Four fundamental skills of software engineers: (1) how to identify, evaluate, choose and implement appropriate methods and tools (2) how to select languages (3) how to test software (4) how to maintain software. I think how to maintain software is most important.

Software engineering (SE) adopts engineering approaches, such as established methodologies, processes, tools, standards, organization methods, management methods, quality assurance systems, and the like, in the development of large-scale software seeking to result in high productivity, low cost, controllable quality, and measurable development schedule.

Methodologies of SE: structured methodologies, object-oriented methodologies and agile mythologies.

Knowledge engineering (KE) is another computer science field that shares some of the SE objectives. KE is required when the software to be developed has to behave in a heuristic way.

Difference between computer science and software engineering: from the perspectives of research vs implementation. Computer science: how computers work in theory. (Alan Turing and Von Neumann). Software engineering: build applications (Steve Wozniak and Larry Page).

Software categories: by copyrights: commercial, shareware, freeware and public domain; Divided by applications (such as banking and finance).

**Software quality: definitions and strategic issues (check)**

**Computer ethics**

Data types: (1) Time orientation refers to past, present or future requirements of a proposed application. For example, sometimes past data needs to be saved separately with a timestamp; other times past data only needs to be updated or to be overwritten. (2) Structure of information refers to the extent to which the information can be classified in some way. For example, personal data (e.g. names and date of birth) can be grouped together and have only one row; their contact information (e.g. addresses and email addresses) can be grouped together and can have multiple rows. (3) Completeness refers to the extent to which all desired information is present. For example, for personal data, name is required but date of birth may not be required. If required, there must be some validation methods to ensure completeness. (4) Ambiguity is a property of data such that it is vague in meaning or is subject to multiple meanings. For example, if data is blank, it can mean that the data are missing or it can mean it’s a default value. (5) Semantics is the study of development and change in the meaning of words. I think this is the most important of all 6 attributes of data types. How things are defined and named are the major difficulties in reading any code. Especially if abbreviations are used, it does not always convey the meaning that the software engineer have in mind when they build the variable. More comments are usually needed to clarify the meaning of the variable. (6) Volume is the number of business events the system must cope with in the same period. I think this is important when designing and building software that will be expected to be scaled up. Scalability may be inherent to the system design for tradeoffs of other functions such as security.

Data collection techniques: seven techniques for data collections during application development: interviews, group meetings, observation, review software, questionnaire, temporary assignment, review internal documents, review external documents. Each of the techniques has strengths and weaknesses.

Data type and application type

Application types are transaction processing (TPS), query, decision support (DSS), group decision support (GDSS), executive information (EIS), and expert system (ES).

TPS contain predominantly known, current, structured and complete information. I think many accounting and payment software that records transaction data are TPS. DSS are statistical analysis tools that allow development of information that aids the decision process. I think SPSS and SAS is an example of DSS. GDSS are meeting facilitation tools for groups of people. I think Zoom and WebEx belong to the category of GDSS. EIS are future-oriented applications that allow executives to scan the environment and identify trends, economic changes or other industry activity that affect the governance of the people. I think software like Tableau with dashboard or reporting functions are examples of EIS. Expert systems manage and reason through semi-structured, incomplete, ambiguous and variable semantic data. I think expert systems are usually customized software or add-ons that are built for the specific needs of an organization.

Software Development Life Cycle (SDLC): **(1) Planning**: Customer talk with project manager about the requirements of application: User registration, Login, Logout, Dashboard landing page (2) Requirement analysis: software development team discusses the requirements and create tickets for these requirements in the project management system: a) User registration: Username input field, password field, checkbox: accept terms and conditions, submit button, save user in database; **b) Login**: Username input field, password field, submit button, read user from database, log user into the system; c) Logout: Logout button, clear session, prevent account theft; d) Dashboard landing page: Main home page, new users should redirect here, existing users redirect here after login; **3) Design**: This phase may include business rules, user interface layouts, color schemes, program languages to use, frameworks, system server design, database relationships, application architecture, mobile apps, supported browsers, etc; **4) Implementation and coding**: Set up physical hardware, write the code, design user interface, analyze requirements and start building test cases for test plans; **5) Testing**: Test all test cases on the application, verify all the requirements are met, make sure all functionalities working as expected, find as many bugs as possible (bug life cycle); **6) Deployment**: Application becomes live for real users; 7) Maintenance: Monitor the statistics of the server, production support;

Software development methodologies:

|  |  |  |
| --- | --- | --- |
|  |  |  |

**Waterfall model**: sequential approach: one step at a time, move to the next step only if the current step is completed. In other words, the output of each phase feeds into the next phase, while phases are modified via feedback produced during the verification and validation process. **Agile Method**: discover a feature, design, develop and test a small chunk of software in every cycle. This development strategy can save a lot of time because customer’s demand can change very quickly. If customer is not satisfied with any of the functions, developers can tweak them without wasting time developing things that customers do not need. **DevOps**: similar to agile method, the only difference is immediate package and release to the market after developing a feature. I think WebEx uses this approach because updates are released immediately after being developed.

Iterative Project Life Cycle: 1) A cyclic repetition of analysis and design events (sometimes called prototyping or spiral approach). 2) Development in a short period of time without formal written specifications. 3) Use to test proof of utility, availability or appropriateness of hardware, software or design concepts.

Quality Management

1) Value: the degree of importance you give to something. (Good to know!) 2) ISO: International Organization for Standardization 3) ISO 9000: a family of Quality Management System (QMS) standards created by ISO in 1987; comprising ISO 9001, ISO 9002 and ISO 9003 for different types of industries. Business can choose to be certified against one of them based on business processes. ISO is meant to certify the processes and the system of the organization, not the product or service itself. 4) Total Quality Management (TQM): a strategic approach to management aimed at embedding awareness of quality in all organizational processes. 5) Poka-yoke: a methodology of using low-cost techniques to error proof production processes.