1.1 Software Processes

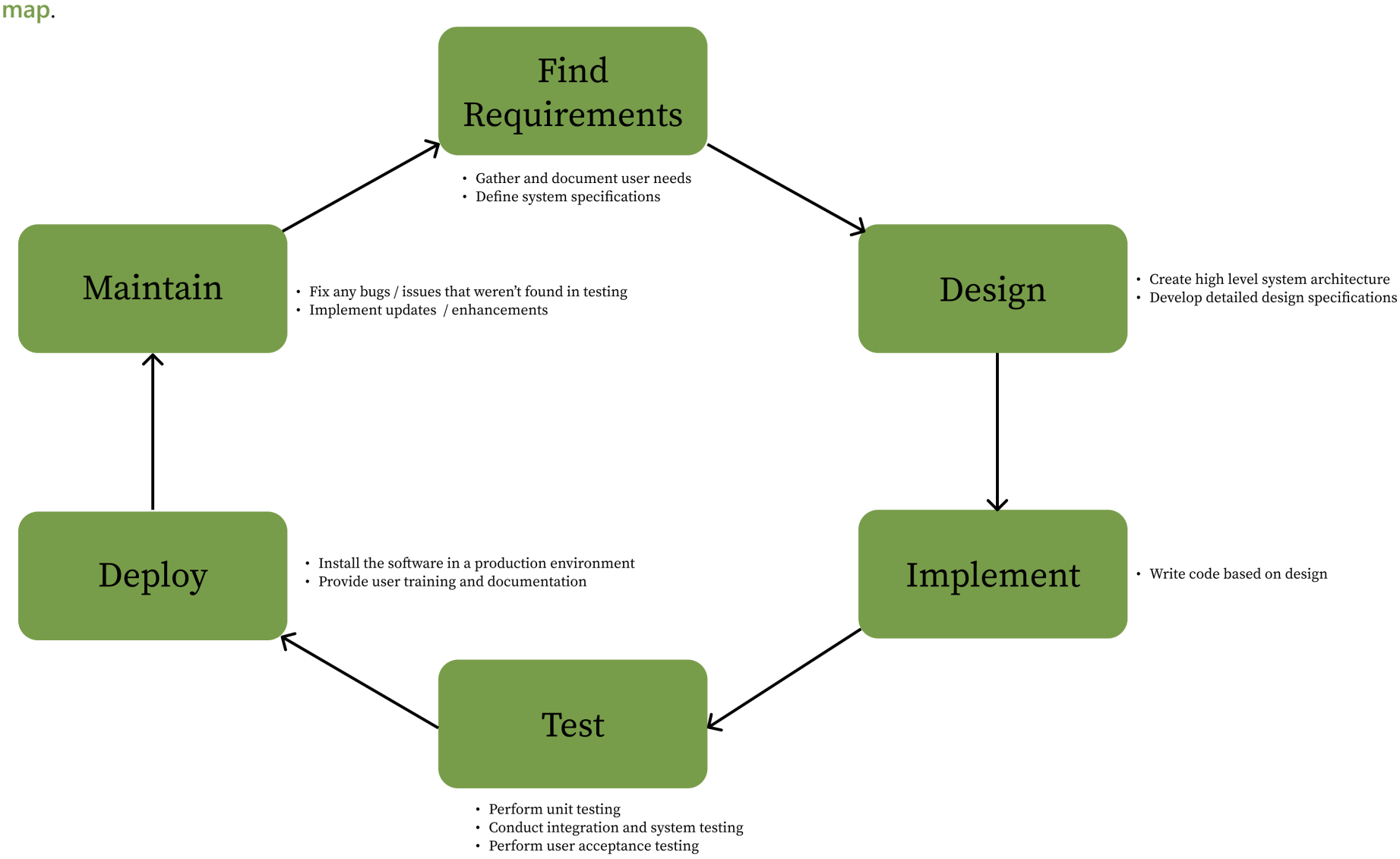
Software Processes

A structured set of activities to produce or maintain a software product. These processes allow us to:

- Improve efficiency in software development
- Ensure quality and reliability of software products
- Manage resources effectively
- Meet project deadlines and milestones
- Facilitate communication among team members
- Standardize development practices
- Track progress and identify issues early

Software Process Model

An abstract representation of a process used to describe steps and guide software teams throughout development. It acts as a roadmap.



Main Software Development Methodologies

Two primary approaches to software development processes:

Waterfall Model

- Example: Developing software for a heart surgery robot
  - Requirements fully defined before design begins
  - Rigorous testing before deployment
  - Minimal changes after implementation
- Resembles a waterfall cascading from one stage to the next
- Key memory aid:
  - Lots of work at the top (extensive upfront planning)
  - Once flowing, it easily progresses downward (because gravity!)
  - Difficult to go back upstream (challenging to make changes)
  - Kinda like eating all of your vegetables at dinner first (challenging at first, but smooth sailing after)

Agile Methodology

- Example: Developing an online multiplayer game
  - Features added and balanced iteratively
  - Regular updates based on player feedback
  - Continuous refinement of gameplay mechanics
- Named for its ability to move quickly and easily
- Key memory aid: Think of an agile athlete, able to change direction rapidly

Aspect	Waterfall (Heart Surgery Robot)	Agile (Online Multiplayer Game)
Requirements	Fully defined upfront	Evolve as development progresses
Testing	Comprehensive testing phase	Continuous testing throughout
Delivery	Single release after full development	Regular updates and patches
Risk Management	Extensive initial risk assessment	Ongoing risk evaluation and mitigation
Changes	Difficult and costly to implement	Expected and easily accommodated
User Feedback	Limited, mainly in initial stages	Continuous through beta testing and live service
Documentation	Extensive and detailed	Lean, focuses on essential information
Project Predictability	High, with less flexibility	Lower, but with higher adaptability