MCSE 544
Software Design and Integration

Chapter 5: Feasibility Study & Requirements Definition

Dr. Mehedi Hasan
Stamford University Bangladesh

Feasibility Study

Before beginning a project, a short, low-cost study to **identify**

- Client
- Scope
- Potential benefits
- Resources needed:

  staff, time, equipment, etc.
- Potential obstacles

Where are the risks? How can they be minimized?

Feasibility Study

A feasibility study leads to a **decision:**

go ahead
do not go ahead
think again

In production projects, the feasibility study often leads to a **budget request**.

In research, a feasibility study is often in the form of a **proposal.**

CSI 331: Client

In CSI 331, you have two clients:

- The client for the project
- The professor for the course

Can you satisfy them both?

Scope

What are the boundaries of the project?

CSI 331 Examples:

- Static web pages with open access on the Web [Web Profiler]
- Used by the general public [Digital Collections]
- Varying data formats [Legal Information]
- Thousands of sensors [Data mining]
- Support for Windows, Mac, Unix [SALSA]

Potential Benefits

Why are you doing this project?

Examples

- Create a marketable product
- Improve the efficiency of an organization
- Control a system that is too complex to control manually
- New or improved service
- Safety or security
- **Get a good grade on CSI 331**

Resources

Examples: CSI 331

Staff: 4 to 6 students, with some help. How many hours per week? What skills do people have?

Time: Must be completed by end of semester, including operational system, documentation, presentation

Equipment and software: What special needs are there?

Client: Will the client be sufficiently available and helpful?

Obstacles

CSI 331 projects

Start-up time. Creating a team, scheduling meetings, acquiring software, learning new systems, ...

Business considerations. Licenses, trade-secrets, ...

Too ambitious. Nothing to show at the end of the semester.

Changing circumstances. Client leaves the university, ...

What else?

How to Minimize Risk?

CSI 331 Projects

- Several target levels of functionality:
  required, *desirable*, *optional*
- Visible software process: intermediate deliverables
- Good communication within team and with Teaching Assistant

Good processes lead to good software

Good processes reduce risk

Feasibility Report

A written document

- For a general audience: client, financial management, technical management, etc.
- Short enough that everybody reads it
- Long enough that no important topics are skipped

In CSI 331, I am looking for a well written, well presented document.

Feasibility Study

- Analysis undertaken before committing to a project
- Leads to a yes/no/rework decision
- Often contains a projected budget, and leads to a formal budget request
- Sometimes takes the form of a proposal

Major Difficulty: Uncertainty

- Benefits are hard to quantify
- Approach is usually ill-defined. Cost, risk and timetable estimates are very rough at this point
- Organizational/external changes may be needed for the project
- Feasibility studies depend heavily on the judgment of experienced leaders
- Mistakes made early in a project are often expensive to correct

The Importance of Advocacy

- Advocacy is often a major factor in overcoming the inaction that naturally accompanies uncertainty
- Enthusiasm is necessary, but enthusiasts under-emphasize costs and risks (sometimes intentionally; always subconsciously)
- The people evaluating a project often have an interest in seeing it go forward

The Decision Maker's Criteria

- Client: Who is the project for? Do we care?
- Scope: What are the boundaries of the project?
- Benefits: Can the benefits be quantified?
- Technical: Do we have a clear sense for how to implement the project?
- Resources: Estimated staff, time, equipment needs?
- Alternatives: What are the options if we don't do the project?

Decision Makers Hate Risk

- Technical
  - There must be a rough initial plan for completing the project
  - The plan must acknowledge substantial uncertainties
- External
  - Every project depends to some extent on external stakeholders
  - Are there potential users? people who don’t want to see the project succeed?

System Ecosystem

- Systems live in an ecosystem. Is there a friendly one for this project?
  - Management expertise in developing organization?
  - Technical expertise in-house or contracted?

Example 1

- Decision before feasibility report
- Government agency that manages a huge number of documents is slowly moving from paper to digital storage

Chronology

- University S developed a prototype system
- Funds were “procured” from congress to develop a major system
- The National Academy of Sciences was commissioned to do a feasibility study
- Problems:
  - The money was already there!
  - Feasibility study only looked at technical issues

Obvious Problems

- Organizational
  - The agency management were not prepared to lead a large technology project
  - No thought given to workflow and job changes
- Preparation
  - No preliminary study to evaluate the properties of the documents to be managed (volume, privacy, secrecy, etc)

Outcome: Deeply Flawed System

- No one wants to return money to congress
- Feasibility study was given short-shrift
- Agency adopted a pure waterfall model
- The system has had limited success

Scope

- The boundaries of the system
  - Included and excluded high-level functionality
  - Dependencies
  - Current systems to be replaced or augmented
- Confusion over scope is very common

Example 2

- Government library contracted out for a “repository system” to manage data
- Contractor built software to store and manipulate complex digital material
- Nobody built a system to load and validate the database in the first place
- The library thought that was implicit in the project
- The contractor disagreed

Benefits

- Can potential benefits be quantified?
  - Marketable product? Market size?
  - Improved efficiency?
  - New capability?
  - Improved safety or security?
- Personal development is not a good justification for a project, but organizational development can be

Technical Feasibility

- Rough draft of the requirements
- Rough draft of high level design (i.e. architecture)
- List of expected frameworks/libraries to be acquired/used
- High-level scale estimates (number of users, size of data, etc)
- Assessment of existing team’s experience

Planning and Resources

- A feasibility study must have a rough plan
- Estimates for staffing, equipment needs, project timetable
- Major milestones and decision points
- Interactions with external stakeholders
- Deliverables and delivery dates
- (Later lecture on planning details)

Alternatives and Risks

- Major alternatives
  - Revise or rewrite?
  - In-house or contracted out?
  - Delivery phases and points to change course
- Risks
  - What can go wrong?
  - How will we know (visibility)?
  - Are there palatable alternatives?

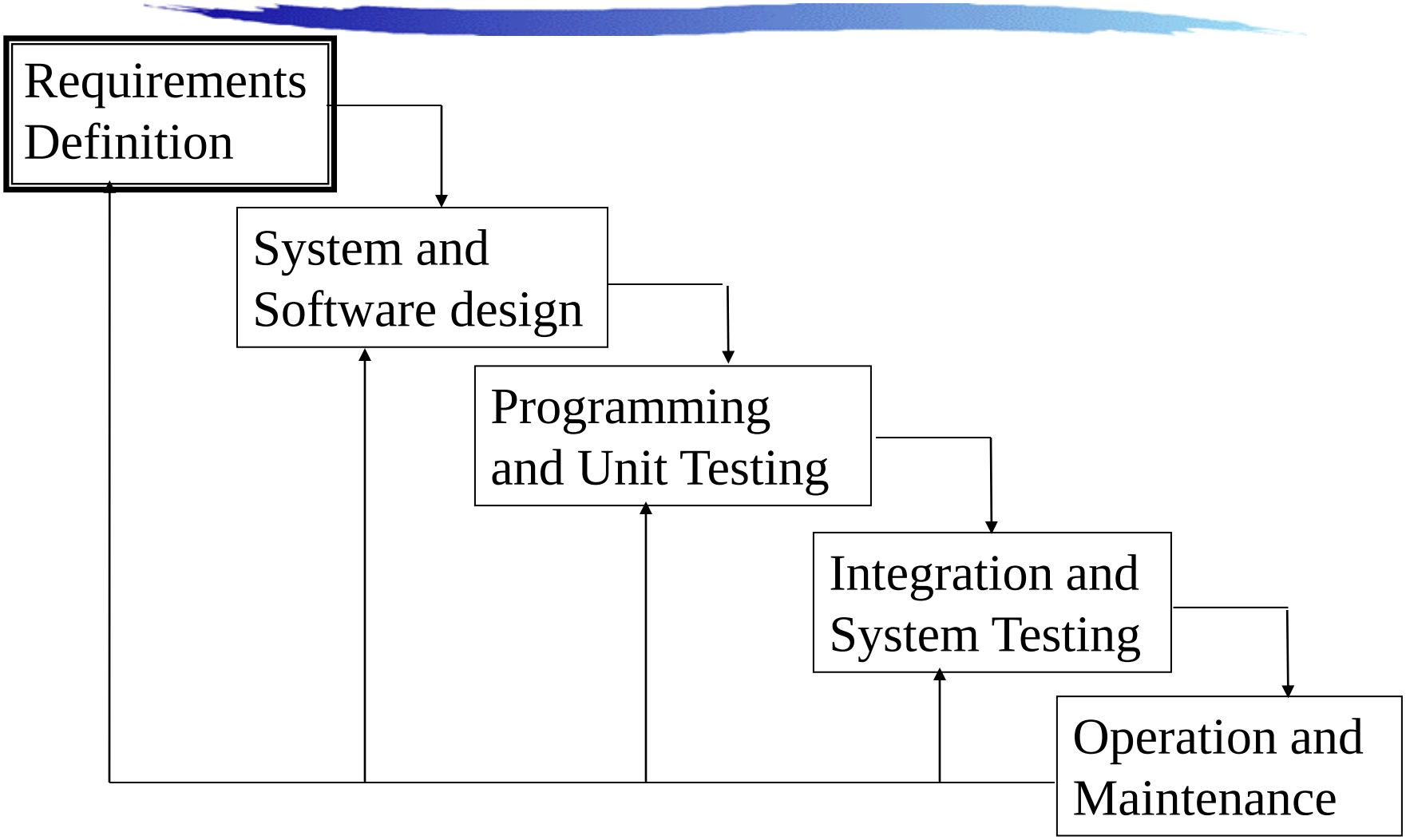
The Report

- A written document to be archived with the project
- Must be accessible to all stakeholders
- **Short** enough that people will read it
- Clear writing is crucial
  - Should spend more time writing and revising than any other document in the project

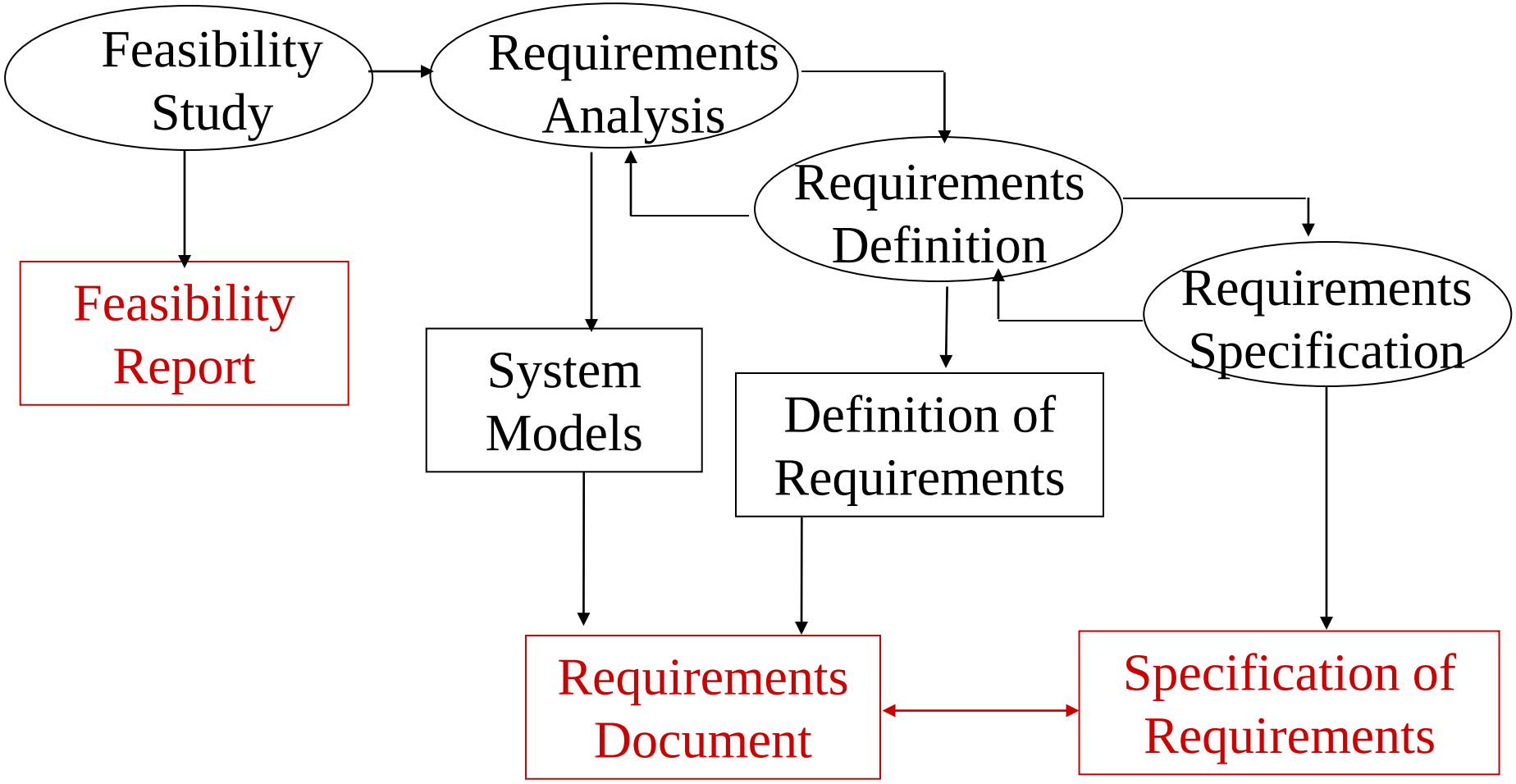
Use Case Stories

- Powerful tool for bridging the gap between users/clients and developers
- Write short “stories” with “characters” using the software you are going to develop
- Keep them short!
- Add just enough fun detail to get the brain in story mode

Requirements Definition and Analysis



The Requirements Process



Requirements Definition

High-level abstract description of requirements:

- Specifies external system behavior
- Comprehensible by customer, management and users

Should reflect accurately what the customer wants:

- Services that the system will provide
- Constraints under which it will operate

Example: Library of Congress (A Partial Failure)

Outline Description

The Library of Congress requires a repository system to store and make accessible very large amounts of highly varied material over long periods of time.

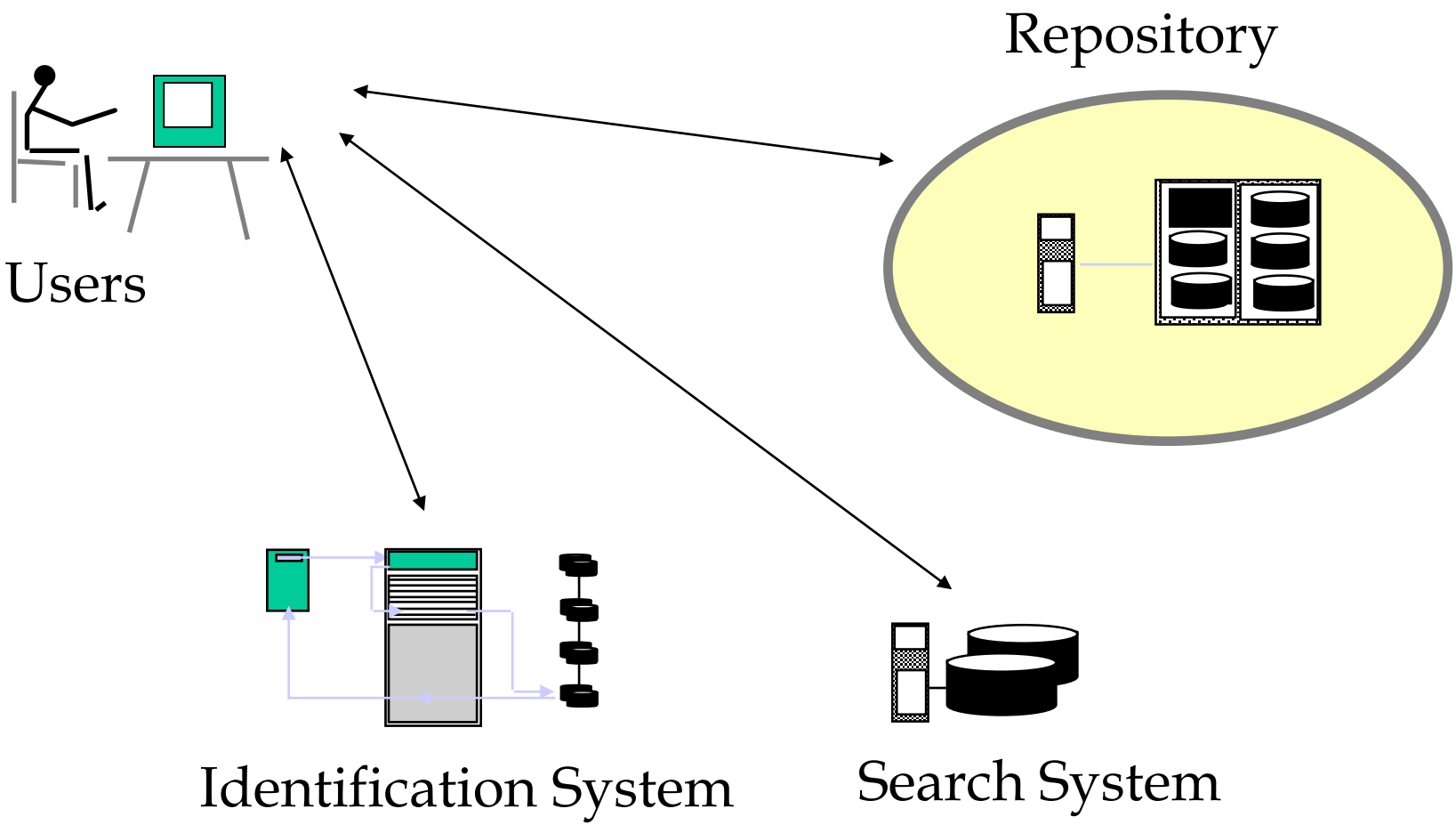
Chronology

1993-94 CNRI carries out research on architectures for digital libraries

1995-97 CNRI implements prototype repository for Library of Congress

1998 CNRI and Library of Congress carry out requirements definition

The Repository



Storage and Representation of Complex Objects

Data

Several representations:

thumbnail image
reference image
archival image

Metadata

Each representation may have its own metadata

Repository: Research Achievements

1. CORBA implementation of repository access protocol.
2. Integration of persistent naming through handle system.
3. Use of structural metadata to describe complex objects, elementary typology.
4. Access management framework and implementation.
5. Applet-based middleware for user interfaces.
6. Information visualization program to view the structure of large collections.

Good Discoveries During Prototype

- Structuring complex information in digital libraries
- Data driven digital library interfaces
- Comparison of object-oriented, relational, and file based storage systems
- Naming and identification of library objects
- Boundaries of required repository system

Bad Discoveries During Prototype

- Resistance to change within Library of Congress
- Technical weakness of Library of Congress
- Gaps in CNRI architecture

Mistakes

- Confusion of objectives (research and implementation)
- Failure to involve all stakeholders
- Over-ambitious (no proper feasibility study)

Library of Congress Requirements Study

Team (all experienced): Librarian, Software Engineer (CNRI), Computing Project Leader (Library of Congress), + 2 others

Advisors: Mailing list of about 20 knowledgeable stakeholders.

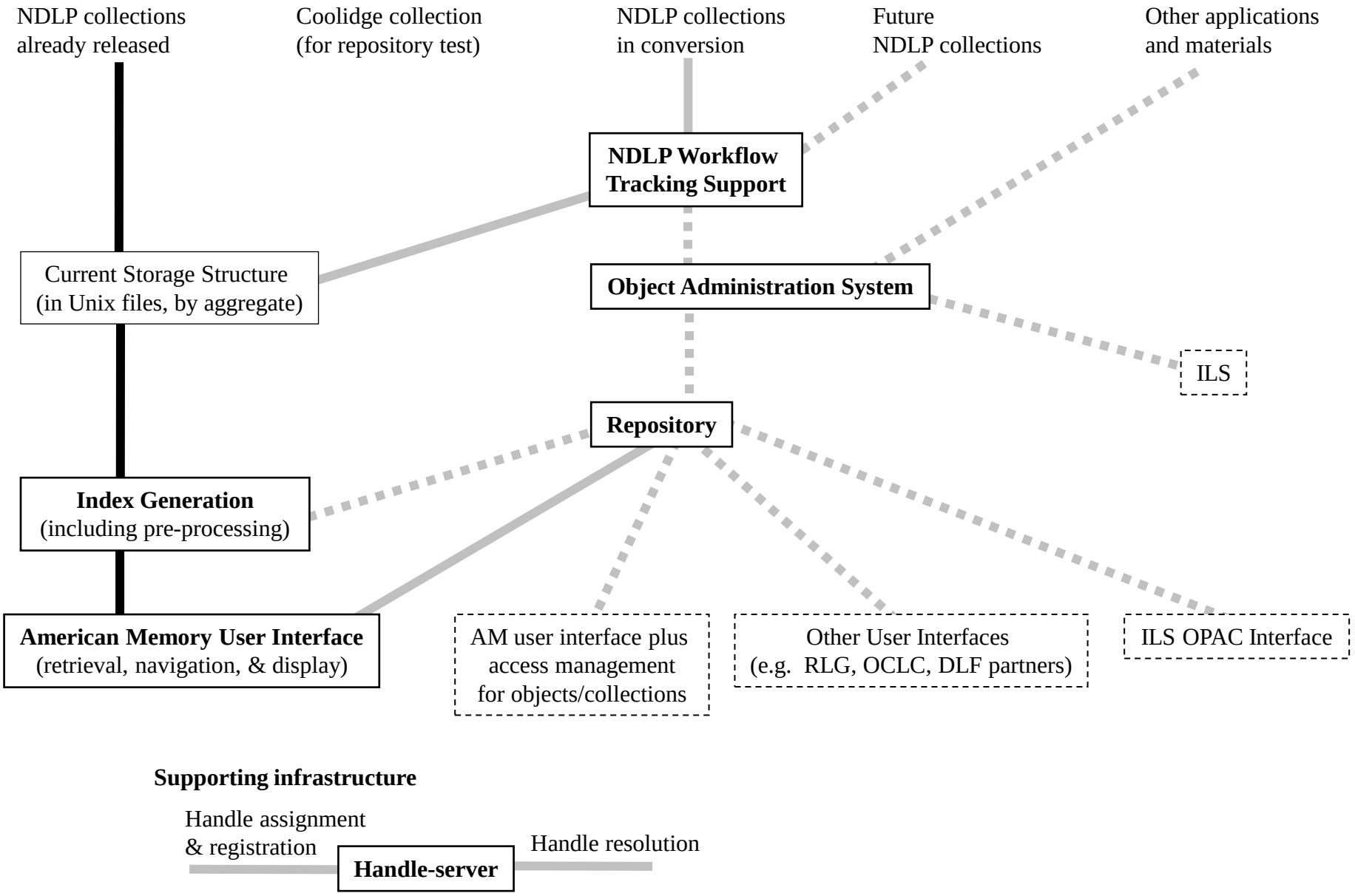
Timetable: Preliminary report (2 months). Final report (1 month).

Functional Requirements

Example: Library of Congress repository

- Support for complex digital objects
- Access management
- Identification
- Information hiding
- Open protocols and formats
- Integration with other systems (scope)

DRAFT OVERVIEW OF ITS SUPPORT FOR NDLP PRODUCTION AND DELIVERY OF AMERICAN MEMORY



Non-functional Requirements

Environment:

- Estimates of sizes, numbers of users, etc.
- Reliability and performance measures and targets

Preferred:

Example: Library of Congress repository

- Hardware and software systems (e.g., IBM/Unix)
- Database systems (e.g., Oracle)
- Programming languages (e.g., C and C++)

Evolution of Requirements

- If the requirements definition is wrong, the system will be a failure.
- With complex systems, understanding of requirements always continues to improve.

Therefore...

- The requirements definition must evolve.
- Its documentation must be kept current (but clearly identify versions).