

16 Software Support

Can be considered an umbrella activity that includes:

- Change management
- Proactive risk management
- Process management
- Configuration management
- Quality assurance
- Release management: Process that brings high quality code from developer's workspace to the end user includes:
 - Code change integration
 - Continuous integration
 - Build system specifications
 - Infrastructure as code
 - Deployment and release
 - Retirement

Software supportability:

- Capability of supporting software over its whole lifetime
- Implies satisfying all necessary requirements and also the provision of resources, support infrastructure, additional software, and manpower needed to ensure software is capable of performing its functions

Software Maintenance:

Software is released to end-users, and:

- Within days, bug reports filter back to the software engineering organization
- Within weeks, one class of users indicates that the software must be changed so that it can accommodate the special needs of their environment

Reverse engineering - process of analyzing a software system to create representations of the system at a higher level of abstraction.

Refactoring - process of changing a software system to improve its internal structure without altering its external behaviour.

Reengineering (evolution) - process of taking an existing software system and generating a new system that has the same quality as software created using modern software engineering practices.

Agile Maintenance

- Use sprints to organize the maintenance work. Balance the goal of keeping customers happy with technical needs of the developers.
- Allow urgent customer requests to interrupt scheduled maintenance sprints, make time for them during sprint planning.
- Balance the use of written documentation with face to face communication to ensure planning meeting time is used wisely

Software Analytics and Proactive Maintenance:

- Be sure you are using analytics to identify meaningful development problems
- The analytics must make use of application domain knowledge to be useful to developers.
- Make sure the analytics are scalable to larger problems and customizable to incorporate new discoveries made over time.

Data Refactoring:

- Data refactoring should be preceded by source code analysis
- Data analysis requires the evaluation of programming language statements containing data definitions, file descriptions, I/O, and interface descriptions are evaluated.
- Data redesign involves a data record standardization which clarifies data definitions to achieve consistency among data item names or physical record formats, data name rationalization ensures that data naming conventions conform to local standards.

Code Refactoring: Is performed to yield a design that produces the same function but with higher quality than the original program.

Architectural Refactoring: trade off options for dealing with a messy program:

- You can struggle through modification after modification, fighting the ad hoc design and tangled source code to implement the necessary changes.
- You can attempt to understand the broader inner workings of the program to make modifications more effectively.

Software Evolution

Reengineering Process Model

Inventory Analysis

Document Restructuring

Code Restructuring

Data Restructuring

Forward Engineering

