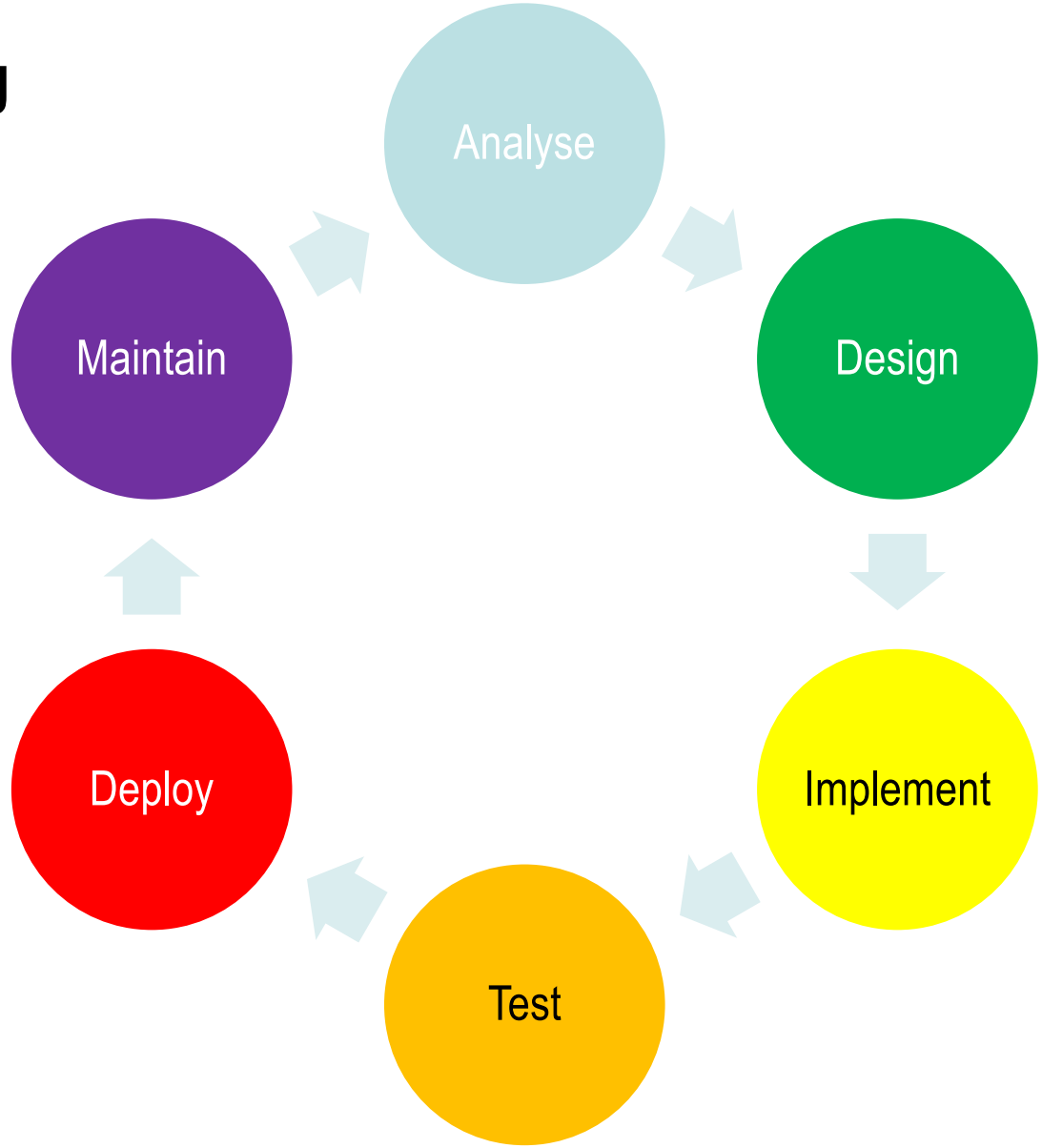
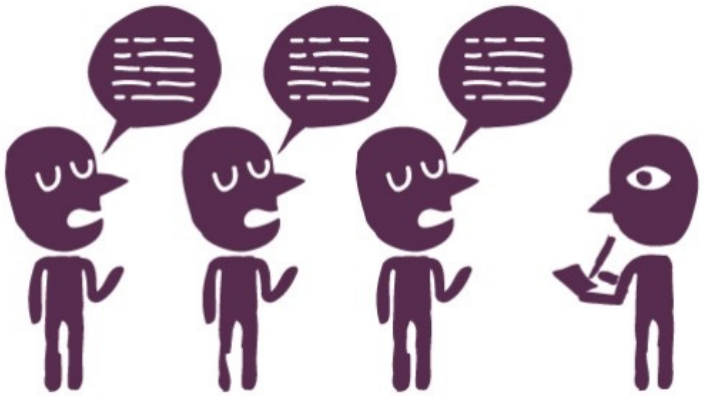
The Software Development Life Cycle

- Requirements gathering
- Design
- Implementation
- Quality Assurance
- Deployment
- Maintenance



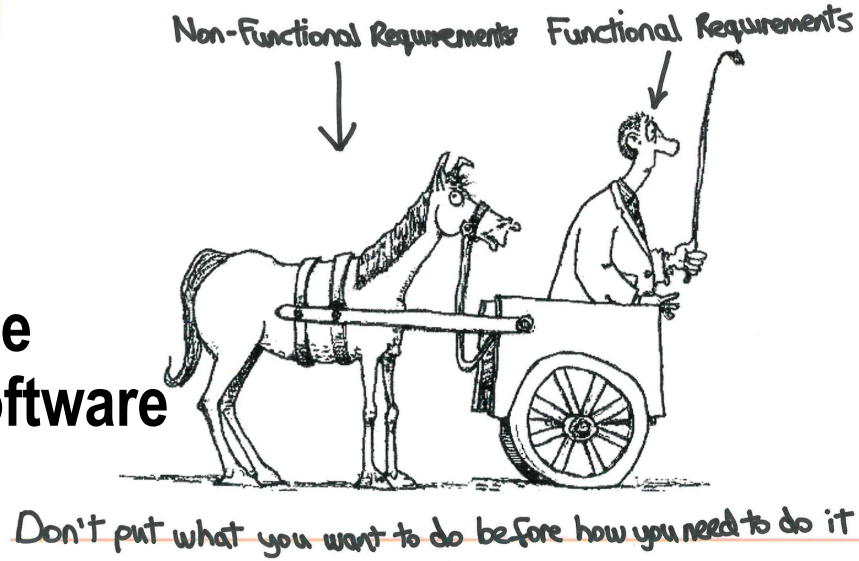
Requirements Gathering

- The first step of any software project is to figure out what it is that you are building
  - Functional requirements: What should it do?
  - Nonfunctional requirements: Other
- Requirements are gathered from:
  - RFP: Request for Proposal
  - Existing user base
  - Prospective user base
  - Management
  - Etc.



- A Software Requirements Specification (SRS) is a document that specifies the software to be built: all functional and non-functional requirements

Functional Requirements



- Functional requirements describe the functions of the proposed software
  - What should it do?
  - What resources can it use?
  - What performance benchmarks should it meet?
  - What other systems should if integrate with?
  - What security standards should it meet?
- Example: The assignment and practicum specifications:
  - Input
  - Processing
  - Output
- This does not typically specify the “how”, just the “what”

Nonfunctional Requirements

- Nonfunctional requirements specify requirements not related to software behaviour
  - Language requirements
  - Budgetary requirements
  - Implementation deadlines
  - Etc
- Examples:
  - You have 2 week to implement Assignment 6
  - You have 90 minutes for Practicum 5
  - You must use Java to implement your assignments and practicums

Object Oriented Design

- Once we have a requirements specification, what's next?
- Assumption: We will use Object Oriented Design
- Hence, we will need to create classes and objects that collaborate to meet the requirements
- Hence. We will need to figure out
  - What classes (and objects) we need?
  - What do these classes and objects do?
  - How do these classes and objects collaborate?