NAME: Shivam Pawar **UID: 2019230068**

NAME: Vishal Salvi **UID: 2019230069**

CLASS: TE COMPS **BATCH: C**

EXPERIMENT No 2

A Study on the Role of Software Architecture in the Evolution and Quality of Software

Aim: To identify latest software engineering development Research paper (stage 1)

Problem Statement:

Software Architecture serves as a blueprint for a system. It provides an abstraction to manage the system complexity and establish a communication and coordination mechanism among components. Software system's architecture has a significant impact on its evolution. Software architecture enables us to reason about the functionality and properties of a software system without getting involved in low-level source code and implementation details.

Methodology:

The first component is Co-change Extractor, which searches source code repositories and retrieves the groups of files which have been changed together. It identifies the co-changes by going through the developer commits to the SVN repository and extracting the groups of files in the same commit transaction that have been modified together. This component has a modular design, and can be easily extended to support other source code repositories as well.

The second component is Defect Extractor, which parses the commit logs of projects and identifies the software changes which introduced the defects/bugs in the system. Defect Extractor and Co-Change Extractor components are synchronized with each other, to implement an n-months data collection approach, where the co-changes are extracted from the first n-months after a certain release and the introduced bugs are retrieved from the next n-months after the co-changes are retrieved. While Co-change Extractor component obtains the information of co-changes from the source code repository, the Defect Extractor component retrieves the information from the next n-months, and finds which of the original co-changed files has introduced defects in the next n-months' time slice.

To examine the effects of co-change dispersion among the system's architectural modules on defects, we incorporated a third component, Architecture Explorer, which reconstructs the module view of architecture Architecture Explorer component thus utilizes different reverse engineering approaches and obtains several Surrogate Views that approximate the system's architecture. The surrogate views are then used in the last experimental module, Hypothesis Testing, where the effects of software co-change dispersion are examined from an architectural perspective

Conclusion:

As the software architecture plays an important role in the construction and maintenance of software. We believe that the architecture provides an appropriate level of granularity for understanding the root cause of a large class of defects that are due to bad architectural choices and Software Architecture is one of the key factors affecting the quality of a changing software system.

Scope:

If the software architecture is properly developed it will help in the discovery of architectural bad smells, i.e., architectural choices that have detrimental effects on system lifecycle properties.

A good software architecture helps to evolve the system as it allows us to adopt new features such as a different front end, or adding a process rule are easier to achieve, as the software architecture creates a clear separation of concerns.

Well developed architecture enables more reliable assessment of system quality attributes like performance, security, interoperability, reliability, availability.