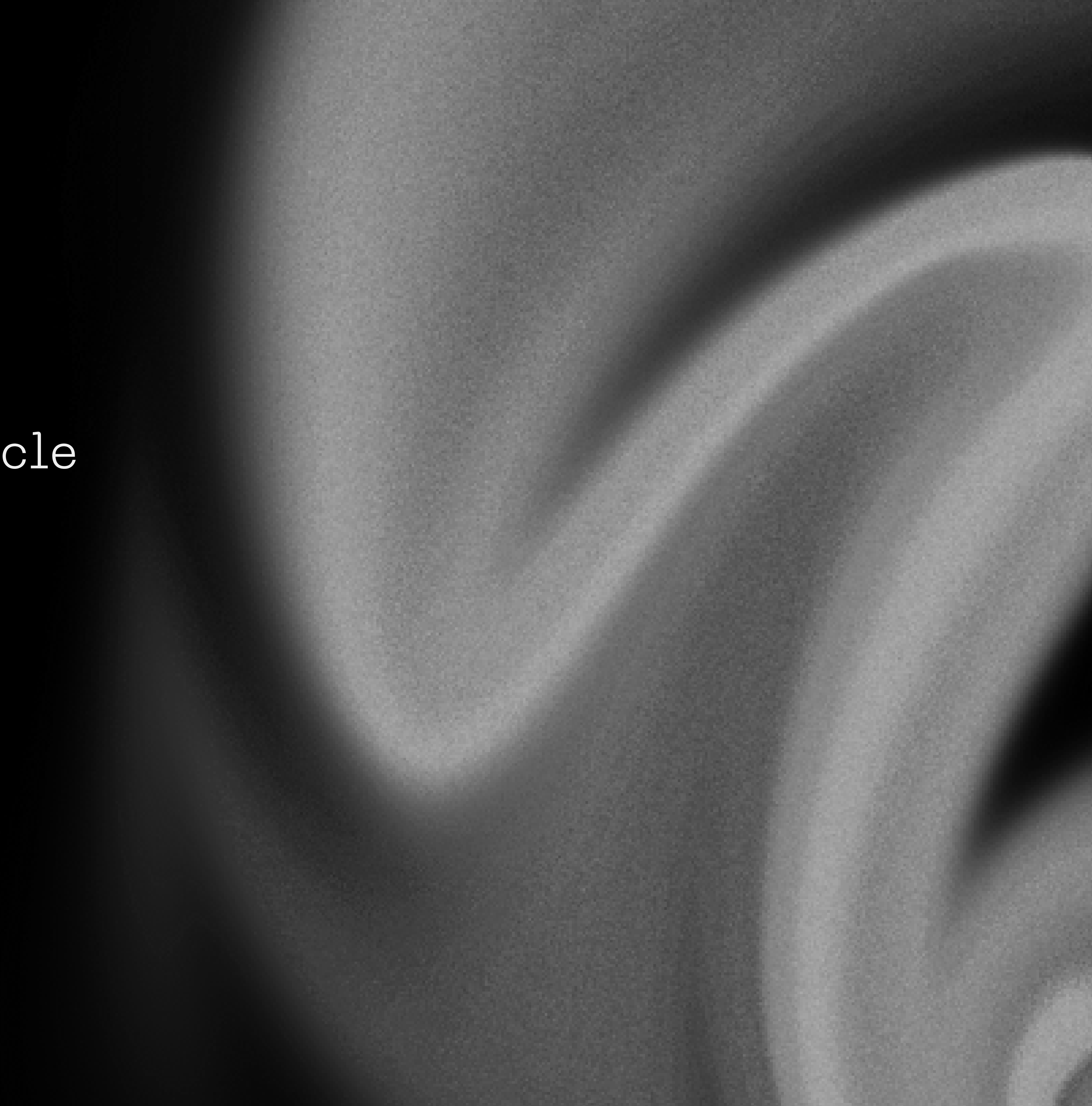
SDLC

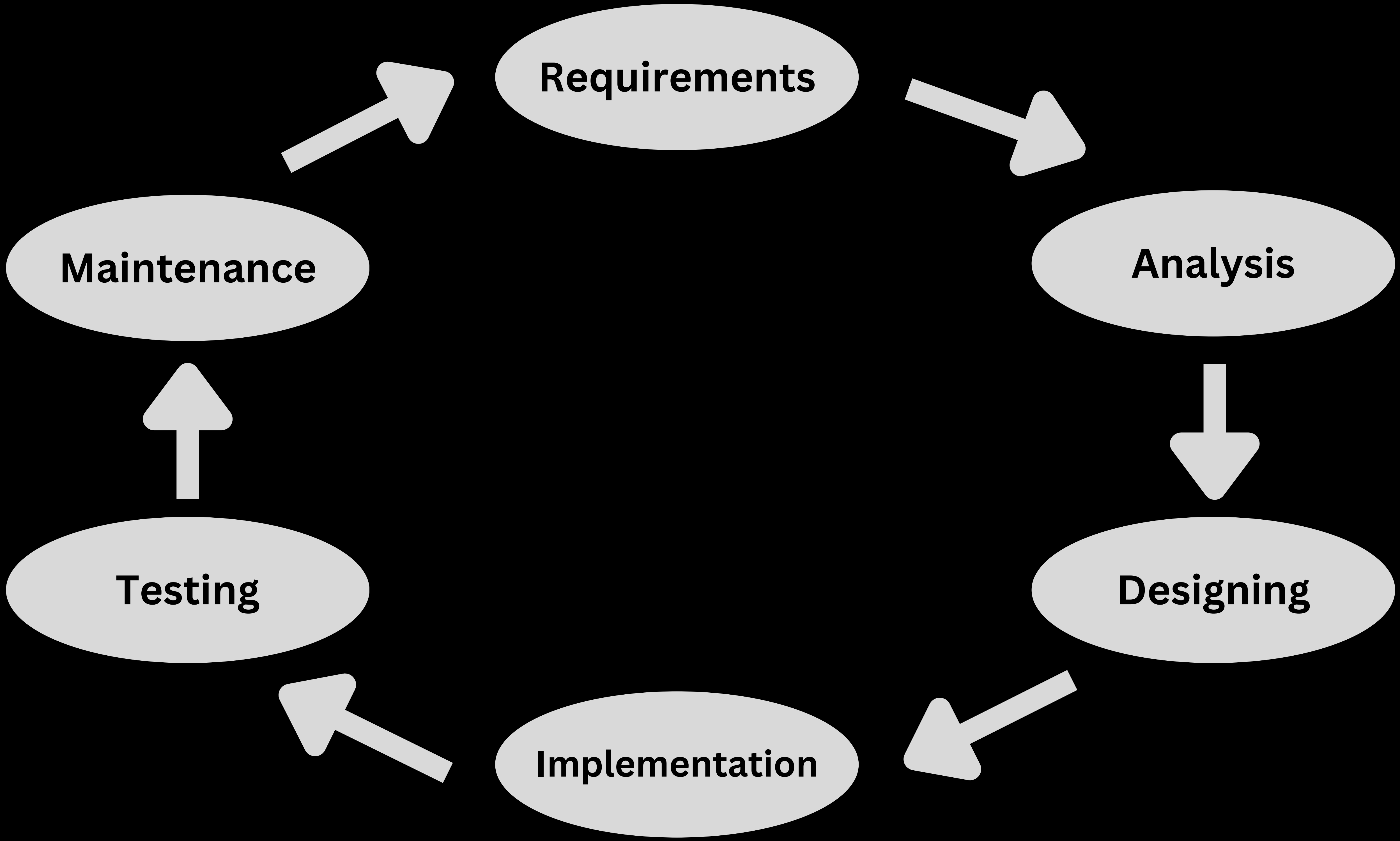
Software Development Life Cycle



SDLC Methodology

- The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software. in detail, the SDLC methodology focuses on the six phases of software development:

SDLC Methodology



Requirements

Analysis

Designing

Implementation

Testing

Maintenance

1. Requirement Gathering

Phase of collecting requirements from client Will be done by business analyst of company.

He will create questioner, in which put answers from client.

2. Analysis & SRS (Software Requirement Specification)

- Collected requirements will be analyzed for time limit, budget and market trade will be filtered and SRS will be created as result will be discussed with client.
- Based on requirements users, their activities and flow of data, modules will be defined.
- Will be done by system analyst.
- Based on diagrams all will be done.

3. Designing

- The Design document should reference what you are going to build to meet the requirements.

- Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary.

3. Designing

These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

4. Implementation (Coding)

- To launch the coding phase, develop a shell program that is then put under some form of version control.
- This phase includes the set up of a development environment, and use of an enhanced editor for syntax checking.

5. Testing

- Each developer insures that their code runs without warnings or errors and produces the expected results.
- The code is tested at various levels in software testing.
- Types of testing: Defect testing, Path testing, Data set testing, Unit testing, System testing,Automation testing, Performance testing, etc.

6. Maintenance

- User’s guides and training are developed to reflect any new functionality and changes which need to be identified to the production staff.
- Any changes needed to operations and/or maintenance need to be addressed.
- Every run in production needs to be verified. Any problems with production need to be addressed immediately.

Thank you!