Software Engineering in Practice
Software construction

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

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

2023-04-03

Team Presentations

- Evaluate the characteristics of a popular open source project with respect to the construction:
  - Look for construction guidelines and standards. How do they enable the validation of the software's correctness?
  - Identify reusable and reused software components.
  - Look for code quality analysis techniques.
  - Try to improve the system's construction quality. Contribute to a change that you propose.

Software construction

- Software construction fundamentals
- Managing construction
- Practical considerations
- Construction technologies
- Software construction tools

Software construction fundamentals

- Minimizing complexity
- Anticipating change
- Constructing for verification
- Constructing with/for reuse

Standards in construction

- Communication methods
- Programming languages
- Coding standards
- Platforms (e.g. J2EE, .NET, POSIX)
- Tools

Construction languages

- Configuration
- Toolkit (application programming)
- Scripting languages
- Visual (e.g., MatLab)
- Linguistic (e.g., C/C++, Java)
- Formal (e.g., Event-B)

Coding

- Naming conventions
- Use of classes, types, enumerations, variables, named constants
- Error handling
- Security
- Resource usage
- Source code organization
- Code documentation
- Code tuning

Testing

- Unit testing
- Integration testing (it will be covered in a separate session)

Construction for reuse

- (Mainly internal)
- Parameterization
- Conditional compilation
- Encapsulation
- Packaging
- Documentation

Construction with reuse

- Selection of reusable units
- Handling dependencies
- Handling updates
- Legal issues

Construction quality

- Tests
- Assertions

- Inspections
- Static analysis (it will be covered in a separate session)

Construction technologies

- Application programming interface (API) design and use
- Polymorphism
- Parameterization and Generics
- State based (or automata based) programming
- Table-driven programming
- Internationalization and localization
- Grammar-based input processing
- Concurrency primitives
- Middleware (e.g. ESB)
- Performance analysis and tuning
- Test-driven development (TDD)

Software construction tools

- Integrated development environments (IDEs)
- Graphical user interface (GUI) builders
- Unit testing
- Profiling and performance analysis
- Static analysis

Preparation for the next lecture (1)

- Study Chapter 4: “Software Testing” from SWEBOK v 3.0
- Assignment (Software Testing)

  Perform the following tasks on a popular open source project:

  - identify the software testing layers,
  - study which testing techniques are used,
  - present metrics that evaluate the testing coverage for this open source project with respect to basic characteristics of the project (i.e., lines of code, reported issues, number of test cases, etc.), and
  - propose additional testing techniques and tool categories that can be used on this project.

Preparation for the next lecture (2)

- Video (Software Testing) https://www.youtube.com/watch?v=wEhu57pih5w

Distribution License