Software and the Nature of Software

Madiba Hudson-Quansah

Contents

Chapter 1

The Nature of Software

1.1 Defining Software

Definition 1.1.1: Software

Software is instructions, that when executed provide desired features, function and performance; data structures that enable the programs to adequately manipulate information in both hard copy and virtual forms that describes the operation and use of the programs.

The characteristics of software:

Software is developed or engineered; it is not manufactured in the classical sense - There may exist similarities between the development and manufacture of software and hardware respectively, like quality through design, but the manufacturing phase for hardware can introduce quality problems that are non-existent in software contexts.

Software doesn't "wear out" - Usually hardware exhibits relatively high failure rates early in it's life, and corrected and the failure rate drops to a steady-state level for some period of time. Software does not behave the same, with decreasing failure rate with spikes when when defects are discovered as changes are made.

Although the industry is moving towards component-based construction, most software continues to be custom built - As the field matures more and more best practices are codified and a set of standard design components are established. This allows the engineer to focus on the truly innovative elements of a design.

1.2 Software Application Domains

System Software - A collection of programs written to service other programs. Characterized by heavy interaction with computer hardware, heavy usage by multiple users, concurrent operations that require scheduling, complex data structures, and multiple external interfaces.

Application Software - Stand-alone programs that solve a specific business need. Process business or technical data in a way that facilities business operations or management/technical decision making.

Engineering / Scientific Software - Characterized by number crunching algorithms, which application areas ranging from astronomy to volcanology, from automotive stress analysis to space shuttle orbital dynamics, and from molecular biology to automated manufacturing.

Embedded software - Resides within a product or system and is used to implement and control features and functions for the end user and the system itself.

Product-line software - Designed to provide a specific capability for use by many different customers.

Web Applications - A set of linked hypertext files that represent information using text and graphics.

Artificial Intelligence software - Makes use of non-numerical algorithms to solve complex problems that re not amenable to computation or straightforward analysis.

Chapter 2

Software Engineering

Definition 2.0.1: Software Engineering

The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is the application of engineering to software.

Software Engineering, like an onion, has layers and with any engineering approach has it's foundation in an organizational commitment to quality.

Process Layer - The foundation of software engineering, enables rational and timely development of computer software by defining a framework that must be established for effective delivery of software engineering technology.

Method Layer - Provide technical descriptions and guides for building software, encompassing a broad array of tasks including communication, requirement analysis, design modelling, program construction, testing and support.

Tools Layer - Provide automated or semi-automated support for the process and the methods.

2.1 The Software Process

Definition 2.1.1: Process

A collection of activities, actions, and tasks that are performed when some work product is to be created