

# Software Engineering Practice

Suradet Jitprapaikulsarn

Derived from Roger S. Pressman, *Software Engineering: A Practitioner's Approach, 6th Edition*, McGraw-Hill, 2005

---

---

---

---

---

---

---

---

## What is Practice?

- Way of doing something
- Concepts, principles, methods, and tools
- Technical considerations and How-to's

---

---

---

---

---

---

---

---

## The Essence of Practice

1. Understanding the problem
2. Devising the plan
3. Executing the plan
4. Checking the results

Derived from George Polya, *How to Solve It*, 2nd ed., Princeton University Press, 1957

---

---

---

---

---

---

---

---

## Understanding the Problem

- What are the unknowns?
- What are the data?
- What is the condition?
- Who involves?
- Can the problem be divided?
- Can the problem be represented graphically?

---

---

---

---

---

---

---

---

## Devising the plan

- Are there any similar problems?
- Have they been solved? How?
- Can subproblems be defined?
- Can we develop a solution?
- Are there any alternative solutions?

---

---

---

---

---

---

---

---

## Executing the plan

- Taking action
- Tracking the result
- Is the plan being followed?
- Can the solution be traced to the model?
- Does the solution conform to the plan?

---

---

---

---

---

---

---

---

## Checking the results

- Can we check the results?
- Can the solution be derived differently?
- Can we reuse the solution?
- Dose the solution satisfied the stakeholders?

---

---

---

---

---

---

---

---

## Seven Principles of Software Development

1. The Reason It All Exists
2. KISS (Keep It Simple, Stupid)
3. Maintain the Vision
4. What You Produce, Others Will Consume
5. Be Open to the Future
6. Plan Ahead for Reuse
7. Think!

---

---

---

---

---

---

---

---

## Communication Practices

1. Listen
2. Be prepared
3. Have facilitator
4. Communicate face-to-face
5. Document
6. Collaborate
7. Focus
8. A picture is worth a thousand words
9. Move on
10. Win-Win is the best

---

---

---

---

---

---

---

---

## Planning Practices

1. Know the boundary
2. Involve the users
3. Planning is iterative
4. Estimate  $\neq$  Guess
5. Don't forget the risks
6. Be realistic
7. Different detail at different time
8. Define the quality assurance
9. Prepare for the changes
10. Measure and adjust

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ศูนย์วิจัยและพัฒนาวิศวกรรมปิโตรเลียม

10

---

---

---

---

---

---

---

---

---

---

## Modeling Practices

- Analysis models
  - Information domain
  - Functional domain
  - Behavior domain
- Design models
  - Architecture
  - User Interface
  - Component

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ศูนย์วิจัยและพัฒนาวิศวกรรมปิโตรเลียม

11

---

---

---

---

---

---

---

---

---

---

## Analysis Modeling Principles

1. Understand the information domain
2. Define the functions
3. Represent the behavior
4. Organize the representation
5. From simple to detail

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ศูนย์วิจัยและพัฒนาวิศวกรรมปิโตรเลียม

12

---

---

---

---

---

---

---

---

---

---

## Design Modeling Principles

1. Compatibility with analysis model
2. Think about architecture
3. Balance the importance of data and functions
4. Watch out the interfaces
5. Tailor the UI to users
6. Functionally independent component design
7. Loosely couple component among others and to external environment
8. Easy to comprehend representation
9. Iterative design
10. Make it simple but not simpler

---

---

---

---

---

---

---

---

## Construction Practice

- Coding Principles
- Testing Principles

---

---

---

---

---

---

---

---

## Coding Principles: Preparation

1. Understand the problem
2. Understand the design, its principles and concepts
3. Select an appropriate developing tool
4. Select a suitable developing environment
5. Create unit tests

---

---

---

---

---

---

---

---

## Coding Principles: Coding

1. Constrain the curiosity
2. Compatibility to the design
3. Create consistent software with conformance to the architecture
4. Strive for simplicity
5. Strive for testability and traceability
6. Follow the coding standard and pick the right name
7. Self-documenting code
8. Code with visual layout

---

---

---

---

---

---

---

---

## Coding Principle: Verification

1. Conduct a code walkthrough
2. Carry out the unit tests
3. Refactor the code

---

---

---

---

---

---

---

---

## Testing Principles

1. Compatibility to specification
2. Planned tests
3. 80:20 principle
4. From small to large
5. Although exhaustive testing is not possible, adequate testing is doable

---

---

---

---

---

---

---

---

## Deployment Practices

1. Manage the user expectations
2. Check the delivery package and environment
3. Establish the support mechanism before delivery
4. Instructional document and training must be provided
5. Fix before send

---

---

---

---

---

---

---

---