Knowledge Infrastructure for Project Management[1]

Pankaj Jalote

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
Kanpur, India – 208016
Jalote@iitk.ac.in

Abstract

In any organization, past experience plays a key role in improvement and management. How effectively past experience can be leveraged depends on how well this experience is captured and organized to enable learning and reuse. Systematically recording data from projects, deriving lessons from it, and then making the lessons available to other projects can enhance this reuse. In this paper we discuss three key approaches for organizing and using past experience and how they are employed in Infosys Technologies Ltd. – a large software house which has been assessed at level 5 of the CMM. First, we discuss the process infrastructure which encapsulates the past experience in the form of processes and supporting templates and checklists. Second, we discuss the process database that contains metrics from past projects. Finally, we discuss the body of knowledge system that is used to record experience of people in problem solving in a variety of areas. We will also briefly discuss how this knowledge infrastructure is used for managing a project.

Introduction

An organization is a cohesive entity that has some mission or defined goals. The organization (or the people in it) performs some tasks to achieve these goals. Knowledge helps perform these tasks better, faster, cheaper. The main goal of knowledge management is to help reduce cost, reduce cycle time, or improve quality through the effective use of knowledge. In an organization which is in the business of software development, as the main assets are the intellectual capital, knowledge management is particularly important [14].

Knowledge can be external, i.e. which is produced by people outside the organization. This type of knowledge resides in books, journals, magazines, etc. Knowledge can also be internal, i.e. the knowledge that is created primarily within the organization, largely through experience and experimentation. Generally, the goal of knowledge management within an organization is to manage the internal knowledge of the organization (creation of which uses external knowledge.) Leveraging experiential knowledge is the focus in the experience factory model [1], and is envisaged at the higher levels of the capability maturity model [15]. In this paper we also focus on the management of internal knowledge, particularly the knowledge that is useful in project management, i.e. use of which can make project management more effective.

Suppose in a software organization, there exists a "super" project manager who consistently executes projects successfully, whose estimates are generally on target, and who seems to avoid the "fire fighting" mode most of the time. Clearly, this project manager has acquired the knowledge to properly perform the various tasks associated with project planning and execution through experience.

Clearly, the organization will want this experience to be available to other project managers so they can also execute projects successfully. One way to achieve this is to have the super project manager available as a "consultant" to other project managers. This approach is not scalable. Having knowledge reside with an individual also has other undesirable side effects. The goal of knowledge management (for project

---

[1] This article is based on some chapters in *CMM in Practice* and in *Software Project Management in Practice*, both by P. Jalote and published by Addison Wesley.

management purposes) is to preserve and leverage experience of individuals, such as this super project manager, for the benefit of all project managers.

Hence the basic objective of knowledge management is to compile and organize internal knowledge such that it resides in systems and is available for use by project managers. Consequently, the key elements of knowledge management are collecting and organizing the knowledge, making it available through some knowledge infrastructure, and then using the knowledge to improve the execution of projects.

The center piece of a knowledge infrastructure for project management is the processes and related process assets. Processes describe how different tasks are to be executed and encapsulate the knowledge the organization has for efficiently performing that task. Process assets are documents that aid in the use of processes. Besides process and process assets, metrics knowledge from past projects is invaluable for new projects – both for planning and project monitoring. Hence, another key element in knowledge infrastructure for project execution is the process database which keeps the summary of the past projects. Process assets and process databases capture the key elements but still leave some things un-captured. Hence a system to capture the rest of knowledge that may be of use is needed. We discuss these three elements of knowledge infrastructure in the rest of this paper. These elements are based on how they are supported at Infosys, a large software house that has been assessed at level 5 of the CMM. Further details on these are give in [8, 10].

Process Specification and Process Assets

A process-oriented approach for project execution forms the foundation, the backbone, of any knowledge management system. Without defined processes for executing different tasks, it is not even possible for a project manager to ask the question "how can I use past experience to perform this task better?" This is because implicit in this question is the existence of some method which the project manager is to use and which he wants to improve! Hence the centerpiece of any knowledge management system for project execution is the processes defined to perform different tasks in a project.

And what is a process? Technically, a process for a task is comprised of a sequence of steps that should be followed to execute that task. For an organization, however, the processes is recommended for use by its engineers and project managers are much more than a sequence of steps—they encapsulate what the engineers and project managers have learned about successfully executing projects. Through the processes, which cover engineering as well as project management tasks, the benefits of experience are conferred to all, including a newcomer in the organization. These processes help managers and engineers emulate past successes and avoid the pitfalls that lead to failures. Hence, processes are the main means of packaging and reusing past knowledge.

For an organization, the standard processes that have to be followed by a project have to be properly specified and documented. Different approaches are possible to precisely and succinctly specify a process. At Infosys, processes are organized in a top-down manner. A process consists of stages or phases, a stage (phase) consists of activities, and each activity could be further broken down into sub-activities. The formal process definition specifies the top three levels only – further details are specified as checklists. The definition for each stage generally follows the ETVX (Entry, Task, Verification, and eXit) model [12], and specifies the following.

1. **Overview** - a brief description of the stage
2. **Participants** - all the participants that take part in executing the various activities in the stage
3. **Entry Criteria** - the pre-requisites that must be satisfied before this stage can be started
4. **Inputs** - all the inputs needed to execute the stage
5. **Activities** - list of all activities (sometimes also important sub-activities) that are performed in this stage

6. **Exit Criteria** - The conditions that the outputs of the stage must satisfy in order to consider the stage as completed
7. **Outputs** - all the outputs of the stage
8. **Measurements** - all the measurements that must be done during the execution of the stage
9. **Special Verification**
10. **References**

With a specification like this for each stage, the dependence between stages is explicitly specified in the form of entry criteria. The order in which the stages are presented in the process definition is merely for documentation convenience. Note that this specification captures past experience not only about the sequence of steps that should be used, but also about entry and exit criteria that should be satisfied, what measurements to take, what outputs should be produced, etc.

At Infosys, various processes are specified. These include processes relating to both the engineering tasks of the project, as well as the tasks related to project management. For example, for a development project, the recommended life cycle process is specified as the development process. There are other processes for different types of projects, for example the reengineering process and the maintenance process. The project management process, which covers project planning as well as project monitoring and closure, is the main management related process. There are other supporting processes like the configuration management and review process.

A process specification encapsulates an organization's experience in form of "successful recipes". Process descriptions, however, are generally succinct and do not give detailed steps on how to execute different tasks or how to document their outputs. In order to facilitate the use of processes on projects, guidelines, checklists, and templates usually provide useful support. These together are called *process assets*, and are generally present in many high maturity organizations [9].

Guidelines usually give rules and procedures for executing some step in the process. For example, a step in project planning process is "Estimate effort". But to actually execute this step, a project manager will need some guidelines. Checklists are usually of two types – activity checklists and review checklists. An activity checklist is, as the name suggests, a list of activities that should be done while performing a process step. The purpose of review checklists is to draw the attention of reviewers to the defects that are likely to be found in an output. Templates essentially provide the structure of the document in which the output of a process or a step can be captured. The relationship between process and these assets is shown in Figure 1.

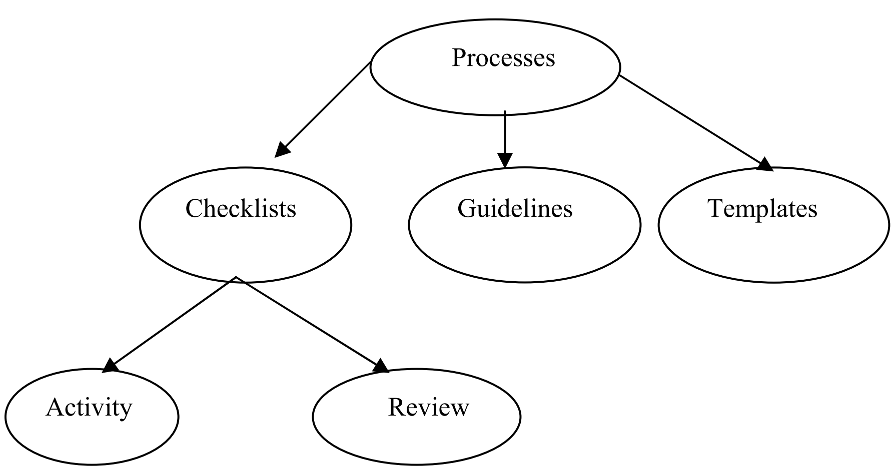

Figure 1: Process and Process assets

The main purpose of these process assets, which capture specific aspects of organizational knowledge, is to facilitate the use of processes and to save effort. For example, creating a document with a template can be so much easier and less time consuming than creating it from scratch. These assets also help improve the quality by minimizing the number of defects injected by providing proper guidelines and activity checklists, and by catching the injected defects early by aiding reviews. It should be clear that to derive full benefits from a process-oriented approach for project execution, process assets are extremely important. At Infosys, all guidelines, checklists, and templates are available on-line and are regularly updated. A sample of some of the process assets that are used in project management are shown in Table 1.

Table 1: Process assets for project management

Guidelines	Checklists	Templates/Forms
• Effort and schedule estimation guidelines • Group review procedure • Process tailoring guidelines • Defect estimation and monitoring guidelines • Guidelines for measurements and data analysis • Risk management guidelines • Guidelines for requirement traceability • Defect prevention guidelines	• Requirements analysis checklist • Unit test and system test plan checklists • Configuration management checklist • Status report checklist • Requirement review checklist • Functional design review checklist • Project plan review checklist • Code review checklist for C++	• Requirements specification document • Unit test plan document • Acceptance test plan document • Project management plan • Configuration management plan • Metrics analysis report • Milestone status report • Defect prevention analysis report

In the context of knowledge management, the guidelines for process tailoring deserves special menion. Any defined process will not apply to all situations and all projects. Tailoring is the process of adjusting a previously defined process of the organization to obtain a process that is suitable for the particular business or technical needs of a project. To allow proper tailoring of previously defined processes, tailoring guidelines are provided. These guidelines define under what conditions which type of changes should be done to a standard process. In essence, they define a set of "permitted deviations" of the standard process to suit the needs of a project. These guidelines are themselves based on experience and encapsulate the past experience of project managers regarding how to tailor the process under different circumstances. Figure 2 illustrates the role of tailoring guidelines for a project. More information on the tailoring guidelines at Infosys is given in [8, 10].

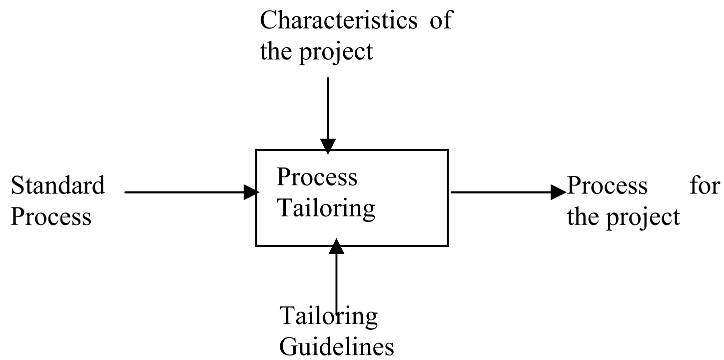

Fig 2: Process tailoring

In addition to these generic assets, if a project manager finds that a past project was similar in some respects, he may want to use some of its outputs. Reusing artifacts can save effort and increase productivity. To promote this goal, process assets from projects may also be collected when the projects terminate. The assets that are typically collected, and made available through a separate system, include project management plan, configuration management plan, schedules, standards, checklists, guidelines, templates, developed tools, training material, and other documents that could be used by future projects

Process Database

The process database (PDB) is a repository of process performance data from projects, which can be used for project planning, estimation, analysis of productivity and quality, and other purpose [7]. The PDB consists of data from completed projects and forms the quantitative knowledge about experience in project execution. As can be imagined, to populate the PDB, data is collected in projects, analyzed, and then organized for entry into the PDB [6]. Many high maturity organizations have some form of process database [9]. Here we discuss what the PDB at Infosys contains. We do not discuss how measurements are done in projects and refer the reader to [8, 10] for more information.

Overall, the data captured in the PDB at Infosys can be classified into the following categories:

- Project characteristics
- Project schedule
- Project effort
- Size
- Defects

Data on project characteristics consists of the project name, the names of the project manager and module leaders (so they can be contacted for further information or clarifications about the project), the business unit to which the project belongs (to permit business-unit-wise analysis), the process being deployed in the project (which allows analysis for different processes to be done separately), the application domain, the hardware platform, the languages used, the DBMS used, a brief statement of the project goals, information about project risks, the duration of the project, and team size.

The data on schedule is primarily the expected start and end dates for the project, and the actual start and end dates. The data on project effort includes data on the initial estimated effort and the total actual effort, and the distribution of the actual effort among different stages e.g. project initiation, requirements management, design, build, unit testing, and other phases. This data is useful in estimating the effort or the schedule of a new project.

The size of the software developed may be in terms of LOC, the number of simple, medium, or complex programs, or a combination of these. Even if function points are not used for estimation, a uniform metric for productivity may be obtained by representing the final size in function points. The final size in function points is usually obtained by converting the measured size of the software in LOC to function points, using published conversion tables. Size data is always required for comparison purposes and building models.

The data on defects includes the number of defects found in different defect detection activities, and the number of defects injected in different stages. Hence, the number of defects of different origins found in requirements review, design review, code review, unit testing, and other phases is recorded. (The detailed data on reviews, however, is kept in a separate review database, which is used for analyzing the review process and setting suitable guidelines for controlling reviews [8].) Defect data can be used for quality planning for a project and evaluating the effectiveness of the various quality activities, when performed in the project.

This information in PDB allows a project manager to obtain data on "similar" past projects – information that is most often sought by project managers when planning a new project. With this type of information, a project manager can search and find information on all projects that focused on a particular business application, used a particular database management system, operating system, hardware, language, etc., was of certain size or duration, and so on. The screen that can be used by project managers to generate this report is shown in Figure 3.

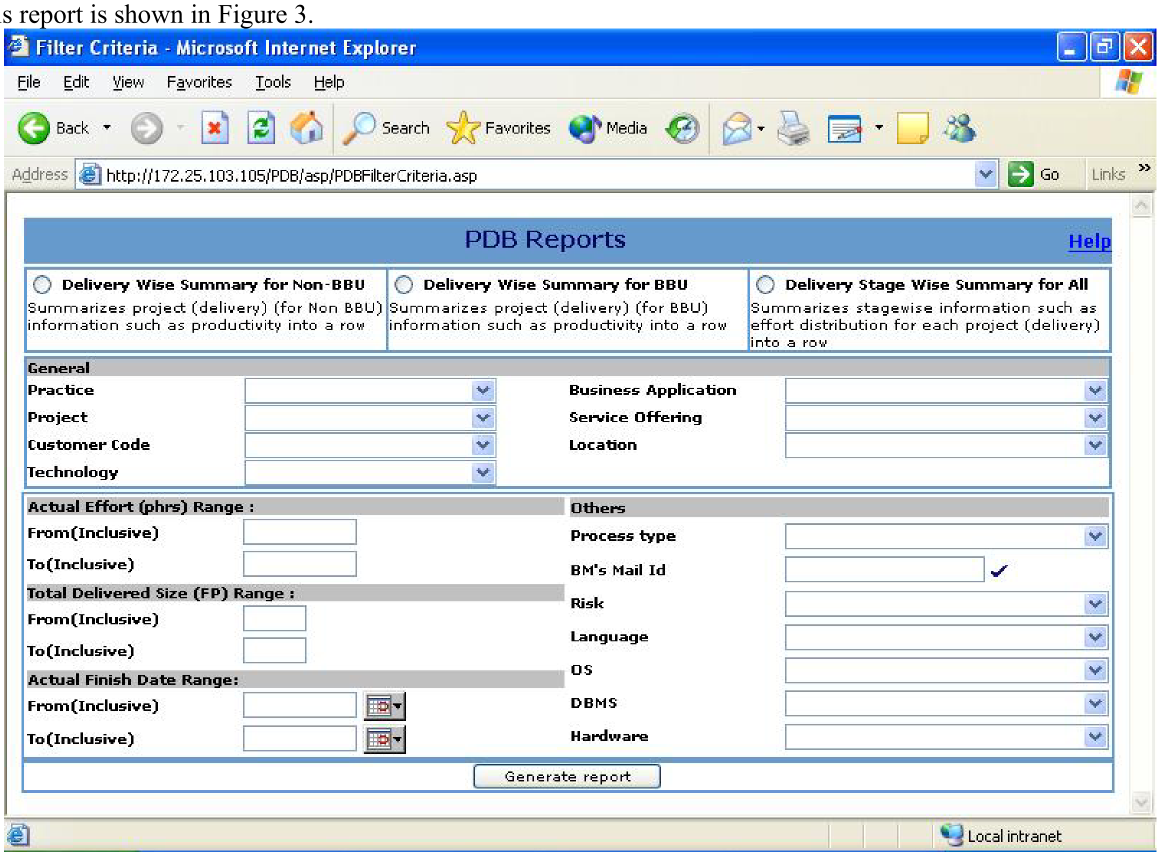

Figure 3: Screen for generating reports from process database

How is the data for the PDB obtained from projects? As part of the standard process for executing projects, project personnel are required to enter data on effort, defects, and schedule. There are tools for each of these. For schedule management, most project use the Microsoft Project in which all the tasks and milestones are enumerated, along with their dates and the resources of each task. For effort collection, an in-house tool called the Weekly Activity Report (WAR) system is used. The WAR system for a person shows all the tasks assigned to him/her, and requires the person to enter the hours spent on the different tasks every day. To ensure consistency in usage, codes are specified for various activities. Every week, each person has to submit his/her WAR, which is then used for analysis. For defect tracking and analysis, a

PC-based commercial tool called Defect Control System was earlier used, but has now been replaced by a web-based in-house tool. For defects, pre-defined categories exist for severity, type, stage of injection, stage of detection, etc. More details about data collection is given in [8, 10].

At the end of the project, the raw data on effort, schedule, defects, and size is analyzed and summarized during a postmortem analysis [2, 3, 4], whose objective is to derive lessons from the project based on what worked and what did not. That is, the purpose of having an identified completion analysis activity, rather than just saying, "the project is done," is clearly not to help this project, but to help future projects by leveraging the "lessons learned" in this project. This type of learning can be effectively supported by analysis of data from completed projects. At Infosys, it is done as part of project closure analysis. After closure analysis, the results are packaged in a manner such that they can be used by others through the PDB (packaging is an important step in knowledge management and is also a key step in the quality improvement paradigm [1]). Full examples of closure analysis reports are given in [8, 10]. The relationship between projects and PDB is shown in Figure 4.

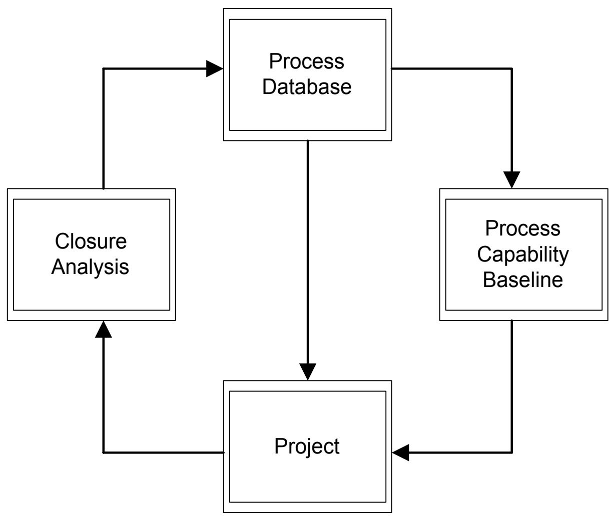

Figure 4 Process database and closure analysis

The data from PDB can also be used to understand the capability of the process in quantitative terms. The capability of a process is essentially the range of expected outcomes that can usually be expected if the process is followed. In other words, if a project follows a process, the process capability can be used to determine the range of possible outcomes that the project can expect. A process capability baseline (PCB) is the snapshot of the capability of the process. At Infosys, the PCB specifies the capability of the process for parameters like delivered quality, productivity, effort distribution, defect injection rate, defect removal efficiency, defect distribution, etc.

PCB is essentially a summary of knowledge about process capability. This knowledge can be used in various ways in project management. For example, productivity data can be used to estimate the effort for the project from the estimated size, and distribution of effort can be used to estimate the effort for the various phases of the project and for making staffing plans. Similarly, defect injection rate can be used to estimate the total number of defects a project is expected to have, and the distribution of defects can be used to estimate the defect levels for different defect detection activities. Overall defect removal efficiency or quality can be used for estimating the number of defects that may be expected after the software is delivered and can be used to plan for maintenance. The PCB also plays an important role in overall process management within the organization.

Body of Knowledge

Though the processes and process assets capture experience related to how different tasks should be done, they still leave information that cannot be generalized or "processized". For example, specific information about how to use a particular tool, how to "get around" some problem in a new compiler, how to tune an application, etc. It is hard to put a process-assets like framework for such knowledge. To capture this type of un-structured knowledge, some other mechanism is needed. At Infosys, another system called the Body of Knowledge (BOK) is used to encapsulate experience. Now this system has been enhanced with other knowledge management initiatives into a K-Shop [13].

The knowledge in BOK, which is primarily in the form of articles, is organized by different topics. Some of the topics are requirement specification, tools, build, methodologies/techniques, design, testing, quality assurance, productivity, and project management. In the BOK system, articles relating to "lessons learned" and "best practices" are posted. Tutorials and articles on trends are also available. The BOK system is Web-based, with its own keyword or author based search facility. The top-level screen of the K-shop is shown in Figure 5.

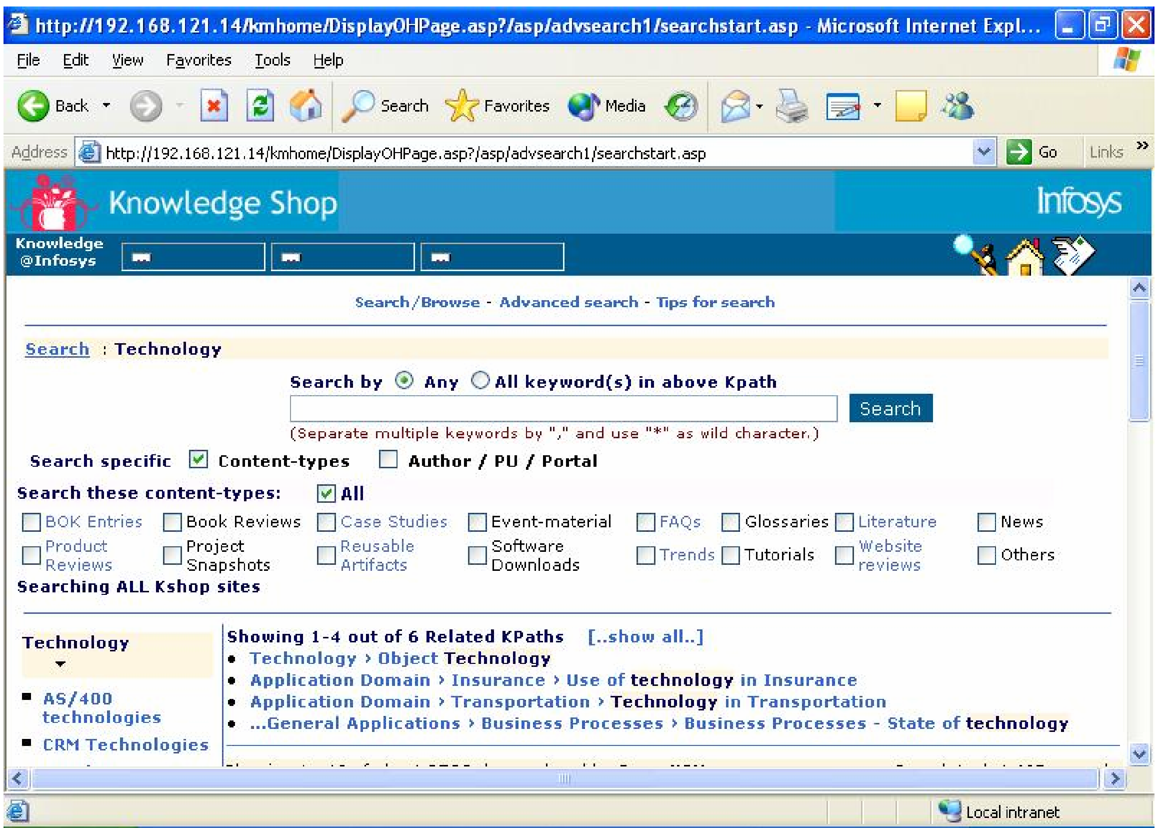

Figure 5: Search screen for the K-shop

Any member of the organization can submit an entry for inclusion in BOK. A template for submitting a BOK entry has been provided. Each submission undergoes a review, which focuses on usefulness, generality, changes required, etc. and an editorial control is maintained to ensure the quality of the entries. Financial incentives have been provided for employees to submit to BOK, and the department that manages BOK actively pursues people to submit. To further the cause, submission to BOK is also one of the factors that is considered during the yearly performance appraisal. A quarterly target for BOK entries is set for the organization.

In addition to the K-shop, there is also a People Knowledge Map [13]. This system keeps the skills and expertise of the different people in the organization. Using this, a project manager can quickly identify and contact specialists. This is what is referred to as competence management in [14]. Such a system helps speed up learning through informal knowledge transfer between people. It also helps the organization identify areas where it is lacking the necessary skills and where expertise needs to be built.

Use of Knowledge Infrastructure in Projects

We briefly discuss how these elements of the knowledge infrastructure are used in projects. The user is referred to [8, 10] for more details. The main use of this knowledge is in project planning and project monitoring. During project planning, a number of tasks need to be performed for which some elements of this knowledge infrastructure are used.

For planning the process that should be used in the project, and for performing the different tasks in the project, processes and process assets are used extensively. All projects use one of the defined standard processes, and use the defined guidelines for tailoring the process. During the execution of the project, guidelines, checklists, and templates are used heavily for most of the tasks. And during reviews, review checklists are used. It is fair to say that for most of the major tasks, projects rely heavily on past experience in the form of checklist, process, and templates.

During project planning, once the process planning is done, some of the key tasks are effort and schedule estimation, risk management planning, and quality planning. In Infosys, for many of these tasks there are guidelines which specify how these tasks are to be done. These guidelines are such that they explicitly make use of past data from process database or the PCB.

We can get a sense of the usage of these elements from their access figures. The overall quality system documentation, which contains all the main documents, is accessed about 4000 times each month. Within this documentation, many of the templates like the ones for CM plan, unit testing plan, project management plan are accessed about 100 to 200 times each month, the common processes (like the CM process, project management process, development process) are accessed about 100 to 200 times each month, various estimation guidelines are accessed about 150 times each month, metrics analysis spreadsheet is accessed about 150 times per month, a typical programming language standard is accessed about 100 time a month, and a typical checklist is accessed about 50 to 100 times a month. The PCB, which summarizes the data in the PDB and which is used heavily for estimation, is accessed about 1,500 times each month.

The metrics infrastructure is also used for project monitoring. As discussed earlier, metrics data on size, effort, schedule, and defects is collected regularly in projects. These data are regularly analyzed to evaluate the health of the project. These evaluations include actual vs. estimated analysis for effort and schedule. For some quality activities also where defect predictions are made, similar analysis is done. Earlier, these analysis were done through reports that were generated at milestones. Now an integrated project management (IPM) system has been developed which integrates all the tools for collecting different metrics. Based on the data, it generates project health report on demand. Hence, the health of the project can be checked at any time. The IPM system also shows the audit and review reports. The top-level screen of the IPM system is shown in Figure 6.

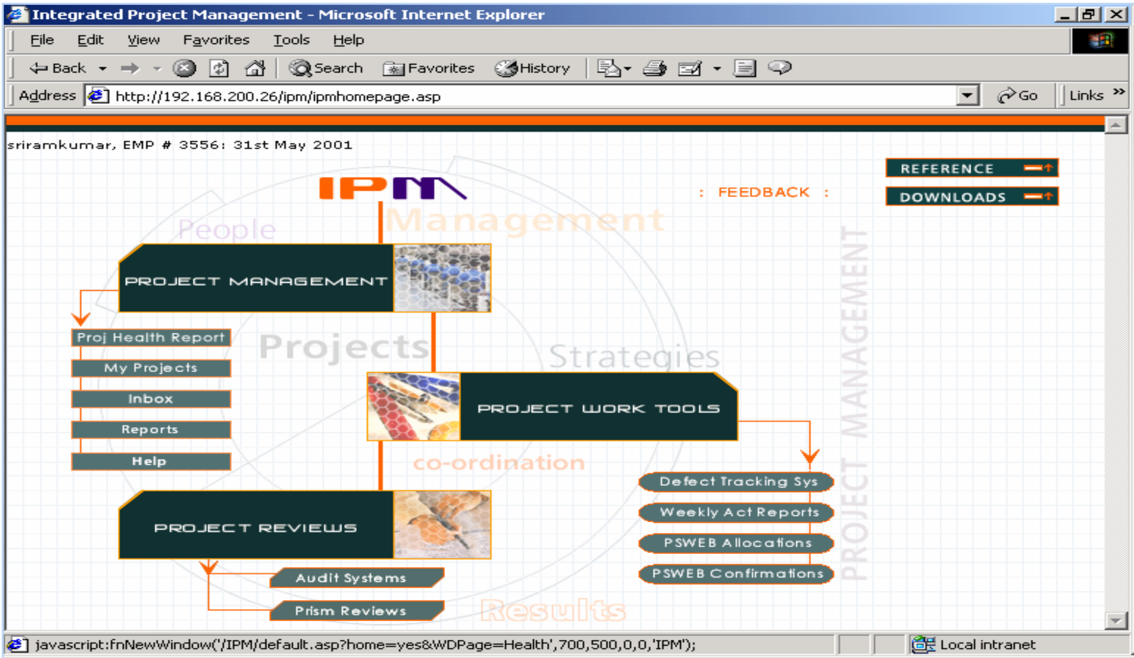

Figure 6: Integrated project management system

Metrics from the project throw up some numbers about the various parameters being measured. But interpretation of these numbers require past experience. For example, when is the measured attribute "too large" or "too small" to deserve management intervention requires past experience. At Infosys, guidelines have been provided for evaluating various parameters. For example, thresholds have been set based on past experience on how much deviation of actual from planned is acceptable [8].

Similarly, when using statistical process control for controlling the process, control limits need to be set. Setting of these limits require past data on the process execution. (We refer the reader to [5, 11, 16] for concepts related to statistical process control and control charts.) At Infosys, analysis of data in PDB and review database is used for setting these limits. Further details on this can be found in [8].

What is the cost of all this knowledge infrastructure? At Infosys, most of this infrastructure is managed by the software engineering process group (SEPG), whose strength is about 0.5% of the engineering staff strength (but the SEPG also does other tasks besides managing this infrastructure.) In addition, task forces from across the organization are formed for a limited period for special initiatives needed from time to time. This overhead, clearly, is quite small for an organization. However, there is also a small overhead in using these systems, entering the data, doing data analysis, and so on. This cost is hard to measure. However, we suspect that it is no more than an hour or so per person per week.

Regarding the benefits, we have already mentioned the access data earlier. The benefit in using this information is also hard to quantify. However, the heavy usage seems to suggest that project managers do find it useful. The senior management also swear by it – without these systems they feel that the large number of projects being executed at Infosys cannot be kept in tight control.

Summary

The main purpose of knowledge infrastructure for project management is to leverage past experience of the organization to improve the execution of new projects. To achieve this objective, the knowledge infrastructure have to compile and organize internal knowledge such that it resides in systems and is available for use by project managers. Consequently, the key elements of building knowledge infrastructure are collecting and organizing the knowledge, making it available through systems, and reusing it to improve the execution of projects. In this chapter we have discussed three main knowledge management systems that are used at Infosys – processes and process assets, process database, and body of knowledge.

The center piece of a knowledge management set up for project management is the processes and related process assets. Processes describe how different tasks are to be executed and encapsulate the knowledge the organization has for efficiently performing that task. Process assets are guidelines, checklists, and templates to support the use of the processes. With processes and process assets, past experience can be effectively used by a new project as help is available on how to execute a task, how to review it, how to document the output, etc.

Besides process and process assets, metrics knowledge from past projects is invaluable for new projects – both for planning and project monitoring. Hence, another key element in knowledge management for project execution is the process database which keeps the summary of the past projects. The process database is a repository of process performance data from projects, which can be used for project planning, estimation, analysis of productivity and quality, and other purpose. The required metrics from a project are captured through various tools as the project proceeds. The captured data is also used for monitoring a project through the integrated project management tool.

Process assets and process database capture the key elements but still leave some things un-captured. Hence a system to capture the rest of knowledge that may be of use is needed. The Body of Knowledge system is where such knowledge is preserved. It is a web-based system with its own keyword or author based search facility. The knowledge in BOK is primarily in the form of articles describing best practices, lessons learned, "how to", etc. on a range of topics.

In the end, it is worth pointing out something that perhaps is quite obvious. The knowledge captured in various systems is dynamic and keeps changing. In the software world, the change is even more rapid. Hence, maintaining the knowledge and keeping it current by enhancing it, adding more useful knowledge and removing information that is not of use is a task that has to be undertaken within the organization. In other words, knowledge management is not free. However, the gains from the use of knowledge captured in these systems should pay many times more than the cost of setting and maintaining these systems.

References

1. V. R. Basili and H. D. Rombach, The experience factory, in The Encyclopedia of Software Engineering, John-Wiley and Sons, 1994.
2. Birk, A.; Dingsoyr, T.; Stalhane, T. Postmortem: never leave a project without it, IEEE Software, Vol 19, No 3, May/June 2002, Pp: 43 -45.
3. E. J. Chikofsky, Changing your endgame strategy, IEEE Software, Nov. 1990, pp. 87, 112.
4. B. Collier, T. DeMarco, and P. Fearey, A defined process for project postmortem review, IEEE Software, pp. 65-72, July 96.
5. W. A. Florac and A. D. Carleton. *Measuring the Software Process – Statistical Process Control for Software Process Improvement*, Addison Wesley, SEI Series on Software Engineering, 1999.
6. R. Grady and D. Caswell, *Software Metrics: Establishing a Company-Wide Program*, Prentice Hall, 1987.
7. W. Humphrey, *Managing the software process*, Addison-Wesley, 1989.
8. P. Jalote, *CMM in Practice – Processes for Executing Projects at Infosys*, Addison-Wesley (SEI Series on Software Engineering), 1999.
9. P. Jalote, Use of metrics in high maturity organizations, SEPG 2K, Seattle, March 2000.
10. P. Jalote, *Software Project Management in Practice*, Addison-Wesley, 2002.
11. D. C. Montgomery. *Introduction to Statistical Quality Control*, third edition. John Wiley and Sons, 1996.
12. R. A. Radice, et. Al. "A programming process architecture", IBM Systems Journal, vol 24, no. 2, 1985.
13. S. Ramasubramanian and G. Jagadeesan, "Knowledge management at Infosys", IEEE Software, Vol 19, no 3, May/June 2002, 53-55.
14. Rus and M. Lindvall, Knowledge management in software engineering, IEEE Software, Vol 19, No 3, May/June 2002, pp 26-38.
15. Software Engineering Institute, *The Capability Maturity Model – Guidelines for Improving the Software Process,* Addison Wesley, 1995.
16. D. J. Wheeler and D. S. Chambers. *Understanding Statistical Process Control*, second edition. SPS Press, 1992.