Governing Software Process Improvements in Globally Distributed Product Development

Abstract
Continuous software process improvement (SPI) practices have been extensively prescribed to improve performance of software projects. However, SPI implementation mechanisms have received little scholarly attention, especially in the context of distributed software product development. We took an action research approach to study the SPI journey of a large multinational enterprise that adopted a distributed product development strategy. We describe the interventions and action research cycles enacted over a period of five years in collaboration with the firm, which resulted in a custom SPI framework that catered to both the social and technical needs of the firm’s distributed teams. Institutionalizing the process maturity framework got stalled initially because the SPI initiatives were perceived by product line managers as a mechanism for exercising wider controls by the firm’s top management. The implementation mechanism was subsequently altered to co-opt product line managers, which contributed to a wider adoption of the SPI framework. Insights that emerge from our analysis of the firm’s SPI journey pertain to the integration of the technical and social views of software development, preserving process diversity through the use of a multi-tiered, non-blueprint approach to SPI, the linkage between key process areas and project control modes, and the role of SPI in aiding organizational learning.
**Existing System:**

Most of the existing furthermore, the cycle time and defect density of the components developed by the distributed teams was more than two times the existing organizational benchmarks. There was a huge backlog of quality and reliability complaints from customers that were not resolved by the distributed product teams (on average about 20 conformance quality complaints).

**Proposed System:**

The development of the new system contains the following activities, which try to automate the entire process keeping in view of the database integration approach.

1. The system makes the overall project management much easier and flexible.
2. The contract signed clients details are readily available at any moment when the information is necessary.
3. The schedule of the presently working software engineers, Team leaders and project leaders can be known is a flash of a second.
4. The generic schedules and the future availability of the man power calculation can take place in a short time.
5. There is no risk of data mismanagement at any level while the project development is under process.
6. The library search with regard to the previously developed modules will become much easier and faster.
7. The associated relationship with respect to a module, component or element design can be found out very easily and very quickly.

8. The application at peak stages can make the Bug tracking and maintenance to be more specific and more proper.

System Requirements:

Hardware Requirements:

- System: Pentium IV 2.4 GHz.
- Hard Disk: 40 GB.
- Floppy Drive: 1.44 Mb.
- Monitor: 15 VGA Colour.
- Mouse: Logitech.
- Ram: 512 Mb.

Software Requirements:

- Operating system: Windows 7. 32 bit
- Coding Language: C#.net 4.0
- Data Base: SQL Server 2008