***Software Engineering*** – The application of scientific principles to the design and creation of software  
- Uses a systematic approach to collect business requirements and design, build, and test software

**Computer-Aided Software Engineering (CASE):**

- Tools that came to prevalence in the 196- 1980’s to alleviate the “software crisis” and help align discordant software engineering frameworks  
- Can be divided into six categories:

1. Business Analysis & Modeling  
2. Debugging Environments  
3. Verification and Validation tools  
4. Configuration Management  
5. Metrics and Measurement  
6. Project Management

**Software Engineer vs Software Developer:**

**A screenshot of a computer

Description automatically generated  
A blue and white text on a white background

Description automatically generated  
A blue background with white text

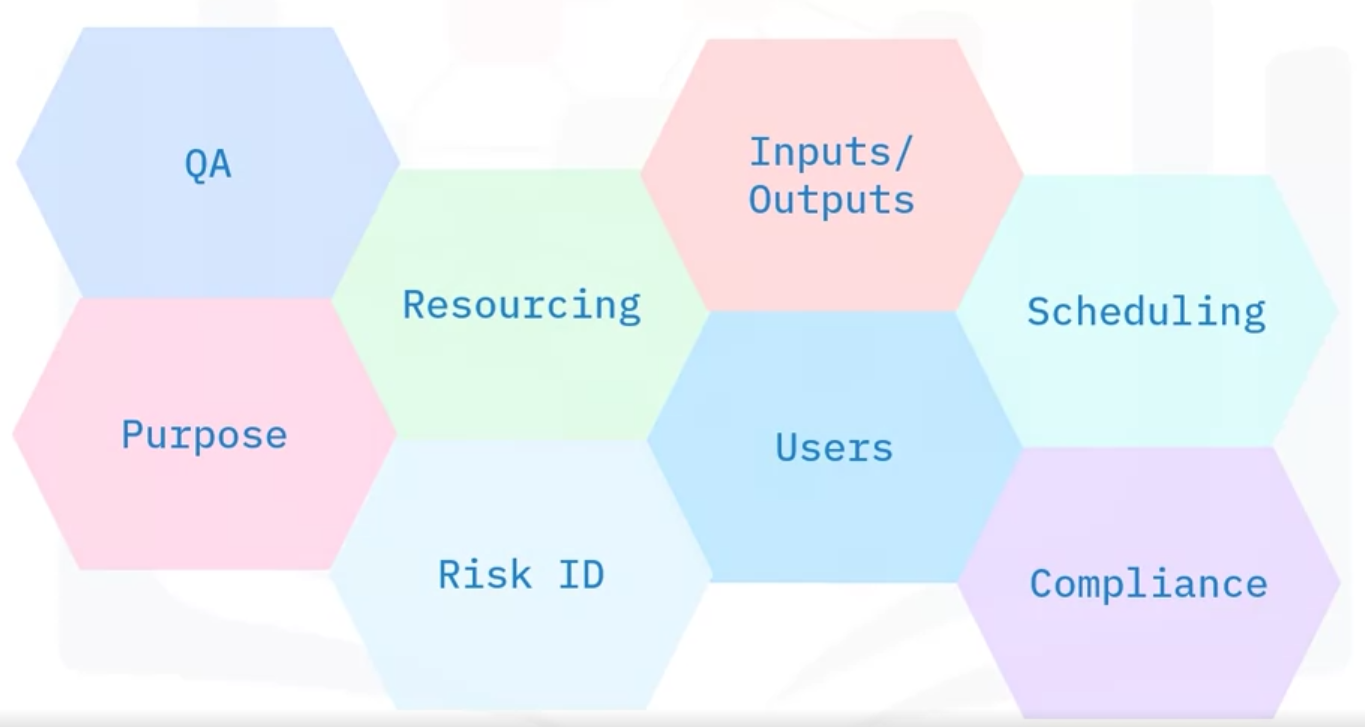
Description automatically generated**

- Engineering encompasses system design and architecture   
- Goes beyond just ‘writing code’

**Software Development Life Cycle (SDLC)**

- A scientific approach to software development  
- Identifies steps needed to develop high quality software and guides the software development process to meet a client’s requirements  
- Defines phases with their own processes and deliverables (initially used waterfall method for development but now uses Agile and DevOps – more iterative methods)  
- Six Phases that are all independent on the one before being complete – utilize agile or waterfall methods  


**Phase 1 – Planning:**

- Requirements are gathered, analyzed, documented, and prioritized  
- Need to consider all of these factors in this phase  


- Labor and material costs are identified and weighed against project time constraints  
- Project teams are identified and individual roles are listed  
- Development team can produce a prototype during the planning stage  
- **Software Requirements Specification (SRS)** – All requirements are documented in this document and all stakeholders must agree to it

**Phase 2 – Design:**  
  
- **Design Document** - Requirements from SRS are used to develop the software architecture designwhich is maintained in this doc  
- Can also design a prototype in this stage

**Phase 3 – Development:**

- Developers begin coding process using the Design Document  
- Project planners determine and assign program tasks

**Phase 4 – Testing:**

- Code needs to be tested to ensure it is stable, secure, and meets requirements outlined in SRS  
- Testing can be manual, automated, or a hybrid of both   
- Bugs are reported, tracked, fixed, and retested until software is stable  
- Testing levels include: Unit, Integration, System, and Acceptance

**Phase 5 – Deployment:**

- Code is deployed into production environment and made available to users  
- Usually done in stages with code being deployed to a **User Acceptance Testing Environment** (beta/gamma) and once customer accepts functionality it is then fully deployed into production

**Phase 6 – Maintenance:**

- Occurs after code has been deployed to prod  
- Used to find other bugs, identify user interface issues, and fast follow new/changing requirements  
- Also includes additional code enhancements that didn’t make it into MVP of product