Software Re-Engineering

Lecture: 16

Dr. Syed Muazzam Ali Shah

Department of Software Engineering

National University of Computer & Emerging Sciences

muazzam.ali@nu.edu.pk

Sequence [Todays Agenda]

Content of Lecture

Restructuring & Refactoring

Restructuring & Refactoring

Refactoring

- Developers continuously modify, enhance, and adapt software to new requirements and execution environments.
- The software evolves with time, and, most likely, it deviates from its intended design.

Restructuring & Refactoring

Refactoring

- As the software further evolves and strays too far away from its original design, three important things happen to the software.
  - **Decreased understandability:** It becomes increasingly difficult to understand the software and it becomes less maintainable.
  - **Decreased reliability:** The reliability of the software decreases. As the software deviates from its original design and as documentations become out-of-date, faults are inadvertently introduced into the software during maintenance.
  - **Increased maintenance cost:** The cost of maintaining the software rises in the absence of preventive measures.

Restructuring & Refactoring

Refactoring

- The difficulty in understanding the software is due to the:
  - Increased complexity of the code during the maintenance phase;
  - Out-of-date documentation;
  - Code not conforming to standards.
- Therefore, there is a need to decrease the complexity of software by improving its internal quality.
- The internal quality of software is improved of restructuring the software.
- If restructuring is performed on an object-oriented software, then it is called refactoring.

Restructuring & Refactoring

Restructuring

- Restructuring means reorganizing software to give it a different look, or structure.
- Source code is restructured to improve some of its non-functional requirements, without modifying its functional requirements.
  - For example, one may restructure source code to improve its readability, extensibility, maintainability, and modularity.
- Software restructuring is informally stated as the modifications of software in order to make it:
  - Easier to understand;
  - Easier to change;
  - Easier to update its internal and external documentations; and
  - Less susceptible to faults when changes are made in the future.

Restructuring & Refactoring

Why Restructuring?

- A higher level goal of software restructuring is to increase the software value as explained in the following:
  - **External software value:** This is the value of the software as perceived by the customers.
    - For example, a software with faults may fail to satisfy the business needs of the customers. Such software may be seen by the customers to have less value.
  - **Internal software value:** This represents the cost saving due to three factors:
    - Maintenance cost saving due to a good structure of the software;
    - Cost saving due to potential reuse of components of a software with good structure;
    - Cost saving due to extended use of a software.

Restructuring & Refactoring

Restructuring activities

- Some simple restructuring activities are as follows:
  - **Pretty printing:** Align code statements so that code becomes easier to understand as logical units.
  - **Meaningful names for variables:** Variable names are chosen to give an indication of programming plans.
  - **One statement per line:** Write one code statement in one line, as opposed to many statements in one line.

Restructuring & Refactoring

Reasons – Software Restructuring

- Developers and managers should be aware of software restructuring for the following reasons.
  - **Better understanding:** By reorganizing software with easily traceable structures, it becomes easier for programmers to understand it.
  - **Keep pace with new structures:** As time passes, new generations of programmers are taught new software structures.
  - **Better reliability:** It is easier to locate and fix bugs in well-structured, well understood software than in poorly understood software.
  - **Longer lifetime:** The lifetimes of software can be increased by making them maintainable by means of improving their structures.
  - **Automated analysis:** Programs with good structures are more amenable to automatic analysis than unstructured programs.

Restructuring & Refactoring

Summarize: characteristics- restructuring and refactoring

- The objective of restructuring and refactoring is to improve the internal and external values of software.
- When a subject program is transformed into a new program, the original program's external behavior is preserved by the new program. In other words, the two programs are functionally identical from the viewpoint of users.
- Restructuring does not normally involve code transformation to implement new requirements. Rather, restructuring can be performed without adding new requirements to the existing system.
- When a subject program is transformed into a new program, the relative level of abstraction is preserved. For example, a program in C is transformed into another C program, rather than a program in an assembly language.

Restructuring & Refactoring

Activities : Refactoring Process

- To restructure a software system, programmers follow a process with well defined activities. Those activities are as follows:
  - **Identify what to refactor.**
  - **Determine which refactorings should be applied.**
  - **Ensure that refactoring preserves the software's behavior.**
  - **Apply the refactorings to the chosen entities.**
  - **Evaluate the impacts of the refactorings.**
  - **Maintain consistency.**


Thank You!