

An Indian-Australian research partnership



Industry sponsored project by Infosys

## Project title: Architecture centric Software Evolution

**Project number:** IMURA0179

**Monash University supervisors:** Professor Shonali Krishnaswamy, Prof Sita Ramakrishnan

**Monash University contact:** Email: Shonali.Krishnaswamy@infotech.monash.edu.au

**IITB supervisors:** Rushikesh K Joshi,

**IITB contact:** Department of Computer Science and Engineering.; Email: rkj@cse.iitb.ac.in

**Infosys supervisors:** Dr. Srinivas Padmanabhuni, Mr Naveen Kulkarni

---

### Research Academy theme/s

List only the research academy theme/s that is relevant to the project

1. **Advanced computational engineering, simulation and manufacture**
2. Infrastructure engineering
3. Clean energy
4. Water
5. Nanotechnology
6. Biotechnology and stem cell research

### The research problem

The state of global software development, has reached its pinnacle especially in area of software maintenance, and today major Fortune 1000 companies have significant portions of their IT portfolio being maintained by global software corporations. In view of this, innovations and improvements in area of software evolution are crucial to help derive maximal savings for clients, and at the same improve overall time taken to improve average time taken to solve tickets for client software maintenance projects.

There is constant search for techniques and approaches to help enhance overall software maintenance effectiveness and reduce cost.

### Project aims

One recent view emerging is that of applying conformance techniques while doing software refactoring, a key approach for software preventive maintenance, can greatly impact the overall effectiveness of maintenance. While there are several approaches in literature for code refactoring, one of the key issues is the level of freedom given to programmers to do the refactoring. One way to control the refactoring, is to ensure a kind of a gating criteria to provide a filter to the refactoring task. In this direction, we are looking at research and deep dive approaches to involve software architecture and design to guide the refactoring process. The advantage of a gating approach via architecture/design is the potential savings in downstream maintenance costs due to “conformant” code. We are looking at approaches to help bridge this gap of architecture centric software evolution.