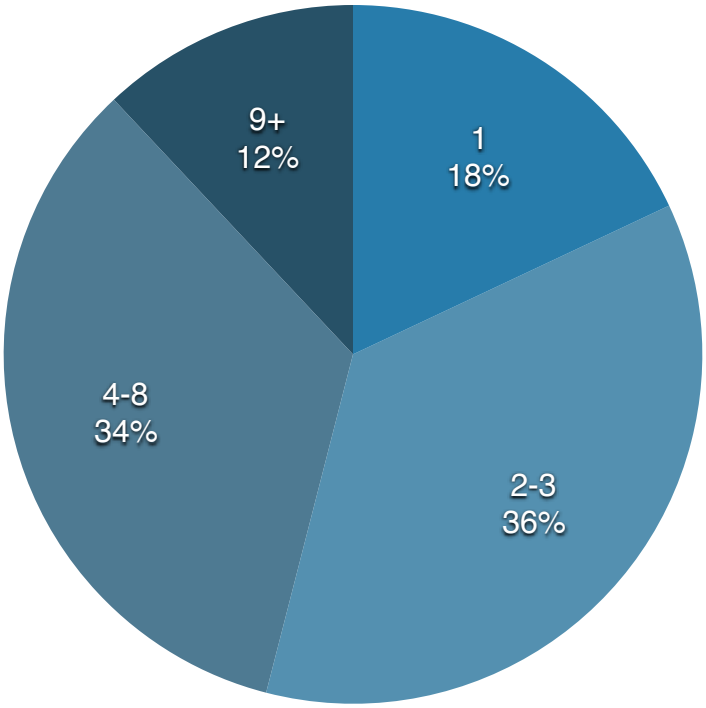
Developers Per Project

Answer	Response	%
1	48	37%
2-3	54	42%
4-6	27	21%
7+	1	1%
Total	130	100%



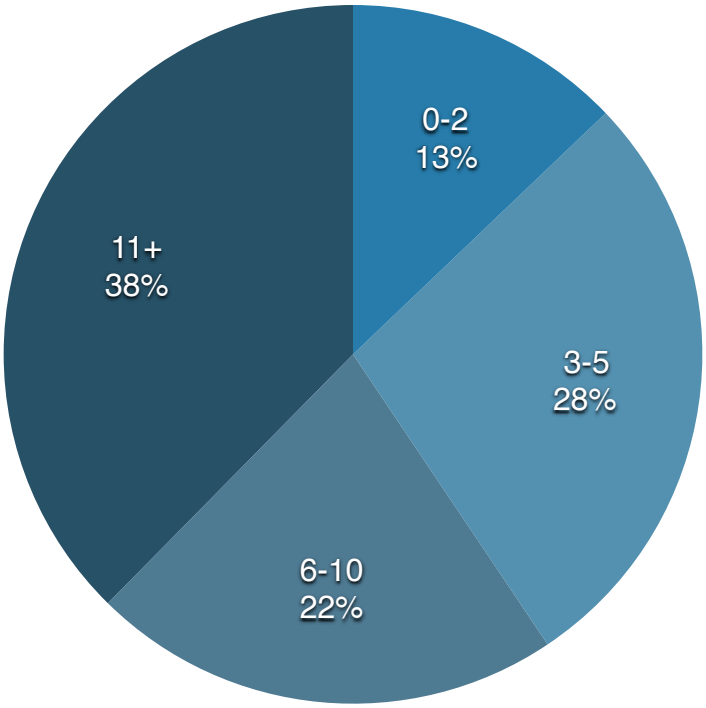
Team Size

Answer	Response	%
1	23	18%
2-3	46	36%
4-8	44	34%
9+	16	12%
Total	129	100%



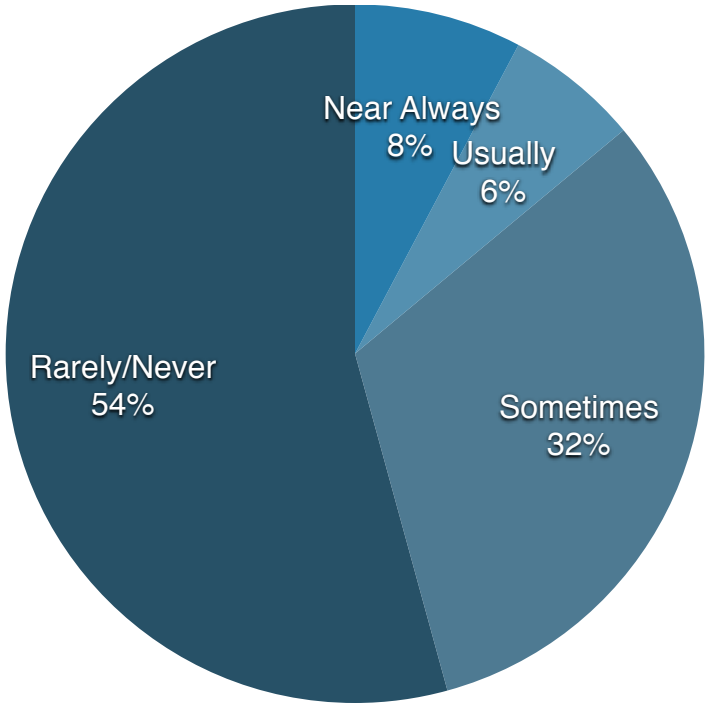
Application Count

Answer	Response	%
0-2	17	13%
3-5	36	28%
6-10	28	22%
11+	49	38%
Total	130	100%



Uses Designers

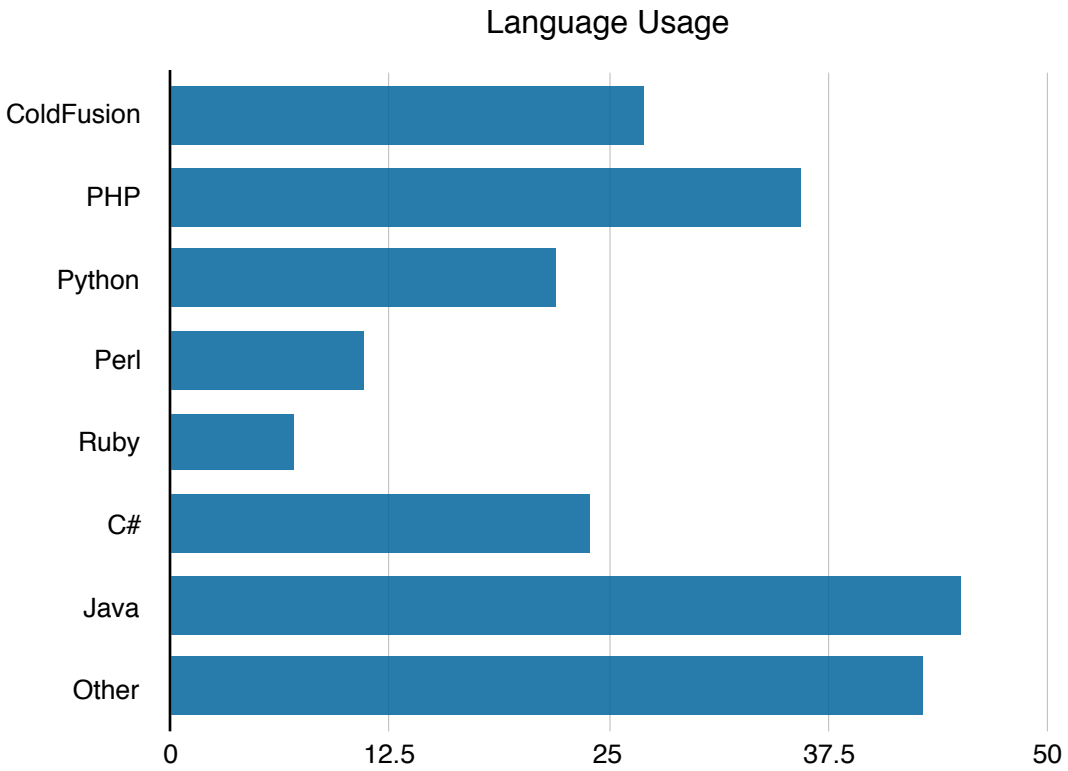
Answer	Response	%
Near Always	10	8%
Usually	8	6%
Sometimes	41	32%
Rarely/Never	70	54%
Total	129	100%



Technology Stack

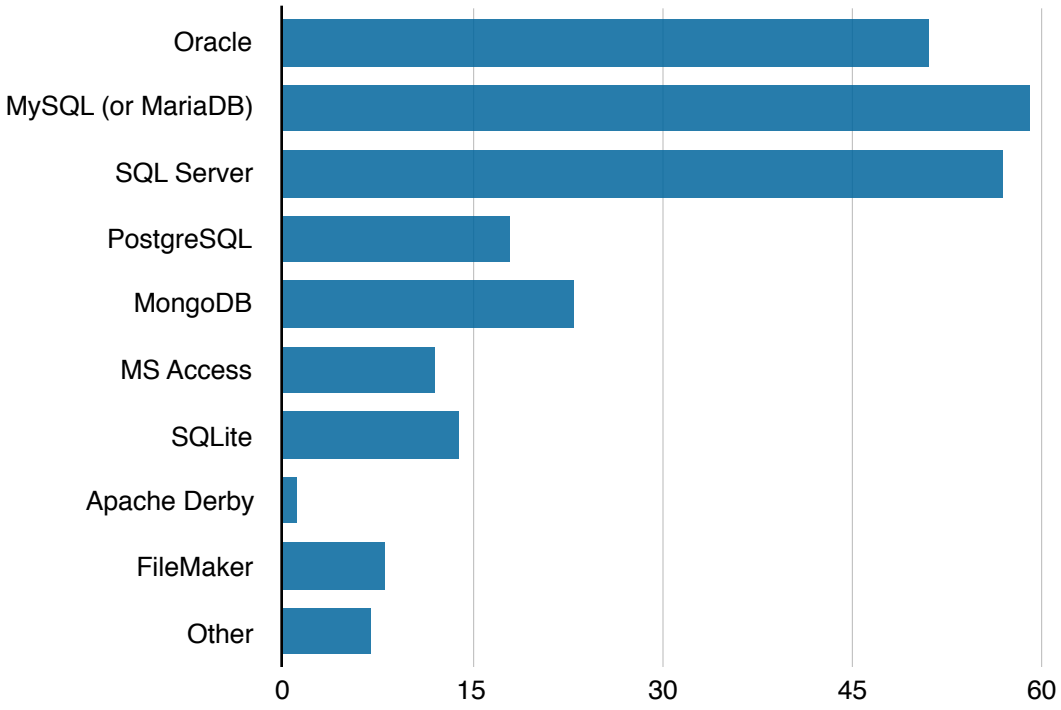
Language Usage

Answer	Response	%
ColdFusion	27	23%
PHP	36	30%
Python	22	18%
Perl	11	9%
Ruby	7	6%
C#	24	20%
Java	45	38%
Javascript	97	81%
Other	43	36%



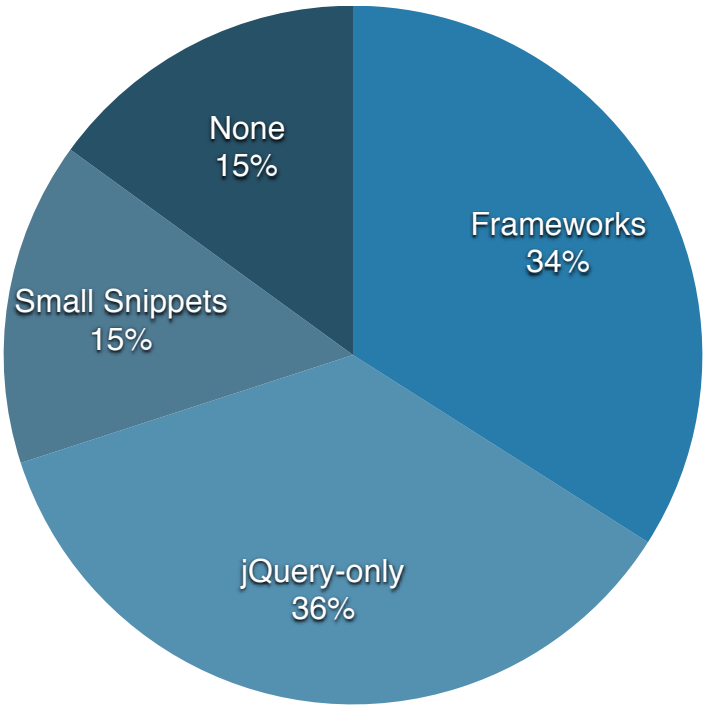
Database Usage

Answer	Response	%
Oracle	51	42%
MySQL (or MariaDB)	59	49%
SQL Server	57	47%
PostgreSQL	18	15%
MongoDB	23	19%
MS Access	12	10%
SQLite	14	12%
Apache Derby	1	1%
FileMaker	8	7%
Other	7	6%



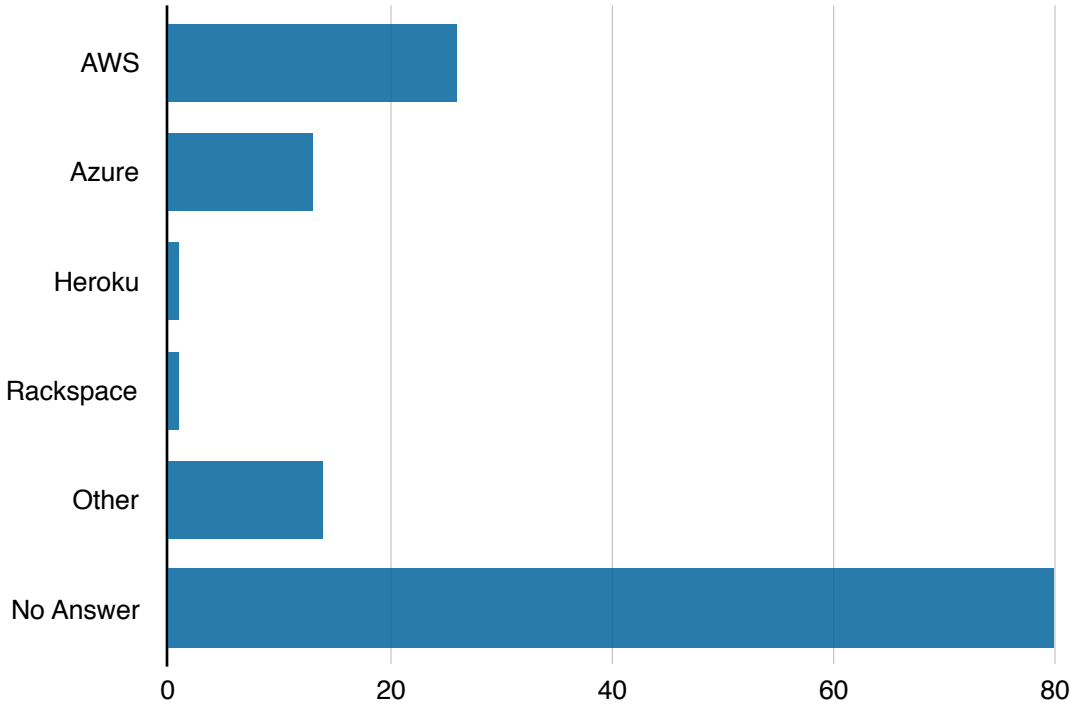
Front-end JS Development

Answer	Response	%
Frameworks	40	34%
jQuery-only	42	36%
Small Snippets	17	15%
None	18	15%
Total	117	100%



Cloud Hosting

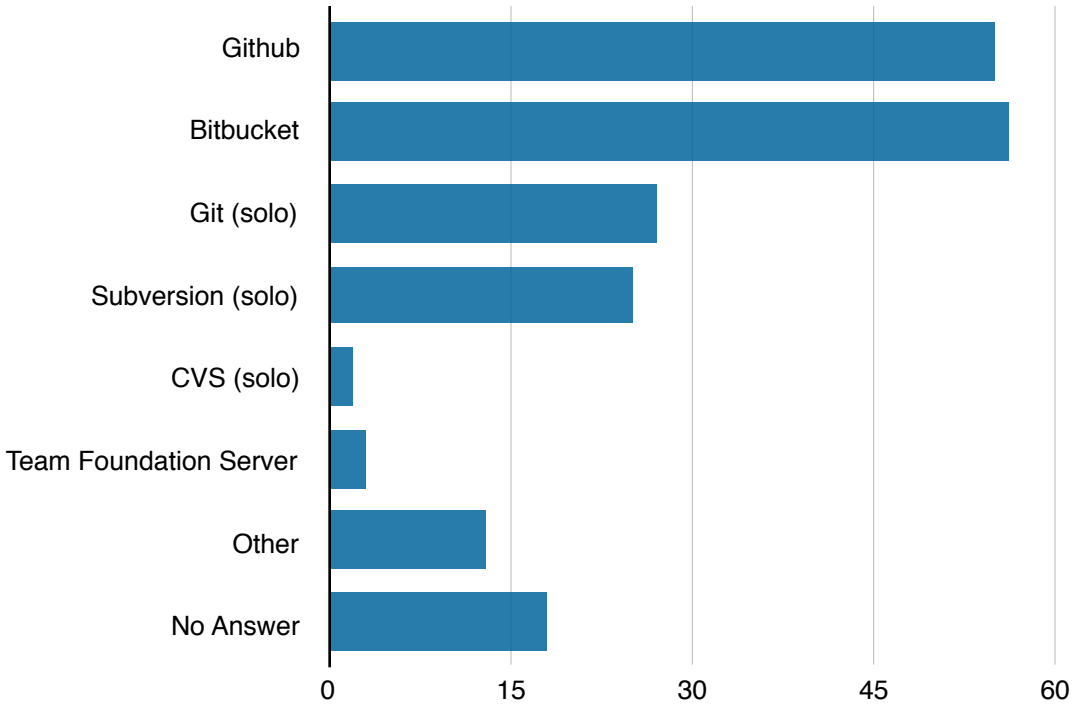
Answer	Response	%
AWS	26	57%
Azure	13	28%
Heroku	1	2%
Rackspace	1	2%
Other	14	30%
No Answer	80	



Software Lifecycle

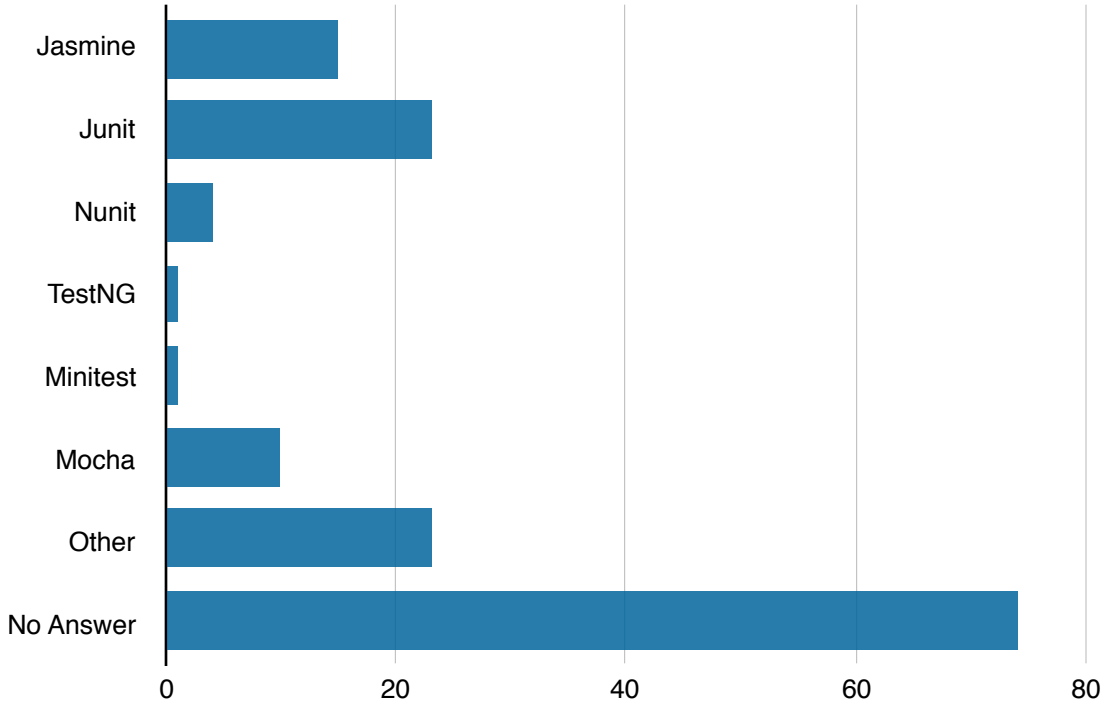
Version Control Usage

Answer	Response
Github	55
Bitbucket	56
Git (solo)	27
Subversion (solo)	25
CVS (solo)	2
Team Foundation Server	3
Other	13
No Answer	18



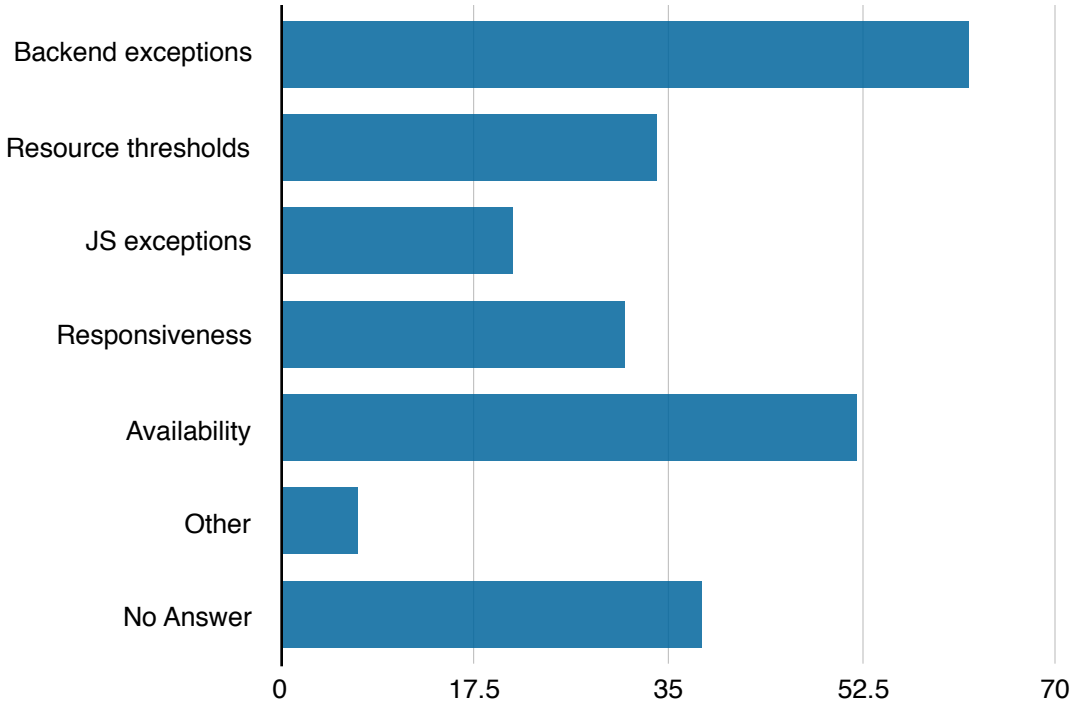
Test Usage

Answer	Response
Jasmine	15
Junit	23
Nunit	4
TestNG	1
Minitest	1
Mocha	10
Other	23
No Answer	74



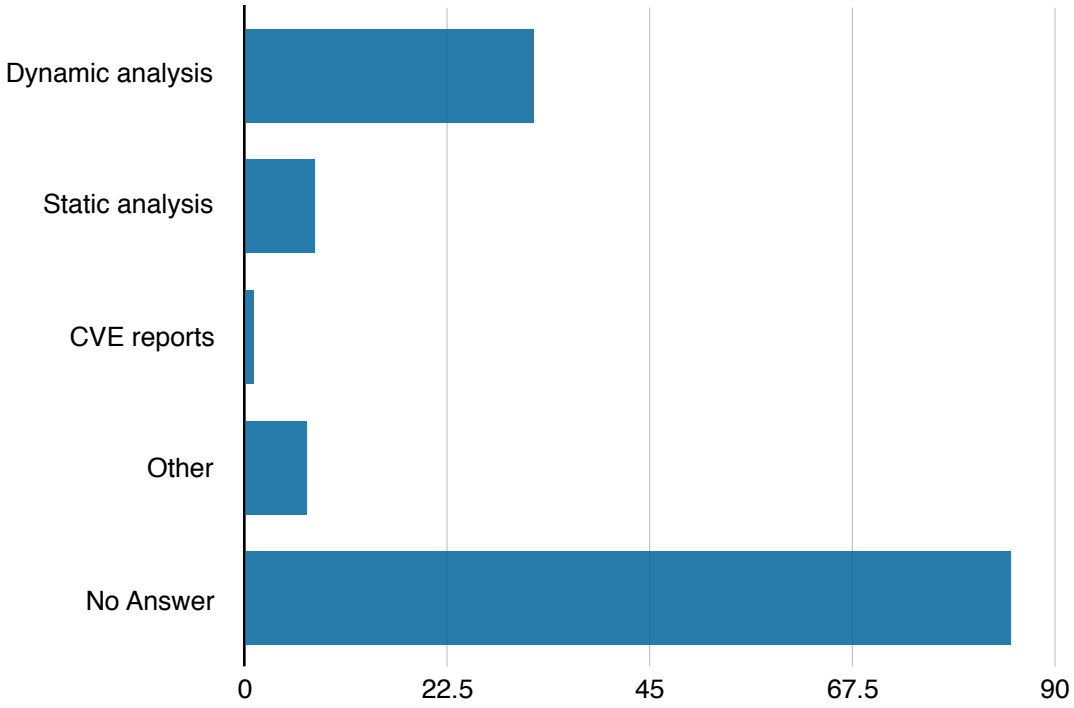
Monitoring

Answer	Response
Backend exceptions	62
Resource thresholds	34
JS exceptions	21
Responsiveness	31
Availability	52
Other	7
No Answer	38



Security Scanners

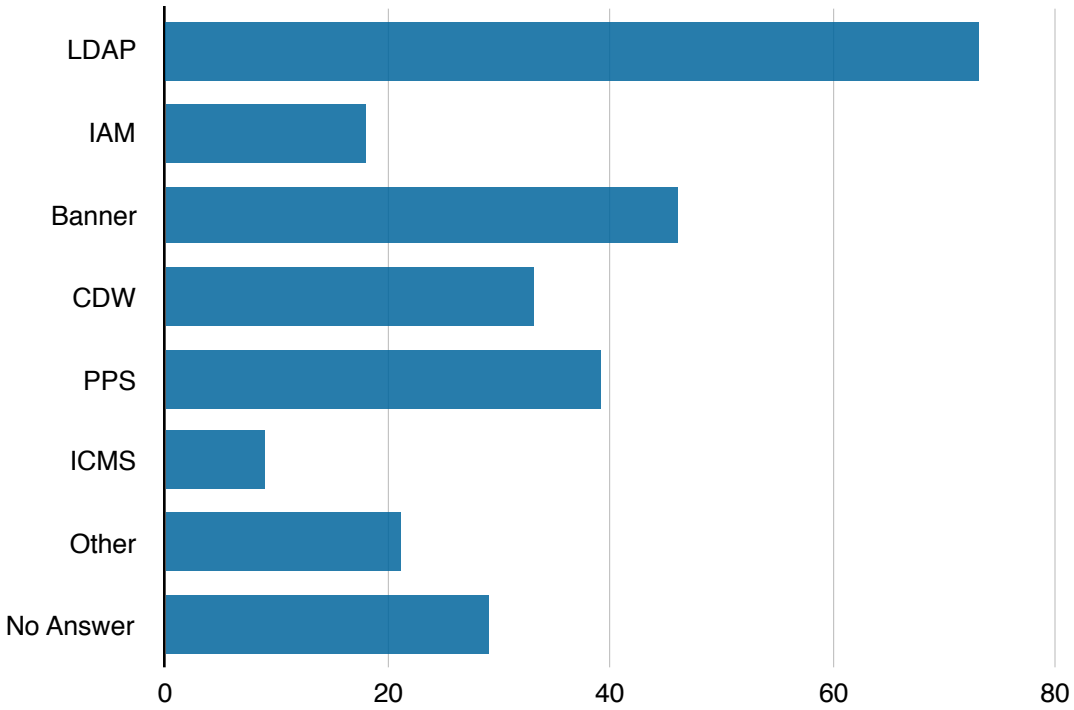
Answer	Response
Dynamic analysis	32
Static analysis	8
CVE reports	1
Other	7
No Answer	85



Community

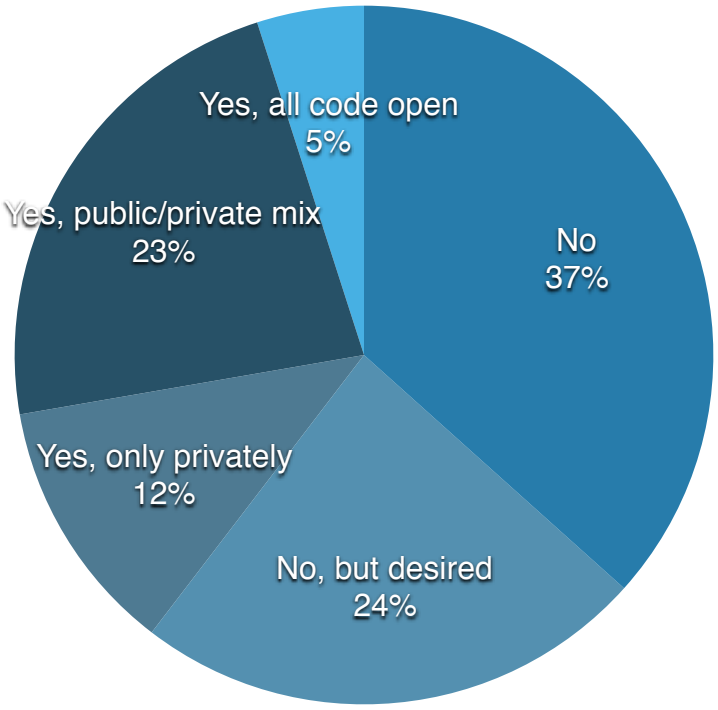
Campus Data Source Usage

Answer	Response
LDAP	73
IAM	18
Banner	46
CDW	33
PPS	39
ICMS	9
Other	21
No Answer	29



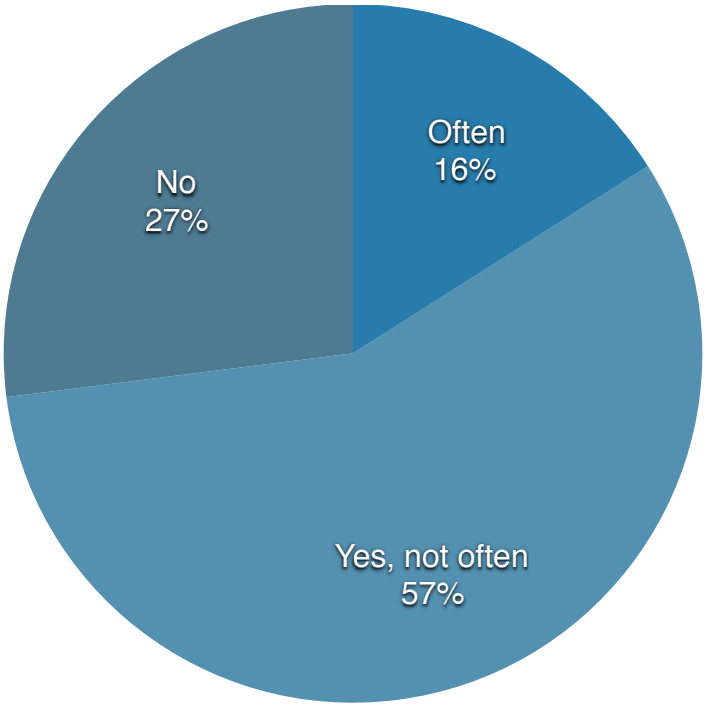
Open Source Publishing

Answer	Response	%
No	44	37%
No, but desired	29	24%
Yes, only privately	14	12%
Yes, public/private mix	27	23%
Yes, all code open	6	5%
Total	120	100%



Existing Collaboration

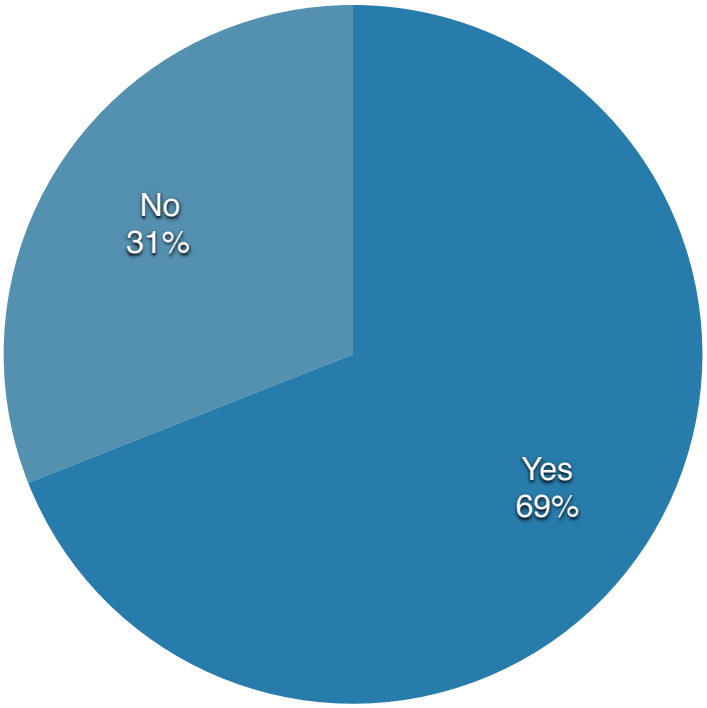
Answer	Response	%
Often	20	16%
Yes, not often	69	57%
No	33	27%
Total	122	100%



Community Engagement

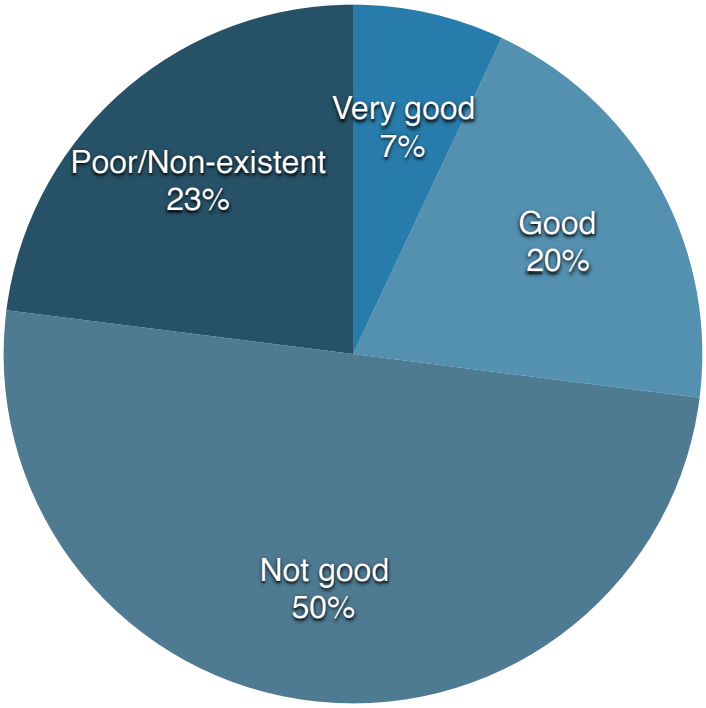
Answer	Response	%
Yes	85	69%
No	38	31%
Total	123	100%

Engages in UCD IT Community (e.g. tsp-share)



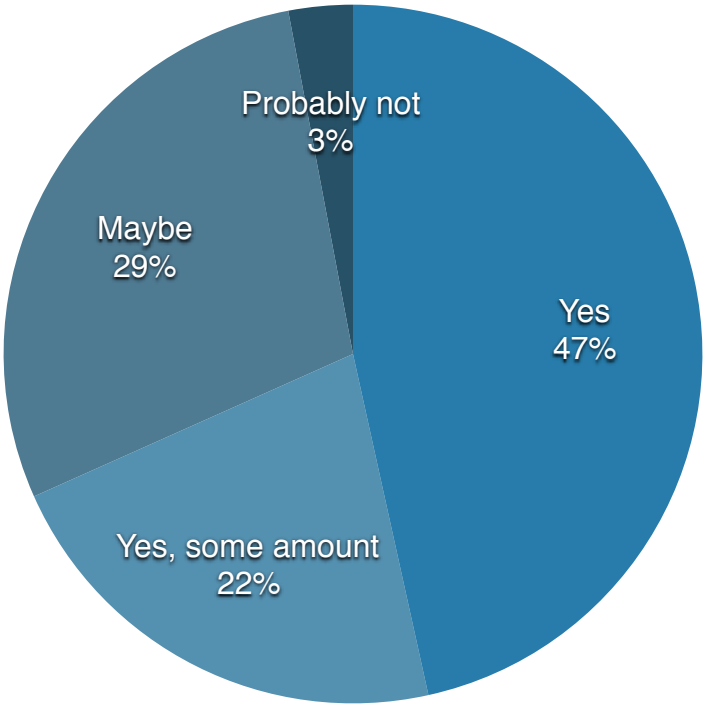
Sense of Connectedness

Answer	Response	%
Very good	9	7%
Good	24	20%
Not good	61	50%
Poor/Non-existent	28	23%
Total	122	100%



Interest in Community Building

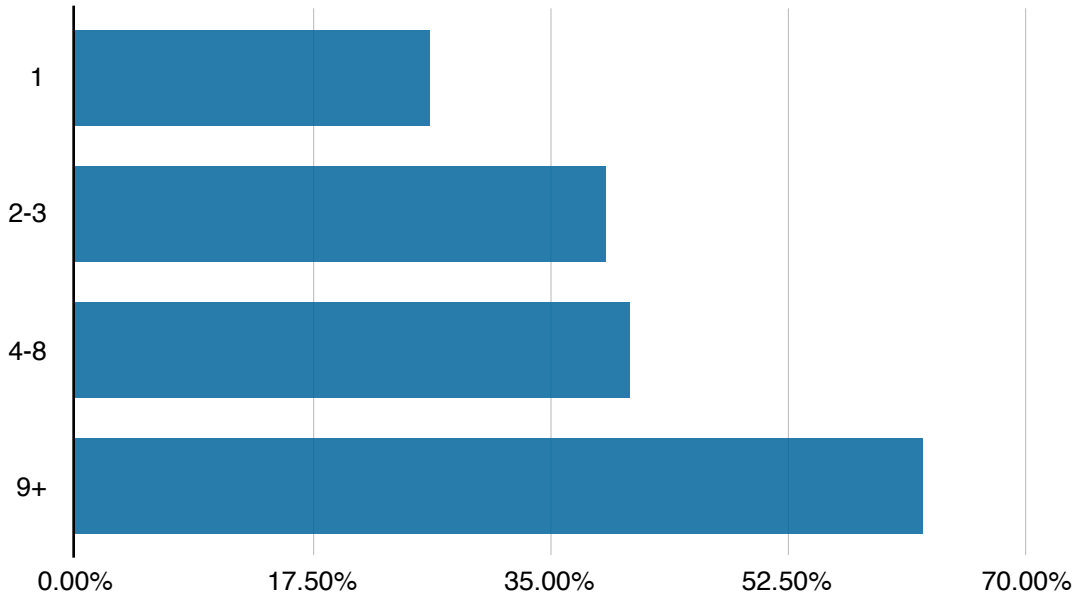
Answer	Response	%
Yes	56	47%
Yes, some amount	26	22%
Maybe	34	29%
Probably not	3	3%
Not at all	0	0%
Total	119	100%



Team Size vs. Software Lifecycle Practices

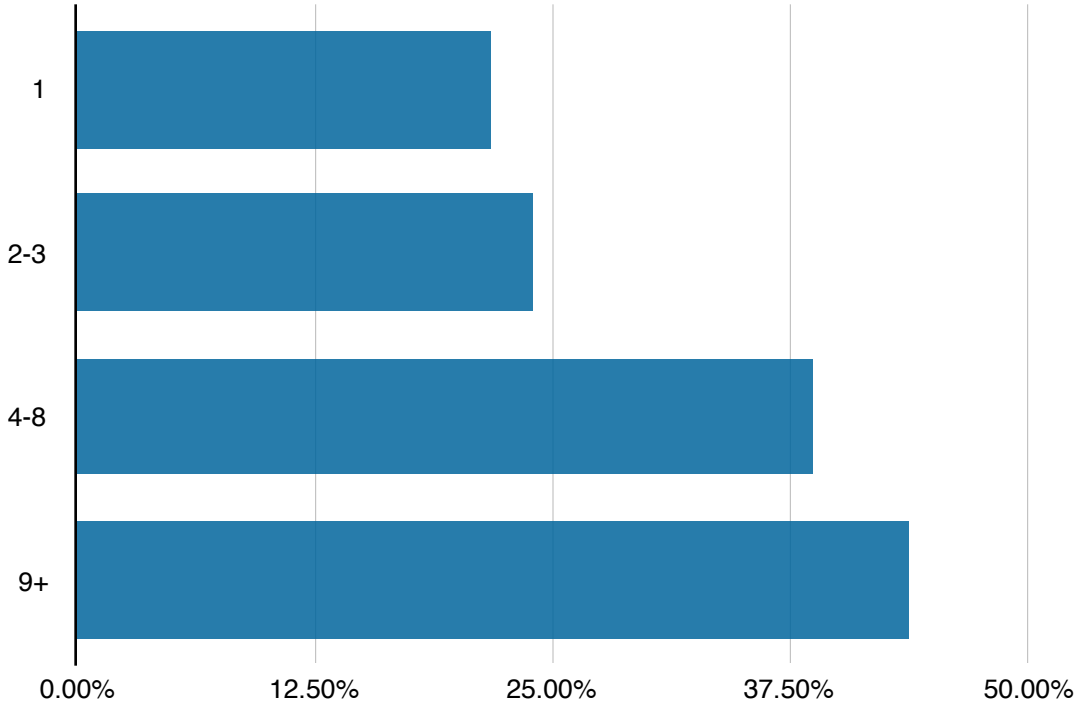
Team Size vs. Automated Testing

	1	2-3	4-8	9+
Yes	6	18	18	10
No	17	28	26	6
% Automated Testing	26.09%	39.13%	40.91%	62.50%



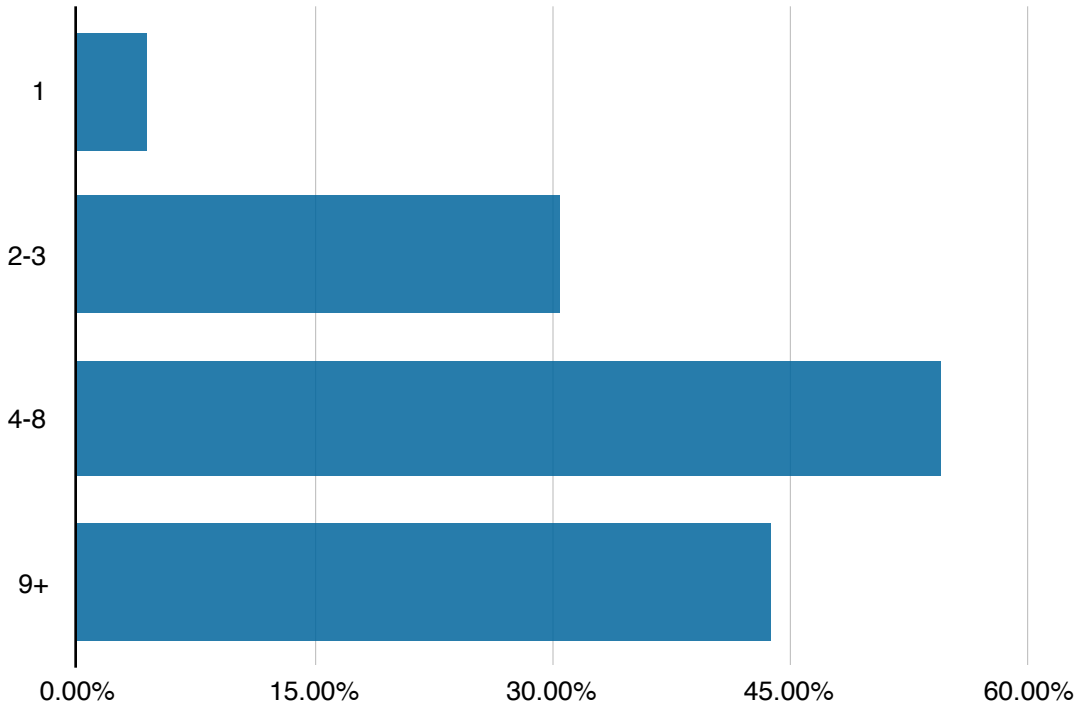
Team Size vs. Security Scanning

	1	2-3	4-8	9+
Yes	5	11	17	7
No	18	35	27	9
% Scanning	21.74%	23.91%	38.64%	43.75%



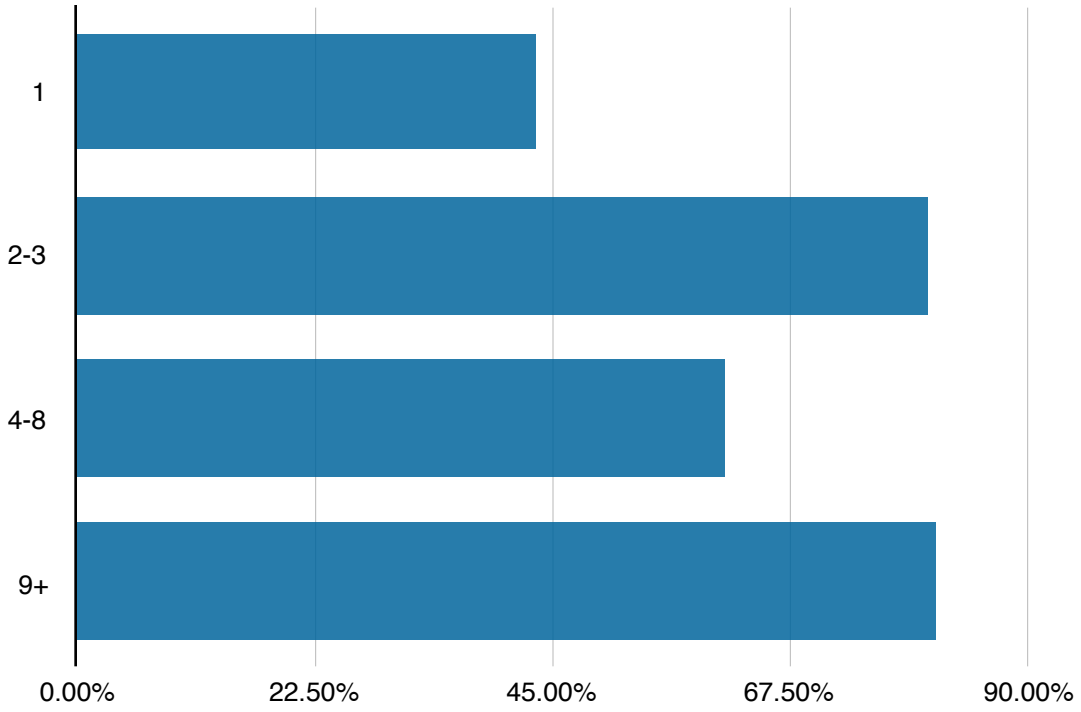
Team Size vs. Cloud Hosting

	1	2-3	4-8	9+
Yes	1	14	24	7
No	22	32	20	9
% Cloud Usage	4.35%	30.43%	54.55%	43.75%



Team Size vs. Monitoring

	1	2-3	4-8	9+
Yes	10	37	27	13
No	13	9	17	3
% Monitoring	43.48%	80.43%	61.36%	81.25%



APPENDIX

Survey Methodology

The survey utilized a set of 40 title codes at UC Davis resulting in 302 individuals. The survey was solicited to these individuals via e-mail and hosted using Qualtrics. It was also solicited via the campus TSP Share e-mail list, a community resource utilized by members of the UCD IT family. The survey received 156 responses over the course of two weeks.

Salary Investment Calculation

The salary investment figure mentioned in the Execute Summary is the result of calculating the average salary based on public salary grade ranges for the 302 targeted individuals mentioned in the Survey Methodology section. The minimum investment figure was $17,633,386 and the maximum was $37,020,462.

CREDIT

Survey Authors

Christopher Thielen (Lead Developer, Division of Social Sciences IT)
Scott Kirkland (Enterprise Application Architect, College of A&ES)

Special Thanks

Jeremy Phillips (Director of IT, Division of Social Sciences IT)
Steve Pigg (Executive Director of IT, College of Engineering)
UCD Application Developer SIG members