Introduction of Software Engineering

Chapter 5:

Modeling Principles

VŨ THỊ TRÀ

CONTENTS

Principles That Guide Process

1. Be agile
2. Focus on quality at every step
3. Be ready at adapt
4. Build an effective team
5. Establish mechanisms for communication and coordination
6. Manage changes
7. Assset rish
8. Create work products that provide values for others

Principles That Guide Practice

1. Divide and conquer
2. Understand the use of abstraction
3. Strive for consistency
4. Focus on the transfer of information
5. Build software that exhibits effective modularity
6. Look for patterns
7. When possible, represent the problem and its solution from a number of different perspectives
8. Remember that someone will maintain the software

CONTENTS

Communication Principles

1. Listen
2. Prepare before you communication
3. Someone should facilitate the activity
4. Face-to-face communication is best
5. Take notes and document decisions
6. Strive for collaboration
7. Stay focused and modularize your discussion
8. If someone is unclear, draw a picture
9. (a) One you agree to something, move on. (b) If you can't agree to something, move on. (c) If a feature or function is unclear and cannot be clarified at the moment, move on.
10. Negotiation is not a contest or a game. It work best when both parties win.

Planning Principles

1. Understand the scope of the project
2. Involve stakeholder in the planning activity
3. Recognize that planning is iterative
4. Estimate based on what you know
5. Consider risk you define the plan
6. Be realistic
7. Adjust granularity as you define the plan
8. Define how you intend to accommodate change
9. Track the plan frequently and make adjustments as required

Modeling Principles

1. The primary goal of the software team is to build software, not create model
2. Travel light – don't create more models than you need
3. Strive to produce the simplest model that will describe the software
4. Build models in a way that makes them amenable to change
5. Be able to state an explicit purpose for each model that is created
6. Adapt the models you develop to the system at hand
7. Try to build useful models, but forget about building perfect models

Modeling Principles

8. Don’t become dogmatic about the syntax of the model. If it communicates content successfully, representation is secondary
9. If your instincts tell you a model isn’t right even though it seems okay on the paper, you probably have reason to be concerned
10. Get feedback as soon as you can

Requirement Modeling Principles

1. The information domain of a problem must be represented and understood
2. The functions that the software performs must be defined
3. The behavior of the software (as a consequence of external events) must be represented
4. The models that depict information, function, and behavior must be partitioned in a manner that uncovers detail in a layered (or hierarchical) fashion
5. The analysis task should move form essential information toward implementation detail

Design Modeling Principles

1. Design should be traceable to the requirements model
2. Always consider the architecture of the system to be built
3. Design of data is as important as design of processing functions
4. Interfaces (both internal and external) must be designed with care
5. User interface design should be turned to the needs of the end user. However, it should stress easy of use
6. Component-level design should be functionally independent

Design Modeling Principles

7. Components should be loosely coupled to one another and to the external environment
8. Design representatives (models) should be easily understandable
9. The design should be developed iteratively
10. Creation of a design model does not produce an agile approach

Construction Principles

1. Preparing Principles
2. Coding Principles
3. Validation Principles
4. Testing Principles

Testing Principles

1. All test should be traceable to customer requirement
2. Tests should be planned long before testing begins
3. The Pareto principle applies to software testing
4. Testing should begin " in the small"   and progress toward testing "in the large"
5. Exhaustive testing is impossible
6. Apply to each module in the system a testing effort commensurate with its expected fault density
7. Statistic testing techniques should be yield high results
8. Track defects and look for patters in defects uncovered by testing
9. Include test cases that demonstrate software is behavior correctly

(Everret and Meyer, 1999)

Deployment Principles

1. Customer expectation for the software must be managed
2. A complete delivery package should assembled and tested
3. A support regime must be established before the software is delivery
4. Appropriate instructional materials must be provided to end user
5. Buggy software should be fixed first, delivered later