C – 4: PIR Lead Process Improvement (PIRL)

Post Implementation Reviews (PIRs , V- 1) are a reliable source of first hand information about the performance of software development working practices. Attendance at a PIR gives insights into working practices difficult to get any other way. The problem is that many of the lessons learned are soon forgotten and improvement suggestions not implemented. At the end of PIRs someone will usually make a comment along the lines of 'that was good – what are you going to do about it?'. PIRL provides a simple mechanism to ensure that something is done about it; that PIRs are followed up and lessons learned are remembered by the organization.

What it is for:	To ensure lessons learned by PIRs are not forgotten and that improvements suggestions are implemented
When to introduce	When PIRs are in use.
When to use	Ongoing – process group periodically reviews lessons learned and project review lessons learned at project startup.
When not to use	When no PIRs are performed.

PIRs can supply an organization with high quality information about its software development working practices. This ranges from best practices through to issues and problems. To enable an organization to exploit this information it is necessary to ensure that the information is not lost or forgotten ands that it is used to direct process improvements.

Prerequisites for PIRL are the routine performance of PIRs, someone in the organization interested in the performance and improvement of software development working practices – a member of a process group, if this exists, and an ability to make improvement to working practices, for example process workshops, TCM or something similar.

PIRs generate reports. These reports will contain descriptions of good and bad points, improvement suggestions. At the end of every PIR a copy of the PIR report should be sent to the process group, or whoever is concerned with improving the organizations working practices. It should be reviewed to see if any unusually urgent or important observations have been recorded. The process group should ensure that all received PIR reports are made accessible to everyone in the organization. Periodically, quarterly is a reasonable period, all recent reports should be reviewed and their content analysed for any patterns or trends

common to all software development activities[1]. If PIRs are routinely performed the PIR agendas and report structures will tend to reflect issues of concern to the organization. Areas that are always useful to consider include requirements, project schedule, internal and external project communications, roles and associated responsibilities and customer satisfaction. The analysis should consider both the number of occurrences of types of issues, as well as their severity. Large numbers of relatively low priority issues may, collectively, have a significant impact on the organizations ability to deliver software.

A Pareto type analysis will usually reveal a dominant issue recurring on most projects. This issue will significantly impact the organization's ability to deliver and support software and is a clear candidate for process improvement.

These patterns, revealed by routine PIRs, offer an insight into the requirements, or business drivers, of the organization's software development capability that are often unstated or tacit. (Ideally the requirements of the organization's development capability should be made explicit by the senior management, but this is rarely the case.) If, for example, scheduling keeps recurring as a common theme in PIRs, and scheduling usually means development timescales are too short, then time is a driver for the organization. Recurring testing or rework problems may indicate quality problems. By presenting patterns and trends in the data this 'bottom up' approach to process improvement can be as effective as policy or model driven process improvement, and has the advantage of being founded upon good data.

The periodic analyse of PIR reports will identify candidate aspects of working practices for improvements. These candidate areas should be considered carefully by those responsible for process improvements, and the most cost effective solutions selected; process improvement resources will always be in short supply and the process improvement itself may be risky o to implement or take too long to deliver benefits.

When areas for improvement have been selected then suitable tools to implement them should be deployed – process workshops for well defined improvements, TCM for work needing some investigation and a larger investment in developing and planning the fix.

In this manner PIRL provides a feedback loop back into development with improvements being made that are directed by the experience of those doing software development.

[1] This presumes some form of categorization of classification of the data. This is an important activity in itself making the data more understandable and manageable – it is impossible to perform even a Pareto analysis without some form of categorization.

There are two other, smaller feedback loops. The first can be considered part of the PIRs themselves. These are the ‘quick fixes’. During the course of PIRs minor issues will be identified and can be resolved immediately by those affected by them, for example including someone on a distribution list, or posting documents on the web for information, making test scripts more widely available. These are fixes of ‘special causes’ where the process improvements initiated by the process group are repairs of ‘common causes’. Such minor changes can be made easily and all help improve ways of working. The second is part of project planning. PIRs reports capture ‘lessons learned’. This feedback loops ensures lessons are remembered. During the early planning of software projects one of the planner’s checklist items should be a review of PIR reports or analyses to ensure that problems of the past are recognized and risks of encountering them again can be mitigated.