BECOMING

Barberotheca

Project Charter

Project Management Plan and Software Engineering Specifications for 'Barberotheca'. Curated for the “Project Management and Software Engineering for Cultural Heritage” course.

a.y. 2025/2026

Tommaso Barbato
tommaso.barbato@studio.unibo.it

Martina Uccheddu
martina.uccheddu@studio.unibo.it

Table of contents

About the Project Charter

The following document is a shared effort to effectively manage the Barberotheca project by actively adopting a top-down planning approach. The document begins with a comprehensive project charter that defines the project description, scope, deliverables, activities, estimated resources, schedule, and risks at a high level.
Regarding the planning process, the curators avoided fragmenting the effort into distinct sub-system charters. Instead, a 'divide and conquer' strategy was applied to technical knowledge, requiring the authors to bridge the project manager and developer perspectives simultaneously throughout the planning process and documentation production. The result is a unified, detailed charter for the entire system.
This unified planning lays the groundwork for the project's execution, supporting the adoption of a team-managed agile process model. Given the project's nature, which mixes data-oriented needs with process-oriented goals, this methodology provides the necessary flexibility to handle partial requirements as development progresses.

Project scope

Project Name	Barberotheca
Project ID	BRBTHC-001
Grantor or Customer	Fondazione Eventi Città di Sarzana Srl

Document Version	**Date**	**Contributors**	**State**
1.0	28/01/2026	Martina Uccheddu, Tommaso Barbato	Final

Brief description

Barberotheca is a semantic digital library hosting the transcription of every *lectio magistralis* by historian Alessandro Barbero held at the yearly event in Sarzana called “Festival della Mente”. The project aims to automate the workflow of acquiring, transcribing and semantically enriching audio content, extracted from original video sources and to deploy a web application for the dissemination and semantic exploration of these digital assets.

The system leverages artificial intelligence (OpenAI Whisper for audio transcription) and natural language processing techniques (Spacy NLP) to perform statistical analysis and extraction of recurrent keywords and entities. The inferred entities are semantically reconciled and linked to external authority control systems (Wikidata, VIAF, GeoNames).

On the user side, Barberotheca provides a web interface that enables multidimensional navigation. The system offers Javascript-powered interactive visual tools, provides information retrieval capabilities, including filtering and fuzzy search logic. Furthermore, it facilitates deep engagement through semantic contextualization. This allows users to traverse complex historical connections, transforming the archive into an active research tool.

Scope and objectives

The primary goal is to develop a software able to ingest raw media and disseminate the enriched digital assets. To manage the complexity of this project the scope and subsequent objectives are strategically divided into two functional areas, corresponding to the data-oriented and process-oriented goals mentioned above.

The semantic data infrastructure

The project builds a coherent corpus of audio, text, and metadata starting from heterogeneous online sources. **Audio tracks** are systematically extracted from selected lectures, normalized through semantic filenames, and preserved in a high-quality version for transcription. Automatic speech-to-text generates **full textual transcriptions**, which are then reviewed and corrected when necessary. Natural language processing techniques are applied to the transcripts to identify recurring keywords and named entities, producing **structured semantic metadata**. Human curation refines these results to ensure conceptual relevance and consistency and finally audio files are compressed for efficient storage and distribution, while all data and metadata are consolidated into structured, interoperable formats suitable for reuse and dissemination. A **configuration guide** is then produced to document the process so that new data and metadata can be added in the future coherently.

The dissemination platform

The goal is to deploy a web-based user application focused on digital resource access and dissemination. This platform will allow the users to browse the data in a multidimensional way. The main outcome is an **interactive interface** that enables the user to search and read the enhanced transcription and features: advanced retrieval, alternative browsing, semantic contextualization. Furthermore, a **user manual** will be designed to facilitate the effective use of the browsing and research tools.

Project deliverables

The following table identifies each deliverable with a unique code, a descriptive title, its type as internal or external outputs, covering the entire lifecycle of the software project from the initial project summary to the deployment of the semantic data infrastructure and the dissemination platform.

Code	Name	Description	Type
D01	Project summary	Document describing the project context, scope and objectives to be discussed with stakeholders.	Internal (documentation)
D02	Project	Document detailing the WBS,	Internal

	management plan	schedule (Gantt Diagram), resources allocation and cost estimates.	(documentation)
D03	Software requirement specification	Guide for the development team including requirements, use cases, diagrams.	Internal (Documentation)
D04	Kick-off presentation	Slideshow to introduce the project domain and roadmap to the team.	Internal (Documentation)
D05	Audio corpus	The collection of extracted, normalized, and compressed audio tracks.	Internal (Dataset)
D06	Transcription corpus	The complete set of verified text transcriptions generated from the audio sources.	Internal (Dataset)
D07	Enriched metadata dataset	Structured dataset containing extracted keywords, named entities, and semantic links to external authorities.	Internal (Dataset)
D08	Configuration guide	Documentation of the data ingestion process to ensure new data and metadata can be added coherently in the future.	Internal (Documentation)
D09	End user application	Deployed web platform allowing users to search, browse, and read the enriched transcriptions.	External (Software)
D10	User manual	Features guide designed to assist end-users.	External (Documentation)

Project activities

To identify project activities, an objective-oriented approach has been adopted. This method lists all the products expected from the project and associates a specific activity to each. Accordingly, starting from the list of expected products, the WBS results from the set of activities and tasks required to obtain them. The following is the written version of the WBS that maps the diagram structure: outputs, activity, task. The original diagram is available at: https://github.com/metamuses/becoming-barberotheca/blob/main/docs/work_breakdown_structure.pdf.

BARBEROTHECA

METADATA SYSTEM

Requirements analysis

- Identify sources
- Define metadata needed structure
- Compile initial raw dataset

Tools & process development

- Harmonize semantic naming
- Extract keyword and entities
- Reconcile authority controlled entities
- Research further concepts connections

Validation and materialization

- Review automated metadata extraction
- Validate authority control mappings
- Provide convertible formats

DATA SYSTEM

Requirements analysis

- Identify relevant audio sources
- Define desired types and formats
- Establish required tools

Pipelines development

- Acquire audio files
- Perform speech-to-text transcription
- Optimize data for storage

Validation and provisioning

- Verify audio integrity
- Review transcription quality
- Produce final data assets for publication

CONFIGURATION GUIDE

Key processes analysis

- Identify pipeline key points and inputs
- Identify pipeline main pitfalls

Content development

- Document audio ingestion steps
- Document transcription workflow
- Document metadata production
- Document scripts and tools usage

Review and finalization

- Test-drive the guide

USER APPLICATION

Requirements analysis

- Define Use Cases
- Analyze Data Requirements

System design

- UX design and layout definition
- Visual identity and UI Definition

Development

- Core Layout Implementation
- Data integration and logic
- Search and information retrieval development
- Advanced visualization and navigation features
- Media rendering component

Testing and validation

- Test search functionalities
- Cross-browser and cross-device testing
- User testing

USER MANUAL

Requirements analysis

- Select critical function/feature

Content development

- Define document template
- Draft instructions

Review and finalization

- Check accuracy
- Editorial review

Project estimation

The goal of this section is to define the total estimated costs. To achieve this, a **top-down approach** is adopted, where higher-level activities are estimated first as macro-aggregates, and micro-activities are re-estimated during the process. The estimation begins with the drafting of a comprehensive **list of human and technical resources** required to support the project's scope. By combining this resource list with the project's operational needs, an **activity-based personnel budget** is developed, estimating costs for macro-activities based on daily rates. Finally, recognizing that human effort is the primary direct cost, these figures are aggregated into a **category-based budget**, which adds costs for infrastructure, marketing services, as well as general operating expenses.

Project Resources

List of human resources

The selection of human resources is driven by the specific needs of the project's lifecycle, from data ingestion to final user interaction.

A **Project Manager** is essential to coordinate the team, plan, report and schedule the activities, allocate and monitor costs and resources, interact with stakeholders, manage risks, ensuring the project stays on track. Given the centrality of the digital assets, a **Data Steward** is required to oversee the quality, normalization, and long-term preservation of the media assets and metadata. A **Solution Architect** is included to design the complex technical pipeline (integration of OpenAI Whisper and semantic enrichment) and ensure system scalability. The development phase necessitates a **Web Developer** for the implementation of the platform and a **UX/UI Designer** to ensure the interface is user-friendly and the manuals are well-designed. To guarantee the semantic accuracy of the historical content, a **Domain Expert** is crucial for validating the entity extraction and linking. Finally, a **Quality Assurance (QA) Engineer** is appointed to test both the software functionality and the integrity of the data outputs.

Resource role	Resource code
Project Manager	PM
Data Steward	DS

Solution Architect	SA
Web Developer	DEV
UX/UI Designer	UX/UI
Domain Expert	EXP
Quality Assurance Engineer	QA

List of technical resources and tools

The list of technical resources serves two practical purposes. First, it helps identify the concrete technical skills required from the team, making it easier to match tools with the competencies of the selected personnel. Second, it supports a realistic estimation of infrastructure costs by clarifying which resources are used for computation, storage, and hosting, and for how long, allowing the project to estimate costs and technical needs in a controlled way.

Name	**Description**
Python	Core scripting language for data pipelines, NLP processing, automation, and format conversions, with a wide ecosystem of libraries
yt-dlp	Reliable, open-source tool to extract high-quality audio from online video sources at scale
OpenAI Whisper	Speech-to-text engine used to produce accurate transcriptions from audio sources
FFmpeg	General-purpose audio processing tool for compression, normalization, and format optimization
Figma	Collaborative design tool for UX flows, wireframes, and interface prototyping
GitHub	Version control and collaboration platform for code, data, and documentation
AWS	High-performance cloud compute machines for fast audio download, transcription, and NLP pipelines, plus S3 storage for final audio files
DigitalOcean	Cost-effective cloud hosting for deployment of the final web application

Project Budget

Activity-based personnel costs budgeting

The personnel cost estimation is derived from the specific human effort required for each higher-level activity, quantified in person-days under the assumption of a standard 8-hour workday and a 5-day work week.

Drawing on past projects' timeline, the overall execution is projected to span six weeks. This duration has been explicitly calculated as 27 working days. Accordingly, the budget is calculated by multiplying the estimated effort in days by the daily cost rate assigned to each role, ensuring the financial plan accurately reflects the necessary resource allocation for the project's duration.

The daily rates have been estimated using average annual salaries for each role in Italy, sourced from Glassdoor. The daily personnel cost is derived assuming a working year of 190 days.

It is important to note that the activities listed in this budget do not strictly mirror the high-level activities nor the tasks defined in the Work Breakdown Structure (WBS). This deviation is intentional: while the Gantt diagram provides the exact correspondence between activity, task owner and timeline, the scope of this estimation table is solely to provide a financial assessment of the required human resources.

Activity	Effort (days)	Resource code	Cost per day	Activity Cost
Project management	6	PM	€ 226	€ 1.356
Data strategy and process documentation	9	DS	€ 56	€ 504
Data pipeline development and execution	21	SA	€ 253	€ 5.313
Domain research and validation	12	EXP	€ 195	€ 2.340
Quality assurance and editorial review	8	QA	€ 174	€ 1.392
User experience design and documentation	10	UX/UI	€ 184	€ 1.840
Application engineering	12	DEV	€ 211	€ 2.532
			TOTAL COST	**€ 15.277**

Category-based budgeting

Since personnel costs compose the most significant portion of the project's financial requirements, the budgeting process begins with this figure as the primary cost driver. To calculate the final total estimated cost, this base cost is then aggregated with other estimated expenses. The following table reports the direct costs for infrastructure, as well as indirect costs for general operating expenses (overheads), providing a complete financial overview.

Category	**Estimated costs**
Personnel	€ 15.277
Infrastructure	€ 2.000
General and operating expenses	€ 1.500
TOTAL ESTIMATED COST	**€ 18.777**

Gantt Diagram

The Gantt diagram serves as the visual roadmap for the project's six-weeks execution, integrating the activities defined in the Work Breakdown Structure (WBS) with the estimated person-day effort and allocated human resources. In this visualization, key **deliverables** are explicitly marked with a diamond symbol (♦), while the qualitative and quantitative **milestones** required to achieve them are inherently met through the completion of each iterative step.

Agile Timeline

This agile timeline structures the project into **four iterations**, each delivering a usable increment of the system. Several tasks are intentionally **iterative**, reappearing across iterations with increasing depth, from initial feasibility tests to full-scale implementation. This **spiral approach** enables early validation, gradual automation, and controlled growth of both technical and semantic components, while keeping risks manageable and ensuring steady progress toward the final system. The Gantt diagram is available at: https://github.com/metamuses/becoming-barberotheca/blob/main/docs/gantt_diagram.pdf.

Iteration 1

Concept validation & basic MVP

1. Identify sources
[Metadata system → Requirements analysis]
(Domain Expert + PM, Data Steward) - **2 days**
Select representative lecture sources for feasibility testing.

2. Define metadata needed structure
[Metadata system → Requirements analysis]
(Data Steward + Domain Expert) - **2 days**
Define minimal metadata fields for a single item.

3. Compile initial raw dataset
[Metadata system → Requirements analysis]
(Data Steward + Domain Expert) - **3 days**
Manually compile metadata for the test lecture.

4. Define desired types and formats
[Data system → Requirements Analysis]
(Solution Architect + Data Steward) - **1 day**
Define initial audio and transcription formats, producing sample data.

5. Establish required tools
[Data system → Requirements Analysis]
(Solution Architect + PM) - **2 days**
Select tools for download, transcription, and processing.

6. Acquire audio files
[Data system → Pipelines Development] **(iter. 1)**
(Solution Architect + Data Steward) - **1 day**
Tentative acquisition of a single audio file to validate the pipeline.

7. UX design and layout definition
[User application → System Design] **(iter. 1)**
(UX/UI + Domain Expert) - **2 days**
Sketch basic layout and interaction for audio and transcript.

8. Media rendering component
[User application → Development] **(iter. 1)**
(Web Developer) - **2 days**
Implement a basic audio player to validate playback feasibility.

Iteration 2

Pipeline structuring & basic automation

1. Acquire audio files
[Data system → Pipelines Development] (**iter. 2**)
(Solution Architect + Data Steward) - **2 days**
Acquire a limited set of audio files to test repeatability.

2. Perform speech-to-text transcription
[Data system → Pipelines Development] (**iter. 1**)
(Solution Architect + Data Steward) - **3 days**
Test transcription automation and models on the limited corpus.

3. Harmonize semantic naming
[Metadata system → Tools & process development]
(Data Steward + Domain Expert) - **1 day**
Apply consistent naming across audio and transcripts.

4. Extract keyword and entities
[Metadata system → Tools & process development] (**iter. 1**)
(Solution Architect + Data Steward) - **2 days**
Develop a tool to test automated keyword and entity extraction.

5. UX design and layout definition
[User application → System Design] **(iter. 2)**
(UX/UI + Domain Expert, Web Developer) - **2 days**
Refine layout to support multiple items and navigation.

6. Data integration and logic
[User application → Development] (**iter. 1**)
(Web Developer) - **2 days**
Develop an initial integration of audio, transcripts, and enriched metadata.

Iteration 3

Semantic enrichment & application growth

1. Acquire audio files
[Data system → Pipelines Development] (**iter. 3**)
(Solution Architect) - **3 days**
Acquire the whole corpus of audio files.

2. Verify audio integrity
[Data system → Validation and Provisioning]
(QA) - **2 days**
Check completeness and correctness of audio assets.

3. Perform speech-to-text transcription
[Data system → Pipelines Development] (**iter. 2**)
(Solution Architect) - **4 days**
Finalize transcriptions on the whole corpus.

4. Review transcription quality
[Data system → Validation and Provisioning]
(Domain Expert + QA) - **3 days**
Validate transcription accuracy and consistency.

5. Media rendering component
[User application → Development] (**iter. 2**)
(Web Developer) - **3 days**
Extend the audio player with full controls and transcript synchronization.

6. Data integration and logic
[User application → Development] (**iter. 2**)
(Web Developer + Solution Architect) - **3 days**
Integrate audio, transcripts, and enriched metadata.

7. UX design and layout definition
[User application → System Design] (**iter. 3**)
(UX/UI) - **2 days**
Improve usability, readability, and navigation clarity.

8. Extract keyword and entities
[Metadata system → Tools & process development] (**iter. 2**)
(Solution Architect) - **3 days**
Finelize automated keyword and entity extraction tool.

9. Reconcile authority controlled entities
[Metadata system → Tools & process development]
(Domain Expert + Data Steward) - **3 days**
Validate entities against authority sources.

Iteration 4

Finalization, documentation, validation

1. Review automated metadata extraction
[Metadata system → Validation and materialization]
(QA) - **2 days**
Validate metadata consistency and correctness.

2. Optimize data for storage
[Data system → Pipelines Development]
(Solution Architect) - **2 days**
Introduce compression and storage conventions.

3. Research further concepts connections
[Metadata system → Tools & process development]
(Domain Expert) - **2 days**
Explore and formalize semantic relationships.

4. Review authority control connections
[Metadata system → Validation and materialization]
(Domain Expert + QA) - **2 days**
Check semantic links and entity mappings.

5. Provide format convertibility
[Metadata system → Validation and materialization]
(Solution Architect) - **1 day**
Ensure metadata export in reusable formats.

6. Produce final data assets for publication
[Data system → Validation and Provisioning]
(Solution Architect) - **1 days**
Finalize audio and transcription datasets.

7. User testing
[User application → Testing and Validation]
(UX/UI + Web Developer, QA) - **2 days**
Validate usability and overall coherence.

8. Document audio ingestion steps
[Configuration guide → Content Development]
(Data Steward) - **1 day**
Describe the audio acquisition workflow.

9. Document transcription workflow
[Configuration guide → Content Development]
(Data Steward) - **1 day**
Describe transcription parameters and process.

10. Document metadata production
[Configuration guide → Content Development]
(Data Steward + Domain Expert) - **1 day**
Describe metadata creation and validation steps.

11. Test-drive the guide
[Configuration guide → Review and Finalization]
(QA + Solution Architect) - **1 day**
Validate documentation through a test extension.

12. Select critical function/feature
[User manual → Requirement analysis]
(Web Developer + UX/UI, QA) - **1 day**
Select a specific functionality to document.

13. Define document template
[User manual → Content Development]
(UX/UI) - **1 day**
Create an ideal structure for manual.

14. Draft instructions
[User manual → Content Development]
(UX/UI + QA) - **1 day**
Create an ideal structure for manual.

15. Check accuracy
[User manual → Review and finalization]
(QA + UX/UI, Web Developer) - **1 day**
Validate the content of the manual.

16. Editorial review
[User manual → Review and finalization]
(QA + UX/UI) - **2 days**
Finalize the guide for release.

Risk analysis

Risk score table

Risk management involves identifying uncertain events or conditions that, if they occur, could have a positive or negative effect on the project's objectives. The identified risks correspond to three different high-level uncertainty types.

Unrealistic time estimation and **technical integration difficulties** are classified as process uncertainty, as they relate to the efficiency and effectiveness of the workflow.

External service unavailability is also categorized as a process uncertainty, given its potential to disrupt the operational pipeline. **Low AI transcription** quality represents a product uncertainty, as it concerns the risk of the project's outputs (the transcription corpus) not matching expectations. Finally, **personnel shortfalls** are identified as resource uncertainty, addressing potential gaps in team availability.

The following table details the risk analysis. Each identified risk was prioritized by assigning a specific score derived from the product of its likelihood (probability of occurrence) and its potential impact (severity of effect on budget).

Risk	**ID**	**Likelihood**	**Impact level**	**Score (LxI)**
Unrealistic time estimations	R1	Significant (5)	Significant (6)	**30**
Low AI transcription quality	R2	Moderate (4)	Significant (6)	**20**
Technical integration difficulties	R3	Moderate (3)	High (8)	**24**
Personnel shortfall	R4	Low (2)	Significant (9)	**18**
External services unavailability	R5	Low (2)	High (8)	**16**

Risk mitigation plan

The following table outlines the risk mitigation plan, detailing the corrective actions and strategies adopted to handle the identified risks. For each potential issue, specific mitigation techniques are proposed, aimed at minimizing the likelihood of occurrence or reducing the negative impact on the project's objectives.

ID	Mitigation
R1	**Unrealistic time estimations**: adopt incremental development to release core features first.
R2	**Low AI transcription quality:** improve software evaluation and assessment by testing the Whisper model on a sample dataset early; implement a strict human-in-the-loop validation process.
R3	**Technical integration difficulties**: conduct early feasibility and technical analysis of the semantic reconciliation APIs; create a prototype of the entity extraction pipeline.
R4	**Personnel shortfalls**: Identify key personnel early and focus on job matching and teambuilding to ensure retention.
R5	**External service unavailability:** perform infrastructure assessment to ensure fallback options (e.g., local caching of authority data) are in place; conduct early technical analysis of API stability.