Key Terms

Here are some basic definitions and abbreviations that are useful for studying.

Definitions

Software Engineering - The Application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software.

Milestone - Checkpoints throughout the life of a project. Identify when one or multiple groups of activities have been completed thus implying that a notable point has been reached in the project.

Deliverable - Measurable and tangible outcome of the project. Developed by project team members in alignment with the goals of the project.

Gold-plating - adding shiny new features that aren't necessary

Scope Creep - uncontrolled growth in scope after project begins

Crunch - Extended working hours

Software Entropy - Measure of code complexity

Yak Shaving - Endless series of small tasks before next big milestone

Artefacts - (belong in a museum) byproduct of the development process

Timescale - the time allowed for or taken by a process/task

User Stories - stories describing each requirement - like use cases but lighter

Refactoring - change internal code structure so it is easier to understand and cheaper to change, without modifying behaviour of program

Software Architecture - set of principal design decisions about software systems

Architecture Description - artefacts to express and document architectures.

High Cohesion - degree of intra-class communication

Low Coupling - degree of inter-class communication

Domain Specific Software Architecture (DSSA) - idea that for any given business, there will be existing architectures relevant to that business's domain which can be used to build an architecture tailored for that business

Vertical architecture - layered, each layer uses previous layer, each layer has its own function

Horizontal architecture - multi-tiered, can have aspects of application layer across all layers/tiers

Abbreviations

Contraction	Definition
SDLC	Software Development Life Cycle
UP	Unified Process
PERT	Program Evaluation and Review Technique
RDD	Responsibility Driven Development
GRASP	General Responsibility Assignment Software Patterns