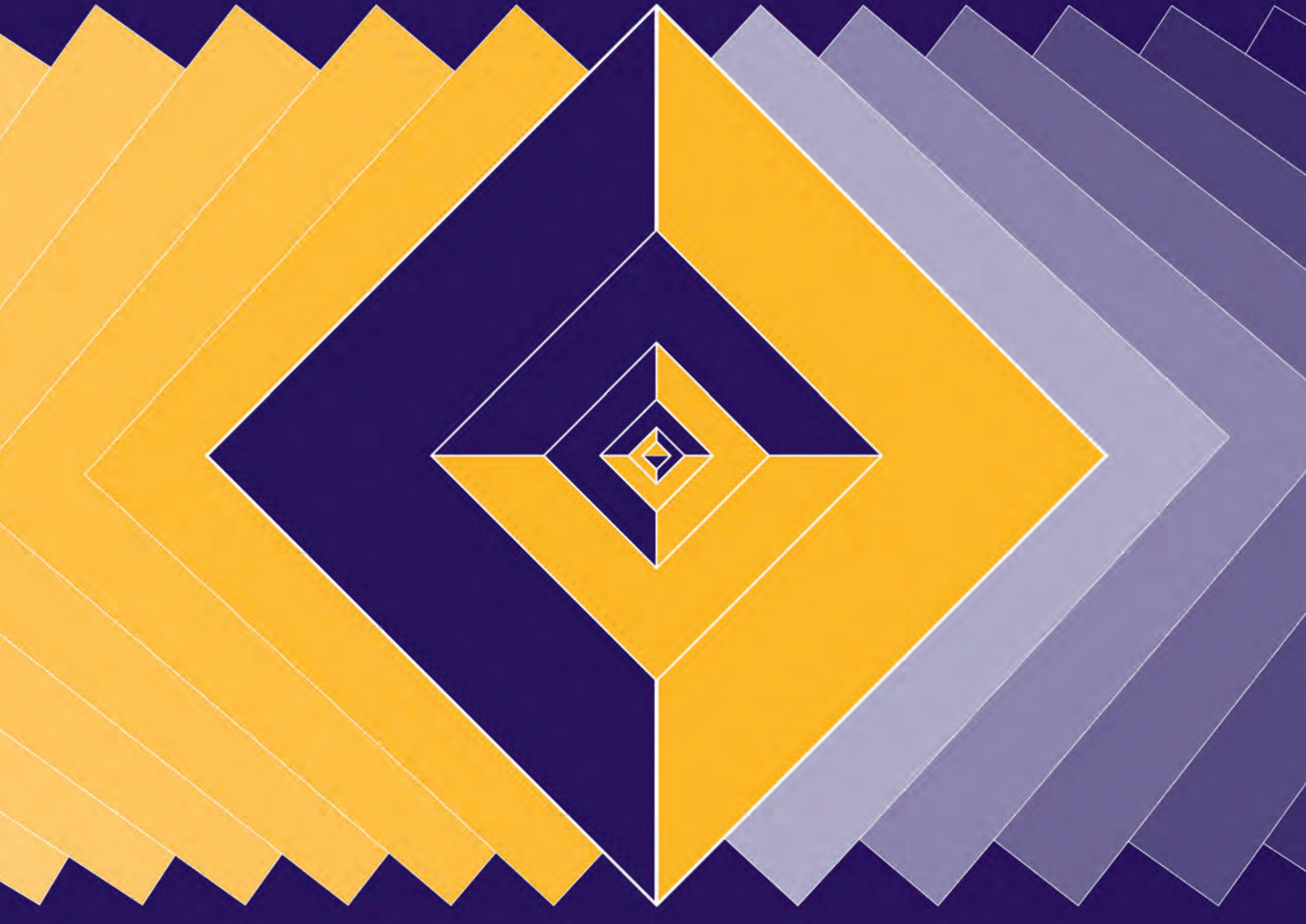
SPEC

Kit 340

Open Source Software
July 2014



Association of Research Libraries

SPEC Kits

Supporting Effective Library Management for Over 40 Years

Committed to assisting research and academic libraries in the continuous improvement of management systems, ARL has worked since 1970 to gather and disseminate the best practices for library needs. As part of its commitment, ARL maintains an active publications program best known for its SPEC Kits. Through the Collaborative Research/Writing Program, librarians work with ARL staff to design SPEC surveys and write publications. Originally established as an information source for ARL member libraries, the SPEC Kit series has grown to serve the needs of the library community worldwide.

What are SPEC Kits?

Published six times per year, SPEC Kits contain the most valuable, up-to-date information on the latest issues of concern to libraries and librarians today. They are the result of a systematic survey of ARL member libraries on a particular topic related to current practice in the field. Each SPEC Kit contains an executive summary of the survey results; survey questions with tallies and selected comments; the best representative documents from survey participants, such as policies, procedures, handbooks, guidelines, Web sites, records, brochures, and statements; and a selected reading list—both print and online sources—containing the most current literature available on the topic for further study.

Subscribe to SPEC Kits

Subscribers tell us that the information contained in SPEC Kits is valuable to a variety of users, both inside and outside the library. SPEC Kit purchasers use the documentation found in SPEC Kits as a point of departure for research and problem solving because they lend immediate authority to proposals and set standards for designing programs or writing procedure statements. SPEC Kits also function as an important reference tool for library administrators, staff, students, and professionals in allied disciplines who may not have access to this kind of information.

SPEC Kits are available in print and online. The executive summary for each kit after December 1993 can be accessed online free of charge. For more information visit: **http://www.arl.org/publications-resources**.

SPEC Kit 340

Open Source Software

July 2014

J. Curtis Thacker and Dr. Charles D. Knutson

Brigham Young University

Mark Dehmlow

University of Notre Dame

Association of Research Libraries®

Series Editor: Lee Anne George

SPEC Kits are published by the

Association of Research Libraries
21 Dupont Circle, NW, Suite 800
Washington, DC 20036-1118
P (202) 296-2296 F (202) 872-0884
http://www.arl.org/publications-resources
pubs@arl.org

ISSN 0160 3582
ISBN 1-59407-921-8 / 978-1-59407-921-4 print
ISBN 1-59407-922-6 / 978-1-59407-922-1 online

∞ The paper used in this publication meets the requirements of ANSI/NISO Z39.48-1992 (R1997) Permanence of Paper for Publications and Documents in Libraries and Archives.

SPEC Kit 340

Open Source Software

July 2014

SURVEY RESULTS

REPRESENTATIVE DOCUMENTS

OSS Contributor Agreements

OSS Licenses

OSS Copyright Notices

OSS Adoption Decision

Job Descriptions of OSS Contributors

Organization Charts

SELECTED RESOURCES

SURVEY RESULTS

EXECUTIVE SUMMARY

Introduction

Open source software (OSS) "licenses must permit non-exclusive commercial exploitation of the licensed work, must make available the work's source code, and must permit the creation of derivative works from the work itself." [St. Laurent, Andrew M. (2008). Understanding Open Source and Free Software Licensing. O'Reilly Media, p 8. ISBN 9780596553951].

The emergence of OSS increases collaboration among research libraries, providing greater control of library tools, as well as improving usability and quality of library resources. This collaborative approach fits neatly with the knowledge and resource sharing ideology of libraries. While OSS is ostensibly "free," adoption of OSS within an organization is not without significant support, integration, and development costs.

The purpose of this survey was to study ARL member libraries' adoption and/or development of OSS for functions such as an integrated library system (ILS), discovery layer, electronic resource management, inter-library loan, digital asset management, institutional repository, course reserve, streaming media, study room scheduler, digital preservation, publishing, floor maps, data warehouse, and other library-related purposes. We wanted to understand organizational factors that affect decisions to adopt OSS, the cost of OSS, and the awareness of OSS systems already in use. With regard to development of OSS, we wanted to understand: 1) research libraries' policies and practices on open sourcing their code; 2) the frequency of research library contributions to open source projects; 3) the reluctance of research libraries to make their code openly available; and 4) the most common benefits and challenges encountered when research libraries open source their code.

This survey was distributed to the 125 ARL member libraries in February 2014. Seventy-seven libraries (62%) responded to the survey by the March 17, 2014 deadline.

Library IT Staff

The 66 responding academic and public libraries reported between two and 50 staff with IT responsibilities as all or part of their duties, with an average of 16 and a median of 14. Three national libraries reported between 130 and 350 IT staff. This bimodal distribution is stark, with the national libraries an order of magnitude larger than their university counterparts. Despite this difference in staff size, we find no statistically significant differences in the relative participation in OSS projects.

Seventy respondents (91%) develop software in-house. Of those, the most common software development practices include using version control (86%) and performing usability tests (86%). The least common practices include the use of independent quality assurance (24%), adherence to a formal, written code reuse policy (10%), and the presence of a committee or working group to encourage code reuse (7%). The most common other software practices mentioned by respondents were agile/scrum development methodologies (5 of 15 respondents) and pair programming (2 respondents). Most respondents reported that their library IT staff are encouraged to experiment with new technologies (75 or 99%), and prototype potential projects (62 or 82%).

As expected, we found a strong positive correlation between staff size and support for software development best practices (particularly creation of software documentation and specifications, creation of user documentation, performing code reviews, using version control, practicing casual code reuse, and standardizing development by utilizing a common framework).

When asked how users give feedback to IT staff, several findings emerged:

- Library employees most commonly give feedback through a helpdesk or bug tracking system (69 respondents, or 91%) and by emailing or calling the system manger/developer directly (67 or 88%).
- Employees of the parent institution give feedback through a form on the library website (54 or 71%), through subject librarians (44 or 59%), by emailing or calling the system manger/developer directly (39 or 51%), and through a helpdesk or bug tracking system (35 or 46%).
- In-library patrons most commonly give feedback through a form on the library website (59 or 78%) and through subject librarians (58 or 76%).
- Remote users most commonly give feedback through a form on the library website (60 or 79%), and through subject librarian (49 or 64%)

In-library users and remote users most commonly use the same feedback methods, suggesting that proximity to the physical library may not significantly impact feedback channels.

In our review of organizations that contribute to open source projects, software development staff ranged from one or two to as many as 14. While organizations that contribute to large scale, formal open source projects were clearly investing heavily in programming staff, it was also clear that a few organizations who didn't have resources for large technology staffs could still contribute to projects with as few as one programmer. The median number of staff reported as working on OSS projects was two, with an average of nearly four.

Organizational structures varied considerably. Within smaller organizations, single programmers are often located in library systems or web units. Within larger organizations, software development staff are often clustered together in application development units located in digital library, digital projects, or library technology branches of the organization.

Library Software

The survey asked respondents to provide information about the type of software used for various library purposes. All 76 respondents use one or more vended products, 72 identified types of open source software used by the library, and 50 identified software that was built in-house. Below are some of the highlights of the range of applications being used.

- Fifty-eight respondents (76%) use a vended, locally hosted integrated library system (ILS). No respondents use an ILS built in house, but five use an open source ILS.
- Forty-five respondents (59%) use a vended, locally hosted interlibrary loan (ILL) system and 29 (38%) license a software as a service (SaaS) ILL system.
- Forty-nine respondents (64%) use a SaaS discovery layer, 17 (22%) use a vended, locally hosted discovery layer, and 10 (13%) use a discovery layer that is built in house. Several respondents indicated that their discovery layer was both a vended, locally hosted system and also built in house, suggesting significant customizations to a vended product.
- Forty-seven respondents (62%) use a locally hosted and supported OSS institutional repository.
- Forty respondents (53%) use a locally hosted and supported OSS digital preservation system.
- Fifty-one institutions (67%) have adopted a system that is open source and supported by a third party.
- The most commonly built in-house systems were floor maps (28 respondents) and digital asset management systems (19 respondents).

- The most frequently adopted OSS systems include institutional repositories (52 respondents), blogging (50 respondents), digital preservation (47 respondents), and publishing (40 respondents).

OSS Adoption

Seventy-four of the 76 responding libraries (97%) report having adopted open source software. Of these, only five (plus one parent institution) have a formal written policy related to adoption of OSS. Twenty-five libraries (34%) have an informal policy, but the other 43 (59%) have no OSS adoption policy. Several respondents reported that policies were currently being created, but could not be shared at the time of their response.

Most respondents indicated their institution had neither a sustainability strategy (50 of of 71 respondents, or 70%) nor an exit strategy (53, or 75%). Reported strategies include minimizing customizations, providing sufficient staffing with needed expertise, and only adopting systems with good documentation and an active community. More than half of the respondents who commented on their exit strategy emphasized the criticality of data migration (8 of 15 relevant comments).

Survey respondents were then asked to identify the system they had most recently adopted and to provide the number of staff and hours required to implement that system. A wide variety of projects were identified, the most common being Drupal, Blacklight, Omeka, and DSpace. Respondents reported from one to eight staff members dedicated to implementation, with a mean and median of three staff. The number of hours required for initial implementation varied dramatically, ranging from 0.75 hours to 9,000 hours with a mean of 573 hours and a median of 160 hours.

Respondents were asked to identify the open source system they most recently adopted that is still in production and to describe the resources needed to support that system. For most respondents, the system referred to in this question was the same system described in the implementation question above. The number of staff required to maintain this system ranges from 0 (for a digital exhibit) to 10 (for a CMS) with a mean of 2.1 and a median of 2. The number of hours required to support the same system ranged from 0 (for the exhibit) to 512 (for a digital repository) per month, with a mean of 68 hours and a median of 20 hours.

Only 16 of 72 respondents (22%) were able to track the costs of either adopting or contributing to an OSS system. Ten respondents who could track the cost of their most recently adopted OSS system reported that expenses ranged from $400 to over $600,000 and, in some cases, represented a multi-year investment. These funds covered a variety of expenses including staff time, hosting, travel, and consulting. The nearly universal source of funding for adopting or contributing to an OSS system was the library's operating budget (69 of 70 respondents, or 99%). A few had additional funding from grants, their university, or a consortium. One ArchivesSpace project received only consortium and grant funding.

The survey asked respondents to describe three benefits and three challenges associated with adopting OSS. The most common benefit is the ability to customize the software (50 responses). Other common themes include low cost or time to implement (27 responses) and the association with an active community (27 responses). The most common challenge was the need for highly skilled staff who could provide support for the OSS system (40 responses). Other commonly cited challenges include poor documentation (19 respondents), a need for additional training or expertise (16 respondents), and substandard development practices (12 respondents).

OSS Contribution

Fifty-six of the responding libraries (78%) have contributed resources to an open source project. The number of projects contributed to by each library ranges from 1 to 20, with an average of 4.6 and a median of 3. Thirty-two libraries report being the primary code contributor for at least one project; a different set of 32 libraries (with significant overlap) identified themselves as the original developer of an open source project.

Commonly reported examples of projects include DSpace (12 respondents), Fedora (11 respondents), Hydra (9 respondents), Kuali (6 respondents), Blacklight (5 respondents), and ArchivesSpace (4 respondents). Below are some of the highlights of library contributions to the projects.

- The most common contributions involved code or developer time (47 respondents), funding (36 respondents), hosting (36 respondents), and testing (8 respondents).
- Across all types of contributions, the most common types of projects included institutional repositories (38 respondents), digital preservation (30 respondents), digital asset management (22 respondents), discovery layer (15 respondents), publishing (13 respondents), authentication/identity management (10 respondents), and electronic resource management (10 respondents).
- Code was most commonly contributed to projects on institutional repositories (32 respondents), digital preservation (22 respondents), digital asset management (20 respondents), and discovery layers (11 respondents).
- Digital preservation and institutional repository projects most often received funding via monetary contributions (19 and 18 respondents, respectively), followed by digital asset management projects (8 respondents).
- Hosting was contributed most often to digital preservation projects (9 respondents), followed by repository and publishing projects (5 respondents each).

When asked about reasons for open sourcing their project, respondents listed the following as being "important" or "very important": a belief that open sourcing would lead to better software (30 respondents), a desire to contribute to an open source community (29 respondents), and shared effort in development and quality assurance of the project (27 respondents).

Sixty respondents (78%) develop plugins, extensions, or customizations for a library-related proprietary or vended system. Of these, 31 (54%) indicated vendors allowed them to distribute the code under an open source license.

As was the case with OSS adoption policies, 44 respondents indicated their library has no policy in place for contribution to open source projects, while 20 respondents have an informal policy. Thirty-four respondents stated that they have no tech transfer policy, while 23 respondents indicated that their parent institution has a formal, written tech transfer policy.

Respondents were asked to describe three benefits and three challenges associated with **contributing** to OSS. The benefit most commonly cited was engagement in the open source community (38 respondents). Other common themes included control of product features and direction (25 respondents), and recognition/reputation (14 respondents). The most common challenge was allocating sufficient staff time to make meaningful contributions (24 respondents). Other commonly cited challenges included writing generalized software for use by a larger community, and securing the financial resources needed to support the open source project and community (7 respondents each).

Since open source project members are rarely collocated, a variety of tools have been employed to help coordinate development efforts. Common tools used include shared version control (37 of 45 respondents, or 82%), an issue tracker (36 or 80%), a mailing list, (32 or 71%), and a wiki (25 or 56%). Forty-one respondents (79%) use a public repository or forge to share their open source code; Github was by far the most common (38 of 41 respondents, or 93%).

The most common licenses used by respondents were GNU Public License v3 (16 respondents), Apache, and Creative Commons (15 responses each).

Respondents were asked to rank a set of success indicators in terms of their importance for the respondent's institution. A significant number (41 or 80%) identified as most important that the functionality better suits their institution's needs.

Respondents were asked if any of their in-house software could have been, but has not yet been, released under an open source license. The 53 respondents (69%) who answered in the affirmative expressed concerns about the staff time commitment required to support the community (41 or 77%), the readiness of code quality for public adoption (39 or 74%), and dependence on other internal systems (30 or 57%).

Conclusion

This survey reveals that nearly all of the responding ARL member libraries are developing custom software and/or adopting one or more open source systems. Contribution to OSS projects is also common, with more than three quarters of respondents actively contributing to OSS projects.

Many respondents expressed a desire on the part of their developers to share with and participate in one or more OSS communities. Larger organizations committed more resources to OSS projects than smaller organizations, but we found no significant correlations suggesting a disproportionate level of commitment to OSS projects as a function of IT staff size. The nearly universal adoption of OSS systems and the high level of contribution to OSS projects may suggest that adoption of and contribution to OSS projects has entered the mainstream for libraries. Simply stated, libraries that develop software also predominantly contribute to OSS projects.

The results of this survey suggest that libraries view organizational behaviors surrounding the adoption of open source software separate from contribution to OSS projects. For example, while respondents view OSS adoption as a means of saving time and resources, contributing to OSS projects is viewed as being advantageous for different reasons, primarily engagement in an OSS community. For developers, the sense of social involvement in a community represented by an OSS project can be a positive source of professional satisfaction, ultimately leading to greater productivity and a return on investment for the library.

Control of software emerged as a theme common to both adopting and contributing to OSS projects. Those adopting OSS systems felt that access to source code gave them greater control, allowing them to change the software as needed, rather than being subject to the whims of a proprietary solution. Those that contributed to OSS projects felt that they gained greater opportunity to influence product direction, especially with respect to product features. In both cases, they perceived a sufficient benefit to their overall productivity to justify the expense of their involvement (as adopters, contributors, or both) in OSS systems.

Page intentional left blank.

SURVEY QUESTIONS AND RESPONSES

The SPEC Survey on Open Source Software was designed by **J. Curtis Thacker**, Discovery Systems Manager at Brigham Young University's Harold B. Lee Library, **Dr. Charles D. Knutson**, Associate Professor of Computer Science at Brigham Young University, and **Mark Dehmlow**, Program Director for Information Technology at the University of Notre Dame's Hesburgh Libraries. These results are based on data submitted by 77 of the 125 ARL member libraries (62%) by the deadline of March 18, 2014. The survey's introductory text and questions are reproduced below, followed by the response data and selected comments from the respondents.

Open source software (OSS) is software that adheres to the following principles: "open source licenses must permit non-exclusive commercial exploitation of the licensed work, must make available the work's source code, and must permit the creation of derivative works from the work itself."[St. Laurent, Andrew M. (2008). ***Understanding Open Source and Free Software Licensing***. O'Reilly Media, p 8. ISBN 9780596553951].

The emergence of OSS has increased collaboration among research libraries, providing greater control of library tools, as well as improving usability and quality of library resources. This collaborative approach fits neatly with the knowledge and resource sharing ideology of libraries. While OSS is ostensibly "free," adoption of OSS within an organization is not without significant support, integration, and development costs.

The purpose of this survey is to study ARL member libraries' adoption and/or development of OSS for functions such as ILS, discovery layer, electronic resource management, inter-library loan, digital asset management, institutional repository, course reserve, streaming media, study room scheduler, digital preservation, publishing, floor maps, data warehouse, or other library-related purposes. We would like to understand organizational factors that affect decisions to adopt OSS, the cost of OSS, and the awareness of OSS systems already in use. With regard to development of OSS, we would like to understand: 1) research libraries' policies and practices on open sourcing their code; 2) the frequency with which research libraries contribute to open source projects; 3) whether research libraries are reluctant to make their code openly available; and 4) the most common benefits and challenges encountered when research libraries open source their code.

IN-HOUSE SOFTWARE DEVELOPMENT

1. How many individuals in your library are responsible for information technology as all or part of their duties? ("Library IT staff" could be a well-defined department or a small part of one person's duties.) N=69

All Respondents

Minimum	Maximum	Mean	Median	Std Dev
2	350	31.78	15.0	72.43

Academic Library Respondents N=65

Minimum	Maximum	Mean	Median	Std Dev
2	50	15.89	14.0	10.62

Number of IT Staff	Responses
2	1
3	1
4	2
5	3
6	3
7	3
8	4
9	5
10	3
11	1
12	3
13	2
14	3
15	3
16	3
17	2
18	4
19	4
20	2
21	1
23	1
25	3
26	1
30	1
36	1

Number of IT Staff	Responses
38	2
40	1
50	2

Nonacademic Library Respondents N=4

Minimum	Maximum	Mean	Median	Std Dev
30	350	190.00	190.00	139.52

Number of IT Staff	Responses
30	1
130	1
250	1
350	1

2. Do library IT staff develop any in-house software? N=77

Yes	70	91%
No	7	9%

If yes, which of the following software development practices do library IT staff employ? Check all that apply. N=70

Usability testing	60	86%
Version control	60	86%
Software documentation and specifications	55	79%
Iterative releases (i.e., small and frequent releases)	53	76%
Reuse of in-house code libraries	52	74%
Reuse of shared framework(s)	51	73%
Casual code reuse between developers	50	71%
User documentation	49	70%
Developer unit testing	44	63%
Accessibility testing	39	56%
Code reviews	38	54%
Coding style guidelines	35	50%
Code commenting guidelines	33	47%
Independent quality assurance	17	24%
Reuse of purchased code libraries	13	19%
A formal written code reuse policy	7	10%
A committee or working group to encourage reuse and oversee shared code	5	7%
Other software development practice(s)	15	21%

Please briefly describe the other software development practice(s) your library IT staff employ. N=15

Acceptance testing, pair programming, community code review, continuous integration, DevOps practices

Agile/Scrum project management practices

Agile development

Agile development methodology with active involvement of customer

Agile Project management

Agile Scrum development methodology. Also note that not all practices checked above are applied universally across all projects.

Continuous integration, bug/enhancement tracking, backlog management

Deployment strategies, such as Capistrano

Experimental software as part of research projects

Functional testing. Virtualized development environments and code driven environment configuration. Design patterns. Agile approach, trying to implement a 2–3 week cycle for milestones. Frequent standups, not daily but certainly when issues arise. Iterative development with incremental feedback.

Informal usability test

Modify open source code for library use.

Pair programming

Pair programming, interaction design (personas, user stories, prototyping), TDD

Security checks, penetration testing

3. Which of the following activities are library IT staff encouraged to participate in? Check all that apply. N=76

Experimenting with new technologies	75	99%
Prototyping for potential projects	62	82%
Rewriting existing systems to make them easier to support	57	75%
Collaborating on projects that are not part of their specific responsibility	56	74%
Other related activity	10	13%

Please briefly describe the other related activity. N=10

Collaborating with developers outside the Libraries, participating in open-source developer communities, attending developer users' groups meetups.

Configuring, customizing, and extending existing systems.

DevOps work to support operations staff.

Existing systems are rewritten only when there is a need.

Inter-campus work, marketing department and ITS

Other responsibilities as assigned/needed.

Professional conferences

Streamline services, decommission paid services, security review.

Training on related emerging software technologies and platforms.

We work to keep applications supportable in the library by choosing technologies and languages that can be supported by more than one person in IT, and through cross training on those technologies.

4. How do users of library systems give feedback to your library IT staff? Check all that apply. N=76

Feedback Method	Library employees	Institution employees	In-library patrons	Remote users	N
Through a helpdesk or bug tracking system	69	35	25	31	71
Emailing or calling the system manager/developer directly	67	39	16	23	68
Through a web form built into the library website	48	54	59	60	65
Through subject librarians	33	44	58	49	65
There is no established method	—	—	1	—	1
Other method	5	3	6	6	8
Number of Responses	76	69	75	71	76

If you selected "Other method" above, please specify the user group and briefly describe that method. N=12

"Contact us" link and Chat

Emails or chat notes or phone messages forwarded by other library employees.

In person

In person discussions [with library employees]

Our public feedback takes place through email to support web sites, or notes in suggestion boxes. Our system user feedback takes place through the Help Desk.

Service teams for our major brands who help assess requests for features, problems, projects, etc.

Through library public service staff (not all of them necessarily subject librarians).

User research, informal conversations with members of various groups

We have a User Experience department that employs several methods for gathering feedback of existing services, as well as feedback and input on services as they are being implemented.

We have an extensive release testing process that involves faculty and staff throughout the libraries.

We no longer have a web form for tech support; it was replaced with a web helpdesk ticketing system. The IT ticketing system has many different categories of help, and it is used by a variety of campus departments. Help requests are triaged to the appropriate campus department based on need.

We occasionally hold focus group sessions with student users (generally undergraduates). These are sometimes very informal introductions to prototypes on which we gather first-reaction comments to inform further development, at other times, these are more structured formal feedback opportunities.

SYSTEMS BUILT IN-HOUSE THAT AREN'T OPEN SOURCED

5. Has your library built in-house any library-specific systems that could be, but have not been, released as open source? N=77

Yes	53	69%
No	24	31%

If yes, what are the primary reasons for not releasing it as open source? Check all that apply. N=53

Concerns about staff time commitment required to support the community	41	77%
Concerns that the code quality is not ready for public adoption	39	74%
Dependence on other internal systems	30	57%
It didn't occur to us	7	13%
Seeking to license or sell the system	2	4%
A competitive desire to have the best system	1	2%
Other reason(s)	12	23%

Please briefly describe the other reason(s) for not open sourcing the system. N=12

Highly customized to address local requirements.

Lack of clarity about campus policies for licensing and intellectual property ownership.

Legal considerations.

Narrow niche applications where a community is unlikely to develop.

Not approved for release.

Not documented for external audiences.

Often these systems reflect local practices. We've not viewed them as useful beyond our local environment.

Planning to release a service as open source, working on appropriate licensing language at this time.

Security

Security concerns related to embedded information.

Technology Commercialization Office needs to review any software developed at the university.

Time needed for review of and compliance with licenses of third-party components.

CUSTOMIZING PROPRIETARY SYSTEMS

6. Does your library develop plugins, extensions, or customizations for any proprietary or vended systems? N=77

Yes	60	78%
No	17	22%

If yes, do those vendors allow the code you developed to be openly distributed with OSS licensing? N=57

Yes	31	54%
No	26	46%

Comments N=16

Customizations are specific to our institution's unique requirements and would not be generally useful to others. Some customizations would not be supported by organization for security and support reasons.

Ex Libris allows/encourages development and customization of their systems, but sharing is limited to other Ex Libris user institutions via CodeShare on the password-protected Ex Libris EL Commons web site.

In some cases, we are not sure, because we have not specifically asked the vendor. In the case of our ILS vendor, their willingness to have our code openly distributed depends upon how much proprietary information about the system would be divulged by the new software, i.e., the nature of the software and how it interacts with the proprietary system.

Most do allow for this. Or, they at least have an established community of their customers where code can be shared. We attempt to write code that is mostly generalizable to any like system, in order to allow ourselves the flexibility to changes systems later on with fewer dependencies on custom development.

Not all our vendors allow this. Some applications would reveal proprietary information about the data model used in vendor product.

Not sure if it's allowed (haven't asked).

Some allow this, some do not.

Some vendors allow it, others do not. Ability to redistribute is not a major factor in determining whether we develop plugins, extensions, or customizations.

Some vendors do, some vendors don't.

The library has developed plugins for use with its proprietary ILS software (Voyager) and has shared the plugins with other libraries. They are considered a federal employee product, therefore public domain.

The library IT staff has plans to develop plugins, extensions, or customization for the ILS. The ILS vendor does allow APIs to be openly distributed.

Unsure [whether vendor allows this].

We do provide the extensions without a license but we include a disclaimer.

We have a couple of vendors that have taken contributions from our teams but that code is not openly distributed with OSS licensing.

We primarily build them for us and share them if we can. Some vendors allow for semi-open sharing.

With the signing of appropriate releases and/or agreements.

LIBRARY SOFTWARE

7. Please identify the type of software used by your library for each of the following purposes. Check all that apply. N=76

Purpose	OSS (locally hosted, locally supported)	OSS (locally hosted, supported by a third party)	OSS (hosted and supported by a third party)	Vended product (locally hosted)	Vended product (hosted by the vendor or SaaS)	Built in-house	N/A	N
Inter-library loan	2	—	1	45	29	4	3	76
Institutional repository	47	1	6	5	12	14	7	76
Digital preservation	40	10	7	11	3	15	19	76
ILS	3	1	2	58	17	—	1	75
Discovery layer	16	2	3	17	49	10	2	75
Course reserve	2	—	2	43	16	12	7	75
Electronic resource management	8	—	1	18	38	13	3	74
Streaming media	16	1	—	33	18	5	12	74
Blogging	38	2	13	11	8	1	9	74
Authentication/identity management	25	7	8	33	8	11	7	74
Digital asset management	33	3	2	20	11	19	9	73
Study room scheduler	17	—	1	13	20	13	14	73
Publishing	36	3	4	4	10	5	19	73
Link resolver	5	1	4	22	43	7	3	73
Floor maps	8	—	—	8	5	28	28	71
Web analytics	15	2	7	10	47	4	—	71
Data warehouse	11	1	2	7	4	10	43	69
ELMS	4	1	2	11	6	2	45	68
Data analysis	6	1	1	17	11	8	36	68
Visualization	10	—	—	15	8	3	40	67
Other purpose	13	1	—	4	3	9	10	31
Number of Responses	70	22	40	76	73	50	67	76

If you indicated above that the library is using any software for an "Other purpose," please briefly describe that purpose. N=25

Archival description software (ICA-AtoM for archival finding aids)

Archival Management for managing archival data.

Citation Fox and IL Fox

Content management system

Course reserve is Blackboard, hosted by university IT, not the library.

Database software (MySQL), Web Server (Apache), Exhibits (Omeka), Timeline & Map web support (Neatline)

Electronic Finding Aids: currently use Archon, will move to ArchivesSpace in the future.

Enterprise service bus and rapid application development environment afforded by Kuali Rice.

FYI, we are considering vended product/hosted by vendor to include the university's central IT unit (Office of the Chief Information Officer) and central academic computing unit (Office of Distance Education and E-Learning).

Here are some top software products the Libraries have developed to fulfill our needs: research consultation services, equipment management, trouble ticket, feedback, hours, event administration, news/alerts, reference transactions, spam blocking, reminders. Also, we have a vendor product for single-sign on for our ILS. Lastly, there are, additionally, more campus central IT run services that the Libraries use.

Just wanted to note an additional dimension to consider. We make use both of very library-specific software primarily managed by the Libraries but are also heavy users of software provided by our university's central IT dept. In some cases the relationship is somewhere in between a locally hosted and vendor hosted situation.

Many of the choices above do not allow for accurate categorization of our environment.

Monitoring, performance analysis, metrics, digital signage

Note: Dataverse (Data Warehouse) and geospatial software (Data Analysis) on shared consortial system: From Scholars Portal, of the Ontario Council of University Libraries.

Offsite storage inventory, RFID, self-checkout.

Omeka for online exhibits

Other purpose is Digital Collections application and ContentDM for metadata management.

Persistent identifier software

Research guides/FAQs, digital exhibits, EAD repository, staff directory, Database A-Z

Resource annotation and analysis tool (RUanalytic). Metadata and resource handling application (OpenWMS) and ETD submission system (RUetd)

Scientific data analysis, text mining

Social media archiving, and social media display/sharing

Subject-specific databases/portals, electronic access

We also have several productivity tools that are small productivity applications, such as tools for replacement materials workflows, another for reformatting. Our subject pages are driven by the MyLibrary toolkit. We use Library a la Carte for subject guides.

We use OSS and in-house software for many other needs: lots of back end server stuff like sharing data between systems, and frontend custom displays for various resources.

8. **Please indicate how important each of the following software selection criteria is to your library. Please make one selection per row. N=76**

Criteria	1 Not Important	2	3	4	5 Very Important	N
Functionality that best meets our needs	—	—	1	14	61	76
Staff time to support	—	2	13	35	26	76
Control and customizability	—	1	13	36	26	76
Monetary cost for support and maintenance	—	—	14	40	22	76
Staff time to implement	—	3	21	31	21	76
Monetary cost for implementation and licensing	—	2	14	31	27	74
Other criteria	1	—	2	6	12	21
Number of Responses	1	6	42	65	70	76

If you indicated above that the library is using any "Other criteria" to select library software, please briefly describe the criteria. N=17

Academically developed and controlled to reduce risk. We do buy vendor solutions but with intention and critical analysis due to the amount of data we have and priority to preserve and make that information available.

ADA compliant, standards based, interoperable with other systems, meets security standards

Adoption of the software in the wider (library) community. Whether or not the software is actively being maintained.

Compatibility with existing systems

Compliance with industry standards for system interoperation

Integration with complex information environment; ability to extend software beyond library to provide services to other departments and institutions; opportunities afforded for professional development in open- and community-sourced software.

Integration with existing systems

Integration with other library systems. Community of software users and evidence of development.

Interoperability with existing systems. Community around an OSS project.

Interoperability with other systems; sustainability

Is it open source?

It is important for any systems to meet accessibility standards.

Safety and security of the software (impact on IT security at the library)

Software quality and reliability

Use of open data standards

Vendor responsiveness for vended products or a robust user community or user groups for OSS.

We try to insure that all components of our cyberinfrastucture, whether developed in house or not, work well together to fit within the RUcore architectural framework. All tools and services can then be managed together and receive upgrades/enhancements on the same schedule. Our commercial ILS, Sirsi/Dynix does not support this and one IMPORTANT reason we are moving to Kuali OLE is the ability to integrate all our cyberinfrastructure into a coherent platform where the focus can be an integrated approach to user needs.

Please select the correct statement about the use of OSS at your library. N=76

Our library is using open source software	74	97%
Our library is NOT using any open source software	2	3%

If your library is using open source software, you will continue to questions about OSS Policies.

If your library is NOT using any open source software, you will skip to the screen Library Doesn't Use OSS.

OSS POLICIES

9. Please indicate the kinds of policies your institution has related to OSS. Check all that apply. N=73

OSS Policy Content	Formal, written library policy	Formal, written parent institution policy	Informal library policy	Informal parent institution policy	No policy	N
Adoption of OSS developed elsewhere	5	1	25	7	43	73
Development of OSS in-house	3	4	20	10	44	73
Contributing resources to OSS projects	4	5	20	6	44	73
Technology transfer	2	23	4	8	34	69
Number of responses	7	24	32	16	59	73

Comments N=8

I am not aware of any official or documented policy regarding OSS at the institution at this time.

The library has policies and procedures for making library-produced open source code available outside the library. The policies are currently under editorial revision and are expected to be released later in 2014.

Not aware of university policy though it may exist.

Our library informally supports and greatly encourages IT staff to use and contribute to OSS projects.

We are just beginning to develop policies in this area.

We have no formal policies with regards to OSS. We are pragmatic in our approach to open source software, and compare with vended solutions based on criteria noted earlier in this survey.

We know from experience there is a process, but could not locate the policies.

Whether a commercial vendor or OSS product best meets a given need is determined on a case-by-base basis.

10. Does your institution have either a sustainability or exit strategy related to OSS projects? N=71

Strategy	Yes	No
Sustainability strategy	21	50
Exit strategy	18	53

If there is either a sustainability or an exit strategy, and a document that describes the strategy, please include the document in the Call for Documents at the end of the survey.

If there is a strategy, but no document, please briefly describe the strategy below.

Sustainability Strategy N=15

Informal. Must be sustainable. Implementing department is accountable.

Minimize customization.

Platform review on a regular basis (~five year cycle).

Provide staff support for ongoing development of our open source content management system (Drupal) and ongoing support and development of our institutional repository (if we stay with an open source product after our pilot project).

Staff to support; minimum customization; data management a requirement.

Stated in strategic plan and through staffing, but no formal document.

Supported as a strategic application, that is, assigned as primary responsibility for a group or person in IT.

The closest is the Hydra partner agreement https://wiki.duraspace.org/display/hydra/Hydra+Community+Framework

The Kuali OLE project, not yet in production, is developing a sustainability plan to grow and sustain the software for at least a decade. This includes ongoing support, in cash and in-kind, from partners, attracting new partners, and partnering with commercial affiliates for software support, training, implementation, and development contributions.

The way in which we contribute and leverage OSS assures that the university has access to all OSS and can continue to maintain, develop, or discard that technology according to our needs and priorities. We are involved in the strategic steering, operational and development of the majority of OSS that we use.

We adopt only OSS projects that have a healthy, active community for collaboration/support. We also choose projects with methods for contributing code back, and with good documentation so in-house work can begin quickly.

We avoid making extreme customizations that are super specific or require extensive changes to the base code, hence sustaining our OSS from one version to another is relatively flexible.

We plan out sustainability in the same manner as other software implementations and development activities.

We will adopt an enterprise OSS system or component only if it is developed within the narrow range of technologies—languages and deployment platforms—in which we have expertise and experience, and only if the system or component is supported by an established, stable community. We follow best practices, particularly around testing and engineering for stability and scalability, in order to minimize support and maintenance costs. We move support out of the development group and into a support group (with partial success).

When adopting OSS or engaging in development of OSS, we look for and/or try to establish a broadly-based community of support in order to mitigate risks of being too dependent on one institution's/individual's resource commitment.

Exit Strategy N=15

Data migration is mandatory.

Exit strategy only concerning ability to export all data and relationships from software.

For the eXtensible Catalog (XC), our exit strategy (which we are now implementing) involves moving all infrastructure support for the software to a library consortium (CARLI) that has been a major partner in developing the system. Our strategy also has included a detailed communication plan for notifying all stakeholders. We have not deployed XC locally. For IR+, we are now discussing possible options for future actions that may include a formal exit strategy.

Informal. Must have a reasonable exit strategy. Implementing department is accountable.

Native export tools/XML, etc. unique to each application

No formal exit strategy. We do choose software with open data standards so that our information can be exported on a whim and used in different software.

Not only with OSS, but with all software systems, we develop such that dependencies are not vendor or product specific, but could allow for replacement of a part of our infrastructure with a like service without having to redesign the whole.

Our data adheres to open standard policies, so if we ever need to migrate out or exit out of the OSS, our data would be compatible with any other system.

The plan will include an exit strategy to allow either end-of-life of the software, or mechanism for turning over software to other interested parties.

To ensure that our data are portable, we require that an open source software be capable of exporting our data in a standard data exchange format.

Use of a software system whether OSS or vended requires data export capability.

We always look at an exit strategy when making a decision about a particular technology solution, regardless of whether it is open source or not.

We keep data and presentation layers separate, so that migration out is easier. We choose OSS with data storage techniques that allow for complete export of all relevant data in a format for easy migration.

We may resort to a hosted/vended product for our institutional repository if we're not satisfied with the results of our pilot project using an open source software repository product.

We regularly evaluate our needs against the technologies we are using and are aware of alternatives. Because we are involved in the strategy and development of most of the OSS, we are also aware of the threats for the OSS that we use. Use of OSS affords us greater time to plan migration or alternative strategies. We have experience and expertise with vended solutions that offered minimum time and therefore forced quick migration and alternative solutions that in some cases have proven to not meet our needs.

REASONS FOR ADOPTING OSS

11. Please identify the open source software that your library has adopted. N=66

Apache, Eventum, Movable Type

Archivists' Toolkit

AutoDewey: software was created at Northwestern University Libraries, adapted at LC.

AWStats, DSpace, Islandora, Fedora Commons, ICA-AtoM, Archivematica, Drupal, Apache Solr, Apache Lucene, Apache, Squid, KeePass, Nagios, PuTTY, MongoDB

Blacklight content management system, Google Map viewer API, California Digital Library Micro Services, Archivists' Toolkit, ArchivesSpace, DSpace, LibStats, Drupal, Omeka, Linux, Apache, LOCKSS

Blacklight, Fedora Commons, DSpace, Handles, WordPress

Blacklight, Fedora Commons

Blacklight, Hydra, Solr, Fedora Commons, DSpace, Opencast Matterhorn, Avalon Media System, Variations Digital Music Library. Many utilities/tools such as ffmpeg, JHOVE, etc.

Digital Library Extension Service (DLXS), Fedora Commons, Omeka, Guide on the Side, Apache, Tomcat, Wikimedia, Linux

Drupal

Drupal, PHP, phpScheduleIt, Blacklight

Drupal, CORAL, Guide on the Side, ArchivesSpace

DSpace (2 responses)

DSpace, Open Journal System (OJS)

DSpace, Drupal

DSpace and several others

DSpace, Fedora Commons, Hippo CMS, Drupal, Open Journal System (OJS)

DSpace, Fedora Commons, Hydra, Apache, MySQL, Solr, Linux, Open Journal System (OJS), Python, R, Ruby, Archivists' Toolkit, ArchivesSpace, WordPress, Drupal, Tomcat

DSpace, Islandora, Fedora Commons, Drupal, Tesseract, ICA-AtoM, Open Journal System (OJS), Open Book System, Manitobia, LOCKSS, PostgreSQL, MySQL, Apache suite of applications, Python, Redmine (Ruby), Git

DSpace, Omeka, MDID

DSpace, Omlaut, WordPress

DSpace, Open Journal System (OJS), and VuFind

DSpace, Open Journal System (OJS), Archivematica, ICA-AtoM, LOCKSS, WordPress, MediaWiki

DSpace, Open Journals System (OJS), eXtensible Text Framework (XTF), Omeka, WordPress, Drupal

DSpace, Fedora Commons, Archivematica, ResourceSpace; Public Knowledge Project (PKP) including Open Monograph Press (OMP), Open Journal System (OJS), Open Conference System (OCS), General Transit Feed Specifications (GTFS), RefStat, Suma, Xibo, Mondo Grinder, phpScheduleIt, software for hours and locations

DSpace. File Analyzer, Archivists' Toolkit, LOCKSS

Fedora Commons

Fedora Commons, Hydra, CORAL, Apache, Puppet

Fedora Commons, Blacklight, Hydra, Solr, Avalon, WordPress, ArchivesSpace (soon), Piwik, MySQL, Apache, Neatline, and many other components for transforming or disseminating information.

Fedora Commons, DSpace, Open Journal System (OJS), Open Conference System (OCS)

Fedora Commons, DSpace, Umlaut, Shibboleth, Xerxes, Blacklight, Vireo, Hydra, Solr. As well we have adopted several OSS, such as Tomcat and Apache, that do not seem to be the focal point of this survey.

Apache, Drupal, Webinator, Fedora Commons, WordPress, Omeka, BuddyPress, Avalon Media System, eXtensible Text Framework (XTF), Bugzilla, Handles, PostgreSQL, PHP, Perl, Linux

Hydra, Blacklight, Solr, Drupal

Hydra, DSpace, Drupal, WordPress, LC Newspaper Viewer, Archivist ToolKit, VireoCat, various open source utilities

Hydra, Fedora Commons, Solr, Blacklight, phpScheduleIt, Open Harvester, WordPress, others

Hydra, Omeka, Drupal, Shibboleth

Islandora

Koha, Fedora Commons, Xerxes, Library a la Carte, WordPress, MyLibrary, eReserves, Blacklight, VuFind, Hydra, CORAL

Linux, django, Python, Solr, Lucene, Nginx, PostgreSQL, various support libraries and toolkits

LOCKSS, Public Knowledge Project (PKP), Omeka, Plone

Lots: Drupal, EZProxy when it was OSS, our web stack, our Moodle LMS, our IR, others.

Open Journal System (OJS) and Omeka; CORAL ERMS

Open Journal System (OJS), DSpace, Omeka

Open Journal System (OJS), Open Monograph Press (OMP), Drupal, WordPress, Dokuwiki, MediaWiki, Islandora, Fedora Commons, Spiceworks, Piwik, Omeka, Archivists Toolkit

Omeka, Avalon, WordPress, Silverstripe, DSpace, Open Journal System (OJS), Open Conference System (OCS)

Open Journal System (OJS)

Open Journal System (OJS), eXtensible Text Framework (XTF), AWStats, Daily Stats, WordPress, Webilizer, GoogleAnalytics, MySQL, PHP

Open Journal System (OJS)

phpScheduleIt, Omeka, WordPress, Archon, ArchivesSpace, Blackligh, SubjectsPlus, Variations, Avalon, Fixity, Assana, MarcEdit, DMPTool, Lucene/Solr, EzProxy, E-Prints

PHP, Blacklight, MongoDB, PostgreSQL, MySQL, Northwestern U Book Viewer, Solr, Lucene, GSearch, Djatoka, Fedora Commons. SciDB, Openstack, django, Openshift, Drupal, CentOS, Cassandra, sqe, Ruby, Python (and libraries), Perl and libraries, many Apache tools, GNU tools, Nagios Open Monitoring Distribution (OMD), Spacewalk, OCS Inventory

PHP, MySQL, Linux, Apache, Drupal

Public Knowledge Project (PKP), Research Project Calculator (Assignment Calculator), ArchivesSpace, Apache, Linux, MySQL, PostgreSQL, Hydra, Blacklight, Fedora Commons, Solr, PersistantURLs (PURLZ), Omeka, Open Journal System (OJS)

Streetprint, DSpace, OS Ticket, Dokuwiki, Guide on the Side

SuraSpace products, SugarCRM, ArchivesSpace

The main library-specific OSS we use: VuFind (and Solr), DSpace, LOCKSS. We make heavy use of other general open source software including Ubuntu, Apache, Tomcat, WordPress, etc.

This list could go on for pages: Apache, Fedora Commons, DSpace, Islandora, WordPress, Drupal, MySQL, Linux, Docker, Redmine, OpenLDAP, VuFind, Arduino IDE, Open Journal System (OJS), Raspbian, OpenOffice, GIMP, etc. We have both servers and desktops running various Linux flavours; nearly every piece of software on them is by nature OSS.

Too many to mention. But here are some: Ubuntu, Apache, PostgreSQL, Python, django, Perl, PHP, Java (openjdk), Solr, jQuery, D3, postfix, Nagios, phpScheduleIt, DSpace, Drupal, MySQL, osTickets.

UCLA MWF, DSpace, MySQL, Apache, PHP, SAMBA, Open SSL, Open SSH, Linux (CentOS and Ubuntu), Sendmail, Solr,, Nutch, Tomcat, WINE, VirtualBox, KeePass, PuTTY, Pidgin, Stat Transfer, WinSCP, 7zip, Firefox, Thunderbird, SPSS, Audacity, MarcEdit, FreeMind, Gimp

Umlaut, Blacklight, Xerxes, Fedora Commons, Solr, DSpace, Drupal, WordPress, Rails, Jenkins, djatoka, OpenLayers, Git, Linux, PHP, Java, Apache, Tomcat, GNU Compiler Collection (GCC)

VuFind

VuFind to develop our discovery layer, Shibboleth for identity management (this is the standard at our parent institution and it has been integrated with library systems).

VuFind, Drupal, CORAL, ARC, Omeka, Solr

VuFind, DSpace, Open Journal System (OJS), Papyrus, Islandora

Webcalendar, Hydra

WordPress, XTF, Omeka, Nagios, PKP OAI Harvester

12. Please indicate how important each of the following reasons for adopting OSS over a competing vended product is to your library. Please make one selection per row. N=72

Reasons	1 Not Important	2	3	4	5 Very Important	N
The functionality of the open source system best meets our needs	—	1	3	14	54	72
Greater control and customizability	1	—	5	26	40	72
Lower monetary cost for implementation and licensing	2	6	25	18	21	72
Lower monetary cost for support and maintenance	2	8	23	25	14	72
Library or institutional policies encourage the use of OSS	27	15	18	11	—	71
Desire to contribute to the library OSS community	6	15	22	18	9	70
Less staff time to implement	2	18	32	10	7	69
Less staff time to support	4	11	31	17	4	67
Other reason(s)	3	—	4	—	3	10
Number of Responses	31	37	65	61	67	72

If you indicated above that the library has other reason(s) for adopting OSS over a competing vended product, please briefly describe the reason(s). N=7

3 Important

Limited availability of software

Ongoing economic sustainability is critical for determination to adopt OSS or a vended product. All public facing web applications must be made accessible for disabled users, so control of this is vital for our institution.

OSS implementations relate to gaps in the vended market.

Staff familiarity with OSS systems.

5 Very Important

Better integration with RUcore cyberinfrastructure.

Freedom to study, copy, modify, and redistribute. Availability of potential staff candidates familiar with free software options. Trust in the respective developer communities.

Resourcing: Leveraging pooled resources within community, which decreases cost for cross training and ensures forward movement and support during staff shortages. Training & retention: staff have a ready network of peers and training opportunities which greatly supports skill building, impact of work, visibility of their work and professional networking.

Additional Comments N=4

For above statements, don’t necessarily agree, e.g., “less staff time to implement”—generally takes more time to implement an OSS—so not important is what was selected.

Security, analytics, integration with older systems

We disagree with the statements above that OSS takes less time to implement and less staff time to support, and so were unsure how to respond to them. Saying that they are "not important" to us would be misleading, so we left them blank.

We like our OSS to have a robust developer community.

13. Please identify your most recently adopted OSS system **that has been deployed**, and indicate how many staff and how many hours of staff time were required to complete the initial production deployment. An estimate of the number of hours is acceptable. N=63

OSS System	Staff	Staff hours	Comments
Archivematica			
ArchivesSpace			
ArchivesSpace	2	160	
Archivists' Toolkit	1	100	Customization was contracted out.
Blacklight	3	1500	
Blacklight	4	100	The work was done in two 2-week sprints of about 25 hrs/wk. Part of the experience was getting used to Blacklight as a development environment, in addition to developing the intended discovery piece.
Blacklight	8	9,000 (very rough estimate)	Work on this project spanned many groups and involved work across several units of our organization. This estimate is likely to be fairly inaccurate.
Blacklight			We cannot share cost related information at this time.
Blacklight, Fedora Commons, djatoka, Lucene, Book Viewer	2	Approximately 2,000 hours	OSS allowed team to select best components for specific parts of project to meet project goals of this major development effort. OSS allowed us to greatly customize presentation and functionality. Functional changes are more easily achieved with OSS than a vended product, but of course requires in-house development staff.
CORAL (e-resource management)	1	30	Does not include hours spent with data management from Technical Services; just the time the developer spent.
DAMS – Islandora, Fedora Commons	1	630	
Dokuwiki	1	8	
Drupal	2	500	Change platform for library website.
Drupal	3		
Drupal	3	1000	
Drupal	3	at least 240 hours	Three staff members were involved in the implementation of Drupal, but only a portion of their time for a period of about three months.
Drupal	5	3500	Library website development and deployment.
Drupal	3		Number of hours was not tracked.
DSpace	2	40	

OSS System	Staff	Staff hours	Comments
DSpace	4	200	
DSpace	5	1000	Hours calculated on 4 hours of work per week spread across 5 staff for one year. This relates to a grant project has been going on for several years. 1000 hours is probably a conservative estimate. We have not been formally tracking personnel time for OSS projects.
DSpace	4	200	Mostly one IT staff implementing configurations and changes and two librarian/admin staff making design decisions and testing. Sysadmin time during startup.
Fedora Commons	3	80	
Fedora Commons	4	unknown	
File Analyzer	1	5	
Guide on the Side	3	500	This is a piece of software that we actually developed, so the number of staff hours is very high due to the development time.
Guide on the Side	3	2	Staff included 1 technical resource and 2 librarians.
Hippo CMS	5	2500	Very rough estimate; also includes building the html/CSS for new website from scratch.
ICA-AtoM	3	700	
Islandora	2	many	We can't calculate staff hours with any accuracy, as we haven't been systematically keeping track.
Islandora	2	16	We are counting server build only. Software install was completed by support vendor. We are not counting system evaluation prior to purchase of vendor support or customizations/configuration/initial material ingest.
Islandora	3		Difficult to estimate; deployment bleeds into other issues, such as metadata import, etc.
Islanora	4	160	We have four full time staff developing on the Islandora stack. This includes efforts for Drupal, Solr, and Fedora Commons, which comprise Islandora.
Koha	7	130	For Jerusalem site
LC Newspaper Viewer	4	100	
Linux/Apache/django stack for library widget	2	0.75	
Movable Type			Project occurred 8 years ago; estimate of staff time unknown.
obento (our in-house developed bento search)	4	500 (approx.)	
Open Journal System (OJS)	3	100	
Omeka	1.5	40	

OSS System	Staff	Staff hours	Comments
Omeka	2	60	Developer created an accessible fork of Omeka, called Omeka_a11y, for use in our library, then removed institution-specific changes and released the fork on GitHub.
Omeka	3	20	
Omeka	5	450	
Omeka		301	One digital exhibit
Open Journal System (OJS)	2	50	
Open Journal System (OJS)	2	400	
Papyrus	2	210	
phpScheduleIT	4	400	
ResourceSpace	1	8	
Room Booking	2	60	
RUanalytic	3	400	
Shibboleth	N/A	N/A	The development was driven by the university's Middleware Group, so it is difficult to estimate library time on the project.
Social Feed Manager	2	40	
UCLA Mobile Web Framework	1	40	Software started at UCLA to create a framework to have web sites work well on a mobile device without having to create apps for devices.
Vireo	2	200	Times are grossly estimated for the last question.
Vireo	2	120	For ETD management
VIVO	4	100	Deployment was spread over several months.
VIVO	6	250	
VuFind	2	500	
WebCalendar	1		
WordPress	1		
WordPress	2	25–35	We were already using WordPress on a limited scale for blogs and some web pages, but recently fully adopted WordPress for our library web site. Hours are based only on the time to setup and configure a new web server environment and WordPress instance for the intended use. Time spent creating and adding content was in addition and significantly greater.
Xerxes	2	2 * 280 hours	

Additional Comment

We do not have a metric for this at this time because it is not useful to capture unless we are comparing two similar scoped systems (OSS vs. Vendor). Much also depends on the type of application and needs it presents: rebrand requirements, training requirements, configuration and sometimes development to utilize.

14. Please identify your most recently adopted OSS system **that is still in production**, and indicate how many staff and how many staff hours per month are required to maintain the system. An estimate of the number of hours is acceptable. N=56

OSS System	Staff	Staff hours per month	Comments
ArchivesSpace	5	15	We are still in the process of migrating from Archon to ArchivesSpace.
Archivists' Toolkit	1	100	
Blacklight	3	200	
Blacklight	4	300	The system, though deployed, is still under active development. We cannot separate development from support.
Blacklight			We cannot share cost related information at this time.
CORAL	1	2	
DAMS – Islandora, Fedora Commons	2	280	The number of staff hours includes more than maintenance because the system is continually being developed for use beyond the library, to the entire enterprise. The two staff are working full time on the system, migrating digital assets from other legacy and proprietary systems into the DAMS, implementing authentication, user-centered interface and navigation, writing bulk ingesters, creating testing scripts, distributed solutions, data preservation processes, etc.
Droid	2	200	
Drupal	1	20	
Drupal	2	30–40	
Drupal	2	75	Two staff members are involved with maintaining Drupal, but not full time. It adds up to about .5 FTE.
Drupal	5	125	Library web site
Drupal	5	100	
Drupal	3		Hours unknown
DSpace	1	2	
DSpace	1	5	
DSpace	2	32	We are not currently tracking maintenance time for OSS systems.
DSpace	2	120	
DSpace	2	10	One Sysadmin handling patches/updates/security and one Developer handling feature requests and fixes.
eReserves	2	250	This is a locally developed system that we don't open source currently.
Fedora Commons	3	80	
Fedora Commons	4	512	
File Analyzer	1	20	
Guide on the Side	1	<10	

OSS System	Staff	Staff hours per month	Comments
Hippo CMS	10	40	Includes maintenance and occasional upgrades; does not include development of new website features.
Hydra	1	60	By "in production," in this question, it appears to us you actually mean still in development prior to deployment or in the earliest stages of deployment.
Hydra	3	100	
ICA-AtoM	2	20	
Islandora	1	70	
Islandora	2		We can't calculate staff hours with any accuracy as we haven't been systematically keeping track.
Nagios	0.25	1	For this OSS component, there only requires minimal effort to maintain, just the application of system patches.
obento (our in-house developed bento search)	2	20	
Open Journal System (OJS)	1	10	
Open Journal System (OJS)	3	75	24 instances; customer support and updates to software
Omeka	—	—	One digital exhibit
Omeka	1	10	The active installation requires minimal work. We are in the midst of a version update, to replace the current production installation—that is a larger time commitment, but I view it as a "project" not "support".
Omeka	1	2	Most effort spent sporadically when software needs to be upgraded.
Omeka	1.5	2	Very difficult to give staff hours per month; depends very much on the release cycle for product and status of projects being implemented.
Omeka	3	10	
Open Journal System (OJS)	1	8	Hours/Staff do not include continued development time.
Open Journal System (OJS)	2	44	
Open Journal System (OJS)	2	50	
RUanalytic	2	40	We are currently enhancing it via an NSF grant so spending more time on it than normal, particularly in response to feedback from grant P.I.
Shibboleth	N/A	N/A	This is incremental process, since we are supporting the university's single sign-on initiative. Library use of Shibboleth is being gradually phased in, with the goal of Shibboleth becoming the standard.
Social Feed Manager	1	2	
Solr/Nutch	3	20	Apache-based product to create a search index for our public web site.

OSS System	Staff	Staff hours per month	Comments
Spiceworks	2	4	For this question, we are assuming that "in production" means systems that we are actually depending upon, as opposed to systems that we have installed but not started to actively use ("deployed"), as in the previous question.
Umlaut	2	2 * 21 hours	
Vireo	2	< 10	
Vireo2.2	4	10	
VIVO	1	10	
VIVO	3	180	
WordPress, Confluence, JIRA, Jenkins	1 to 2	20	
WordPress	1	25	
WordPress	1		We have one full-time webmaster who spends the majority of his time doing custom design, maintenance, etc. on our WordPress site, as well as many other library staff who spend smaller percentages of their time creating content (blog posts, web pages, etc.)
WordPress	2	15–20	This is time spent maintaining the web server and WordPress environments and does not include time spent maintaining web site content.

Additional Comment

We do not have figures for separating software only maintenance and support and again is not useful unless comparing to something similar that offers the same functions. Much of the software we develop does not have vendor alternatives and our requirements go beyond just what the software delivers.

COST OF ADOPTING OSS

15. Were you able to track the costs of the most recently adopted and deployed OSS system? N=71

Yes	10	14%
No	61	86%

If yes, please indicate the costs of adopting that OSS system, and briefly describe what expenses were covered (e.g., staff time, equipment, training, travel, etc.) N=10

Cost	Expenses Covered
$400	Server hosting agreement for VM with university central IT department; cost here doesn't include staff time.

Cost	Expenses Covered
$646,119.07 over 4 years (yearly average cost $161,529.76)	Staff (IT, Archival, Tech Services), 3rd party developers, Amazon cloud hosting & storage
$3,800	3800
Approximately $8,000	Staff time
$50,000	Consulting, hosting, staff time, training, travel
$17,000	Vendor installation and support, virtual server, travel. Other costs not tracked so not included.
$40,000	Staff development time - NSF grant budget
$45,500	Staff time
We cannot share cost related information at this time.	We cannot share cost related information at this time.
Approximately $200,000	Staff time, equipment

What was the source of the funds for adopting this OSS system? Check all that apply. N=70

Library's operating budget	69	99%
Grant(s)	6	9%
Parent institution	4	6%
Consortial budget(s)	4	6%
Gift(s)	1	1%
Other funding source(s)	3	4%

Please specify the other funding source(s). N=3

2014 expenses will be reduced by the Amazon cloud hosting, storage, and back-up costs ($130,034.16) because the university's central IST department will provide these services locally.

We are able to track project costs but our practice is not to track time spent to implement.

We have library staff working on this project, but we have not tracked their hours, since it is part of their day-to-day duties.

BENEFITS AND CHALLENGES OF ADOPTING OSS

16. Please briefly describe up to three benefits your library enjoys as a result of adopting OSS systems. N=65

A cost effective means to deploy business critical software and services. Ability to customize for internal uses. Ability to serve users of the digital library with software standards and standard interfaces.

A single system hosts many formats; still images, books, newspapers, audio, video and manages all associated files, derivatives, preservation data. The core system was further developed to meet specific local functional requirements of users without waiting for vendor releases. The system is scalable to millions of objects and can provide a single enterprise solution for the whole university.

Ability to contribute bug fixes and enhancements desired at our institution. Lower initial cost outlay. Control over support and maintenance costs.

Ability to customize/extend the software to meet local needs. Easier to evaluate/test/prototype different options. Staff experience gained from working with the source code.

Ability to have applications that better meet the library's needs. Accessibility and usability are usually better for library patrons. Inline with library values to support open access.

Ability to have solutions more customized to our and our users' needs. Ability to provide innovative services beyond the reach of commercial products. Reduced dependency on vendor changes in products and priorities.

Ability to modify or change software based on specific needs. Community-based support and knowledge availability. Reduced/eliminated licensing costs.

Ability to rapidly respond to local needs/issues. Ability to configure/customize service to local needs. Local knowledge of interoperability issues with other systems in use by institution.

Because we have a local software development shop, we can adjust OSS systems to meet our requirements, and have succeeded in deploying systems that we believe are superior to commercial systems. The quality of OSS systems is often very high. OSS systems can evolve rapidly in response to new ideas and trends.

Better engagement with the communities doing the work. Ability to contribute to the improvement of systems used by libraries and archives. Better able to recruit and maintain developers from a wider circle of practitioners.

Built for a specific need. Cost of licensing.

Can customize to fit our requirements. Broader base of software support.

Community of Support. Better understanding of the technology. Good exit strategy.

Configurable. Broad user base. Ease of use.

Control and customizability. Speed to adopt. Ability to participate in community and shape direction.

Control of functionality. Participation in community over roadmap. Flexibility of customization.

Control over customization and software direction. Less effort to support. Functionality meets our needs.

Control over discovery system. Ability to expand scope of discovery system. Unlinking back end from discovery.

Control over system features and design. Reduced time to fix issues or troubleshoot.

Creation of highly collaborative environments. Increased knowledge/skills. Having a foundation on which modifications can be made to address local needs.

Customization. Connection to current systems. Ownership of data.

Customization. Community participation.

Developing and adopting OSS affords us flexible, sustainable solutions that meet complex problems facing libraries, archives, and museums. Reduces risk by affording control over the solutions that meet our needs and control over when and how to use them. Staff are working on solutions that have impact beyond our institution, have a professional network, higher visibility of the work they do while the library can save in training, resourcing, and stop gap measures during staff shortages.

Flexibility. Reduced cost and purchasing wait time. Community support.

Flexibility. Low risk in the case of project failure, due to nature of projects chosen. Customizability.

Flexibility in responding to changing needs. Opportunities to look for added value enhancements to services. Engagement with a wider community of library developers.

Flexibility to customize. Licenses are cost effective. Software easy to require.

Freedom to use, study, copy, modify, and redistribute solutions that work for us. Rapid access to really good ideas by people who don't work here with us. Implied membership in development communities.

Functionality that meets our needs. Ability to integrate software into our infrastructure, and with other library and university systems. Professional development opportunities from participation in the community.

Functionality that was not present in affordable commercial software. Ability to customize to meet our needs. Ability to integrate with local software.

Greater control of implementation timeframes. Lower up front costs. More flexibility with regard to customization.

Greater flexibility. No similar vended tools. Ability to develop new tools as needed from the OSS system.

Having access to a wide network of support for a system. Participating in a large community of developers with library-centric OSS expertise. Having more control over features and interfaces.

Improved quality. Customizability. Cross application integration.

Integration with other library systems. Opportunity to test software with little investment; low-cost testing/adoption.

Involvement at the national/international level. Can move to another product with no contractual lock-in. Opportunity to improve the product.

It gives us greater control over the implementation. There can be greater interoperability with OSS systems. The cost is internal; it generally includes staff time and training.

Less staff time to modify and support OSS systems when compared to creating homegrown products. We have better control over OSS software and our data than we do with vended products. OSS communities tend to have vibrant and engaged members, which can be a good support resource.

Leverage adoption community support. Attract applied research funding for OSS projects. Align with institute mission to share knowledge.

Lower acquisition cost. Complete control over user experience and user privacy. Flexibility.

Lower cost. Customizability. More control.

Lower licensing and maintenance cost. Fast deployment. Functionality sharing.

Many choices available. Allows for quick prototyping. Ability to modify to environment.

More options to choose from than just those provided by commercial vendors. Can frequently implement without need of identifying and budgeting funds to purchase product. Can implement more quickly because there is no need to go through a complicated and time-consuming licensing process.

No purchase cost. Community support. Flexibility to modify.

No purchase price. More control.

Obtaining functionality that best meets our needs. Control and customizability. Community participation.

Opportunity to contribute code that meets not only our specialized needs, but those of other institutions. Opportunity for developer to join a community of developers (professional development). Reflects our commitment to the values/mission of the university and library profession.

Opportunity to influence future directions. Opportunity to increase staff expertise through reviewing and extending OSS code. Opportunity to leverage work at other institutions and contribute back to product.

Out of the box, relatively quick to install. Robust development community. Customizable face.

Prototyping; ability to try before you buy the "free puppy." Ability to customize to meet our needs. No licensing fees.

Provide additional services to user community. Less expensive. Greater ability to customize.

Quality of software. Ability to customize. Lower cost.

Rapid prototyping/updating. Community support. Reduced cost.

Save on licensing costs. Ability to customize, integrate with other library system. Research and publishing opportunities.

Shared expertise with other libraries. Customizability. Extensibility.

Software that is developed to meet the needs of the community rather than being profit motivated. Software that can be customized. Strong support community.

Speed of adoption. Services provided that would not otherwise be available. Good community support.

Staff development: increasing skill and knowledge. Flexibility in terms of being able to change without penalty. Rapid deployment: always faster to use OSS than a vendor solution for most anything.

Sustainability and influence in directing future development. More easily able to integrate other library platforms. Financial.

The ability to customize the product. The ability to influence the direction of development.

The ability to respond quickly and effectively to the needs of our user community. The ability to troubleshoot our systems because of the deep understanding we have of the software. OSS developer communities are more responsive than most vendors' support systems (at least in our experiences).

Tools and services that are designed and customized to real faculty and student workflow needs. Tools and services that integrate into a coherent and cohesive cyberinfrastructure. Reusable code that can enable building other things.

Using WordPress instead of our parent institution's commercial content management system allows us to develop a web site that is more attractive, more customizable, and meets our needs.

We have the ability to do deep customization without waiting for a vendor. We keep fixed costs down by avoiding proprietary licensing and support fees. We help improve the library OSS ecosystem by sharing our code and reusing other code.

17. Please briefly describe up to three challenges your library encountered as a result of adopting an OSS system and the strategies employed to overcome these challenges. N=64

Adapting the service for multiple users has been a challenge; we've addressed it by assessing user needs and conducting training. Systems security is a concern. We've addressed it through the use of penetration testing.

Adopting open source software isn't free. There are support costs. We schedule regular maintenance of our software. Some vendors have more resources and can be quicker to market to meet a need or respond to changing environment. To deal with this, we always keep our options open to swapping pieces between OSS and vended solutions.

Although we try to minimize support costs through good engineering, we nevertheless have to support the applications. We move most application support to a support group after deployment, but some support issues require developer attention, taking time away from development efforts on other projects. The time to deployment can be long depending on the level of development or customization we undertake.

Bad software. Bad documentation. Too much staff time needed to get application running.

Bugs

Change in mindset on part of technical staff to contribute to open source communities.

Changing code: careful tracking of changes. Pressure to always provide latest version: lots of testing.

Compatibility. Waiting for developers to make/implement fixes. Staff support.

Complex environment: use virtualized environment. Poor documentation: staff enhance documentation through various means. Rapid change: each successive version of a software is not necessarily implemented; assessed to determine the added value.

Configuration and customization may take time and may not be possible to customize to satisfaction. Idiosyncratic code which will need to be documented and systemized. Attitude that open source may mean an inferior product.

Continued maintenance. Documentation.

Coordinating activities across developers not in the same location. Managing expectations for features and delivery dates. Finding qualified developers and keeping them in the library.

Creation of new tools needs deeper understanding of the OSS system.

Customizability and time to maintain customizations. Resource time to support users in using as the software is somewhat unintuitive.

Deciding whether to develop custom extensions or install existing. Resolved through cost benefit analysis.

Difficulty in getting timely accurate support. Requires developing in-house deep understanding to support. Finding clearly written documentation. Building a documentation system to accompany OSS systems necessary. Understanding limitations in the feature set of an application. Building prototypes and involving stakeholders in pre-production testing.

Difficulty with interoperability. More staff overhead for maintenance and support. Unclear migration path.

Documentation. Adoption.

Documentation: develop local documentation; contribute testing, bug reports, and documentation to project. Incomplete functionality: develop alternative workflows, contribute enhancements. Poorly developed or managed code contribution process: minimize customization of software.

Ensuring enough cross training, especially to ensure continuity in case of staff loss. Handling non-core customizations in upgrades of core. Occasional gaps in documentation of OSS systems.

Finding and selecting products with the appropriate functionality. Discovery committees are usually tasked with the assessment and evaluation process. Conveying support knowledge from an experienced staff member to an inexperienced staff member. In-house modifications to the OSS software can make this more challenging. The strategy for overcoming this challenge is to make extensive comments within the changed coding.

Gap in web design skills. Had to use existing resources. Difficult to organize functional teams to create requirements or user-stories. Developers filled gaps. Lack of a mature service model to offer support.

Having the skill sets to support the product over the long term. Having a voice in governance within the open source community. Software bugs with little or no support to fix issues. To overcome, we try to purchase vendor/3rd party support.

Highly skilled in-house staff required in lieu of vendor support. Deep customizations can create a local fork that is hard to upgrade for a new upstream release. The power to customize is addicting. Sometimes it's better to adjust the local workflow to fit a 90% good enough tool than to spend time building that last 10%.

Immature technology: chose only established and mainstream product. Lack of support: chose only product with available paid support. Lack of control on product and feature direction.

Increased deployment time for unfamiliar products: admins must spend more time learning software upfront. Users expect sys admins to be source of expertise for deployed products: have to educate users about becoming self-servant with available documentation and knowledge bases. Alignment of local project timelines with those of OSS products.

Initial hardware needs: repurposed hardware from other project. Reliance on locally developed expertise: limit the amount of customization.

Institutional IT department has had difficulty supporting large data, bandwidth, and open source philosophy in general. Core system needed considerable development beyond basic functions. Version updates not always scheduled or based on an upgrade path. Poor implementation and documentation.

It still creates IT debt that we need to manage. The communities are not big enough to always add value. We have a greater need for technical documentation when we release an OSS software.

Keeping up with software updates. Training overhead for new staff.

Lack of documentation: communication on listservs and forums.

Lack of documentation and support can slow adoption. Sustainability problems can lead to abandoned projects. Skepticism on part of non-technical stakeholders.

Lack of necessary elements: have developed our own or contributed to community work to do same. Lack of documentation.

Lack of staffing: we haven't really resolved this. Lack of training in specific areas: fortunately our location between two large metropolitan areas has made this fairly easy to obtain. Lack of policies and procedures for OSS: we have established a work team and are starting to address this.

Learning curve. Staff time. Server capacity.

Learning curve: overcome by online training resources. Recovering from patches to customized software: overcome by before/after detailed checklists. Training and maintenance: overcome by building in new routine tasks for maintenance and cutting back on other services.

Maintain thorough documentation of local implementation & customization decisions. Failsafe upgrades: need to make sure locally developed plugins, etc. don't crash with each new upgrade. Maintain sandbox environment to thoroughly test upgrades before pushing to production. Version control of development vs. production servers.

Managing all the associated software components of a software package. Getting the organization to make the appropriate level of investments. Free software does not mean no cost. Have to monitor security patches more closely.

Metrics that can be used to compare against commercial software since much of what we develop and use is done by OSS communities. We are not merely shopping, adopting, and tailoring; we are building it together and have no access to all the information needed for valid metrics. Strategy: gather information on cost for solutions that only serve a portion of needs and be able to articulate that against ballpark expense of equivalent OSS. Getting software developers from commercial sector to understand that the return on investment for day-to-day work is not exact. When you preserve cultural heritage or the scholarly record, the impact on research or learning is very difficult to measure; there is no clear profit margin in terms of money. Strategy: make applicants aware of the mission and strategy of the organization, be transparent about the institution and how the organization fits within the institution and the larger educational community. Managing expectations: since we have OSS, people believe they can have everything but we aim to standardize practices within our national and international communities so we have to manage expectations on how much customization and one-off design is sustainable and practical. Strategy: engage early, often, and be transparent into why and how work is being accomplished.

More complexity in implementation, configuration. Accommodating local customizations at time of software upgrade.

More up-front development work: it's all our responsibility. "Forking" code: ending up with code that is removed from the open source core.

Need to grow staff expertise: grew it.

New development method (agile) employed. Managing scope. Prioritizing desired enhancements.

Newer versions no longer supporting important features: overcome by changing to a different system. Minimal to no support: overcome by increasing our knowledge and expertise, or securing third-party support where available. Lack of availability of formal training in system use: overcome by taking a deep breath and figuring it out as we go.

Open source is not free. Infrastructure costs and developer salary/benefits add up over time. Keeping up with upgrades. Future of the product is not entirely up to us and may go in an undesired direction.

Personnel to sustain systems: proposal to administration to re-hire. Priority conflicts with multiple systems: working with leadership to implement portfolio management. No clarity on system expectations and service design when OSS solutions are requested from the IT department: working with leadership to implement project management.

Poor documentation for the software: our Systems Department was helpful getting the server ready, then we depended on an active and enthusiastic user group. Minimal tech support: we depended on fellow-users because help from the software was limited.

Problems must be resolved by staff: network with community of users. Documentation lacking: network with community of users; acquire reviews of OSS. Maintenance and upgrades: don't be the first.

Software ceasing to be developed by the community. Software being developed for technology stacks that we don't run. Inconsistent documentation.

Some software can have a steep learning curve.

Staff and consultant time spent on debugging and customization. Cost of implementation and support not much less than commercial product. Product looks behind-the-times.

Staff cost. Long-term stability and robustness of software. Open source licenses can be variable.

Staff time. Lack of support. Lack of clear documentation.

Support for changes, bug fixes is dependent upon user community. Future development can be taken in a different direction than desired, or stopped completely. Learning curve in the organization for production implementation & support after development. Not all open source software is documented well.

The main supporting group provides poor support or abandons the software. Dependence on technologies that are not well known within the library. Ability to both customize the system and track future releases.

Time to deploy. Compatibility among modules. Lack of documentation.

Total cost of ownership can be higher. Replacement of knowledge when staff involved in OSS project leaves. More difficult to justify investment in OSS over vended solution in face of budget cuts/constraints.

Transition plans for stranded (abandoned) OOS systems. In-house resources to support and extend OSS system hard to cultivate. Upgrade cycles are resource-intensive.

Trial and error approach is sometime necessary: need to have a tolerance for failure. Lack of community support at times. Development takes time.

Understanding features and capabilities of OSS now and in the future so we do requirements analysis and trial implementation. OSS can't be included as part of a formal RFP process: no strategy to overcome. Understanding the total cost of ownership for OSS: no strategy to overcome.

Unplanned costs associated with maintaining and customizing the code.

Variable level of support from the community, especially with older versions. Strategy: upgrade often! Sometimes missing one or two key features that are beyond the library's ability to develop in-house. Strategy: contract out to third parties. Greater staff time required to support. Strategy: ensure staff know the system thoroughly.

We locally customized one system and are a bit stuck with our fork now, but it's a tradeoff we manage just fine. Very good modern software tools often don't fit our legacy data; e.g., django requires utf8 db connections but Voyager requires us7ascii.

WordPress is not supported by our parent institution (university), so if we lost our in-library webmaster we would have no support.

LIBRARY CONTRIBUTIONS TO OSS PROJECTS

18. Has your library contributed to any library-related OSS projects (either your own or another organization's project) in any way (e.g., code or developer time, money, hosting)? N=72

Yes	56	78%
No	16	22%

If you answered Yes, you will continue to additional questions about your library's contributions to OSS projects.

If you answered No, you will skip to the section Additional Comments.

19. Please identify the open source software your library has contributed to. N=50

ArchivesSpace, Hydra

Avalon, Variations (testing partner)

Blacklight Reserves Direct OLE

Blacklight, Solr, Hydra, Vireo, Umlaut

Code for custom functions of our ILS

Developing a crowd-sourced transcription tool

Digital Preservation Network (DPN)

Droid, Pronom, storage Resource Baker, iRODS

Drupal, Citation Fox, IL Fox, Movable Type

Drupal, Omeka, DSpace, APTrust, Digital Library Extension Service (DLXS), Copyright Review Management System (CRMS), MPach, VuFind, Sakai, Solr, Lucene, Kaltura

DSpace (3 responses)

DSpace and File Analyzer

DSpace, Kuali, Fedora Commons, Hydra, django

DSpace, Silverstripe

DSpace, Vireo, CORAL

Evergreen, Islandora, Docker

eXtensible Text Framework (XTF). The work is in progress as of the end of February 2014.

EZProxy Wondertool, Mondo License Grinder, Archivematica

Fedora Commons (2 responses)

Fedora Commons, DuraSpace, ArchivesSpace

Fedora Commons, Blacklight, Hydra, Avalon, Hydramata, ArchivesSpace, APTrust, Digital Preservation Network (DPN), Solrmarc, Tracksys

Fedora Commons, Islandora

Guide on the Side

Hydra (2 responses)

Hydra, Blacklight

Hydra, CORAL, MyLibrary

Hydra, Blacklight, Umlaut, Xerxes, Drupal, ArchivesSpace, Archivists' Toolkit, Capistrano

In-house link tracking software, in-house map software, other contributions to VuFind

IR+. eXtensible Catalog, DSpace

Islandora, Archivematica, ICA-AtoM

KentDSS https://github.com/ksulibraries/KentDSS

Kuali Financial Systems, Shibboleth

Kuali OLE, Sobek, ASERL Disposition Database, jrnl

Kuali OLE, Avalon Media System, Fedora Commons, Hydra, Hydramata, Variations Digital Music Library, METS Navigator, Sakai

Kuali OLE, Global Open Knowledgebase (GOKb), LOCKSS, Solr, VIVO

LOCKSS (Private LOCKSS network)

Manakin (DSpace)

Manitobia, DSpace, ICA-AtoM, Islandora, Fedora Commons, LOCKSS, Drupal, Open Journal System (OJS)

Omeka

One example: Viewshare

SRA toolkit, BLAST, C++ toolkit, variety of scientific tools

SubjectsPlus, Remixing Archival Metatdata Project (RAMP), Variations Digital Music Library System, Avalon, Kuali OLE

There’s a long list at https://github.com/gwu-libraries/

UCLA MWF, DSpace

VIVO, Fedora Commons

Voyager

20. Please indicate how your library is contributing to each of the following types of OSS projects. Check all that apply. N=56

Type of OSS Project	Code (i.e., developer time)	Money	Hosting	Other contribution	N/A	N
Institutional repository	32	18	5	10	14	52
Digital preservation	22	19	9	11	19	49
Digital asset management	20	8	4	5	26	48
Discovery layer	11	3	2	5	32	47
Publishing	5	5	5	3	34	47
ILS	6	5	—	7	37	46
Streaming media	7	4	2	3	37	46
Study room scheduler	5	—	—	1	39	45
Link resolver	3	1	1	1	41	45
Authentication/identity management	8	—	1	2	35	45
Inter-library loan	2	1	3	3	39	44
Data analysis	5	1	2	2	39	44

Type of OSS Project	Code (i.e., developer time)	Money	Hosting	Other contribution	N/A	N
Blogging	2	2	1	—	40	44
Electronic resource management	6	—	2	4	33	43
Course reserve	4	—	—	2	39	43
Floor maps	4	—	1	1	38	43
Data warehouse	6	—	2	1	37	43
ELMS	3	1	—	1	39	43
Visualization	4	1	1	2	39	43
Web analytics	3	—	1	1	38	43
Other type of project	15	5	2	6	16	30
Number of Responses	47	36	16	27	45	56

If you selected "Other contribution" above, please briefly describe the contribution the library makes to each corresponding project. N=25

Adding modules, patches as well as providing whole libraries (SRA toolkit, C++ toolkit, etc.)

Beta test institution

Blacklight: regularly host and organize committer calls; hosted Blacklight developer conference. Vireo: participate in the governance of the user community. Duraspace: Silver sponsors. Public Knowledge Project (PKP): Silver sponsors.

Both Kuali and Shibboleth are systems that are used university-wide. The Libraries is responsible for integrating these systems into our existing technology environment.

Consultation, organization

Contributing Omeka_a11y to the Omeka Project and ShadowPage, a page-turning plugin for content presentation in Omeka.

Contributing to and testing enhancements.

Creating software that intersects with OSS to enhance functionality.

Developing a crowd-sourced transcription tool.

Discovery layer, ILL, and "Other type of project": the library has contributed leadership, project management, governance, HR, financial management, and IT infrastructure support via the eXtensible Catalog Project, which developed four toolkits that fit within these various categories.

Feedback and bug reports for release candidates/new releases, contributing to support forms and listservs.

For both Citation Fox and IL Fox, library staff have provided training and given presentations at regional conferences.

Functional requirements, technical requirements, advisory role

Functional requirements, testing

ILS: project management, providing use cases. Electronic resource management: project management. Institutional repository: community membership.

Kuali OLE [ILS, ERM, Course Reserves]: participate to provide use cases, functional spec teams, testing of releases. Digital Music Library System, Avalon Media System [streaming media]: provide use cases, feedback on development priorities, release testing.

Legal advice, business/sustainability

Participation in architecture/design sessions, participation in pilot deployments

Release coordinator, educational efforts

Strategic direction, project management, research & development, grant
management

Streaming media: bug reporting and testing (Kaltura). Digital preservation: we manage and offer fee-based support to this project.

Testing, feature requests/requirements development

We have a heavily customized VuFind instance. We share our changes on a publicly accessible source control server, but we're not pushing our changes up to mainstream VuFind (our customizations are too local-specific).

We have contributed to community engagement, hosted community meetings, facilitated planning teleconferences, and advanced the designs, strategic plan, and architecture of these projects.

We have participated in testing the Fedora Commons repository software.

If you selected "Other type of project" above, please briefly describe the project and the corresponding contribution the library makes. N=15

Archival management system: contributed to support forums/listers

Bibapp: Campus Research Gateway and Expert Finder

Citation Fox is open source software that organizes citations into four broad categories. IL Fox is open source software that provides users with tools related to information literacy.

Developing a crowd-sourced transcription tool.

Digital Humanities, Digital Scholarship tools

ICS-AtoM Archival records management system: code development, testing, feature requests/requirements

Omeka is an online exhibit building tool that the Libraries are using to support Digital Scholarship in the arts, humanities, and social sciences.

Scientific data analysis, text mining

Social media viewing/sharing and harvesting for archives: coding, project and community management

SubjectsPlus [research guides, FAQs, staff directory, database A-Z]: primary code development, documentation, distribution, support. RAMP [used to generate authority records for creators of archival collections (using EAC-CPF) and then take that structured data and transform it into wiki markup to facilitate the creation or enhancement of Wikipedia pages for those creators; also facilitates examination of names/organizations for quality control, data visualization]: development, distribution, support.

The eXtensible Catalog's Metadata Services Toolkit is a platform to transform library metadata into a variety of formats. The library contributed in all of the above areas to the development of this software.

VIVO: researcher profiles

We also contribute to a project called VecNet that isn't library related.

We are eliminating frames and developing the capability for responsive web interface design. We anticipate this to be included in the next version release of XTF.

Website content management system (Silverstripe) module

21. Please indicate how many OSS projects the library has contributed to and for how many projects your library was the *primary* code contributor. N=50

	Minimum	Maximum	Mean	Median	Std Dev
Projects	1	20	4.64	3.00	3.95
Primary Code Contributor	0	20	1.86	1.00	3.11

22. Please indicate how many library staff and about what percent of their time are dedicated to contributing to the development of OSS projects. N=46

	Minimum	Maximum	Mean	Median	Std Dev
Staff	1	14	3.89	2.00	3.34
% of Time	0.05	90	30.67	25.00	25.61

Number of Library Staff	% of Time
1	0.05
1	3
1	5
1	5
1	5
1	10
1	10
1	25
1	30
1	50
1	50
1	60
2	3
2	5
2	5
2	10
2	10
2	20

Number of Library Staff	% of Time
2	25
2	25
2	25
2	50
2	50
2	80
3	10
3	20
3	50
3	90
4	5
4	25
4	90
5	10
5	50
5	50
5	55
6	4
6	25
7	50
8	10
8	15
8	80
10	20
10	50
10	60
12	varies
14	50

LIBRARY AS ORIGINAL DEVELOPER OF OSS PROJECTS

23. Is your library the original developer for any of the OSS project(s) in which you participate? N=56

Yes	32	57%
No	24	43%

If yes, please identify the software. N=31

Archivists' Toolkit, ArchivesSpace

Avalon Media System

Avalon Media System, Variations Digital Music Library, METS Navigator

Blacklight for displaying complex digital objects, Oral History Management Software

BLAST, C++ toolkit, SRA toolkit, PubReader

Citation Fox, IL Fox

Co-primary developer of Fedora Commons 4

Curator's Workbench

Custom Voyager Reports Server

Developing a crowd-sourced transcription tool

Discovery: a Solr-based discovery tool that generalizes an index, search, browse, and deliver framework that can work with content such as MARC records or EAD finding aids, but also including non-library context such as open access publication of scholarly research, and a working catalog of global language observations by an international community of scholars.

Digital Library Extension Service (DLXS)

DSpace

ETD-db, ETD-db 2.0

EZProxy Wondertool, Mondo License Grinder

Guide on the Side

https://github.com/ksulibraries/KentDSS

Hydra (parts of), CORAL, MyLibrary, VecNet

In coordination UVa with Cornell: Fedora Commons; in coordination UVa with Stanford and University of Hull: Hydra; UVa: Blacklight; UVa: Solrmarc; UVa:Tracksys; in coordination UVa with Roy Rosenzweig Center for History and New Media: Neatline.

IR+. eXtensible Catalog

RAMP SubjectsPlus

See https://github.com/gwu-libraries

Simple Archive Format Packager: a tool to support batch ingest of content into the institutional repository (DSpace) (in Java)

Sobek, ASERL Disposition Database, jrnl

Sufia (a Hydra-based repository application)

Suma (mobile space assessment toolkit), lentil (Instagram viewing/sharing, and harvesting for archives), djatoka Ruby gem (Image server wrapper)

Umlaut was originally developed by Ross Singer. We took it over very early on and have been the principal developers since. Our library is the primary developer for the Data Conservancy.

Viewshare is the LC instance of the Recollection OSS software, so not totally created ab novo at LC but considered an LC product now.

Vireo, Collaborative Book Reader (CoBRe)

VuFind, Papyrus, Isladora

We created link-tracking software and map software that is OSS but currently only in small release (code shared upon request). We plan to clean up these projects (and several others) to move them to a public GitHub repository.

Please indicate how important each of the following reasons for deciding to open source the project is to your library. Please make one selection per row. N=43

Reasons	1 Not Important	2	3	4	5 Very Important	N
Shared effort in development and quality assurance of the product	4	5	7	13	14	43
A desire to contribute to an open source community	1	3	10	15	14	43
A belief that open sourcing would lead to better software	1	6	5	17	13	42
A need for expertise not available in your institution	11	9	11	6	4	41
At the request of another institution	14	7	12	6	2	41
Other reason(s)	2	—	1	3	6	12
Number of Responses	22	23	29	31	31	43

If you indicated above that the library has other reason(s) for deciding to open source the project, please briefly describe the reason(s). N=10

Ability for others to adapt tools to meet their needs. Provide support for platforms and services that are not required by our institution.

Assistance with ongoing sustainability of the product.

Demonstrate expertise of library staff to project in a non-library context. Develop an alternative business to deepen the libraries' engagement with researchers and scholars.

How good the system is.

Need for tools not otherwise available.

Other libraries have shared generously before us. We have the expertise and feel some duty to share alike.

Requirements of granting agencies that software developed with grant funds be shared under an open source license.

Risk reduction with resourcing, sustainability and exit strategy.

There was nothing available at the time that ETD-db was developed. Its recent rewrite was entirely for the external use community.

Training aid, set an example

COST OF CONTRIBUTING TO OSS PROJECTS

24. Were you able to track the costs of your most recent contribution to an OSS project? N=53

Yes	10	19%
No	43	81%

If yes, please identify the most recent OSS project, indicate the costs of contributing to that project, and briefly describe what expenses were covered (e.g., staff time, equipment, training, travel, etc.) N=10

OSS Project	Costs	Expenses Covered
Avalon Media System	Not available	Travel to meetings and conferences
Crowd-sourced transcription tool	$7500	Consultant, in-house staff time
Custom Voyager Reports Server	Staff time and equipment	Staff time and equipment
DSpace REST API	Approx. $10,000	Salary/benefits (2 months developer time)
Fedora Commons 4	Pending	Pending
Fedora Commons	We cannot share cost information at this time.	We cannot share cost information at this time.
Open Journal System (OJS)	5% of developer time	Staff time, travel
Open Journal System (OJS)	$2750	Conduct design work, client meetings, programming, testing, troubleshooting, and documentation
Papyrus	N/A	Staff time
Vireo	1 FTE for 1 year	Wages, travel, training

What was the source of the funds for contributing to this OSS project? Check all that apply. N=45

Library's operating budget	43	96%
Grant(s)	10	22%
Parent institution	3	7%
Consortial budget(s)	2	4%
Gift(s)	1	2%
Other funding source(s)	2	4%

Please specify the other funding source(s). N=2

Able to track, chose not to track. Would come from library's operating budget.

Funded by another university division (Technology Services)

BENEFITS AND CHALLENGES OF CONTRIBUTING TO OSS PROJECTS

25. Please briefly describe up to three benefits your library enjoys as a result of contributing to OSS projects. N=44

Ability to enhance product and influence its direction. Sharing with community.

Ability to influence project outcome.

Ability to lend expertise to peer or smaller institutions. Mutual benefit from reusing working solutions.

Avoids data lock-in. While it may not be any less expensive/time consuming to migrate data out of an open source system than a proprietary system, at least with open source there will always be the technical possibility. User communities and developer communications tend to be better formed, enabling better DIY support, and not being totally reliant on a single vendor. Open source values (access to and right to share information) map closely to library values.

Becoming an active part of worthwhile communities. Helping make products we and others use better. Increase our skills and expertise and inspire productive creativity.

Better service offerings. Alignment with institute mission. Collaboration with non-library departments and peer institutions.

Broadens their perspective as developers, product owners, and project managers. Meets the strategic needs of the organization to engage with the world and our communities. Helps us build better solutions with like-minded people and institutions.

Collaborating with other institutions to address common areas of need. Involvement of library staff in intellectually engaging and useful work. Ending up with a more sustainable product than if we had done it just on our own.

Collaboration of common tasks. Faster return on requested features. Giving back.

Community is able to benefit from our developments. Forces us to write cleaner code that is generalizable and fits with our strategies for replaceable parts.

Contributing code helps to meet our specialized needs. We participate in a community of experts. Contributing to the project is in accordance with the Libraries' and university's mission.

Contributing to the library community. Developing local expertise. Recognition.

Contributing, even in a small way, to non-commercial, inexpensive, and highly functional alternatives to expensive commercial software that drain our budgets. Good press for the university, and for the Libraries. Providing software to fill needs of other institutions.

Control of product design. Functionality meets our needs.

Credibility in OSS Developer community. Ability to share problems. Modeling good behavior.

Customization for our exact needs.

Enhanced quality of software through collaboration. Leveraging effort from multiple institutions. Ability to use work from other organizations.

Ensures product remains stable and useful. Fulfill our obligation as a user of the OSS. Improved understanding of the OSS.

Freedom to use, study, copy, modify, redistribute our solutions. Participation in a broader community. Visibility in that community as a contributor.

Functionality that best meets our needs is built into the software. Community participation. Identification and reporting of bugs and new features.

Gain respect as industry leader. Community enrichment. Education.

Good library citizens/community contribution. Having features released that we require. Exposure to new ideas and professional learning and sharing from a broader community.

Increased visibility. Added enhancements.

Institutional needs more likely to be accommodated.

Institutional recognition. Creating a better product than what was currently available. Opportunities for collaboration both within the US and abroad.

Latest software releases. Ability to help steer direction of software development. Ability to tailor software to local needs.

Our monetary contribution helps to sustain the open source federation.

Prestige. Providing direction. Collegial atmosphere.

Pride. Forces rigor.

Providing flexible solutions to solve common library issues or service requirements. Professional development of team members & providing exciting/challenging work environment.

Recognition. Control of budget.

Recognition and community building. Opportunity to influence product development.

Recognition as a source of expertise. Input into direction of software development.

Reduced support costs: others can adapt tools rather than requesting us to make changes. Ability for others to enhance and expand on previous efforts.

Safety in numbers: use helps to ensure viability of the solution. Revenue from offering support. Bug reports and occasional code contributions.

Shared development

Staff development. Reputation. Collaboration building.

Sustainable solutions: together we go farther. Sum is greater than the parts: quality solutions that meet our needs. Investment in our staff: more meaningful work, deepening skills, end of isolation.

Tool is available to meet our needs. Customizability. Ability to add features as needed.

Visibility and participation in the community. Investments benefit other libraries and can lead to partnerships, other collaboration.

We are part of the OSS community.

We helped the Avalon and Variations projects through testing.

We use software to solve our problems that others have written. Better code is written when you have an external audience of coders reviewing your contribution. There's lots of it that's relevant to an academic library.

We want to be able to influence the direction of the effort to align it with our needs. By participating in a larger community, we can contribute the good ideas of our staff and in turn learn from the good ideas of others.

26. Please briefly describe up to three challenges your library encountered as a result of contributing to OSS projects and the strategies employed to overcome these challenges. N=37

Adhering to community standards that differ from in-house. Committing the resources to develop contributions. Understanding the code base and requirements according to the community need.

Agreement of product direction. Coordinate development.

Assessing value to OSS project. Confidence in coding standards. Compliance with OSS review process.

Contribution of developer time can compete with other local project priorities. Remote/asynchronous collaboration: might have to wait a long time for responses. No clear, quantifiable ROI.

Coordinating effort across institutions challenging/varying opinions on functionality. Finding financial sources. Maintaining and supporting software.

Coordination/management of developers. Getting good functional requirements.

Developer/programmer will graduate. Staff required to learn programming of system. Need to document every phase.

Developing a product that is generic enough to meet needs of multiple institutions. Supporting and growing the community around the project. Sustainability: securing ongoing funding to support the software.

Difficult to make substantial contribution without more dedicated time to devote to it.

Extra time. Convincing stakeholders of value. Coming to terms with applicable licensing models.

Finding staff time to contribute. Disconnect between OSS priorities, which may be based on the funder's priorities, and our institutional needs. Ongoing financial commitment as OSS moves to a community source model.

Finding time and resources to devote to development process. Feature creep.

Finding time to contribute. Time to support and answer questions. Removing localization.

Getting library staff familiar with OSS/collaborative ways of working. Lack of control of timelines of collaborative OSS projects, need to readjust expectations. Not enough staff time to both participate actively in OSS projects and continue local responsibilities.

Increased time spent in detailed documentation.

Internal buy-in to benefit of time spent on OSS projects: communication about project at all levels of institution; reaching out to potential stakeholders early in process. General Consul was concerned about our distribution of code, especially with development contributed by faculty who don't have code development built into their job description. The faculty had to sign a release before we could contribute the code.

It can take more work to contribute well to a public project, but that can tend to produce better results. We need to review legal guidelines around assigning copyright to external organizations.

It is more expensive to write code that is generalizable than custom code for your institution. The development process is slower and requires a higher mind.

Larger than expected contribution time required of local resources.

Legal and licensing issues. Strategy: involvement of in-house legal expertise (our Director of Copyright and Digital Scholarship) and coordination with the university Technology Transfer office. Need to provide support or decide how much support to provide. Strategy: clearly communicate expectations regarding level of support provided. Need to support a wider range of environments than would be necessary for an internal-only deployment. Strategy: reducing over-dependence on current architecture can actually reduce costs over the full life of a project.

Maintenance of contributed code to fill the needs of the outside community. Monitoring feedback through multiple channels (pull requests, forum posts, IRC, etc.)

Managing expectations, sometimes you have to compromise. Strategy: engage with people and be transparent. Determining which projects to engage and to what degree. Strategy: stay connected at a management level, know your strategic objectives, know your staff and what culture is a good fit for your resources. Resources. Strategy: be able to show value toward strategic objectives for the resource investment.

Meeting expectations of adopters when we are the primary contributors.

More meetings take time away from local development.

Not having solid business models to refer to showing the real costs of developing, supporting, using OSS. Not being able to devote enough staff effort to OSS projects. When they are on a project less than 50% their return on investment is not as great. Getting institutional support beyond the library for certain solutions. Many administrators seem to prefer vendor provided out-of-the-box solutions.

Opportunity cost: developers not able to contribute to local initiatives.

Partner reliability.

Product was too narrowly focused for our exact needs to be worthy of sharing out to the community.

Some open source applications don't have formal paid support options available, so support risks are transferred from a vendor to the institution: careful evaluation of the risk, and level of risk before making the decision to do an OSS project. Sometimes a lack of understanding that open source doesn't equal free. The cost to the institution may be the same or even greater than a proprietary solution, just the money is spent on different aspects of the project: discussions with library stakeholders to make sure everyone clearly understands the full cost of OSS projects. Lack of institutional understanding to the open source model and licenses can hinder contributions of code back to the community.

Staff time: we just juggle this part with regular projects.

Support requests related to OSS projects takes some time.

Time and effort for creating it. Maintenance.

Time and resource commitment

Time spent to keep track of project.

Time to develop: fit in around other responsibilities. Time to support/answer questions: make part of professional development responsibilities.

Time: overcome only by choosing not to move forward on other projects at that time.

Uses valuable staff time: overcome by making sure we only contribute time we can afford and/or that will provide a desirable return on investment.

TOOLS FOR OSS PROJECTS

27. Does your library use a public repository or forge (e.g., Github, Sourceforge, Google Code, Bitbucket) to share your open source code? N=52

Yes	41	79%
No	11	21%

If yes, please identify the repository or forge. N=41

Github	38
Google Code	3
SourceForge	3
Bitbucket	2
Drupal GIT	1
RedMine	1
Subversion	1

Comments

Currently not, but we're moving to Github.

We're exploring doing this in a more standardized, regular way, but are exploring security concerns.

28. What tools does your library use to facilitate collaboration on the OSS projects your library contributes to? Check all that apply. N=45

Shared version control	37	82%
An issue tracking software package	36	80%
A mailing list	32	71%
A wiki	25	56%
A forum	12	27%
Other tool(s)	10	22%

Please briefly describe the other tool(s) your library uses to facilitate collaboration on OSS projects. N=10

Conference calls

Google Docs

IRC

IRC for chat collaboration

IRC, Google Hangouts, Adobe Connect, Skype

PivotalTracker

Project management tools (e.g., Trello)

Skype

Trello

Virtual tools for the team, project management software

LICENSING MODEL FOR DISTRIBUTION OF OSS

29. What licensing models does your organization recommend for distribution of software? Check all that apply. N=42

GNU Public License (GPL) version 3	16	38.1%
Apache	15	35.7%
Creative commons	15	35.7%
MIT	12	28.6%
GNU Public License (GPL) version 2	11	26.2%
BSD 3 Clause	3	7.1%
BSD 2 Clause	2	4.8%
Other licensing model	12	28.6%

Please briefly describe the other licensing model. N=12

Educational Community License (ECL) - ECL 2 (2 responses)

Educational Community License (ECL)

I wouldn’t say that we’ve come across this very often or that we have a strong opinion of which licenses to recommend. If asked, I’d recommend that we evaluate these options and use the license that best fits the software. Much of the code we write falls under the license used by the platform or libraries that we leverage. Further, we haven’t really been open sourcing any internally developed applications.

Internally developed Rights Statement based very closely on CC.

OSS produced at LC is generally considered federal work product and public domain.

Public domain

Public domain (Creative Commons - CC 0)

There is no organizational policy on licensing models.

This is just what we’ve used; there is no standard license that we would necessarily recommend.

We don't recommend it per se, rather we use an MIT-style license on our own software, as approved by the university.

We have no formal recommendation.

OSS PROJECT ASSESSMENT

30. Please indicate how important each of the following indicators that your contribution to an OSS project has been successful is to your library. Please make one selection per row. N=51

Reasons	1 Not Important	2	3	4	5 Very Important	N
The functionality better suits our institution's needs	—	—	1	8	41	50
Amount of community contribution/involvement	1	8	14	17	10	50
Number of project adopters	2	8	15	18	7	50
Number of project releases	4	11	23	9	3	50
Ease of support	—	2	21	15	11	49
Staff time savings	5	7	17	14	6	49
Monetary savings	4	13	10	17	5	49
Other indicator(s)	2	—	1	1	1	5
Number of Responses	11	22	45	40	46	51

If you indicated above that the library relies on other indicator(s) that your contribution to an OSS project has been successful, please briefly describe the indicator(s). N=3

Community interest in project [altmetrics, conference presentations, articles]

We are concerned to ensure that software systems are section 508 compliant, this indicator of success is not necessarily subsumed under "functionality."

Sustainability in terms of direction and responsiveness to meet evolving needs.

Additional Comments

Again, we don't agree that OSS results in staff time savings or ease of support, so did not respond to those two statements.

Who has adopted, and not just the number of adopters.

LIBRARY DOESN'T USE OSS

31. Please briefly describe why your library is not using any open source software. N=2

We don't have a sufficient IT support to develop, customize, and maintain OSS software.

We have not done any major software selection processes in over five years, and the OSS products have not historically had the functions we required. That may be changing looking forward.

ADDITIONAL COMMENTS

32. Please enter any additional information that may assist the survey authors' understanding of your library's use of open source software. N=19

I forgot to add that we developed a collection directory application, currently used for two projects, WAAND (Women Artists Archives National Directory) and NAP (Newark Archives Project).

Last August we hired a programmer with Drupal skills to assist in the library's web site redesign. We are trying to get colleagues to use Gimp because the licensing fees for Adobe Photoshop are prohibitive. Needless to say, Gimp is not being well received, yet. The campus and university system procurement office is trying to negotiate a campus and system-wide license.

OSS allows for greater customizations that fulfill the needs of so many library patrons and employees. We are lucky enough to have enough staff to get started on these projects, but it was very important for us to agree on some core OSS elements to make it easier to maintain in the long run. A good example of this our use of PHP and Apache. Focusing on this as a core allows for a smaller number of programmers to turn out and support a large number of applications. I will note that we have a smaller use for MySQL as there is a significant cost reduction in licensing Microsoft SQL for the university system. Therefore, we are not in the norm in that our Linux, PHP, and Apache works more with Microsoft SQL than MySQL.

OSS is a cost effective way to provide solutions that can be customized to local needs. The various components can be used to build products and solutions large and small. A staff of skilled software developers is required to use the tools, and products. It also requires system support staff to learn and support new tools, especially database systems.

OSS is used to support operations. Currently, not a major focus. Generally not using because of development and maintenance costs (staff time).

The availability of staff skilled in OSS technology remains the one hurdle to implementing more OSS as a strategy for the library. There is great interest in utilizing OSS more widely as a part of our technology strategy, but balancing availability of skillsets vs. demand will be challenging.

The Libraries and Academic Computing and Networking Services (ACNS) both report to the Vice President for Information Technology/Dean of Libraries.

The library has the will to participate in OSS if we had the staff time and resources to commit to OSS projects.

The use of OSS is very important to our mission, resource, and risk management.

This survey didn't ask about future projections of OSS use. We currently have DSpace but are devoting several full-time staff to developing Fedora Commons and Hydra. IT staff are divided between the ITS department and the Center for Digital Research and Scholarship.

We are a typical large research university. The use of OSS for interface to the digital library (REST APIs) allow for our research faculty to create content with whatever tools they are comfortable with. We encourage use of our standards, but if they use the API, they can do what they please with our digital assets.

We are very supportive of OSS but ultimately use the products that best meet our needs. Sometimes this is OSS but sometimes it is a commercial vendor product as there are advantages and disadvantages to both.

We believe in it deeply. It's what we do. We'd be up a creek without it.

We have no preference for OSS over vendor software. We use what works best and what we can afford.

We learned (the hard way) from our first experience with putting OSS developed elsewhere into production (about 10 years ago) that having vendor support and an active community around an OSS application are very important. With the OSS that we have developed locally (eXtensible Catalog and IR+), we have been unable to provide either of these things to potential users of our software, and have thus found ourselves in this same position with our own software of being unable to sustain the software on our own. While we still strongly support OSS and continue to implement additional OSS applications, we now make sure that vendor support and an active user community are already in place before we proceed with deploying the software.

We take a broad view of OSS and answered based on that approach, not limiting the scope to library-specific OSS. Our answers would be different were this more clearly defined, perhaps. Also, it suffices to say that our philosophy is simple: open source first, vendor only when there's no viable OS option. For example, we run our own data centre, and for that infrastructure from operating system to virtualization platform, it is all OS; there's no VMware, Citrix, etc.

We're transitioning from using mostly closed software to preferring mostly open software, so we're not yet where we want to be. We're working out more formal policies with campus technology transfer to allow us to release GPL software at our own discretion. We choose to use more OSS than vendor software because we have a tight budget but a great IT staff. With much of our software support burden being internal, it doesn't leave a lot of time to take the extra steps to polish, release, and support OSS software. But it's still a major goal for us.

While we use OSS, our unwritten policy is to use hosted, out-of-the-box solutions wherever possible. OSS is used to fill in the gaps.

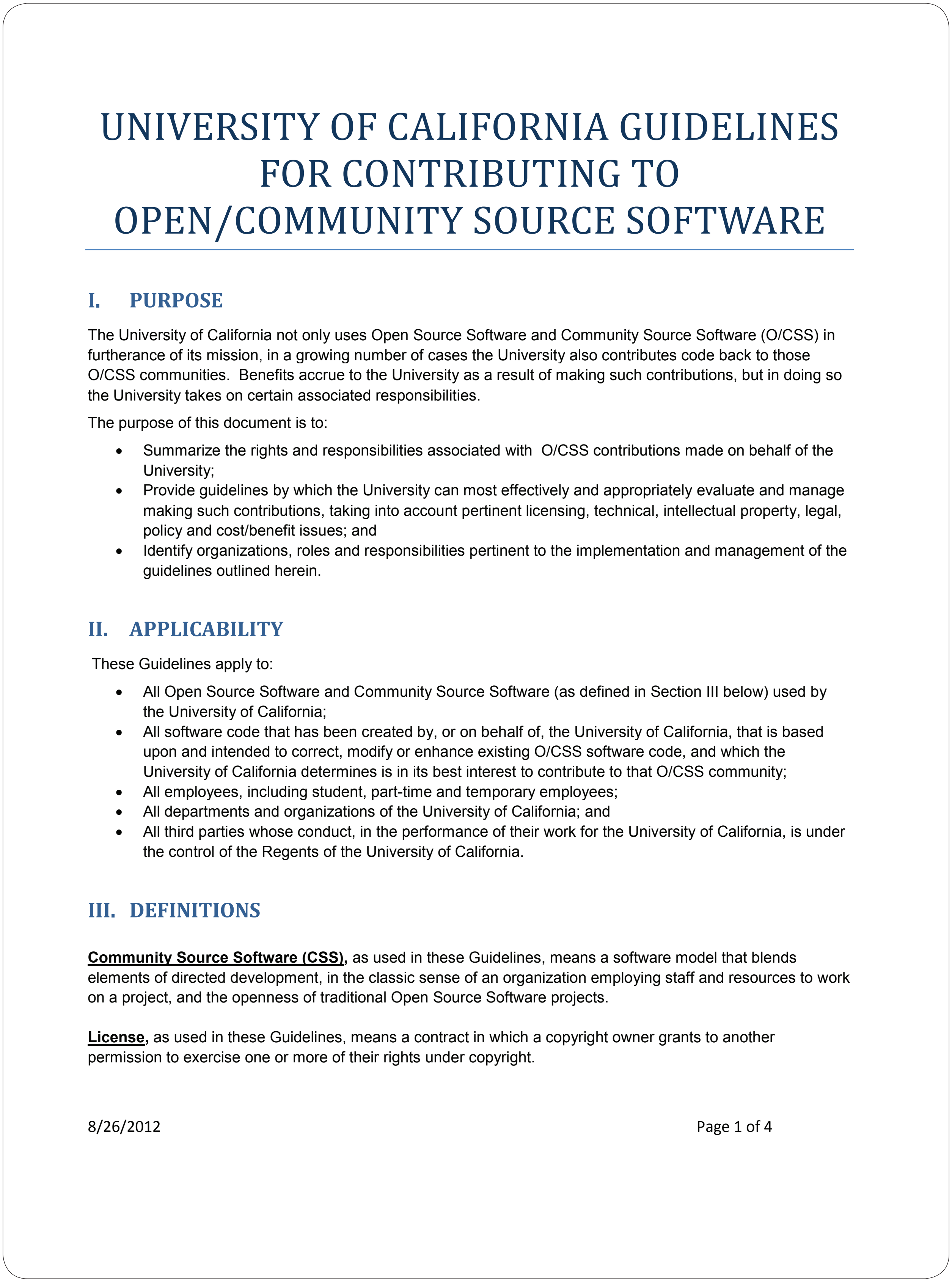
RESPONDING INSTITUTIONS

Arizona State University
Auburn University
Boston Public Library
Boston University
Brigham Young University
University of British Columbia
Brown University
University of Calgary
University of California, Irvine
University of California, Los Angeles
Case Western Reserve University
University of Colorado at Boulder
Colorado State University
Columbia University
University of Connecticut
Duke University
University of Florida
George Washington University
Georgetown University
Georgia Institute of Technology
University of Guelph
University of Hawaii at Manoa
Indiana University Bloomington
Iowa State University
Johns Hopkins University
University of Kansas
Kent State University
University of Kentucky
Library of Congress
University of Louisville
McMaster University
University of Manitoba
University of Maryland
Massachusetts Institute of Technology
University of Miami
University of Michigan
National Archives and Records Administration
National Library of Medicine
University of Nebraska–Lincoln
University of New Mexico
New York University
University of North Carolina at Chapel Hill
North Carolina State University
Northwestern University
University of Notre Dame
Ohio University
Ohio State University
University of Oklahoma
Oklahoma State University
University of Oregon
University of Ottawa
University of Pennsylvania
Pennsylvania State University
Purdue University
Queen's University
University of Rochester
Rutgers University
University of Saskatchewan
University of Southern California
Southern Illinois University Carbondale
University at Albany, SUNY
Syracuse University
Temple University
University of Tennessee
Texas A&M University
Texas Tech University
Tulane University
University of Utah
Vanderbilt University
University of Virginia
Virginia Tech
Washington State University
Washington University in St. Louis
Western University
University of Wisconsin–Madison
York University

REPRESENTATIVE DOCUMENTS

OSS Contributor Agreements

UNIVERSITY OF CALIFORNIA
Guidelines for Contributing to Open/Community Source Software
http://uctas.ucop.edu/documents/uc-guidelines-contributing-oss-communities.pdf



UNIVERSITY OF CALIFORNIA GUIDELINES FOR CONTRIBUTING TO OPEN/COMMUNITY SOURCE SOFTWARE

I. PURPOSE

The University of California not only uses Open Source Software and Community Source Software (O/CSS) in furtherance of its mission, in a growing number of cases the University also contributes code back to those O/CSS communities. Benefits accrue to the University as a result of making such contributions, but in doing so the University takes on certain associated responsibilities.

The purpose of this document is to:

- Summarize the rights and responsibilities associated with O/CSS contributions made on behalf of the University;
- Provide guidelines by which the University can most effectively and appropriately evaluate and manage making such contributions, taking into account pertinent licensing, technical, intellectual property, legal, policy and cost/benefit issues; and
- Identify organizations, roles and responsibilities pertinent to the implementation and management of the guidelines outlined herein.

II. APPLICABILITY

These Guidelines apply to:

- All Open Source Software and Community Source Software (as defined in Section III below) used by the University of California;
- All software code that has been created by, or on behalf of, the University of California, that is based upon and intended to correct, modify or enhance existing O/CSS software code, and which the University of California determines is in its best interest to contribute to that O/CSS community;
- All employees, including student, part-time and temporary employees;
- All departments and organizations of the University of California; and
- All third parties whose conduct, in the performance of their work for the University of California, is under the control of the Regents of the University of California.

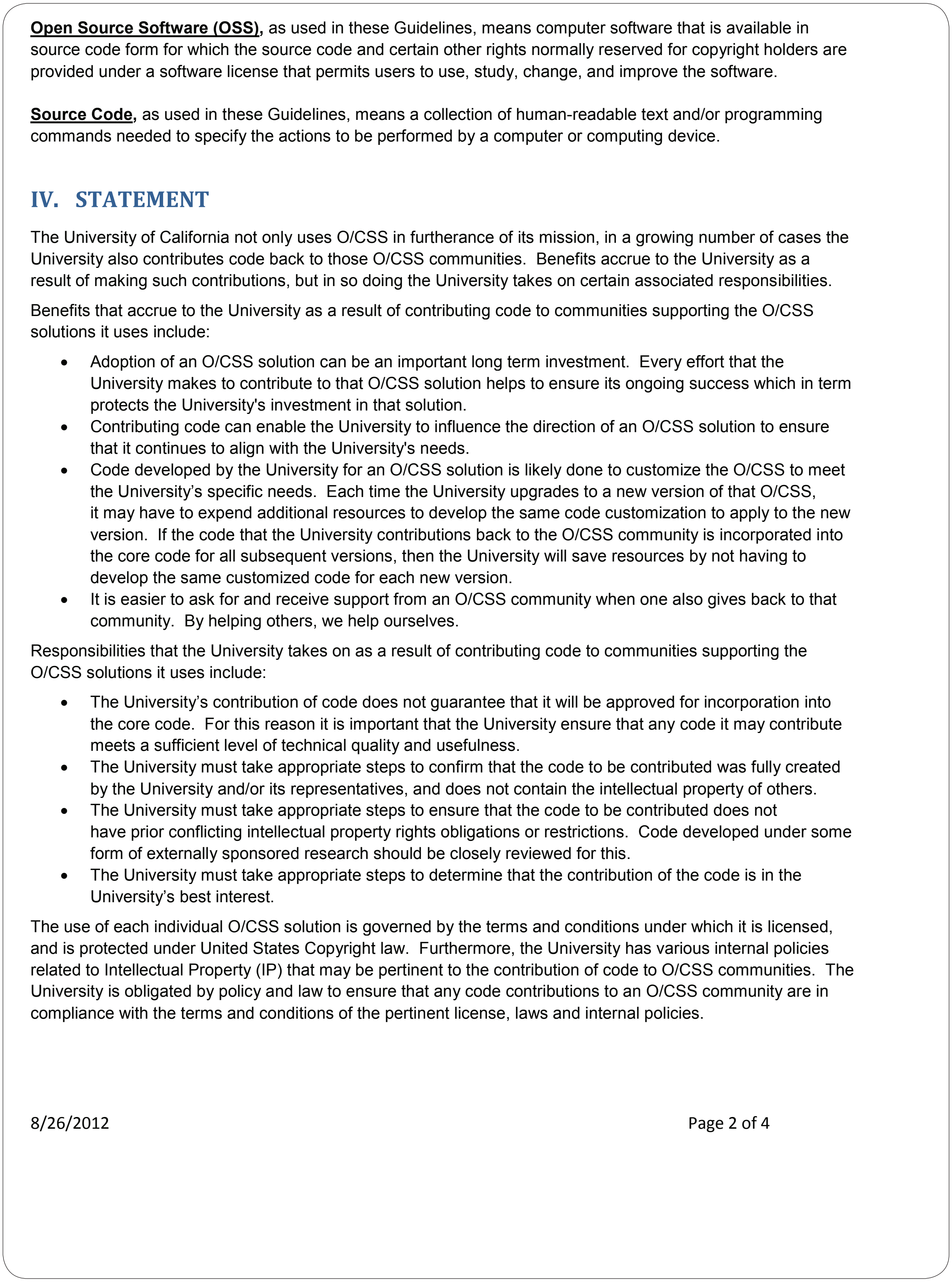
III. DEFINITIONS

Community Source Software (CSS), as used in these Guidelines, means a software model that blends elements of directed development, in the classic sense of an organization employing staff and resources to work on a project, and the openness of traditional Open Source Software projects.

License, as used in these Guidelines, means a contract in which a copyright owner grants to another permission to exercise one or more of their rights under copyright.

UNIVERSITY OF CALIFORNIA

Guidelines for Contributing to Open/Community Source Software
http://uctas.ucop.edu/documents/uc-guidelines-contributing-oss-communities.pdf



Open Source Software (OSS), as used in these Guidelines, means computer software that is available in source code form for which the source code and certain other rights normally reserved for copyright holders are provided under a software license that permits users to use, study, change, and improve the software.

Source Code, as used in these Guidelines, means a collection of human-readable text and/or programming commands needed to specify the actions to be performed by a computer or computing device.

IV. STATEMENT

The University of California not only uses O/CSS in furtherance of its mission, in a growing number of cases the University also contributes code back to those O/CSS communities. Benefits accrue to the University as a result of making such contributions, but in so doing the University takes on certain associated responsibilities.

Benefits that accrue to the University as a result of contributing code to communities supporting the O/CSS solutions it uses include:

- Adoption of an O/CSS solution can be an important long term investment. Every effort that the University makes to contribute to that O/CSS solution helps to ensure its ongoing success which in term protects the University's investment in that solution.
- Contributing code can enable the University to influence the direction of an O/CSS solution to ensure that it continues to align with the University's needs.
- Code developed by the University for an O/CSS solution is likely done to customize the O/CSS to meet the University's specific needs. Each time the University upgrades to a new version of that O/CSS, it may have to expend additional resources to develop the same code customization to apply to the new version. If the code that the University contributions back to the O/CSS community is incorporated into the core code for all subsequent versions, then the University will save resources by not having to develop the same customized code for each new version.
- It is easier to ask for and receive support from an O/CSS community when one also gives back to that community. By helping others, we help ourselves.

Responsibilities that the University takes on as a result of contributing code to communities supporting the O/CSS solutions it uses include:

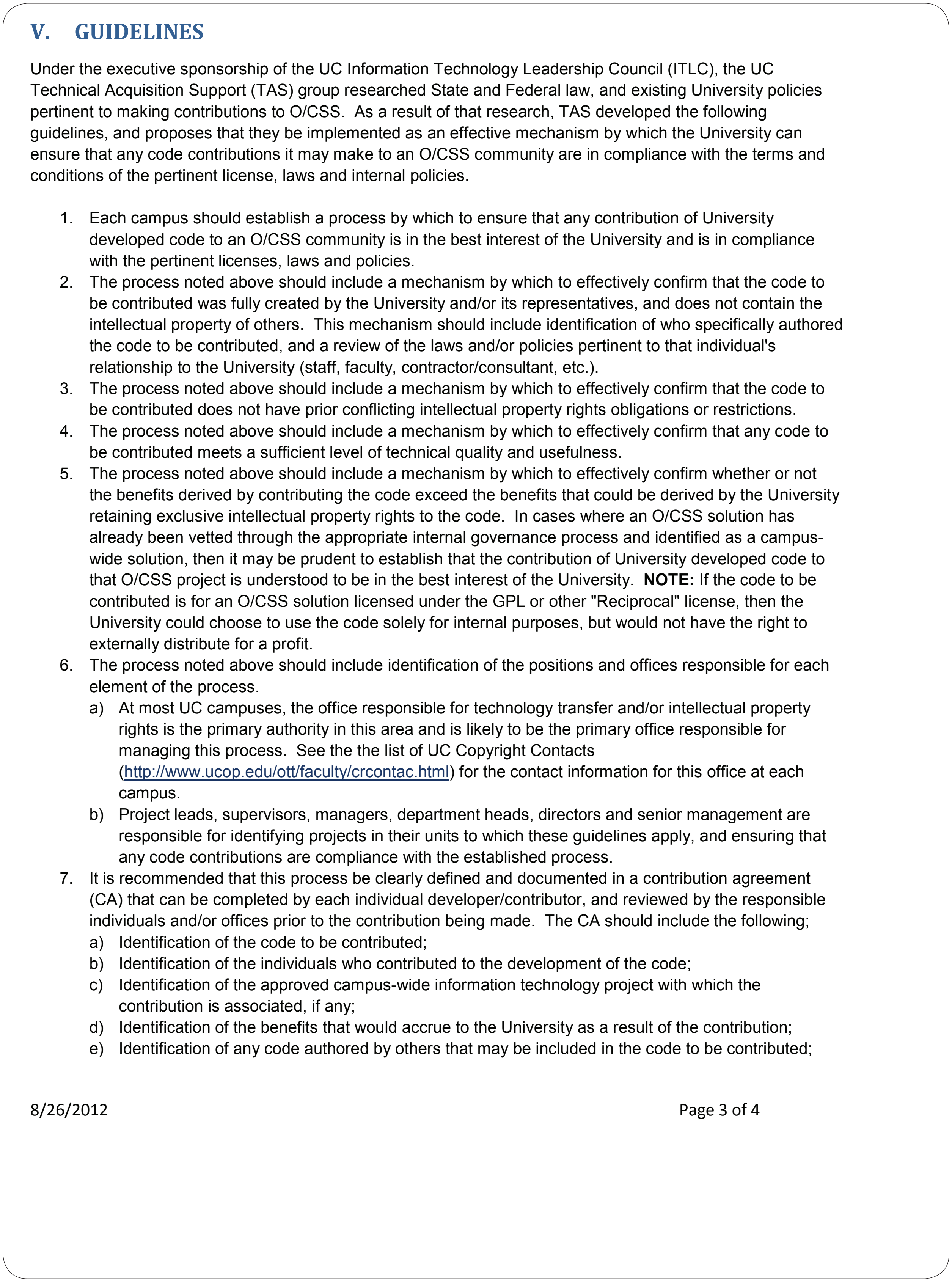
- The University's contribution of code does not guarantee that it will be approved for incorporation into the core code. For this reason it is important that the University ensure that any code it may contribute meets a sufficient level of technical quality and usefulness.
- The University must take appropriate steps to confirm that the code to be contributed was fully created by the University and/or its representatives, and does not contain the intellectual property of others.
- The University must take appropriate steps to ensure that the code to be contributed does not have prior conflicting intellectual property rights obligations or restrictions. Code developed under some form of externally sponsored research should be closely reviewed for this.
- The University must take appropriate steps to determine that the contribution of the code is in the University's best interest.

The use of each individual O/CSS solution is governed by the terms and conditions under which it is licensed, and is protected under United States Copyright law. Furthermore, the University has various internal policies related to Intellectual Property (IP) that may be pertinent to the contribution of code to O/CSS communities. The University is obligated by policy and law to ensure that any code contributions to an O/CSS community are in compliance with the terms and conditions of the pertinent license, laws and internal policies.

8/26/2012 Page 2 of 4

UNIVERSITY OF CALIFORNIA

Guidelines for Contributing to Open/Community Source Software
http://uctas.ucop.edu/documents/uc-guidelines-contributing-oss-communities.pdf



V. GUIDELINES

Under the executive sponsorship of the UC Information Technology Leadership Council (ITLC), the UC Technical Acquisition Support (TAS) group researched State and Federal law, and existing University policies pertinent to making contributions to O/CSS. As a result of that research, TAS developed the following guidelines, and proposes that they be implemented as an effective mechanism by which the University can ensure that any code contributions it may make to an O/CSS community are in compliance with the terms and conditions of the pertinent license, laws and internal policies.

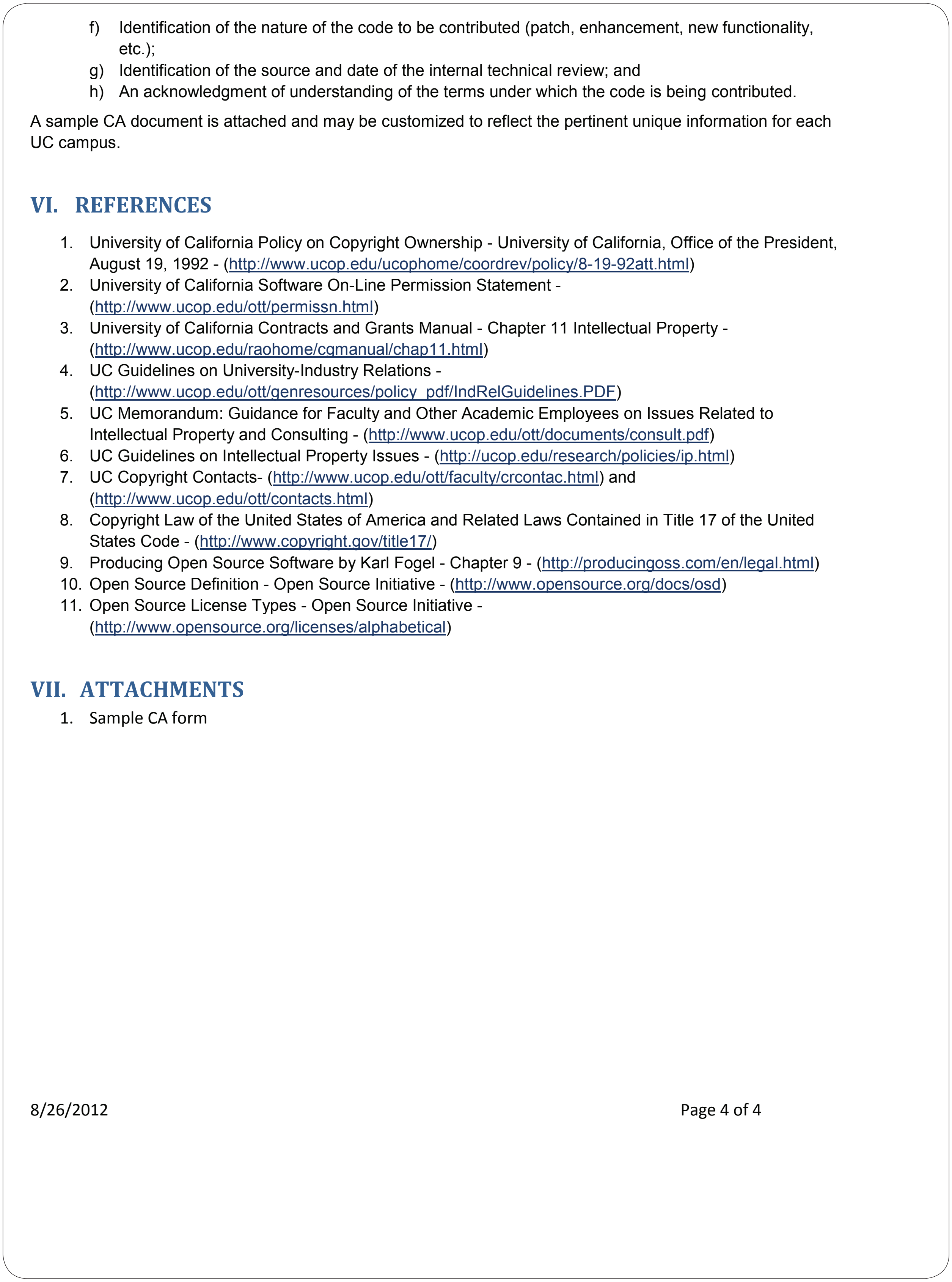
1. Each campus should establish a process by which to ensure that any contribution of University developed code to an O/CSS community is in the best interest of the University and is in compliance with the pertinent licenses, laws and policies.
2. The process noted above should include a mechanism by which to effectively confirm that the code to be contributed was fully created by the University and/or its representatives, and does not contain the intellectual property of others. This mechanism should include identification of who specifically authored the code to be contributed, and a review of the laws and/or policies pertinent to that individual's relationship to the University (staff, faculty, contractor/consultant, etc.).
3. The process noted above should include a mechanism by which to effectively confirm that the code to be contributed does not have prior conflicting intellectual property rights obligations or restrictions.
4. The process noted above should include a mechanism by which to effectively confirm that any code to be contributed meets a sufficient level of technical quality and usefulness.
5. The process noted above should include a mechanism by which to effectively confirm whether or not the benefits derived by contributing the code exceed the benefits that could be derived by the University retaining exclusive intellectual property rights to the code. In cases where an O/CSS solution has already been vetted through the appropriate internal governance process and identified as a campus-wide solution, then it may be prudent to establish that the contribution of University developed code to that O/CSS project is understood to be in the best interest of the University. **NOTE:** If the code to be contributed is for an O/CSS solution licensed under the GPL or other "Reciprocal" license, then the University could choose to use the code solely for internal purposes, but would not have the right to externally distribute for a profit.
6. The process noted above should include identification of the positions and offices responsible for each element of the process.
   a) At most UC campuses, the office responsible for technology transfer and/or intellectual property rights is the primary authority in this area and is likely to be the primary office responsible for managing this process. See the the list of UC Copyright Contacts (http://www.ucop.edu/ott/faculty/crcontac.html) for the contact information for this office at each campus.
   b) Project leads, supervisors, managers, department heads, directors and senior management are responsible for identifying projects in their units to which these guidelines apply, and ensuring that any code contributions are compliance with the established process.
7. It is recommended that this process be clearly defined and documented in a contribution agreement (CA) that can be completed by each individual developer/contributor, and reviewed by the responsible individuals and/or offices prior to the contribution being made. The CA should include the following;
   a) Identification of the code to be contributed;
   b) Identification of the individuals who contributed to the development of the code;
   c) Identification of the approved campus-wide information technology project with which the contribution is associated, if any;
   d) Identification of the benefits that would accrue to the University as a result of the contribution;
   e) Identification of any code authored by others that may be included in the code to be contributed;

8/26/2012 Page 3 of 4

UNIVERSITY OF CALIFORNIA

Guidelines for Contributing to Open/Community Source Software

http://uctas.ucop.edu/documents/uc-guidelines-contributing-oss-communities.pdf



f) Identification of the nature of the code to be contributed (patch, enhancement, new functionality, etc.);

g) Identification of the source and date of the internal technical review; and

h) An acknowledgment of understanding of the terms under which the code is being contributed.

A sample CA document is attached and may be customized to reflect the pertinent unique information for each UC campus.

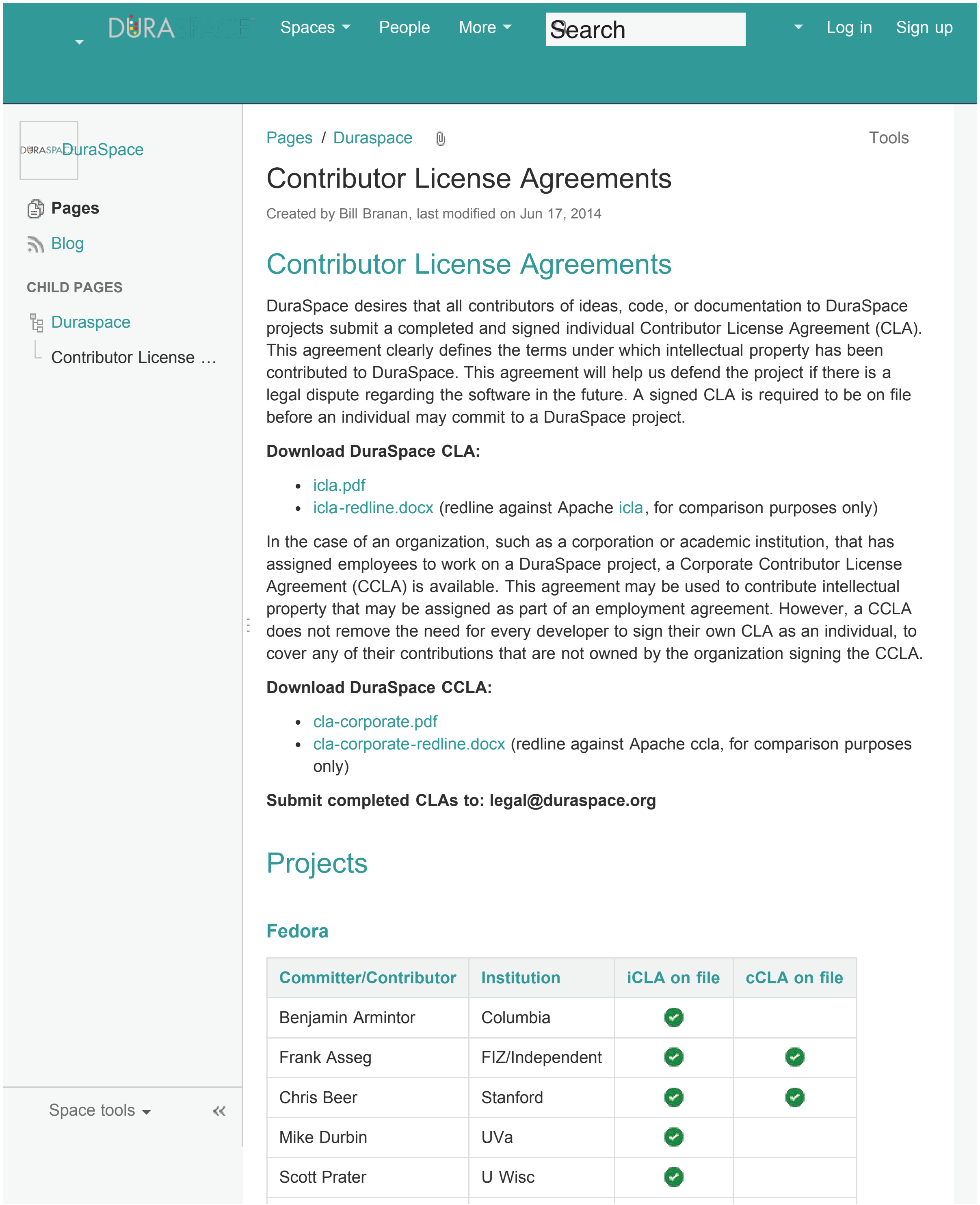
VI. REFERENCES

1. University of California Policy on Copyright Ownership - University of California, Office of the President, August 19, 1992 - (http://www.ucop.edu/ucophome/coordrev/policy/8-19-92att.html)
2. University of California Software On-Line Permission Statement - (http://www.ucop.edu/ott/permissn.html)
3. University of California Contracts and Grants Manual - Chapter 11 Intellectual Property - (http://www.ucop.edu/raohome/cgmanual/chap11.html)
4. UC Guidelines on University-Industry Relations - (http://www.ucop.edu/ott/genresources/policy_pdf/IndRelGuidelines.PDF)
5. UC Memorandum: Guidance for Faculty and Other Academic Employees on Issues Related to Intellectual Property and Consulting - (http://www.ucop.edu/ott/documents/consult.pdf)
6. UC Guidelines on Intellectual Property Issues - (http://ucop.edu/research/policies/ip.html)
7. UC Copyright Contacts- (http://www.ucop.edu/ott/faculty/crcontac.html) and (http://www.ucop.edu/ott/contacts.html)
8. Copyright Law of the United States of America and Related Laws Contained in Title 17 of the United States Code - (http://www.copyright.gov/title17/)
9. Producing Open Source Software by Karl Fogel - Chapter 9 - (http://producingoss.com/en/legal.html)
10. Open Source Definition - Open Source Initiative - (http://www.opensource.org/docs/osd)
11. Open Source License Types - Open Source Initiative - (http://www.opensource.org/licenses/alphabetical)

VII. ATTACHMENTS

1. Sample CA form

DURASPACE
Contributor License Agreements
https://wiki.duraspace.org/display/DSP/Contributor+License+Agreements



Contributor License Agreements

Created by Bill Branan, last modified on Jun 17, 2014

Contributor License Agreements

DuraSpace desires that all contributors of ideas, code, or documentation to DuraSpace projects submit a completed and signed individual Contributor License Agreement (CLA). This agreement clearly defines the terms under which intellectual property has been contributed to DuraSpace. This agreement will help us defend the project if there is a legal dispute regarding the software in the future. A signed CLA is required to be on file before an individual may commit to a DuraSpace project.

Download DuraSpace CLA:

- icla.pdf
- icla-redline.docx (redline against Apache icla, for comparison purposes only)

In the case of an organization, such as a corporation or academic institution, that has assigned employees to work on a DuraSpace project, a Corporate Contributor License Agreement (CCLA) is available. This agreement may be used to contribute intellectual property that may be assigned as part of an employment agreement. However, a CCLA does not remove the need for every developer to sign their own CLA as an individual, to cover any of their contributions that are not owned by the organization signing the CCLA.

Download DuraSpace CCLA:

- cla-corporate.pdf
- cla-corporate-redline.docx (redline against Apache ccla, for comparison purposes only)

Submit completed CLAs to: legal@duraspace.org

Projects

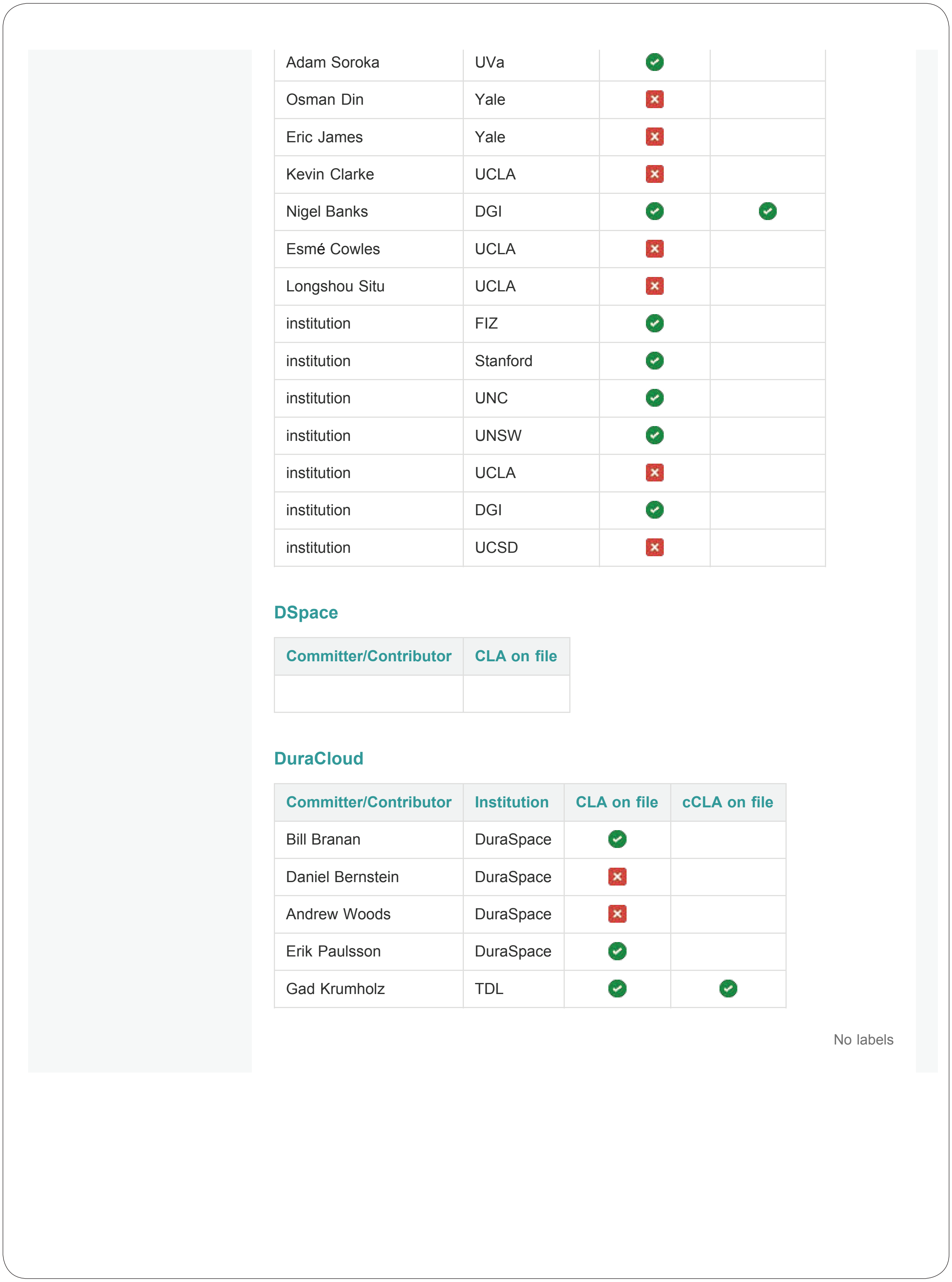
Fedora

Committer/Contributor	Institution	iCLA on file	cCLA on file
Benjamin Armintor	Columbia	✓	
Frank Asseg	FIZ/Independent	✓	✓
Chris Beer	Stanford	✓	✓
Mike Durbin	UVa	✓	
Scott Prater	U Wisc	✓	

DURASPACE

Contributor License Agreements

https://wiki.duraspace.org/display/DSP/Contributor+License+Agreements



Adam Soroka	UVa	✓	
Osman Din	Yale	✗	
Eric James	Yale	✗	
Kevin Clarke	UCLA	✗	
Nigel Banks	DGI	✓	✓
Esmé Cowles	UCLA	✗	
Longshou Situ	UCLA	✗	
institution	FIZ	✓	
institution	Stanford	✓	
institution	UNC	✓	
institution	UNSW	✓	
institution	UCLA	✗	
institution	DGI	✓	
institution	UCSD	✗	

DSpace

Committer/Contributor	CLA on file

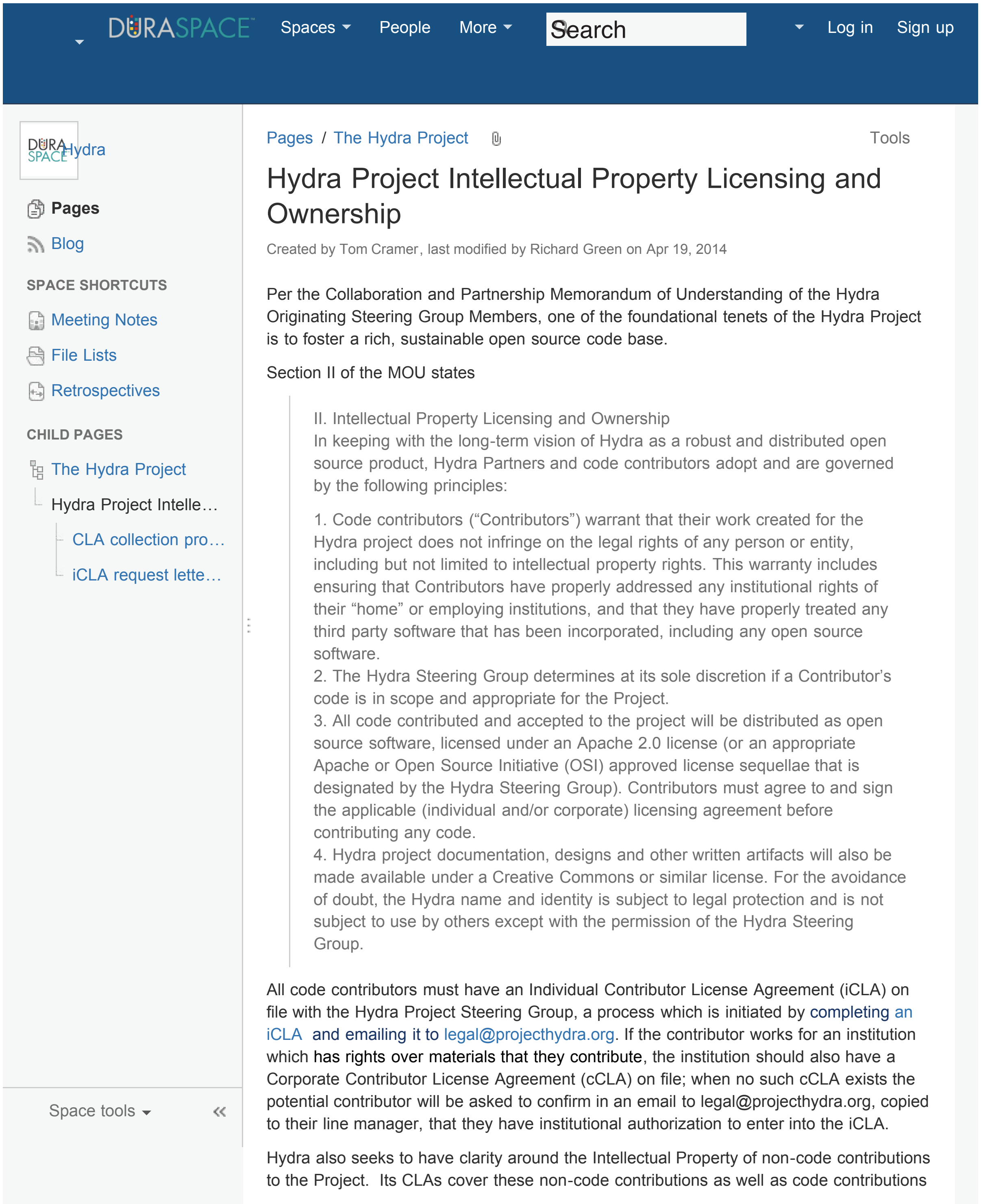
DuraCloud

Committer/Contributor	Institution	CLA on file	cCLA on file
Bill Branan	DuraSpace	✓	
Daniel Bernstein	DuraSpace	✗	
Andrew Woods	DuraSpace	✗	
Erik Paulsson	DuraSpace	✓	
Gad Krumholz	TDL	✓	✓

DURASPACE

Hydra Project Intellectual Property Licensing and Ownership

https://wiki.duraspace.org/display/hydra/Hydra+Project+Intellectual+Property+Licensing+and+Ownership



Hydra Project Intellectual Property Licensing and Ownership

Created by Tom Cramer, last modified by Richard Green on Apr 19, 2014

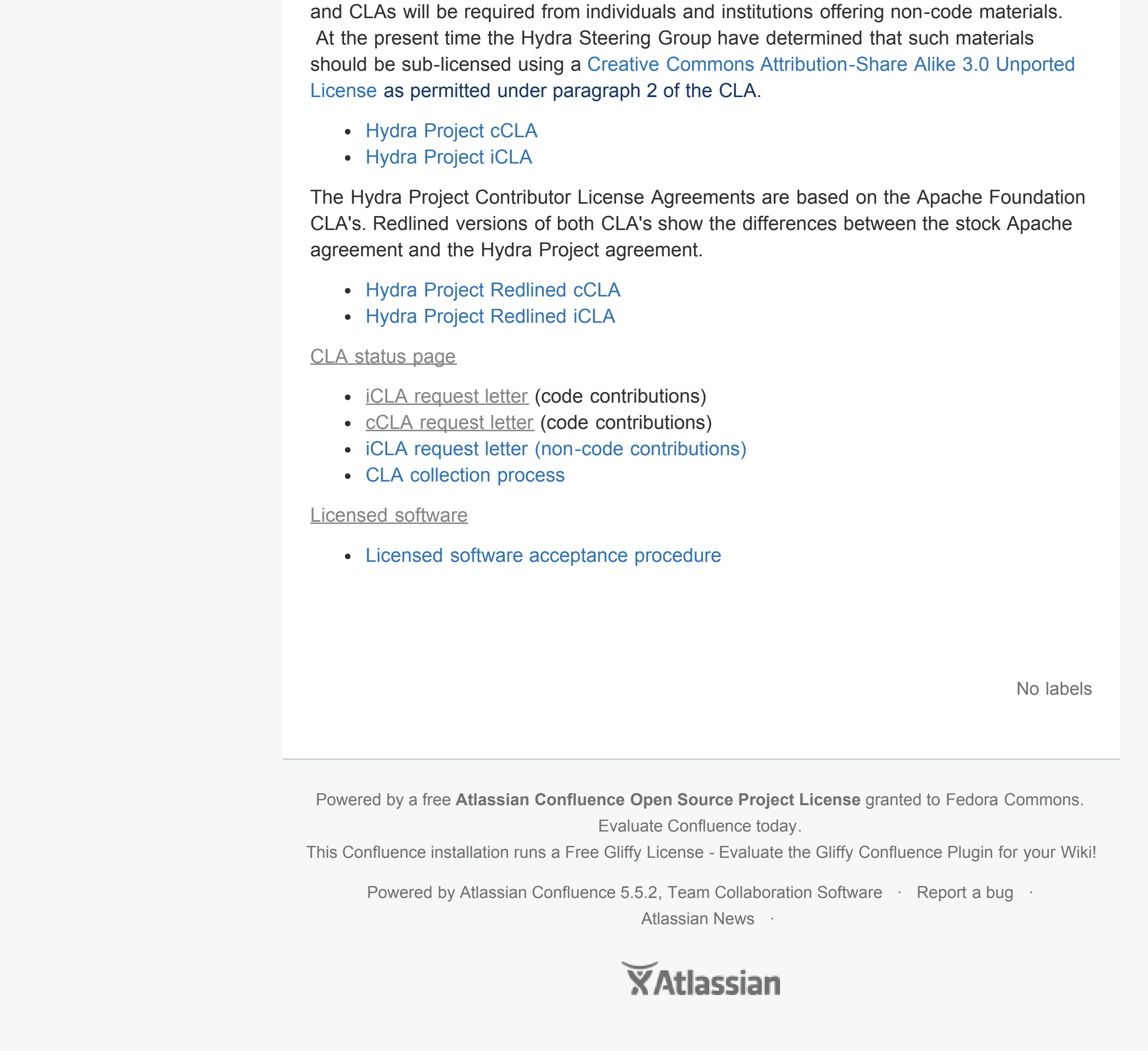
Per the Collaboration and Partnership Memorandum of Understanding of the Hydra Originating Steering Group Members, one of the foundational tenets of the Hydra Project is to foster a rich, sustainable open source code base.

Section II of the MOU states

> II. Intellectual Property Licensing and Ownership
> In keeping with the long-term vision of Hydra as a robust and distributed open source product, Hydra Partners and code contributors adopt and are governed by the following principles:
>
> 1. Code contributors (“Contributors”) warrant that their work created for the Hydra project does not infringe on the legal rights of any person or entity, including but not limited to intellectual property rights. This warranty includes ensuring that Contributors have properly addressed any institutional rights of their “home” or employing institutions, and that they have properly treated any third party software that has been incorporated, including any open source software.
> 2. The Hydra Steering Group determines at its sole discretion if a Contributor’s code is in scope and appropriate for the Project.
> 3. All code contributed and accepted to the project will be distributed as open source software, licensed under an Apache 2.0 license (or an appropriate Apache or Open Source Initiative (OSI) approved license sequellae that is designated by the Hydra Steering Group). Contributors must agree to and sign the applicable (individual and/or corporate) licensing agreement before contributing any code.
> 4. Hydra project documentation, designs and other written artifacts will also be made available under a Creative Commons or similar license. For the avoidance of doubt, the Hydra name and identity is subject to legal protection and is not subject to use by others except with the permission of the Hydra Steering Group.

All code contributors must have an Individual Contributor License Agreement (iCLA) on file with the Hydra Project Steering Group, a process which is initiated by completing an iCLA and emailing it to legal@projecthydra.org. If the contributor works for an institution which has rights over materials that they contribute, the institution should also have a Corporate Contributor License Agreement (cCLA) on file; when no such cCLA exists the potential contributor will be asked to confirm in an email to legal@projecthydra.org, copied to their line manager, that they have institutional authorization to enter into the iCLA.

Hydra also seeks to have clarity around the Intellectual Property of non-code contributions to the Project. Its CLAs cover these non-code contributions as well as code contributions



and CLAs will be required from individuals and institutions offering non-code materials. At the present time the Hydra Steering Group have determined that such materials should be sub-licensed using a Creative Commons Attribution-Share Alike 3.0 Unported License as permitted under paragraph 2 of the CLA.

- Hydra Project cCLA
- Hydra Project iCLA

The Hydra Project Contributor License Agreements are based on the Apache Foundation CLA's. Redlined versions of both CLA's show the differences between the stock Apache agreement and the Hydra Project agreement.

- Hydra Project Redlined cCLA
- Hydra Project Redlined iCLA

CLA status page

- iCLA request letter (code contributions)
- cCLA request letter (code contributions)
- iCLA request letter (non-code contributions)
- CLA collection process

Licensed software

- Licensed software acceptance procedure

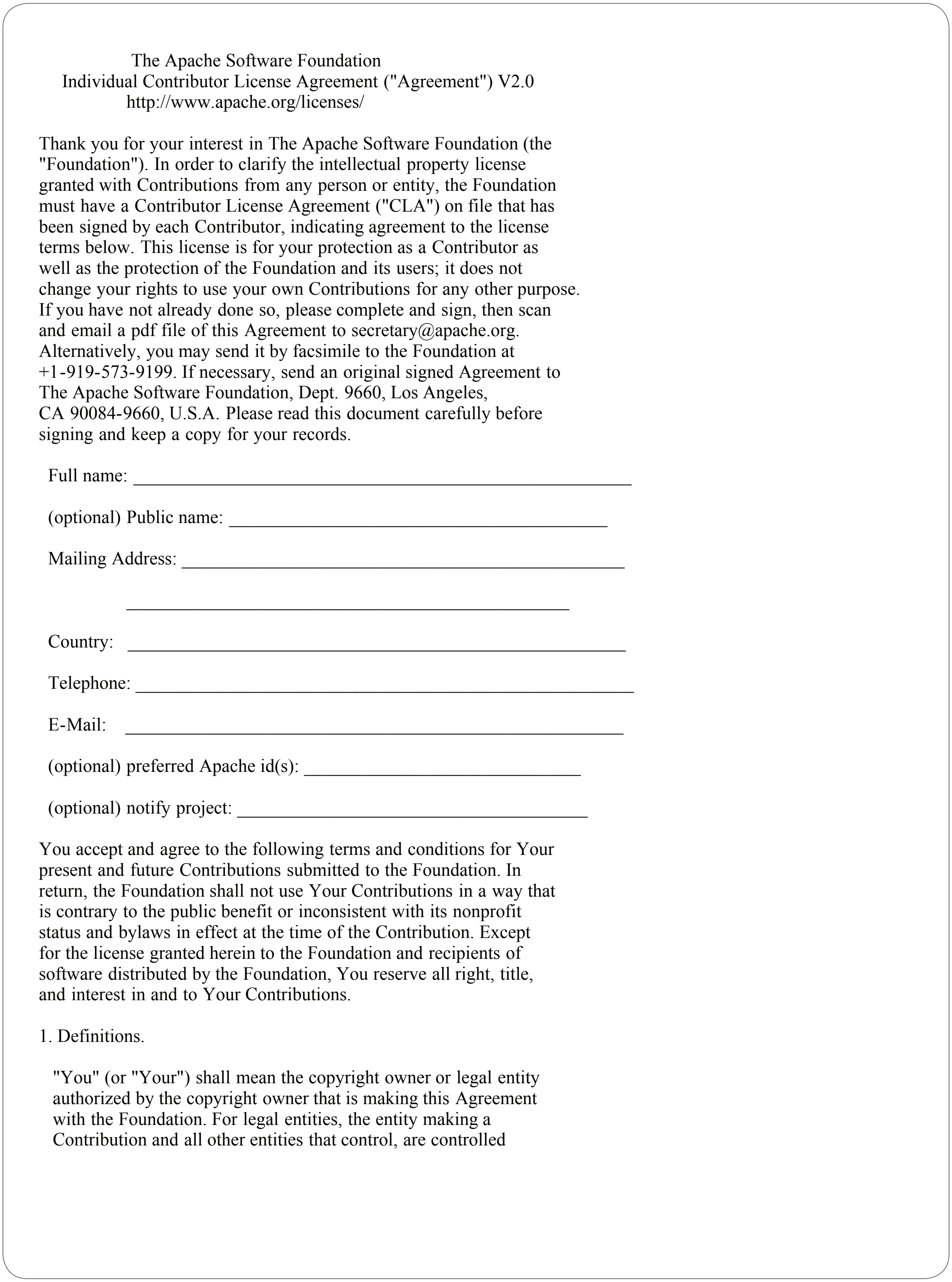
No labels

Powered by a free **Atlassian Confluence Open Source Project License** granted to Fedora Commons. Evaluate Confluence today.
This Confluence installation runs a Free Gliffy License - Evaluate the Gliffy Confluence Plugin for your Wiki!

Powered by Atlassian Confluence 5.5.2, Team Collaboration Software · Report a bug · Atlassian News ·

THE APACHE SOFTWARE FOUNDATION

Individual Contributor License Agreement (“Agreement”) V2.0
http://www.apache.org/licenses/icla.txt



The Apache Software Foundation
Individual Contributor License Agreement ("Agreement") V2.0
http://www.apache.org/licenses/

Thank you for your interest in The Apache Software Foundation (the "Foundation"). In order to clarify the intellectual property license granted with Contributions from any person or entity, the Foundation must have a Contributor License Agreement ("CLA") on file that has been signed by each Contributor, indicating agreement to the license terms below. This license is for your protection as a Contributor as well as the protection of the Foundation and its users; it does not change your rights to use your own Contributions for any other purpose. If you have not already done so, please complete and sign, then scan and email a pdf file of this Agreement to secretary@apache.org. Alternatively, you may send it by facsimile to the Foundation at +1-919-573-9199. If necessary, send an original signed Agreement to The Apache Software Foundation, Dept. 9660, Los Angeles, CA 90084-9660, U.S.A. Please read this document carefully before signing and keep a copy for your records.

Full name: ____

(optional) Public name: ____

Mailing Address: ____

____

Country: ____

Telephone: ____

E-Mail: ____

(optional) preferred Apache id(s): _______________________________

(optional) notify project: _______________________________________

You accept and agree to the following terms and conditions for Your present and future Contributions submitted to the Foundation. In return, the Foundation shall not use Your Contributions in a way that is contrary to the public benefit or inconsistent with its nonprofit status and bylaws in effect at the time of the Contribution. Except for the license granted herein to the Foundation and recipients of software distributed by the Foundation, You reserve all right, title, and interest in and to Your Contributions.

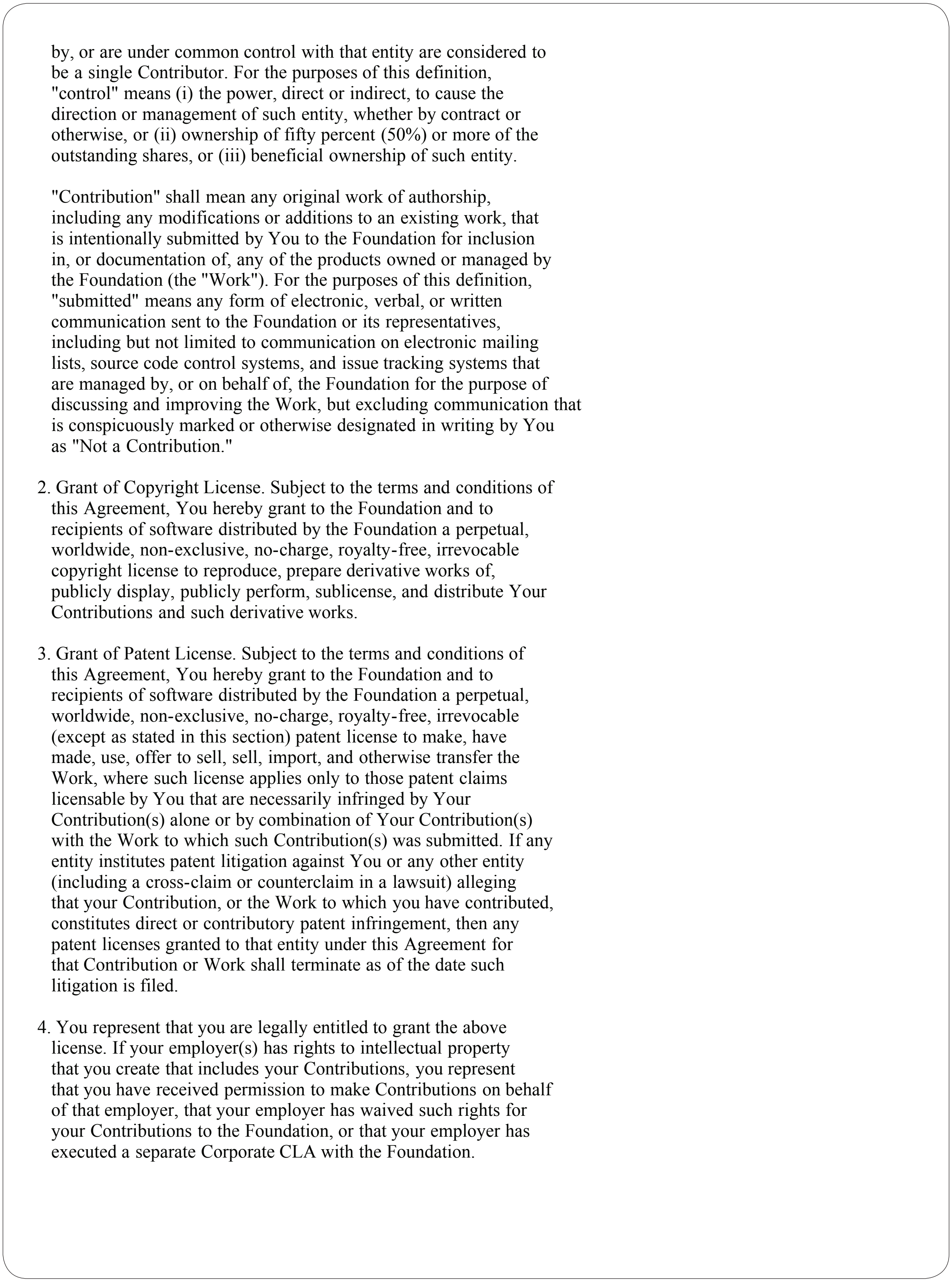
1. Definitions.

"You" (or "Your") shall mean the copyright owner or legal entity authorized by the copyright owner that is making this Agreement with the Foundation. For legal entities, the entity making a Contribution and all other entities that control, are controlled

THE APACHE SOFTWARE FOUNDATION

Individual Contributor License Agreement (“Agreement”) V2.0

http://www.apache.org/licenses/icla.txt



by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"Contribution" shall mean any original work of authorship, including any modifications or additions to an existing work, that is intentionally submitted by You to the Foundation for inclusion in, or documentation of, any of the products owned or managed by the Foundation (the "Work"). For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Foundation or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Foundation for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by You as "Not a Contribution."

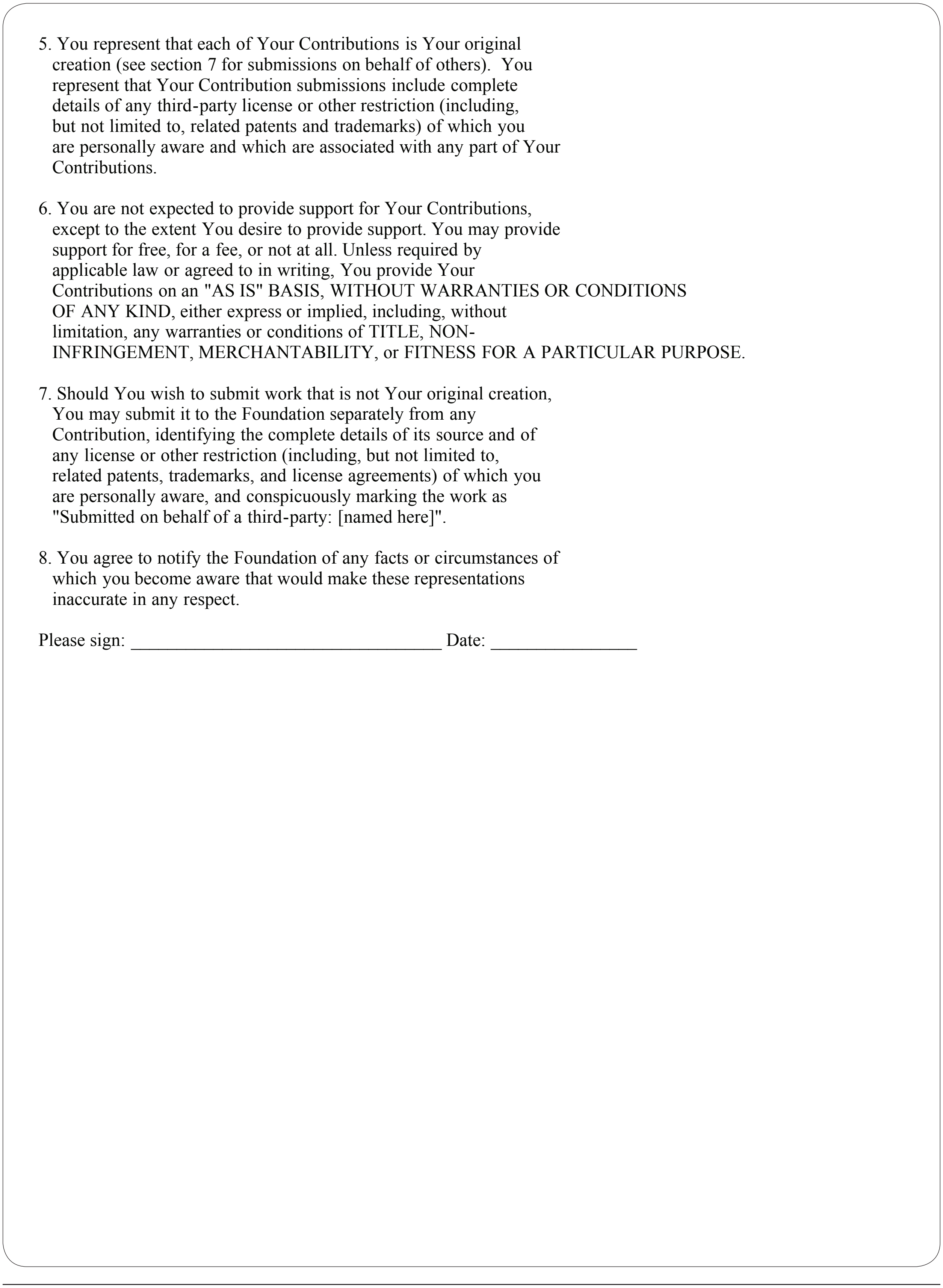
2. Grant of Copyright License. Subject to the terms and conditions of this Agreement, You hereby grant to the Foundation and to recipients of software distributed by the Foundation a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works.

3. Grant of Patent License. Subject to the terms and conditions of this Agreement, You hereby grant to the Foundation and to recipients of software distributed by the Foundation a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by You that are necessarily infringed by Your Contribution(s) alone or by combination of Your Contribution(s) with the Work to which such Contribution(s) was submitted. If any entity institutes patent litigation against You or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that your Contribution, or the Work to which you have contributed, constitutes direct or contributory patent infringement, then any patent licenses granted to that entity under this Agreement for that Contribution or Work shall terminate as of the date such litigation is filed.

4. You represent that you are legally entitled to grant the above license. If your employer(s) has rights to intellectual property that you create that includes your Contributions, you represent that you have received permission to make Contributions on behalf of that employer, that your employer has waived such rights for your Contributions to the Foundation, or that your employer has executed a separate Corporate CLA with the Foundation.

THE APACHE SOFTWARE FOUNDATION

Individual Contributor License Agreement (“Agreement”) V2.0
http://www.apache.org/licenses/icla.txt



5. You represent that each of Your Contributions is Your original creation (see section 7 for submissions on behalf of others). You represent that Your Contribution submissions include complete details of any third-party license or other restriction (including, but not limited to, related patents and trademarks) of which you are personally aware and which are associated with any part of Your Contributions.

6. You are not expected to provide support for Your Contributions, except to the extent You desire to provide support. You may provide support for free, for a fee, or not at all. Unless required by applicable law or agreed to in writing, You provide Your Contributions on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

7. Should You wish to submit work that is not Your original creation, You may submit it to the Foundation separately from any Contribution, identifying the complete details of its source and of any license or other restriction (including, but not limited to, related patents, trademarks, and license agreements) of which you are personally aware, and conspicuously marking the work as "Submitted on behalf of a third-party: [named here]".

8. You agree to notify the Foundation of any facts or circumstances of which you become aware that would make these representations inaccurate in any respect.

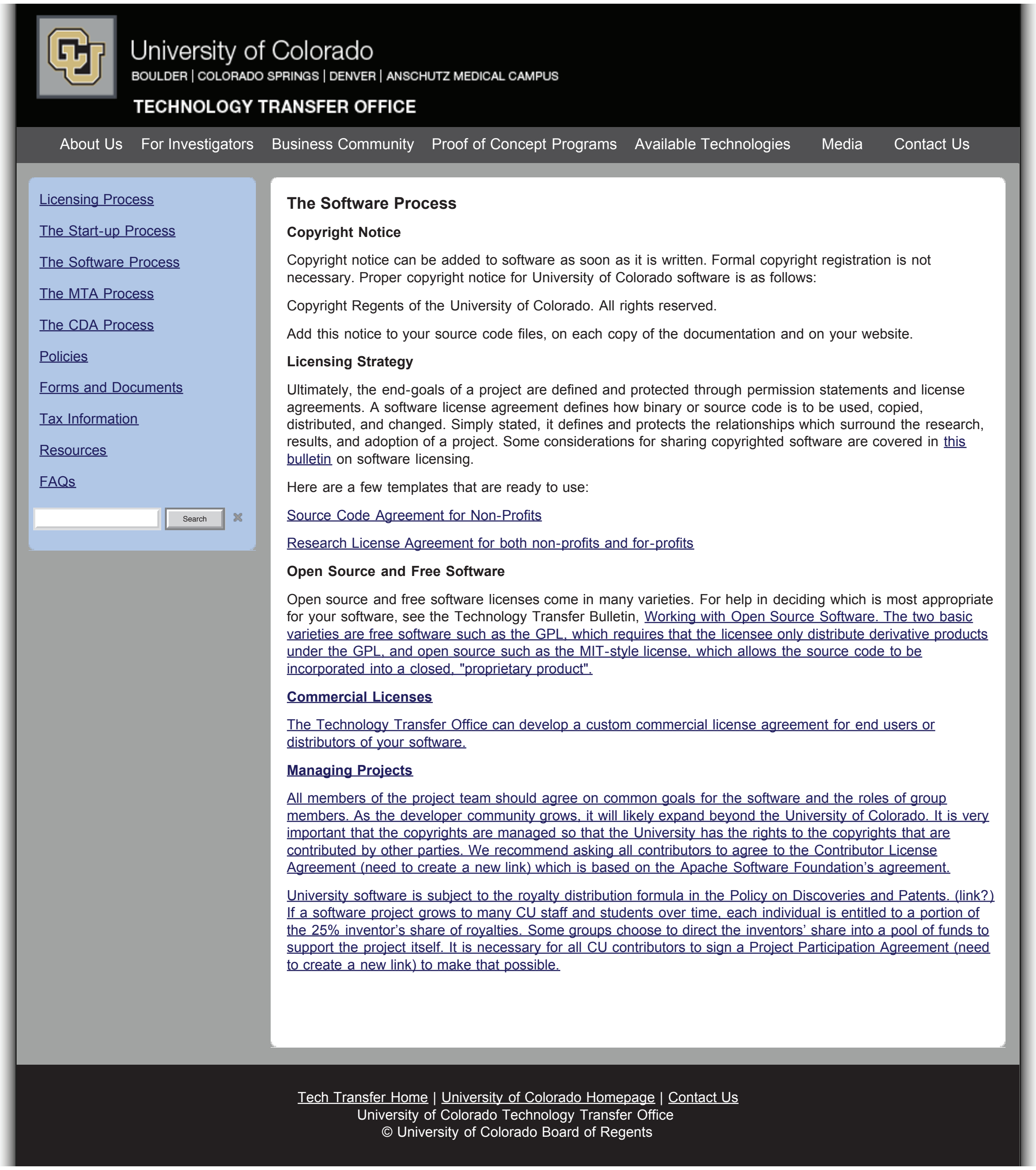
Please sign: __________________________________ Date: ________________

OSS Licenses

UNIVERSITY OF COLORADO

Technology Transfer Office. The Software Process

https://content.cu.edu/techtransfer/inv/software_process.html



The Software Process

Copyright Notice

Copyright notice can be added to software as soon as it is written. Formal copyright registration is not necessary. Proper copyright notice for University of Colorado software is as follows:

Copyright Regents of the University of Colorado. All rights reserved.

Add this notice to your source code files, on each copy of the documentation and on your website.

Licensing Strategy

Ultimately, the end-goals of a project are defined and protected through permission statements and license agreements. A software license agreement defines how binary or source code is to be used, copied, distributed, and changed. Simply stated, it defines and protects the relationships which surround the research, results, and adoption of a project. Some considerations for sharing copyrighted software are covered in this bulletin on software licensing.

Here are a few templates that are ready to use:

Source Code Agreement for Non-Profits

Research License Agreement for both non-profits and for-profits

Open Source and Free Software

Open source and free software licenses come in many varieties. For help in deciding which is most appropriate for your software, see the Technology Transfer Bulletin, Working with Open Source Software. The two basic varieties are free software such as the GPL, which requires that the licensee only distribute derivative products under the GPL, and open source such as the MIT-style license, which allows the source code to be incorporated into a closed, "proprietary product".

Commercial Licenses

The Technology Transfer Office can develop a custom commercial license agreement for end users or distributors of your software.

Managing Projects

All members of the project team should agree on common goals for the software and the roles of group members. As the developer community grows, it will likely expand beyond the University of Colorado. It is very important that the copyrights are managed so that the University has the rights to the copyrights that are contributed by other parties. We recommend asking all contributors to agree to the Contributor License Agreement (need to create a new link) which is based on the Apache Software Foundation’s agreement.

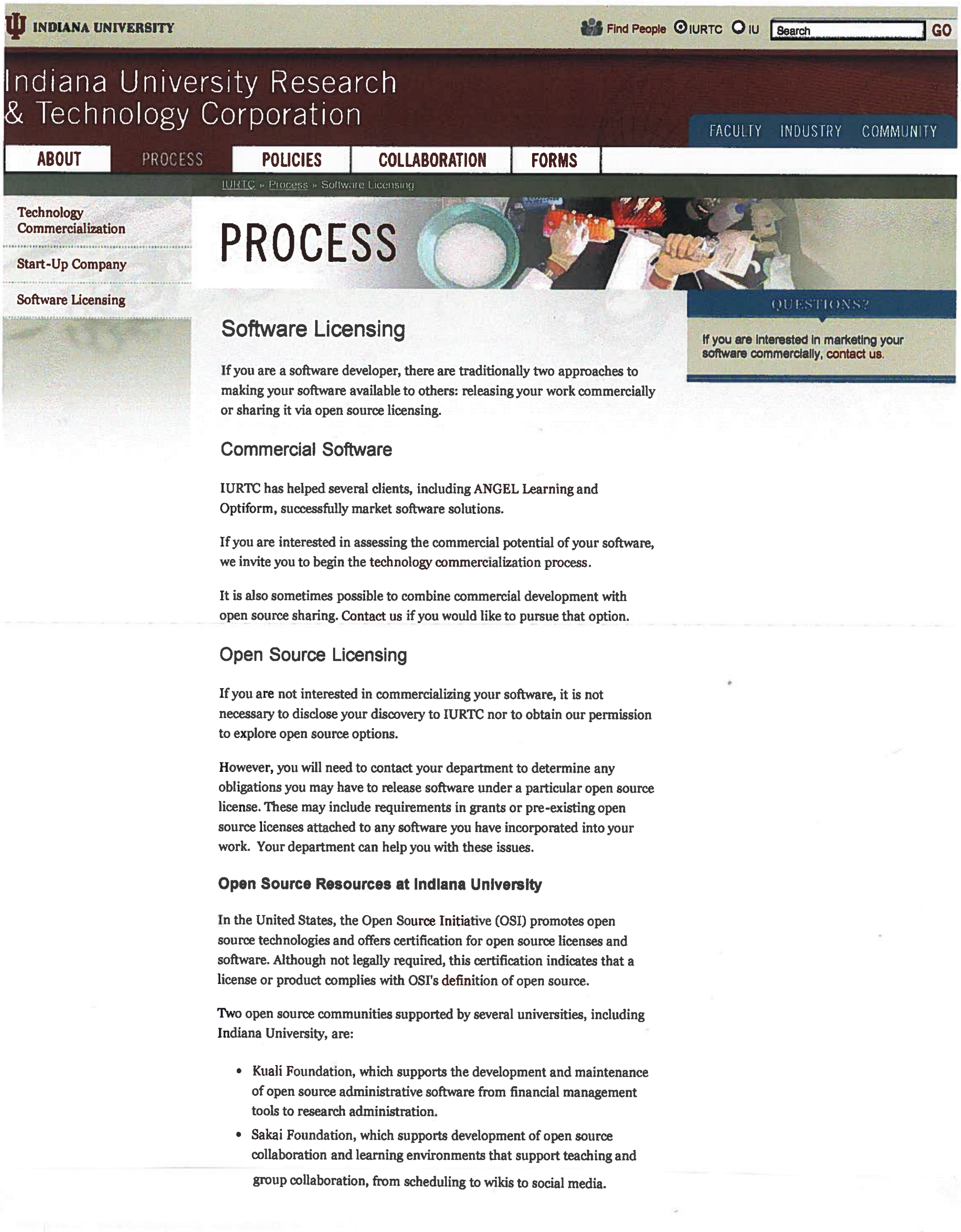
University software is subject to the royalty distribution formula in the Policy on Discoveries and Patents. (link?) If a software project grows to many CU staff and students over time, each individual is entitled to a portion of the 25% inventor’s share of royalties. Some groups choose to direct the inventors’ share into a pool of funds to support the project itself. It is necessary for all CU contributors to sign a Project Participation Agreement (need to create a new link) to make that possible.

Tech Transfer Home | University of Colorado Homepage | Contact Us
University of Colorado Technology Transfer Office
© University of Colorado Board of Regents

INDIANA UNIVERSITY

Research & Technology Corporation. Process. Software Licensing

http://iurtc.iu.edu/process/software-licensing/index.shtml



If you are interested in marketing your software commercially, contact us.

Software Licensing

If you are a software developer, there are traditionally two approaches to making your software available to others: releasing your work commercially or sharing it via open source licensing.

Commercial Software

IURTC has helped several clients, including ANGEL Learning and Optiform, successfully market software solutions.

If you are interested in assessing the commercial potential of your software, we invite you to begin the technology commercialization process.

It is also sometimes possible to combine commercial development with open source sharing. Contact us if you would like to pursue that option.

Open Source Licensing

If you are not interested in commercializing your software, it is not necessary to disclose your discovery to IURTC nor to obtain our permission to explore open source options.

However, you will need to contact your department to determine any obligations you may have to release software under a particular open source license. These may include requirements in grants or pre-existing open source licenses attached to any software you have incorporated into your work. Your department can help you with these issues.

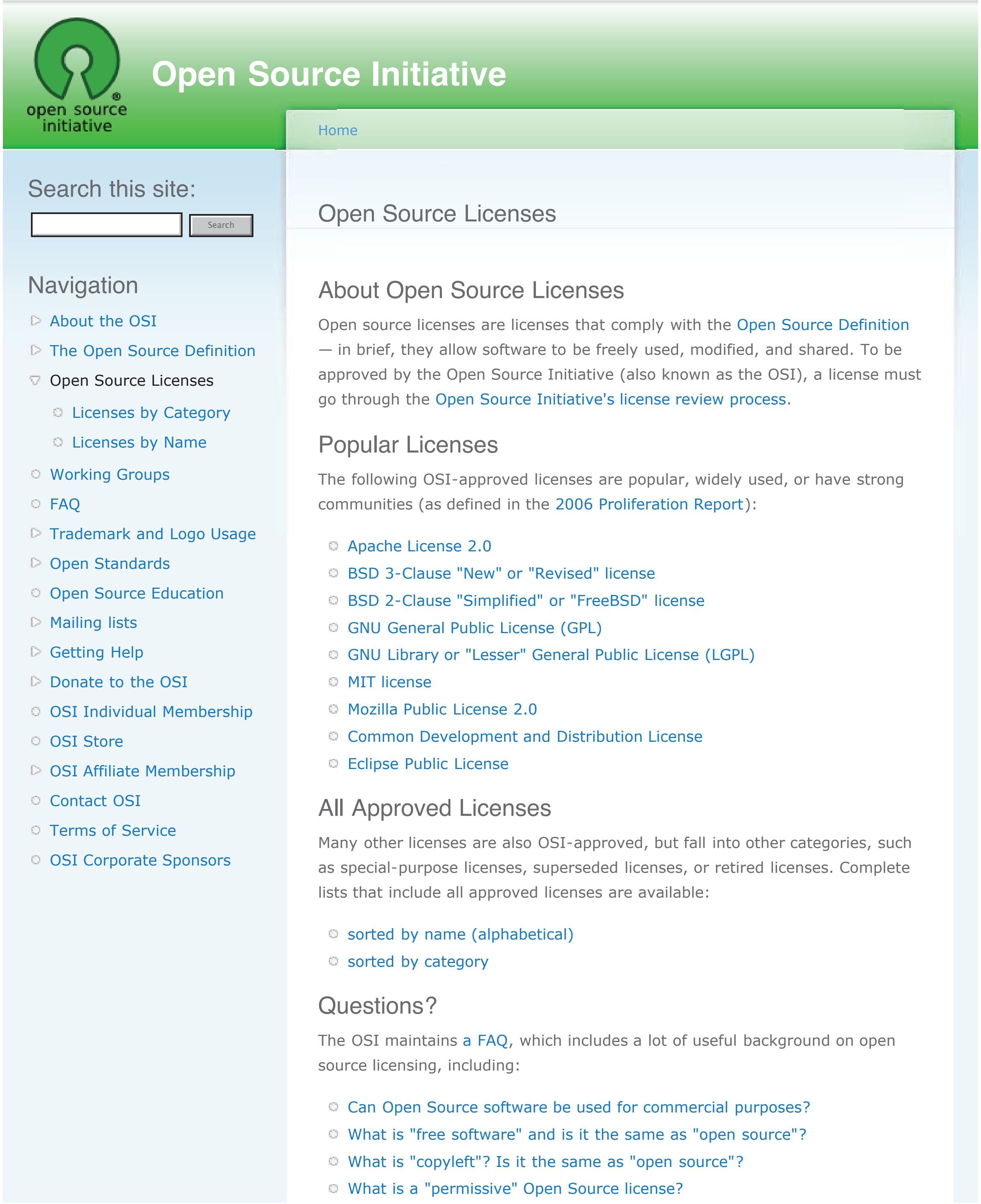
Open Source Resources at Indiana University

In the United States, the Open Source Initiative (OSI) promotes open source technologies and offers certification for open source licenses and software. Although not legally required, this certification indicates that a license or product complies with OSI's definition of open source.

Two open source communities supported by several universities, including Indiana University, are:

- Kuali Foundation, which supports the development and maintenance of open source administrative software from financial management tools to research administration.
- Sakai Foundation, which supports development of open source collaboration and learning environments that support teaching and group collaboration, from scheduling to wikis to social media.

OPEN SOURCE INITIATIVE
Open Source Licenses
http://opensource.org/licenses



Open Source Initiative

Home

Search this site:

Navigation

- About the OSI
- The Open Source Definition
- Open Source Licenses
  - Licenses by Category
  - Licenses by Name
- Working Groups
- FAQ
- Trademark and Logo Usage
- Open Standards
- Open Source Education
- Mailing lists
- Getting Help
- Donate to the OSI
- OSI Individual Membership
- OSI Store
- OSI Affiliate Membership
- Contact OSI
- Terms of Service
- OSI Corporate Sponsors

Open Source Licenses

About Open Source Licenses

Open source licenses are licenses that comply with the Open Source Definition — in brief, they allow software to be freely used, modified, and shared. To be approved by the Open Source Initiative (also known as the OSI), a license must go through the Open Source Initiative's license review process.

Popular Licenses

The following OSI-approved licenses are popular, widely used, or have strong communities (as defined in the 2006 Proliferation Report):

- Apache License 2.0
- BSD 3-Clause "New" or "Revised" license
- BSD 2-Clause "Simplified" or "FreeBSD" license
- GNU General Public License (GPL)
- GNU Library or "Lesser" General Public License (LGPL)
- MIT license
- Mozilla Public License 2.0
- Common Development and Distribution License
- Eclipse Public License

All Approved Licenses

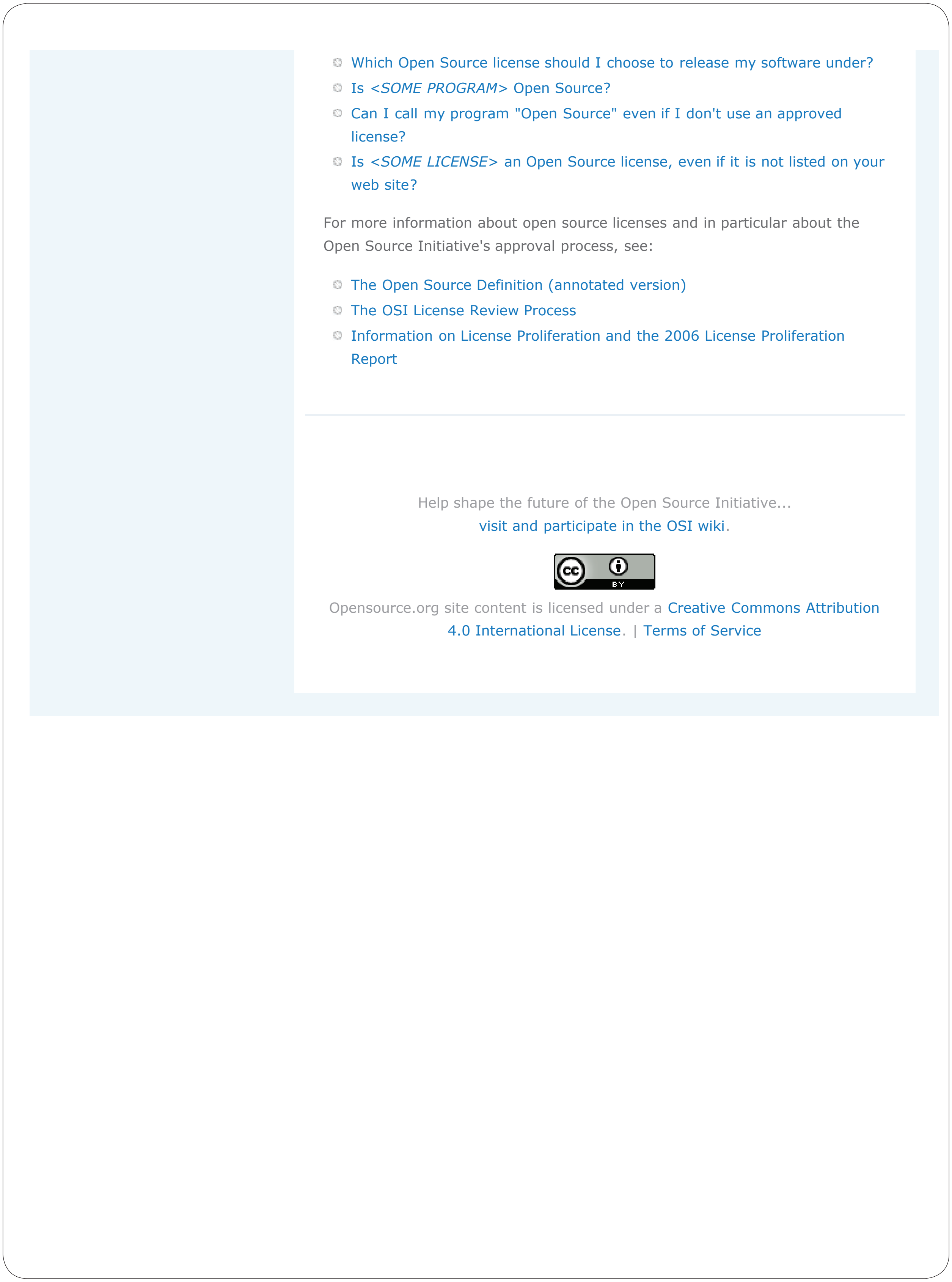
Many other licenses are also OSI-approved, but fall into other categories, such as special-purpose licenses, superseded licenses, or retired licenses. Complete lists that include all approved licenses are available:

- sorted by name (alphabetical)
- sorted by category

Questions?

The OSI maintains a FAQ, which includes a lot of useful background on open source licensing, including:

- Can Open Source software be used for commercial purposes?
- What is "free software" and is it the same as "open source"?
- What is "copyleft"? Is it the same as "open source"?
- What is a "permissive" Open Source license?



- Which Open Source license should I choose to release my software under?
- Is *<SOME PROGRAM>* Open Source?
- Can I call my program "Open Source" even if I don't use an approved license?
- Is *<SOME LICENSE>* an Open Source license, even if it is not listed on your web site?

For more information about open source licenses and in particular about the Open Source Initiative's approval process, see:

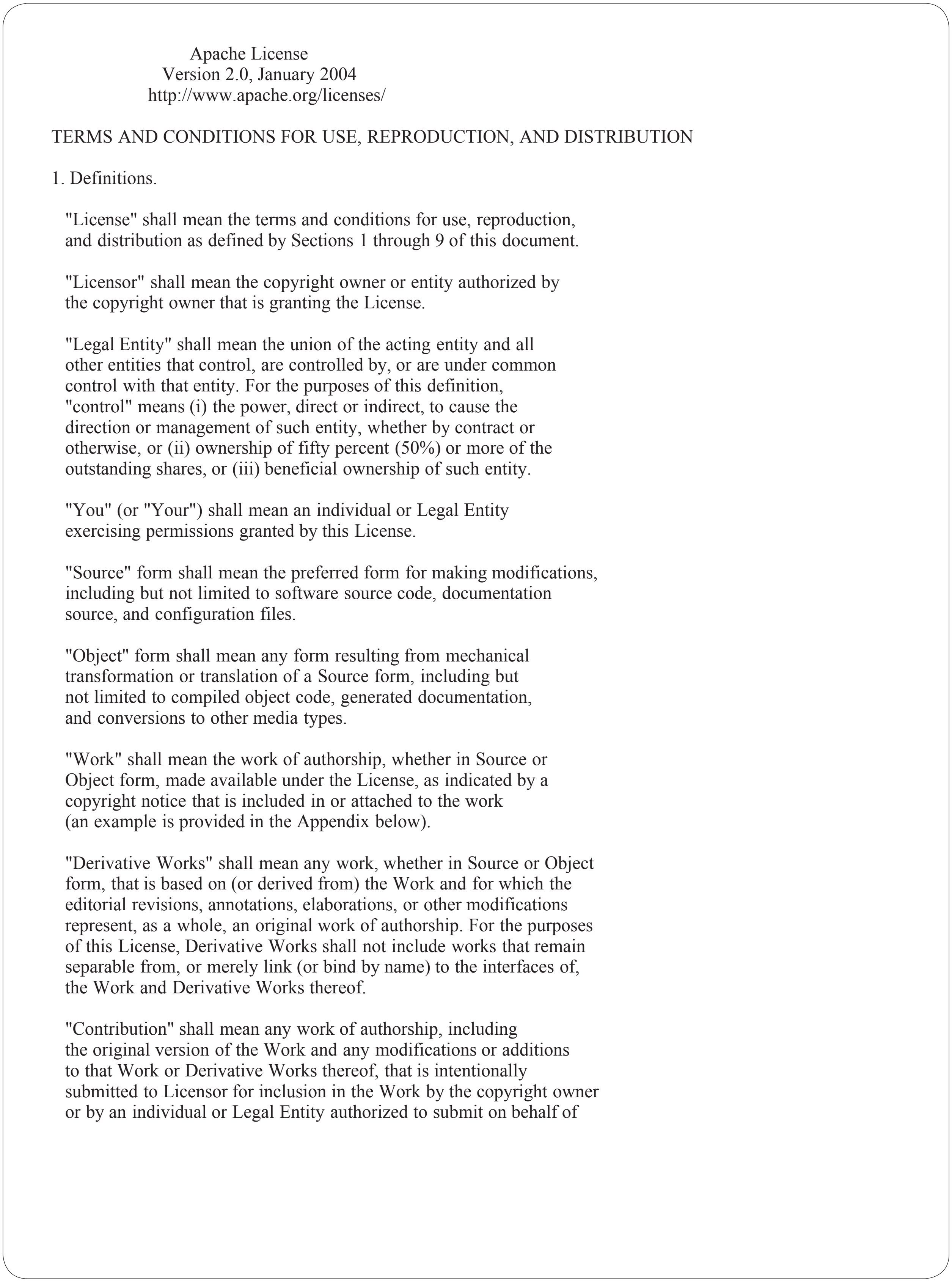
- The Open Source Definition (annotated version)
- The OSI License Review Process
- Information on License Proliferation and the 2006 License Proliferation Report

Help shape the future of the Open Source Initiative...
visit and participate in the OSI wiki.

THE APACHE SOFTWARE FOUNDATION

Apache License Version 2

http://www.apache.org/licenses/LICENSE-2.0



Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

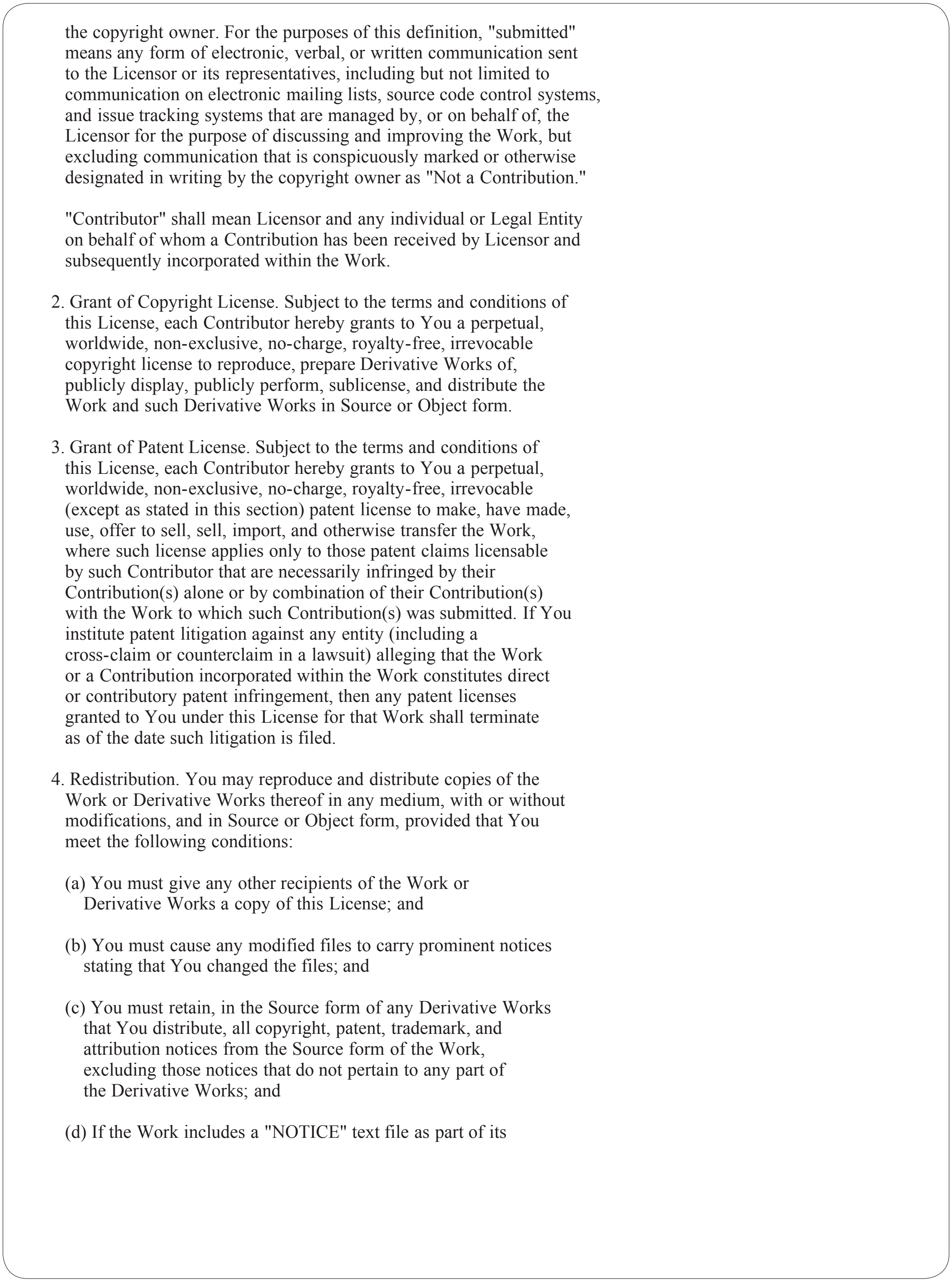
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of

THE APACHE SOFTWARE FOUNDATION

Apache License Version 2

http://www.apache.org/licenses/LICENSE-2.0



the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

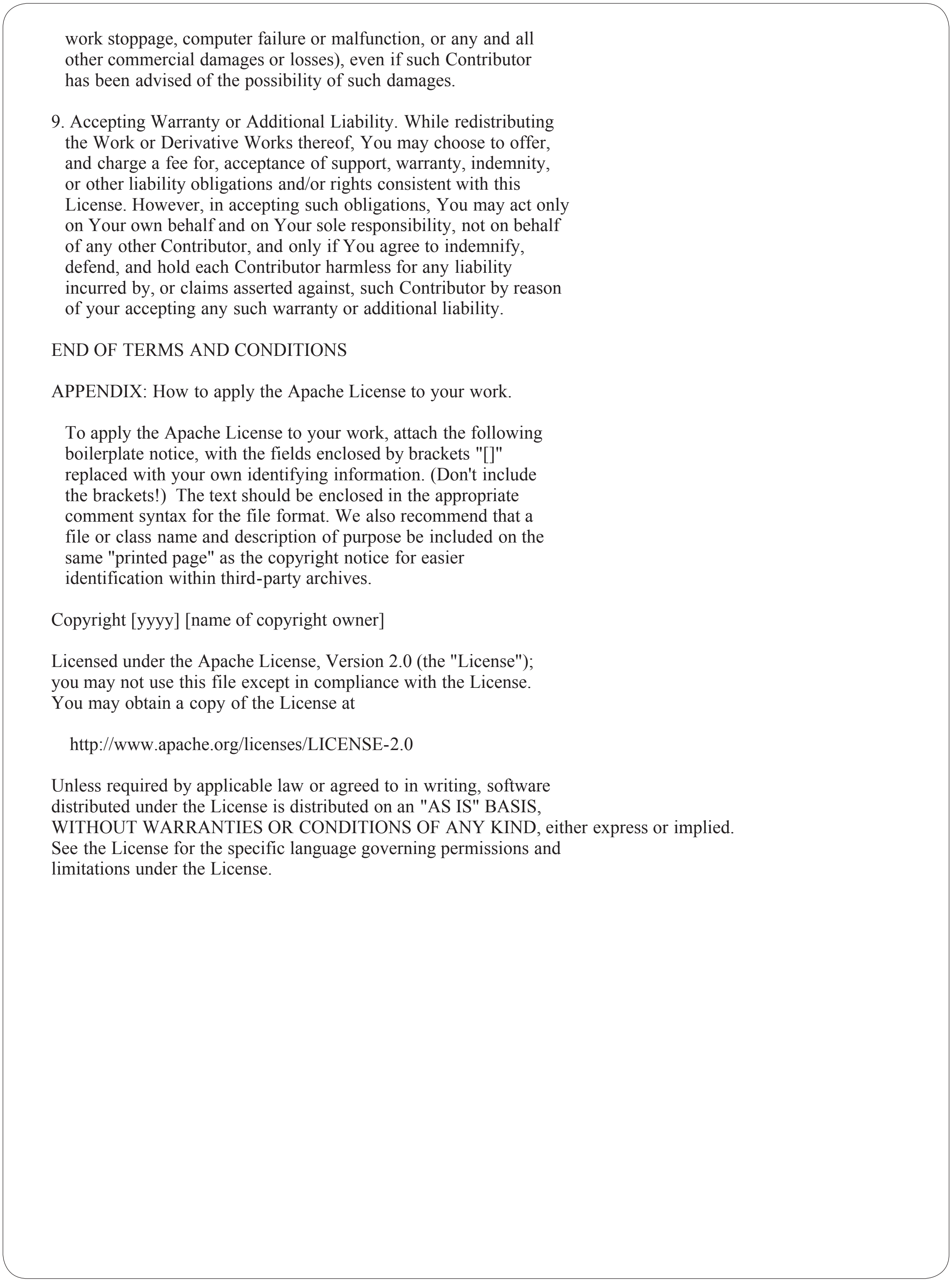
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill,



work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

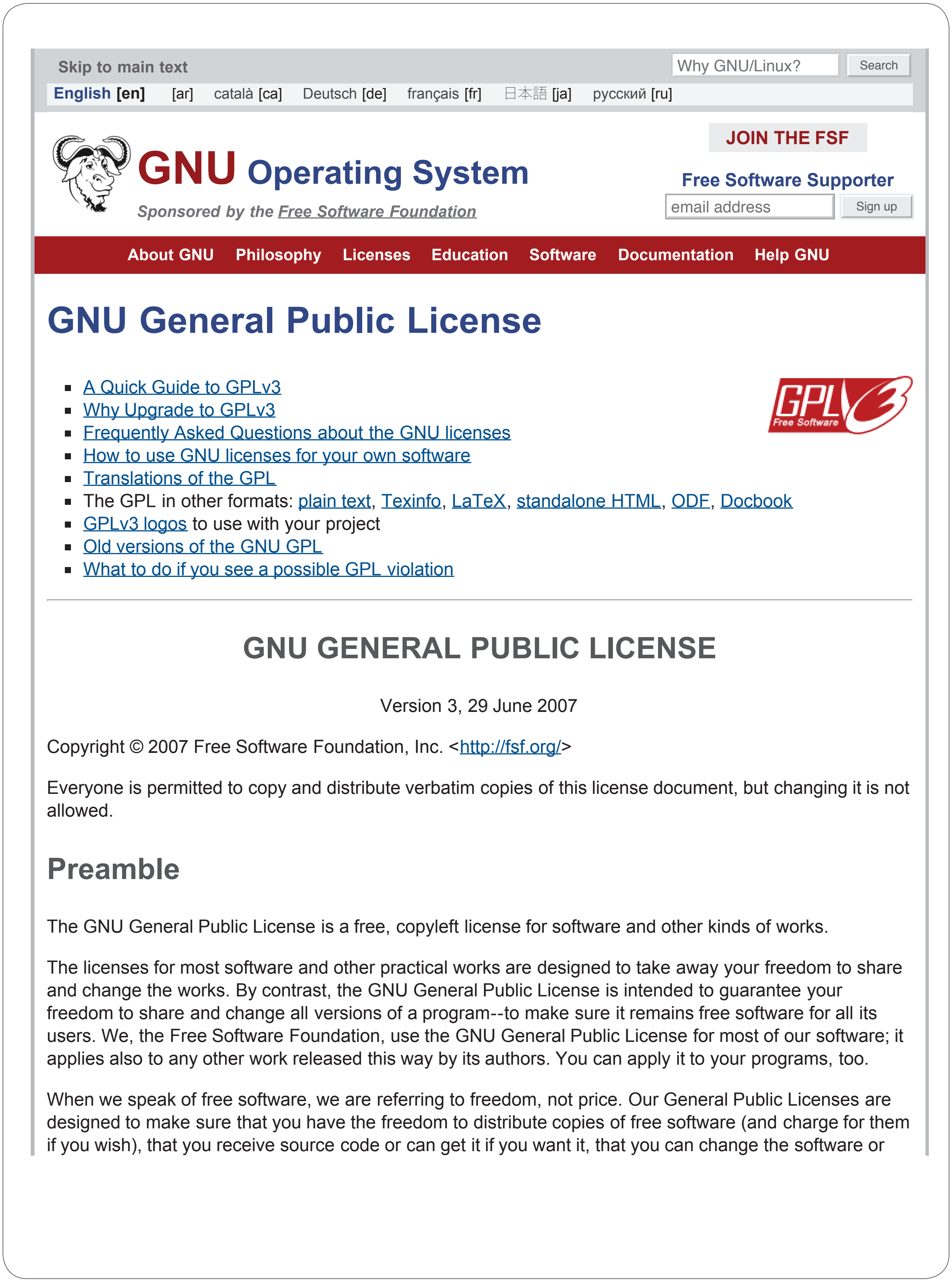
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

FREE SOFTWARE FOUNDATION
GNU General Public License Version 3
http://www.gnu.org/licenses/gpl.html



GNU General Public License

- A Quick Guide to GPLv3
- Why Upgrade to GPLv3
- Frequently Asked Questions about the GNU licenses
- How to use GNU licenses for your own software
- Translations of the GPL
- The GPL in other formats: plain text, Texinfo, LaTeX, standalone HTML, ODF, Docbook
- GPLv3 logos to use with your project
- Old versions of the GNU GPL
- What to do if you see a possible GPL violation

GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

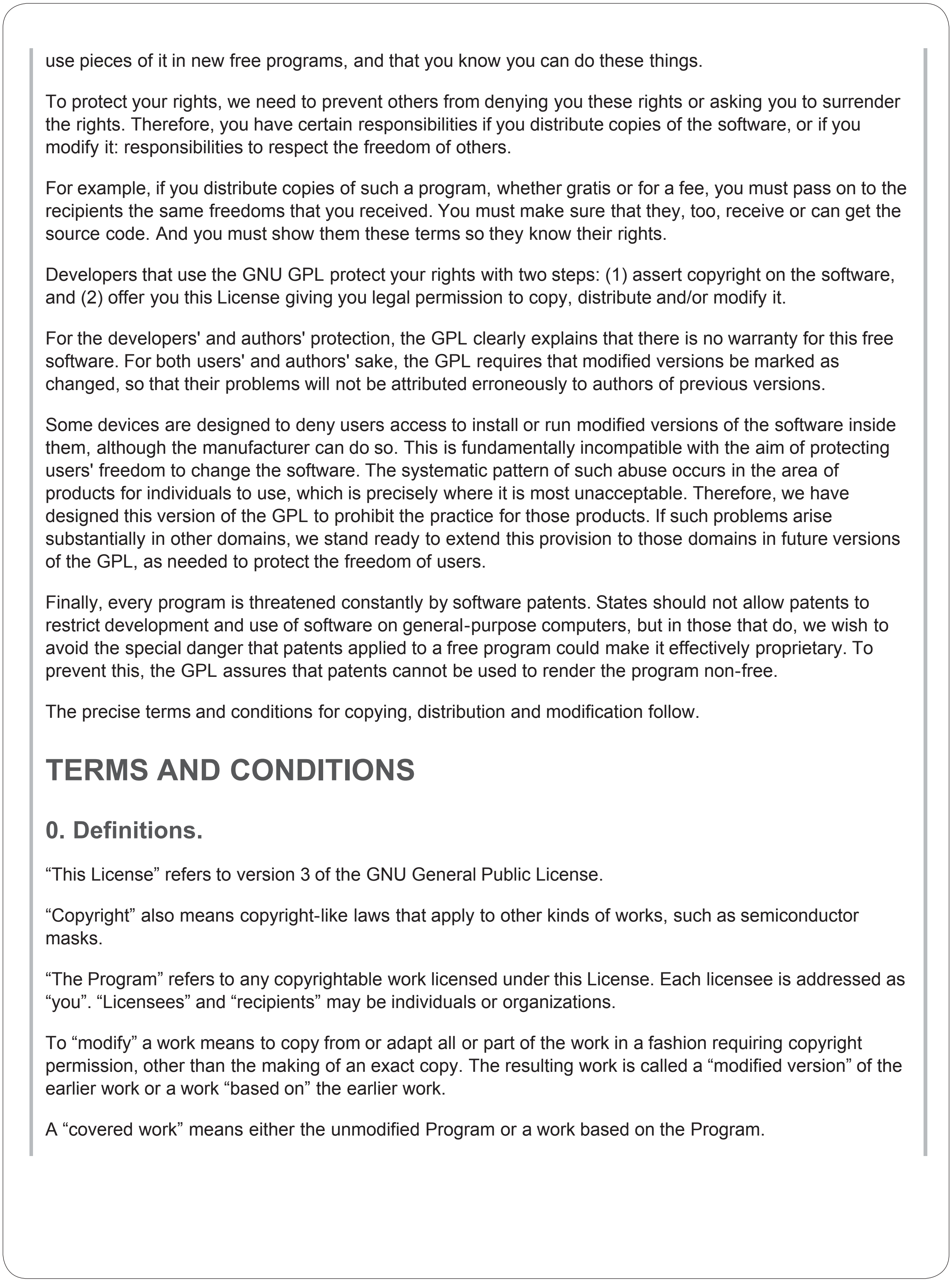
The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or

FREE SOFTWARE FOUNDATION

GNU General Public License Version 3
http://www.gnu.org/licenses/gpl.html



use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

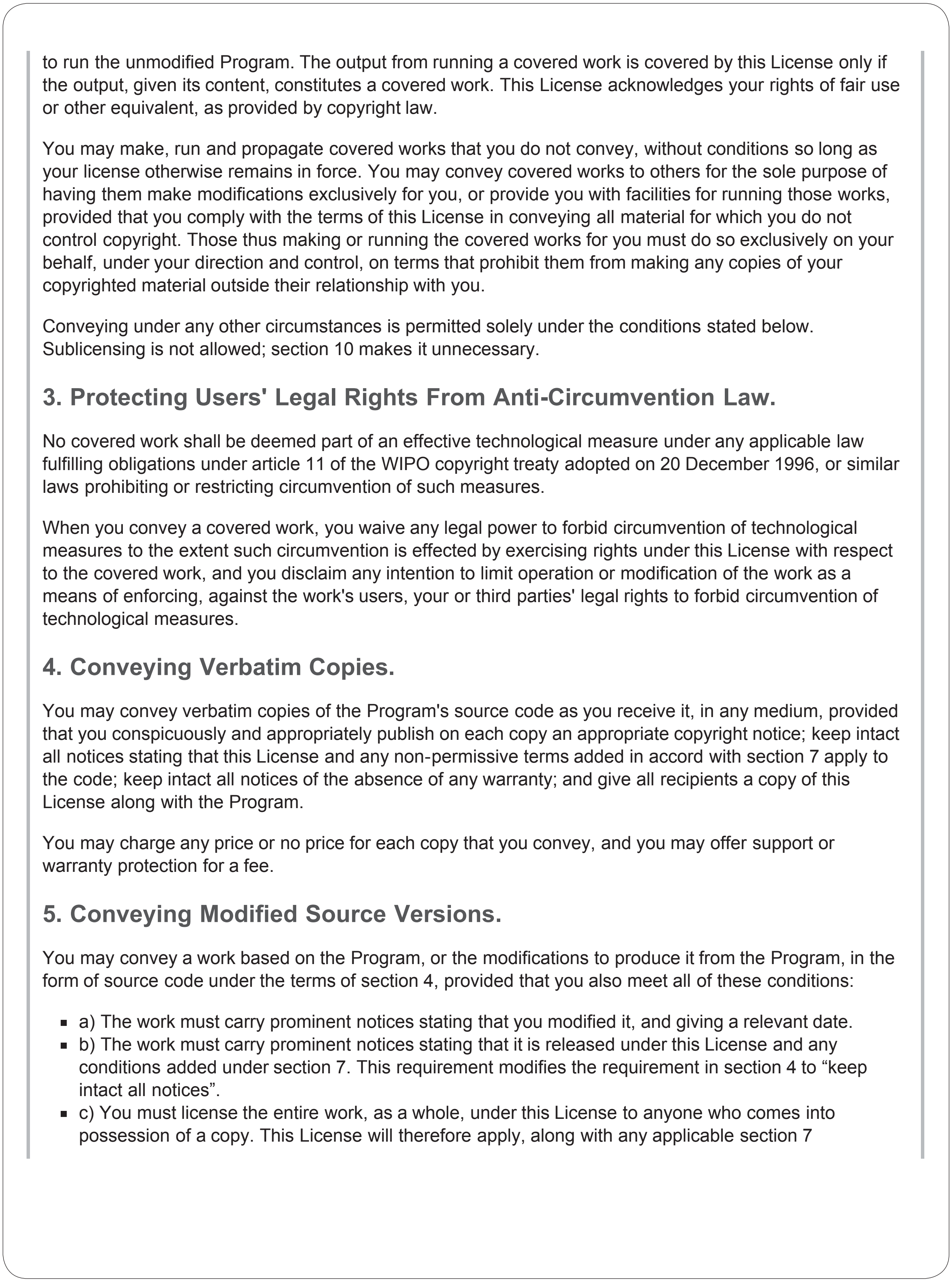
The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission



to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7

additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

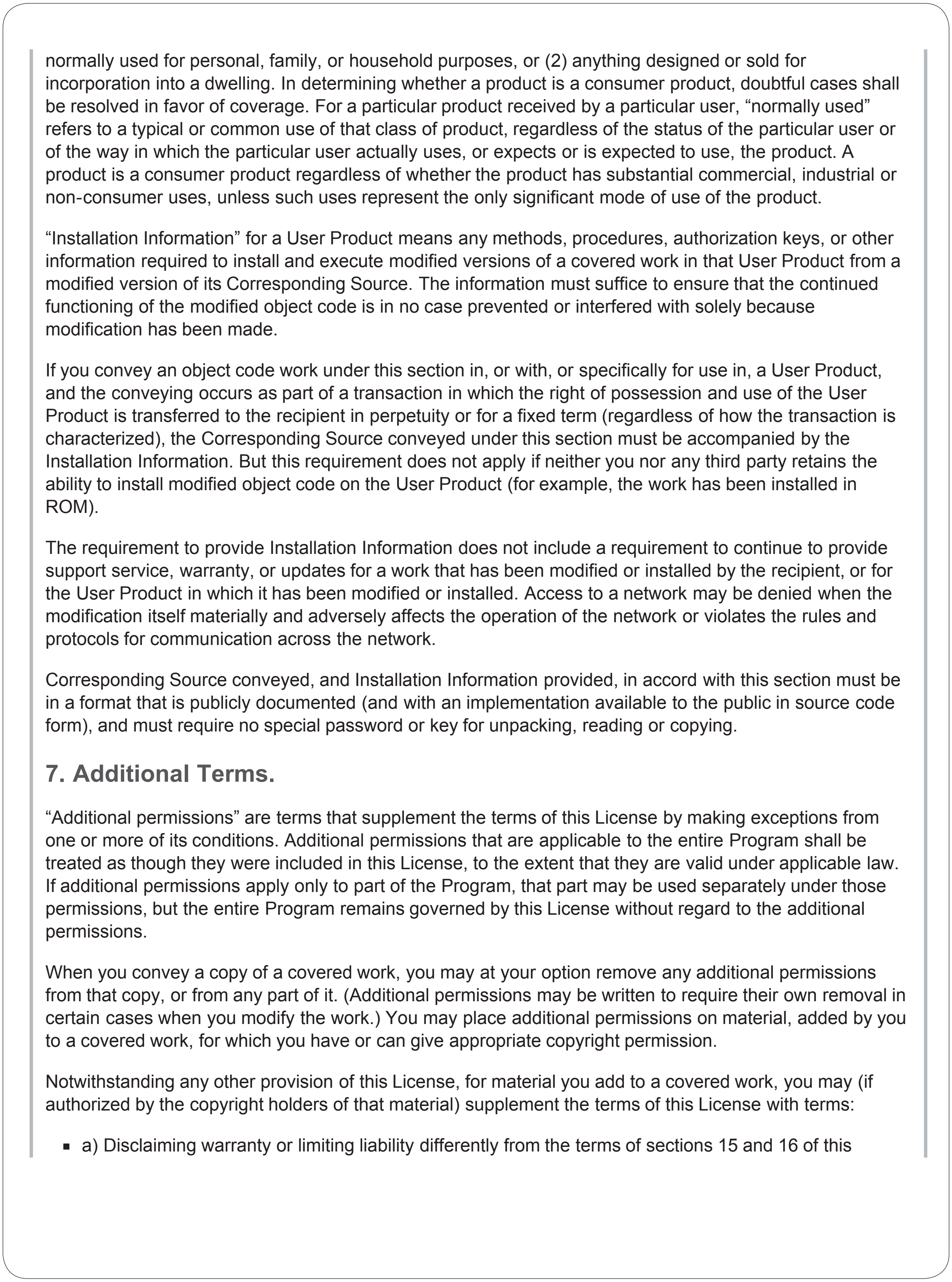
A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is

FREE SOFTWARE FOUNDATION

GNU General Public License Version 3

http://www.gnu.org/licenses/gpl.html



normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this

License; or

- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

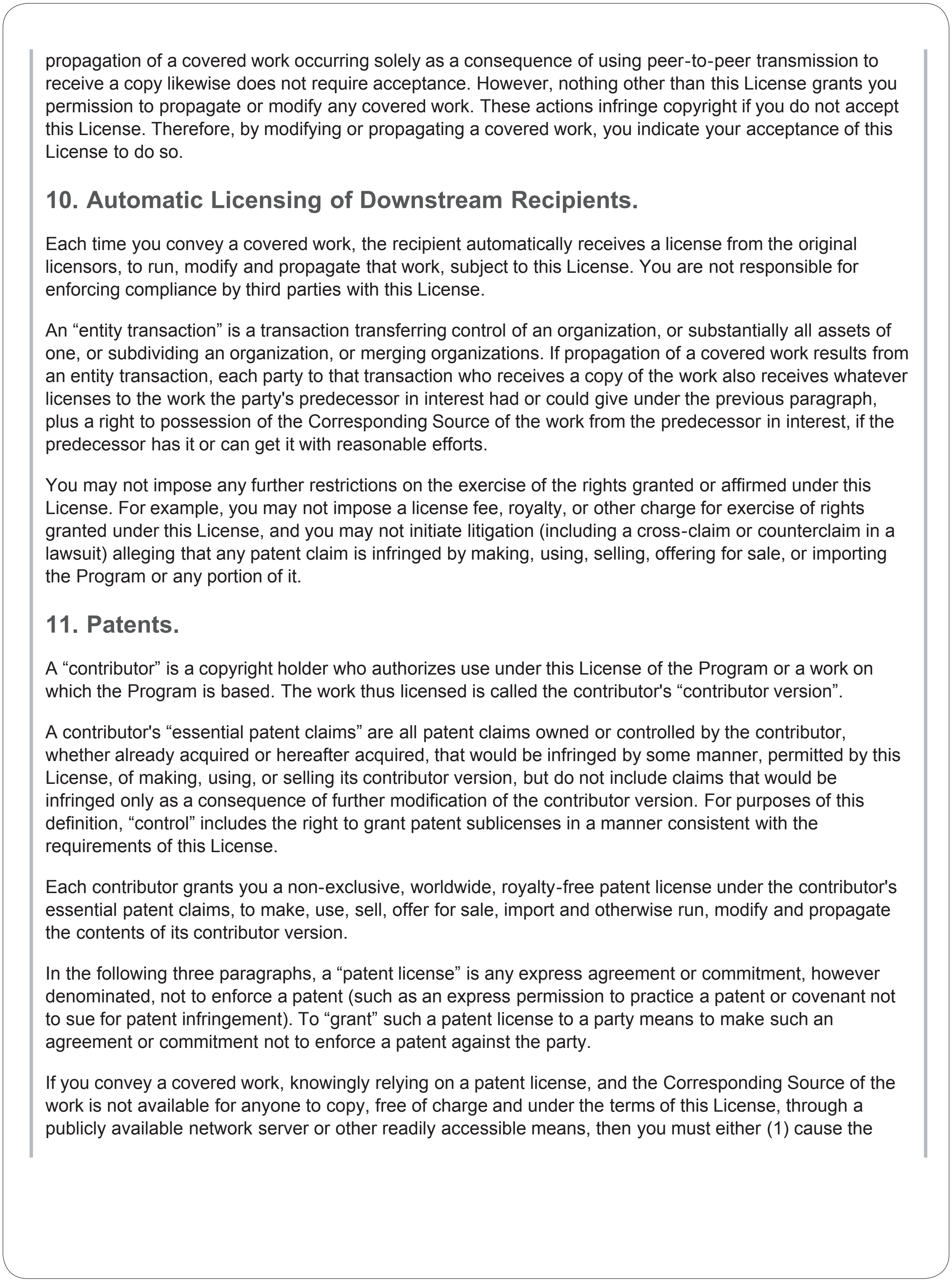
However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary



propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the

Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

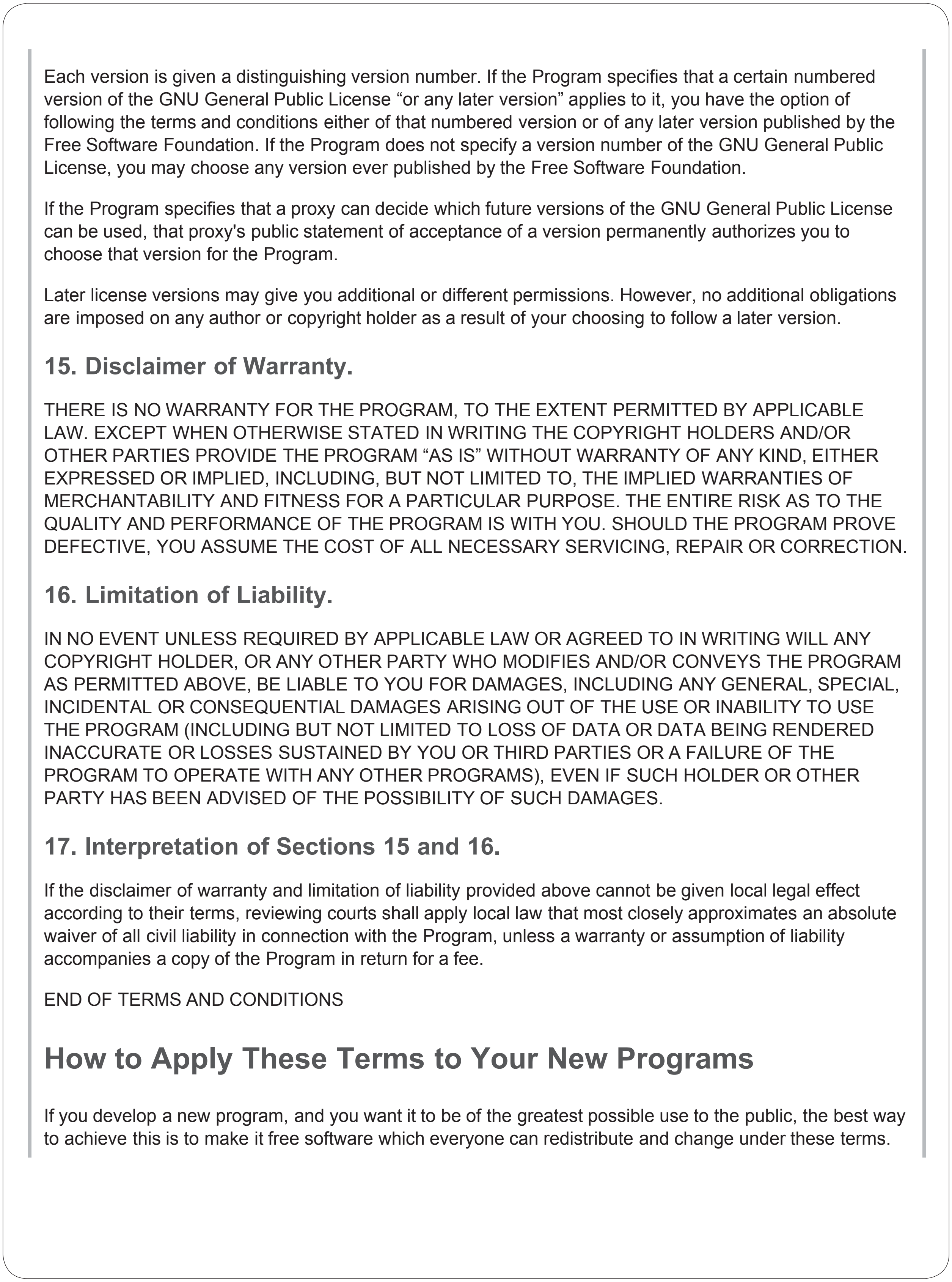
If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.



Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

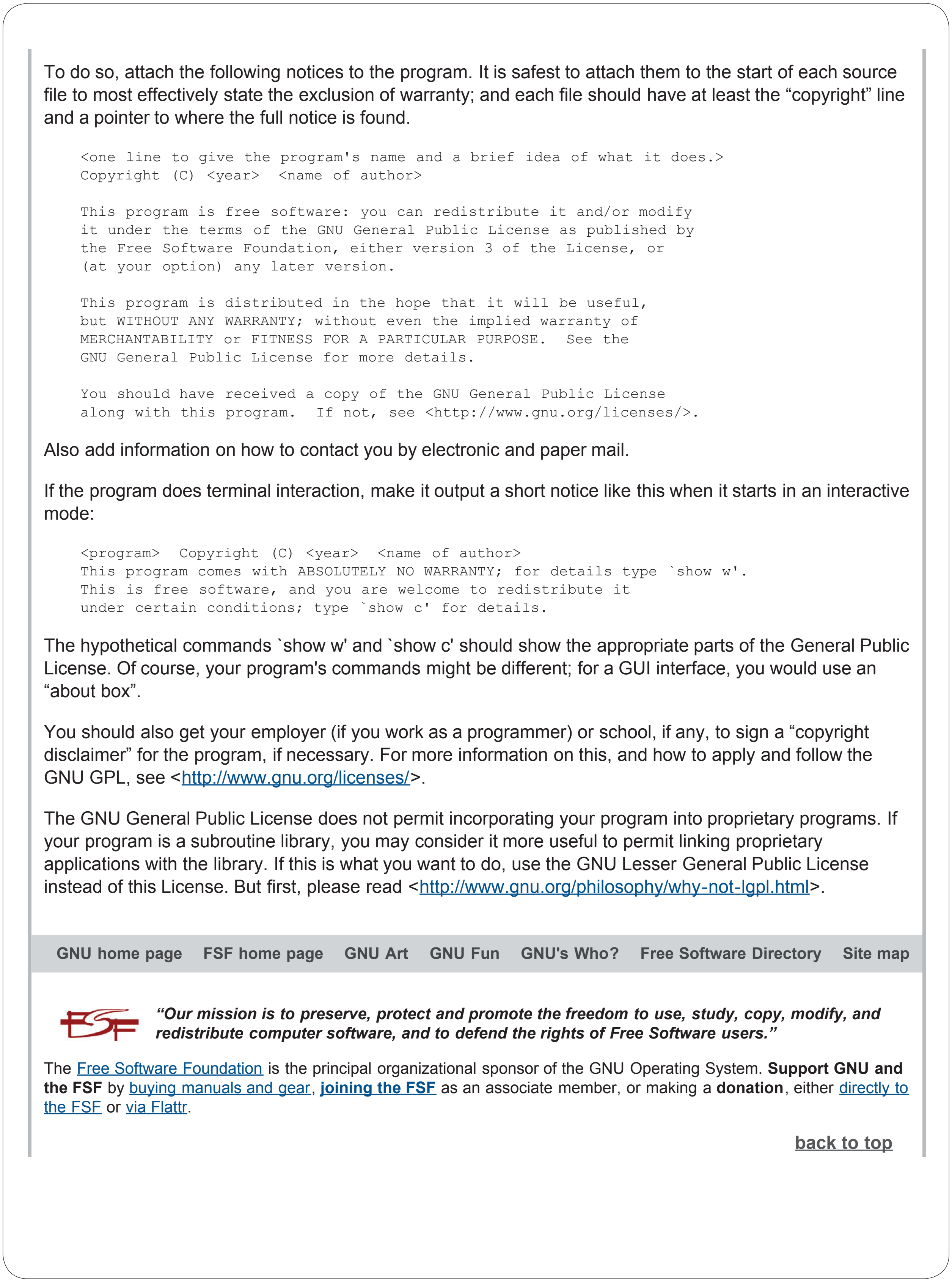
17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.



To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

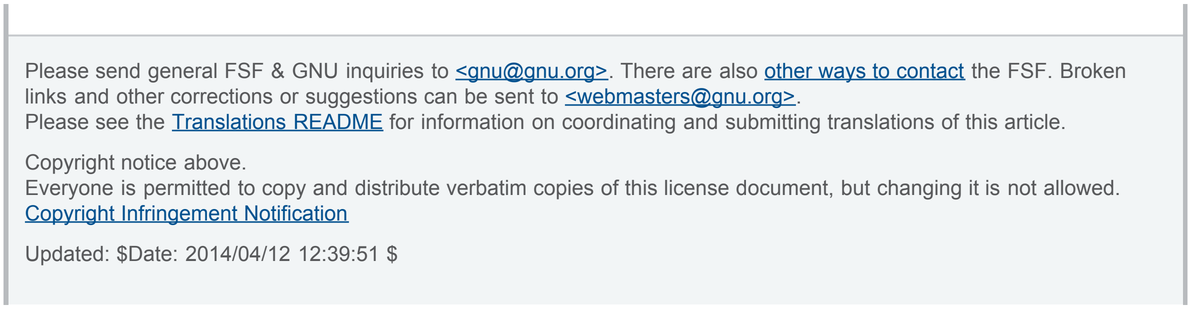
The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

GNU home page   **FSF home page**   **GNU Art**   **GNU Fun**   **GNU's Who?**   **Free Software Directory**   **Site map**

“Our mission is to preserve, protect and promote the freedom to use, study, copy, modify, and redistribute computer software, and to defend the rights of Free Software users.”

The Free Software Foundation is the principal organizational sponsor of the GNU Operating System. **Support GNU and the FSF** by buying manuals and gear, **joining the FSF** as an associate member, or making a **donation**, either directly to the FSF or via Flattr.

back to top



Please send general FSF & GNU inquiries to <gnu@gnu.org>. There are also other ways to contact the FSF. Broken links and other corrections or suggestions can be sent to <webmasters@gnu.org>.
Please see the Translations README for information on coordinating and submitting translations of this article.

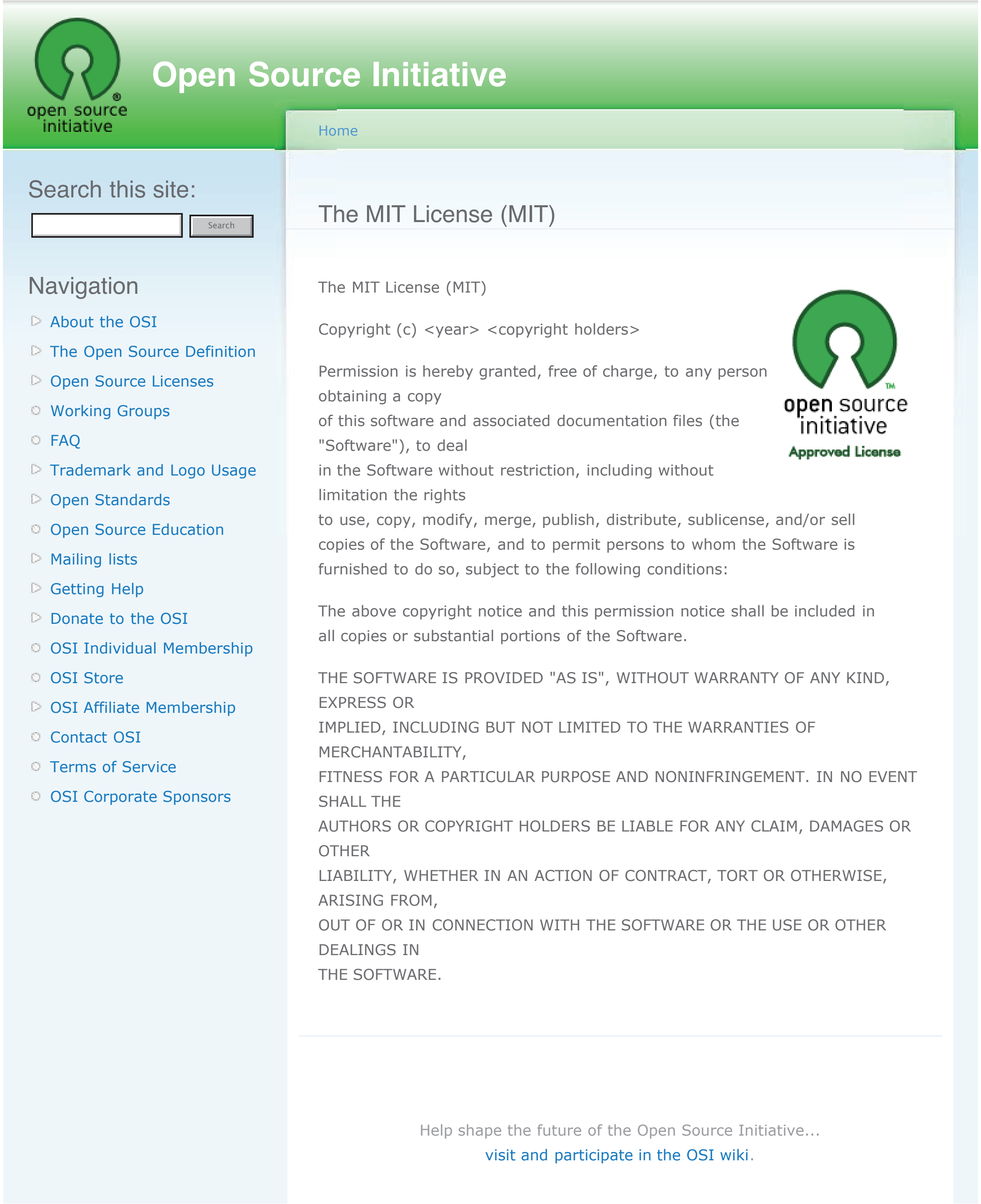
Copyright notice above.
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
Copyright Infringement Notification

Updated: $Date: 2014/04/12 12:39:51 $

OPEN SOURCE INITIATIVE

The MIT License

http://opensource.org/licenses/MIT



The MIT License (MIT)

The MIT License (MIT)

Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

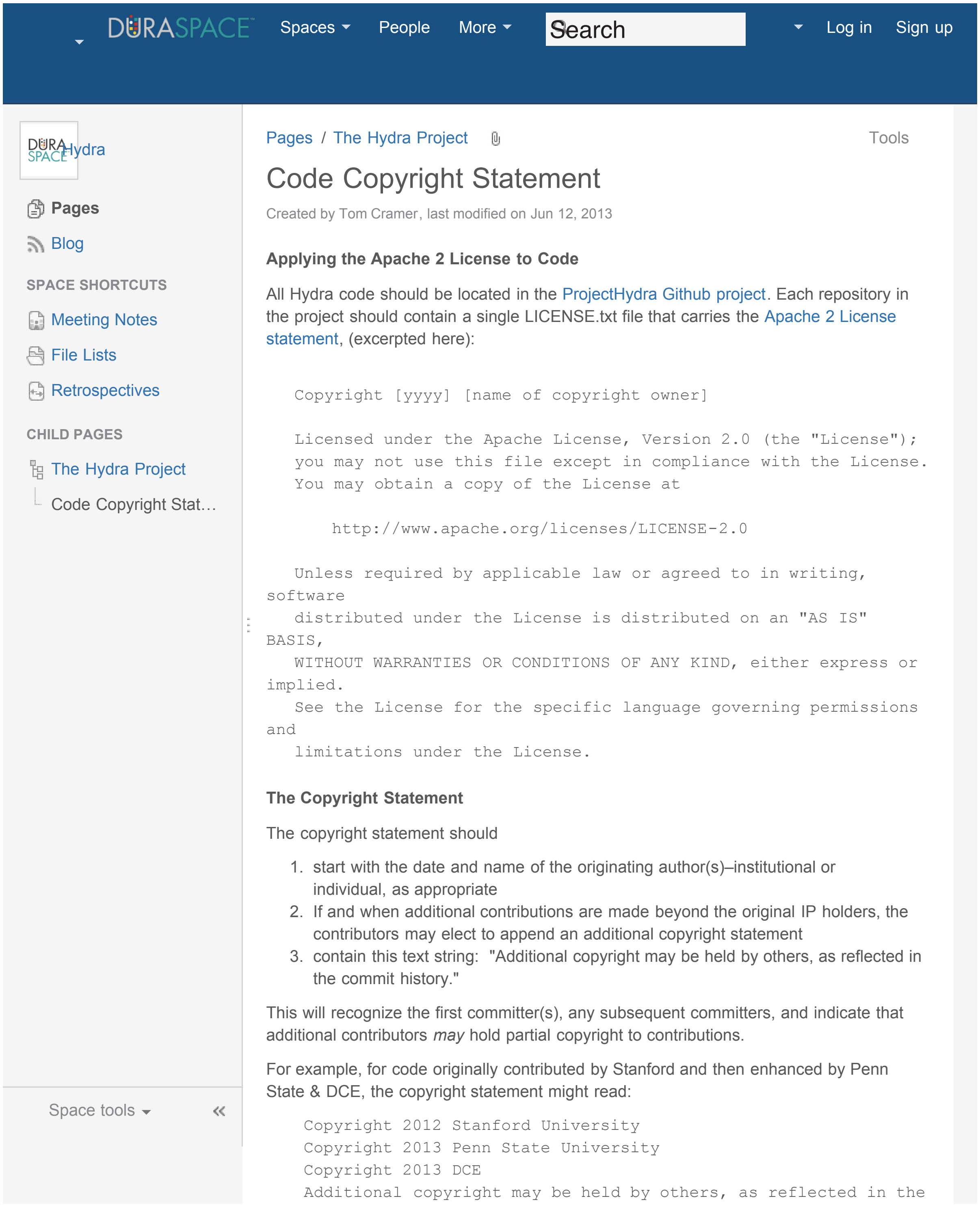
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Help shape the future of the Open Source Initiative...
visit and participate in the OSI wiki.

OSS Copyright Notices

DURASPACE
The Hydra Project Code Copyright Statement
https://wiki.duraspace.org/display/hydra/Code+Copyright+Statement



Code Copyright Statement

Created by Tom Cramer, last modified on Jun 12, 2013

Applying the Apache 2 License to Code

All Hydra code should be located in the ProjectHydra Github project. Each repository in the project should contain a single LICENSE.txt file that carries the Apache 2 License statement, (excerpted here):

```
   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing,
software
   distributed under the License is distributed on an "AS IS"
BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
   See the License for the specific language governing permissions
and
   limitations under the License.
```

The Copyright Statement

The copyright statement should

1. start with the date and name of the originating author(s)–institutional or individual, as appropriate
2. If and when additional contributions are made beyond the original IP holders, the contributors may elect to append an additional copyright statement
3. contain this text string: "Additional copyright may be held by others, as reflected in the commit history."

This will recognize the first committer(s), any subsequent committers, and indicate that additional contributors *may* hold partial copyright to contributions.

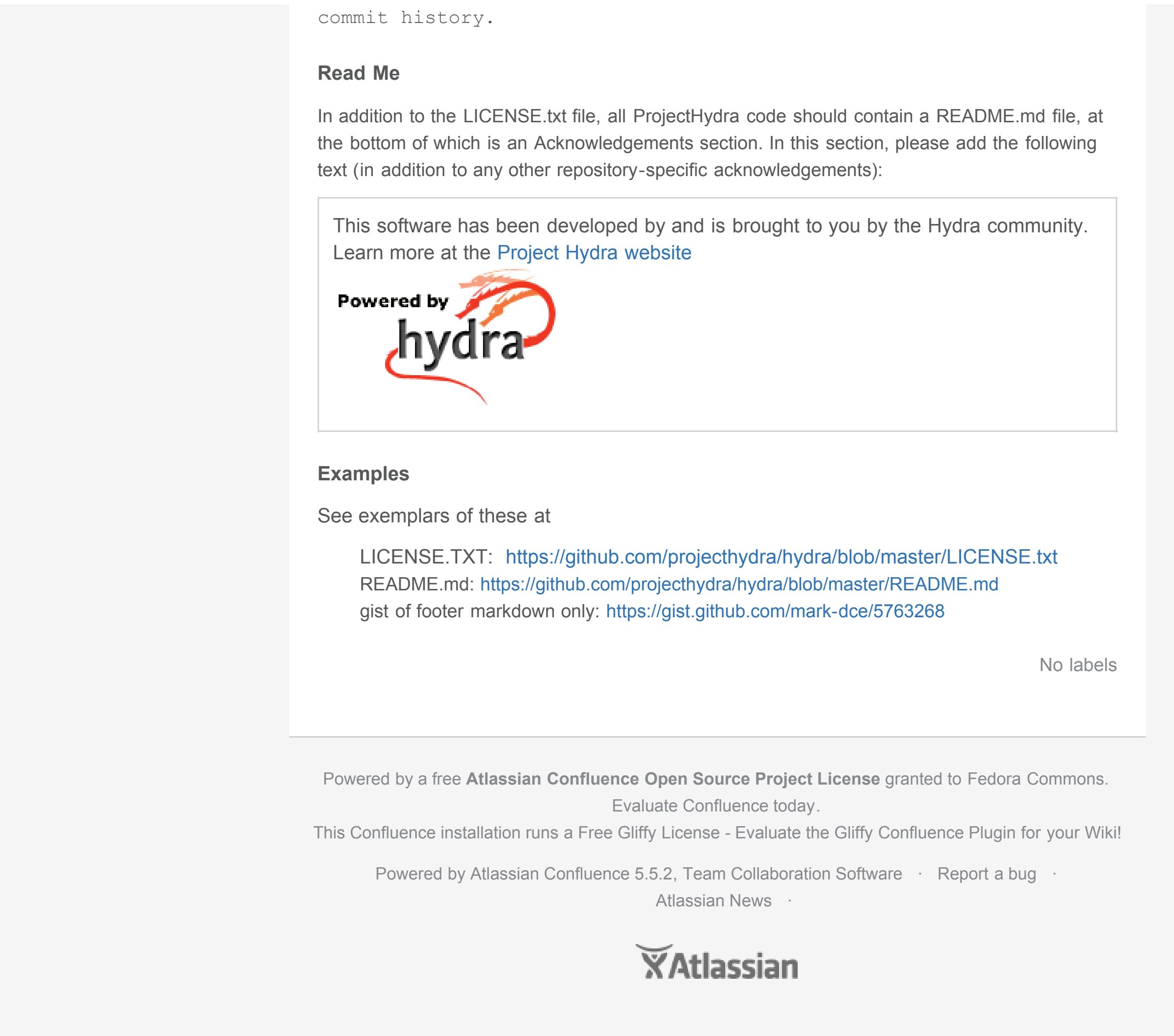
For example, for code originally contributed by Stanford and then enhanced by Penn State & DCE, the copyright statement might read:

```
    Copyright 2012 Stanford University
    Copyright 2013 Penn State University
    Copyright 2013 DCE
    Additional copyright may be held by others, as reflected in the
```

DURASPACE

The Hydra Project Code Copyright Statement

https://wiki.duraspace.org/display/hydra/Code+Copyright+Statement



```
commit history.
```

Read Me

In addition to the LICENSE.txt file, all ProjectHydra code should contain a README.md file, at the bottom of which is an Acknowledgements section. In this section, please add the following text (in addition to any other repository-specific acknowledgements):

> This software has been developed by and is brought to you by the Hydra community. Learn more at the Project Hydra website
>
> Powered by hydra

Examples

See exemplars of these at

LICENSE.TXT: https://github.com/projecthydra/hydra/blob/master/LICENSE.txt
README.md: https://github.com/projecthydra/hydra/blob/master/README.md
gist of footer markdown only: https://gist.github.com/mark-dce/5763268

No labels

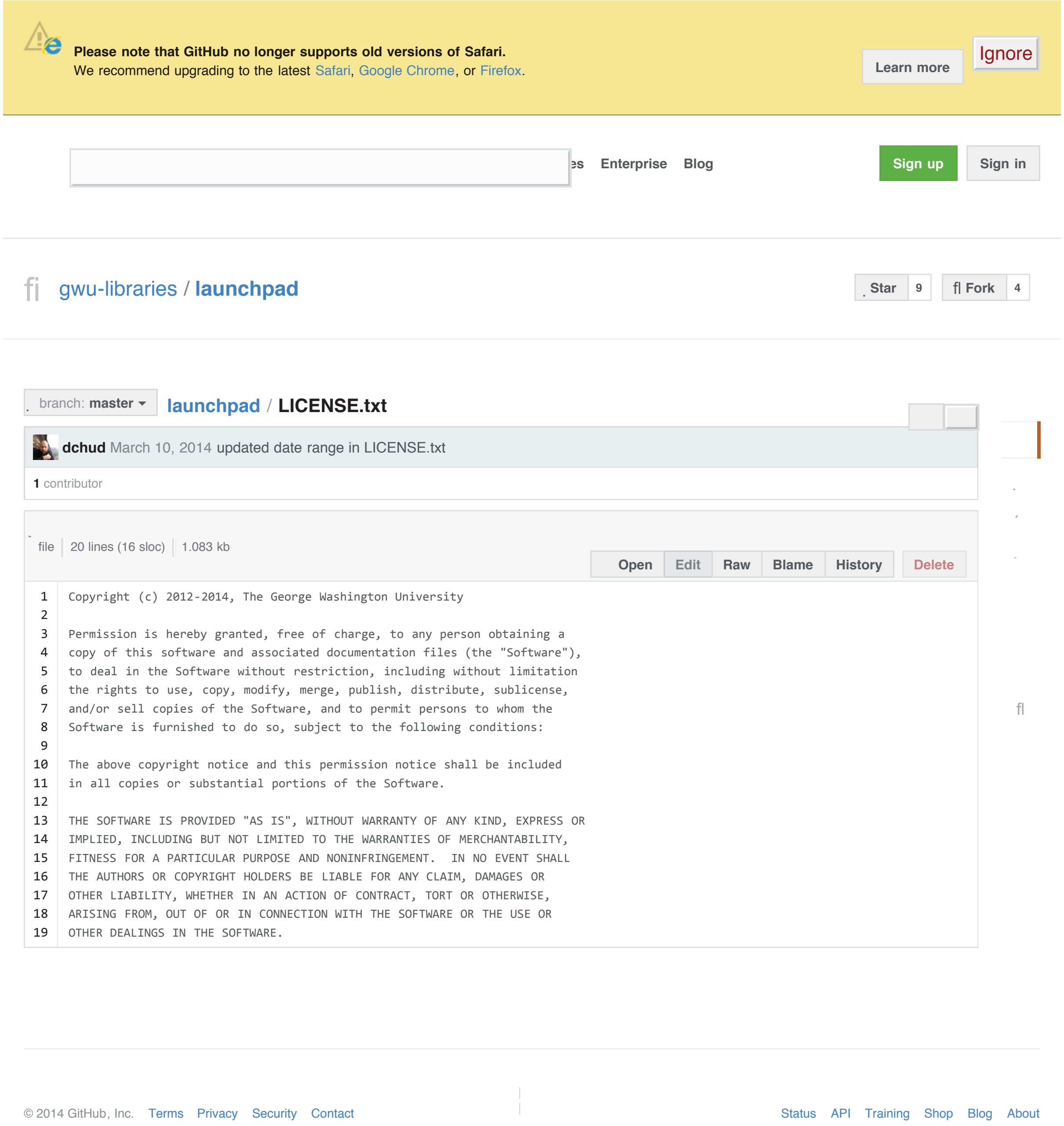
Powered by a free **Atlassian Confluence Open Source Project License** granted to Fedora Commons. Evaluate Confluence today.
This Confluence installation runs a Free Gliffy License - Evaluate the Gliffy Confluence Plugin for your Wiki!

Powered by Atlassian Confluence 5.5.2, Team Collaboration Software · Report a bug · Atlassian News ·

GEORGE WASHINGTON UNIVERSITY

Software Copyright Notice

https://github.com/gwu-libraries/launchpad/blob/master/LICENSE.txt



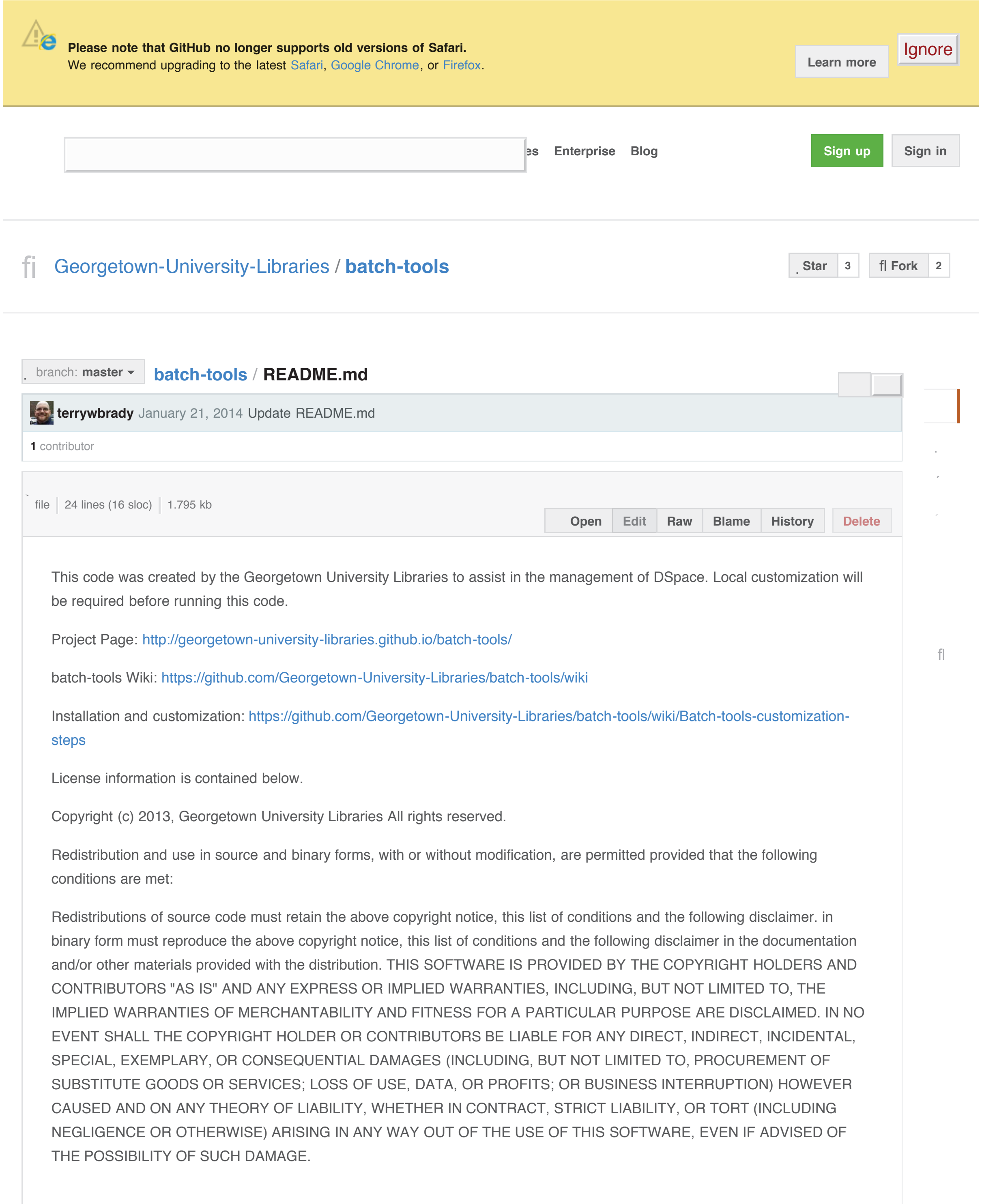
dchud March 10, 2014 updated date range in LICENSE.txt

```
Copyright (c) 2012-2014, The George Washington University

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT SHALL
THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
```

GEORGETOWN UNIVERSITY LIBRARIES

Software Copyright Notice

https://github.com/Georgetown-University-Libraries/batch-tools/blob/master/README.md


Please note that GitHub no longer supports old versions of Safari.
We recommend upgrading to the latest Safari, Google Chrome, or Firefox.
Learn more
Ignore
es
Enterprise
Blog
Sign up
Sign in
Georgetown-University-Libraries / batch-tools
Star
3
Fork
2
branch: master
batch-tools / README.md
terrywbrady January 21, 2014 Update README.md
1 contributor
file
24 lines (16 sloc)
1.795 kb
Open
Edit
Raw
Blame
History
Delete

This code was created by the Georgetown University Libraries to assist in the management of DSpace. Local customization will be required before running this code.

Project Page: http://georgetown-university-libraries.github.io/batch-tools/

batch-tools Wiki: https://github.com/Georgetown-University-Libraries/batch-tools/wiki

Installation and customization: https://github.com/Georgetown-University-Libraries/batch-tools/wiki/Batch-tools-customization-steps

License information is contained below.

Copyright (c) 2013, Georgetown University Libraries All rights reserved.

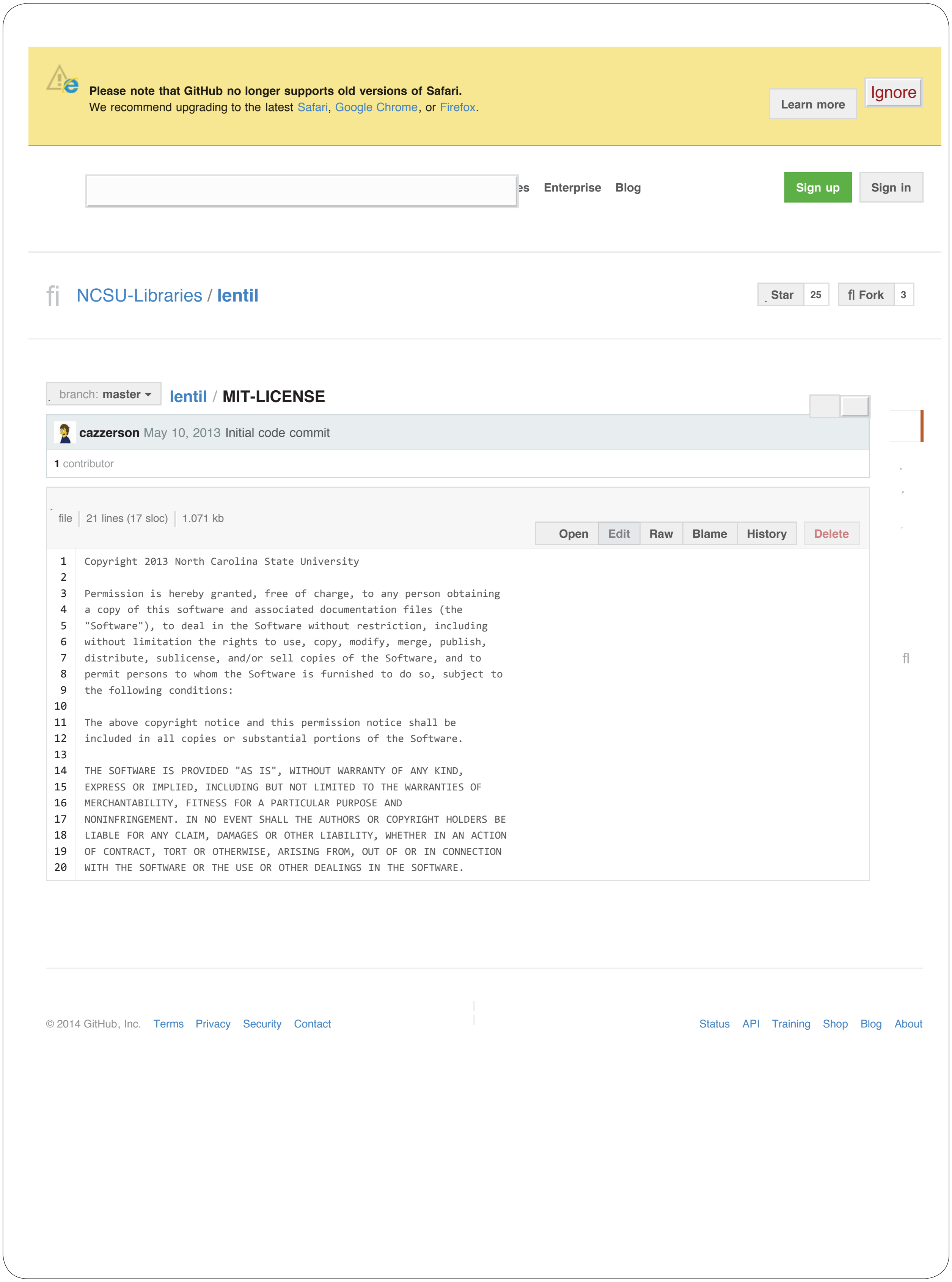
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NORTH CAROLINA STATE UNIVERSITY

Software Copyright Notice

https://github.com/NCSU-Libraries/lentil/blob/master/MIT-LICENSE



Please note that GitHub no longer supports old versions of Safari.
We recommend upgrading to the latest Safari, Google Chrome, or Firefox.

lentil / MIT-LICENSE

cazzerson May 10, 2013 Initial code commit

1 contributor

file | 21 lines (17 sloc) | 1.071 kb

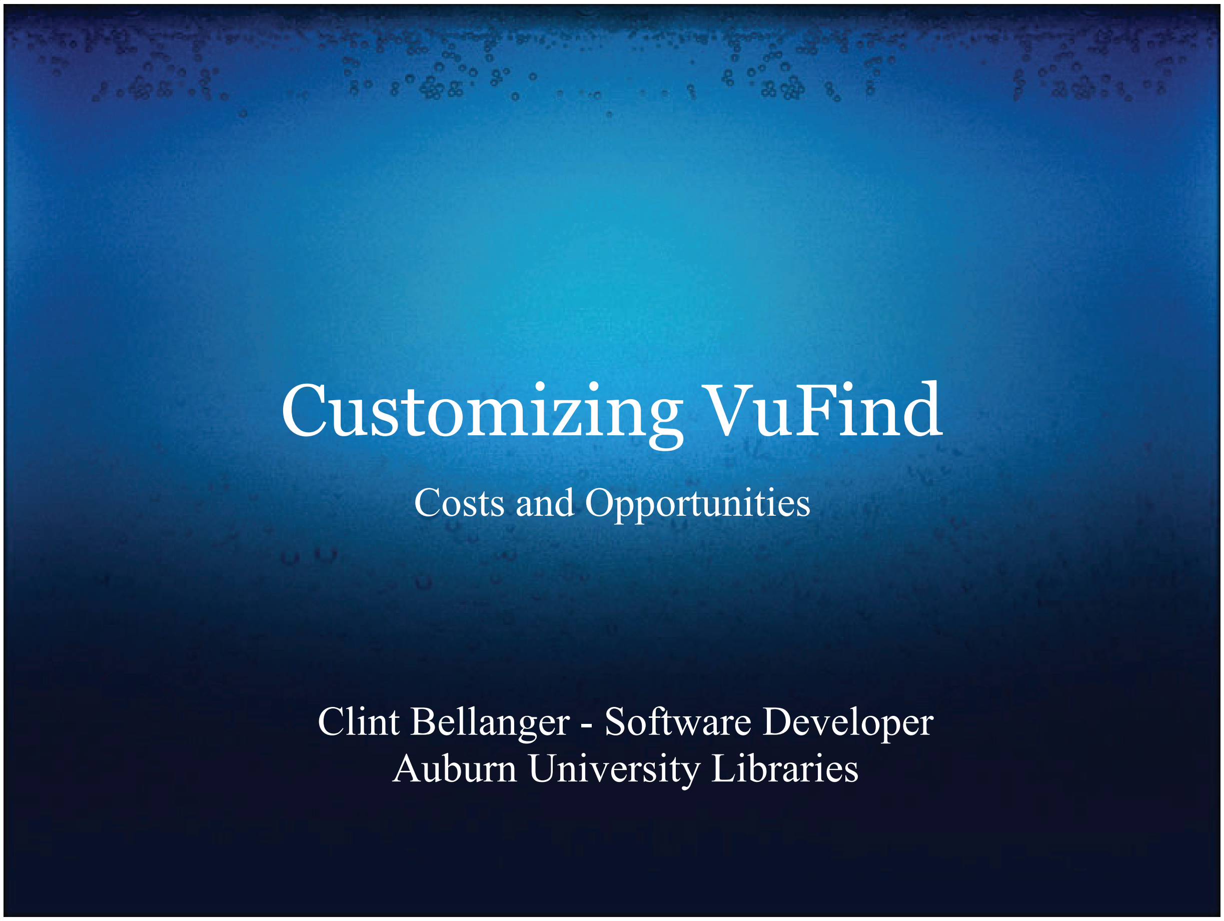
```
Copyright 2013 North Carolina State University

Permission is hereby granted, free of charge, to any person obtaining
a copy of this software and associated documentation files (the
"Software"), to deal in the Software without restriction, including
without limitation the rights to use, copy, modify, merge, publish,
distribute, sublicense, and/or sell copies of the Software, and to
permit persons to whom the Software is furnished to do so, subject to
the following conditions:

The above copyright notice and this permission notice shall be
included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

OSS Adoption Decision

AUBURN UNIVERSITY

Customizing VuFind

http://aurora.auburn.edu/repo/handle/11200/44104

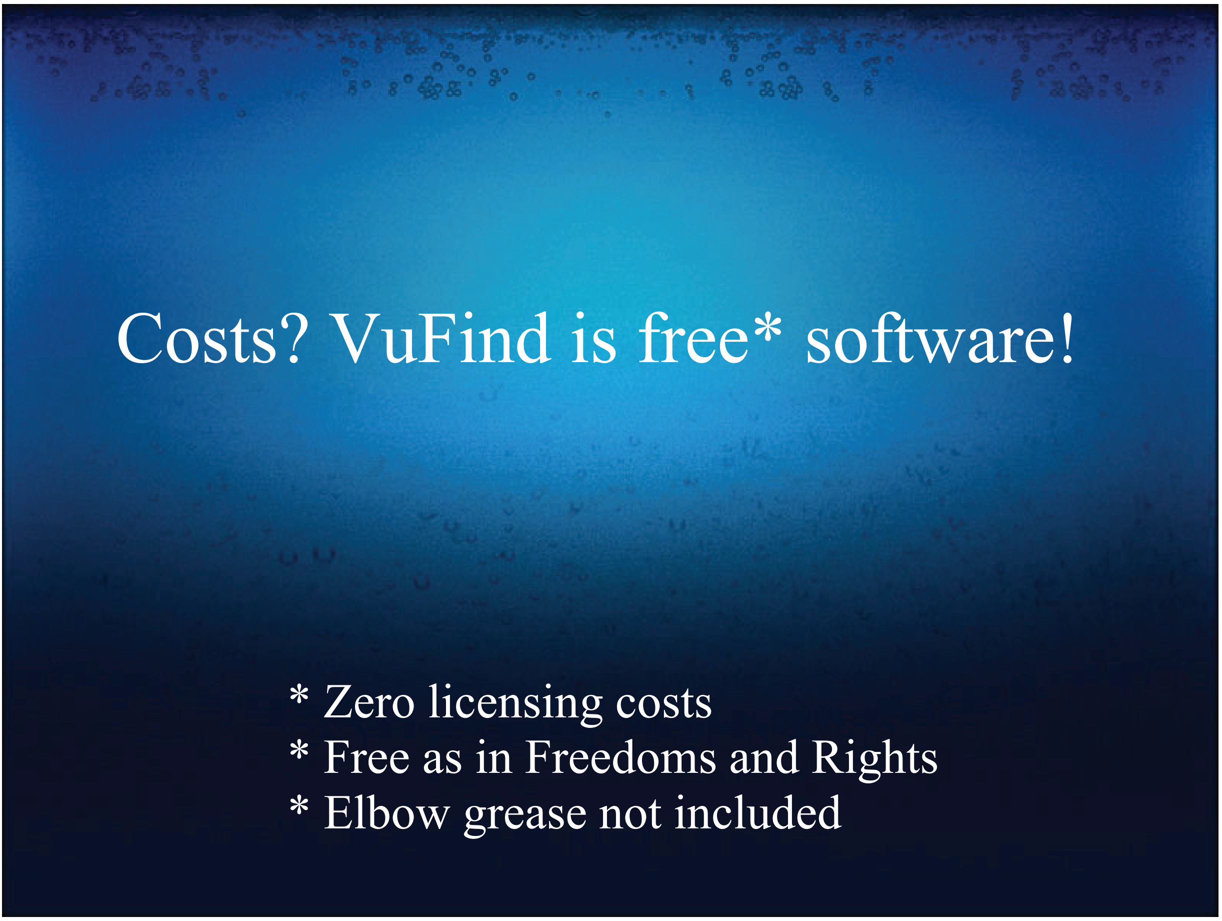


Customizing VuFind

Costs and Opportunities

Clint Bellanger - Software Developer
Auburn University Libraries

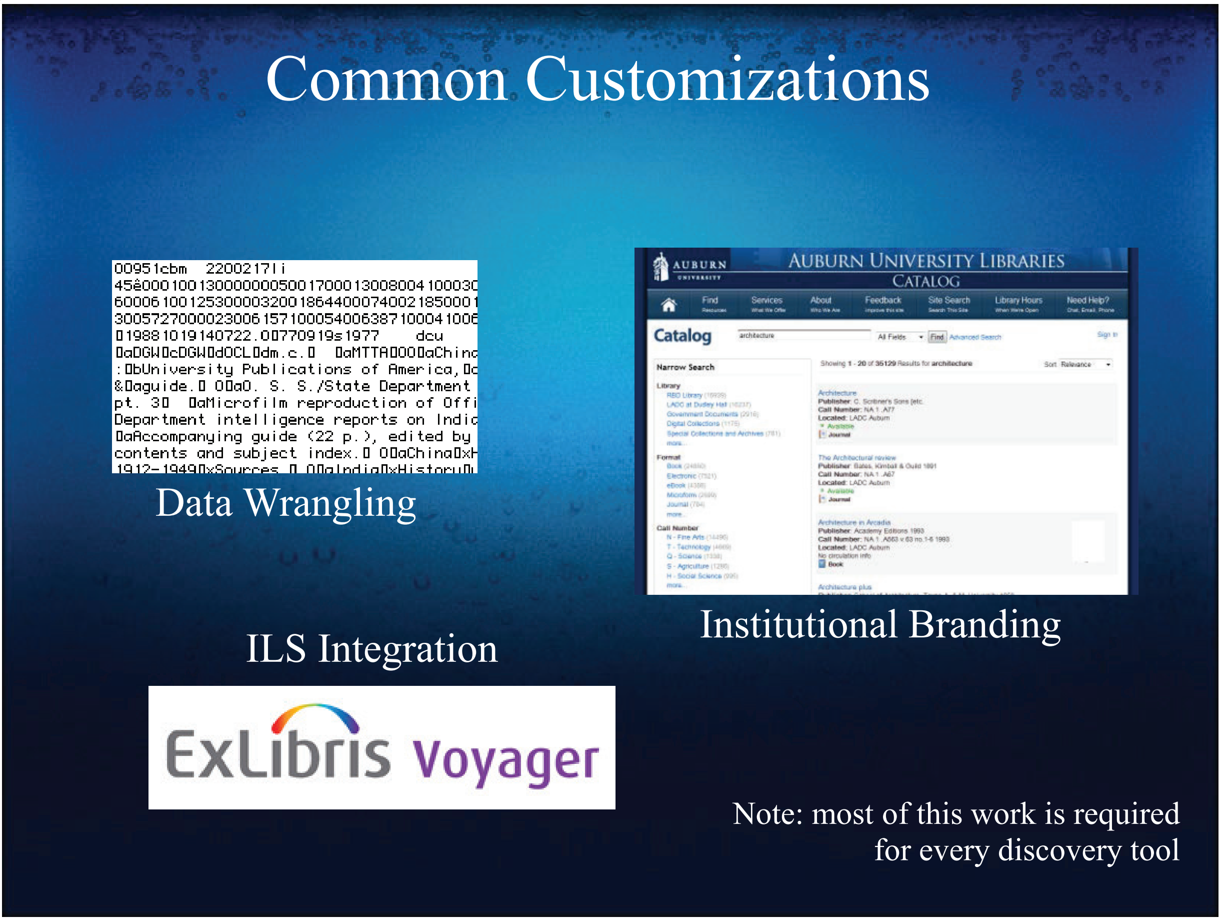
AUBURN UNIVERSITY
Customizing VuFind
http://aurora.auburn.edu/repo/handle/11200/44104



Costs? VuFind is free* software!

* Zero licensing costs
* Free as in Freedoms and Rights
* Elbow grease not included

Customizing VuFind
http://aurora.auburn.edu/repo/handle/11200/44104



Common Customizations

```
00951cbm  2200217li
45ê0001001300000000500170001300800410003C
60006100125300003200186440007400218500001
300572700002300615710005400638710004100E
01988101914072.00770919s1977     dcu
0aDGW0cDGW0dOCL0dm.c.0  0aMTTA0000aChina
:0bUniversity Publications of America,0c
&0aguide.0 00a0. S. S./State Department
pt. 30  0aMicrofilm reproduction of Offi
Department intelligence reports on India
0aAccompanying guide (22 p.), edited by
contents and subject index.0 00aChina0xH
1912-19490xSources.0 00aIndia0xHistory0u
```

Data Wrangling

Showing 1 - 20 of 35129 Results for **architecture**

Architecture
Publisher: C. Scribner's Sons [etc.
Call Number: NA 1 .A77
Located: LADC Auburn
Available
Journal

The Architectural review
Publisher: Bates, Kimball & Guild 1891
Call Number: NA 1 .A67
Located: LADC Auburn
Available
Journal

Architecture in Arcadia
Publisher: Academy Editions 1993
Call Number: NA 1 .A563 v.63 no.1-6 1993
Located: LADC Auburn
No circulation info
Book

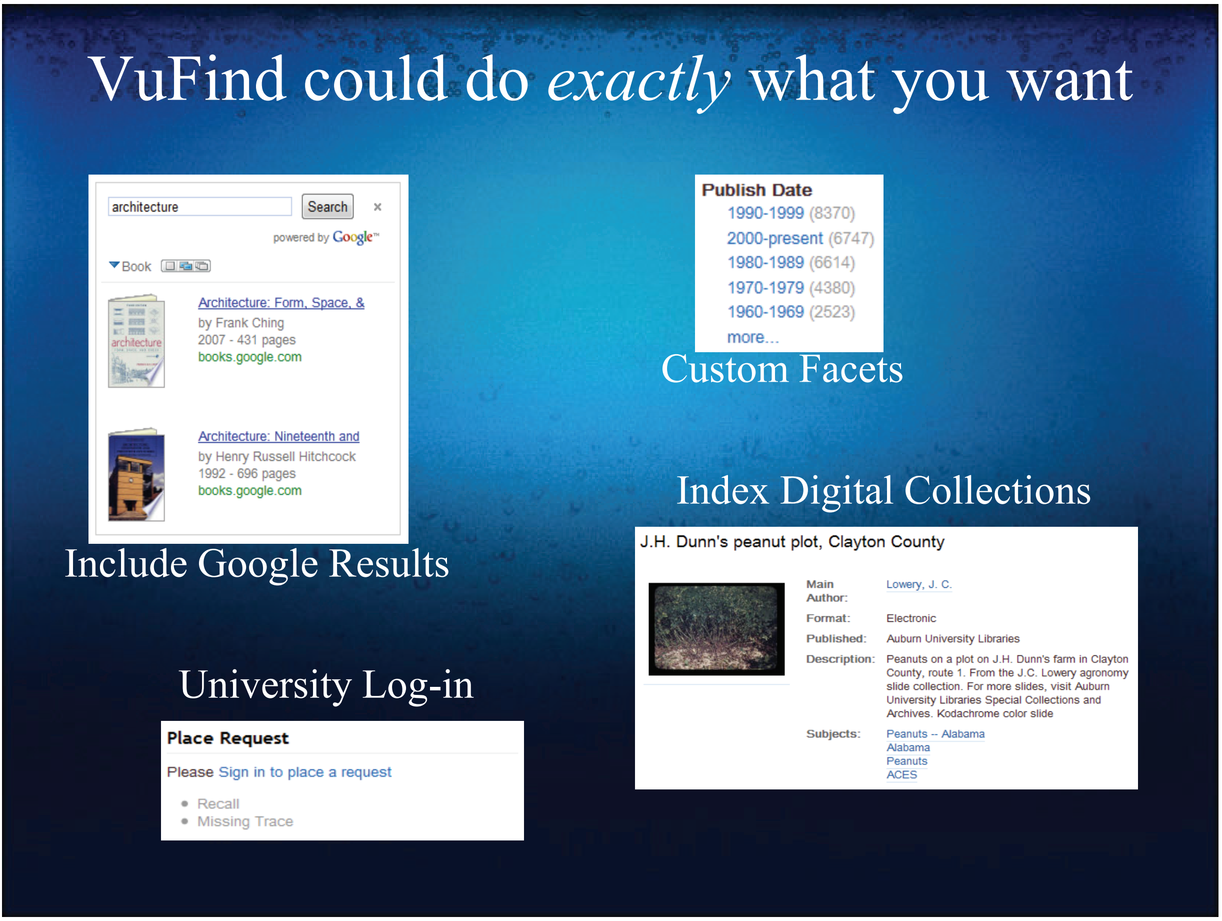
Architecture plus

Institutional Branding

ILS Integration

Note: most of this work is required for every discovery tool

AUBURN UNIVERSITY
Customizing VuFind
http://aurora.auburn.edu/repo/handle/11200/44104



VuFind could do *exactly* what you want

Architecture: Form, Space, &
by Frank Ching
2007 - 431 pages
books.google.com

Architecture: Nineteenth and
by Henry Russell Hitchcock
1992 - 696 pages
books.google.com

Include Google Results

Publish Date

1990-1999	(8370)
2000-present	(6747)
1980-1989	(6614)
1970-1979	(4380)
1960-1969	(2523)
more...	

Custom Facets

Index Digital Collections

J.H. Dunn's peanut plot, Clayton County

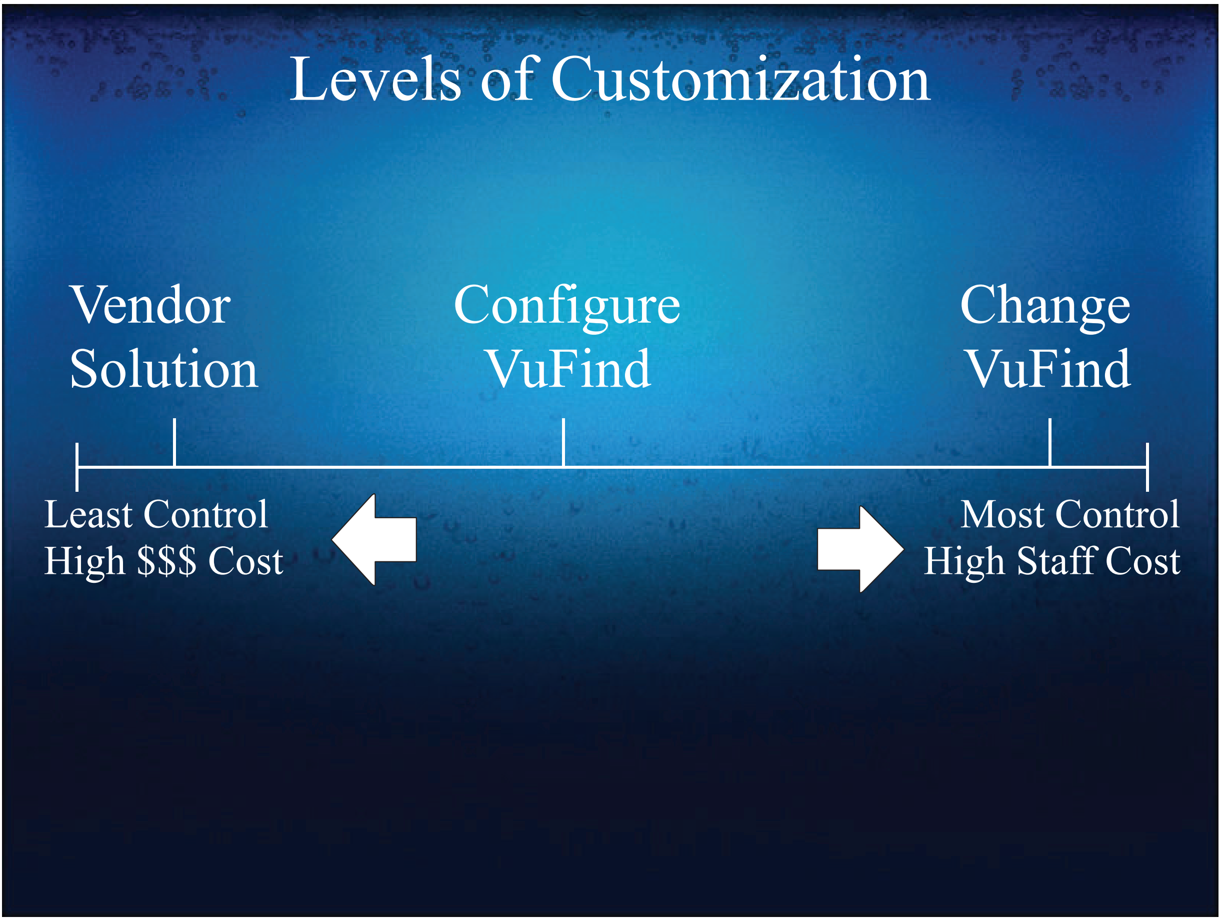
Main Author:	Lowery, J. C.
Format:	Electronic
Published:	Auburn University Libraries
Description:	Peanuts on a plot on J.H. Dunn's farm in Clayton County, route 1. From the J.C. Lowery agronomy slide collection. For more slides, visit Auburn University Libraries Special Collections and Archives. Kodachrome color slide
Subjects:	Peanuts -- Alabama Alabama Peanuts ACES

University Log-in

Place Request

Please Sign in to place a request

- Recall
- Missing Trace



Levels of Customization

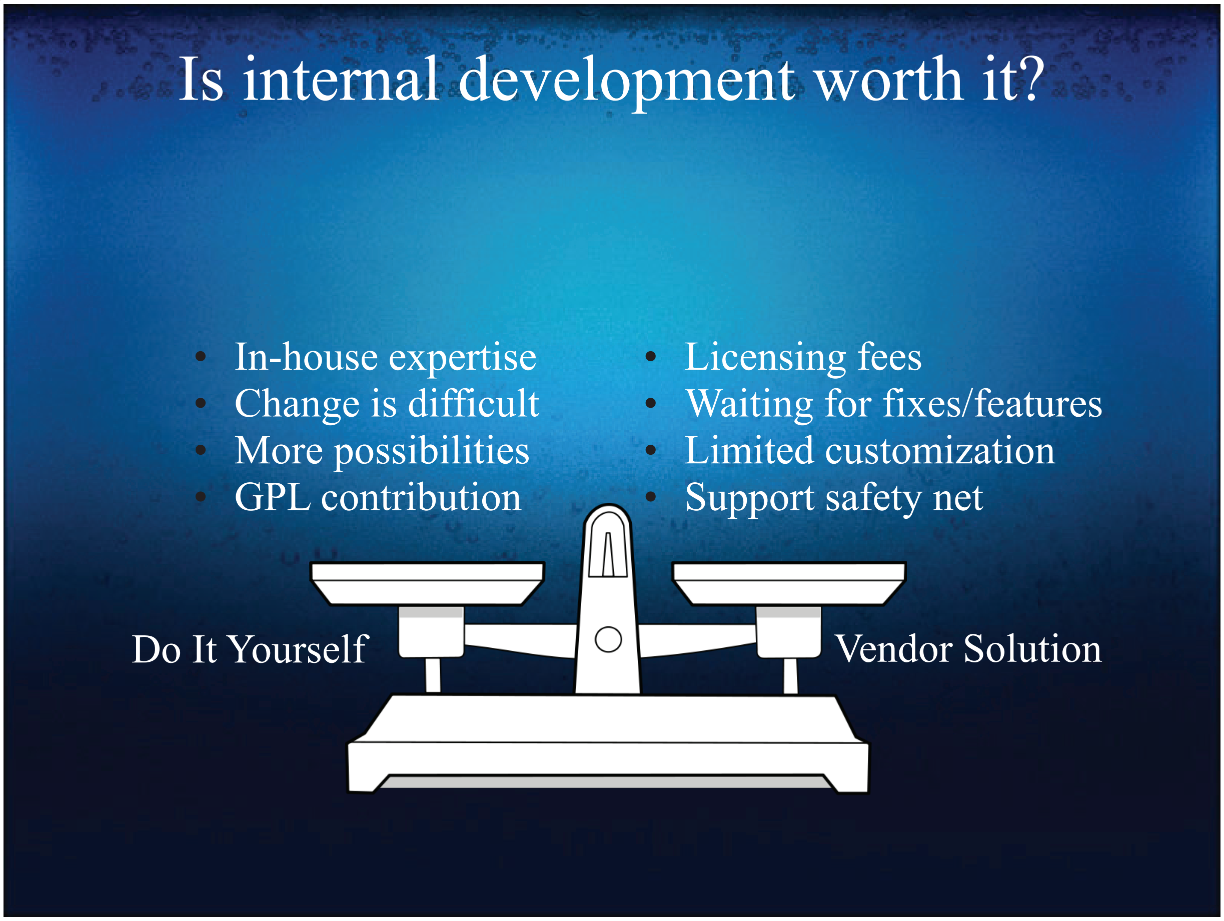
Vendor Solution

Configure VuFind

Change VuFind

Least Control
High $$$ Cost

Most Control
High Staff Cost



Is internal development worth it?

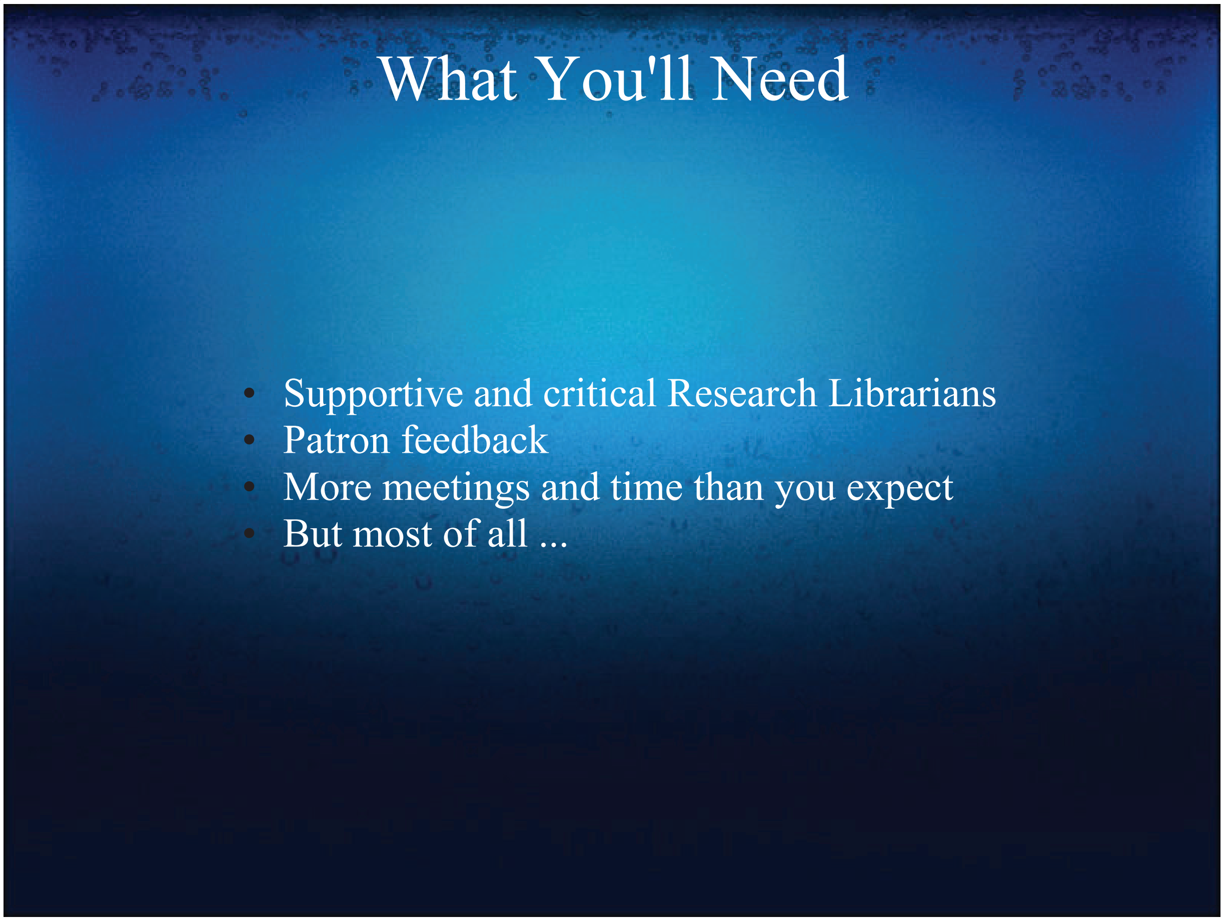
- In-house expertise
- Change is difficult
- More possibilities
- GPL contribution

- Licensing fees
- Waiting for fixes/features
- Limited customization
- Support safety net

Do It Yourself

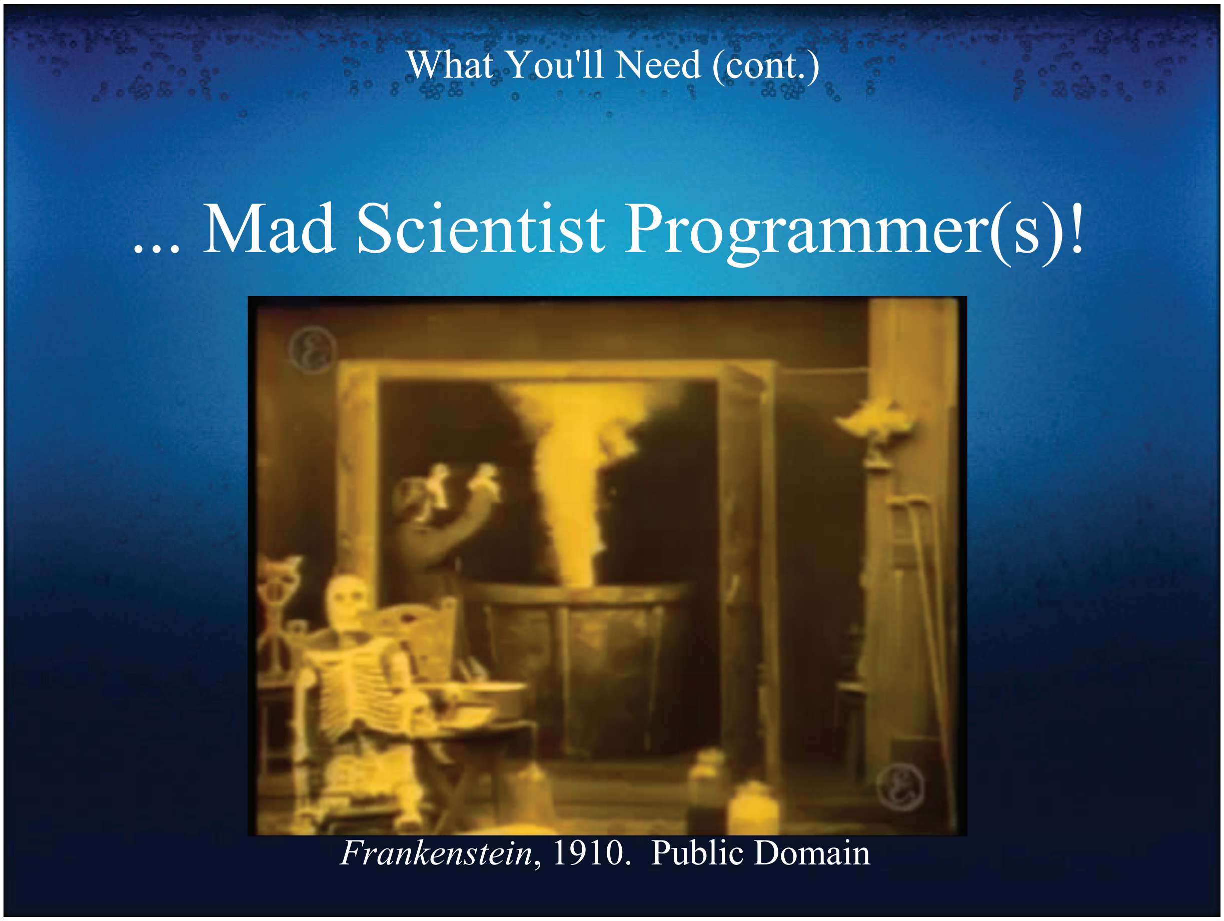
Vendor Solution

AUBURN UNIVERSITY
Customizing VuFind
http://aurora.auburn.edu/repo/handle/11200/44104



What You'll Need

- Supportive and critical Research Librarians
- Patron feedback
- More meetings and time than you expect
- But most of all ...

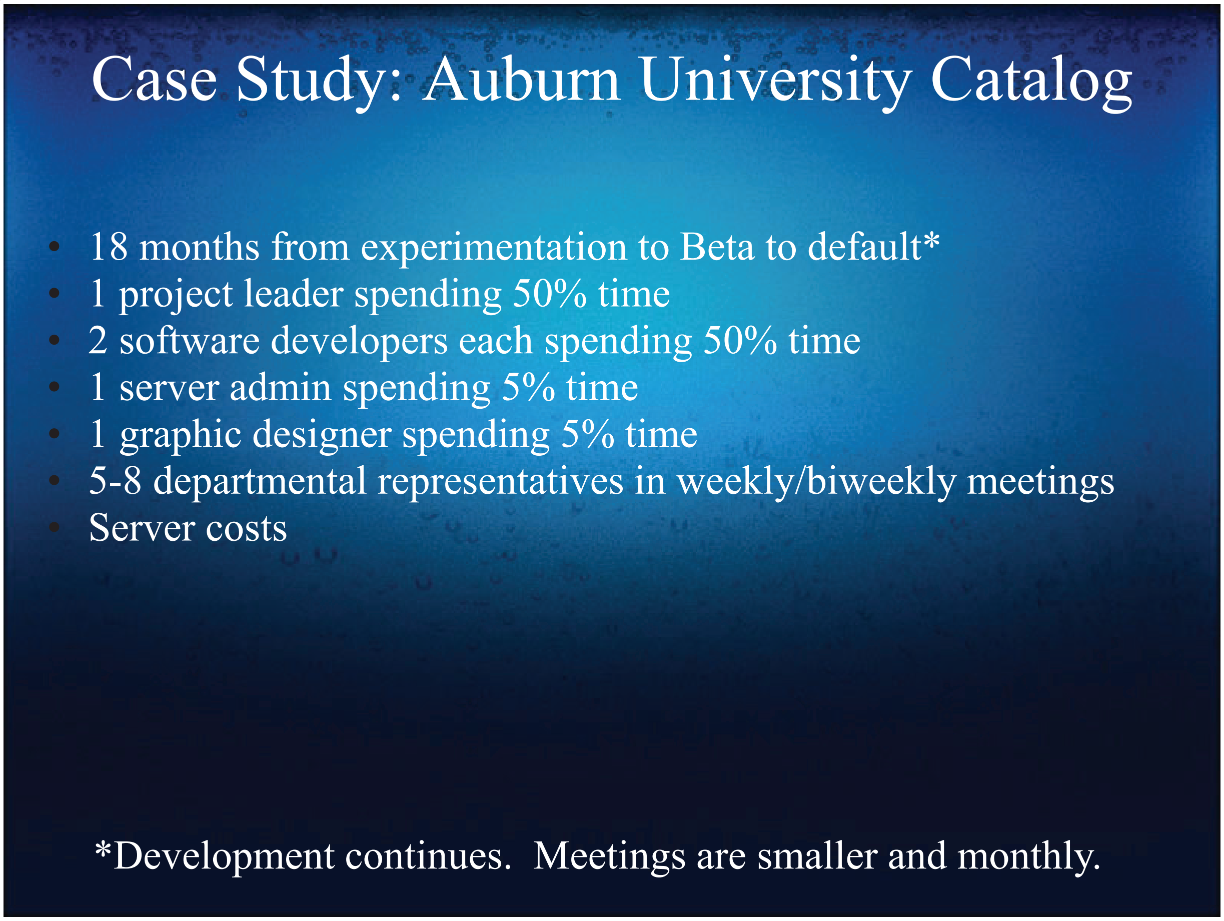


What You'll Need (cont.)

… Mad Scientist Programmer(s)!

Frankenstein, 1910. Public Domain

AUBURN UNIVERSITY
Customizing VuFind
http://aurora.auburn.edu/repo/handle/11200/44104

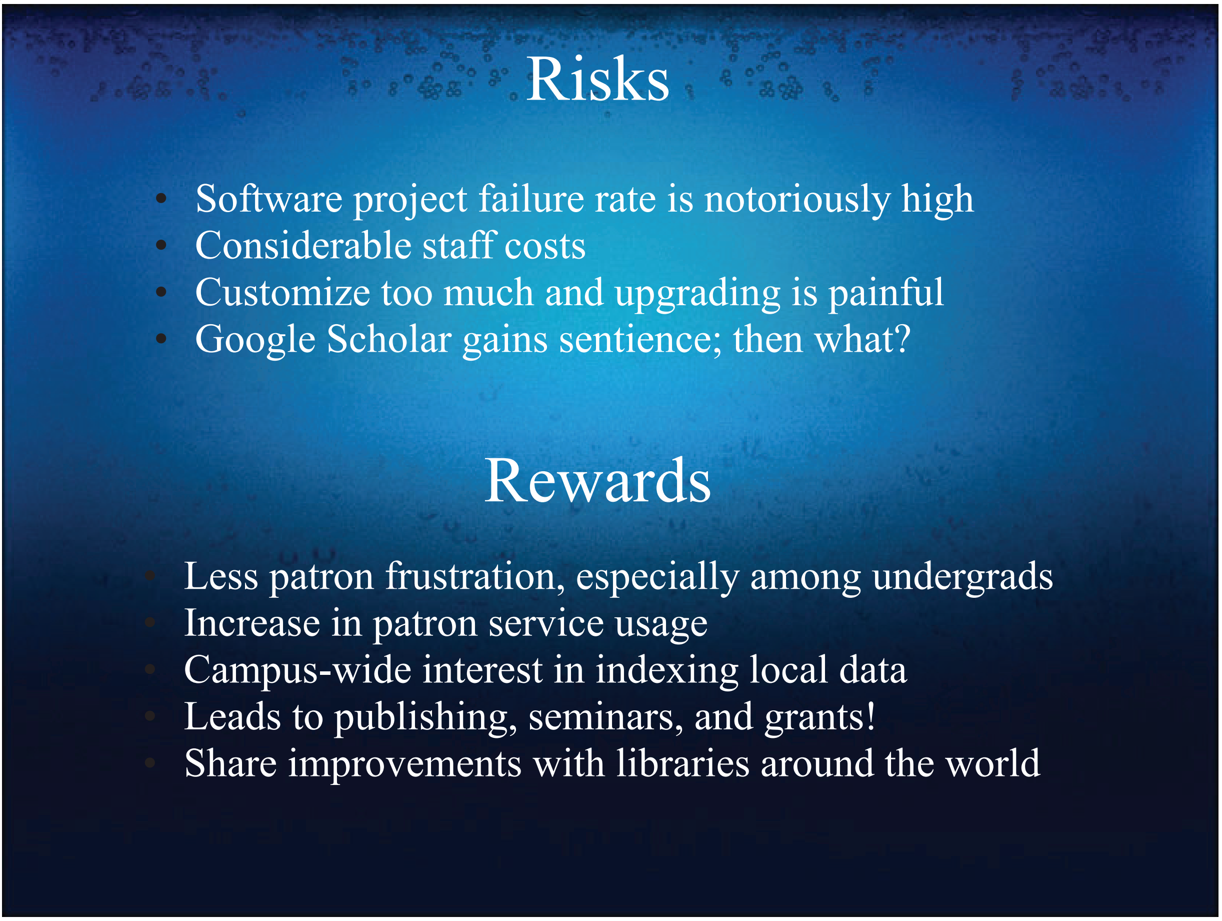


Case Study: Auburn University Catalog

- 18 months from experimentation to Beta to default*
- 1 project leader spending 50% time
- 2 software developers each spending 50% time
- 1 server admin spending 5% time
- 1 graphic designer spending 5% time
- 5-8 departmental representatives in weekly/biweekly meetings
- Server costs

*Development continues. Meetings are smaller and monthly.

AUBURN UNIVERSITY
Customizing VuFind
http://aurora.auburn.edu/repo/handle/11200/44104



Risks

- Software project failure rate is notoriously high
- Considerable staff costs
- Customize too much and upgrading is painful
- Google Scholar gains sentience; then what?

Rewards

- Less patron frustration, especially among undergrads
- Increase in patron service usage
- Campus-wide interest in indexing local data
- Leads to publishing, seminars, and grants!
- Share improvements with libraries around the world

AUBURN UNIVERSITY

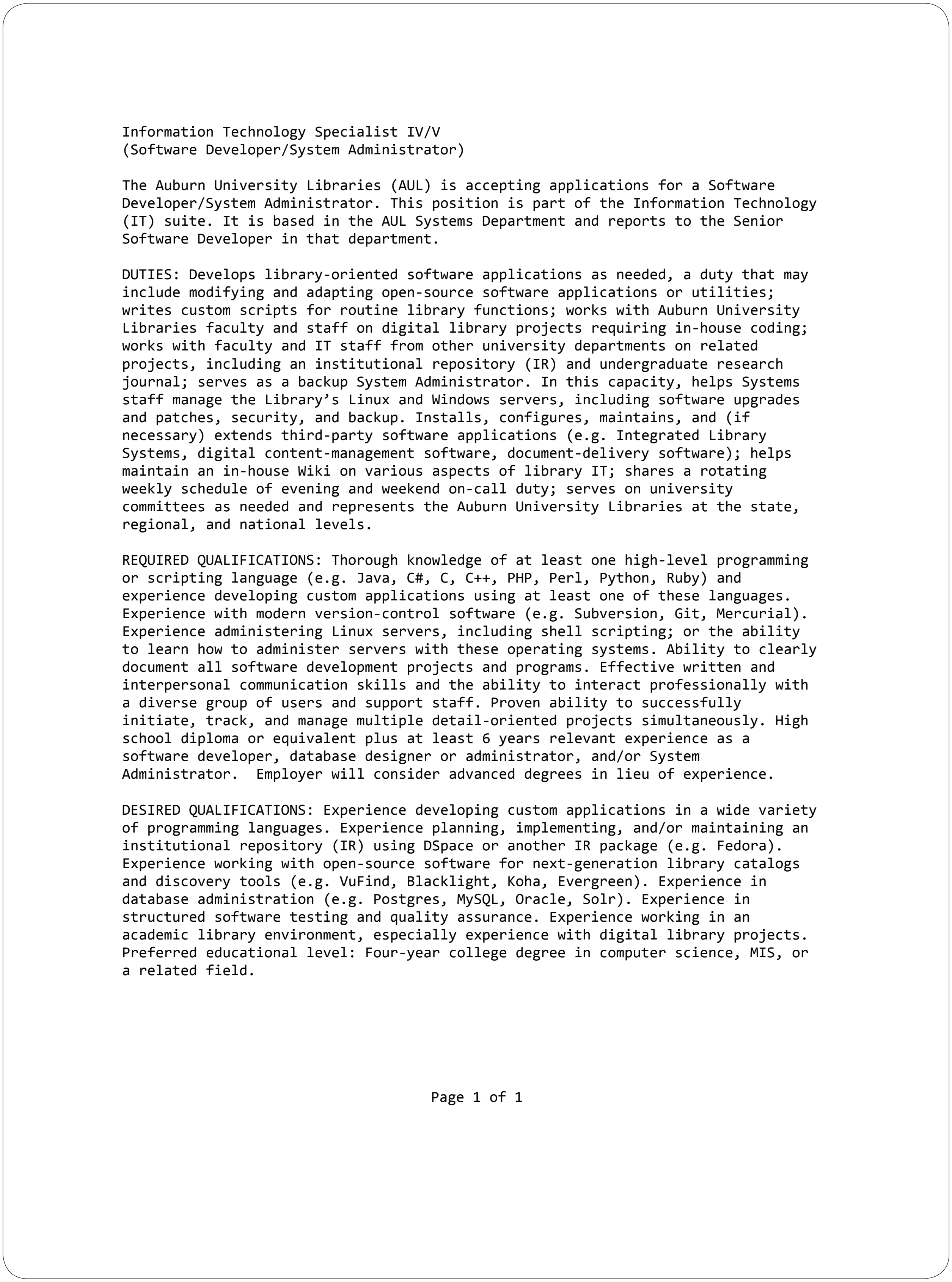
Customizing VuFind

http://aurora.auburn.edu/repo/handle/11200/44104

IT'S ALIVE!

Job Descriptions of OSS Contributors

AUBURN UNIVERSITY
Information Technology Specialist IV/V



Information Technology Specialist IV/V
(Software Developer/System Administrator)

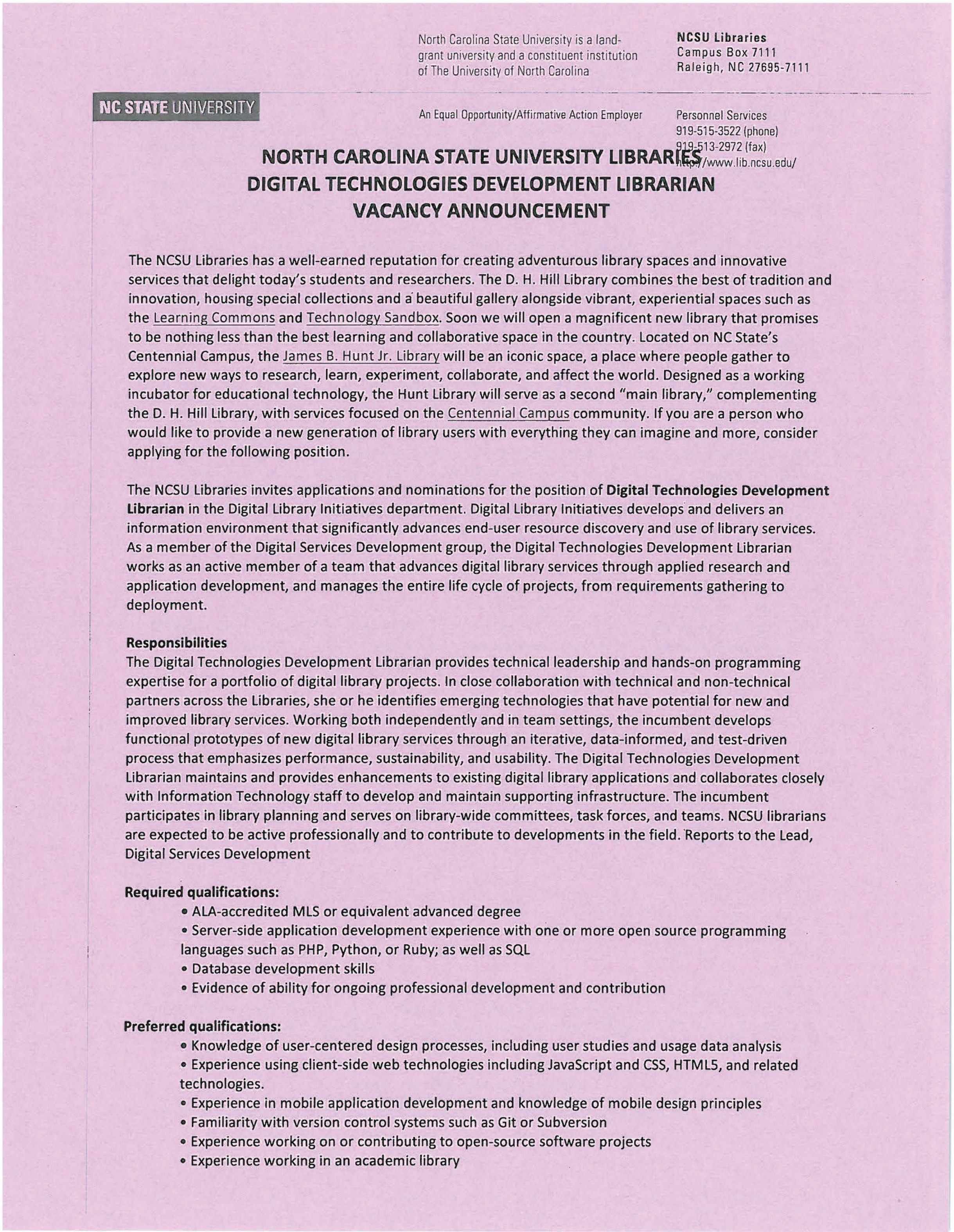
The Auburn University Libraries (AUL) is accepting applications for a Software Developer/System Administrator. This position is part of the Information Technology (IT) suite. It is based in the AUL Systems Department and reports to the Senior Software Developer in that department.

DUTIES: Develops library-oriented software applications as needed, a duty that may include modifying and adapting open-source software applications or utilities; writes custom scripts for routine library functions; works with Auburn University Libraries faculty and staff on digital library projects requiring in-house coding; works with faculty and IT staff from other university departments on related projects, including an institutional repository (IR) and undergraduate research journal; serves as a backup System Administrator. In this capacity, helps Systems staff manage the Library’s Linux and Windows servers, including software upgrades and patches, security, and backup. Installs, configures, maintains, and (if necessary) extends third-party software applications (e.g. Integrated Library Systems, digital content-management software, document-delivery software); helps maintain an in-house Wiki on various aspects of library IT; shares a rotating weekly schedule of evening and weekend on-call duty; serves on university committees as needed and represents the Auburn University Libraries at the state, regional, and national levels.

REQUIRED QUALIFICATIONS: Thorough knowledge of at least one high-level programming or scripting language (e.g. Java, C#, C, C++, PHP, Perl, Python, Ruby) and experience developing custom applications using at least one of these languages. Experience with modern version-control software (e.g. Subversion, Git, Mercurial). Experience administering Linux servers, including shell scripting; or the ability to learn how to administer servers with these operating systems. Ability to clearly document all software development projects and programs. Effective written and interpersonal communication skills and the ability to interact professionally with a diverse group of users and support staff. Proven ability to successfully initiate, track, and manage multiple detail-oriented projects simultaneously. High school diploma or equivalent plus at least 6 years relevant experience as a software developer, database designer or administrator, and/or System Administrator. Employer will consider advanced degrees in lieu of experience.

DESIRED QUALIFICATIONS: Experience developing custom applications in a wide variety of programming languages. Experience planning, implementing, and/or maintaining an institutional repository (IR) using DSpace or another IR package (e.g. Fedora). Experience working with open-source software for next-generation library catalogs and discovery tools (e.g. VuFind, Blacklight, Koha, Evergreen). Experience in database administration (e.g. Postgres, MySQL, Oracle, Solr). Experience in structured software testing and quality assurance. Experience working in an academic library environment, especially experience with digital library projects. Preferred educational level: Four-year college degree in computer science, MIS, or a related field.

Page 1 of 1



North Carolina State University is a land-grant university and a constituent institution of The University of North Carolina

NCSU Libraries
Campus Box 7111
Raleigh, NC 27695-7111

NC STATE UNIVERSITY

An Equal Opportunity/Affirmative Action Employer

Personnel Services
919-515-3522 (phone)
919-513-2972 (fax)
http://www.lib.ncsu.edu/

NORTH CAROLINA STATE UNIVERSITY LIBRARIES
DIGITAL TECHNOLOGIES DEVELOPMENT LIBRARIAN
VACANCY ANNOUNCEMENT

The NCSU Libraries has a well-earned reputation for creating adventurous library spaces and innovative services that delight today's students and researchers. The D. H. Hill Library combines the best of tradition and innovation, housing special collections and a beautiful gallery alongside vibrant, experiential spaces such as the Learning Commons and Technology Sandbox. Soon we will open a magnificent new library that promises to be nothing less than the best learning and collaborative space in the country. Located on NC State's Centennial Campus, the James B. Hunt Jr. Library will be an iconic space, a place where people gather to explore new ways to research, learn, experiment, collaborate, and affect the world. Designed as a working incubator for educational technology, the Hunt Library will serve as a second "main library," complementing the D. H. Hill Library, with services focused on the Centennial Campus community. If you are a person who would like to provide a new generation of library users with everything they can imagine and more, consider applying for the following position.

The NCSU Libraries invites applications and nominations for the position of **Digital Technologies Development Librarian** in the Digital Library Initiatives department. Digital Library Initiatives develops and delivers an information environment that significantly advances end-user resource discovery and use of library services. As a member of the Digital Services Development group, the Digital Technologies Development Librarian works as an active member of a team that advances digital library services through applied research and application development, and manages the entire life cycle of projects, from requirements gathering to deployment.

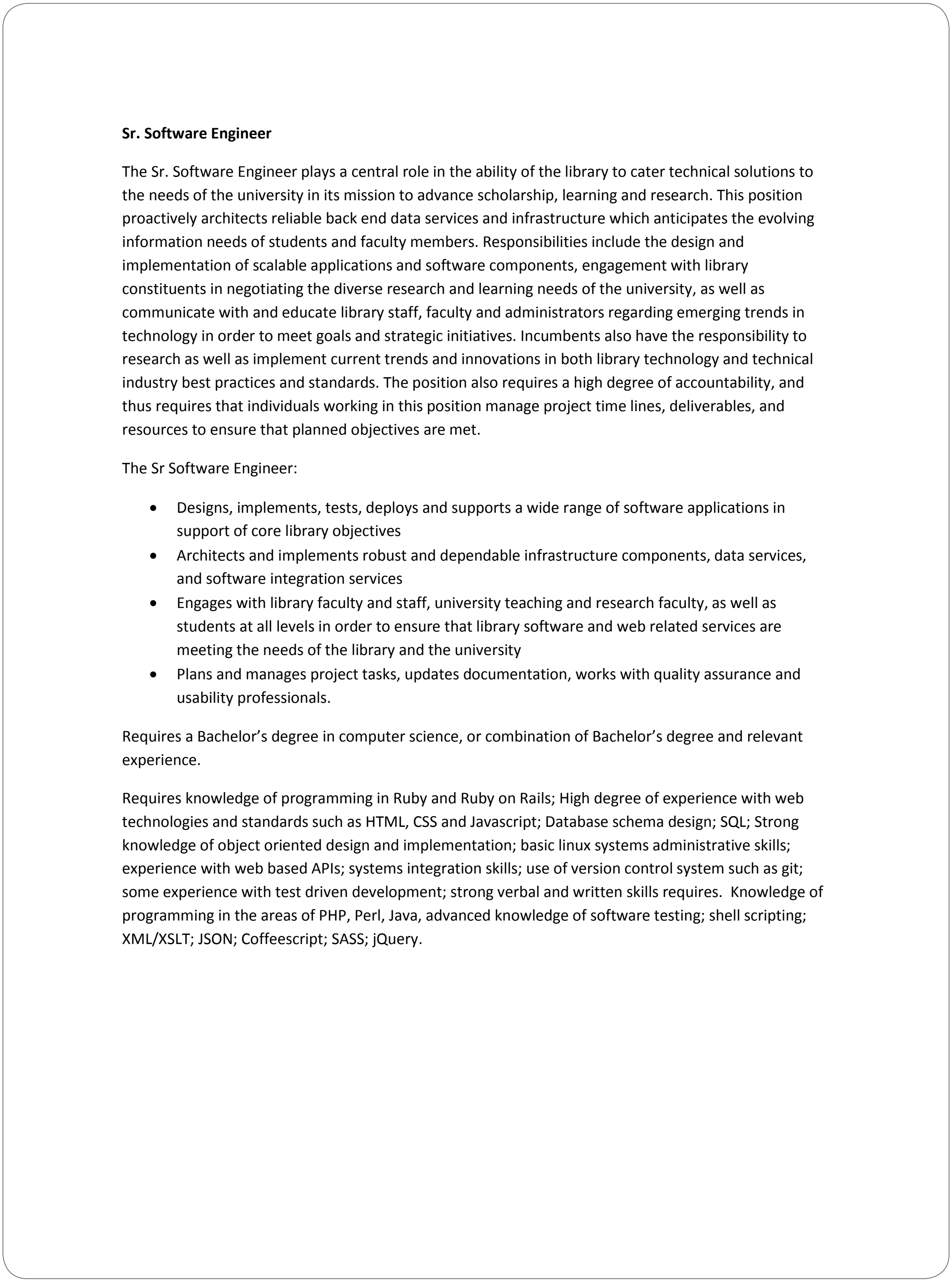
Responsibilities
The Digital Technologies Development Librarian provides technical leadership and hands-on programming expertise for a portfolio of digital library projects. In close collaboration with technical and non-technical partners across the Libraries, she or he identifies emerging technologies that have potential for new and improved library services. Working both independently and in team settings, the incumbent develops functional prototypes of new digital library services through an iterative, data-informed, and test-driven process that emphasizes performance, sustainability, and usability. The Digital Technologies Development Librarian maintains and provides enhancements to existing digital library applications and collaborates closely with Information Technology staff to develop and maintain supporting infrastructure. The incumbent participates in library planning and serves on library-wide committees, task forces, and teams. NCSU librarians are expected to be active professionally and to contribute to developments in the field. Reports to the Lead, Digital Services Development

Required qualifications:

- ALA-accredited MLS or equivalent advanced degree
- Server-side application development experience with one or more open source programming languages such as PHP, Python, or Ruby; as well as SQL
- Database development skills
- Evidence of ability for ongoing professional development and contribution

Preferred qualifications:

- Knowledge of user-centered design processes, including user studies and usage data analysis
- Experience using client-side web technologies including JavaScript and CSS, HTML5, and related technologies.
- Experience in mobile application development and knowledge of mobile design principles
- Familiarity with version control systems such as Git or Subversion
- Experience working on or contributing to open-source software projects
- Experience working in an academic library



Sr. Software Engineer

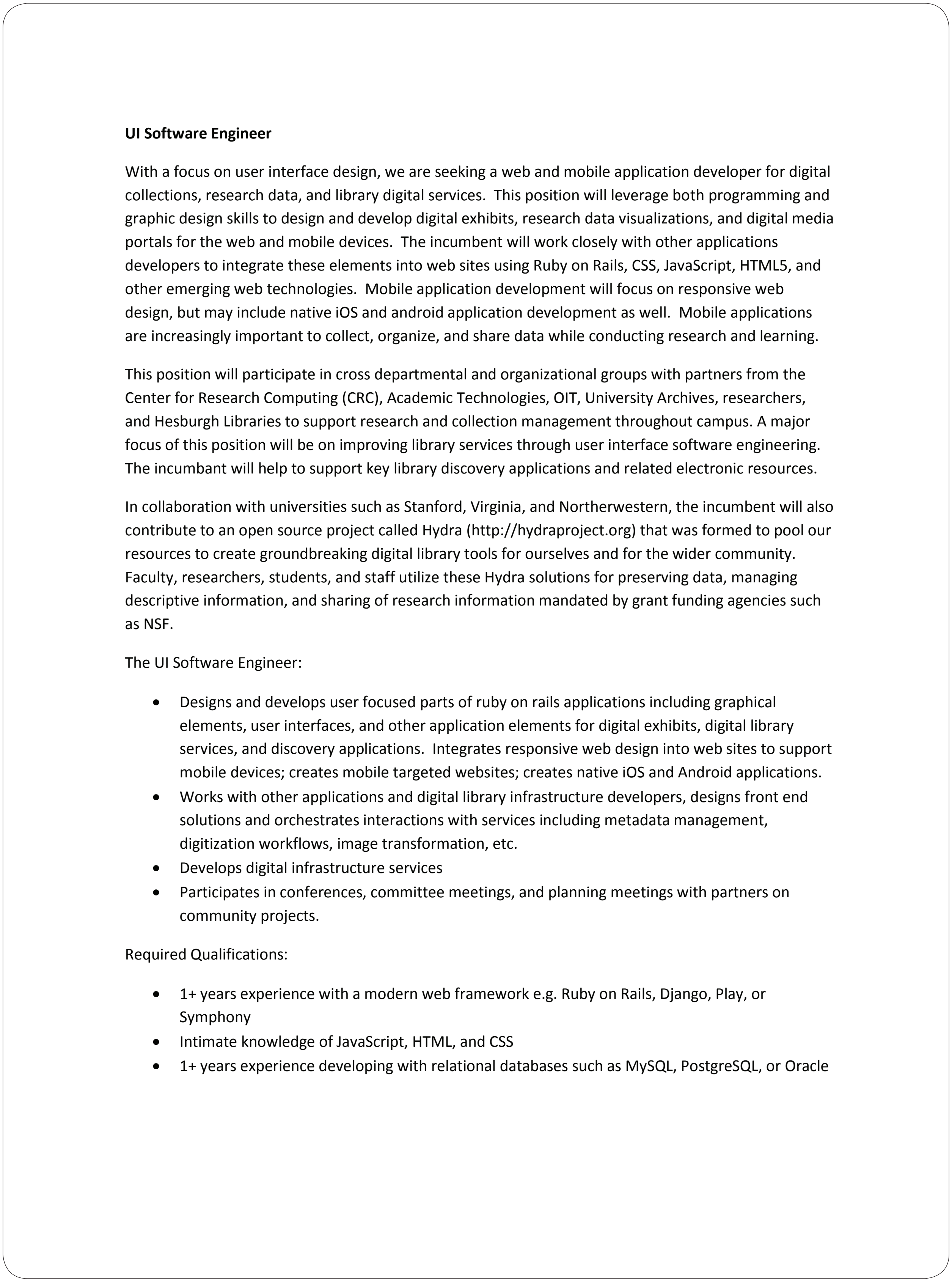
The Sr. Software Engineer plays a central role in the ability of the library to cater technical solutions to the needs of the university in its mission to advance scholarship, learning and research. This position proactively architects reliable back end data services and infrastructure which anticipates the evolving information needs of students and faculty members. Responsibilities include the design and implementation of scalable applications and software components, engagement with library constituents in negotiating the diverse research and learning needs of the university, as well as communicate with and educate library staff, faculty and administrators regarding emerging trends in technology in order to meet goals and strategic initiatives. Incumbents also have the responsibility to research as well as implement current trends and innovations in both library technology and technical industry best practices and standards. The position also requires a high degree of accountability, and thus requires that individuals working in this position manage project time lines, deliverables, and resources to ensure that planned objectives are met.

The Sr Software Engineer:

- Designs, implements, tests, deploys and supports a wide range of software applications in support of core library objectives
- Architects and implements robust and dependable infrastructure components, data services, and software integration services
- Engages with library faculty and staff, university teaching and research faculty, as well as students at all levels in order to ensure that library software and web related services are meeting the needs of the library and the university
- Plans and manages project tasks, updates documentation, works with quality assurance and usability professionals.

Requires a Bachelor’s degree in computer science, or combination of Bachelor’s degree and relevant experience.

Requires knowledge of programming in Ruby and Ruby on Rails; High degree of experience with web technologies and standards such as HTML, CSS and Javascript; Database schema design; SQL; Strong knowledge of object oriented design and implementation; basic linux systems administrative skills; experience with web based APIs; systems integration skills; use of version control system such as git; some experience with test driven development; strong verbal and written skills requires. Knowledge of programming in the areas of PHP, Perl, Java, advanced knowledge of software testing; shell scripting; XML/XSLT; JSON; Coffeescript; SASS; jQuery.



UI Software Engineer

With a focus on user interface design, we are seeking a web and mobile application developer for digital collections, research data, and library digital services. This position will leverage both programming and graphic design skills to design and develop digital exhibits, research data visualizations, and digital media portals for the web and mobile devices. The incumbent will work closely with other applications developers to integrate these elements into web sites using Ruby on Rails, CSS, JavaScript, HTML5, and other emerging web technologies. Mobile application development will focus on responsive web design, but may include native iOS and android application development as well. Mobile applications are increasingly important to collect, organize, and share data while conducting research and learning.

This position will participate in cross departmental and organizational groups with partners from the Center for Research Computing (CRC), Academic Technologies, OIT, University Archives, researchers, and Hesburgh Libraries to support research and collection management throughout campus. A major focus of this position will be on improving library services through user interface software engineering. The incumbant will help to support key library discovery applications and related electronic resources.

In collaboration with universities such as Stanford, Virginia, and Northerwestern, the incumbent will also contribute to an open source project called Hydra (http://hydraproject.org) that was formed to pool our resources to create groundbreaking digital library tools for ourselves and for the wider community. Faculty, researchers, students, and staff utilize these Hydra solutions for preserving data, managing descriptive information, and sharing of research information mandated by grant funding agencies such as NSF.

The UI Software Engineer:

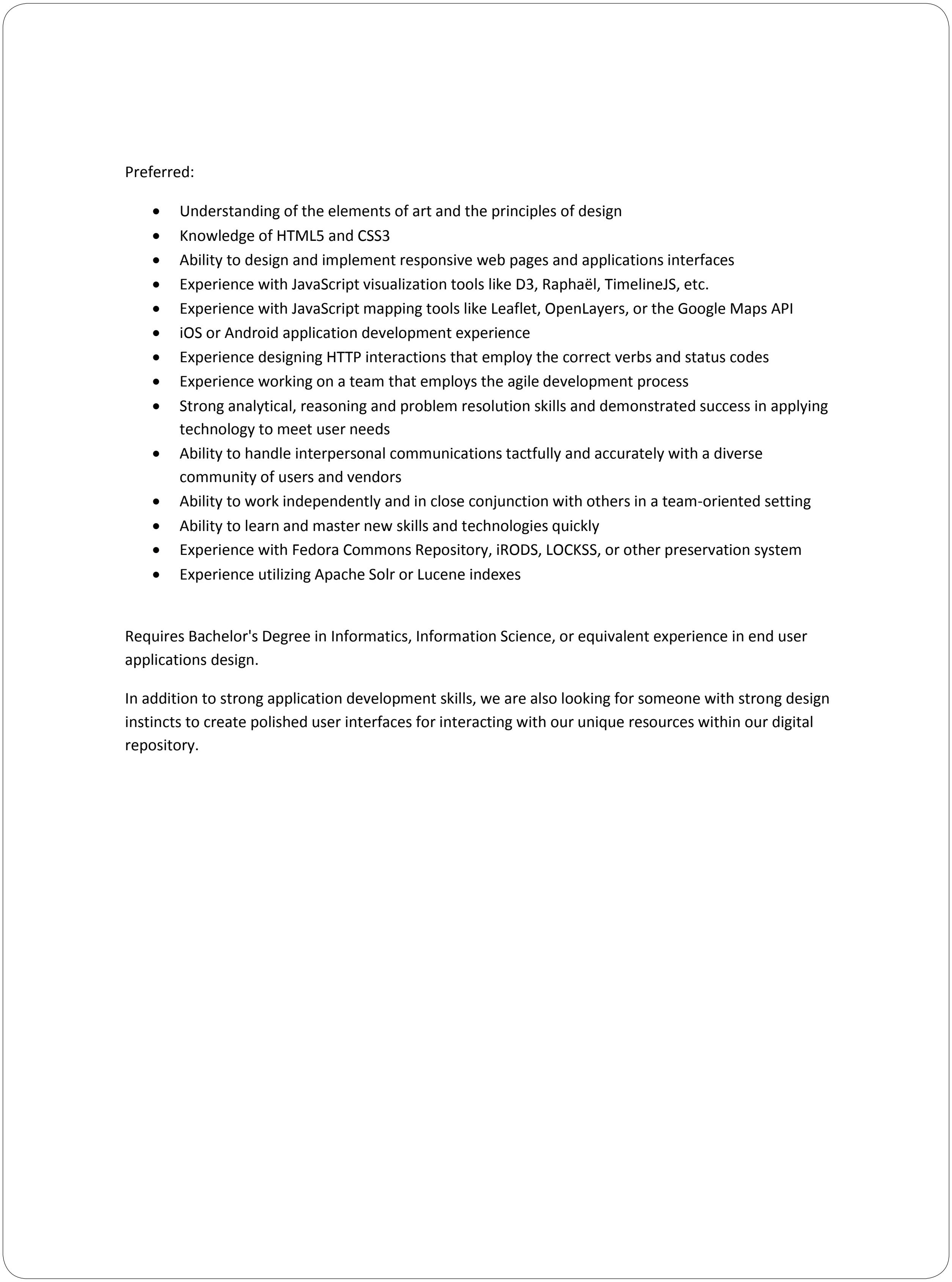
- Designs and develops user focused parts of ruby on rails applications including graphical elements, user interfaces, and other application elements for digital exhibits, digital library services, and discovery applications. Integrates responsive web design into web sites to support mobile devices; creates mobile targeted websites; creates native iOS and Android applications.
- Works with other applications and digital library infrastructure developers, designs front end solutions and orchestrates interactions with services including metadata management, digitization workflows, image transformation, etc.
- Develops digital infrastructure services
- Participates in conferences, committee meetings, and planning meetings with partners on community projects.

Required Qualifications:

- 1+ years experience with a modern web framework e.g. Ruby on Rails, Django, Play, or Symphony
- Intimate knowledge of JavaScript, HTML, and CSS
- 1+ years experience developing with relational databases such as MySQL, PostgreSQL, or Oracle

UNIVERSITY OF NOTRE DAME

UI Software Engineer



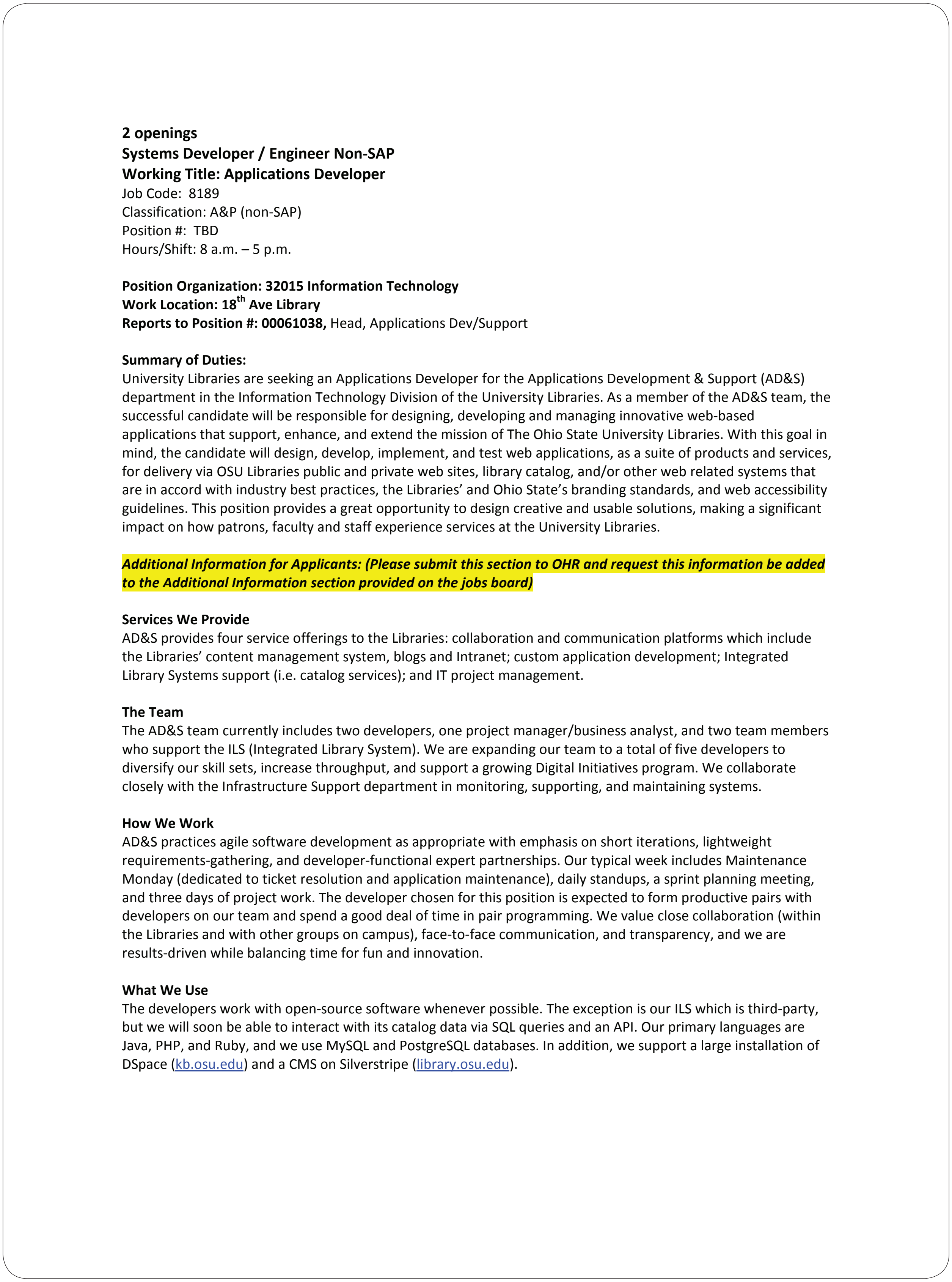
Preferred:

- Understanding of the elements of art and the principles of design
- Knowledge of HTML5 and CSS3
- Ability to design and implement responsive web pages and applications interfaces
- Experience with JavaScript visualization tools like D3, Raphaël, TimelineJS, etc.
- Experience with JavaScript mapping tools like Leaflet, OpenLayers, or the Google Maps API
- iOS or Android application development experience
- Experience designing HTTP interactions that employ the correct verbs and status codes
- Experience working on a team that employs the agile development process
- Strong analytical, reasoning and problem resolution skills and demonstrated success in applying technology to meet user needs
- Ability to handle interpersonal communications tactfully and accurately with a diverse community of users and vendors
- Ability to work independently and in close conjunction with others in a team-oriented setting
- Ability to learn and master new skills and technologies quickly
- Experience with Fedora Commons Repository, iRODS, LOCKSS, or other preservation system
- Experience utilizing Apache Solr or Lucene indexes

Requires Bachelor's Degree in Informatics, Information Science, or equivalent experience in end user applications design.

In addition to strong application development skills, we are also looking for someone with strong design instincts to create polished user interfaces for interacting with our unique resources within our digital repository.

OHIO STATE UNIVERSITY
Systems Developer/Engineer



2 openings
Systems Developer / Engineer Non-SAP
Working Title: Applications Developer
Job Code: 8189
Classification: A&P (non-SAP)
Position #: TBD
Hours/Shift: 8 a.m. – 5 p.m.

Position Organization: 32015 Information Technology
Work Location: 18th Ave Library
Reports to Position #: 00061038, Head, Applications Dev/Support

Summary of Duties:
University Libraries are seeking an Applications Developer for the Applications Development & Support (AD&S) department in the Information Technology Division of the University Libraries. As a member of the AD&S team, the successful candidate will be responsible for designing, developing and managing innovative web-based applications that support, enhance, and extend the mission of The Ohio State University Libraries. With this goal in mind, the candidate will design, develop, implement, and test web applications, as a suite of products and services, for delivery via OSU Libraries public and private web sites, library catalog, and/or other web related systems that are in accord with industry best practices, the Libraries' and Ohio State's branding standards, and web accessibility guidelines. This position provides a great opportunity to design creative and usable solutions, making a significant impact on how patrons, faculty and staff experience services at the University Libraries.

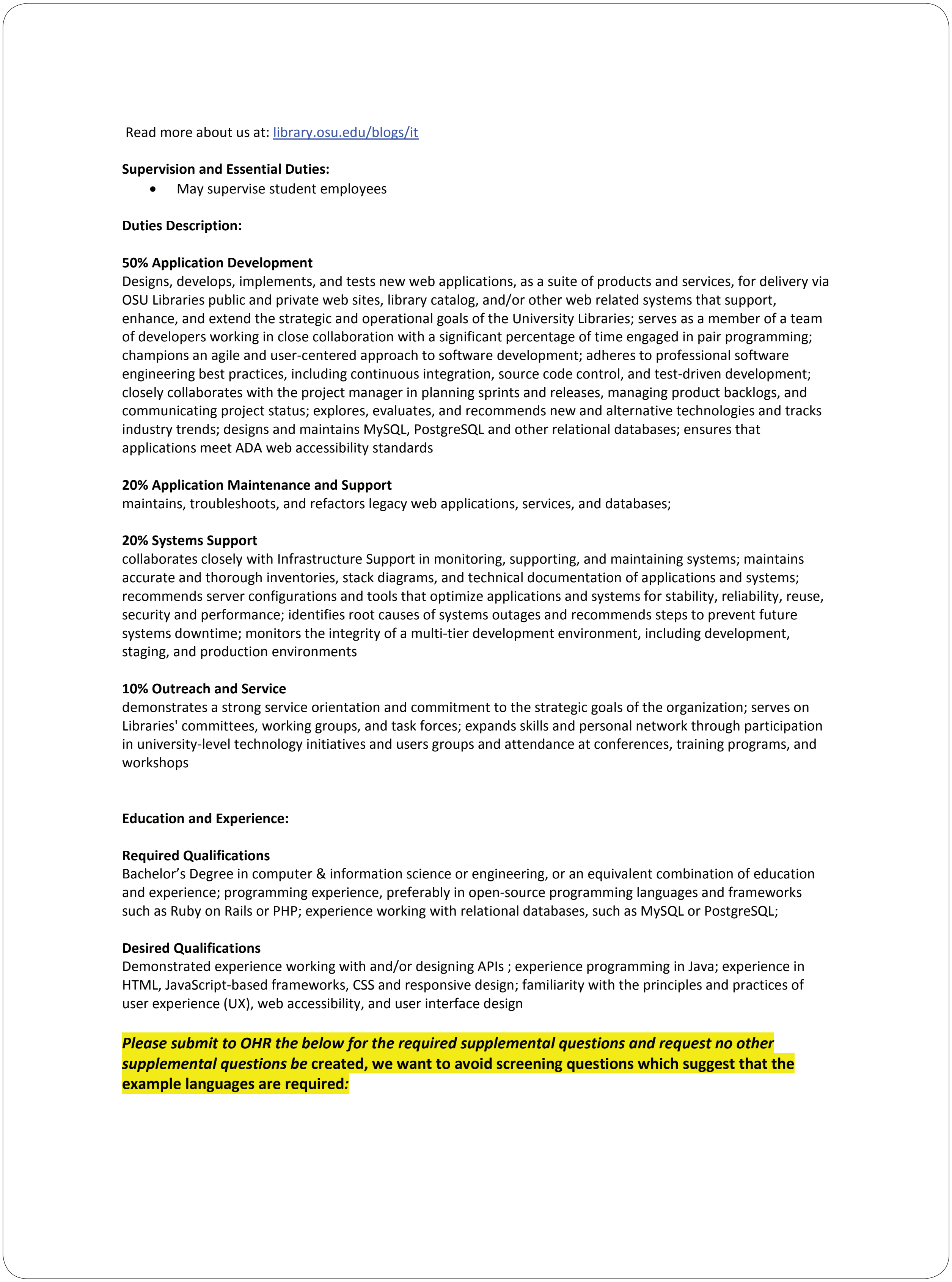
Additional Information for Applicants: (Please submit this section to OHR and request this information be added to the Additional Information section provided on the jobs board)

Services We Provide
AD&S provides four service offerings to the Libraries: collaboration and communication platforms which include the Libraries' content management system, blogs and Intranet; custom application development; Integrated Library Systems support (i.e. catalog services); and IT project management.

The Team
The AD&S team currently includes two developers, one project manager/business analyst, and two team members who support the ILS (Integrated Library System). We are expanding our team to a total of five developers to diversify our skill sets, increase throughput, and support a growing Digital Initiatives program. We collaborate closely with the Infrastructure Support department in monitoring, supporting, and maintaining systems.

How We Work
AD&S practices agile software development as appropriate with emphasis on short iterations, lightweight requirements-gathering, and developer-functional expert partnerships. Our typical week includes Maintenance Monday (dedicated to ticket resolution and application maintenance), daily standups, a sprint planning meeting, and three days of project work. The developer chosen for this position is expected to form productive pairs with developers on our team and spend a good deal of time in pair programming. We value close collaboration (within the Libraries and with other groups on campus), face-to-face communication, and transparency, and we are results-driven while balancing time for fun and innovation.

What We Use
The developers work with open-source software whenever possible. The exception is our ILS which is third-party, but we will soon be able to interact with its catalog data via SQL queries and an API. Our primary languages are Java, PHP, and Ruby, and we use MySQL and PostgreSQL databases. In addition, we support a large installation of DSpace (kb.osu.edu) and a CMS on Silverstripe (library.osu.edu).



Read more about us at: library.osu.edu/blogs/it

Supervision and Essential Duties:

- May supervise student employees

Duties Description:

50% Application Development
Designs, develops, implements, and tests new web applications, as a suite of products and services, for delivery via OSU Libraries public and private web sites, library catalog, and/or other web related systems that support, enhance, and extend the strategic and operational goals of the University Libraries; serves as a member of a team of developers working in close collaboration with a significant percentage of time engaged in pair programming; champions an agile and user-centered approach to software development; adheres to professional software engineering best practices, including continuous integration, source code control, and test-driven development; closely collaborates with the project manager in planning sprints and releases, managing product backlogs, and communicating project status; explores, evaluates, and recommends new and alternative technologies and tracks industry trends; designs and maintains MySQL, PostgreSQL and other relational databases; ensures that applications meet ADA web accessibility standards

20% Application Maintenance and Support
maintains, troubleshoots, and refactors legacy web applications, services, and databases;

20% Systems Support
collaborates closely with Infrastructure Support in monitoring, supporting, and maintaining systems; maintains accurate and thorough inventories, stack diagrams, and technical documentation of applications and systems; recommends server configurations and tools that optimize applications and systems for stability, reliability, reuse, security and performance; identifies root causes of systems outages and recommends steps to prevent future systems downtime; monitors the integrity of a multi-tier development environment, including development, staging, and production environments

10% Outreach and Service
demonstrates a strong service orientation and commitment to the strategic goals of the organization; serves on Libraries' committees, working groups, and task forces; expands skills and personal network through participation in university-level technology initiatives and users groups and attendance at conferences, training programs, and workshops

Education and Experience:

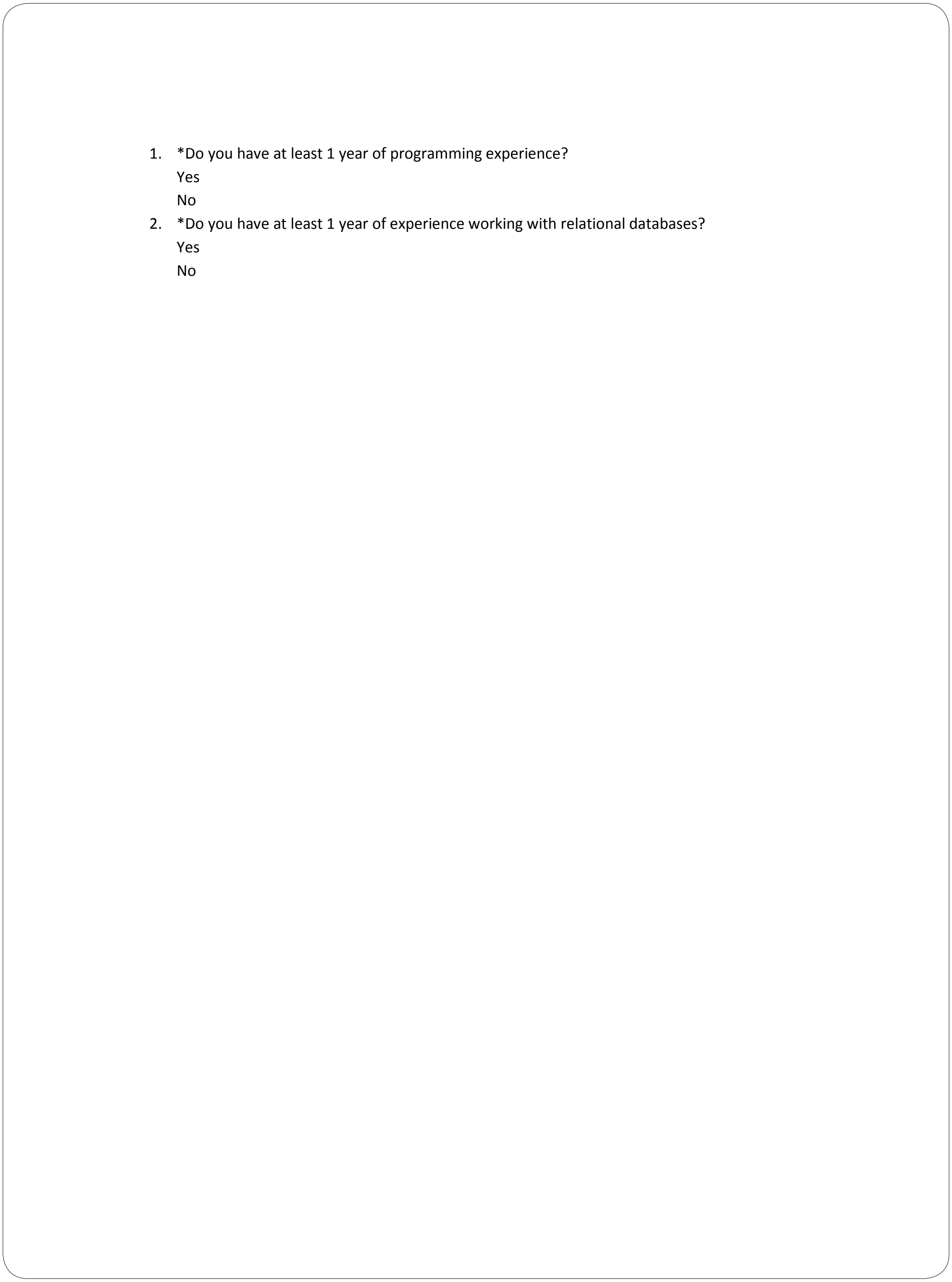
Required Qualifications
Bachelor’s Degree in computer & information science or engineering, or an equivalent combination of education and experience; programming experience, preferably in open-source programming languages and frameworks such as Ruby on Rails or PHP; experience working with relational databases, such as MySQL or PostgreSQL;

Desired Qualifications
Demonstrated experience working with and/or designing APIs ; experience programming in Java; experience in HTML, JavaScript-based frameworks, CSS and responsive design; familiarity with the principles and practices of user experience (UX), web accessibility, and user interface design

Please submit to OHR the below for the required supplemental questions and request no other supplemental questions be **created, we want to avoid screening questions which suggest that the example languages are required*****:***

OHIO STATE UNIVERSITY

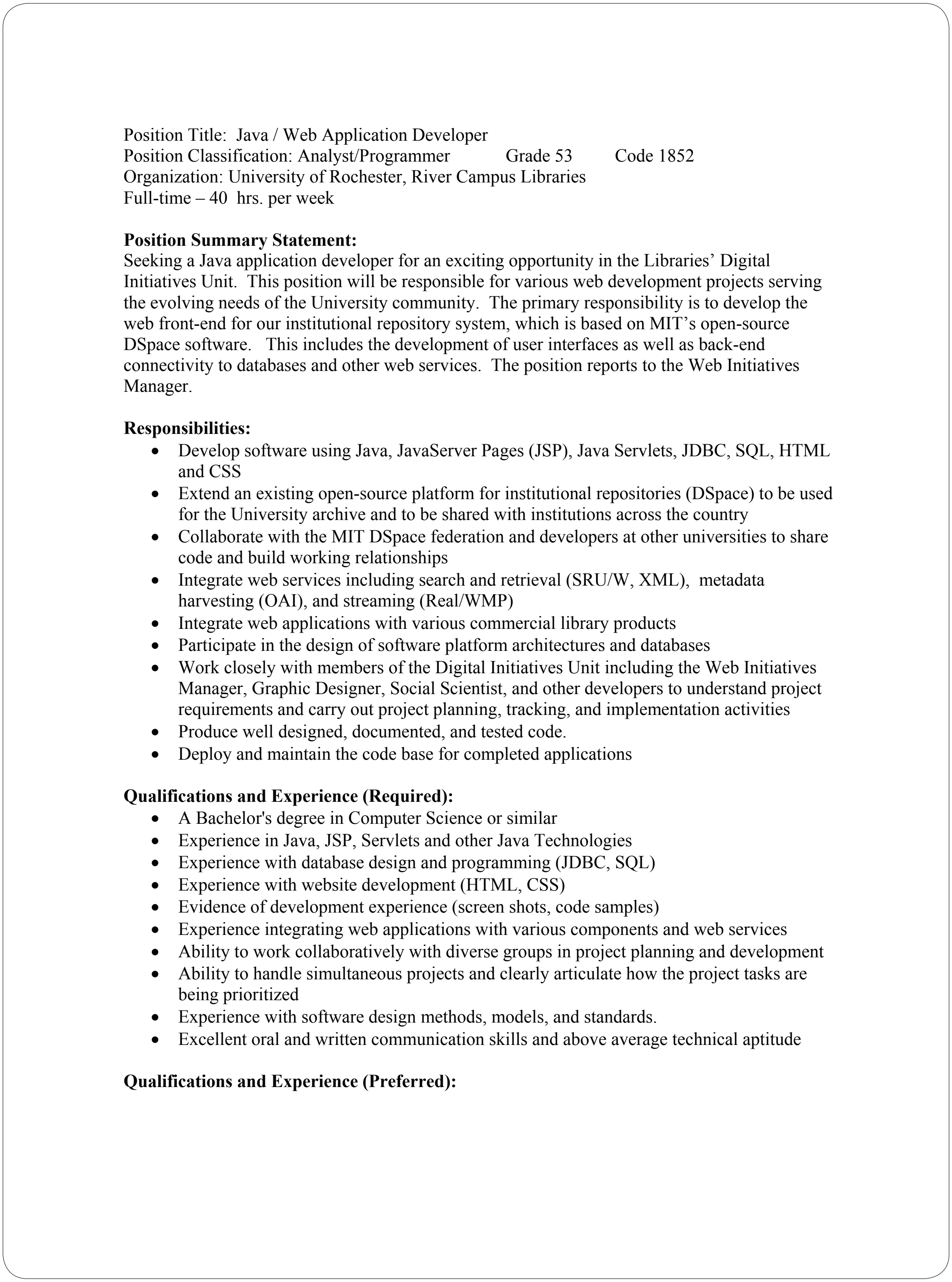
Systems Developer/Engineer



1. *Do you have at least 1 year of programming experience?
   Yes
   No
2. *Do you have at least 1 year of experience working with relational databases?
   Yes
   No

UNIVERSITY OF ROCHESTER

Java/Web Application Developer



Position Title: Java / Web Application Developer
Position Classification: Analyst/Programmer Grade 53 Code 1852
Organization: University of Rochester, River Campus Libraries
Full-time – 40 hrs. per week

Position Summary Statement:
Seeking a Java application developer for an exciting opportunity in the Libraries' Digital Initiatives Unit. This position will be responsible for various web development projects serving the evolving needs of the University community. The primary responsibility is to develop the web front-end for our institutional repository system, which is based on MIT's open-source DSpace software. This includes the development of user interfaces as well as back-end connectivity to databases and other web services. The position reports to the Web Initiatives Manager.

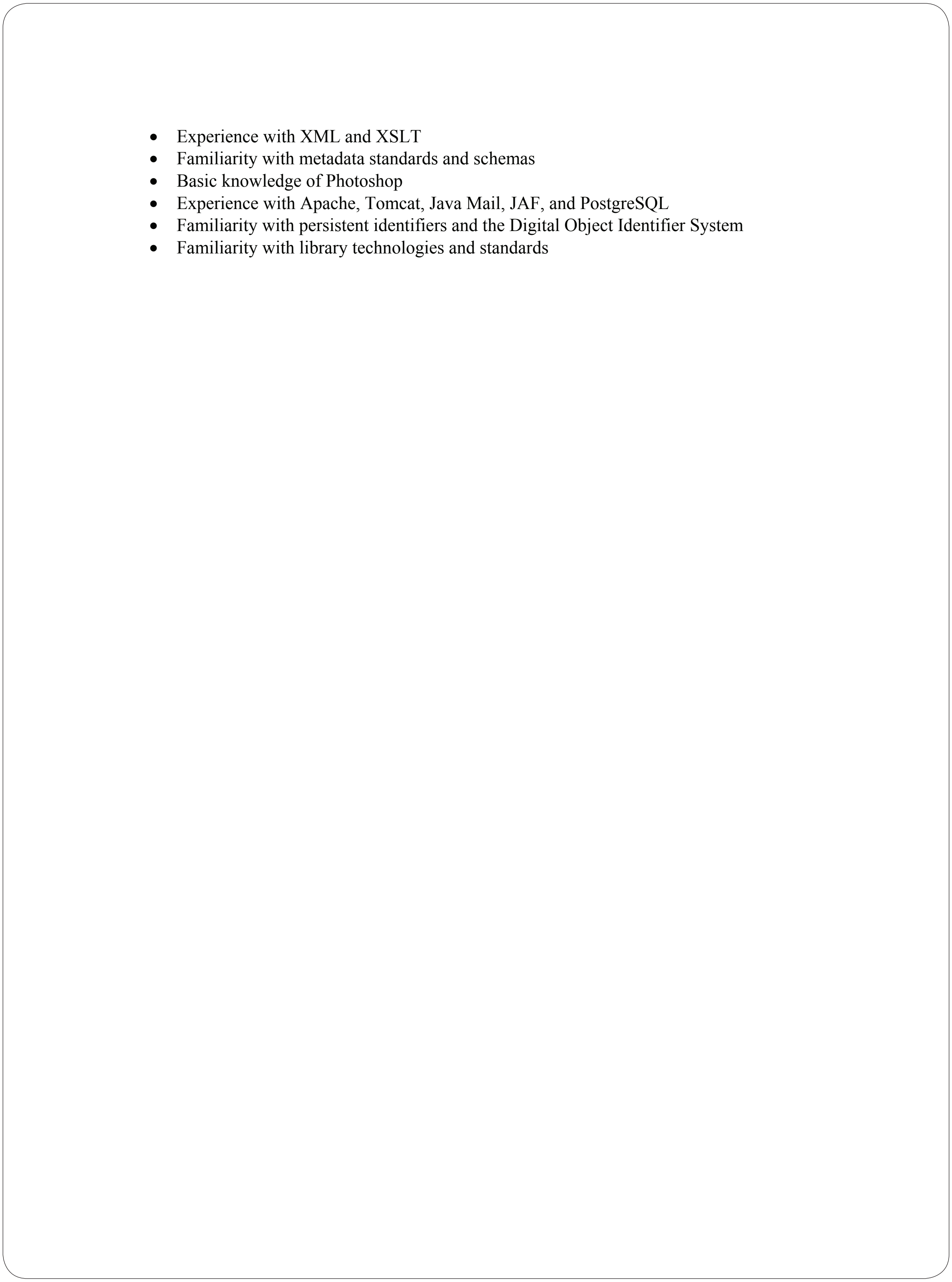
Responsibilities:

- Develop software using Java, JavaServer Pages (JSP), Java Servlets, JDBC, SQL, HTML and CSS
- Extend an existing open-source platform for institutional repositories (DSpace) to be used for the University archive and to be shared with institutions across the country
- Collaborate with the MIT DSpace federation and developers at other universities to share code and build working relationships
- Integrate web services including search and retrieval (SRU/W, XML), metadata harvesting (OAI), and streaming (Real/WMP)
- Integrate web applications with various commercial library products
- Participate in the design of software platform architectures and databases
- Work closely with members of the Digital Initiatives Unit including the Web Initiatives Manager, Graphic Designer, Social Scientist, and other developers to understand project requirements and carry out project planning, tracking, and implementation activities
- Produce well designed, documented, and tested code.
- Deploy and maintain the code base for completed applications

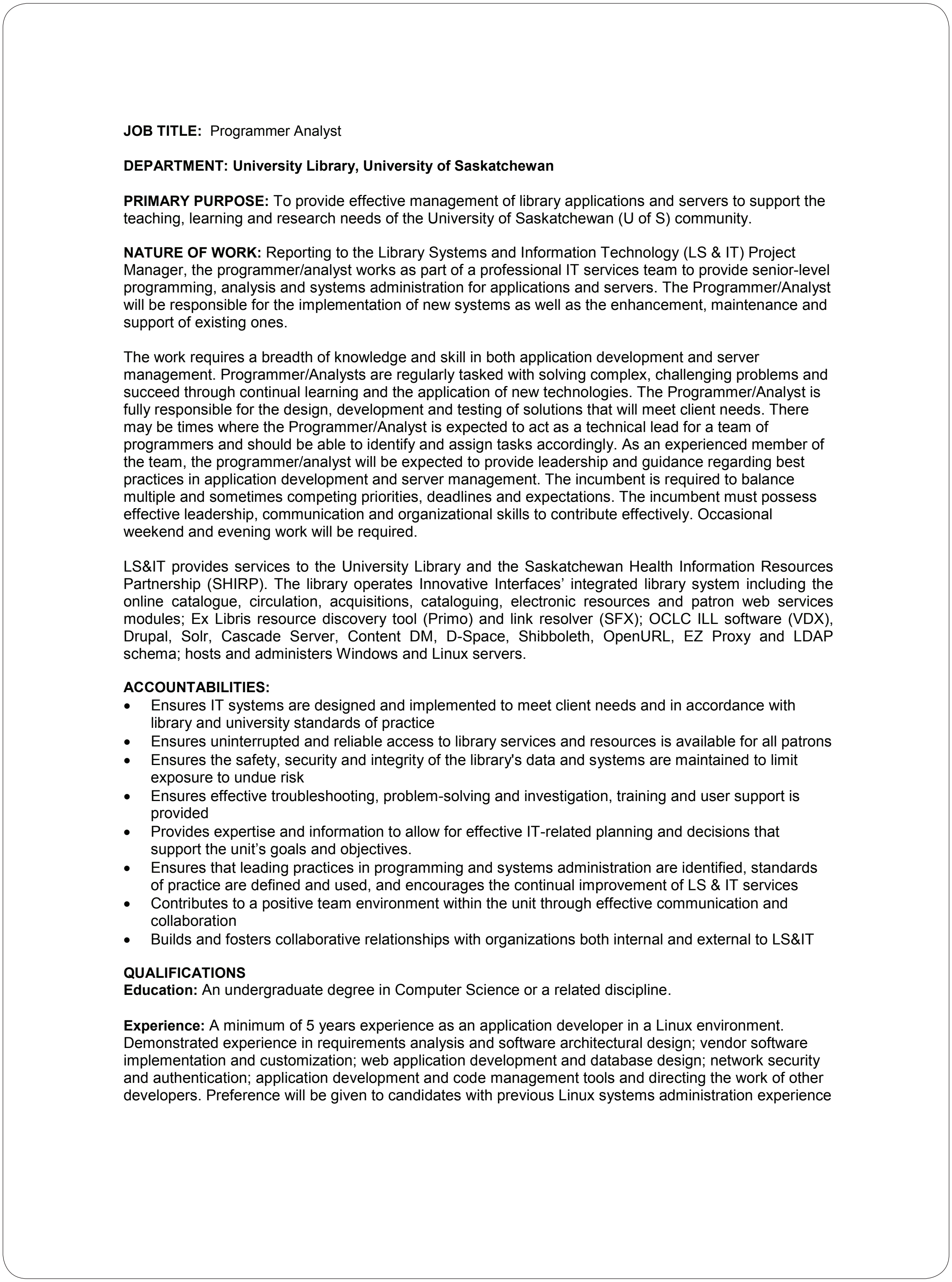
Qualifications and Experience (Required):

- A Bachelor's degree in Computer Science or similar
- Experience in Java, JSP, Servlets and other Java Technologies
- Experience with database design and programming (JDBC, SQL)
- Experience with website development (HTML, CSS)
- Evidence of development experience (screen shots, code samples)
- Experience integrating web applications with various components and web services
- Ability to work collaboratively with diverse groups in project planning and development
- Ability to handle simultaneous projects and clearly articulate how the project tasks are being prioritized
- Experience with software design methods, models, and standards.
- Excellent oral and written communication skills and above average technical aptitude

Qualifications and Experience (Preferred):



- Experience with XML and XSLT
- Familiarity with metadata standards and schemas
- Basic knowledge of Photoshop
- Experience with Apache, Tomcat, Java Mail, JAF, and PostgreSQL
- Familiarity with persistent identifiers and the Digital Object Identifier System
- Familiarity with library technologies and standards



JOB TITLE: Programmer Analyst

DEPARTMENT: University Library, University of Saskatchewan

PRIMARY PURPOSE: To provide effective management of library applications and servers to support the teaching, learning and research needs of the University of Saskatchewan (U of S) community.

NATURE OF WORK: Reporting to the Library Systems and Information Technology (LS & IT) Project Manager, the programmer/analyst works as part of a professional IT services team to provide senior-level programming, analysis and systems administration for applications and servers. The Programmer/Analyst will be responsible for the implementation of new systems as well as the enhancement, maintenance and support of existing ones.

The work requires a breadth of knowledge and skill in both application development and server management. Programmer/Analysts are regularly tasked with solving complex, challenging problems and succeed through continual learning and the application of new technologies. The Programmer/Analyst is fully responsible for the design, development and testing of solutions that will meet client needs. There may be times where the Programmer/Analyst is expected to act as a technical lead for a team of programmers and should be able to identify and assign tasks accordingly. As an experienced member of the team, the programmer/analyst will be expected to provide leadership and guidance regarding best practices in application development and server management. The incumbent is required to balance multiple and sometimes competing priorities, deadlines and expectations. The incumbent must possess effective leadership, communication and organizational skills to contribute effectively. Occasional weekend and evening work will be required.

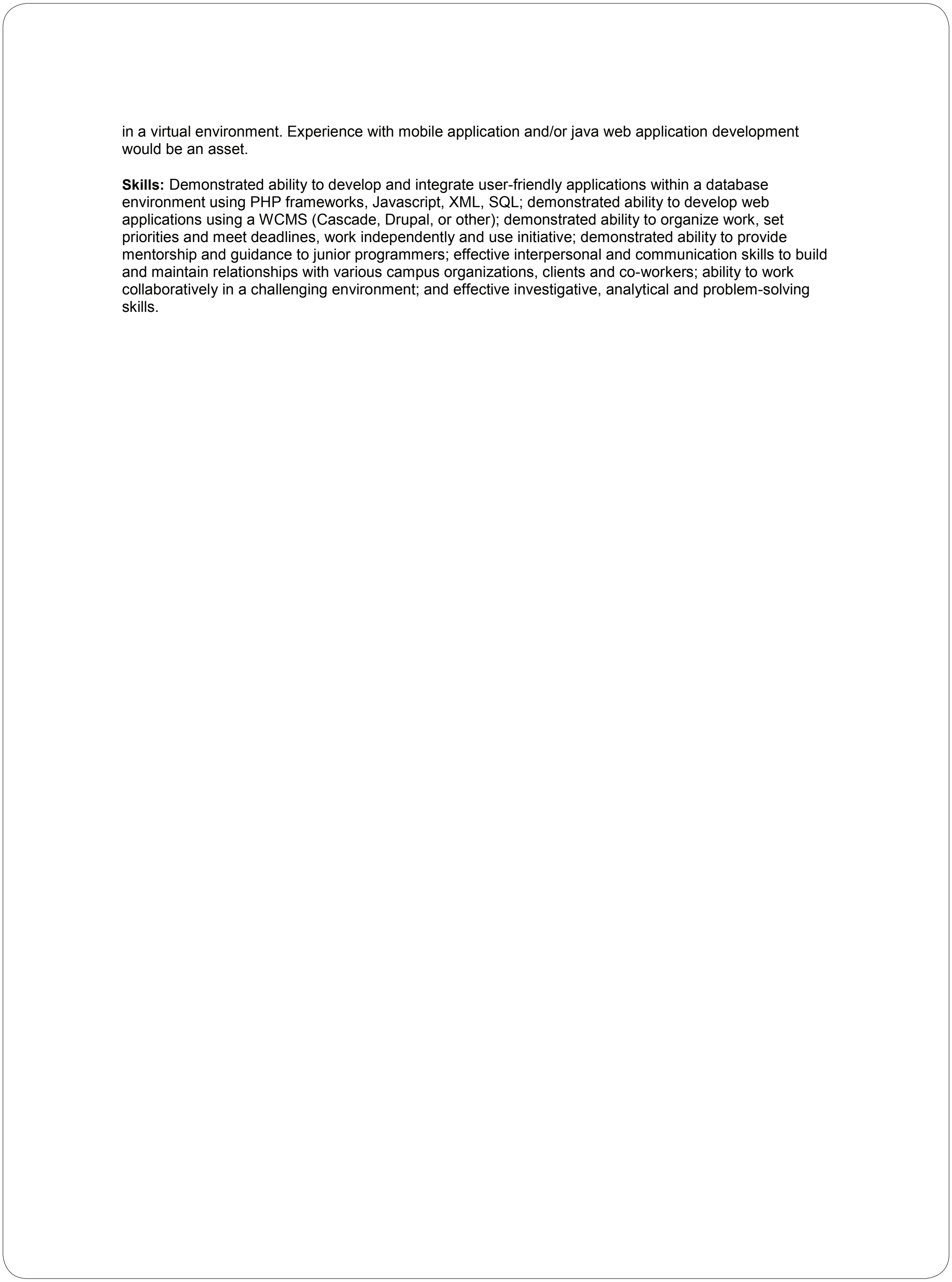
LS&IT provides services to the University Library and the Saskatchewan Health Information Resources Partnership (SHIRP). The library operates Innovative Interfaces' integrated library system including the online catalogue, circulation, acquisitions, cataloguing, electronic resources and patron web services modules; Ex Libris resource discovery tool (Primo) and link resolver (SFX); OCLC ILL software (VDX), Drupal, Solr, Cascade Server, Content DM, D-Space, Shibboleth, OpenURL, EZ Proxy and LDAP schema; hosts and administers Windows and Linux servers.

ACCOUNTABILITIES:

- Ensures IT systems are designed and implemented to meet client needs and in accordance with library and university standards of practice
- Ensures uninterrupted and reliable access to library services and resources is available for all patrons
- Ensures the safety, security and integrity of the library's data and systems are maintained to limit exposure to undue risk
- Ensures effective troubleshooting, problem-solving and investigation, training and user support is provided
- Provides expertise and information to allow for effective IT-related planning and decisions that support the unit's goals and objectives.
- Ensures that leading practices in programming and systems administration are identified, standards of practice are defined and used, and encourages the continual improvement of LS & IT services
- Contributes to a positive team environment within the unit through effective communication and collaboration
- Builds and fosters collaborative relationships with organizations both internal and external to LS&IT

QUALIFICATIONS
Education: An undergraduate degree in Computer Science or a related discipline.

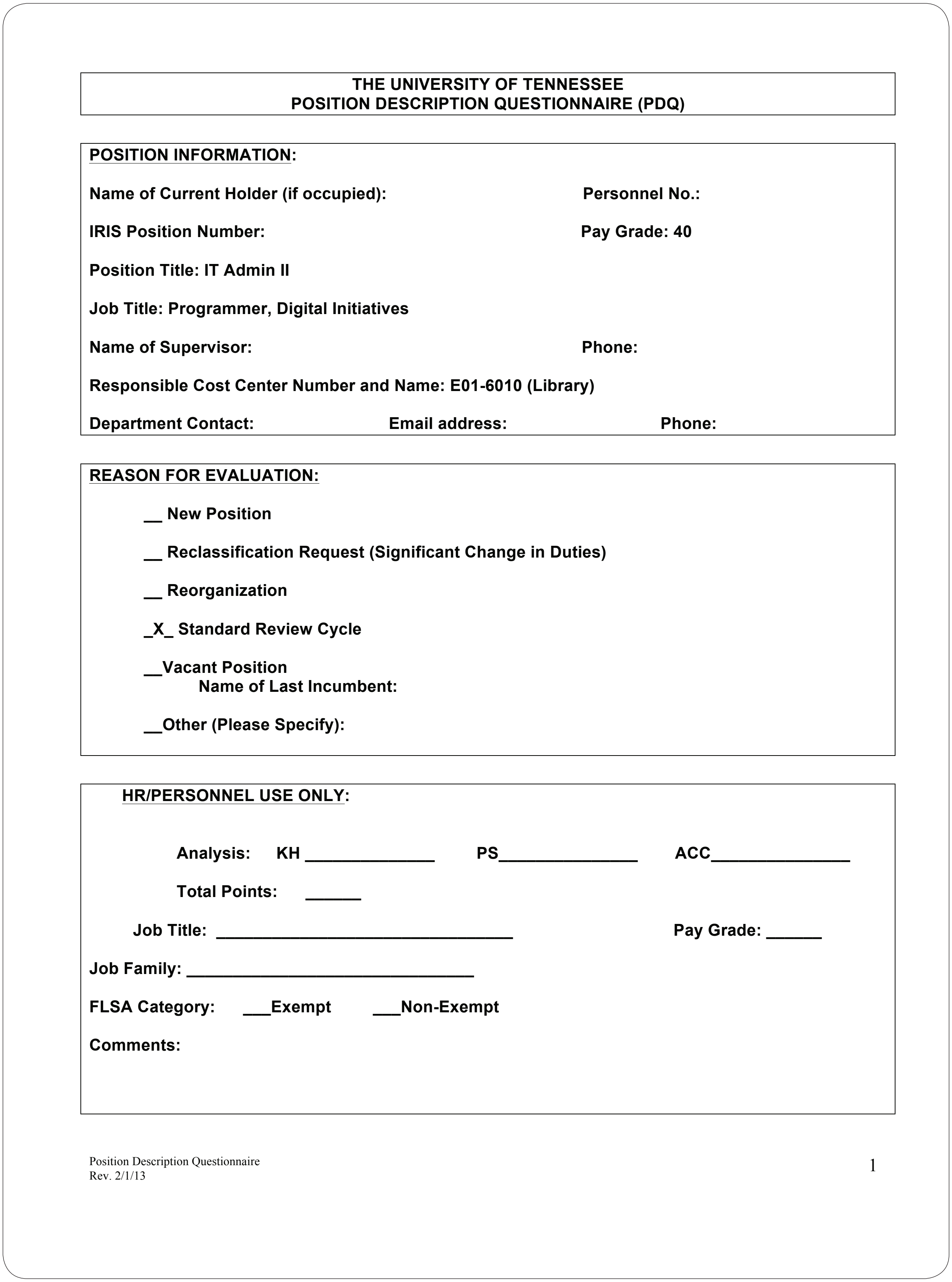
Experience: A minimum of 5 years experience as an application developer in a Linux environment. Demonstrated experience in requirements analysis and software architectural design; vendor software implementation and customization; web application development and database design; network security and authentication; application development and code management tools and directing the work of other developers. Preference will be given to candidates with previous Linux systems administration experience



in a virtual environment. Experience with mobile application and/or java web application development would be an asset.

Skills: Demonstrated ability to develop and integrate user-friendly applications within a database environment using PHP frameworks, Javascript, XML, SQL; demonstrated ability to develop web applications using a WCMS (Cascade, Drupal, or other); demonstrated ability to organize work, set priorities and meet deadlines, work independently and use initiative; demonstrated ability to provide mentorship and guidance to junior programmers; effective interpersonal and communication skills to build and maintain relationships with various campus organizations, clients and co-workers; ability to work collaboratively in a challenging environment; and effective investigative, analytical and problem-solving skills.

UNIVERSITY OF TENNESSEE
Digital Initiatives Programmer 1



THE UNIVERSITY OF TENNESSEE
POSITION DESCRIPTION QUESTIONNAIRE (PDQ)

POSITION INFORMATION:

Name of Current Holder (if occupied): **Personnel No.:**

IRIS Position Number: **Pay Grade: 40**

Position Title: IT Admin II

Job Title: Programmer, Digital Initiatives

Name of Supervisor: **Phone:**

Responsible Cost Center Number and Name: E01-6010 (Library)

Department Contact: **Email address:** **Phone:**

REASON FOR EVALUATION:

__ New Position

__ Reclassification Request (Significant Change in Duties)

__ Reorganization

_X_ Standard Review Cycle

__Vacant Position
Name of Last Incumbent:

__Other (Please Specify):

HR/PERSONNEL USE ONLY:

Analysis: KH ______________ PS______________ ACC______________

Total Points: ______

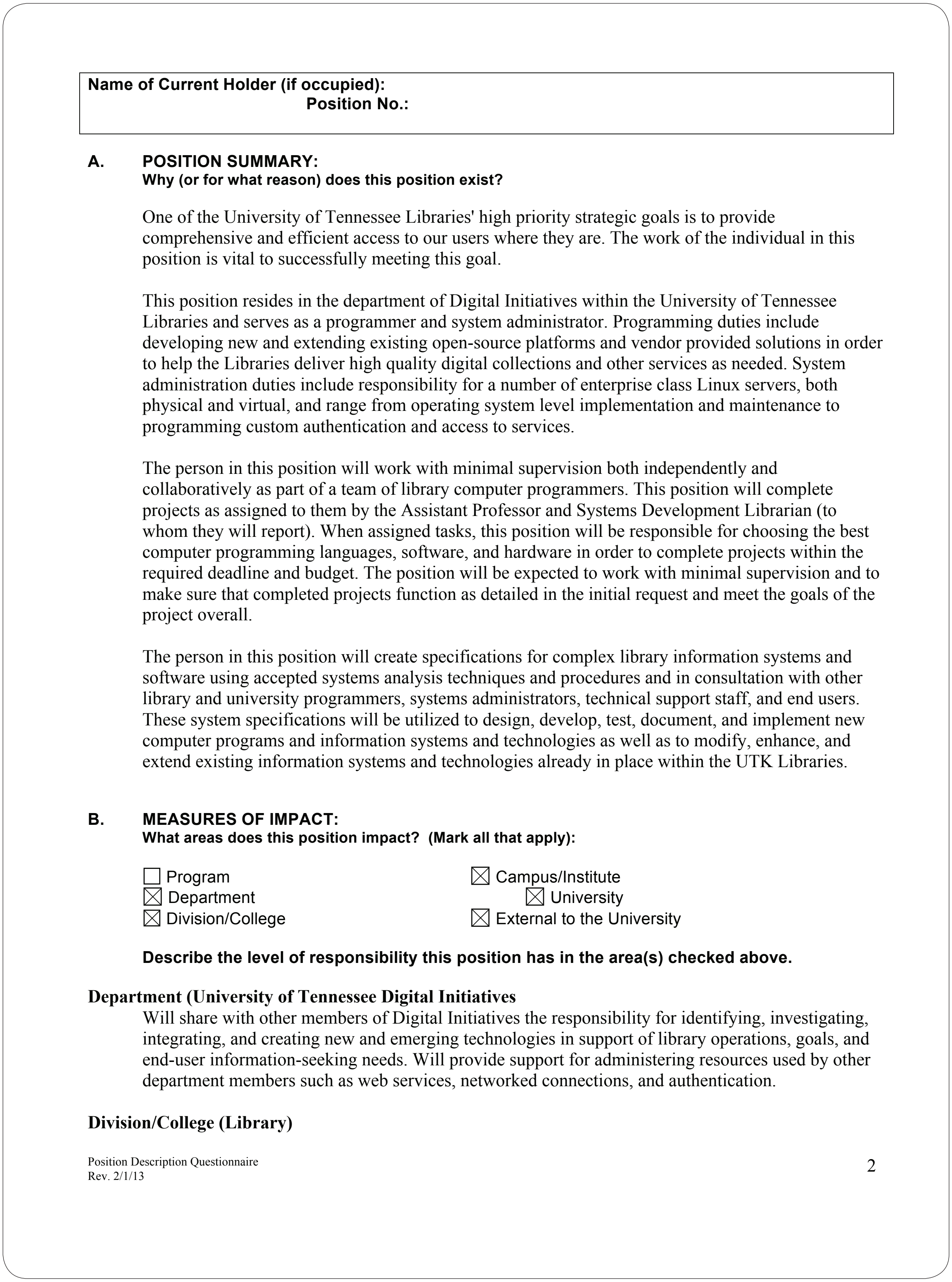
Job Title: ________________________________ Pay Grade: ______

Job Family: _______________________________

FLSA Category: ___Exempt ___Non-Exempt

Comments:

UNIVERSITY OF TENNESSEE
Digital Initiatives Programmer 1



Name of Current Holder (if occupied): **Position No.:**	

A. POSITION SUMMARY:
Why (or for what reason) does this position exist?

One of the University of Tennessee Libraries' high priority strategic goals is to provide comprehensive and efficient access to our users where they are. The work of the individual in this position is vital to successfully meeting this goal.

This position resides in the department of Digital Initiatives within the University of Tennessee Libraries and serves as a programmer and system administrator. Programming duties include developing new and extending existing open-source platforms and vendor provided solutions in order to help the Libraries deliver high quality digital collections and other services as needed. System administration duties include responsibility for a number of enterprise class Linux servers, both physical and virtual, and range from operating system level implementation and maintenance to programming custom authentication and access to services.

The person in this position will work with minimal supervision both independently and collaboratively as part of a team of library computer programmers. This position will complete projects as assigned to them by the Assistant Professor and Systems Development Librarian (to whom they will report). When assigned tasks, this position will be responsible for choosing the best computer programming languages, software, and hardware in order to complete projects within the required deadline and budget. The position will be expected to work with minimal supervision and to make sure that completed projects function as detailed in the initial request and meet the goals of the project overall.

The person in this position will create specifications for complex library information systems and software using accepted systems analysis techniques and procedures and in consultation with other library and university programmers, systems administrators, technical support staff, and end users. These system specifications will be utilized to design, develop, test, document, and implement new computer programs and information systems and technologies as well as to modify, enhance, and extend existing information systems and technologies already in place within the UTK Libraries.

B. MEASURES OF IMPACT:
What areas does this position impact? (Mark all that apply):

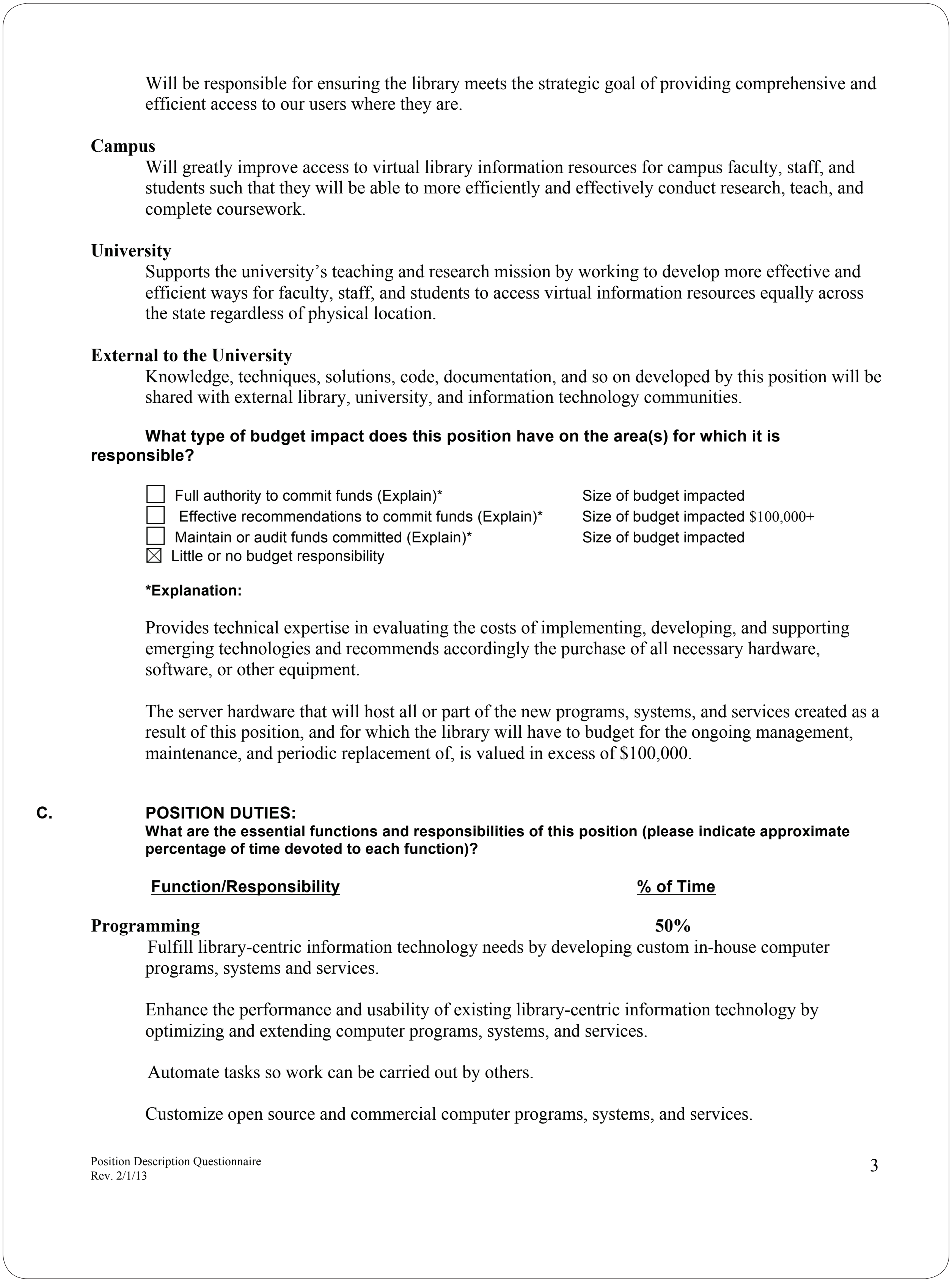
☐ Program
☒ Department
☒ Division/College
☒ Campus/Institute
☒ University
☒ External to the University

Describe the level of responsibility this position has in the area(s) checked above.

Department (University of Tennessee Digital Initiatives

Will share with other members of Digital Initiatives the responsibility for identifying, investigating, integrating, and creating new and emerging technologies in support of library operations, goals, and end-user information-seeking needs. Will provide support for administering resources used by other department members such as web services, networked connections, and authentication.

Division/College (Library)



Will be responsible for ensuring the library meets the strategic goal of providing comprehensive and efficient access to our users where they are.

Campus

Will greatly improve access to virtual library information resources for campus faculty, staff, and students such that they will be able to more efficiently and effectively conduct research, teach, and complete coursework.

University

Supports the university's teaching and research mission by working to develop more effective and efficient ways for faculty, staff, and students to access virtual information resources equally across the state regardless of physical location.

External to the University

Knowledge, techniques, solutions, code, documentation, and so on developed by this position will be shared with external library, university, and information technology communities.

What type of budget impact does this position have on the area(s) for which it is responsible?

□ Full authority to commit funds (Explain)*	Size of budget impacted
□ Effective recommendations to commit funds (Explain)*	Size of budget impacted $100,000+
□ Maintain or audit funds committed (Explain)*	Size of budget impacted
☒ Little or no budget responsibility	

***Explanation:**

Provides technical expertise in evaluating the costs of implementing, developing, and supporting emerging technologies and recommends accordingly the purchase of all necessary hardware, software, or other equipment.

The server hardware that will host all or part of the new programs, systems, and services created as a result of this position, and for which the library will have to budget for the ongoing management, maintenance, and periodic replacement of, is valued in excess of $100,000.

C. POSITION DUTIES:

What are the essential functions and responsibilities of this position (please indicate approximate percentage of time devoted to each function)?

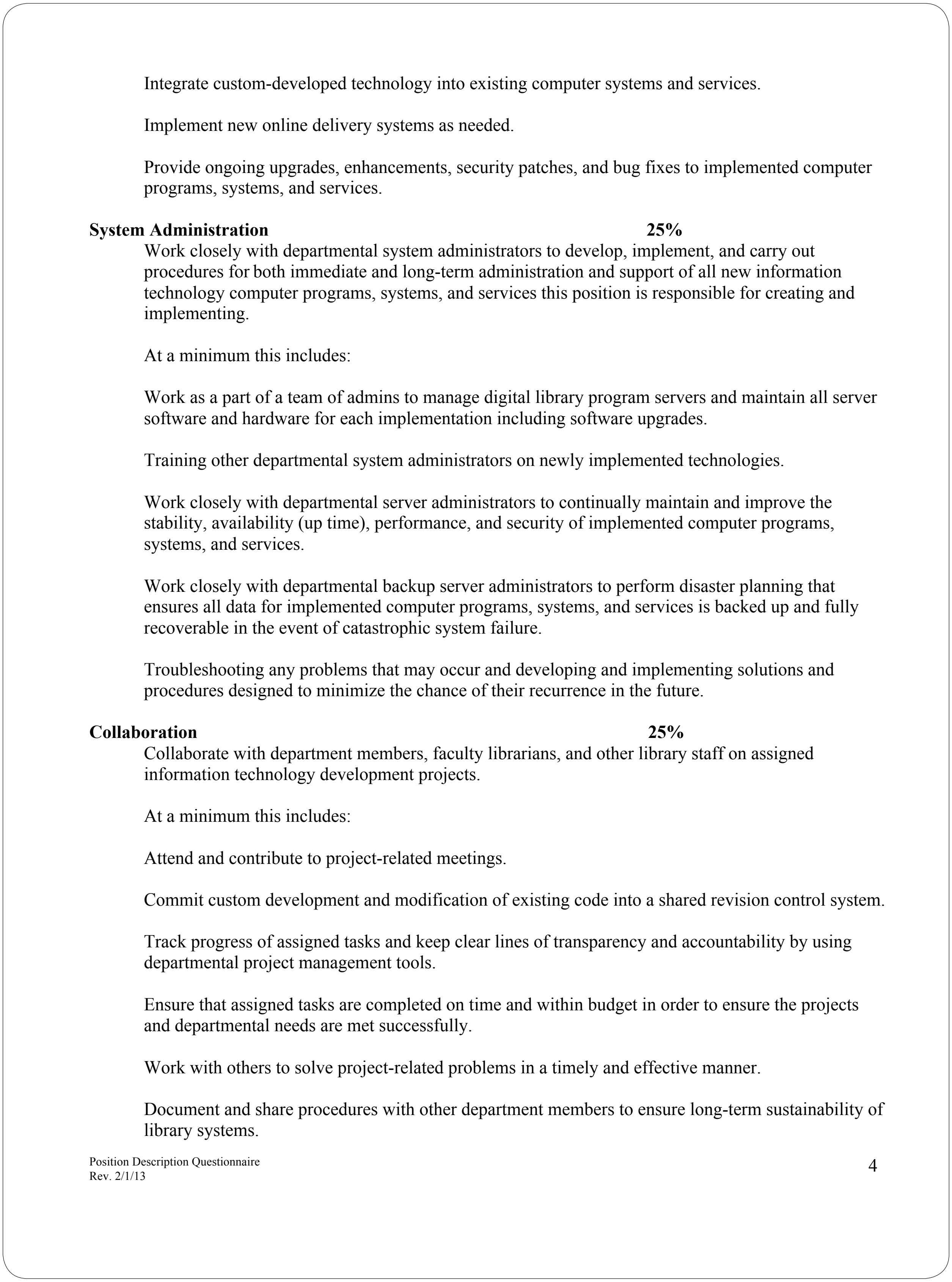
Function/Responsibility	% of Time
Programming	**50%**

Fulfill library-centric information technology needs by developing custom in-house computer programs, systems and services.

Enhance the performance and usability of existing library-centric information technology by optimizing and extending computer programs, systems, and services.

Automate tasks so work can be carried out by others.

Customize open source and commercial computer programs, systems, and services.



Integrate custom-developed technology into existing computer systems and services.

Implement new online delivery systems as needed.

Provide ongoing upgrades, enhancements, security patches, and bug fixes to implemented computer programs, systems, and services.

System Administration **25%**

Work closely with departmental system administrators to develop, implement, and carry out procedures for both immediate and long-term administration and support of all new information technology computer programs, systems, and services this position is responsible for creating and implementing.

At a minimum this includes:

Work as a part of a team of admins to manage digital library program servers and maintain all server software and hardware for each implementation including software upgrades.

Training other departmental system administrators on newly implemented technologies.

Work closely with departmental server administrators to continually maintain and improve the stability, availability (up time), performance, and security of implemented computer programs, systems, and services.

Work closely with departmental backup server administrators to perform disaster planning that ensures all data for implemented computer programs, systems, and services is backed up and fully recoverable in the event of catastrophic system failure.

Troubleshooting any problems that may occur and developing and implementing solutions and procedures designed to minimize the chance of their recurrence in the future.

Collaboration **25%**

Collaborate with department members, faculty librarians, and other library staff on assigned information technology development projects.

At a minimum this includes:

Attend and contribute to project-related meetings.

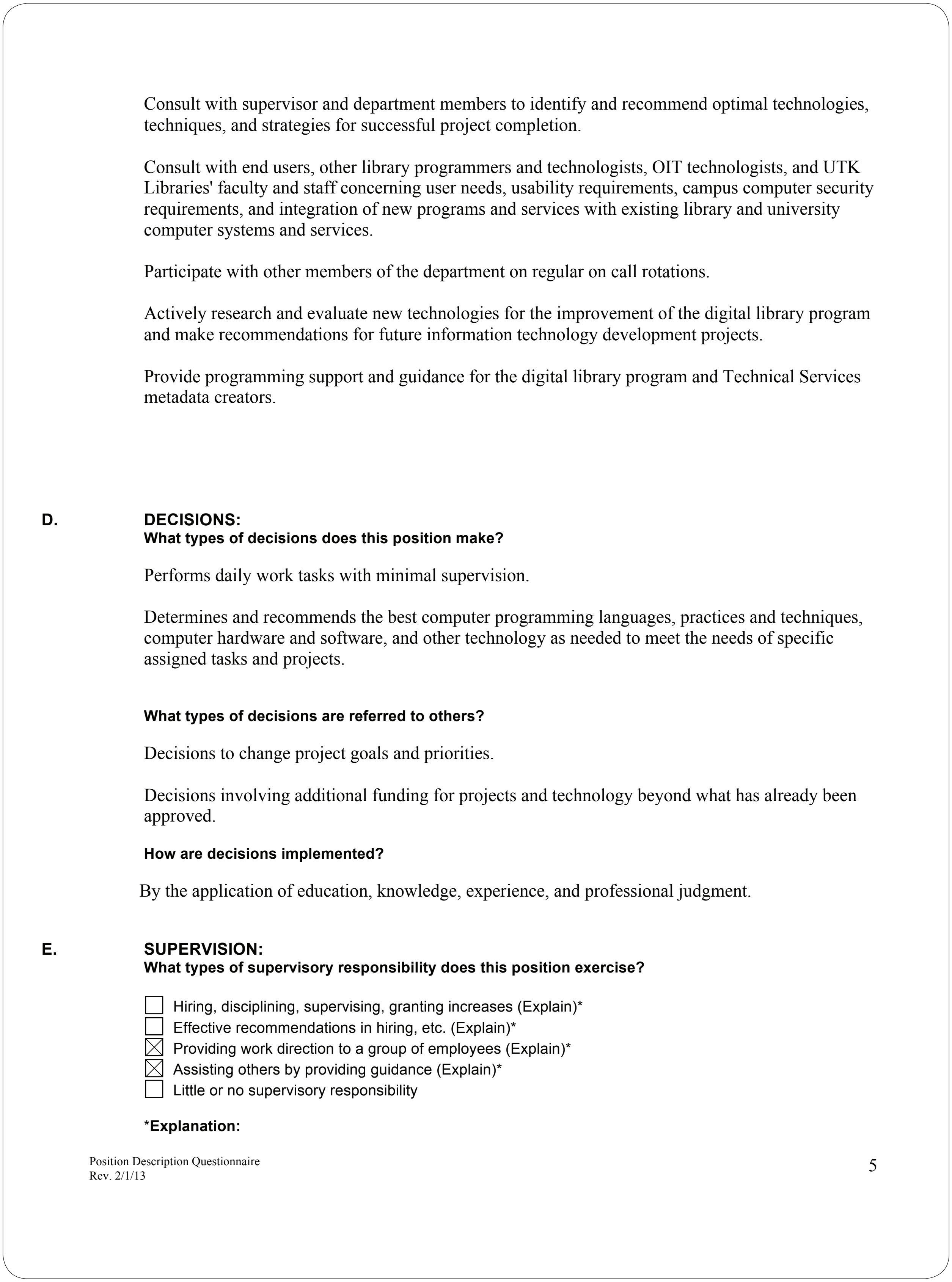
Commit custom development and modification of existing code into a shared revision control system.

Track progress of assigned tasks and keep clear lines of transparency and accountability by using departmental project management tools.

Ensure that assigned tasks are completed on time and within budget in order to ensure the projects and departmental needs are met successfully.

Work with others to solve project-related problems in a timely and effective manner.

Document and share procedures with other department members to ensure long-term sustainability of library systems.



Consult with supervisor and department members to identify and recommend optimal technologies, techniques, and strategies for successful project completion.

Consult with end users, other library programmers and technologists, OIT technologists, and UTK Libraries' faculty and staff concerning user needs, usability requirements, campus computer security requirements, and integration of new programs and services with existing library and university computer systems and services.

Participate with other members of the department on regular on call rotations.

Actively research and evaluate new technologies for the improvement of the digital library program and make recommendations for future information technology development projects.

Provide programming support and guidance for the digital library program and Technical Services metadata creators.

D. DECISIONS:
What types of decisions does this position make?

Performs daily work tasks with minimal supervision.

Determines and recommends the best computer programming languages, practices and techniques, computer hardware and software, and other technology as needed to meet the needs of specific assigned tasks and projects.

What types of decisions are referred to others?

Decisions to change project goals and priorities.

Decisions involving additional funding for projects and technology beyond what has already been approved.

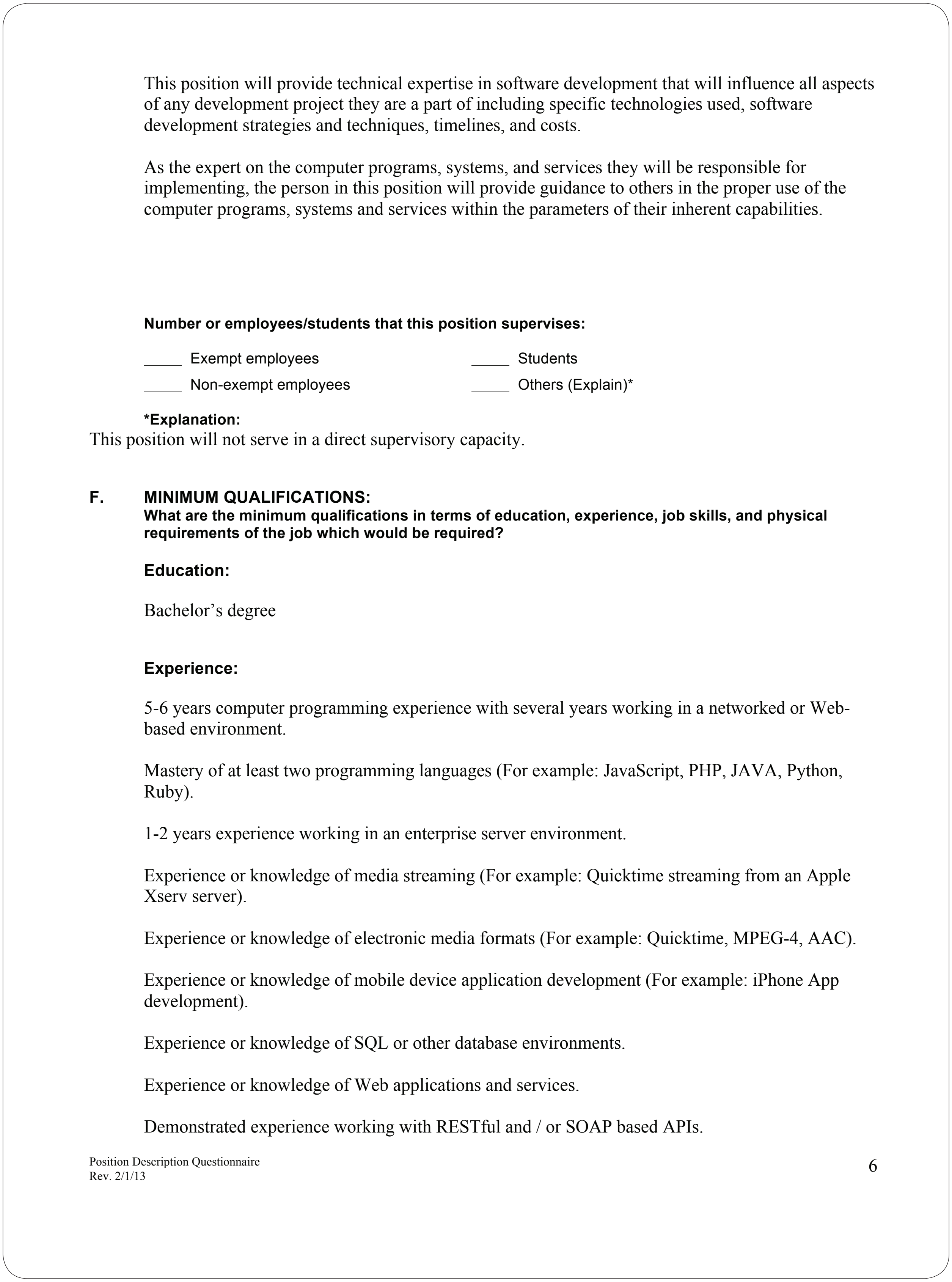
How are decisions implemented?

By the application of education, knowledge, experience, and professional judgment.

E. SUPERVISION:
What types of supervisory responsibility does this position exercise?

☐ Hiring, disciplining, supervising, granting increases (Explain)*
☐ Effective recommendations in hiring, etc. (Explain)*
☒ Providing work direction to a group of employees (Explain)*
☒ Assisting others by providing guidance (Explain)*
☐ Little or no supervisory responsibility

***Explanation:**



This position will provide technical expertise in software development that will influence all aspects of any development project they are a part of including specific technologies used, software development strategies and techniques, timelines, and costs.

As the expert on the computer programs, systems, and services they will be responsible for implementing, the person in this position will provide guidance to others in the proper use of the computer programs, systems and services within the parameters of their inherent capabilities.

Number or employees/students that this position supervises:

_____ Exempt employees _____ Students

_____ Non-exempt employees _____ Others (Explain)*

***Explanation:**

This position will not serve in a direct supervisory capacity.

F. MINIMUM QUALIFICATIONS:

What are the minimum qualifications in terms of education, experience, job skills, and physical requirements of the job which would be required?

Education:

Bachelor's degree

Experience:

5-6 years computer programming experience with several years working in a networked or Web-based environment.

Mastery of at least two programming languages (For example: JavaScript, PHP, JAVA, Python, Ruby).

1-2 years experience working in an enterprise server environment.

Experience or knowledge of media streaming (For example: Quicktime streaming from an Apple Xserv server).

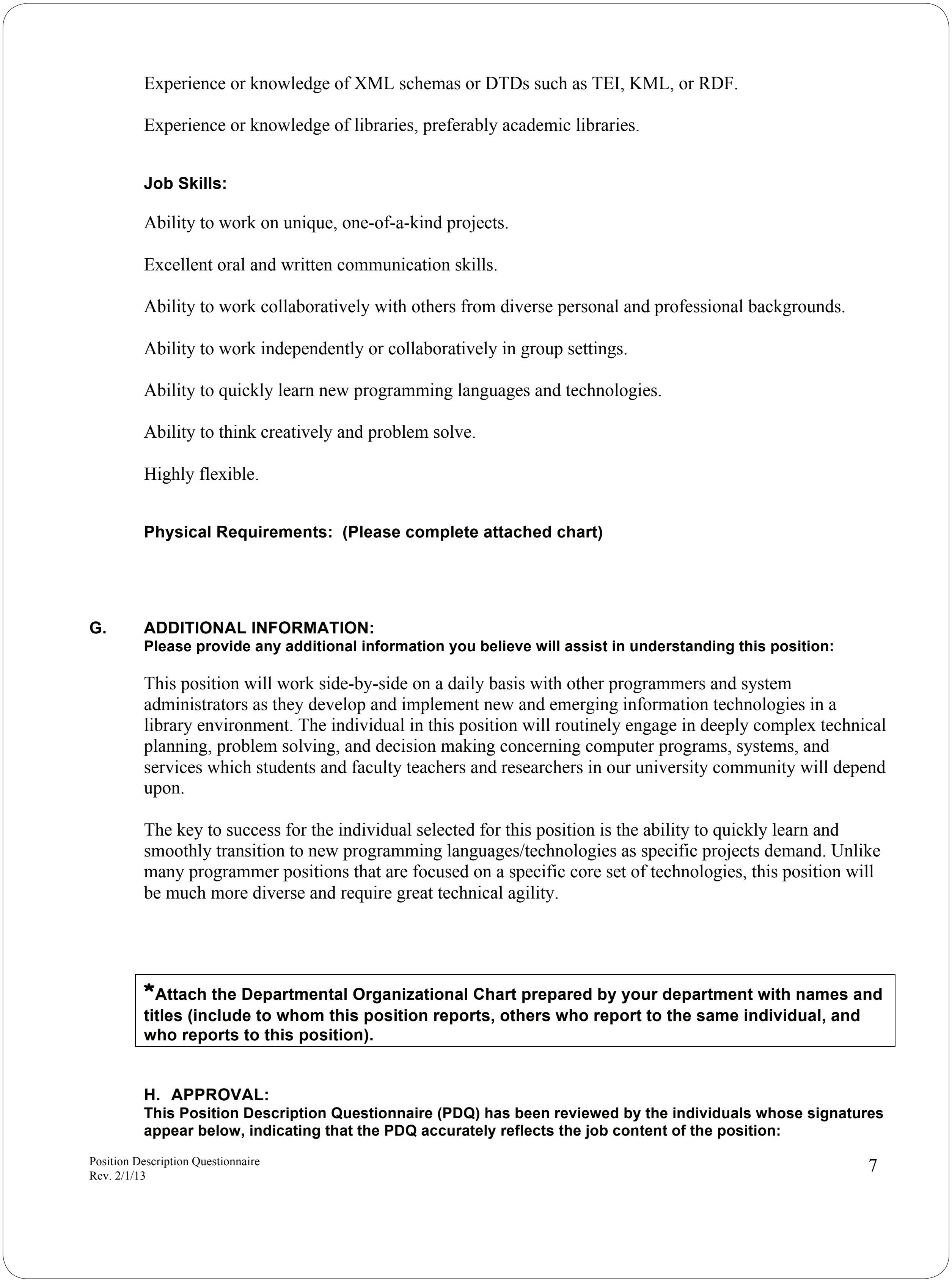
Experience or knowledge of electronic media formats (For example: Quicktime, MPEG-4, AAC).

Experience or knowledge of mobile device application development (For example: iPhone App development).

Experience or knowledge of SQL or other database environments.

Experience or knowledge of Web applications and services.

Demonstrated experience working with RESTful and / or SOAP based APIs.



Experience or knowledge of XML schemas or DTDs such as TEI, KML, or RDF.

Experience or knowledge of libraries, preferably academic libraries.

Job Skills:

Ability to work on unique, one-of-a-kind projects.

Excellent oral and written communication skills.

Ability to work collaboratively with others from diverse personal and professional backgrounds.

Ability to work independently or collaboratively in group settings.

Ability to quickly learn new programming languages and technologies.

Ability to think creatively and problem solve.

Highly flexible.

Physical Requirements: (Please complete attached chart)

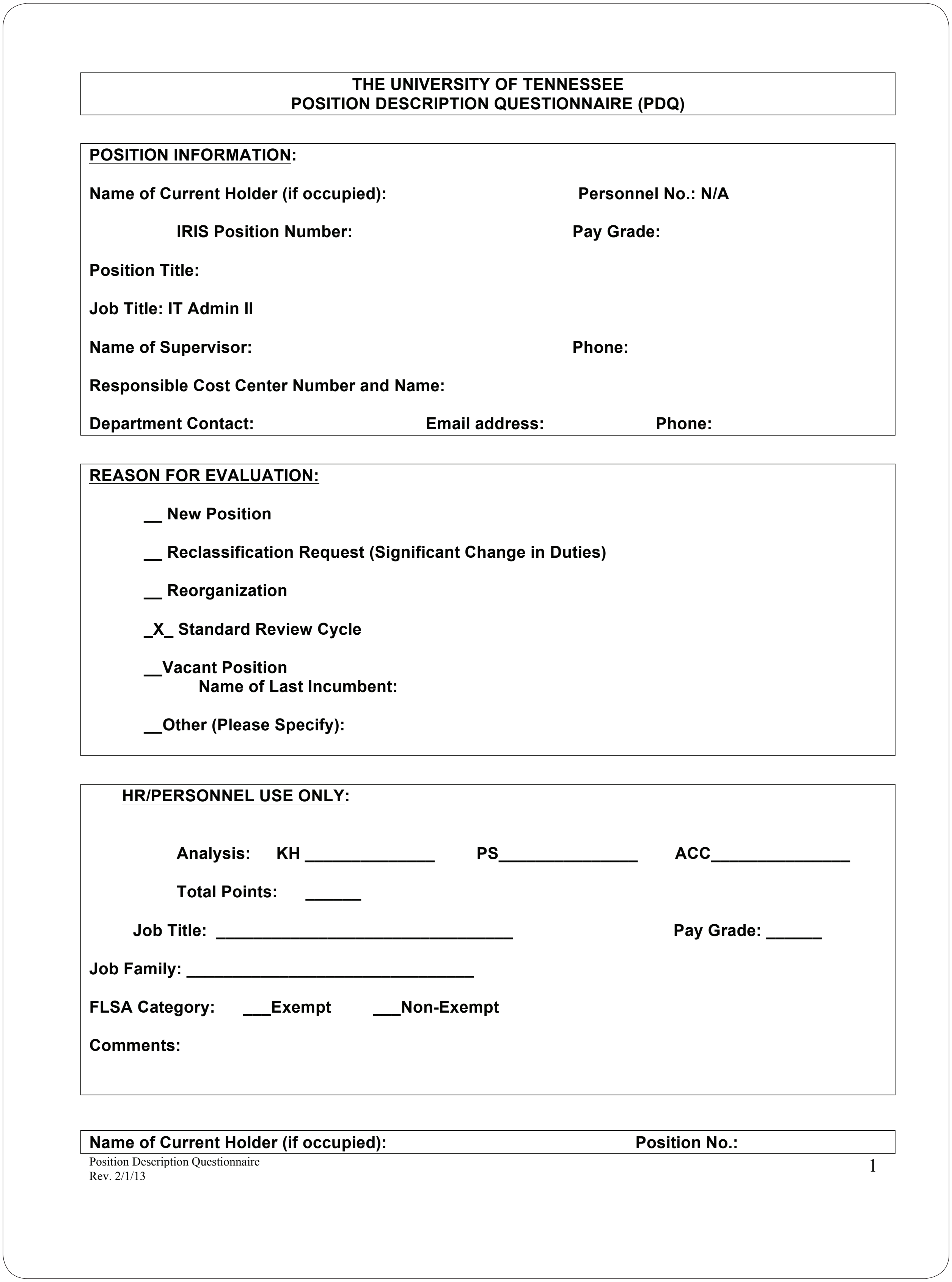
G. ADDITIONAL INFORMATION:
Please provide any additional information you believe will assist in understanding this position:

This position will work side-by-side on a daily basis with other programmers and system administrators as they develop and implement new and emerging information technologies in a library environment. The individual in this position will routinely engage in deeply complex technical planning, problem solving, and decision making concerning computer programs, systems, and services which students and faculty teachers and researchers in our university community will depend upon.

The key to success for the individual selected for this position is the ability to quickly learn and smoothly transition to new programming languages/technologies as specific projects demand. Unlike many programmer positions that are focused on a specific core set of technologies, this position will be much more diverse and require great technical agility.

***Attach the Departmental Organizational Chart prepared by your department with names and titles (include to whom this position reports, others who report to the same individual, and who reports to this position).**

H. APPROVAL:
This Position Description Questionnaire (PDQ) has been reviewed by the individuals whose signatures appear below, indicating that the PDQ accurately reflects the job content of the position:



THE UNIVERSITY OF TENNESSEE
POSITION DESCRIPTION QUESTIONNAIRE (PDQ)

POSITION INFORMATION:

Name of Current Holder (if occupied): **Personnel No.: N/A**

IRIS Position Number: **Pay Grade:**

Position Title:

Job Title: IT Admin II

Name of Supervisor: **Phone:**

Responsible Cost Center Number and Name:

Department Contact: **Email address:** **Phone:**

REASON FOR EVALUATION:

__ New Position

__ Reclassification Request (Significant Change in Duties)

__ Reorganization

_X_ Standard Review Cycle

__Vacant Position
Name of Last Incumbent:

__Other (Please Specify):

HR/PERSONNEL USE ONLY:

Analysis: KH ______________ PS_______________ ACC_______________

Total Points: ______

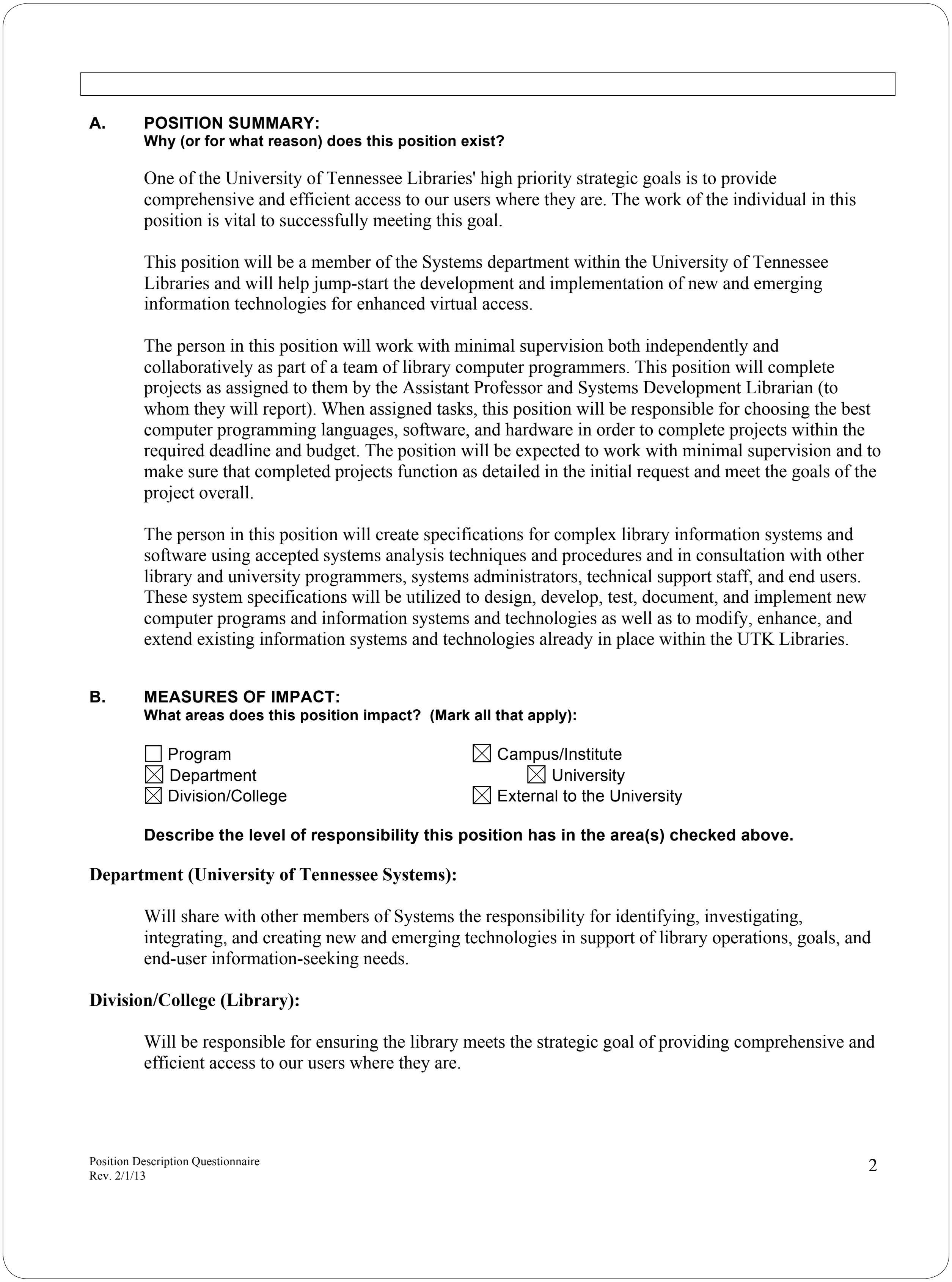
Job Title: ________________________________ **Pay Grade: ______**

Job Family: _______________________________

FLSA Category: ___Exempt ___Non-Exempt

Comments:

Name of Current Holder (if occupied): **Position No.:**



A. **POSITION SUMMARY:**
Why (or for what reason) does this position exist?

One of the University of Tennessee Libraries' high priority strategic goals is to provide comprehensive and efficient access to our users where they are. The work of the individual in this position is vital to successfully meeting this goal.

This position will be a member of the Systems department within the University of Tennessee Libraries and will help jump-start the development and implementation of new and emerging information technologies for enhanced virtual access.

The person in this position will work with minimal supervision both independently and collaboratively as part of a team of library computer programmers. This position will complete projects as assigned to them by the Assistant Professor and Systems Development Librarian (to whom they will report). When assigned tasks, this position will be responsible for choosing the best computer programming languages, software, and hardware in order to complete projects within the required deadline and budget. The position will be expected to work with minimal supervision and to make sure that completed projects function as detailed in the initial request and meet the goals of the project overall.

The person in this position will create specifications for complex library information systems and software using accepted systems analysis techniques and procedures and in consultation with other library and university programmers, systems administrators, technical support staff, and end users. These system specifications will be utilized to design, develop, test, document, and implement new computer programs and information systems and technologies as well as to modify, enhance, and extend existing information systems and technologies already in place within the UTK Libraries.

B. **MEASURES OF IMPACT:**
What areas does this position impact? (Mark all that apply):

□ Program
☒ Department
☒ Division/College

☒ Campus/Institute
☒ University
☒ External to the University

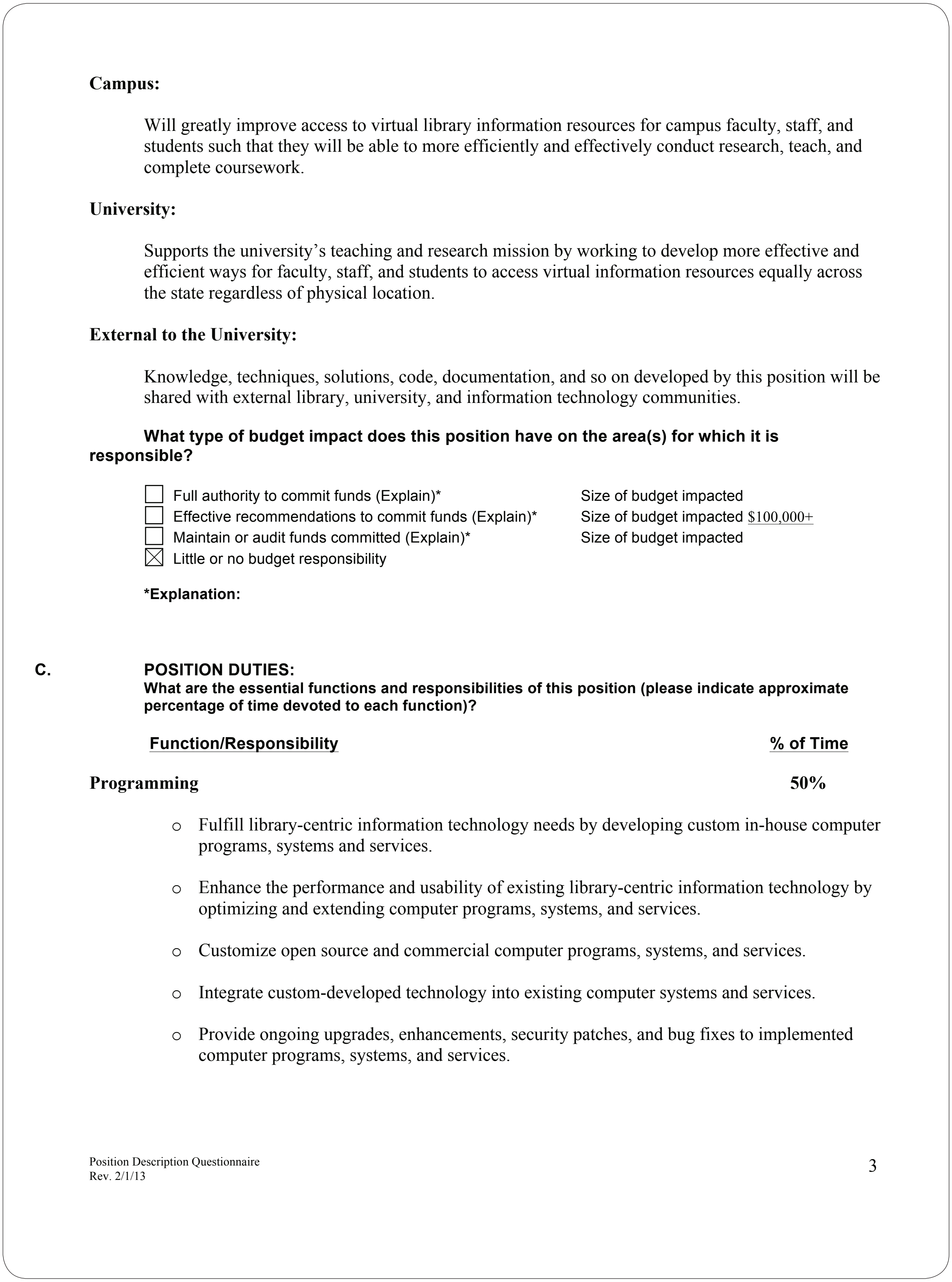
Describe the level of responsibility this position has in the area(s) checked above.

Department (University of Tennessee Systems):

Will share with other members of Systems the responsibility for identifying, investigating, integrating, and creating new and emerging technologies in support of library operations, goals, and end-user information-seeking needs.

Division/College (Library):

Will be responsible for ensuring the library meets the strategic goal of providing comprehensive and efficient access to our users where they are.



Campus:

Will greatly improve access to virtual library information resources for campus faculty, staff, and students such that they will be able to more efficiently and effectively conduct research, teach, and complete coursework.

University:

Supports the university's teaching and research mission by working to develop more effective and efficient ways for faculty, staff, and students to access virtual information resources equally across the state regardless of physical location.

External to the University:

Knowledge, techniques, solutions, code, documentation, and so on developed by this position will be shared with external library, university, and information technology communities.

What type of budget impact does this position have on the area(s) for which it is responsible?

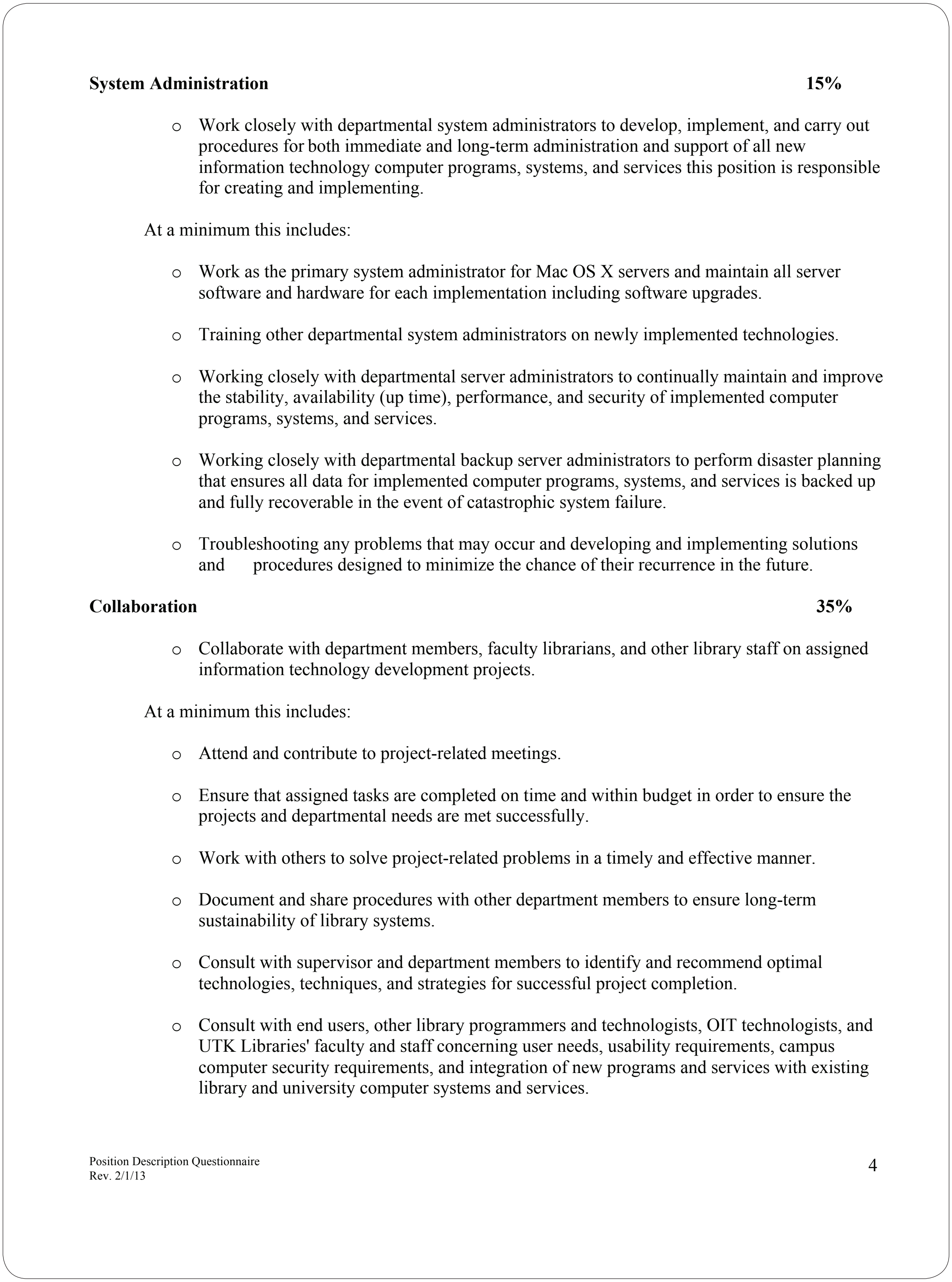
□ Full authority to commit funds (Explain)*	Size of budget impacted
□ Effective recommendations to commit funds (Explain)*	Size of budget impacted $100,000+
□ Maintain or audit funds committed (Explain)*	Size of budget impacted
☒ Little or no budget responsibility	

***Explanation:**

C. POSITION DUTIES:
What are the essential functions and responsibilities of this position (please indicate approximate percentage of time devoted to each function)?

Function/Responsibility	**% of Time**
Programming	**50%**

- Fulfill library-centric information technology needs by developing custom in-house computer programs, systems and services.
- Enhance the performance and usability of existing library-centric information technology by optimizing and extending computer programs, systems, and services.
- Customize open source and commercial computer programs, systems, and services.
- Integrate custom-developed technology into existing computer systems and services.
- Provide ongoing upgrades, enhancements, security patches, and bug fixes to implemented computer programs, systems, and services.



System Administration 15%

- Work closely with departmental system administrators to develop, implement, and carry out procedures for both immediate and long-term administration and support of all new information technology computer programs, systems, and services this position is responsible for creating and implementing.

At a minimum this includes:

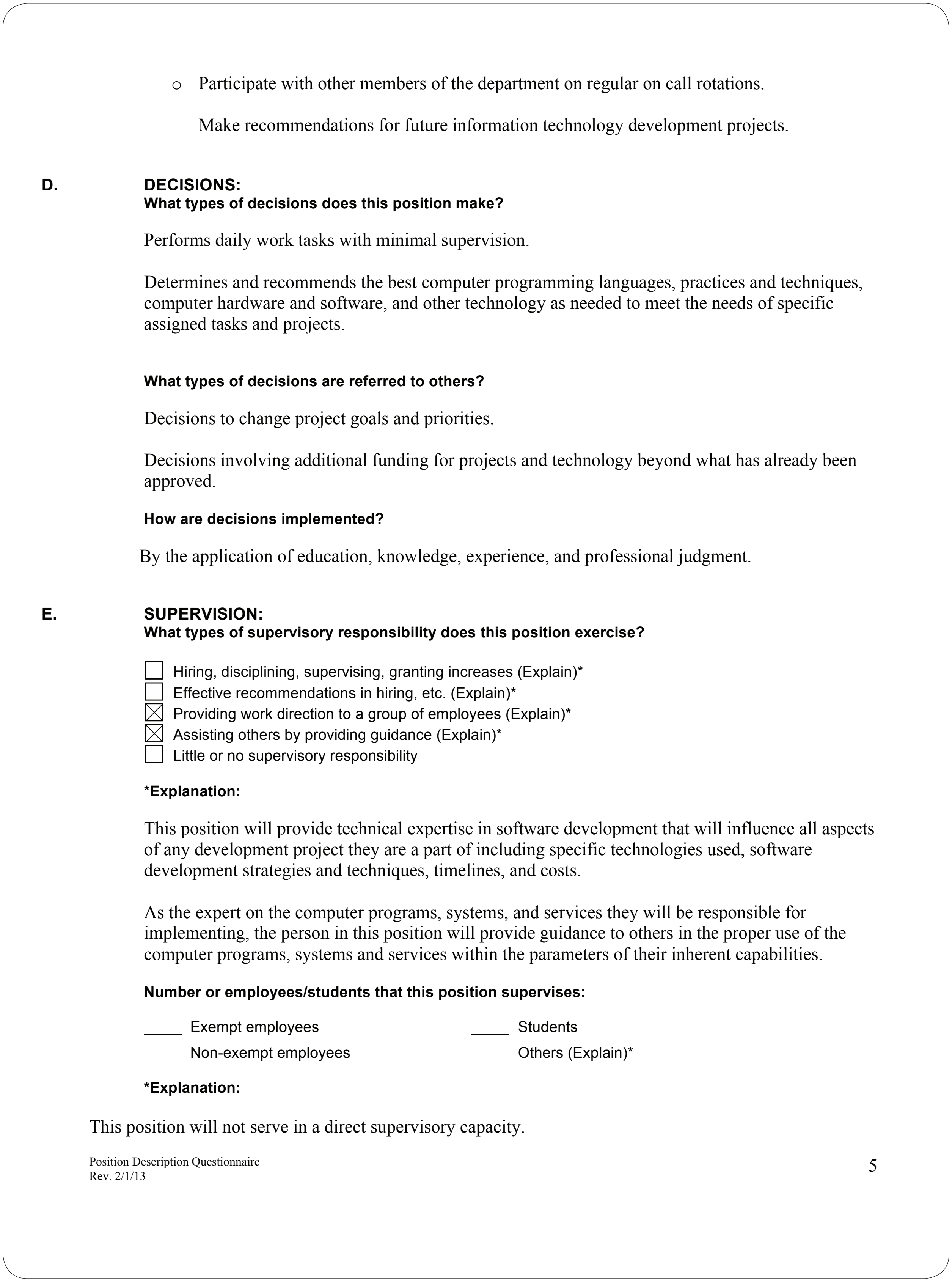
- Work as the primary system administrator for Mac OS X servers and maintain all server software and hardware for each implementation including software upgrades.
- Training other departmental system administrators on newly implemented technologies.
- Working closely with departmental server administrators to continually maintain and improve the stability, availability (up time), performance, and security of implemented computer programs, systems, and services.
- Working closely with departmental backup server administrators to perform disaster planning that ensures all data for implemented computer programs, systems, and services is backed up and fully recoverable in the event of catastrophic system failure.
- Troubleshooting any problems that may occur and developing and implementing solutions and procedures designed to minimize the chance of their recurrence in the future.

Collaboration 35%

- Collaborate with department members, faculty librarians, and other library staff on assigned information technology development projects.

At a minimum this includes:

- Attend and contribute to project-related meetings.
- Ensure that assigned tasks are completed on time and within budget in order to ensure the projects and departmental needs are met successfully.
- Work with others to solve project-related problems in a timely and effective manner.
- Document and share procedures with other department members to ensure long-term sustainability of library systems.
- Consult with supervisor and department members to identify and recommend optimal technologies, techniques, and strategies for successful project completion.
- Consult with end users, other library programmers and technologists, OIT technologists, and UTK Libraries' faculty and staff concerning user needs, usability requirements, campus computer security requirements, and integration of new programs and services with existing library and university computer systems and services.



- Participate with other members of the department on regular on call rotations.

  Make recommendations for future information technology development projects.

D. DECISIONS:
What types of decisions does this position make?

Performs daily work tasks with minimal supervision.

Determines and recommends the best computer programming languages, practices and techniques, computer hardware and software, and other technology as needed to meet the needs of specific assigned tasks and projects.

What types of decisions are referred to others?

Decisions to change project goals and priorities.

Decisions involving additional funding for projects and technology beyond what has already been approved.

How are decisions implemented?

By the application of education, knowledge, experience, and professional judgment.

E. SUPERVISION:
What types of supervisory responsibility does this position exercise?

□ Hiring, disciplining, supervising, granting increases (Explain)*
□ Effective recommendations in hiring, etc. (Explain)*
☒ Providing work direction to a group of employees (Explain)*
☒ Assisting others by providing guidance (Explain)*
□ Little or no supervisory responsibility

***Explanation:**

This position will provide technical expertise in software development that will influence all aspects of any development project they are a part of including specific technologies used, software development strategies and techniques, timelines, and costs.

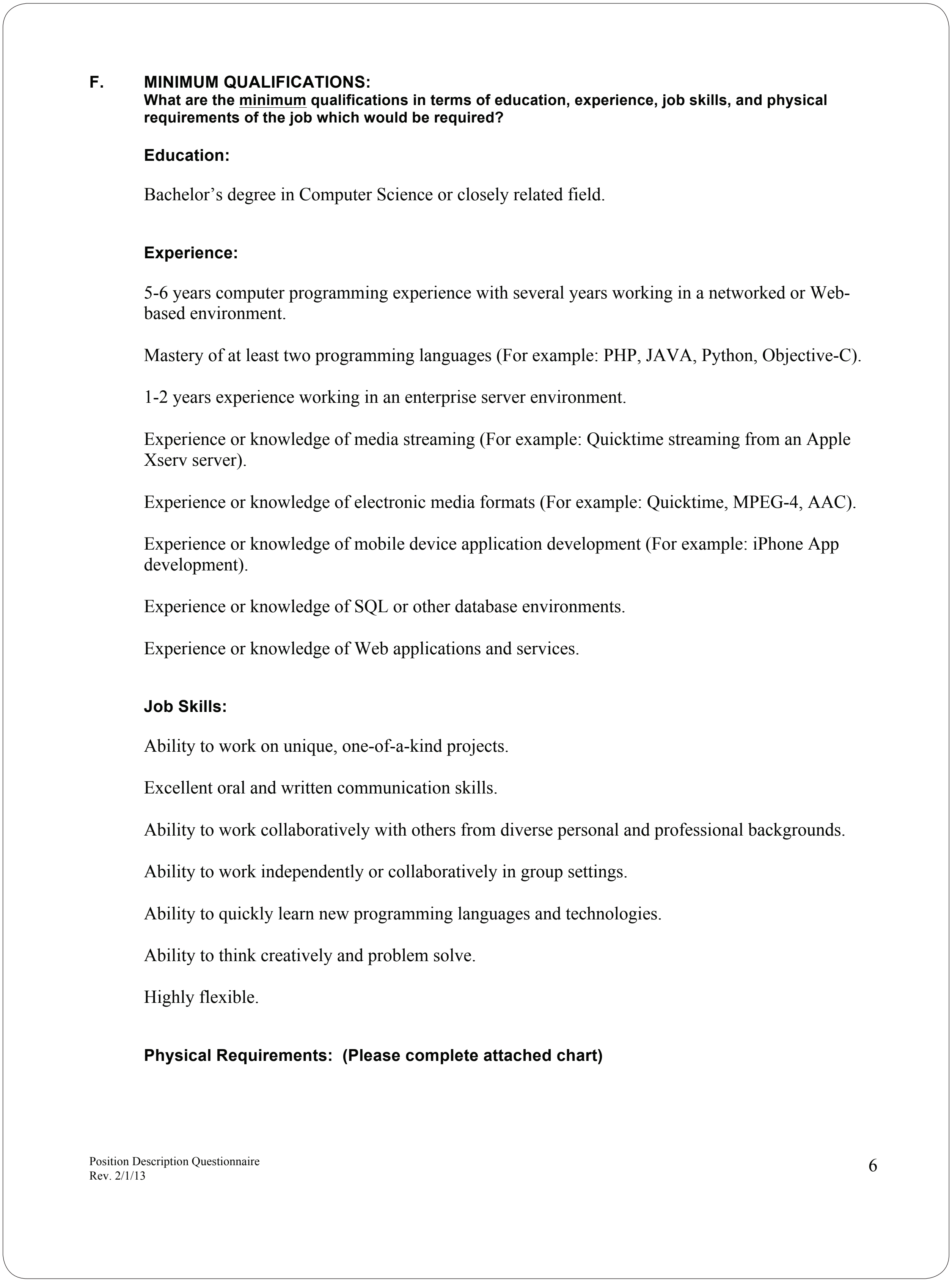
As the expert on the computer programs, systems, and services they will be responsible for implementing, the person in this position will provide guidance to others in the proper use of the computer programs, systems and services within the parameters of their inherent capabilities.

Number or employees/students that this position supervises:

_____ Exempt employees _____ Students
_____ Non-exempt employees _____ Others (Explain)*

***Explanation:**

This position will not serve in a direct supervisory capacity.



F. MINIMUM QUALIFICATIONS:
What are the minimum qualifications in terms of education, experience, job skills, and physical requirements of the job which would be required?

Education:

Bachelor’s degree in Computer Science or closely related field.

Experience:

5-6 years computer programming experience with several years working in a networked or Web-based environment.

Mastery of at least two programming languages (For example: PHP, JAVA, Python, Objective-C).

1-2 years experience working in an enterprise server environment.

Experience or knowledge of media streaming (For example: Quicktime streaming from an Apple Xserv server).

Experience or knowledge of electronic media formats (For example: Quicktime, MPEG-4, AAC).

Experience or knowledge of mobile device application development (For example: iPhone App development).

Experience or knowledge of SQL or other database environments.

Experience or knowledge of Web applications and services.

Job Skills:

Ability to work on unique, one-of-a-kind projects.

Excellent oral and written communication skills.

Ability to work collaboratively with others from diverse personal and professional backgrounds.

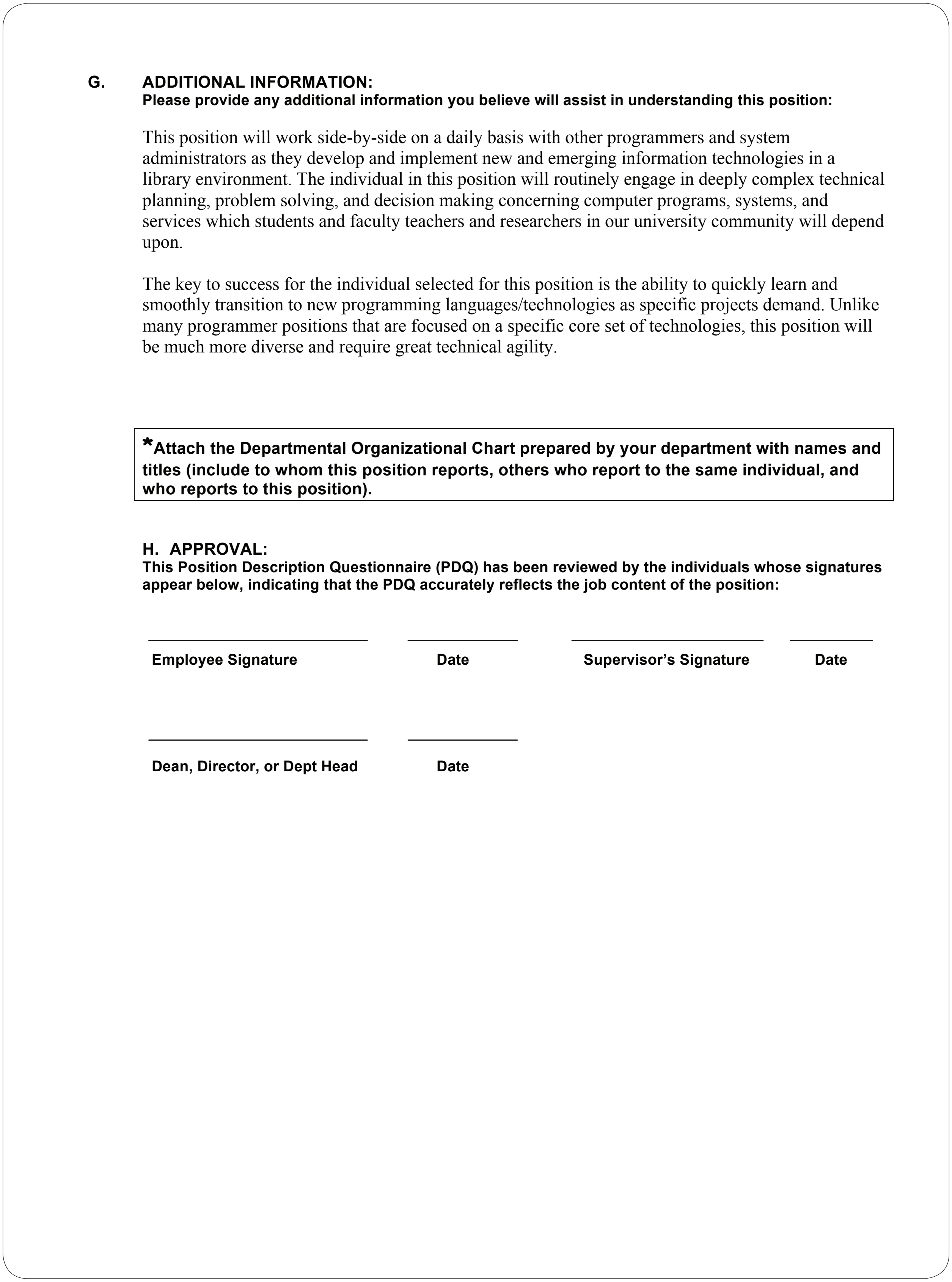
Ability to work independently or collaboratively in group settings.

Ability to quickly learn new programming languages and technologies.

Ability to think creatively and problem solve.

Highly flexible.

Physical Requirements: (Please complete attached chart)



G. ADDITIONAL INFORMATION:
Please provide any additional information you believe will assist in understanding this position:

This position will work side-by-side on a daily basis with other programmers and system administrators as they develop and implement new and emerging information technologies in a library environment. The individual in this position will routinely engage in deeply complex technical planning, problem solving, and decision making concerning computer programs, systems, and services which students and faculty teachers and researchers in our university community will depend upon.

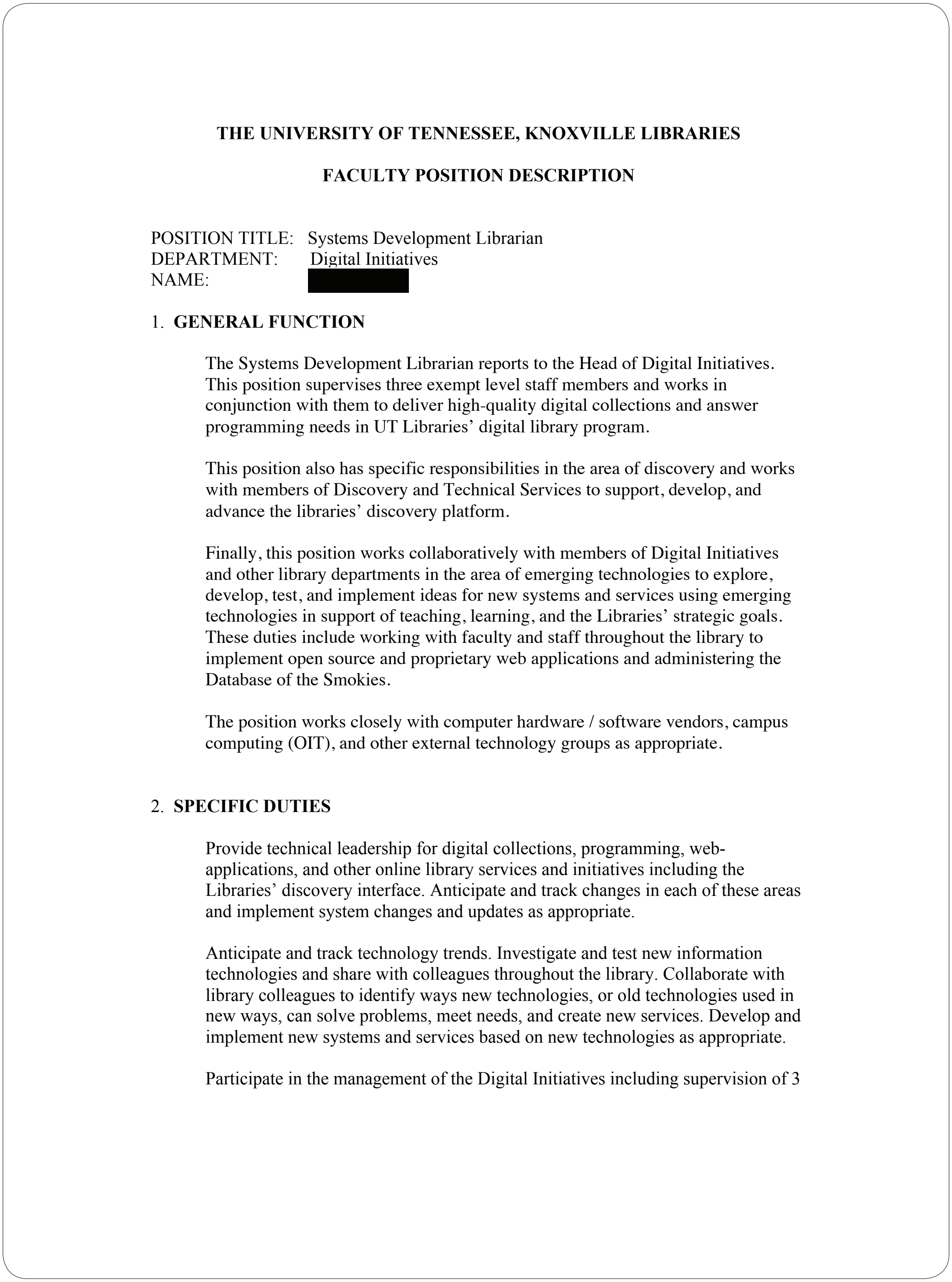
The key to success for the individual selected for this position is the ability to quickly learn and smoothly transition to new programming languages/technologies as specific projects demand. Unlike many programmer positions that are focused on a specific core set of technologies, this position will be much more diverse and require great technical agility.

***Attach the Departmental Organizational Chart prepared by your department with names and titles (include to whom this position reports, others who report to the same individual, and who reports to this position).**

H. APPROVAL:
This Position Description Questionnaire (PDQ) has been reviewed by the individuals whose signatures appear below, indicating that the PDQ accurately reflects the job content of the position:

Employee Signature **Date** **Supervisor's Signature** **Date**

Dean, Director, or Dept Head **Date**



THE UNIVERSITY OF TENNESSEE, KNOXVILLE LIBRARIES

FACULTY POSITION DESCRIPTION

POSITION TITLE: Systems Development Librarian
DEPARTMENT: Digital Initiatives
NAME: [redacted]

1. **GENERAL FUNCTION**

The Systems Development Librarian reports to the Head of Digital Initiatives. This position supervises three exempt level staff members and works in conjunction with them to deliver high-quality digital collections and answer programming needs in UT Libraries' digital library program.

This position also has specific responsibilities in the area of discovery and works with members of Discovery and Technical Services to support, develop, and advance the libraries' discovery platform.

Finally, this position works collaboratively with members of Digital Initiatives and other library departments in the area of emerging technologies to explore, develop, test, and implement ideas for new systems and services using emerging technologies in support of teaching, learning, and the Libraries' strategic goals. These duties include working with faculty and staff throughout the library to implement open source and proprietary web applications and administering the Database of the Smokies.

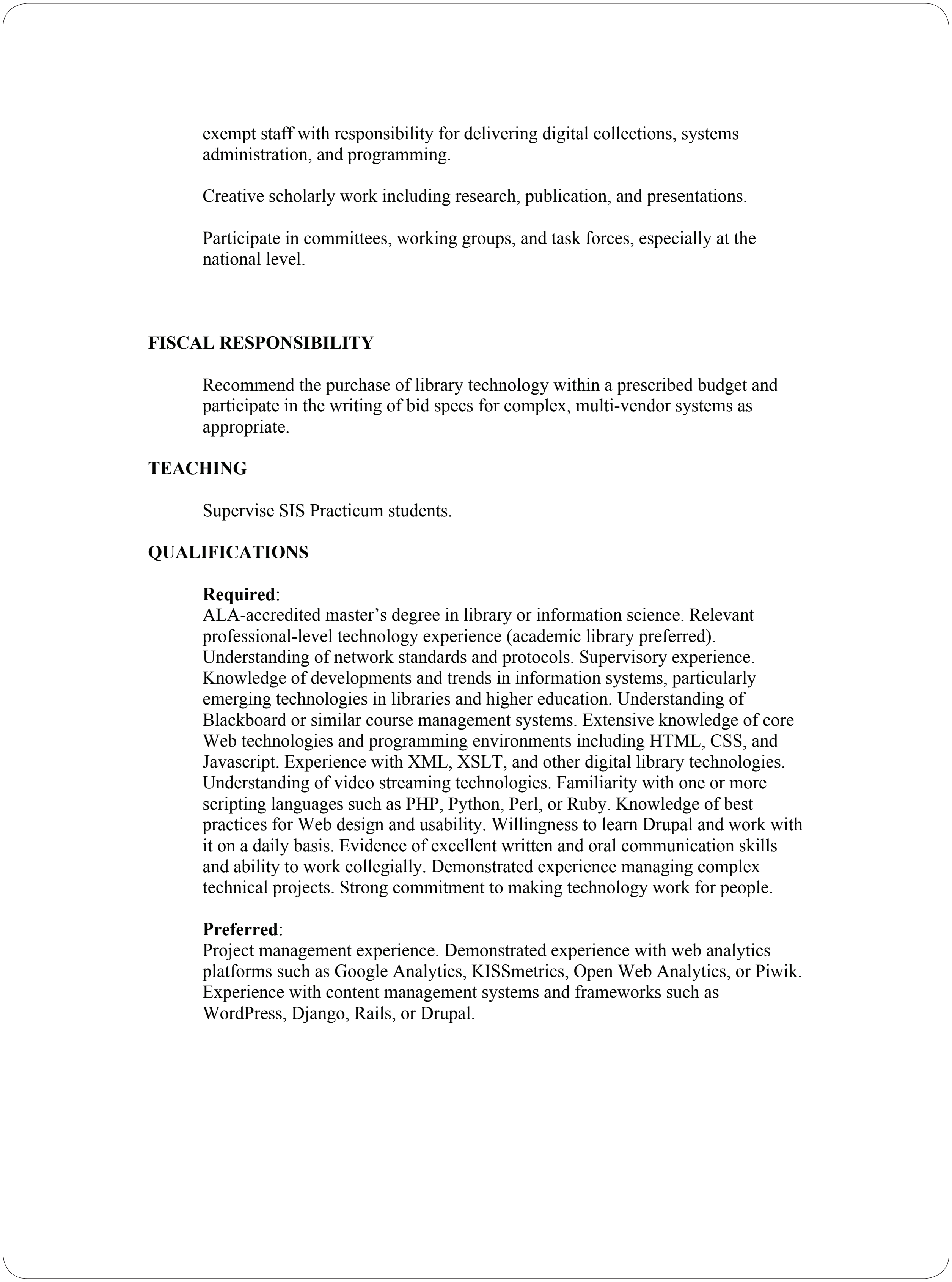
The position works closely with computer hardware / software vendors, campus computing (OIT), and other external technology groups as appropriate.

2. **SPECIFIC DUTIES**

Provide technical leadership for digital collections, programming, web-applications, and other online library services and initiatives including the Libraries' discovery interface. Anticipate and track changes in each of these areas and implement system changes and updates as appropriate.

Anticipate and track technology trends. Investigate and test new information technologies and share with colleagues throughout the library. Collaborate with library colleagues to identify ways new technologies, or old technologies used in new ways, can solve problems, meet needs, and create new services. Develop and implement new systems and services based on new technologies as appropriate.

Participate in the management of the Digital Initiatives including supervision of 3



exempt staff with responsibility for delivering digital collections, systems administration, and programming.

Creative scholarly work including research, publication, and presentations.

Participate in committees, working groups, and task forces, especially at the national level.

FISCAL RESPONSIBILITY

Recommend the purchase of library technology within a prescribed budget and participate in the writing of bid specs for complex, multi-vendor systems as appropriate.

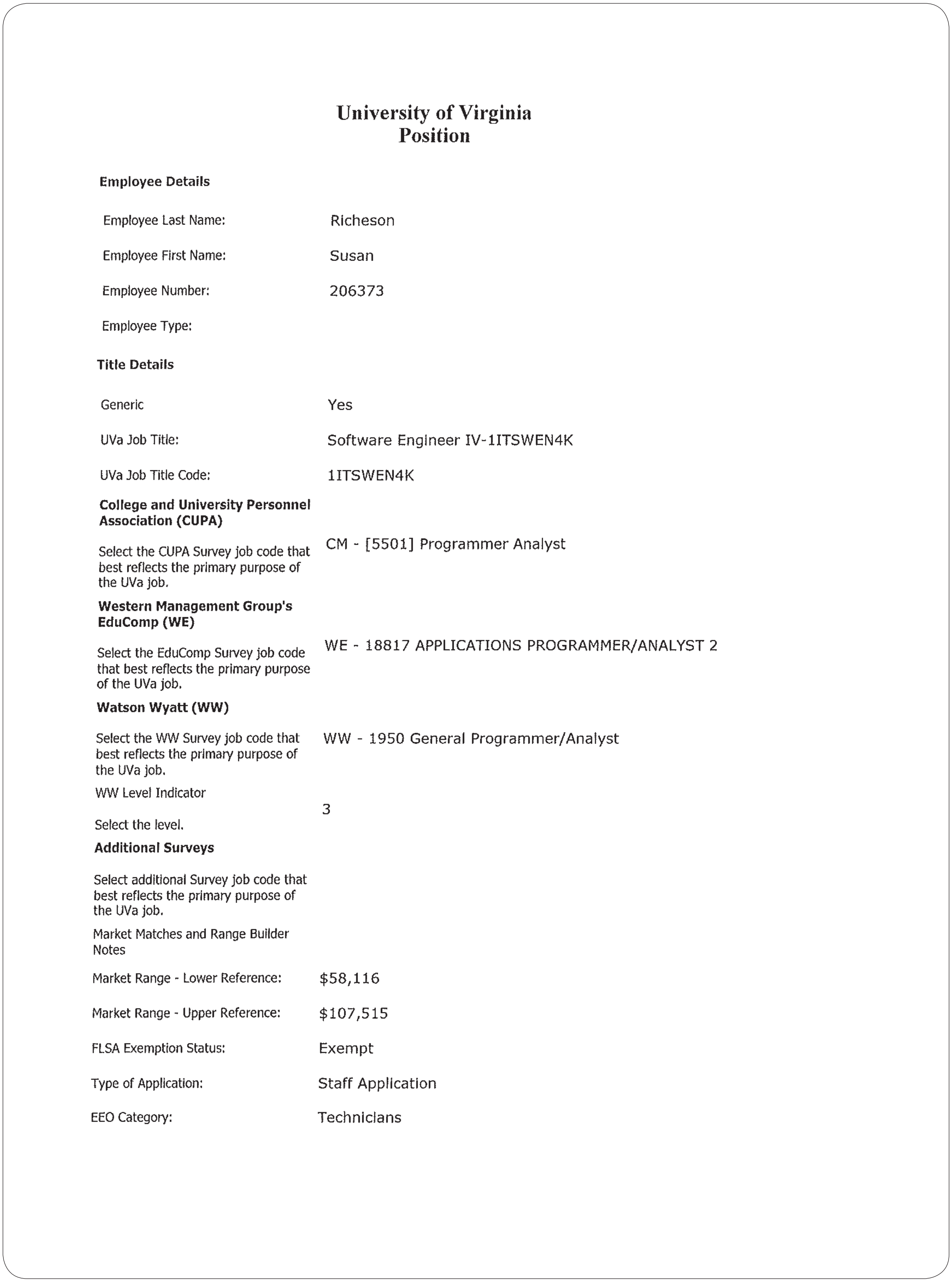
TEACHING

Supervise SIS Practicum students.

QUALIFICATIONS

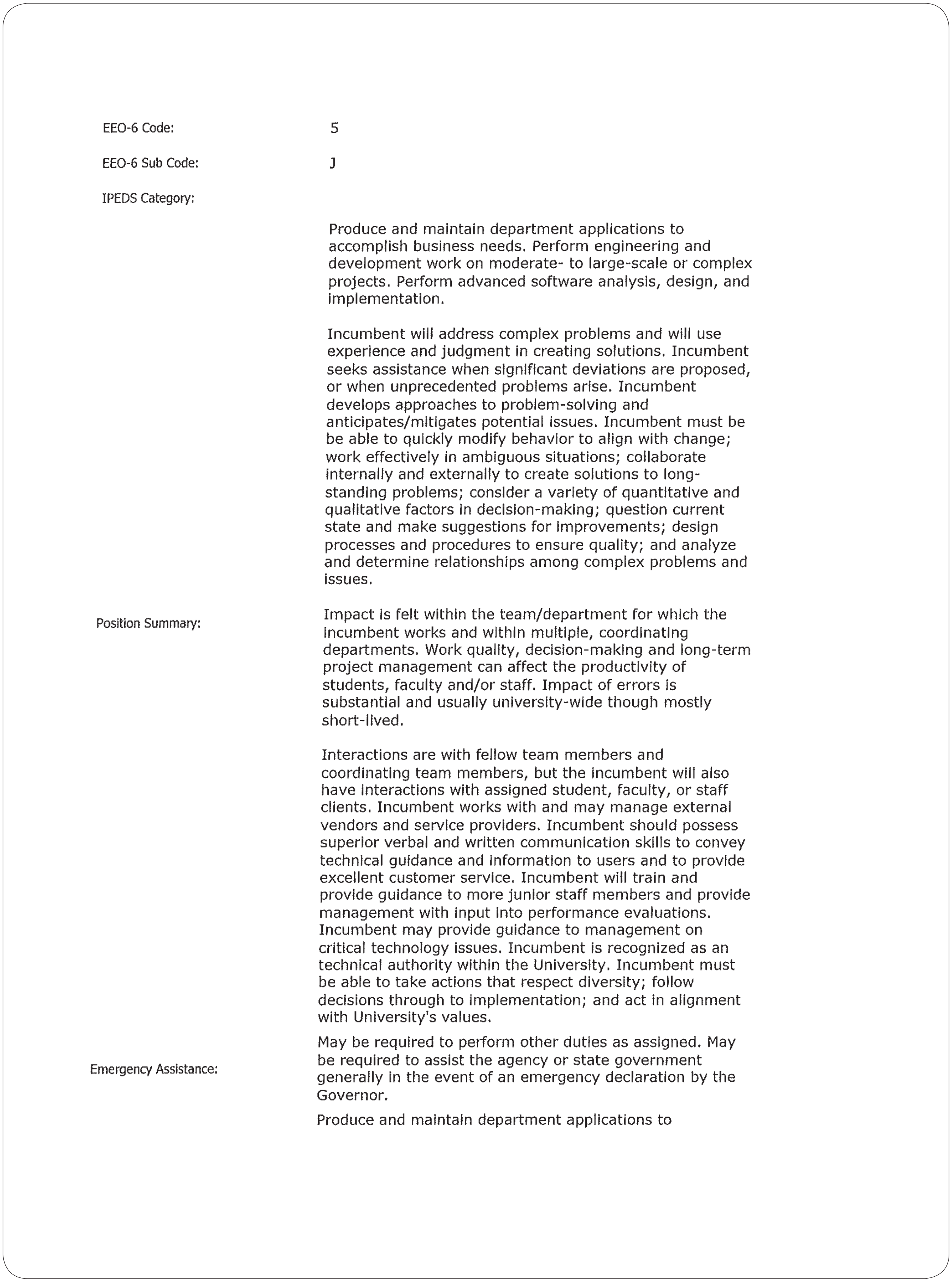
Required:
ALA-accredited master's degree in library or information science. Relevant professional-level technology experience (academic library preferred). Understanding of network standards and protocols. Supervisory experience. Knowledge of developments and trends in information systems, particularly emerging technologies in libraries and higher education. Understanding of Blackboard or similar course management systems. Extensive knowledge of core Web technologies and programming environments including HTML, CSS, and Javascript. Experience with XML, XSLT, and other digital library technologies. Understanding of video streaming technologies. Familiarity with one or more scripting languages such as PHP, Python, Perl, or Ruby. Knowledge of best practices for Web design and usability. Willingness to learn Drupal and work with it on a daily basis. Evidence of excellent written and oral communication skills and ability to work collegially. Demonstrated experience managing complex technical projects. Strong commitment to making technology work for people.

Preferred:
Project management experience. Demonstrated experience with web analytics platforms such as Google Analytics, KISSmetrics, Open Web Analytics, or Piwik. Experience with content management systems and frameworks such as WordPress, Django, Rails, or Drupal.



University of Virginia Position

Employee Details	
Employee Last Name:	Richeson
Employee First Name:	Susan
Employee Number:	206373
Employee Type:	
Title Details	
Generic	Yes
UVa Job Title:	Software Engineer IV-1ITSWEN4K
UVa Job Title Code:	1ITSWEN4K
College and University Personnel Association (CUPA)	
Select the CUPA Survey job code that best reflects the primary purpose of the UVa job.	CM - [5501] Programmer Analyst
Western Management Group's EduComp (WE)	
Select the EduComp Survey job code that best reflects the primary purpose of the UVa job.	WE - 18817 APPLICATIONS PROGRAMMER/ANALYST 2
Watson Wyatt (WW)	
Select the WW Survey job code that best reflects the primary purpose of the UVa job.	WW - 1950 General Programmer/Analyst
WW Level Indicator Select the level.	3
Additional Surveys	
Select additional Survey job code that best reflects the primary purpose of the UVa job.	
Market Matches and Range Builder Notes	
Market Range - Lower Reference:	$58,116
Market Range - Upper Reference:	$107,515
FLSA Exemption Status:	Exempt
Type of Application:	Staff Application
EEO Category:	Technicians



EEO-6 Code: 5

EEO-6 Sub Code: J

IPEDS Category:

Position Summary:

Produce and maintain department applications to accomplish business needs. Perform engineering and development work on moderate- to large-scale or complex projects. Perform advanced software analysis, design, and implementation.

Incumbent will address complex problems and will use experience and judgment in creating solutions. Incumbent seeks assistance when significant deviations are proposed, or when unprecedented problems arise. Incumbent develops approaches to problem-solving and anticipates/mitigates potential issues. Incumbent must be be able to quickly modify behavior to align with change; work effectively in ambiguous situations; collaborate internally and externally to create solutions to long-standing problems; consider a variety of quantitative and qualitative factors in decision-making; question current state and make suggestions for improvements; design processes and procedures to ensure quality; and analyze and determine relationships among complex problems and issues.

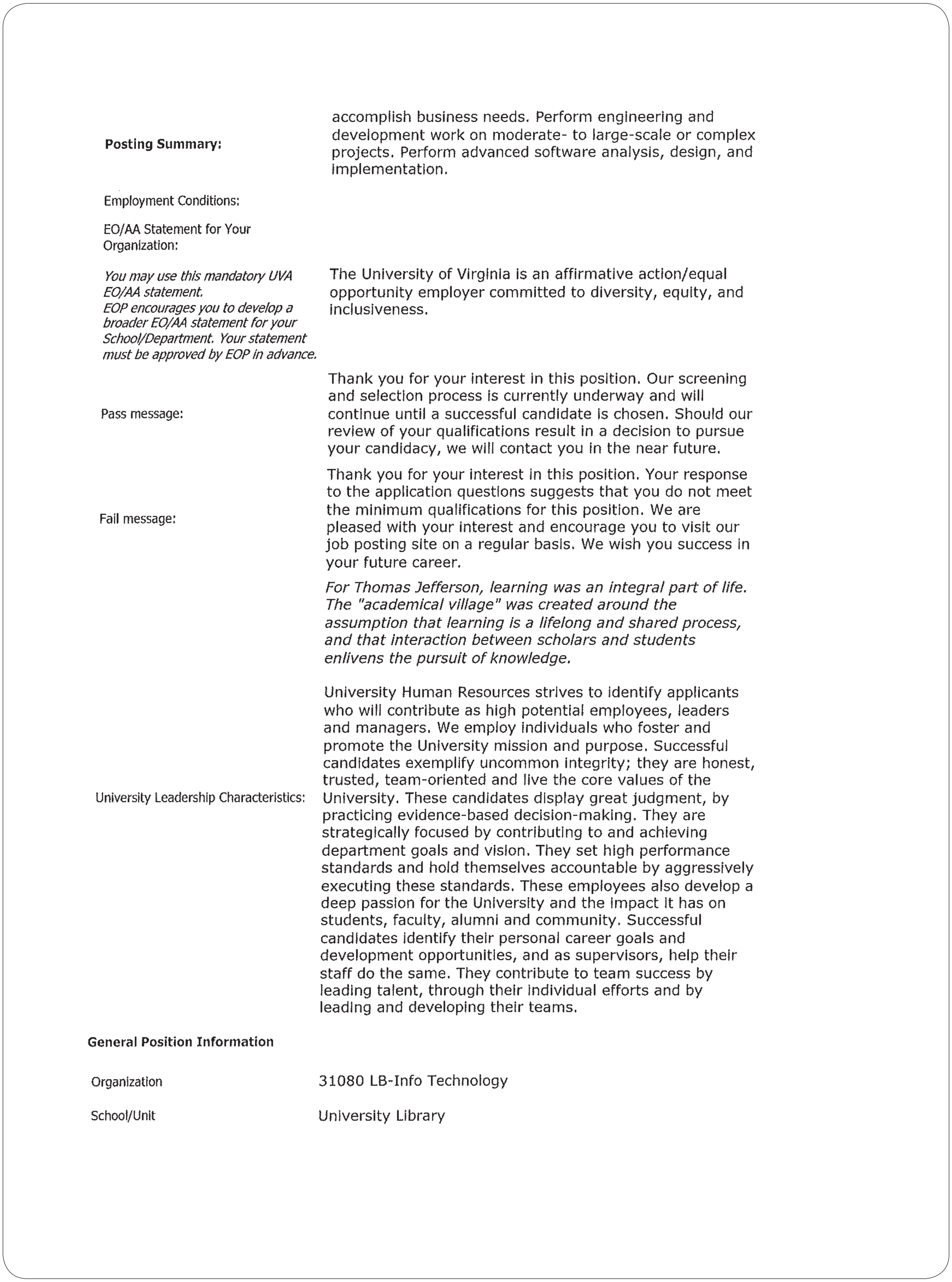
Impact is felt within the team/department for which the incumbent works and within multiple, coordinating departments. Work quality, decision-making and long-term project management can affect the productivity of students, faculty and/or staff. Impact of errors is substantial and usually university-wide though mostly short-lived.

Interactions are with fellow team members and coordinating team members, but the incumbent will also have interactions with assigned student, faculty, or staff clients. Incumbent works with and may manage external vendors and service providers. Incumbent should possess superior verbal and written communication skills to convey technical guidance and information to users and to provide excellent customer service. Incumbent will train and provide guidance to more junior staff members and provide management with input into performance evaluations. Incumbent may provide guidance to management on critical technology issues. Incumbent is recognized as an technical authority within the University. Incumbent must be able to take actions that respect diversity; follow decisions through to implementation; and act in alignment with University's values.

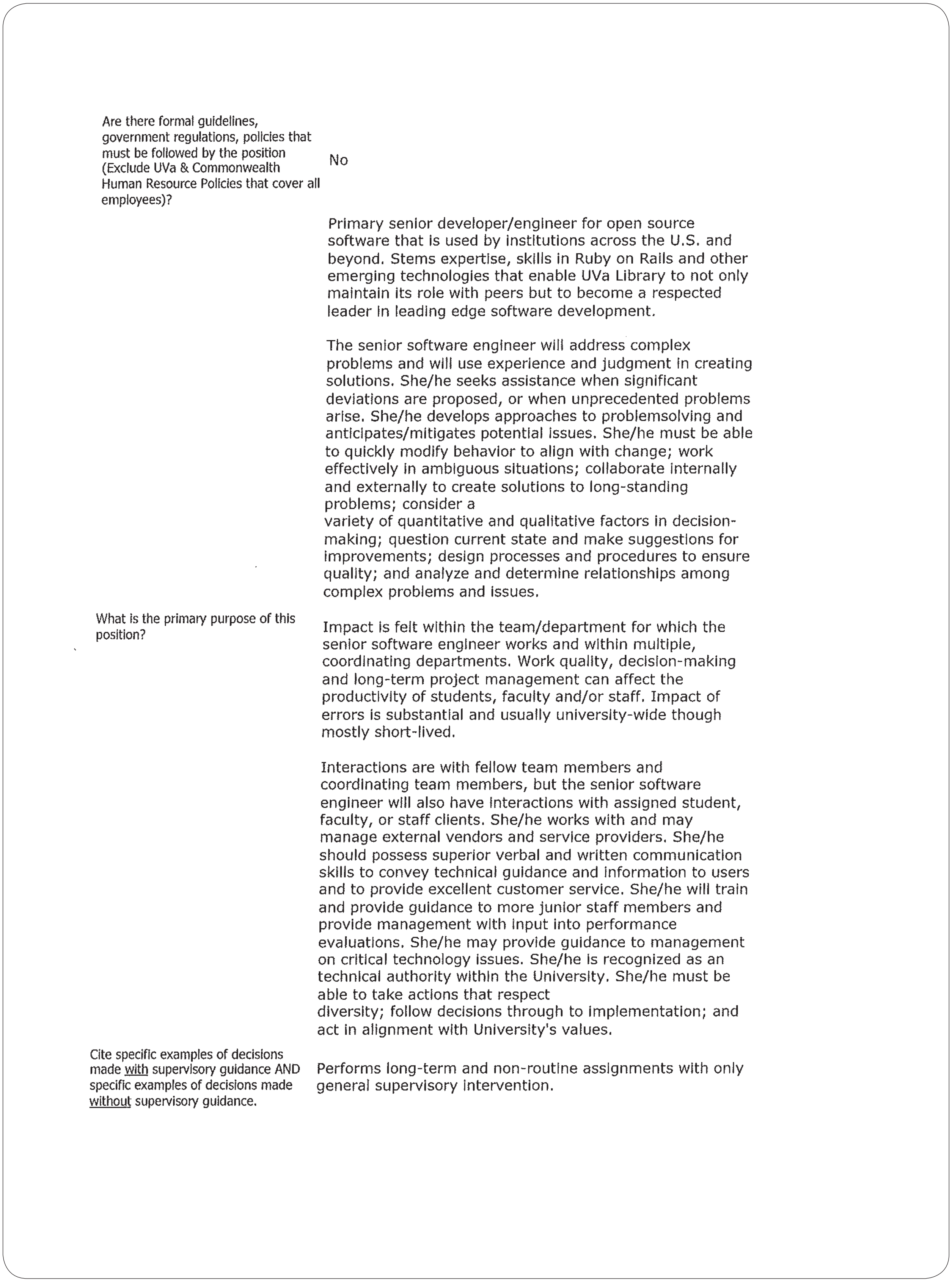
Emergency Assistance:

May be required to perform other duties as assigned. May be required to assist the agency or state government generally in the event of an emergency declaration by the Governor.

Produce and maintain department applications to



Posting Summary:	accomplish business needs. Perform engineering and development work on moderate- to large-scale or complex projects. Perform advanced software analysis, design, and implementation.
Employment Conditions:	
EO/AA Statement for Your Organization:	
You may use this mandatory UVA EO/AA statement. *EOP encourages you to develop a broader EO/AA statement for your School/Department. Your statement must be approved by EOP in advance.*	The University of Virginia is an affirmative action/equal opportunity employer committed to diversity, equity, and inclusiveness.
Pass message:	Thank you for your interest in this position. Our screening and selection process is currently underway and will continue until a successful candidate is chosen. Should our review of your qualifications result in a decision to pursue your candidacy, we will contact you in the near future.
Fail message:	Thank you for your interest in this position. Your response to the application questions suggests that you do not meet the minimum qualifications for this position. We are pleased with your interest and encourage you to visit our job posting site on a regular basis. We wish you success in your future career.
	For Thomas Jefferson, learning was an integral part of life. The "academical village" was created around the assumption that learning is a lifelong and shared process, and that interaction between scholars and students enlivens the pursuit of knowledge.
University Leadership Characteristics:	University Human Resources strives to identify applicants who will contribute as high potential employees, leaders and managers. We employ individuals who foster and promote the University mission and purpose. Successful candidates exemplify uncommon integrity; they are honest, trusted, team-oriented and live the core values of the University. These candidates display great judgment, by practicing evidence-based decision-making. They are strategically focused by contributing to and achieving department goals and vision. They set high performance standards and hold themselves accountable by aggressively executing these standards. These employees also develop a deep passion for the University and the impact it has on students, faculty, alumni and community. Successful candidates identify their personal career goals and development opportunities, and as supervisors, help their staff do the same. They contribute to team success by leading talent, through their individual efforts and by leading and developing their teams.
General Position Information	
Organization	31080 LB-Info Technology
School/Unit	University Library



Are there formal guidelines, government regulations, policies that must be followed by the position (Exclude UVa & Commonwealth Human Resource Policies that cover all employees)?

No

Primary senior developer/engineer for open source software that is used by institutions across the U.S. and beyond. Stems expertise, skills in Ruby on Rails and other emerging technologies that enable UVa Library to not only maintain its role with peers but to become a respected leader in leading edge software development.

The senior software engineer will address complex problems and will use experience and judgment in creating solutions. She/he seeks assistance when significant deviations are proposed, or when unprecedented problems arise. She/he develops approaches to problemsolving and anticipates/mitigates potential issues. She/he must be able to quickly modify behavior to align with change; work effectively in ambiguous situations; collaborate internally and externally to create solutions to long-standing problems; consider a variety of quantitative and qualitative factors in decision-making; question current state and make suggestions for improvements; design processes and procedures to ensure quality; and analyze and determine relationships among complex problems and issues.

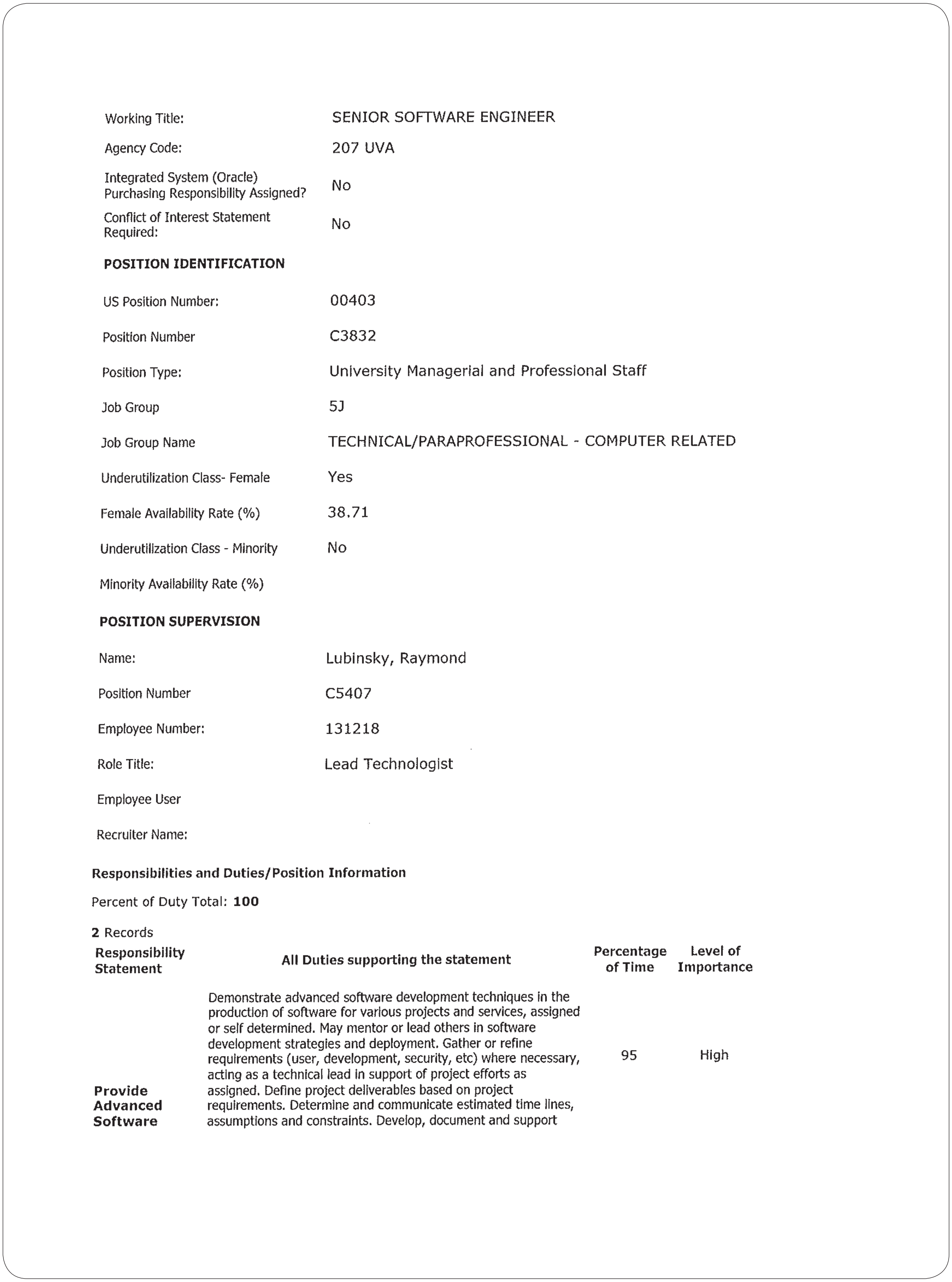
What is the primary purpose of this position?

Impact is felt within the team/department for which the senior software engineer works and within multiple, coordinating departments. Work quality, decision-making and long-term project management can affect the productivity of students, faculty and/or staff. Impact of errors is substantial and usually university-wide though mostly short-lived.

Interactions are with fellow team members and coordinating team members, but the senior software engineer will also have interactions with assigned student, faculty, or staff clients. She/he works with and may manage external vendors and service providers. She/he should possess superior verbal and written communication skills to convey technical guidance and information to users and to provide excellent customer service. She/he will train and provide guidance to more junior staff members and provide management with input into performance evaluations. She/he may provide guidance to management on critical technology issues. She/he is recognized as an technical authority within the University. She/he must be able to take actions that respect diversity; follow decisions through to implementation; and act in alignment with University's values.

Cite specific examples of decisions made <u>with</u> supervisory guidance AND specific examples of decisions made <u>without</u> supervisory guidance.

Performs long-term and non-routine assignments with only general supervisory intervention.



Working Title:	SENIOR SOFTWARE ENGINEER
Agency Code:	207 UVA
Integrated System (Oracle) Purchasing Responsibility Assigned?	No
Conflict of Interest Statement Required:	No

POSITION IDENTIFICATION

US Position Number:	00403
Position Number	C3832
Position Type:	University Managerial and Professional Staff
Job Group	5J
Job Group Name	TECHNICAL/PARAPROFESSIONAL - COMPUTER RELATED
Underutilization Class- Female	Yes
Female Availability Rate (%)	38.71
Underutilization Class - Minority	No
Minority Availability Rate (%)	

POSITION SUPERVISION

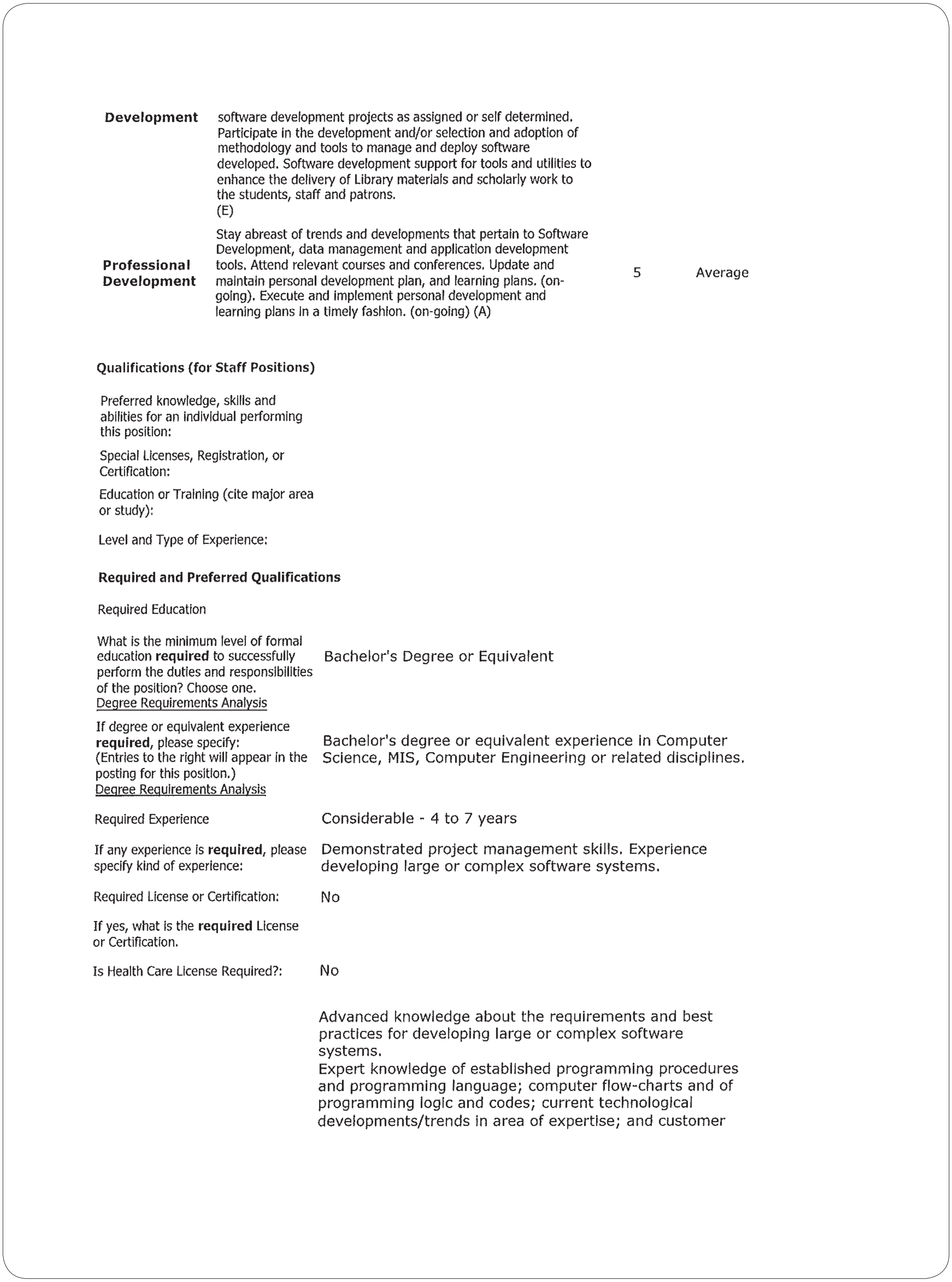
Name:	Lubinsky, Raymond
Position Number	C5407
Employee Number:	131218
Role Title:	Lead Technologist
Employee User	
Recruiter Name:	

Responsibilities and Duties/Position Information

Percent of Duty Total: **100**

2 Records

Responsibility Statement	All Duties supporting the statement	Percentage of Time	Level of Importance
Provide Advanced Software	Demonstrate advanced software development techniques in the production of software for various projects and services, assigned or self determined. May mentor or lead others in software development strategies and deployment. Gather or refine requirements (user, development, security, etc) where necessary, acting as a technical lead in support of project efforts as assigned. Define project deliverables based on project requirements. Determine and communicate estimated time lines, assumptions and constraints. Develop, document and support	95	High



Development	software development projects as assigned or self determined. Participate in the development and/or selection and adoption of methodology and tools to manage and deploy software developed. Software development support for tools and utilities to enhance the delivery of Library materials and scholarly work to the students, staff and patrons. (E)		
Professional Development	Stay abreast of trends and developments that pertain to Software Development, data management and application development tools. Attend relevant courses and conferences. Update and maintain personal development plan, and learning plans. (on-going). Execute and implement personal development and learning plans in a timely fashion. (on-going) (A)	5	Average

Qualifications (for Staff Positions)

Preferred knowledge, skills and abilities for an individual performing this position:

Special Licenses, Registration, or Certification:

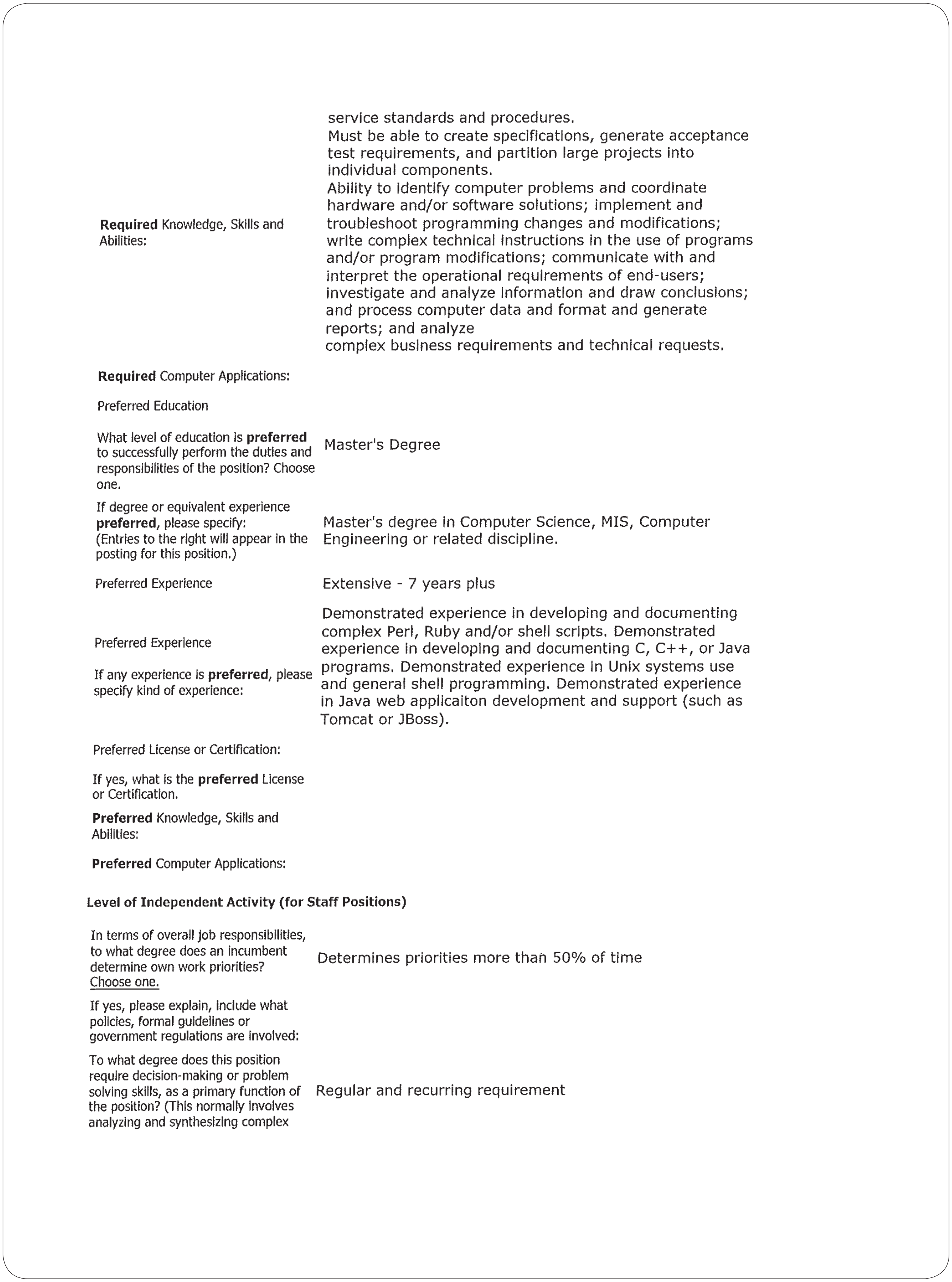
Education or Training (cite major area or study):

Level and Type of Experience:

Required and Preferred Qualifications

Required Education

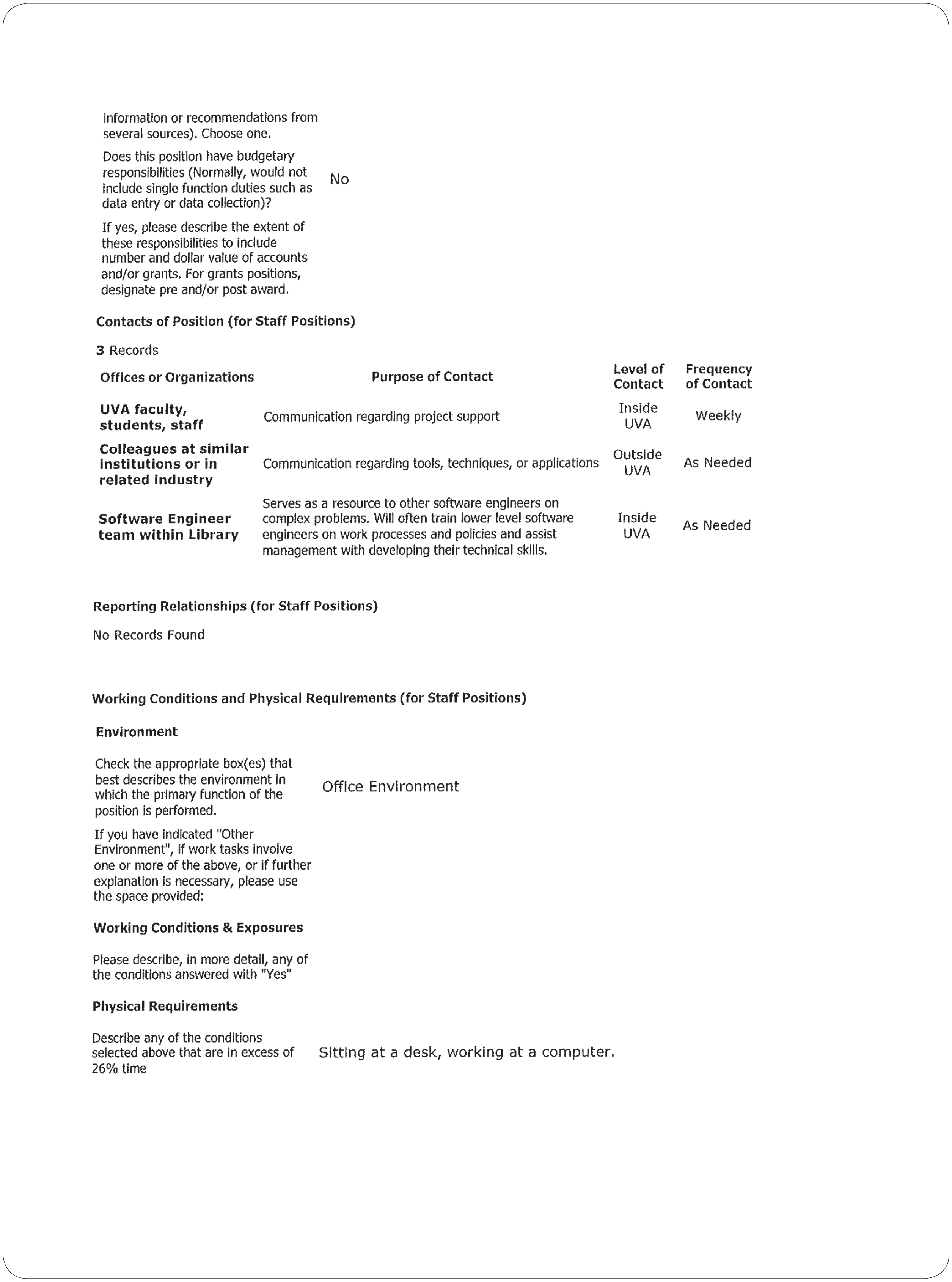
What is the minimum level of formal education **required** to successfully perform the duties and responsibilities of the position? Choose one. Degree Requirements Analysis	Bachelor's Degree or Equivalent
If degree or equivalent experience **required**, please specify: (Entries to the right will appear in the posting for this position.) Degree Requirements Analysis	Bachelor's degree or equivalent experience in Computer Science, MIS, Computer Engineering or related disciplines.
Required Experience	Considerable - 4 to 7 years
If any experience is **required**, please specify kind of experience:	Demonstrated project management skills. Experience developing large or complex software systems.
Required License or Certification:	No
If yes, what is the **required** License or Certification.	
Is Health Care License Required?:	No
	Advanced knowledge about the requirements and best practices for developing large or complex software systems. Expert knowledge of established programming procedures and programming language; computer flow-charts and of programming logic and codes; current technological developments/trends in area of expertise; and customer



	service standards and procedures. Must be able to create specifications, generate acceptance test requirements, and partition large projects into individual components. Ability to identify computer problems and coordinate hardware and/or software solutions; implement and troubleshoot programming changes and modifications; write complex technical instructions in the use of programs and/or program modifications; communicate with and interpret the operational requirements of end-users; investigate and analyze information and draw conclusions; and process computer data and format and generate reports; and analyze complex business requirements and technical requests.
Required Knowledge, Skills and Abilities:	
Required Computer Applications:	
Preferred Education	
What level of education is **preferred** to successfully perform the duties and responsibilities of the position? Choose one.	Master's Degree
If degree or equivalent experience **preferred**, please specify: (Entries to the right will appear in the posting for this position.)	Master's degree in Computer Science, MIS, Computer Engineering or related discipline.
Preferred Experience	Extensive - 7 years plus
Preferred Experience If any experience is **preferred**, please specify kind of experience:	Demonstrated experience in developing and documenting complex Perl, Ruby and/or shell scripts. Demonstrated experience in developing and documenting C, C++, or Java programs. Demonstrated experience in Unix systems use and general shell programming. Demonstrated experience in Java web applicaiton development and support (such as Tomcat or JBoss).
Preferred License or Certification:	
If yes, what is the **preferred** License or Certification.	
Preferred Knowledge, Skills and Abilities:	
Preferred Computer Applications:	

Level of Independent Activity (for Staff Positions)

In terms of overall job responsibilities, to what degree does an incumbent determine own work priorities? Choose one.	Determines priorities more than 50% of time
If yes, please explain, include what policies, formal guidelines or government regulations are involved:	
To what degree does this position require decision-making or problem solving skills, as a primary function of the position? (This normally involves analyzing and synthesizing complex	Regular and recurring requirement



information or recommendations from several sources). Choose one.	
Does this position have budgetary responsibilities (Normally, would not include single function duties such as data entry or data collection)?	No
If yes, please describe the extent of these responsibilities to include number and dollar value of accounts and/or grants. For grants positions, designate pre and/or post award.	

Contacts of Position (for Staff Positions)

3 Records

Offices or Organizations	Purpose of Contact	Level of Contact	Frequency of Contact
UVA faculty, students, staff	Communication regarding project support	Inside UVA	Weekly
Colleagues at similar institutions or in related industry	Communication regarding tools, techniques, or applications	Outside UVA	As Needed
Software Engineer team within Library	Serves as a resource to other software engineers on complex problems. Will often train lower level software engineers on work processes and policies and assist management with developing their technical skills.	Inside UVA	As Needed

Reporting Relationships (for Staff Positions)

No Records Found

Working Conditions and Physical Requirements (for Staff Positions)

Environment

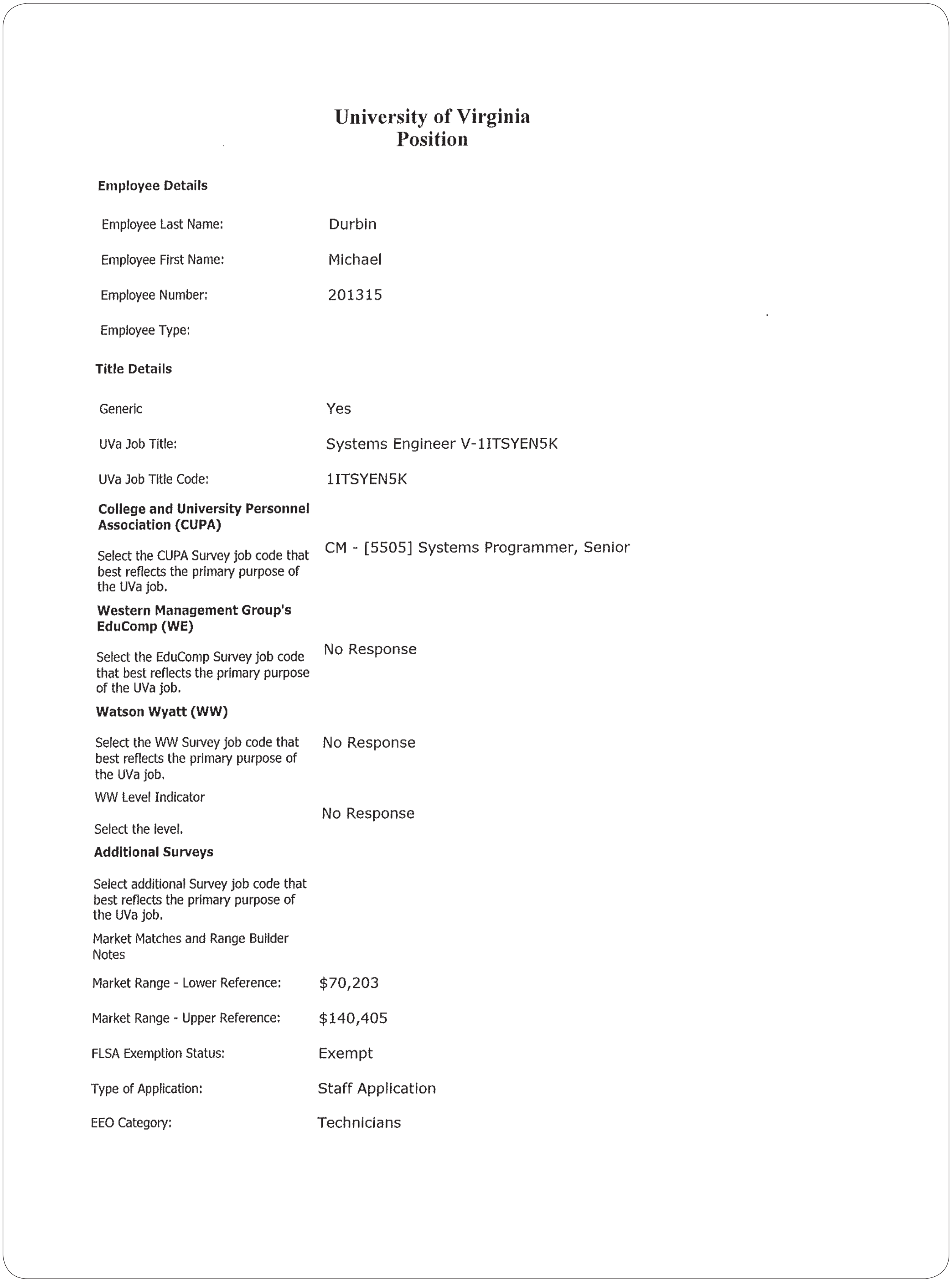
Check the appropriate box(es) that best describes the environment in which the primary function of the position is performed.	Office Environment
If you have indicated "Other Environment", if work tasks involve one or more of the above, or if further explanation is necessary, please use the space provided:	

Working Conditions & Exposures

Please describe, in more detail, any of the conditions answered with "Yes"

Physical Requirements

Describe any of the conditions selected above that are in excess of 26% time	Sitting at a desk, working at a computer.



University of Virginia
Position

Employee Details

Employee Last Name:	Durbin
Employee First Name:	Michael
Employee Number:	201315
Employee Type:	

Title Details

Generic	Yes
UVa Job Title:	Systems Engineer V-1ITSYEN5K
UVa Job Title Code:	1ITSYEN5K

College and University Personnel Association (CUPA)

Select the CUPA Survey job code that best reflects the primary purpose of the UVa job.	CM - [5505] Systems Programmer, Senior

Western Management Group's EduComp (WE)

Select the EduComp Survey job code that best reflects the primary purpose of the UVa job.	No Response

Watson Wyatt (WW)

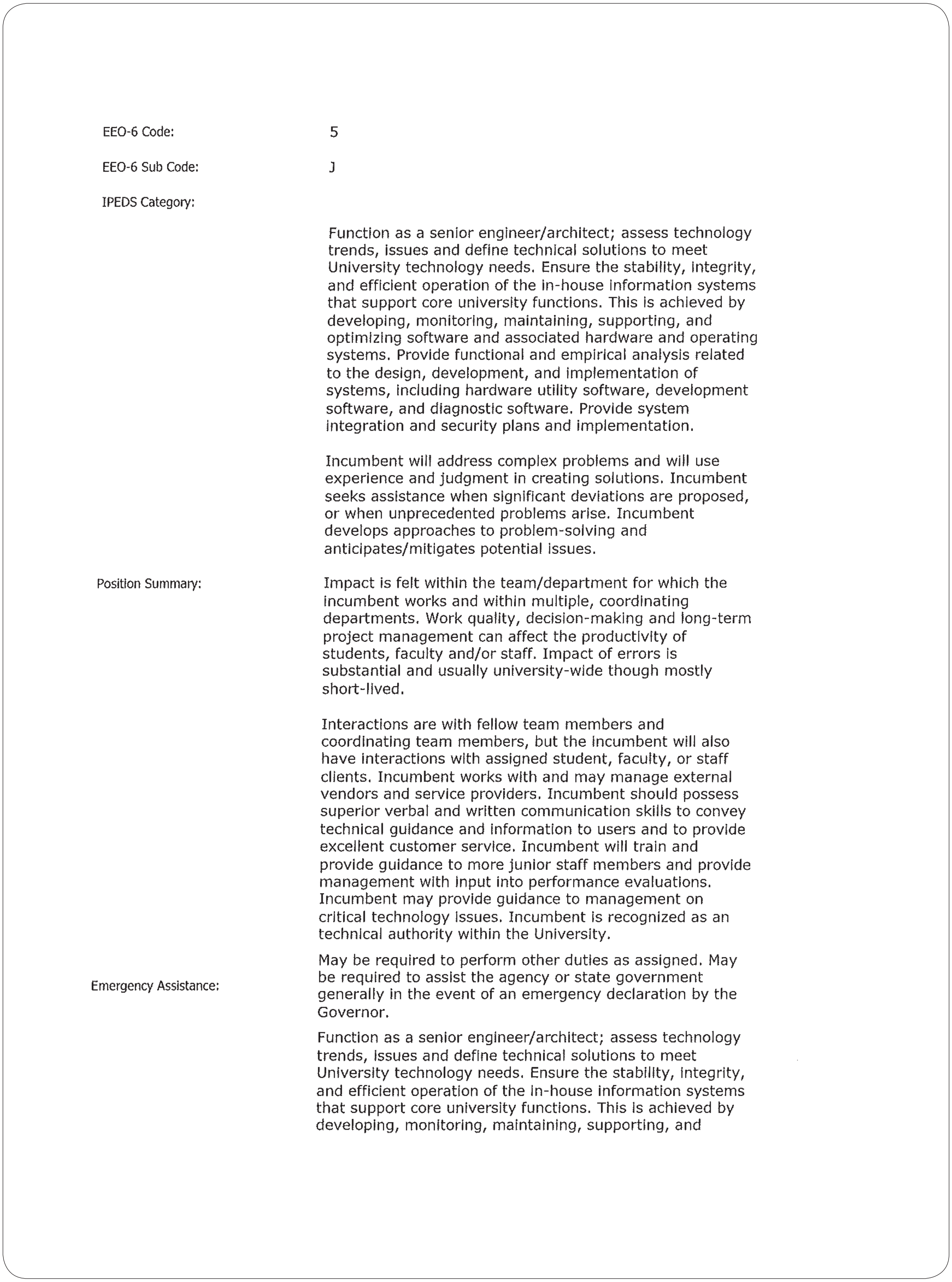
Select the WW Survey job code that best reflects the primary purpose of the UVa job.	No Response
WW Level Indicator Select the level.	No Response

Additional Surveys

Select additional Survey job code that best reflects the primary purpose of the UVa job.	
Market Matches and Range Builder Notes	
Market Range - Lower Reference:	$70,203
Market Range - Upper Reference:	$140,405
FLSA Exemption Status:	Exempt
Type of Application:	Staff Application
EEO Category:	Technicians

UNIVERSITY OF VIRGINIA

Systems Engineer V



EEO-6 Code: 5

EEO-6 Sub Code: J

IPEDS Category:

Position Summary:

Function as a senior engineer/architect; assess technology trends, issues and define technical solutions to meet University technology needs. Ensure the stability, integrity, and efficient operation of the in-house information systems that support core university functions. This is achieved by developing, monitoring, maintaining, supporting, and optimizing software and associated hardware and operating systems. Provide functional and empirical analysis related to the design, development, and implementation of systems, including hardware utility software, development software, and diagnostic software. Provide system integration and security plans and implementation.

Incumbent will address complex problems and will use experience and judgment in creating solutions. Incumbent seeks assistance when significant deviations are proposed, or when unprecedented problems arise. Incumbent develops approaches to problem-solving and anticipates/mitigates potential issues.

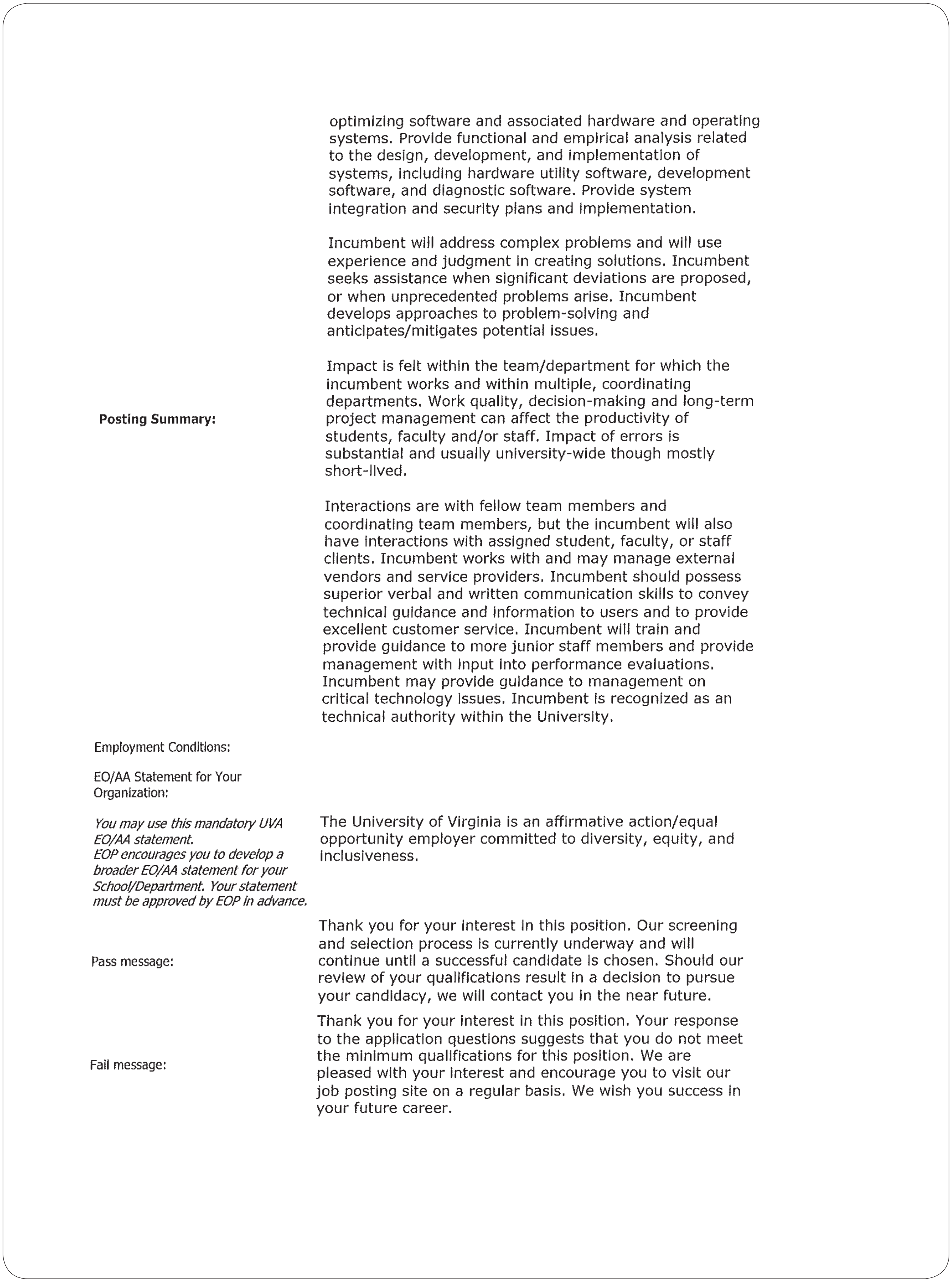
Impact is felt within the team/department for which the incumbent works and within multiple, coordinating departments. Work quality, decision-making and long-term project management can affect the productivity of students, faculty and/or staff. Impact of errors is substantial and usually university-wide though mostly short-lived.

Interactions are with fellow team members and coordinating team members, but the incumbent will also have interactions with assigned student, faculty, or staff clients. Incumbent works with and may manage external vendors and service providers. Incumbent should possess superior verbal and written communication skills to convey technical guidance and information to users and to provide excellent customer service. Incumbent will train and provide guidance to more junior staff members and provide management with input into performance evaluations. Incumbent may provide guidance to management on critical technology issues. Incumbent is recognized as an technical authority within the University.

Emergency Assistance:

May be required to perform other duties as assigned. May be required to assist the agency or state government generally in the event of an emergency declaration by the Governor.

Function as a senior engineer/architect; assess technology trends, issues and define technical solutions to meet University technology needs. Ensure the stability, integrity, and efficient operation of the in-house information systems that support core university functions. This is achieved by developing, monitoring, maintaining, supporting, and



Posting Summary:

optimizing software and associated hardware and operating systems. Provide functional and empirical analysis related to the design, development, and implementation of systems, including hardware utility software, development software, and diagnostic software. Provide system integration and security plans and implementation.

Incumbent will address complex problems and will use experience and judgment in creating solutions. Incumbent seeks assistance when significant deviations are proposed, or when unprecedented problems arise. Incumbent develops approaches to problem-solving and anticipates/mitigates potential issues.

Impact is felt within the team/department for which the incumbent works and within multiple, coordinating departments. Work quality, decision-making and long-term project management can affect the productivity of students, faculty and/or staff. Impact of errors is substantial and usually university-wide though mostly short-lived.

Interactions are with fellow team members and coordinating team members, but the incumbent will also have interactions with assigned student, faculty, or staff clients. Incumbent works with and may manage external vendors and service providers. Incumbent should possess superior verbal and written communication skills to convey technical guidance and information to users and to provide excellent customer service. Incumbent will train and provide guidance to more junior staff members and provide management with input into performance evaluations. Incumbent may provide guidance to management on critical technology issues. Incumbent is recognized as an technical authority within the University.

Employment Conditions:

EO/AA Statement for Your Organization:

You may use this mandatory UVA EO/AA statement.
EOP encourages you to develop a broader EO/AA statement for your School/Department. Your statement must be approved by EOP in advance.

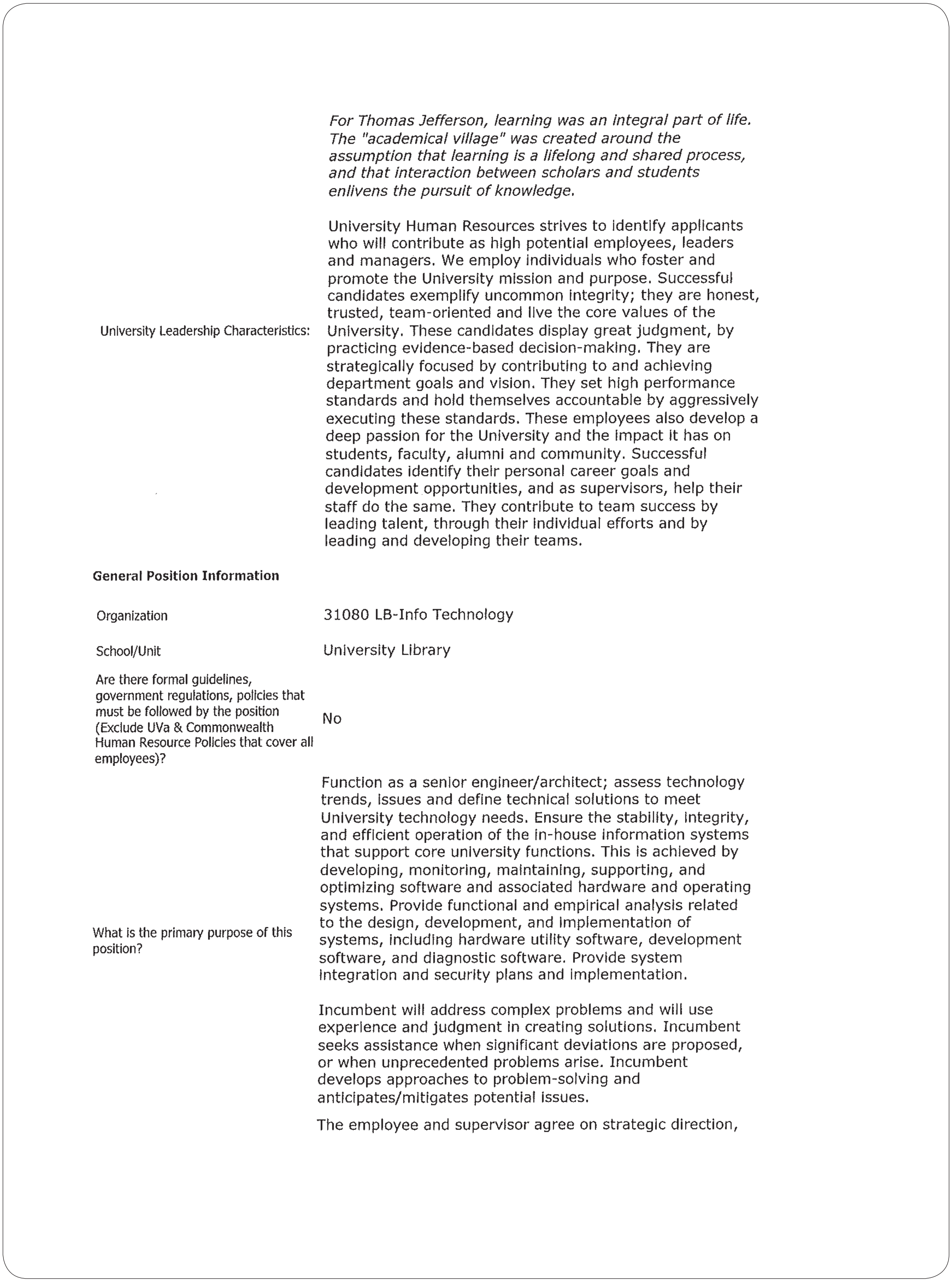
The University of Virginia is an affirmative action/equal opportunity employer committed to diversity, equity, and inclusiveness.

Pass message:

Thank you for your interest in this position. Our screening and selection process is currently underway and will continue until a successful candidate is chosen. Should our review of your qualifications result in a decision to pursue your candidacy, we will contact you in the near future.

Fail message:

Thank you for your interest in this position. Your response to the application questions suggests that you do not meet the minimum qualifications for this position. We are pleased with your interest and encourage you to visit our job posting site on a regular basis. We wish you success in your future career.

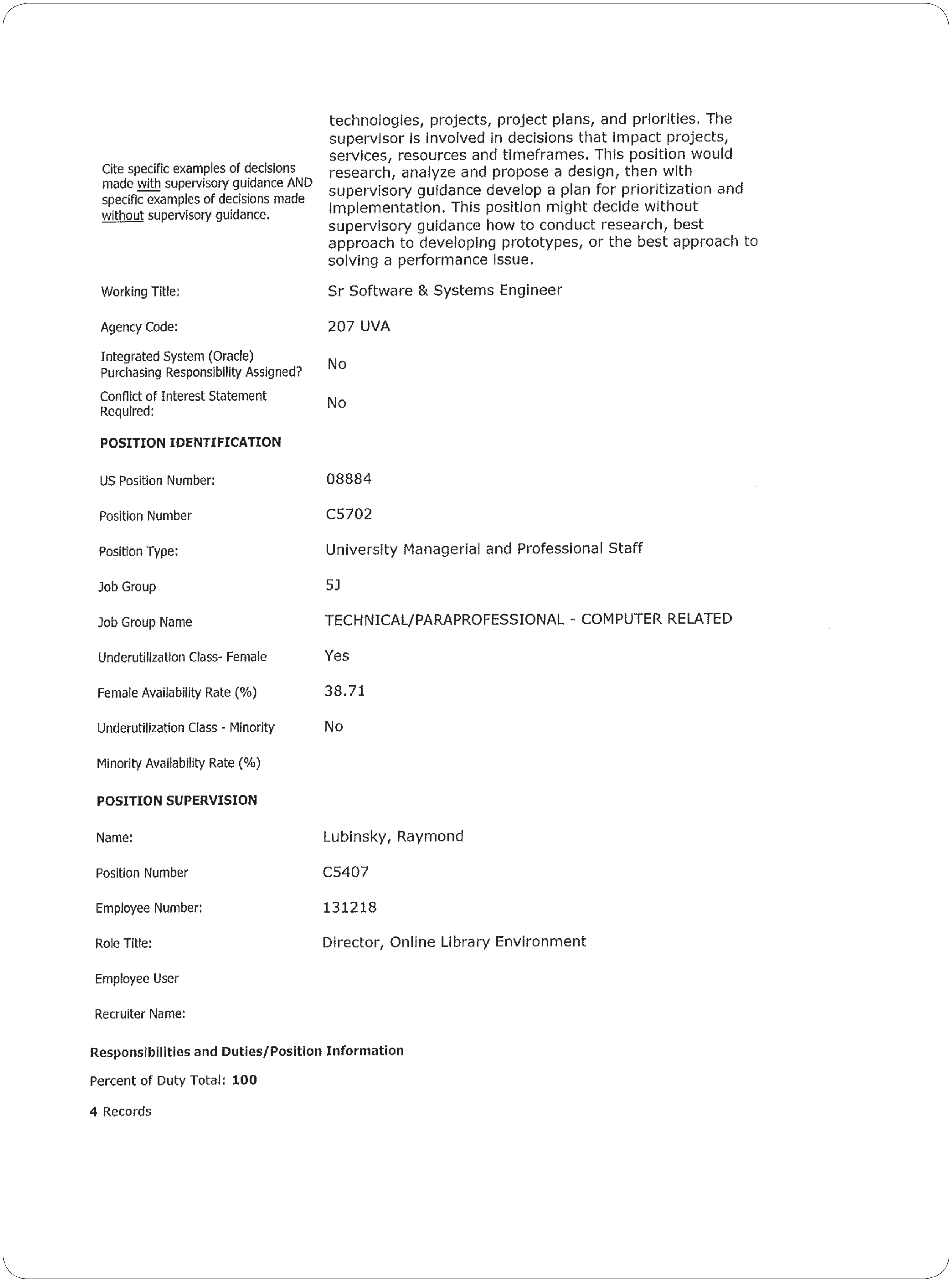


For Thomas Jefferson, learning was an integral part of life. The "academical village" was created around the assumption that learning is a lifelong and shared process, and that interaction between scholars and students enlivens the pursuit of knowledge.

University Leadership Characteristics:	University Human Resources strives to identify applicants who will contribute as high potential employees, leaders and managers. We employ individuals who foster and promote the University mission and purpose. Successful candidates exemplify uncommon integrity; they are honest, trusted, team-oriented and live the core values of the University. These candidates display great judgment, by practicing evidence-based decision-making. They are strategically focused by contributing to and achieving department goals and vision. They set high performance standards and hold themselves accountable by aggressively executing these standards. These employees also develop a deep passion for the University and the impact it has on students, faculty, alumni and community. Successful candidates identify their personal career goals and development opportunities, and as supervisors, help their staff do the same. They contribute to team success by leading talent, through their individual efforts and by leading and developing their teams.

General Position Information

Organization	31080 LB-Info Technology
School/Unit	University Library
Are there formal guidelines, government regulations, policies that must be followed by the position (Exclude UVa & Commonwealth Human Resource Policies that cover all employees)?	No
What is the primary purpose of this position?	Function as a senior engineer/architect; assess technology trends, issues and define technical solutions to meet University technology needs. Ensure the stability, integrity, and efficient operation of the in-house information systems that support core university functions. This is achieved by developing, monitoring, maintaining, supporting, and optimizing software and associated hardware and operating systems. Provide functional and empirical analysis related to the design, development, and implementation of systems, including hardware utility software, development software, and diagnostic software. Provide system integration and security plans and implementation. Incumbent will address complex problems and will use experience and judgment in creating solutions. Incumbent seeks assistance when significant deviations are proposed, or when unprecedented problems arise. Incumbent develops approaches to problem-solving and anticipates/mitigates potential issues. The employee and supervisor agree on strategic direction,



Cite specific examples of decisions made with supervisory guidance AND specific examples of decisions made without supervisory guidance.	technologies, projects, project plans, and priorities. The supervisor is involved in decisions that impact projects, services, resources and timeframes. This position would research, analyze and propose a design, then with supervisory guidance develop a plan for prioritization and implementation. This position might decide without supervisory guidance how to conduct research, best approach to developing prototypes, or the best approach to solving a performance issue.
Working Title:	Sr Software & Systems Engineer
Agency Code:	207 UVA
Integrated System (Oracle) Purchasing Responsibility Assigned?	No
Conflict of Interest Statement Required:	No

POSITION IDENTIFICATION

US Position Number:	08884
Position Number	C5702
Position Type:	University Managerial and Professional Staff
Job Group	5J
Job Group Name	TECHNICAL/PARAPROFESSIONAL - COMPUTER RELATED
Underutilization Class- Female	Yes
Female Availability Rate (%)	38.71
Underutilization Class - Minority	No
Minority Availability Rate (%)	

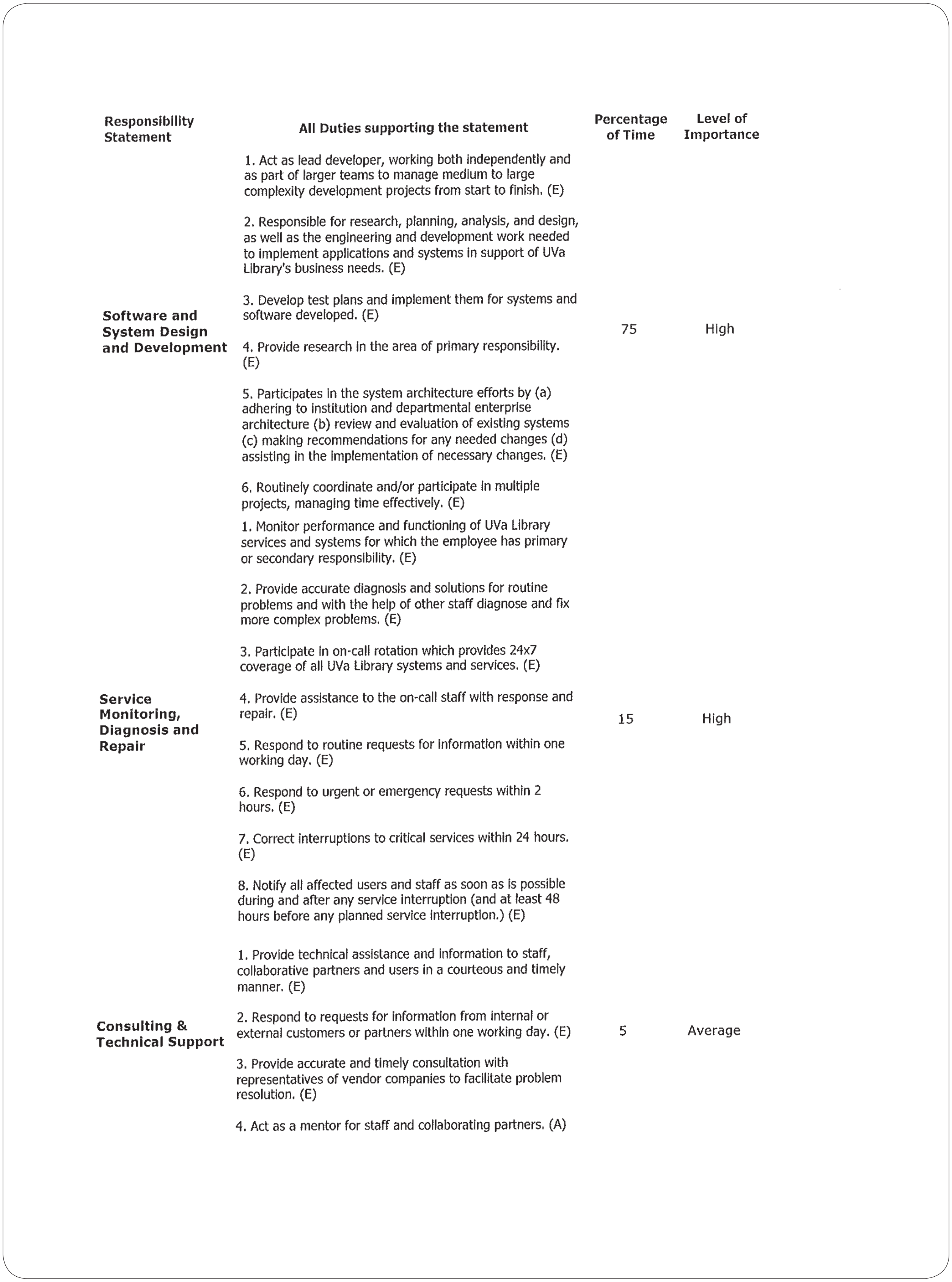
POSITION SUPERVISION

Name:	Lubinsky, Raymond
Position Number	C5407
Employee Number:	131218
Role Title:	Director, Online Library Environment
Employee User	
Recruiter Name:	

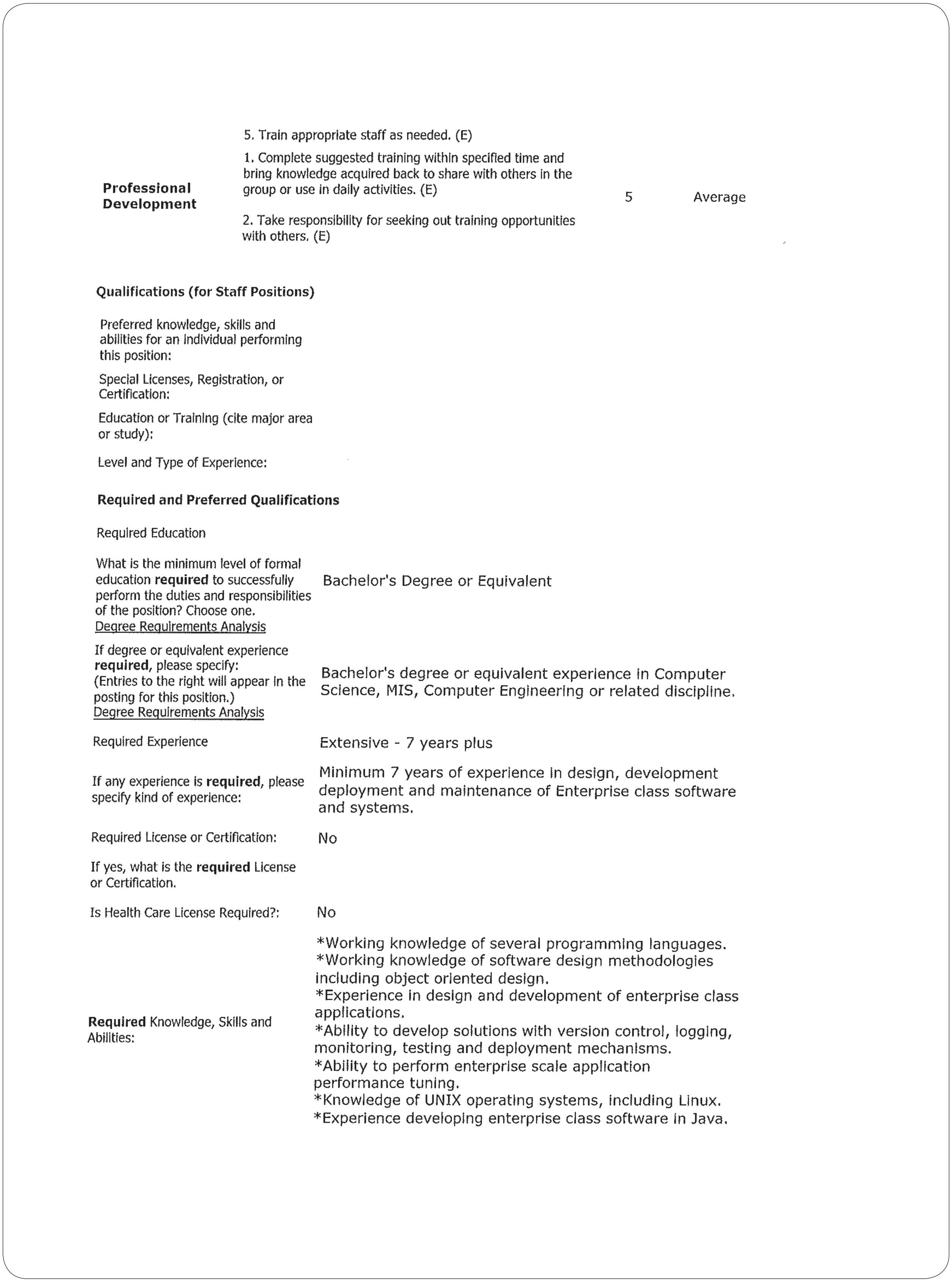
Responsibilities and Duties/Position Information

Percent of Duty Total: **100**

4 Records



Responsibility Statement	All Duties supporting the statement	Percentage of Time	Level of Importance
Software and System Design and Development	1. Act as lead developer, working both independently and as part of larger teams to manage medium to large complexity development projects from start to finish. (E) 2. Responsible for research, planning, analysis, and design, as well as the engineering and development work needed to implement applications and systems in support of UVa Library's business needs. (E) 3. Develop test plans and implement them for systems and software developed. (E) 4. Provide research in the area of primary responsibility. (E) 5. Participates in the system architecture efforts by (a) adhering to institution and departmental enterprise architecture (b) review and evaluation of existing systems (c) making recommendations for any needed changes (d) assisting in the implementation of necessary changes. (E) 6. Routinely coordinate and/or participate in multiple projects, managing time effectively. (E)	75	High
Service Monitoring, Diagnosis and Repair	1. Monitor performance and functioning of UVa Library services and systems for which the employee has primary or secondary responsibility. (E) 2. Provide accurate diagnosis and solutions for routine problems and with the help of other staff diagnose and fix more complex problems. (E) 3. Participate in on-call rotation which provides 24x7 coverage of all UVa Library systems and services. (E) 4. Provide assistance to the on-call staff with response and repair. (E) 5. Respond to routine requests for information within one working day. (E) 6. Respond to urgent or emergency requests within 2 hours. (E) 7. Correct interruptions to critical services within 24 hours. (E) 8. Notify all affected users and staff as soon as is possible during and after any service interruption (and at least 48 hours before any planned service interruption.) (E)	15	High
Consulting & Technical Support	1. Provide technical assistance and information to staff, collaborative partners and users in a courteous and timely manner. (E) 2. Respond to requests for information from internal or external customers or partners within one working day. (E) 3. Provide accurate and timely consultation with representatives of vendor companies to facilitate problem resolution. (E) 4. Act as a mentor for staff and collaborating partners. (A)	5	Average



	5. Train appropriate staff as needed. (E)		
Professional Development	1. Complete suggested training within specified time and bring knowledge acquired back to share with others in the group or use in daily activities. (E) 2. Take responsibility for seeking out training opportunities with others. (E)	5	Average

Qualifications (for Staff Positions)

Preferred knowledge, skills and abilities for an individual performing this position:

Special Licenses, Registration, or Certification:

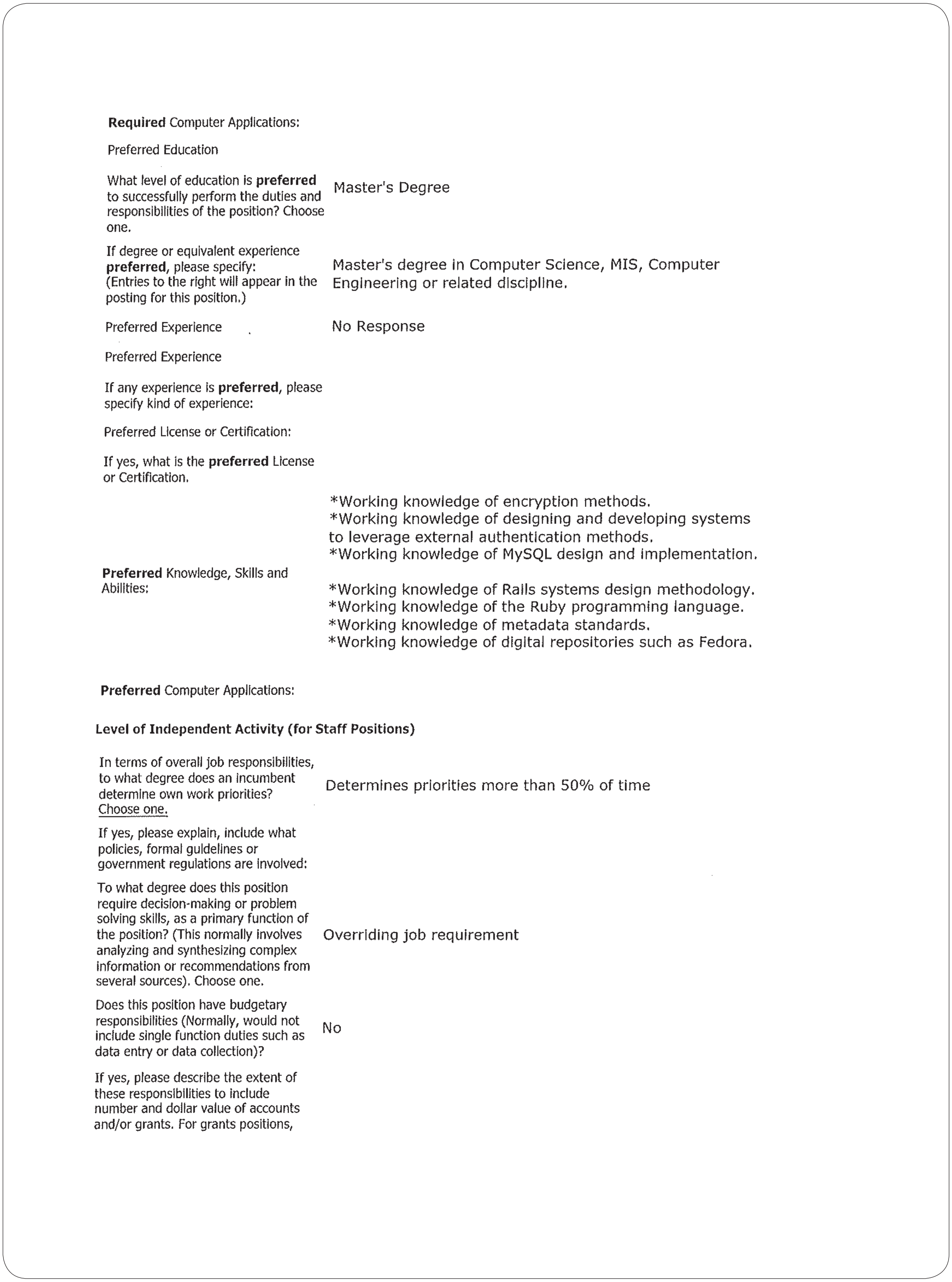
Education or Training (cite major area or study):

Level and Type of Experience:

Required and Preferred Qualifications

Required Education

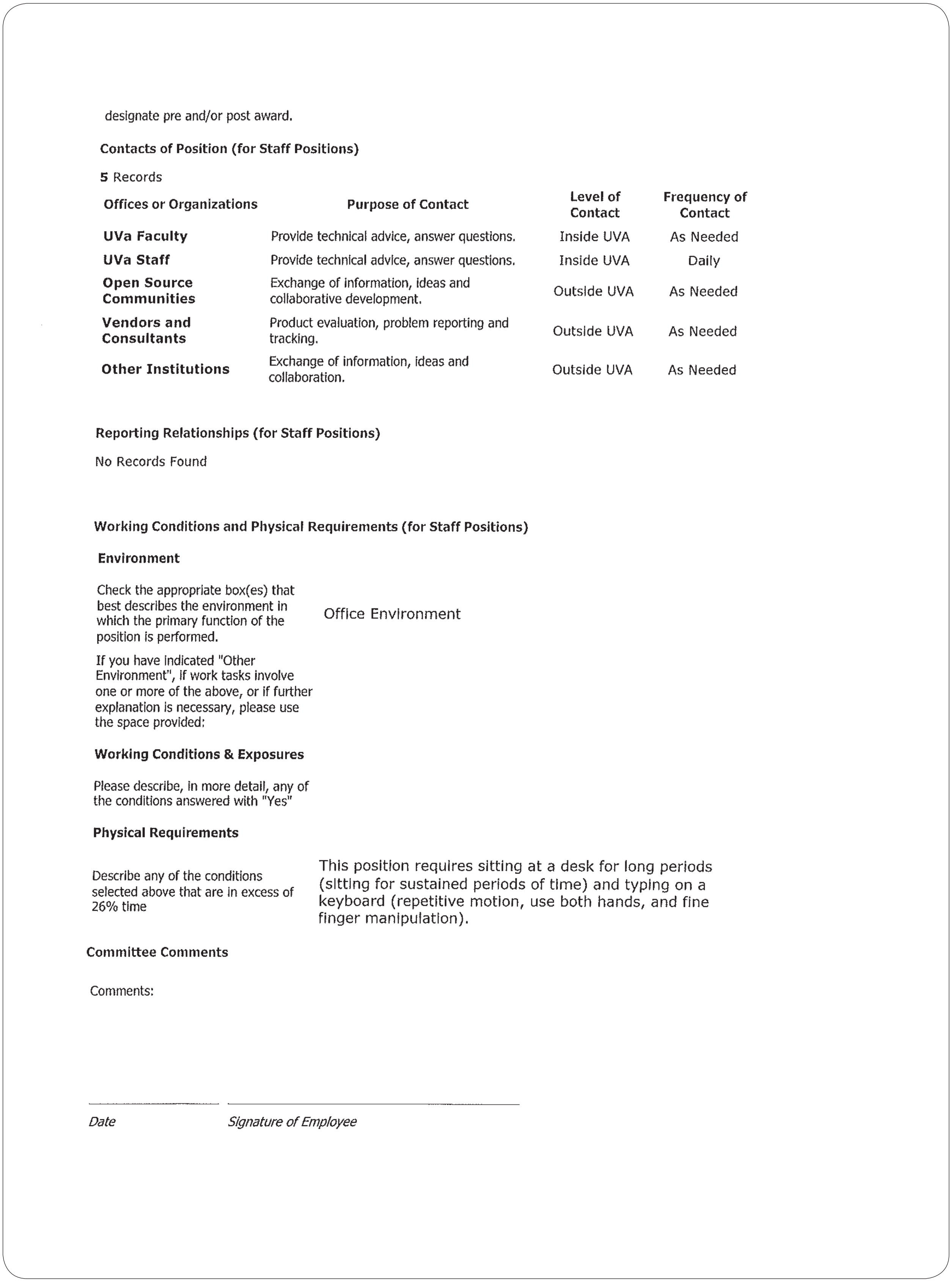
What is the minimum level of formal education **required** to successfully perform the duties and responsibilities of the position? Choose one. Degree Requirements Analysis	Bachelor's Degree or Equivalent
If degree or equivalent experience **required**, please specify: (Entries to the right will appear in the posting for this position.) Degree Requirements Analysis	Bachelor's degree or equivalent experience in Computer Science, MIS, Computer Engineering or related discipline.
Required Experience	Extensive - 7 years plus
If any experience is **required**, please specify kind of experience:	Minimum 7 years of experience in design, development deployment and maintenance of Enterprise class software and systems.
Required License or Certification:	No
If yes, what is the **required** License or Certification.	
Is Health Care License Required?:	No
Required Knowledge, Skills and Abilities:	*Working knowledge of several programming languages. *Working knowledge of software design methodologies including object oriented design. *Experience in design and development of enterprise class applications. *Ability to develop solutions with version control, logging, monitoring, testing and deployment mechanisms. *Ability to perform enterprise scale application performance tuning. *Knowledge of UNIX operating systems, including Linux. *Experience developing enterprise class software in Java.



Required Computer Applications:	
Preferred Education	
What level of education is **preferred** to successfully perform the duties and responsibilities of the position? Choose one.	Master's Degree
If degree or equivalent experience **preferred**, please specify: (Entries to the right will appear in the posting for this position.)	Master's degree in Computer Science, MIS, Computer Engineering or related discipline.
Preferred Experience	No Response
Preferred Experience	
If any experience is **preferred**, please specify kind of experience:	
Preferred License or Certification:	
If yes, what is the **preferred** License or Certification.	
Preferred Knowledge, Skills and Abilities:	*Working knowledge of encryption methods. *Working knowledge of designing and developing systems to leverage external authentication methods. *Working knowledge of MySQL design and implementation. *Working knowledge of Rails systems design methodology. *Working knowledge of the Ruby programming language. *Working knowledge of metadata standards. *Working knowledge of digital repositories such as Fedora.
Preferred Computer Applications:	

Level of Independent Activity (for Staff Positions)

In terms of overall job responsibilities, to what degree does an incumbent determine own work priorities? Choose one.	Determines priorities more than 50% of time
If yes, please explain, include what policies, formal guidelines or government regulations are involved:	
To what degree does this position require decision-making or problem solving skills, as a primary function of the position? (This normally involves analyzing and synthesizing complex information or recommendations from several sources). Choose one.	Overriding job requirement
Does this position have budgetary responsibilities (Normally, would not include single function duties such as data entry or data collection)?	No
If yes, please describe the extent of these responsibilities to include number and dollar value of accounts and/or grants. For grants positions,	



designate pre and/or post award.

Contacts of Position (for Staff Positions)

5 Records

Offices or Organizations	Purpose of Contact	Level of Contact	Frequency of Contact
UVa Faculty	Provide technical advice, answer questions.	Inside UVA	As Needed
UVa Staff	Provide technical advice, answer questions.	Inside UVA	Daily
Open Source Communities	Exchange of information, ideas and collaborative development.	Outside UVA	As Needed
Vendors and Consultants	Product evaluation, problem reporting and tracking.	Outside UVA	As Needed
Other Institutions	Exchange of information, ideas and collaboration.	Outside UVA	As Needed

Reporting Relationships (for Staff Positions)

No Records Found

Working Conditions and Physical Requirements (for Staff Positions)

Environment

Check the appropriate box(es) that best describes the environment in which the primary function of the position is performed.	Office Environment
If you have indicated "Other Environment", if work tasks involve one or more of the above, or if further explanation is necessary, please use the space provided:	

Working Conditions & Exposures

Please describe, in more detail, any of the conditions answered with "Yes"

Physical Requirements

Describe any of the conditions selected above that are in excess of 26% time	This position requires sitting at a desk for long periods (sitting for sustained periods of time) and typing on a keyboard (repetitive motion, use both hands, and fine finger manipulation).

Committee Comments

Comments:

______________ ______________________________

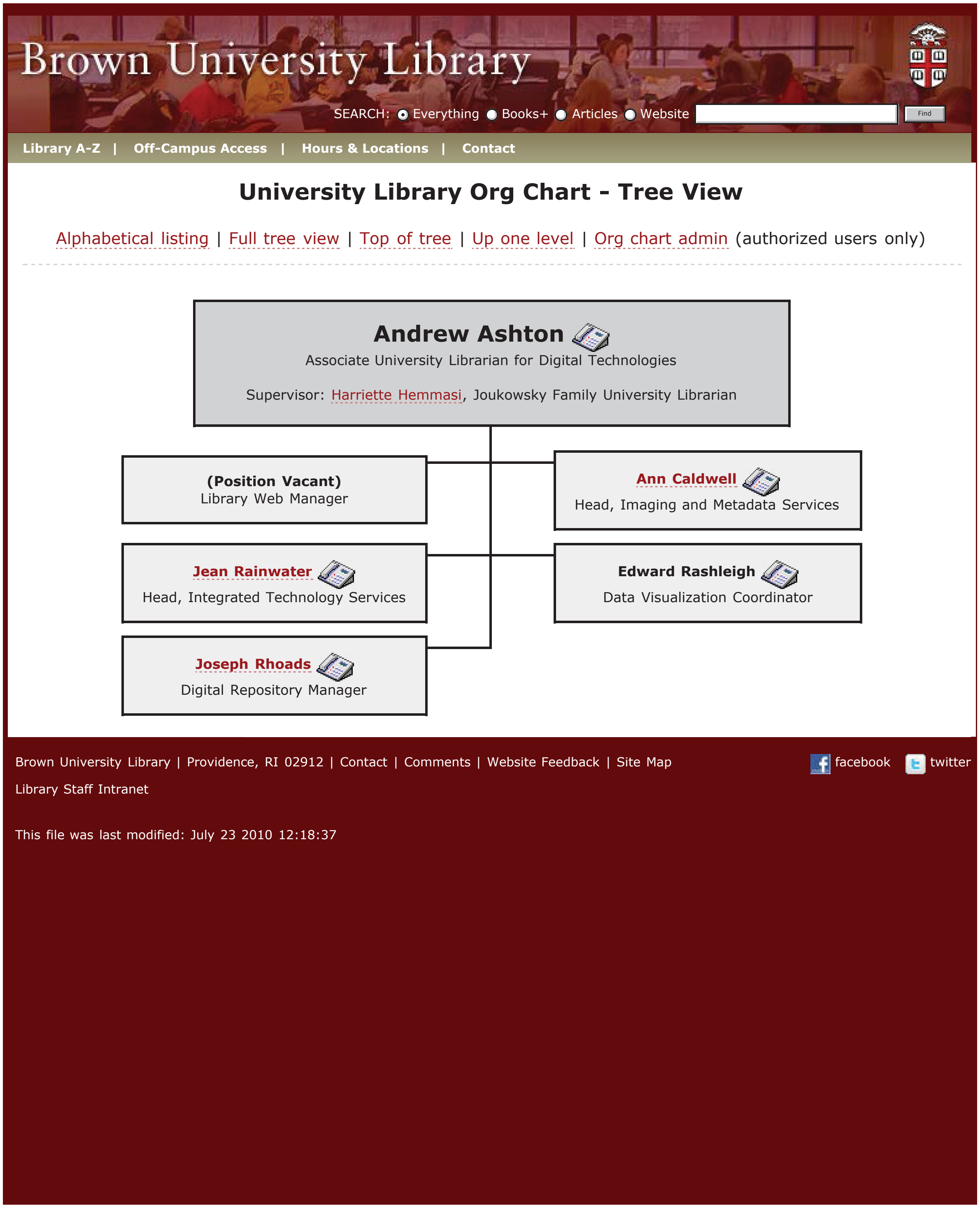
Date *Signature of Employee*

Organization Charts

BROWN UNIVERSITY LIBRARY

Digital Technologies

http://library.brown.edu/orgchart/tree.php?id=173



Brown University Library
SEARCH: ● Everything ○ Books+ ○ Articles ○ Website
Find
Library A-Z | Off-Campus Access | Hours & Locations | Contact

University Library Org Chart - Tree View

Alphabetical listing | Full tree view | Top of tree | Up one level | Org chart admin (authorized users only)

Andrew Ashton ☎
Associate University Librarian for Digital Technologies

Supervisor: Harriette Hemmasi, Joukowsky Family University Librarian

(Position Vacant)
Library Web Manager

Ann Caldwell ☎
Head, Imaging and Metadata Services

Jean Rainwater ☎
Head, Integrated Technology Services

Edward Rashleigh ☎
Data Visualization Coordinator

Joseph Rhoads ☎
Digital Repository Manager

Brown University Library | Providence, RI 02912 | Contact | Comments | Website Feedback | Site Map

Library Staff Intranet

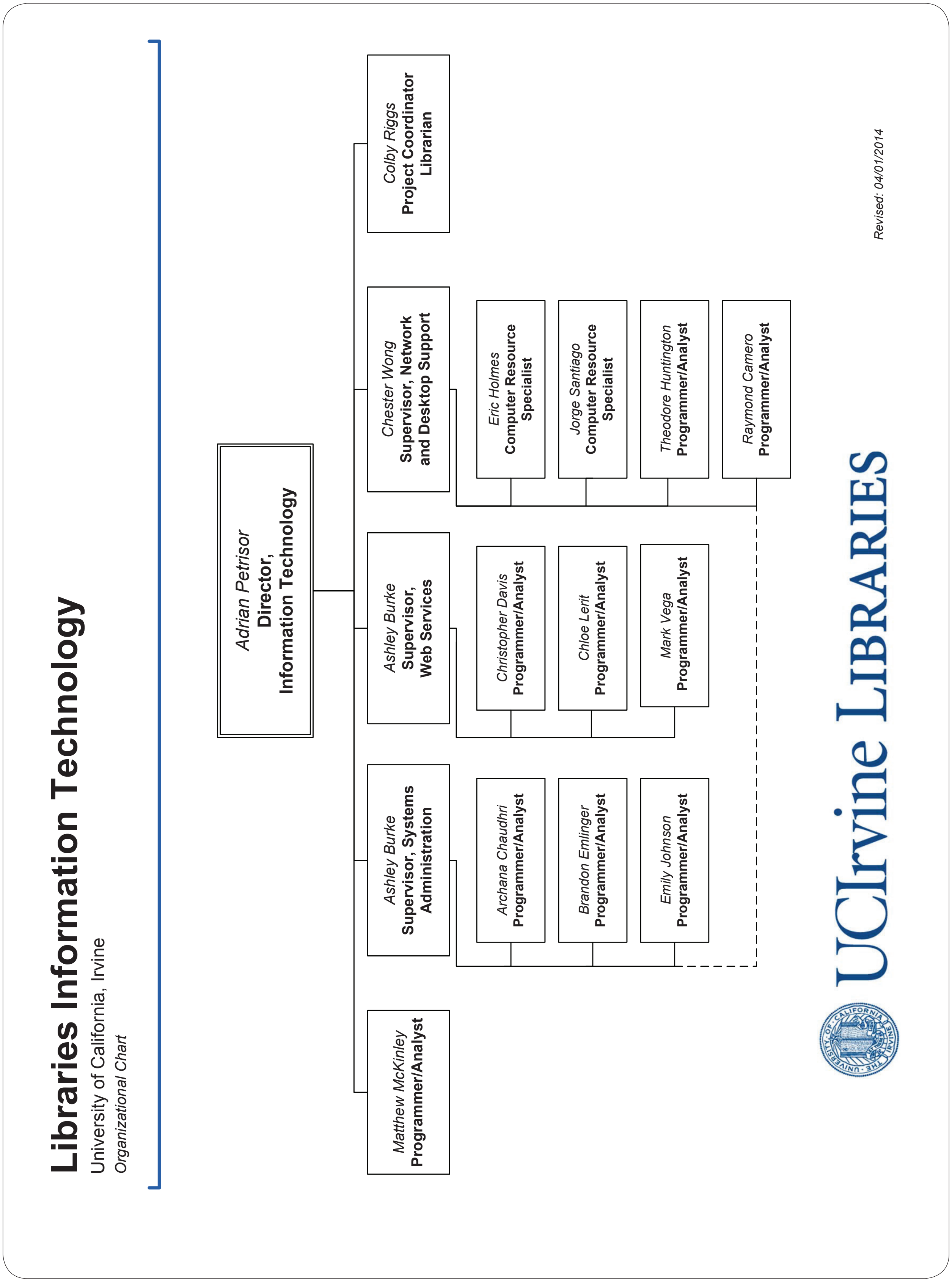
facebook twitter

This file was last modified: July 23 2010 12:18:37

UNIVERSITY OF CALIFORNIA, IRVINE LIBRARIES

Libraries Information Technology

http://staff.lib.uci.edu/departments/orgcharts/docs/it-org-chart.pdf



Libraries Information Technology

University of California, Irvine

Organizational Chart

Adrian Petrisor
Director,
Information Technology

Matthew McKinley
Programmer/Analyst

Ashley Burke
Supervisor, Systems Administration

- *Archana Chaudhri* **Programmer/Analyst**
- *Brandon Emlinger* **Programmer/Analyst**
- *Emily Johnson* **Programmer/Analyst**

Ashley Burke
Supervisor, Web Services

- *Christopher Davis* **Programmer/Analyst**
- *Chloe Lerit* **Programmer/Analyst**
- *Mark Vega* **Programmer/Analyst**

Chester Wong
Supervisor, Network and Desktop Support

- *Eric Holmes* **Computer Resource Specialist**
- *Jorge Santiago* **Computer Resource Specialist**
- *Theodore Huntington* **Programmer/Analyst**
- *Raymond Camero* **Programmer/Analyst**

Colby Riggs
Project Coordinator Librarian

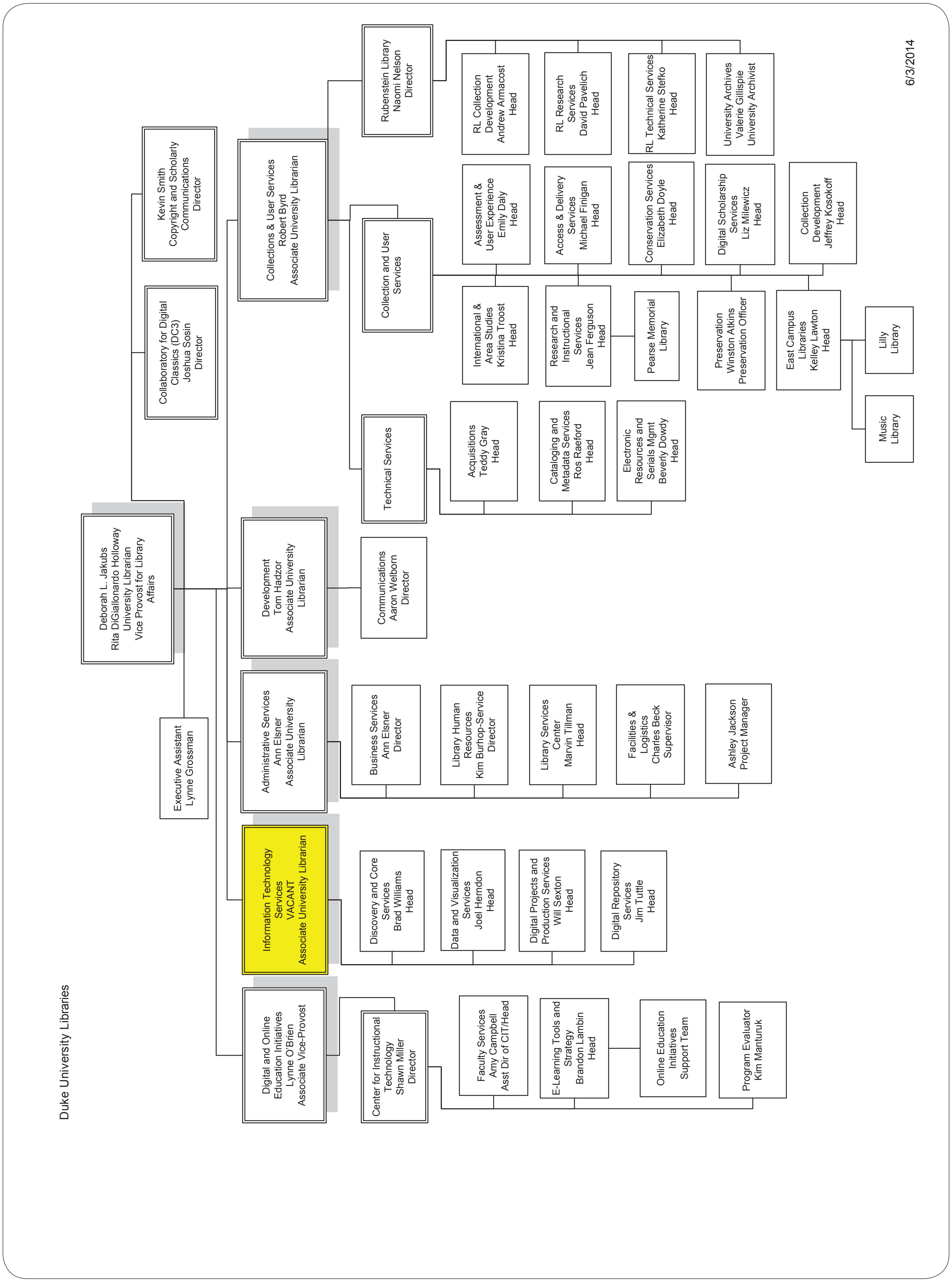
Revised: 04/01/2014

UCIrvine LIBRARIES

DUKE UNIVERSITY LIBRARIES

Org Chart

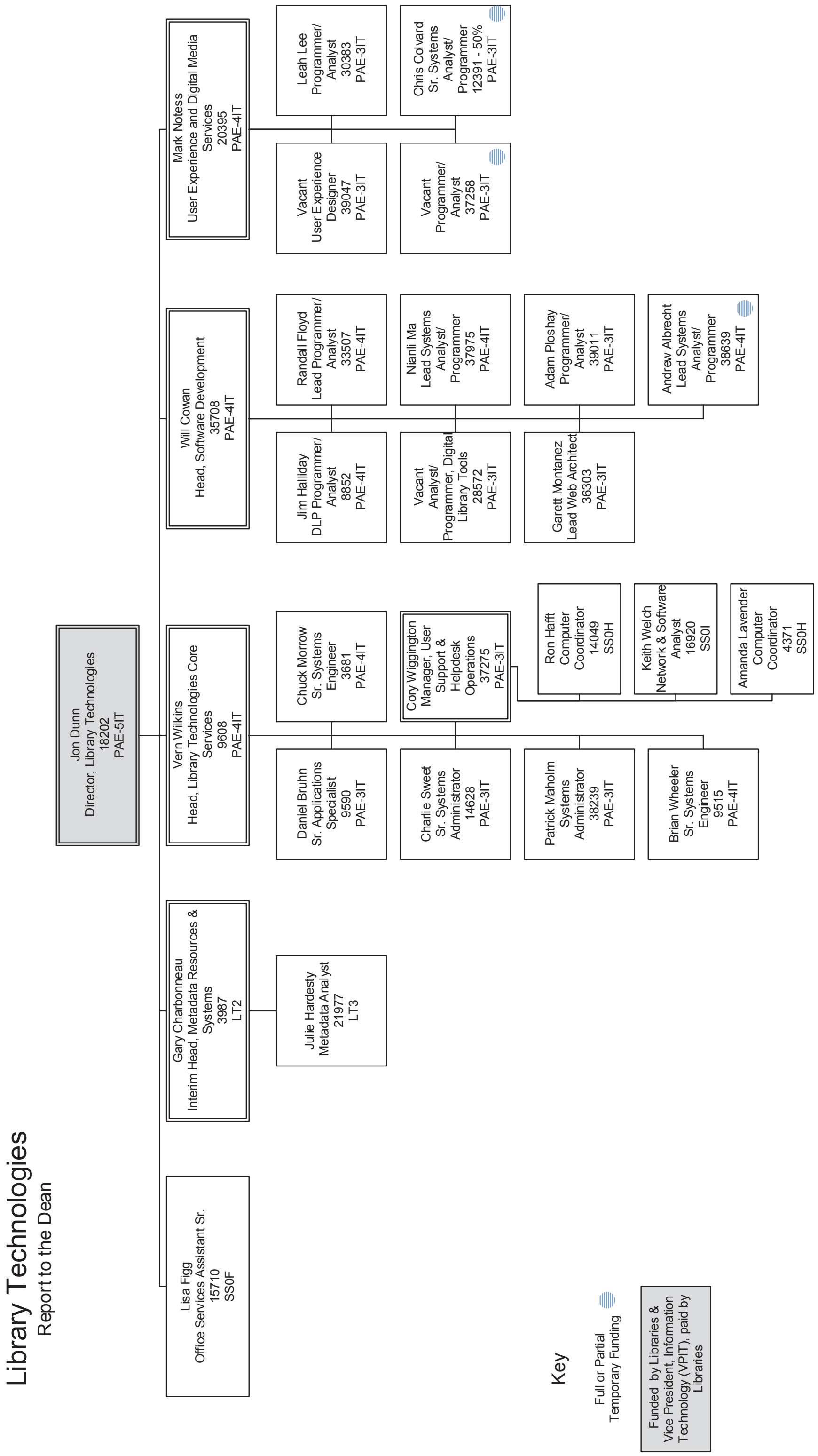
library.duke.edu/sites/default/files/dul/about/orgchart.pdf



INDIANA UNIVERSITY LIBRARIES BLOOMINGTON

Library Technologies

http://www.libraries.iub.edu/secure/defiles/LHR/IUB%20Libraries%202013_11.pdf



Library Technologies

Report to the Dean

▨ Jon Dunn, Director, Library Technologies, 18202, PAE-5IT

- Lisa Figg, Office Services Assistant Sr., 15710, SS0F
- Gary Charbonneau, Interim Head, Metadata Resources & Systems, 3987, LT2
  - Julie Hardesty, Metadata Analyst, 21977, LT3
- Vern Wilkins, Head, Library Technologies Core Services, 9608, PAE-4IT
  - Chuck Morrow, Sr. Systems Engineer, 3681, PAE-4IT
  - Cory Wiggington, Manager, User Support & Helpdesk Operations, 37275, PAE-3IT
    - Ron Hafft, Computer Coordinator, 14049, SS0H
    - Keith Welch, Network & Software Analyst, 16920, SS0I
    - Amanda Lavender, Computer Coordinator, 4371, SS0H
  - Daniel Bruhn, Sr. Applications Specialist, 9590, PAE-3IT
  - Charlie Sweet, Sr. Systems Administrator, 14628, PAE-3IT
  - Patrick Maholm, Systems Administrator, 38239, PAE-3IT
  - Brian Wheeler, Sr. Systems Engineer, 9515, PAE-4IT
- Will Cowan, Head, Software Development, 35708, PAE-4IT
  - Randall Floyd, Lead Programmer/Analyst, 33507, PAE-4IT
  - Nianli Ma, Lead Systems Analyst/Programmer, 37975, PAE-4IT
  - Adam Ploshay, Programmer/Analyst, 39011, PAE-3IT
  - Andrew Albrecht, Lead Systems Analyst/Programmer, 38639, PAE-4IT ●
  - Jim Halliday, DLP Programmer/Analyst, 8852, PAE-4IT
  - Vacant, Analyst/Programmer, Digital Library Tools, 28572, PAE-3IT
  - Garett Montanez, Lead Web Architect, 36303, PAE-3IT
- Mark Notess, User Experience and Digital Media Services, 20395, PAE-4IT
  - Leah Lee, Programmer/Analyst, 30383, PAE-3IT
  - Chris Colvard, Sr. Systems Analyst/Programmer, 12391 - 50%, PAE-3IT ●
  - Vacant, User Experience Designer, 39047, PAE-3IT
  - Vacant, Programmer/Analyst, 37258, PAE-3IT ●

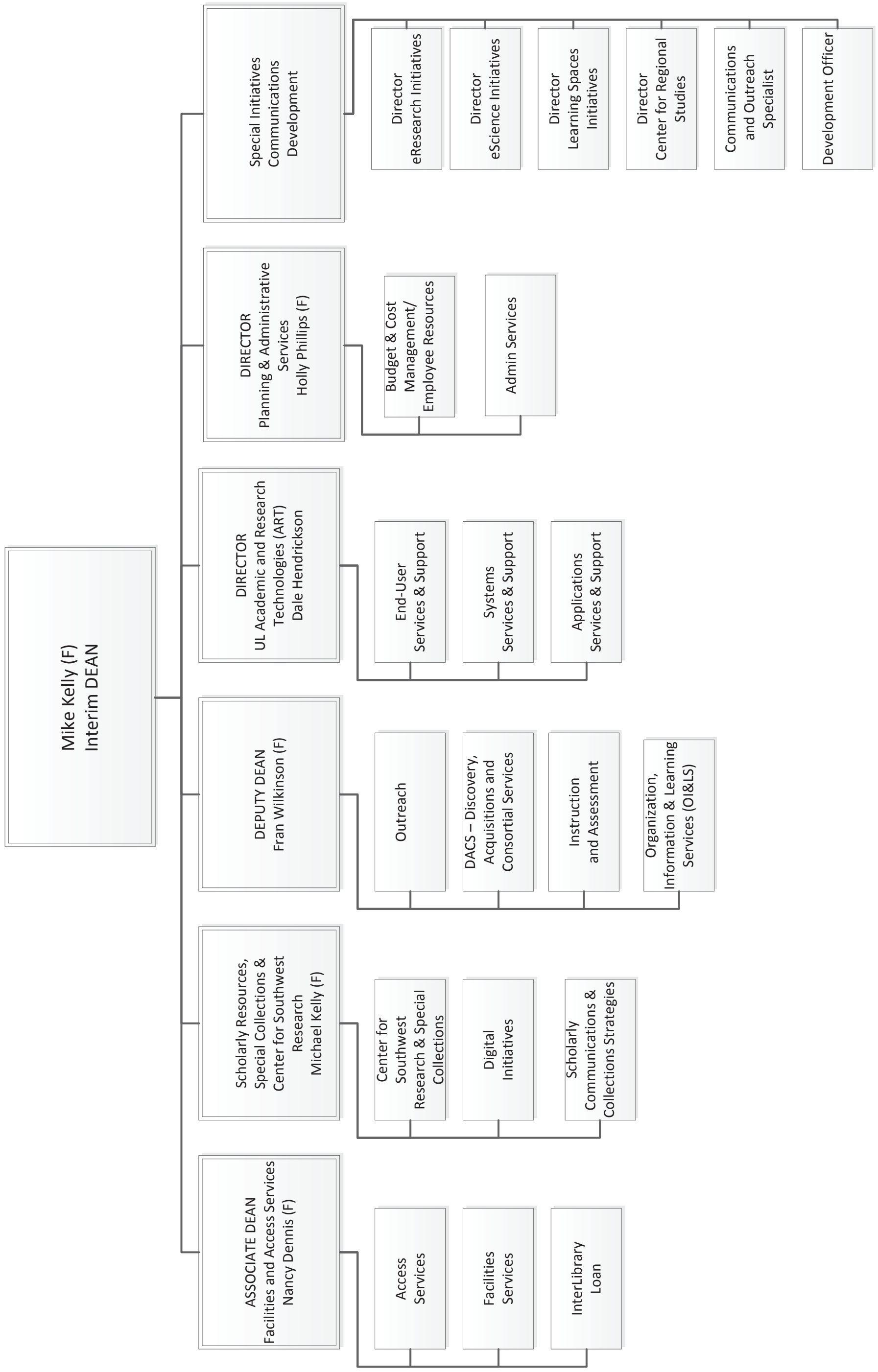
Key

● Full or Partial Temporary Funding

▨ Funded by Libraries & Vice President, Information Technology (VPIT), paid by Libraries

Overview-UL&LS Organizational Structure

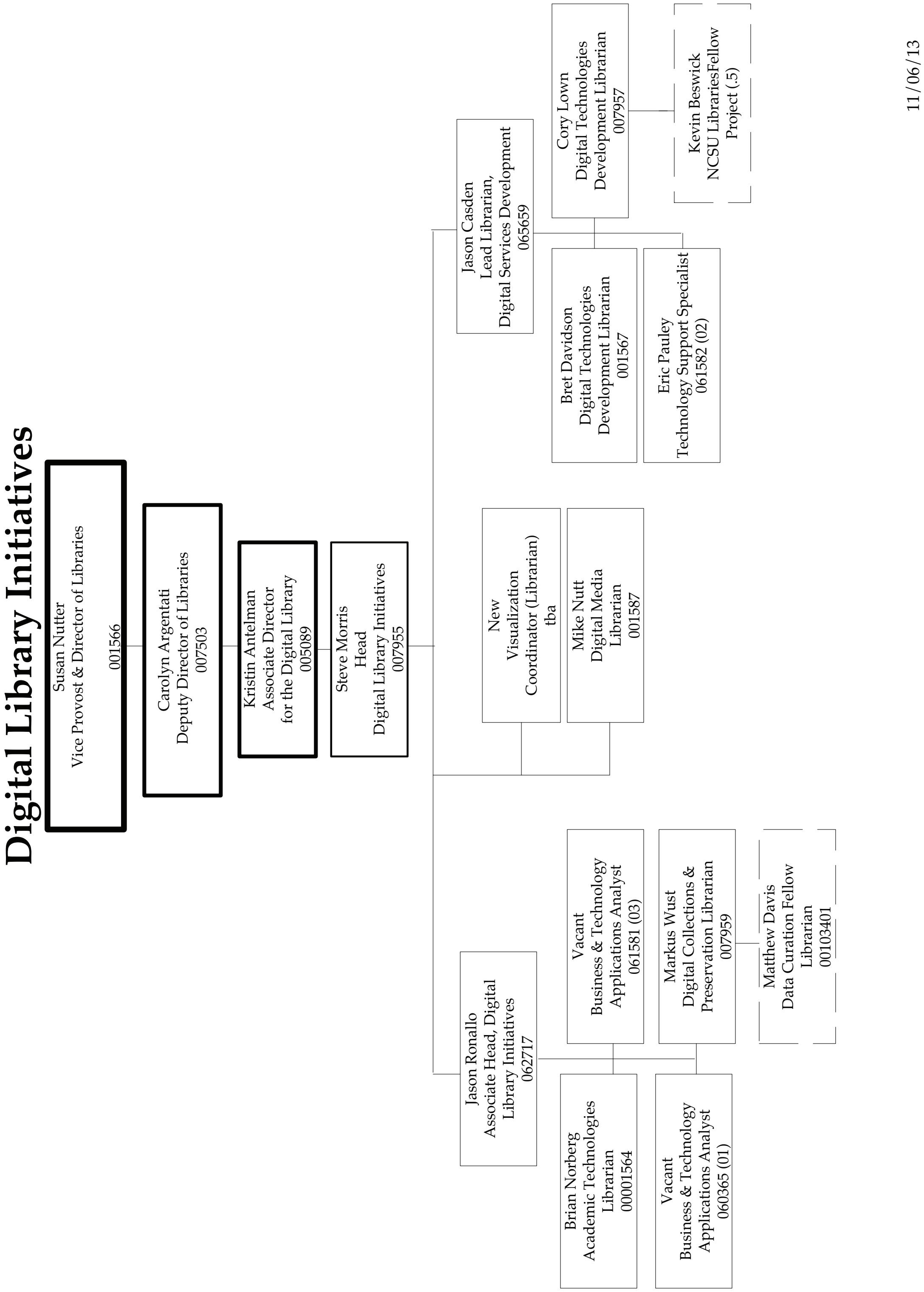
OVERVIEW-UL&LS ORGANIZATIONAL STRUCTURE


Mike Kelly (F)
Interim DEAN
ASSOCIATE DEAN
Facilities and Access Services
Nancy Dennis (F)
Access Services
Facilities Services
InterLibrary Loan
Scholarly Resources, Special Collections & Center for Southwest Research
Michael Kelly (F)
Center for Southwest Research & Special Collections
Digital Initiatives
Scholarly Communications & Collections Strategies
DEPUTY DEAN
Fran Wilkinson (F)
Outreach
DACS – Discovery, Acquisitions and Consortial Services
Instruction and Assessment
Organization, Information & Learning Services (OI&LS)
DIRECTOR
UL Academic and Research Technologies (ART)
Dale Hendrickson
End-User Services & Support
Systems Services & Support
Applications Services & Support
DIRECTOR
Planning & Administrative Services
Holly Phillips (F)
Budget & Cost Management/ Employee Resources
Admin Services
Special Initiatives
Communications
Development
Director
eResearch Initiatives
Director
eScience Initiatives
Director
Learning Spaces Initiatives
Director
Center for Regional Studies
Communications and Outreach Specialist
Development Officer

University of New Mexico Libraries Organization Chart

Revised on 11/5/2013

Page 1 of 12



Digital Library Initiatives

Susan Nutter
Vice Provost & Director of Libraries
001566

Carolyn Argentati
Deputy Director of Libraries
007503

Kristin Antelman
Associate Director
for the Digital Library
005089

Steve Morris
Head
Digital Library Initiatives
007955

Jason Ronallo
Associate Head, Digital
Library Initiatives
062717

Brian Norberg
Academic Technologies
Librarian
00001564

Vacant
Business & Technology
Applications Analyst
060365 (01)

Vacant
Business & Technology
Applications Analyst
061581 (03)

Markus Wust
Digital Collections &
Preservation Librarian
007959

Matthew Davis
Data Curation Fellow
Librarian
00103401

New
Visualization
Coordinator (Librarian)
tba

Mike Nutt
Digital Media
Librarian
001587

Jason Casden
Lead Librarian,
Digital Services Development
065659

Bret Davidson
Digital Technologies
Development Librarian
001567

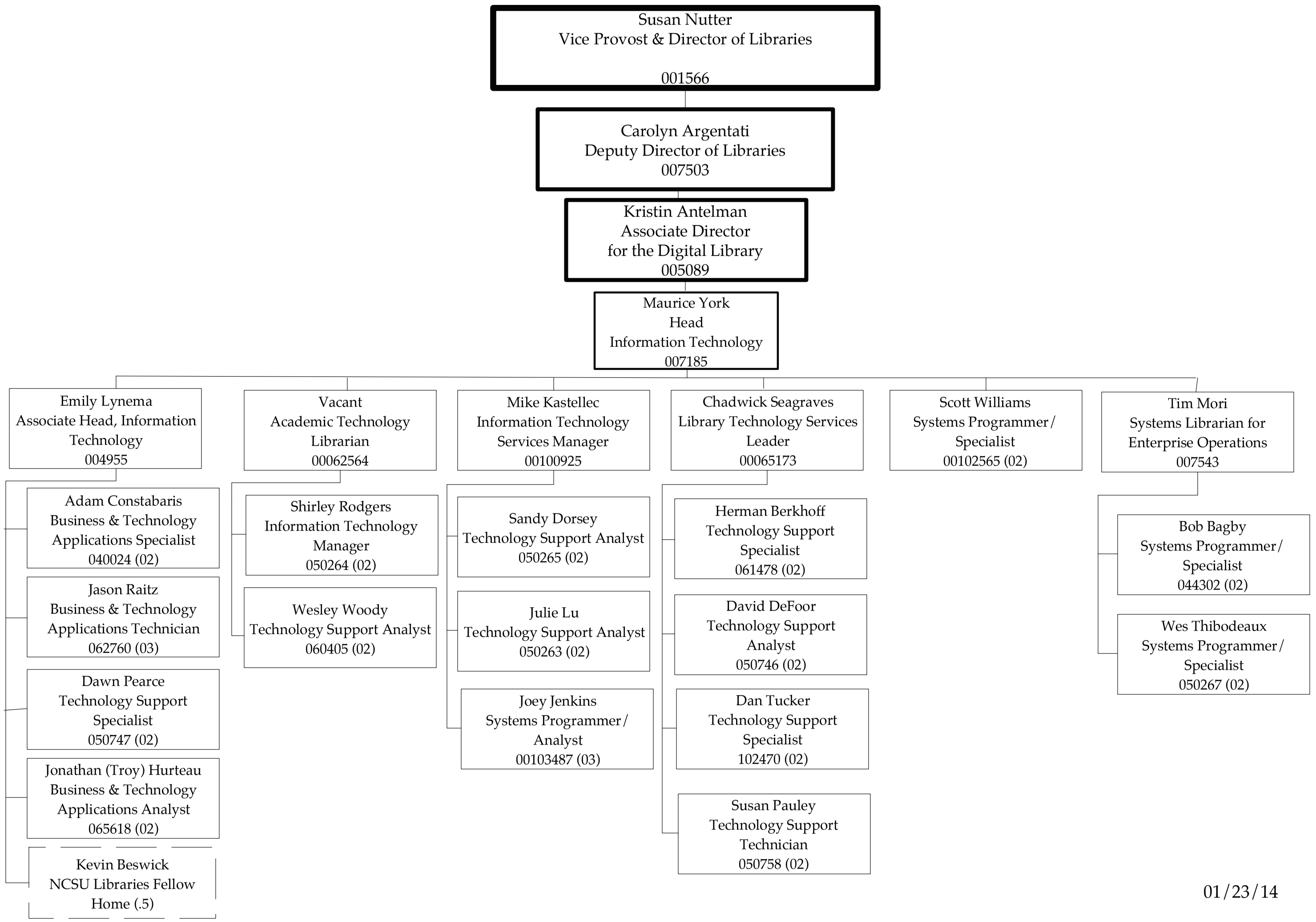
Eric Pauley
Technology Support Specialist
061582 (02)

Cory Lown
Digital Technologies
Development Librarian
007957

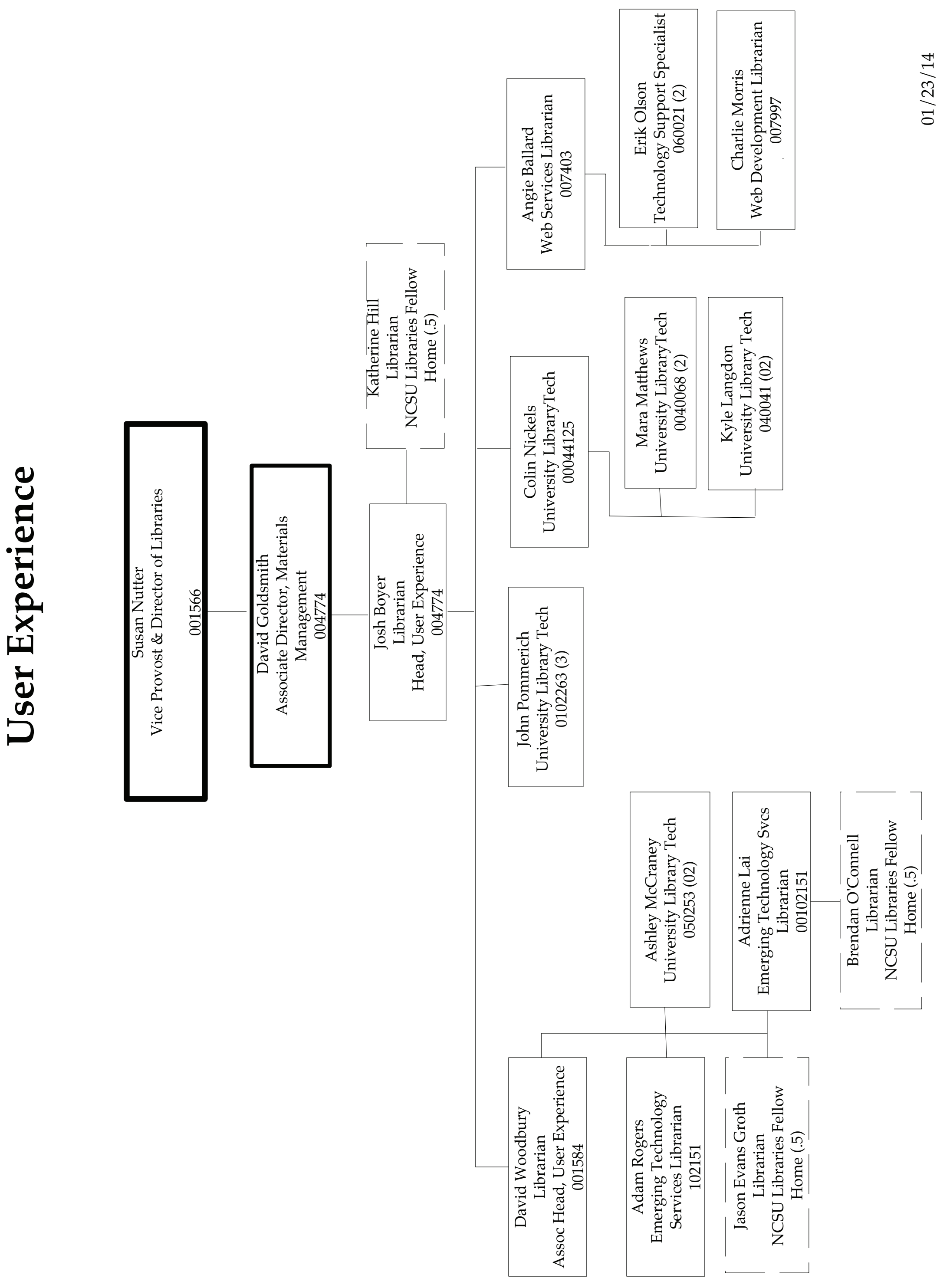
Kevin Beswick
NCSU LibrariesFellow
Project (.5)

11/06/13

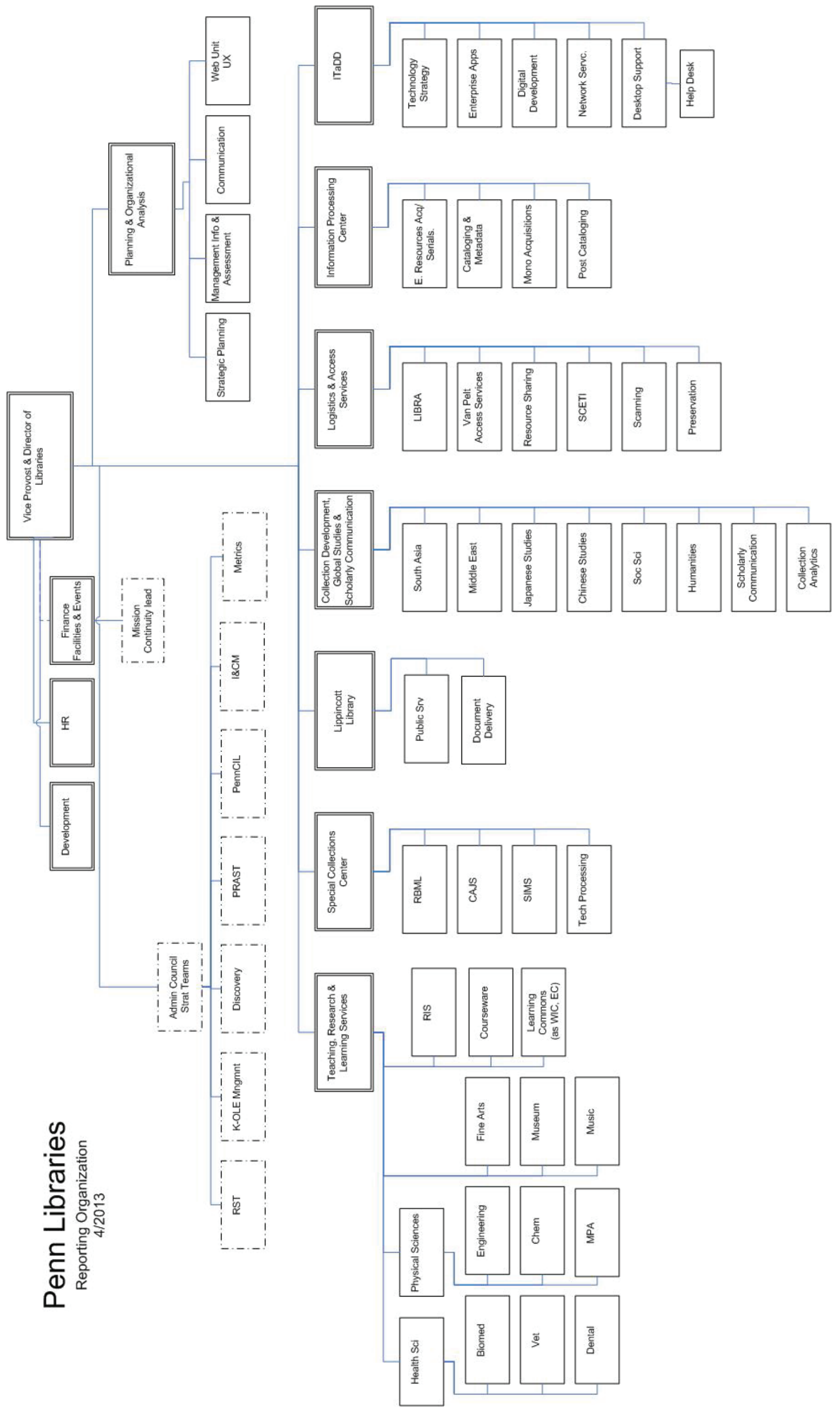
Information Technology


Susan Nutter
Vice Provost & Director of Libraries
001566
Carolyn Argentati
Deputy Director of Libraries
007503
Kristin Antelman
Associate Director
for the Digital Library
005089
Maurice York
Head
Information Technology
007185
Emily Lynema
Associate Head, Information Technology
004955
Adam Constabaris
Business & Technology Applications Specialist
040024 (02)
Jason Raitz
Business & Technology Applications Technician
062760 (03)
Dawn Pearce
Technology Support Specialist
050747 (02)
Jonathan (Troy) Hurteau
Business & Technology Applications Analyst
065618 (02)
Kevin Beswick
NCSU Libraries Fellow
Home (.5)
Vacant
Academic Technology Librarian
00062564
Shirley Rodgers
Information Technology Manager
050264 (02)
Wesley Woody
Technology Support Analyst
060405 (02)
Mike Kastellec
Information Technology Services Manager
00100925
Sandy Dorsey
Technology Support Analyst
050265 (02)
Julie Lu
Technology Support Analyst
050263 (02)
Joey Jenkins
Systems Programmer/Analyst
00103487 (03)
Chadwick Seagraves
Library Technology Services Leader
00065173
Herman Berkhoff
Technology Support Specialist
061478 (02)
David DeFoor
Technology Support Analyst
050746 (02)
Dan Tucker
Technology Support Specialist
102470 (02)
Susan Pauley
Technology Support Technician
050758 (02)
Scott Williams
Systems Programmer/Specialist
00102565 (02)
Tim Mori
Systems Librarian for Enterprise Operations
007543
Bob Bagby
Systems Programmer/Specialist
044302 (02)
Wes Thibodeaux
Systems Programmer/Specialist
050267 (02)

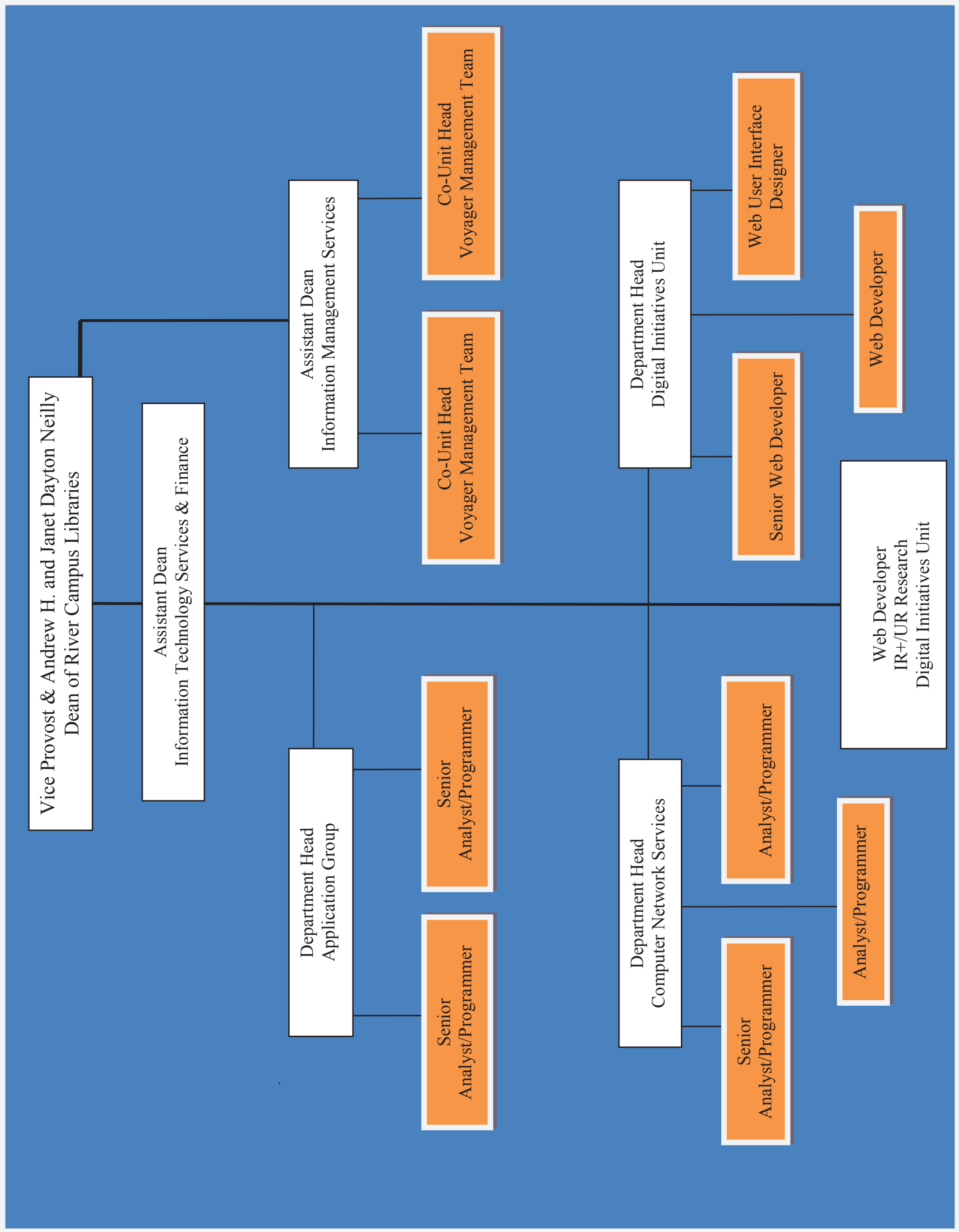
01/23/14

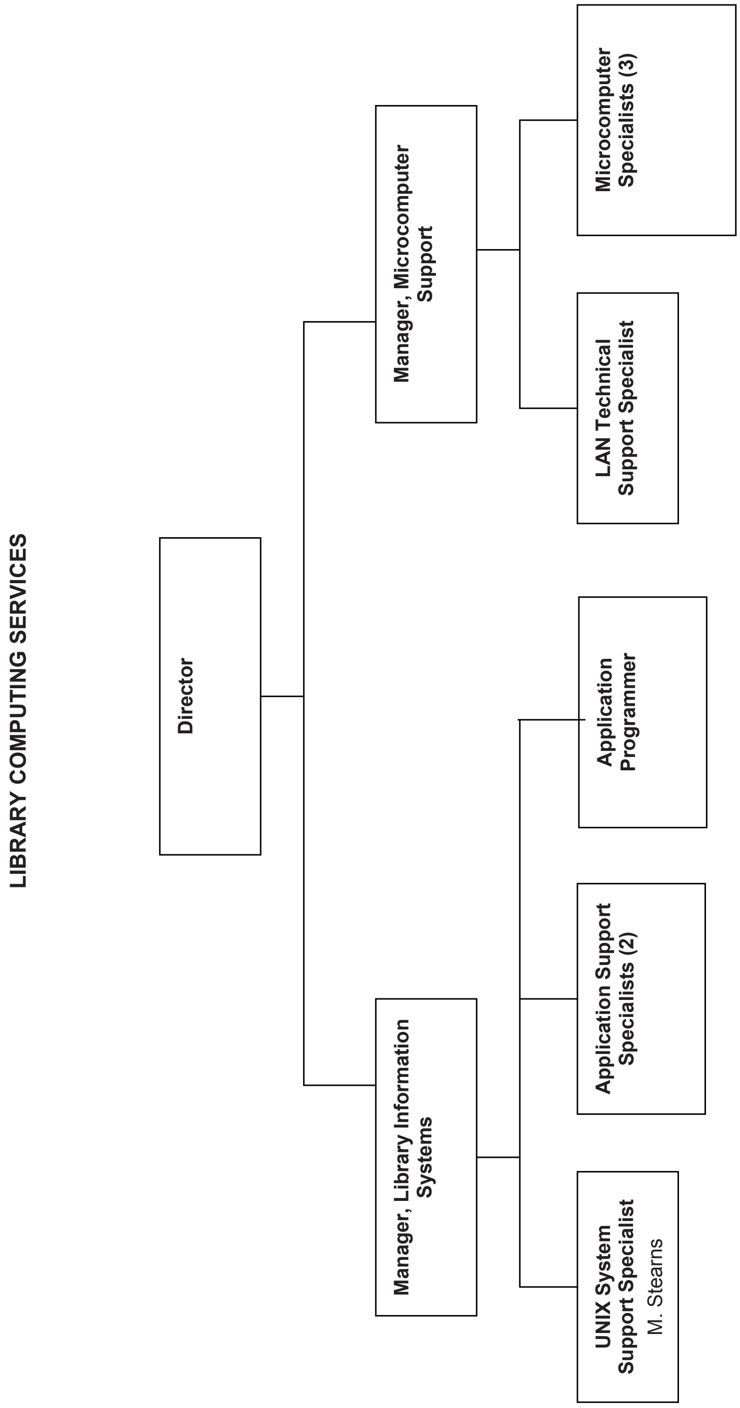

User Experience
Susan Nutter
Vice Provost & Director of Libraries
001566
David Goldsmith
Associate Director, Materials
Management
004774
Josh Boyer
Librarian
Head, User Experience
004774
Katherine Hill
Librarian
NCSU Libraries Fellow
Home (.5)
Angie Ballard
Web Services Librarian
007403
Erik Olson
Technology Support Specialist
060021 (2)
Charlie Morris
Web Development Librarian
007997
Colin Nickels
University LibraryTech
00044125
Mara Matthews
University LibraryTech
0040068 (2)
Kyle Langdon
University Library Tech
040041 (02)
John Pommerich
University Library Tech
0102263 (3)
Ashley McCraney
University Library Tech
050253 (02)
Adrienne Lai
Emerging Technology Svcs
Librarian
00102151
Brendan O'Connell
Librarian
NCSU Libraries Fellow
Home (.5)
David Woodbury
Librarian
Assoc Head, User Experience
001584
Adam Rogers
Emerging Technology
Services Librarian
102151
Jason Evans Groth
Librarian
NCSU Libraries Fellow
Home (.5)
01/23/14

Penn Libraries Reporting Organization



**University of Rochester
River Campus Libraries
Information Technology Services
Organizational Chart**




LIBRARY COMPUTING SERVICES
Director
Manager, Library Information Systems
Manager, Microcomputer Support
UNIX System Support Specialist M. Stearns
Application Support Specialists (2)
Application Programmer
LAN Technical Support Specialist
Microcomputer Specialists (3)

SELECTED RESOURCES

Books and Journal Articles

Askey, Dale. "COLUMN: We Love Open Source Software. No, You Can't Have Our Code." *Code4Lib Journal* 5 (December 15, 2008) http://journal.code4lib.org/articles/527

Breeding, Marshall. "An Update on Open Source ILS." *Information Today* 19, no. 9 (October 2002): 42–43. http://www.infotoday.com/it/oct02/breeding.htm

Breeding, Marshall. "It's Time to Break the Mold of the Original ILS." *Computers in Libraries* 27, no. 10 (November/December 2007): 39–41. http://www.librarytechnology.org/ltg-displaytext.pl?RC=12881

Breeding, Marshall. "The Viability of Open Source ILS." *Bulletin of the American Society for Information Science and Technology* 35, no. 2 (December 2008/January 2009): 20–25. http://www.asis.org/Bulletin/Dec-08/DecJan09_Breeding.html

Chawner, Brenda. "F/OSS in the Library World: An Exploration." *ACM SIGSOFT Software Engineering Notes* 30, no. 4 (July 2005): 1–4. http://dl.acm.org/citation.cfm?id=1083262

Dalling, John. "Open Source, Open Minds? An Investigation into Attitudes towards Open Source Library Management Systems in UK Higher Education Libraries." MSc diss., Aberystwyth University, 2011. http://cadair.aber.ac.uk/dspace/handle/2160/7839

Feller, Joseph, et al. *Perspectives on Free and Open Source Software.* Cambridge, MA: MIT Press 2005. http://mitpress.mit.edu/books/perspectives-free-and-open-source-software

Fogel, Karl. *Producing Open Source Software: How to Run a Successful Free Software Project.* Sebastopol, CA: O'Reilly Media, Inc., 2005. http://shop.oreilly.com/product/9780596007591.do

Goldman, Ron, and Richard P Gabriel. *Innovation Happens Elsewhere: Open Source as Business Strategy.* Burlington, MA: Morgan Kaufmann Publishers, 2005. http://dreamsongs.com/ihe/

Kelty, Christopher M. *Two Bits: The Cultural Significance of Free Software.* Durham, NC: Duke University Press, 2008. http://twobits.net/read/

Meeker, Heather J. *The Open Source Alternative: Understanding Risks and Leveraging Opportunities.* Hoboken, NJ: John Wiley & Sons, 2008. http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470194952.html

Primary Research Group, Inc. *The Survey of Library Use of Open Source Software*. New York: Primary Research Group, 2013. http://www.primaryresearch.com/view_product.php?report_id=407

Ransom, Joann, Chris Cormack, and Rosalie Blake. "How Hard Can It Be?: Developing in Open Source." *Code4Lib Journal* 7 (June 26, 2009): http://journal.code4lib.org/articles/1638

Riewe, Linda M. “Survey of Open Source Integrated Library Systems.” master’s thesis, San Jose State University, 2008. http://scholarworks.sjsu.edu/etd_theses/3481/

Schaefer, Sibyl. 2010. “Challenges in Sustainable Open Source: A Case Study.” *Code4Lib Journal* 9 (March 22, 2010): http://journal.code4lib.org/articles/2493

Schweik, Charles M., and Robert C. English. *Internet Success: A Study of Open-Source Software Commons.* Cambridge, MA: MIT Press, 2012. http://mitpress.mit.edu/books/internet-success

St. Laurent, Andrew M. *Understanding Open Source and Free Software Licensing.* Sebastopol, CA: O’Reilly Media, Inc., 2004. http://shop.oreilly.com/product/9780596005818.do

Stallman, Richard. *Free Software, Free Society: Selected Essays of Richard M. Stallman,* 2nd ed. Boston, MA: GNU Press, 2010. http://shop.fsf.org/product/free-software-free-society-2/

Stol, Klaas-Jan, et al. “Key Factors for Adopting Inner Source.” *ACM Transactions on Software Engineering and Methodology (TOSEM)* 23, no. 2 (March 2014): Article No. 18. http://dl.acm.org/citation.cfm?id=2533685&CFID=364227306&CFTOKEN=76252030

Weber, Steven. *The Success of Open Source*. Cambridge, MA: Harvard University Press, 2005. http://www.hup.harvard.edu/catalog.php?isbn=9780674018587

Technology Transfer and Intellectual Property Policies

Auburn University
Office of Technology Transfer
http://ott.auburn.edu/index.htm

University of California
UC Technology Transfer Program
http://www.ucop.edu/ott/genresources/ttprog.html

University of Louisville
Intellectual Property Policy
http://louisville.edu/research/offices/technology-transfer/ip-policy.html

University of Kansas
Intellectual Property Policy for the Lawrence Campus
http://www.policy.ku.edu/provost/intellectual-property-policy

University of Michigan
Technology Transfer Policy
http://www.techtransfer.umich.edu/resources/policies.php

University of North Carolina at Chapel Hill
Patent and Copyright Policies
www.northcarolina.edu/policy/index.php?pg=dl&id=2787&format=pdf&inline=1

University of Notre Dame
Intellectual Property Policy
http://policy.nd.edu/policy_files/IntellectualPropertyPolicy.pdf

University of Rochester

Policy on Intellectual Property and Technology Transfer
http://www.rochester.edu/ventures/for-ur-innovators/for-inventors-university-policy-on-intellectual-property-and-technology-transfer/

Organization Forge URIs

Auburn University
http://devcat.lib.auburn.edu/cgi-bin/hgwebdir.cgi

Brown University Library
https://github.com/Brown-University-Library

University of Connecticutt
https://github.uconn.edu/

Georgetown University Library Information Technology
https://github.com/Georgetown-University-Libraries

Indiana University Digital Library Program
https://github.com/iudlp/

University of Miami Libraries
https://github.com/umiamilibraries

North Carolina State University
https://github.com/NCSU-Libraries

Northwestern University
https://github.com/orgs/nulib

Penn State Digital Stewardship
http://github.com/psu-stewardship

York University Libraries
https://github.com/yorkulibraries

Project Forge URIs

Avalon Media System
https://github.com/avalonmediasystem/

Blacklight
https://github.com/projectblacklight/blacklight

DSpace
http://github.com/DSpace/DSpace

duke-libraries/dul-hydra
https://github.com/duke-libraries/dul-hydra

mondo-license-grinder
http://code.google.com/p/mondo-license-grinder/

ScholarSpace
http://scholarspace.manoa.hawaii.edu/

wondertool
http://code.google.com/p/wondertool/

xerxes-portal
https://code.google.com/p/xerxes-portal/

Note: All URLs accessed June 24, 2014.