**Software Engineering**:  
Software engineering is the systematic application of engineering approaches to the development of software. It differs from traditional programming in that it involves a structured process that includes planning, development, testing, and maintenance of software, rather than just writing code¹l.

**Software Development Life Cycle (SDLC)\*\*:  
The SDLC phases include:**1. \*\*Planning\*\*: Establishing the scope and defining resources.  
2. \*\*Analysis\*\*: Gathering detailed business requirements.  
3. \*\*Design\*\*: Translating requirements into software architecture.  
4. \*\*Development\*\*: Writing the actual code.  
5. \*\*Testing\*\*: Verifying that the software meets all specifications.  
6. \*\*Deployment\*\*: Releasing the software to users.  
7. \*\*Maintenance\*\*: Ongoing support and updates.

**Agile vs. Waterfall Models**  
Agile is an iterative and incremental approach that emphasizes flexibility and customer collaboration. Waterfall is a linear and sequential approach that is more rigid and structured. Agile is preferred in dynamic environments with changing requirements, while Waterfall is suited for projects with well-defined requirements and less likelihood of change.

**Requirements Engineering**  
Requirements engineering is the process of defining, documenting, and maintaining the requirements of a software system. It's crucial because it ensures the software will meet the needs of users and stakeholders throughout its lifecycle.

**Software Design Principles - Modularity:**  
Modularity in software design refers to dividing software into separate components that can be developed, tested, and maintained independently. It improves maintainability by isolating changes to individual modules and scalability by allowing the system to expand without affecting existing components.

**Testing in Software Engineering**  
Levels of software testing include:  
- \*\*Unit Testing\*\*: Testing individual components.  
- \*\*Integration Testing\*\*: Testing combined components.  
- \*\*System Testing\*\*: Testing the complete system.  
- \*\*Acceptance Testing\*\*: Validation against user requirements.  
Testing is crucial to ensure the software is reliable, secure, and meets user expectations.

**Version Control Systems**:  
Version control systems manage changes to source code over time, enabling collaboration, tracking changes, and facilitating error recovery. Examples include Git, Subversion, and Mercurial, each with features like branching, merging, and commit history.

**Software Project Management**  
A software project manager leads the development team, ensuring projects are completed on time, within budget, and meet quality standards. They handle planning, resource allocation, risk management, and stakeholder communication.

**Software Maintenance**:  
Software maintenance includes:  
- \*\*Corrective\*\*: Fixing defects.  
- \*\*Adaptive\*\*: Updating the software for new environments.  
- \*\*Perfective\*\*: Enhancing features for performance.  
- \*\*Preventive\*\*: Preventing future issues.  
Maintenance is essential to correct faults, improve performance, adapt to environment changes, and prevent potential future problems.

**Ethical Considerations in Software Engineering**:  
Ethical issues include ensuring the public interest, maintaining privacy and security, and avoiding conflicts of interest. Engineers adhere to ethical standards by following professional codes of conduct and considering the impact of their work on society.

Source  
(1) Software engineering - Wikipedia. https://en.wikipedia.org/wiki/Software\_engineering.  
(2) The Software Development Life Cycle (SDLC): 7 Phases and 5 Models. https://theproductmanager.com/topics/software-development-life-cycle/.  
(3) Agile vs. Waterfall | Pros, Cons, and Key Differences - ProductPlan. https://www.productplan.com/learn/agile-vs-waterfall/.  
(4) Requirements Engineering Process in Software Engineering. https://www.geeksforgeeks.org/software-engineering-requirements-engineering-process/.  
(5) Modularity and its Properties - GeeksforGeeks. https://www.geeksforgeeks.org/modularity-and-its-properties/.  
(6) Levels of Software Testing - GeeksforGeeks. https://www.geeksforgeeks.org/levels-of-software-testing/.  
(7) What is version control? Definition, types, systems and tools. https://blog.logrocket.com/product-management/version-control-systems-definition-types/.  
(8) Software Project Manager Roles and Responsibilities. https://project-management.com/project-manager-roles-responsibilities-software-projects/.  
(9) The 4 Main Categories of Software Maintenance - scrums.com. https://www.scrums.com/guides/the-4-main-categories-of-software-maintenance.  
(10) Ethical Considerations In Software Engineering - techmaish.com. https://www.techmaish.com/ethical-considerations-in-software-engineering/.  
(11) What Does a Software Engineer Do? | Coursera. https://www.coursera.org/articles/software-engineer.  
(12) Software engineering Definition & Meaning - Merriam-Webster. https://www.merriam-webster.com/dictionary/software%20engineering.  
(13) What is Software Engineering? - Definition from Techopedia. https://www.techopedia.com/definition/13296/software-engineering.  
(14) Understanding the SDLC: Software Development Lifecycle Explained. https://resources.github.com/software-development/what-is-sdlc/.  
(15) What is the software development lifecycle (SDLC)? Phases and models .... https://blog.logrocket.com/product-management/software-development-lifecycle-sdlc-phases-models/.  
(16) What Is SDLC? Understand the Software Development Life Cycle. https://stackify.com/what-is-sdlc/.  
(17) Software Development Life Cycle (SDLC) Phases & Models - Guru99. https://www.guru99.com/software-development-life-cycle-tutorial.html.  
(18) Effective Modular Design in Software Engineering - GeeksforGeeks. https://www.geeksforgeeks.org/effective-modular-design-in-software-engineering/.  
(19) Modularity - Wikipedia. https://en.wikipedia.org/wiki/Modularity.  
(20) What is Modularity in Software Engineering | Institute of Data. https://www.institutedata.com/us/blog/modularity-in-software-engineering/.  
(21) Modularity - Explanation & Examples | Secoda. https://www.secoda.co/glossary/modularity.  
(22) Agile vs Waterfall – Difference Between Methodologies - Guru99. https://www.guru99.com/waterfall-vs-agile.html.  
(23) Agile vs. Waterfall Methodology – Forbes Advisor. https://www.forbes.com/advisor/business/agile-vs-waterfall-methodology/.  
(24) Agile vs Waterfall: Difference Between Two Powerful Methodologies. https://hygger.io/guides/agile/agile-vs-waterfall/.  
(25) Agile vs Waterfall: A Comprehensive Comparison. https://www.launchnotes.com/blog/agile-vs-waterfall-a-comprehensive-comparison.  
(26) Requirements engineering - Wikipedia. https://en.wikipedia.org/wiki/Requirements\_engineering.  
(27) An Insight into Requirements Engineering Processes. https://link.springer.com/chapter/10.1007/978-3-642-35615-5\_48.  
(28) Requirements Engineering: Fundamentals, Principles, and Techniques .... https://link.springer.com/book/9783642125775.  
(29) What is version control | Atlassian Git Tutorial. https://www.atlassian.com/git/tutorials/what-is-version-control.  
(30) The Importance of Version Control Systems in Software Development. https://moldstud.com/articles/p-the-importance-of-version-control-systems-in-software-development.  
(31) What is version control? | GitLab. https://about.gitlab.com/topics/version-control/.  
(32) Version Control — Why Do We Need It? | by Lance Harvie | Medium. https://medium.com/@lanceharvieruntime/version-control-why-do-we-need-it-1681f4888cec.  
(33) Levels of Testing: A Complete Approach to Quality Assurance. https://testsigma.com/blog/levels-of-testing/.  
(34) The 4 Levels of Testing in Software Engineering Explained. https://fellow.app/blog/engineering/the-levels-of-testing-in-software-engineering-explained/.  
(35) Levels of Testing in Software Testing - Guru99. https://www.guru99.com/levels-of-testing.html.  
(36) Code of Ethics for Software Engineers - IEEE Computer Society. https://www.computer.org/education/code-of-ethics.  
(37) Ethics in the Software Development Process: from Codes of ... - Springer. https://link.springer.com/article/10.1007/s13347-021-00451-w.  
(38) Ethical considerations in software engineering - IEEE Computer Society. https://www.computer.org/csdl/proceedings-article/icse/1991/00130652/12OmNviHKnL.  
(39) Ethical Considerations in Case Studies - IEEE Computer Society. https://www.computer.org/csdl/proceedings-article/rethics/2020/835000a015/1nYsPs1sDVC.  
(40) What is Software Maintenance: Importance, Types, Phases, and Models. https://www.simform.com/blog/software-maintenance/.  
(41) 4 Different Types of Software Maintenance [A Detailed Guide]. https://www.spaceo.ca/blog/types-of-software-maintenance/.  
(42) SDLC Guide: A Comprehensive Guide to Effective Software Maintenance .... https://stratoflow.com/software-maintenance-process/.  
(43) 4 Types of Software Maintenance - Thales Group. https://cpl.thalesgroup.com/software-monetization/four-types-of-software-maintenance.  
(44) Role and Responsibilities of a software Project Manager - Software .... https://www.geeksforgeeks.org/software-engineering-role-and-responsibilities-of-a-software-project-manager/.  
(45) What Is the Role of a Project Manager in Software Development?. https://www.developer.com/project-management/role-of-a-project-manager-in-software-development/.  
(46) What does a software project manager do? (Duties and skills). https://uk.indeed.com/career-advice/finding-a-job/what-does-software-project-manager-do.