Software Engineering in Practice
Software maintenance

Diomidis Spinellis
Department of Management Science and Technology
Athens University of Economics and Business

dds@aueb.gr
http://www.dmst.aueb.gr/dds
@CoolSWEng

2024-02-27

Assignment (Software Maintenance)

- Select a popular open source project and study its maintenance:
  - identify maintenance activities and related techniques,
  - propose appropriate metrics for evaluating the current maintenance of the project,
  - identify requirements for further maintenance and relevant techniques, and
  - classify under the maintenance categories the activities and the techniques that you identified.

Need for maintenance

- Correct faults
- Improve the design
- Implement enhancements
- Adapt program to different:
- hardware
- software
- communication facilities
- New/changed APIs
- Migrate legacy software
- Retire software

Lehman's laws

- Continuous change
- Increasing complexity
- Self-regulation
- Conservation of organizational stability
- Conservation of familiarity
- Continuing growth
- Declining quality

- Feedback system

Characteristics

- Maintaining control over the software's day-to-day functions
- Maintaining control over software modifications
- Perfecting software
- Identifying security threats and fixing vulnerabilities
- Maintaining performance

Categories

- Corrective
- Adaptive
- Perfective
- Preventive

Technical issues

- Software comprehension
- Testing
- Impact analysis:
- Analyze modification requests or problem reports
- Replicate the problem
- Develop and document options for implementing the modification
- Obtain approval for the selected modification
- Organize and perform deployment
- Software maintainability

Management issues

- Alignment with organizational objectives
- Staffing
- Process
- Maintenance team
- Specialization
- Communication
- Egoless programming, collegiate atmosphere
- Reduce dependency on individuals
- Allow for periodic audit checks
- Outsourcing

Software maintenability metrics

- Analyzability (see next slide)

- Changeability
- Stability
- Encapsulation
- Abstraction
- Type checking
- Assertions
- Testability
- Understandability
- Complexity

Analyzability

- Consistency
- Formatting
- Naming
- Code conventions
- Comments
- Length of statements, functions and methods
- Control structure
- Coupling and cohesion
- Locality of dependencies
- Abstraction (e.g., through polymorphism)
- Reviewability

Lifecycle metrics

- Number of change requests
- Average time for impact analysis
- Average time of implementation of change
- Number of pending change requests

The perils of metrics

Googhart's Law

By xkcd: https://xkcd.com/2899/

Activities

- Program understanding
- Transition
- Modification request acceptance/rejection
- Help desk
- Impact analysis
- Service Level Agreement (SLA)

Planning activities

- Organizational level
- Maintenance planning
- Release/version planning
- Individual software change request planning

Release/Version planning

- Date commitment for every request
- Agreement between customer and end-user
- Identify potential conflicts and develop alternatives
- Risk assessment and back-out plan
- Inform all stakeholders

Semantic Versioning

Name versions as MAJOR.MINOR.PATCH, and increment the:

- MAJOR version when you make incompatible API changes,
- MINOR version when you add functionality in a backwards compatible manner, and
- PATCH version when you make backwards compatible bug fixes.

Migration

- Planning
- Documentation
- Requirements
- Tools
- Data upgrade
- Testing
- Support
- Parallel operation
- Training
- Inform users

Preparation for the next lecture (1)

- Study Chapter 6 from SWEBOK v 3.0
- Assignment (Software configuration management):
- Perform the following tasks on a popular open source project:
- Which project elements are under software configuration management control and which are not?
- Show diagrammatically two non-trivial workflows associated with elements that are under configuration management

- Measure and show development process data, obtaining them from a software configuration management system. You may find the Perceval tool useful for this task.
- Identify and categorize the software configuration management tools used.

Additional material

- Ross Anderson on software obsolecence

Preparation for the next lecture (2)

- Video Software Configuration Management

Reminder - Coming soon at SEiP lab

- Hands on Unit Testing
- Tuesday 27/3/2018, 17:00-19:00
- eloi2