You are improving a legacy system. You need to customize a software life cycle for your project. Which activities would you need? In which order?

Step 1:

A legacy system is a piece of computing equipment or software that is no longer supported. The system continues to satisfy the requirements for which it was first intended, but growth is not permitted. A legacy system will only ever do the current tasks for the business.

Step 2:

The three key steps in customising the software life cycle are as follows:

Replace

When you have already undergone a significant level of organisational or technological change and have little internal capacity to tolerate much more change, the replace method can be quite disruptive. Additionally, replacing legacy systems is excessively expensive, while in certain cases it could be the best option.

Extend

You probably wouldn't need to replace your legacy programme if it has been around for a while and continues to carry out its essential responsibilities. You can still update it by expanding its capabilities, though. Other advantages of extending your old software are also available. When you use this approach to successfully upgrade historical programmes, you build momentum for future digitalization.

Migrate

Legacy migration may be the ideal option for your needs if your firm wants to transform important operations right away and your legacy software isn't flexible enough to accommodate the anticipated adjustment or if your legacy application no longer provides the correct user experience. Delivering a more dependable version of the systems your employees are accustomed to while also demonstrating agility, migration of legacy systems, and giving them the flexibility and scalability that only modern infrastructure can offer.