Automating the Assessment of Cultural Value for Digital Software Libraries

Poster submission for the Web Archiving and Digital Libraries (WADL) Workshop 2026

Lars Florian Meiendresch
lars.meiendresch@uni-due.de
University of Duisburg-Essen
Essen, Germany

Michael Goedicke
michael.goedicke@paluno.uni-due.de
University of Duisburg-Essen
Essen, Germany

Abstract

The Software Heritage (SWH) archive pursues the mission of preserving the world's software source code. However, the *AI Code Explosion*—the exponential generation of synthesized code—presents a severe physical and economic challenge for decentralized mirror servers. To complement the "archive everything" strategy, we introduce the Universal Value Index (UVI), an automated, highly scalable graph-topological heuristic system. Inspired by UNESCO World Heritage criteria, the UVI automates the evaluation of software repositories across Impact, Time, Rarity, and Quality. By providing a mathematical framework mapped to cultural heritage standards, the UVI system enables digital archivists to leverage automation for scalable preservation decisions to ensure the survival of critical digital infrastructure.

CCS Concepts

• **Information systems** → **Digital libraries and archives**.

Keywords

Software Preservation, Software Heritage, Digital Archives, Automation, Graph Topology

ACM Reference Format:
Lars Florian Meiendresch and Michael Goedicke. 2026. Automating the Assessment of Cultural Value for Digital Software Libraries: Poster submission for the Web Archiving and Digital Libraries (WADL) Workshop 2026. In *Proceedings of Web Archiving and Digital Libraries Workshop (WADL '26).* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The Software Heritage (SWH) archive acts as the universal digital library for software [1, 5]. However, this mission faces a fundamental storage challenge. With the advent of generative AI, the volume of synthesized source code is growing at an unprecedented rate [2, 3]. This rapid proliferation of code is severely worsened by a recent surge in enterprise storage costs [4]. As the archive grows, storing every single repository across mirror servers becomes increasingly demanding, turning the curation of the software commons into a pressing scalability problem. This challenge is directly evident in our operational experience hosting the SWH.NRW[1] mirror, the third official Software Heritage Foundation mirror worldwide, deployed at the University of Duisburg-Essen.

The archival community must complement the "archive everything" approach with automated tools for intelligent prioritization. Ultimately, the long-term goal is to empower decentralized mirrors to define selective mirroring policies, enabling an institution to state: "We will mirror the most important and culturally relevant X Terabytes or Petabytes of the global archive." Lowering the storage barrier to entry in this way encourages the proliferation of new mirror nodes, paving the way for a robust, truly decentralized network where subsets of the archive can be flexibly distributed across multiple institutions. We ask: how can we automate and objectively quantify the cultural value of a software repository to assist these selective archival efforts? We introduce the **Universal Value Index (UVI)**[2], a heuristic system designed to objectively rank software.

2 The UNESCO Connection

Software is a fragile cultural artifact [6]. UNESCO officially supports the Software Heritage initiative and recognizes source code as a fundamental component of human cultural heritage [7, 8]. The UVI directly maps software characteristics to established UNESCO World Heritage criteria [9], providing a robust, philosophically aligned evaluation framework for digital archivists.

3 The Universal Value Index

The final Universal Value Index score $S_{UVI}(o)$ for a repository origin o is defined using a weighted p-norm of four evaluative dimensions, multiplied by a fork penalty $P_{fork}(o)$ and a bloat penalty $P_{bloat}(o)$:

$$S_{UVI}(o) = P_{fork}(o) \cdot P_{bloat}(o) \cdot \left(\sum_{i=1}^{4} w_i \cdot D_i(o)^p \right)^{\frac{1}{p}} \quad (1)$$

where $D_i \in \{I_{norm}, T_{norm}, R, Q\}$ and $\vec{w}$ is a weighting vector satisfying $\sum w_i = 1$. We assume $p = 2$, which calculates the weighted quadratic mean. The four dimensions are mathematically formulated as follows:

1. Impact (I_{norm}): Impact measures the systemic relevance of a repository through its reference density (in-degree). High-impact

[1] https://swh-nrw.uni-due.de
[2] https://anonymous.4open.science/r/swh-universal-value-index-1AC5

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WADL '26, Dallas, TX, USA
© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2026/10
https://doi.org/XXXXXXX.XXXXXXX

software represents technological milestones that serve as the backbone for modern digital civilization (UNESCO criteria ii, iv, ix). Due to the power-law distribution of software dependencies, this dimension is calculated using a logarithmically scaled in-degree metric:

$$I_{norm}(o) = \frac{\log(1 + D_{in}(o))}{\log(1 + D_{max})} \tag{2}$$

where $D_{in}(o)$ is the absolute number of nodes referencing code entities of o, and D_{max} is the historical maximum in-degree within the archive.

2. Time (T_{norm}): This dimension captures the historical depth of a repository. Code with a high T_{norm} score serves as a historical testimony to early programming paradigms (UNESCO criteria iii, vi, viii). It evaluates the absolute age of the repository and the span of its active maintenance:

$$T_{norm}(o) = \alpha \cdot \frac{Y_{now} - 1970}{Y_{now} - Y_{first}} + (1 - \alpha) \cdot \frac{\text{MaxAge}}{Y_{last} - Y_{first}} \tag{3}$$

where Y_{first} and Y_{last} are the years of the first and last commits, and α balances absolute age and maintenance duration.

3. Rarity (R): To preserve "digital biodiversity," the Rarity dimension prioritizes code written in obsolete or niche programming languages (e.g., COBOL, Fortran), addressing the risk of obsolescence facing niche technologies (UNESCO criteria v, x):

$$R(o) = \sum_{l \in L_o} p_l(o) \cdot \left(1 - \frac{f_l}{f_{max}}\right) \tag{4}$$

4. Quality (Q): Quality assesses the structural integrity of the code. This metric evaluates architectural elegance, documentation, release management, and collaboration diversity to identify "masterpieces of human creative genius" (UNESCO criteria i, vii).

$$Q(o) = c_1 \cdot q_{doc} + c_2 \cdot q_{tags} + c_3 \cdot q_{collab} + c_4 \cdot q_{formal} \tag{5}$$

4 Automated Archival Application

Calculating this metric using external API calls across 400 million repositories is computationally unfeasible. Thus, our tool automates this locally on the highly compressed SWH graph dataset, using efficient topological proxies rather than semantic metadata lookups.

To demonstrate the empirical composition, we calculated the index for a subgraph of 3,000 Python repositories using an optimized weight vector $\vec{w} = (0.50, 0.25, 0.15, 0.10)$ for Impact, Time, Rarity, and Quality.

Table 1: Empirical Examples of UVI Evaluations

Repository Origin	Type	Impact	Time	Rarity	Quality	UVI
`deb://.../sagemath`	OS Pkg	0.90	0.37	0.00	0.01	**0.66**
`github.com/django`	Git	0.16	1.00	0.00	0.60	**0.55**
`github.com/cpython`	Git	0.10	1.00	0.00	0.60	**0.54**
`pypi.org/.../tzlocal`	PyPI	0.10	0.37	0.00	0.01	**0.20**
`deb://.../ghextris`	OS Pkg	0.10	0.08	0.00	0.00	**0.08**

As shown in Table 1, high-impact operational packages like *sagemath* achieve top scores (≈ 0.66) primarily through immense reference density. Foundational frameworks like *django* and *cpython* secure high rankings (≈ 0.55) through decades of maintenance and structural integrity. In contrast, obsolete games or simple distribution wrappers yield much lower scores (≤ 0.10).

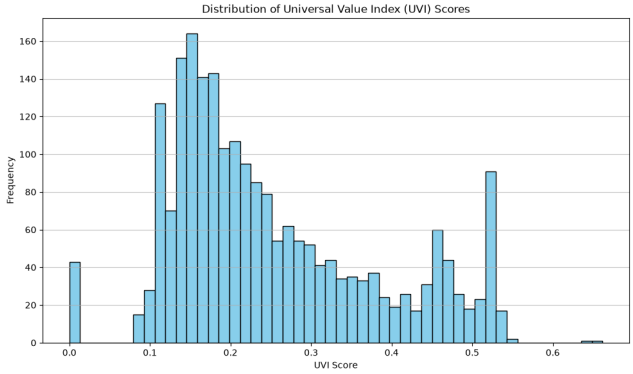


Figure 1: Distribution of calculated UVI scores across the initial test dataset. The right-skewed distribution highlights the rarity of exceptionally high-value repositories.

5 AI-Assisted Verification

To validate if heuristics align with human intuition of "historical value," we used LLMs as an independent jury to semantically rank sampled repositories. Achieving a Spearman correlation of $\rho \approx +0.55$ indicates that structural heuristics can approximate semantic judgement. This highlights a potential *Two-Stage AI Pipeline*: the UVI acts as a scalable topological filter (Stage 1), enabling resource-intensive LLMs to perform deep semantic curation on the surviving candidate repositories (Stage 2).

6 Conclusion

The UVI provides a necessary mathematical foundation for scalable prioritization in software archiving. By translating UNESCO World Heritage concepts into computable mathematical formulations, it offers a pragmatic tool for capacity planning. This purely topological approach serves strictly as a heuristic approximation of long-term archival importance; a full global evaluation is still pending. By definition, such a proxy is never 100% accurate and does not definitively distinguish "good" from "bad" software. Our future work aims to refine these heuristics and deploy this system across federated mirror networks.

References

[1] Roberto Di Cosmo and Stefano Zacchiroli. 2017. Software Heritage: Why and How to Preserve Software Source Code. In *Proceedings of the 14th International Conference on Digital Preservation (iPRES 2017)*.

[2] Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. 2023. The impact of AI on developer productivity: Evidence from GitHub Copilot. *arXiv preprint arXiv:2302.06590* (2023).

[3] Antoine Pietri, Diomidis Spinellis, and Stefano Zacchiroli. 2019. The Software Heritage graph dataset: public software development under one roof. In *Proceedings of the 16th International Conference on Mining Software Repositories*. 138–142.

[4] Lilith Ray. 2026. *Will SSD Prices Drop in 2026?* https://www.oscooshop.com/blogs/blogs/will-ssd-prices-drop-in-2026

[5] Software Heritage Foundation. 2026. *Mission.* https://www.softwareheritage.org/mission/

[6] Software Heritage Foundation. 2026. *Serving the world heritage.* https://www.softwareheritage.org/mission/heritage/

[7] UNESCO. 2017. *Agreement on software preservation signed at UNESCO.* https://www.unesco.org/en/articles/agreement-software-preservation-signed-unesco

[8] UNESCO. 2019. *Paris Call - Software Source Code as Heritage.* https://www.unesco.org/en/paris-call-software-source-code

[9] UNESCO. 2026. The Criteria for Selection. https://whc.unesco.org/en/criteria/.