Supporting Software Preservation Services in Research and Memory Organizations - Dedoose Codebook Export

(last updated 12/18/2020)

[Barriers (parent)](#_heading=h.30j0zll)

[Security Risks (child)](#_heading=h.1fob9te)

[Collaboration (parent, cross-cutting)](#_heading=h.3znysh7)

[Conceptions / Understandings of Software & Software Preservation](#_heading=h.2et92p0)

[Software as a physical thing (child)](#_heading=h.tyjcwt)

[Software as something challenging to conceptualize (child)](#_heading=h.3dy6vkm)

[Software as just another digital thing (child)](#_heading=h.1t3h5sf)

[Software as support for something else (child)](#_heading=h.4d34og8)

[Software as product of research (child)](#_heading=h.2s8eyo1)

[Software as institutional legacy (child)](#_heading=h.17dp8vu)

[Software as intellectual property (child)](#_heading=h.3rdcrjn)

[Software as commercial asset (child)](#_heading=h.26in1rg)

[Software as valuable "in itself" (child)](#_heading=h.lnxbz9)

[Software Preservation Practices (parent)](#_heading=h.35nkun2)

[Storage (child)](#_heading=h.1ksv4uv)

[Reimplementing Software (child)](#_heading=h.44sinio)

[Software Documentation (child)](#_heading=h.2jxsxqh)

[Establishing workflows (child)](#_heading=h.z337ya)

[Description (child)](#_heading=h.3j2qqm3)

[Collecting (child)](#_heading=h.1y810tw)

[Surveying / Inventorying (child)](#_heading=h.4i7ojhp)

[Delivering software-dependent content to users (child)](#_heading=h.2xcytpi)

[Software Citation (child)](#_heading=h.1ci93xb)

[Serving Communities & Training Users (child)](#_heading=h.3whwml4)

[Automation (child)](#_heading=h.2bn6wsx)

[Emulation (child)](#_heading=h.qsh70q)

[Migration (child)](#_heading=h.3as4poj)

[Storytelling (child)](#_heading=h.1pxezwc)

[Assessing value and cost (child)](#_heading=h.49x2ik5)

[Complying with legal mandates (child)](#_heading=h.2p2csry)

[Skills (parent)](#_heading=h.147n2zr)

[Familiarity with software development and/or IT (child)](#_heading=h.3o7alnk)

[Resources (parent)](#_heading=h.23ckvvd)

[SPN's Potential (child)](#_heading=h.ihv636)

[Legal support (child)](#_heading=h.32hioqz)

[Advocating for resources (child)](#_heading=h.1hmsyys)

[Community and Professional Networks (child)](#_heading=h.41mghml)

[Institutional Support (child)](#_heading=h.2grqrue)

[Learning & Training (child)](#_heading=h.vx1227)

[Demoted Codes (parent)](#_heading=h.3fwokq0)

[Perceptions, Motivations, Values (child)](#_heading=h.1v1yuxt)

[Developer/User Practices (child)](#_heading=h.4f1mdlm)

[Software as "time-based media" (child)](#_heading=h.2u6wntf)

[Software as historically specific craft (child)](#_heading=h.19c6y18)

# Barriers (parent)

Description: This code relates to our research goal: " to identify barriers software preservation service providers face in preserving and providing access to software." Use this code with another code (ideally) to identify types of barriers, or use it alone to identify barriers in general.

## Security Risks (child)

Description: This code can be used when participants mention risks related to software like viruses, malware, etc. Relevant terms used by participants might be risk, secure/security, dangerous, hacking, corrupted, virus, malware.

# Collaboration (parent, cross-cutting)

Description: CROSS-CUTTING This code applies to instances of collaboration between software preservationists and other groups/individuals. Use this code when participants talk about collaborating for the purpose of software preservation or learning/talking about software preservation.

# Conceptions / Understandings of Software & Software Preservation

Description:

"Catch-all" code that can be used if a more specific code does not exist. Relates to "what software is", "what software is used for", "how software fits in with broader digital materials and infrastructure", and "why software is important".

## Software as a physical thing (child)

Description:

Captures the idea that preservationists often work with the physical manifestations of software, like boxed software, documents, discs, etc.

## Software as something challenging to conceptualize (child)

Description:

This code aims to capture parts of the conversation in which the participants discuss how they conceptualize software or the challenges of conceptualizing, defining, categorizing, enumerating, or otherwise describing software. Use this code when they are talking about their own experience or just the practice/field in general. The diversity of perceptions and motivations around software can overwhelm the work of preservation.

## Software as just another digital thing (child)

Description:

This code can be used to refer to similarities and differences between software objects and other kinds of digital objects, and between software preservation and digital preservation. Additionally, in many institutions, responsibilities for software and digital preservation are shared by the same set of people, and the infrastructure and community networks that

## Software as support for something else (child)

Description:

This code can be used when a participant discusses software that is preserved in order to achieve a specific purpose rather than for its inherent or evidentiary value, especially if the discussion revolves around the need to preserve software to support access to other digital objects. This could be software that supports a textbook, software that helps install other software or the general notion that software is a helper tool. Relevant terms may include dependencies, interdependency, decoding, or rendering.

## Software as product of research (child)

Description:

This code aims to capture discussions about software as a necessary part of or product of research. Use this code when participants discuss community development of software by a specific discipline, the curation of software as part of a research project, or research data management. This code can also be used for software citation.

## Software as institutional legacy (child)

Description:

This code can be used when a participant discusses the need to preserve software because it contributes to or is part of a specific institution's legacy. Examples of this could be (but are not limited to) an important piece of research software, an interactive game developed for student orientation, a geographic information system designed by a governmental institution, personal papers donated to an archive that include software, or a home-grown information management system or web application.

## Software as intellectual property (child)

Description:

This code can be used to capture the conversation around intellectual property of software. Use this code if participants mention legal issues like copyright, fair use, trademark, or patent law.

## Software as commercial asset (child)

Description:

This code aims to capture entrepreneurial aspects and practices related to software. Software is often commercialized and sold to companies. This code is focused on capturing financial, proprietary, or other economic aspects of software creation and ownership.

## Software as valuable "in itself" (child)

Description:

Use this code when participants discuss the inherent, often cultural, value of software as distinct from its value as support for something else.

# Software Preservation Practices (parent)

Description:

"Catch-all" code that can be used if a more specific code does not exist. These codes are used to describe specific actions or activities and curation related to software preservation including their absence, their necessity, or their current state.

## Storage (child)

Description:

Refers to work of storing things either physically or digitally; preparing things to go into storage.

## Reimplementing Software (child)

Description:

Reimplementing Software for Conservation

## Software Documentation (child)

Description:

This code aims to capture software documentation produced by developers or users NOT service providers. This code relates to the practice of creating and using documents and documentation relating to a piece of software.

## Establishing workflows (child)

Description:

This code relates to planning and organizing people and resources or defining protocols in order to undertake the work of software preservation. Use this code if participants explicitly mention the presence or absence of workflows (written or tacit) or use terms/phrases like preservation planning, documenting our organizational processes, preservation process, "doing" software preservation, process, steps, or system.

## Description (child)

Description:

This code relates to the work of service providers to describe software. Often such description involves the production of metadata.

## Collecting (child)

Description:

This code relates to the work of obtaining and taking custody over materials. This includes the tasks of acquisition and accessioning. Acquisition is about obtaining materials via purchase, donation, or transfer. Accessioning is about taking custody, legal and policy responsibility, over a piece of software. Use this code for any discussion of these practices or general mentions of "what to collect" or collecting policies related to software.

## Surveying / Inventorying (child)

Description:

This code relates to the work of gaining intellectual control over a software in a collection. Use this code if participants mention the explicit practice of surveying or inventorying or if they discuss the need to know what is in their collection.

## Delivering software-dependent content to users (child)

Description:

This code relates to the work of delivering software or software-dependent objects. Use this code when participants discuss responding to reference questions or direct requests for software (or software-dependent materials) or if they are talking in general about the challenges of responding to these requests or enabling access to software.

## Software Citation (child)

Description:

This code relates to the practice of citing software, particularly in the context of academic writing, so that software authors get credit for their work. Can also be used for discussions about citation for reproducibility.

## Serving Communities & Training Users (child)

Description:

This code aims to capture interactions between service providers and users. Users include software developers, software users, researchers, or whole disciplines. Interactions of most interest are ones in which the service provider is providing training on how to use (preserved?) software or understand the context for a particular software or digital object, how to build preservation/curation into user development tasks, and helping users (broadly defined) curate their own work. Some key terms are: training, instruction, outreach (instructional).

## Automation (child)

Description:

This code can be used when participants mention practices of automating software preservation activities. For example: automated file-type identification or metadata generation. Some key terms related to automation are: AI/artificial intelligence, machine learning, machine-actionable.

## Emulation (child)

Description:

This code can be used when participants mention the practice of emulation. This can be used to capture conversations about currently providing emulation services, the desire/need for these kinds of services and participation in projects like EaaSI. Some key terms related to emulation are: virtualization, containerization, operating systems, environments.

## Migration (child)

Description:

This code can be used when participants discuss migration projects or the act of migrating (with minimal transformation) media, software, or data from one format or medium to another. Some key terms related to migration are: migrate, transfer, obsolescence, older files, earlier versions, etc.

## Storytelling (child)

Description:

This code relates to practices of capturing the significance of a piece of software through narratives, personal accounts, oral or written histories etc. Use this code if participants mention storytelling (or some component of stories) as a way to bring awareness of a collection of materials. Some key terms are: stories, and narrative.

## Assessing value and cost (child)

Description:

This code relates to practices of budgeting, determining, and justifying the cost of software preservation activities. Participants may use terms like value, appraisal, cost, monetary value, tradeoff, priority, or phrases like “worth it.” Relates to making decisions about taking action (or not) related to resource allocations for software preservation.

## Complying with legal mandates (child)

Description:

This code relates to any discussion of legal mandates for retention of materials or legal requirements for providing access.

# Skills (parent)

Description:

In the survey, we asked explicitly about skills people needed for "doing" software preservation. This code aims to capture the discussion and nuances related to what kinds of skills are not only necessary but preferred to do this kind of work. This includes "soft skills" and expertise like getting funding, collaborating, advocating, etc.

Use this code for material relevant to RQ3: "to identify what skillsets are needed to understand, collect, preserve, and provide access to software"

## Familiarity with software development and/or IT (child)

Description:

Use this code where interviewees refer to technical expertise in software development and/or computing (IT) as a necessary skill set for software preservation.

# Resources (parent)

Description:

These codes can be used to indicate things that software preservationists need to do their work, or to indicate challenges in obtaining those resources.

## SPN's Potential (child)

Description:

This code can be used anytime a participant mentions SPN and how it can support or benefit them.

## Legal support (child)

Description:

This code can be used when participants discuss help or support from lawyers or legal counsel for purposes of software preservation.

## Advocating for resources (child)

Description:

This code aims to capture discussions of how service providers advocate for resources. Participants might mention arguments, motivations or the case for software preservation and/or how to talk with administrators or other entities in positions of power.

## Community and Professional Networks (child)

Description:

This code is focused on capturing the need for or participation in community and/or professional networks. Participants may use obvious terms like community or networking but they may also mention related terms like mentoring, advice, organization, membership, or connection.

## Institutional Support (child)

Description:

This code aims to capture conversations about receiving institutional support for software preservation including different levels or kinds of support beyond financial support. This includes broader institutional support to the extent that software preservation takes place within broader institutional contexts (like archives, etc.). This code also includes discussions on the absence of support, not just presence.

## Learning & Training (child)

Description:

This code aims to capture resources service providers use for learning about software preservation techniques. Participants may mention terms like teaching, learning, coursework, workshops, or self-paced classes. This code is specific to receiving training, not providing training.

# Demoted Codes (parent)

Description:

Old codes that we aren't so interested in anymore

## Perceptions, Motivations, Values (child)

Description:

CROSS-CUTTING Perceptions motivations and values of either service providers or users of software preservation services.

## Developer/User Practices (child)

Description:

These codes relate to practices of users and developers of preserved/curated software (i.e., not those of service providers).

## Software as "time-based media" (child)

Description:

This code captures discussions of software as a category of media or media art, particularly in the "GLAM" context. ("Time" refers to the idea that software exists in time as it is used or experienced).

## Software as historically specific craft (child)

Description:

This code can be used if the participant is discussing the development of software as a craft in itself or talking about the history of software and/or its development.