27 August 2019

What is software – 3 things we need to remember about software

1. **Code**
2. **Data**
3. **Documentation**
4. Software is developed or engineered, it is not manufactured in the classical sense
5. Software doesn’t “wear out”
6. Although the industry is moving toward component-based construction, most software continues to be custom-built.

\*If software isn’t changing, then it is dead

Software Applications

1. System Software
2. Application software
3. Engineering/Scientific software
4. Embedded Software
5. Product-line software
6. Web/Mobile applications
7. AI Software (Robotics, neural nets, game playing)

What is software Engineering

1. The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

Umbrella Activities for SE Projects

1. Software project tracking and control
2. Risk management
3. Software quality assurance
4. Technical Reviews
5. Measurement
6. Software configuration management
7. Reusability management
8. Work product preparation and production

Software Engineering Practice

1. Polya suggests:
   1. Understand the problem (Communication and analysis)
   2. Plan a solution (modeling and software design)
   3. Carry out the plan (code generation)
   4. Examine the Results

Software Myths

1. If we get behind schedule, we can add more programmers and catch up
   1. False, “The Mythical Man-Month”
2. Read Myths

29 August 2019

IBM

What makes a good software engineer?

* Sense of individual responsibility
* Acutely aware of the needs of team members and stake holders
* Brutally honest about design flaws and offers constructive criticism
* Resilient under pressure
* Heightened sense of fairness
* Attention to detail
* Pragmatic

Team roles

* Ambassador
* Scout
* Guard
* Sentry
* Coordinator

Effective team attributes

* Sense of purpose
* Sense of involvement
* Sense of trust
* Sense of improvement
* Diversity of team member skill sets

Avoid “Toxicity”

Factors affecting team structure

* Difficulty of the problem
* Size of the resultant programs
* Time that the team will stay together
* Degree to which the problem be modularized
* Required quality and reliability
* Rigidity of the delivery date
* Degree of sociability

12, September 2019

Part 1: Software process

Part 2: Modeling (Analysis and Design)

Part 3: Quality (Testing)

Part 4: Project Management

Chapter 7

Principles that guide practice

1. Planning principles
   1. Understand the project scope
   2. Involve the customer in the planning
   3. Estimate based on what you know
   4. Be Realistic
   5. Define how you intend to ensure quality
   6. Track the plan frequently and make adjustments as needed
2. Modeling Principles
   1. Two types of models can be created
      1. Requirements or analysis model
         1. Represents customer requirements by depicting the software in 3 domains
            1. Information Domain

Entity-Relationship diagram

* + - * 1. Functional Domain

Class Diagrams

Code Modules

* + - * 1. Behavior Domain

State Transition Diagram

* + 1. Design model
       1. Represents Characteristics of the software to help construct it effectively
       2. 3 Pieces
          1. Architecture
          2. User interface
          3. Component Level Design

17 September 2019

19 September 2019