Software Re-Engineering

Lecture: 17

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 and Refactoring

Reverse Engineering – Techniques

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 of software artifcats.**

Reverse Engineering – Techniques

Activities : Refactoring Process

❖ Identify what to refactor.

- In this step, the programmer identifies what to refactor from a set of software artifacts.
  - **Some examples of software artifacts that the programmer can consider are source code, design documents, and requirements documents.**
- Having identified the top level artifact, the programmer can focus on specific portions of the chosen artifact for refactoring.
- Specific modules, functions, classes, methods, and data structures can be identified from the source code for refactoring.
- For programs written in non object-oriented languages, restructuring is generally limited to the level of a function or a block of code.

Reverse Engineering – Techniques

Activities : Refactoring Process

❖ Determine which refactorings should be applied

- In this step, the programmer identifies which refactorings to apply to the portions of the software identified in the aforementioned first step.
- A subset of the entire set of refactorings must be carefully chosen, because of the following reasons.
  - **Some refactorings must be applied together.**
  - **Some refactorings must be applied in certain order.**
  - **Some refactroings can be individually applied, but they must follow an order if applied together.**
  - **Some refactroings are mutually exclusive.**

Reverse Engineering – Techniques

Activities : Refactoring Process

❖ Ensure that Refactoring Preserves the Behavior of the Software

- Ideally, the behavior of a program after refactoring should be the same as the behavior before refactoring.
- Program behavior simply referred to input–output behavior.
  - **In other words, for the same set of input values, the programs before refactoring and after refactoring were desired to produce the same output values.**
- However, in many applications preservation of input–output behavior alone is not enough, because preservation of temporal constraints and non-functional requirements of the program may be key to the success of refactoring.

Reverse Engineering – Techniques

Activities : Refactoring Process

- **Apply the refactorings to the chosen entities**
  - His means executing the steps of the refactroings chosen before.
- **Evaluate the Impacts of the Refactorings on Quality**
  - Both internal qualities and external qualities are impacted by refactorings.
  - Some examples of internal qualities are **size, complexity, coupling, cohesion, and testability**.
  - Similarly, some examples of external qualities are **performance, reusability, maintainability, extensibility, robustness, and scalability.**
  - In general, refactoring techniques are highly specialized, which means that one technique is intended to improve a small number—generally one—of quality attributes of the program.

Reverse Engineering – Techniques

Activities : Refactoring Process

❖ Maintain Consistency of Software Artifacts

- A software system is described by many artifacts at different levels of abstractions.
- Those artifacts include **requirements documents, design documents, source code, and test suites.**
- If one kind of artifact is changed, then it is important to change some or all of the other artifacts so that consistency is maintained across the artifacts.


Thank You!