A programming process study

by R. A. Radice
J. T. Harding
P. E. Munnis
R. W. Phillips

A programming Site Study group was convened to look at the work of eight large-system programming development locations within IBM and to evaluate them according to a set of process stages. Eleven attributes were applied to each process stage. The process of the Site Studies is directly transferable to software evaluations on any project in the software industry, and it is believed that the studies are the first step necessary in the evolution of a consistently repeatable and dynamically controllable process of improvement within the industry. The phases of these studies and implementation of the studies are described.

The nature of the software industry has evolved to a homogeneous state in which software development projects follow a basic life cycle definition.[1] This is generally true throughout the industry, and it is specifically true for the software project environments which this paper addresses.

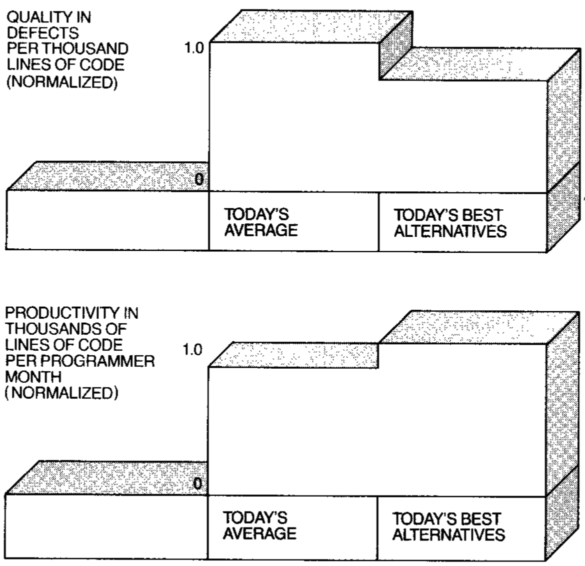
At the same time, it is evident that not all programming projects take full or consistent advantage of the best proven alternatives in tools, methodologies, processes, or practices for developing software. If each project were to exploit these proven and existing alternatives fully, the aggregate quality and productivity would improve to a higher plateau from which still further levels of improvement could be made. As an example, Figure 1 shows the effect on quality and productivity normalized across eight large-system, i.e., Operating System/370 (OS/370), programming sites in IBM.

A Site Study team composed of experts reviews the work of these programming sites to understand two facets of the business: (1) what is actually being done in the sites by the programmers, and (2) what best proven alternatives exist in the sites but are not yet widely practiced throughout the IBM software community. This information is obtained from the sites by interviewing both the programmers and the managers to determine what is actually being done in the development of programming projects versus what is defined as being done or thought to be done. The Site Study team looks at the work of the eight programming laboratory sites to (1) understand how the software is being produced, (2) propagate better alternatives across the sites, and (3) help the sites in the evolution of a consistently repeatable discipline for developing software.

It is essential that the work on the study with each of the sites, and with key representative projects within the sites, be held in absolute confidence and not be presented to anyone outside the project or the site. This confidentiality provides the entree for the Site Study team and allows the uninhibited flow of information from the programmers and from management of the projects. The only information that is communicated outside the project or site are examples of what is the best available in software development today.

From at least one perspective, the Site Study team can be compared to an industrial anthropology group going into software production sites to understand the culture in which software is produced, to

Figure 1 Normalized effect on quality and productivity



objectively evaluate how that software is being produced, and to serve as a conduit for knowledge transfer to accelerate the evolution of the existing programming cultures.

Much as the pioneers Frederick W. Taylor[2] and the Gilbreths went into the engineering communities and manufacturing facilities to better understand how people did their work in the early days of industrial engineering, the Site Study team goes into the IBM programming sites to better understand the work of the programmers, their successes, and the factors inhibiting production of better software. Thus, the team acts as a mirror for information and concerns, reflecting them back to the sites for their use in changing how software is produced. As software engineering consultants to the site, the study team serves as a catalyst for change.

Historical context

What is new about defining a process in the software industry? Certainly this has been done countless times within the industry. We have only to look at the recent work of the U.S. Department of Defense in defining its software engineering environment objectives, and the work of numerous others who have defined how one might produce software from a waterfall model, a serial model, etc.[3–5] An essential difference with the process definition used by the Site Study team is that the working process model was already available. It was not dependent on some innovation that had not yet been proved. The working process definition was built upon the best proven and existing alternatives which were embodied in tools, methodologies, process definitions themselves, and practices presently in use in the software industry.

After the initial model was defined, it was updated as new best available examples became evident. As the team found a new capability that had been proven demonstrably with data to be superior to other existing alternatives, it was added to the working process definition. This updated process definition was then used as the comparative while the next Site Study was performed.

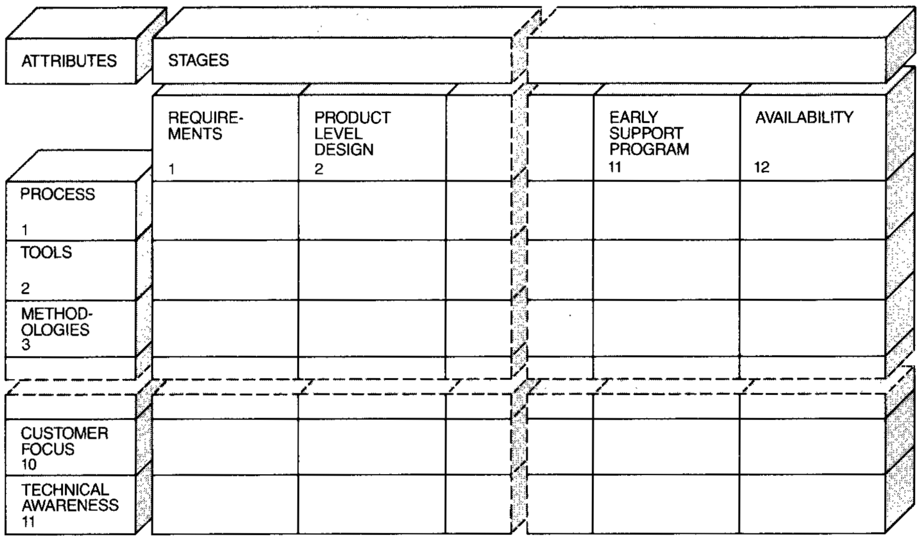
The process definition with which the Site Study team initially went into the sites became a primary input for the Programming Process Architecture defined in another paper in this issue.[6] The working process definition used by the study team was never intended to be accepted or interpreted as a final best-defined or ideal process. Rather, it was treated as a definition of an existing and working process approach which could lead to improved quality and productivity across all products. Because no one product was found to be using all of the tools, methodologies, processes, or practices, the initial process definition became a structure to effect change at the programming sites.

The study implementation

The study always occurred at the site of a product being developed. The Site Study team talked to two groups of people on the project: first, to the management of the project, to obtain a perspective on what was perceived as being done and to understand the rationale for the approaches taken, and then to the individual programmers on the project teams who were doing the work in the different process stages. The project managers did not attend the individual sessions with the programmers.

The study findings were broken up into a matrix of cells (Figure 2) defined by each of the stages for producing the software and a set of common attributes explained later in this paper. This dissection into cells provided a mechanism through which critical items could be immediately targeted for specific focus. It also provided a quantification of the software process which could then be propagated across

Figure 2 Matrix of software cells



the sites as new examples of the best available alternative in software development might become evident.

Findings

This paper will not address the specific ratings given to each of the sites or products, nor will it address examples of the best available alternatives; rather, it speaks to the evaluation process of the studies. The results of these studies have been factored into the process and tools architectures defined in other papers in this issue.

The Site Studies are now occurring on a yearly basis at each of the laboratories and offer one of the best methods for continuously improving the quality, productivity, and schedules of software production. The Site Studies create a continuous focus for changing the culture of the software development process.

Although these studies were instituted within IBM, the methodology is transferable to the evaluation of a software development process on any software project. It is further thought that similar studies are the first step necessary to have a consistently repeatable and dynamically controlled process evolve in the software industry.

Terms and concepts

Before proceeding, we should explore the major concepts and terms that are used in this paper and that formed the essence of the Site Study coverage.

Process stages. Twelve process stages are studied for each product in one-to-one correspondence with the stages of the life cycle as practiced in IBM software production and in the software industry in general. These stages, listed below, are explained fully in the paper on the Programming Process Architecture[6] in this issue.

1. Requirements
2. Product Level Design
3. Component Level Design
4. Module Level Design
5. Code
6. Unit Test
7. Functional Verification Test
8. Product Verification Test
9. System Verification Test
10. Package and Release
11. Early Support Program
12. General Availability

In addition to these 12 stages, the Site Study team evaluates the following functions: (a) performance, (b) build and integration, and (c) publications. It is important to note that although these stages are listed serially, in practice a number of them have a large degree of parallelism. However, for any given function in one module or group of modules, the process stages have a defined or inherent serial definition associated with the function.

We now need to be able to view any given process stage of product development against a set of attri-

Figure 3 Process grid

	TRADITIONAL 5	AWARENESS 4	KNOWLEDGE 3	SKILL & WISDOM 2	INTEGRATED MANAGEMENT SYSTEM 1
PROCESS	NOT DEFINED OR USED	DEFINED BUT INCONSISTENTLY USED	DEFINED BUT STATIC	DEFINED AND IMPROVING	LEADING EDGE AND INTEGRATED INTO BUSINESS
METHODOLOGIES	NOT AWARE OF STATE OF THE ART	AWARE BUT NO USE	AWARE BUT CASUAL USE	MORE USE THAN NOT	FULLY EXPLOITS STATE OF THE ART
ADHERENCE TO PRACTICES	NONE TO LITTLE	SOME BUT NOT CONSISTENT	MORE CONSISTENT THAN NOT	CONSISTENT	CONSISTENT AND TOTAL ACROSS PROCESS
TOOLS	NOT AWARE OF STATE OF THE ART	AWARE BUT NO USE	AWARE BUT CASUAL USE	MORE USE THAN NOT	FULLY EXPLOITS STATE OF THE ART
CHANGE CONTROL	NONE TO LITTLE	INCONSISTENT; LACKS ENFORCEMENT	FOLLOWS A PROCESS BUT DIFFERENT FROM MAINLINE	TENDS TO FOLLOW PROCESS, BUT IS COMPROMISED	FOLLOWS MAINLINE PROCESS, IS NOT COMPROMISED
DATA GATHERING	NONE TO MINIMAL; PRIMITIVE DATA BASES	INCONSISTENT; NON-STATE-OF-THE-ART DATA BASES	ACTIVE ACROSS PARTS OF PROCESS; BEGINNING STATE-OF-THE-ART DATA BASE USE; MORE TRUE THAN NOT	ACTIVE ACROSS PROCESS; STATE-OF-THE-ART DATA BASES	COMPLETELY INTEGRATED AND EVOLVING ACROSS BUSINESS
COMMUNICATION AND USE OF DATA	NONE TO LITTLE	SOME BUT NOT CONSISTENT	CONSISTENTLY USING FEEDBACK	ACTIVELY PURSUING IMPROVEMENT THROUGH INFORMATION FLOW	FULLY INTEGRATED INFORMATION FLOW AS A WAY OF DOING BUSINESS
GOAL SETTING	NONE TO LITTLE	BEGINNING; NO FEEDBACK	ESTABLISHED; LITTLE FEEDBACK	ACTIVE AT ALL LEVELS; SOME FEEDBACK	ACTIVE AND EVOLVING WITH FULL FEEDBACK THROUGH PROCESS
QUALITY FOCUS	NONE TO LITTLE	SOME BUT NOT CONSISTENT	KNOWS HOW TO IMPROVE	ACTIVELY PURSUING IMPROVEMENT	FULLY INTEGRATED AS A WAY OF DOING BUSINESS
CUSTOMER FOCUS	NONE	SOME FOCUS; NO FEEDBACK	SOME FOCUS; SOME FEEDBACK	MAJOR FOCUS WITH SOME FEEDBACK	MAJOR AND INTEGRATED FOCUS; FULL FEEDBACK
TECHNICAL AWARENESS	MINIMAL	AWARE OF OPPORTUNITIES; MINIMAL USE OR PURSUIT	PURSUING OPPORTUNITIES	VIGOROUSLY PURSUING OPPORTUNITIES	FULLY PURSUING INTERNAL AND EXTERNAL PROFESSIONAL AND TECHNICAL OPPORTUNITIES

butes that provide coverage and completeness from an industrial management viewpoint.

Attributes. Eleven attributes are defined and applied to each process stage. These attributes and their definitions are as follows:

1. Process: The systematic flow and relationships of tasks and information needed to produce a product.
2. Methods: The systematic procedures and techniques used to accomplish a task.
3. Adherence to Practices: A properly defined, proven, and commonly understood process ethic that is adhered to by the product development team members, and the consistency with which the process ethic is followed.
4. Tools: The automated support of tasks, methodologies, and practices.
5. Change Control: The methodology by which all changes to the product are controlled.
6. Data Gathering: The collection of appropriate process data and information that illustrates the process performance.

7. Data Communication and Use: The effective analysis and communication of process data and information to improve the process.
8. Goal Setting: The establishment and use of quantifiable goals or targets for the purpose of improving the process.
9. Quality Focus: The pursuit and achievement of product excellence in every process task, including understanding of and action on what must be changed in the process to accomplish product excellence.
10. Customer Focus: The achievement of customer needs and requirements in the product at each process stage. A customer is viewed as both the end user of the product and the next person to work on the development of the product during its life cycle.
11. Technical Awareness: The technical knowledge of the state of the art for products and processes used in the profession.

The maturity grid

Given the dissection of the product life cycle into stages and related attributes, there now exists a matrix of cells that allows for a level of quantifiable evaluation regarding the effectiveness of the work being performed to develop software.

The notion of a "maturity grid" is used. This notion was originally introduced by Phillip Crosby in his evaluation of quality.[7] The maturity grid for the Site Studies was developed such that each cell had a set of attribute characteristics that would delineate the evaluation from a 5 (low) to a 1 (high) level of maturity. The same description was used for each attribute across all the process stages. It was recognized that this practice could imply that each cell carried equal weight for each stage in the evaluation. This implication was clearly not intended and, in fact, was avoided by not aggregating the individual cell scores into one overall product score. Additionally, the product groups were asked to focus on the more relevant cells for each stage and to use the others as ancillary input when the groups worked on their improvements or action plans.

The Evaluation Matrix shown in Figure 3 looks like a spreadsheet for each process stage. Across the top is a grade range with a span of 5 to 1. A 5 means traditional. A 1 means fully integrated into the business. Between these limits are gradients of maturity. The 11 attributes are defined across the side of the matrix. At the intersection of each maturity gradient and attribute heading (the cell) is found the characteristic that defines the degree of maturity of an attribute.

Phases of the study

The product study procedure has six major phases: planning, the on-site study, analysis, initial feedback, final feedback with recommendations, and the action plan developed by the product groups. This section discusses these study phases.

The product study procedure has six major phases.

Phase I—Planning. This phase is concerned with ensuring that all the logistics and necessary people for each product to be studied are in place and that the affected people are aware that a Site Study is about to occur.

Phase II—The study

Who is interviewed. This phase entails talking to planners, designers, coders, testers, technical writers, and others who are working daily with the product at a technical level. Usually, one or more key technical nonmanagement people, such as the "team leader" for each stage, are selected for the interview for each process stage.

The study team is interested in interviewing people who can share technical insights about their work, and who can offer a historical perspective about the way they have executed their process, about the way they are now running the process, and about the way they plan to change the process in the near future for their product area. The actual choice of who is to be interviewed is left to the management of the product under study.

The key management person who is always interviewed is the product manager. In the initial interview, the study team is interested in gaining a perspective of the development process; that is, what the product manager believes is the process that his/her management and technical group are performing. Additionally, the study team will ask for areas on which the manager wants the study to focus. Other managers may be interviewed on the basis of the suggestion of the product manager.

The study team. The study team is made up of three to five experienced programmers with broad programming backgrounds and knowledge of the industry, the local process manager or the equivalent, and one coordinator selected by the process manager whose duties will include arranging the interview schedule, inviting the interviewees, and logging materials provided by the interviewees. One of the members of the study team is the manager, whose primary job is to act as moderator and ensure continuity across the studies.

It is essential that the study team members be trained in interview techniques and also be technically qualified. They should be able to talk easily to and relate to the technical people they will be interviewing and to quickly establish a peer relationship with each of the interviewees. Team members should do nothing to alienate the interviewee. In fact, the challenge for them is to remain unbiased and try to see the work and the work environment as if they were the interviewee. This is an essential difference between this process and other similar forums such as product audits, assurance evaluations, etc.

Instructions to interviewers. When an individual is invited to be interviewed by the study team, he or she is given a set of guidelines for preparation regarding what should be brought to the interview session and what to expect during the session. The Appendix contains an example of the guidelines given to each interviewee. The complete interview will typically last about one hour for each interviewee.

The interview session. The person being interviewed could possibly be intimidated by several people seated around a conference table asking one question after another. For this reason, everything possible is done, in both arranging the room and conducting the session, to eliminate any threatening feeling. Upon arrival, the interviewee is greeted informally by one or two of the study team members. A brief reminder about informality and confidentiality is given.

The interviewee then begins, using the presentation material he or she has brought. As the discussion progresses, the interviewee usually begins to associate with the group, especially when, through the questions asked, it becomes apparent that the study team members are technically competent in the field being discussed, are in fact peers as programmers, and more important, are not outsiders who are only causing more work in already busy schedules.

No fixed set of questions is asked at every session. There is, however, a set of questions from which the study team draws its focus. If the interviewee has discussed all the topics in the question set, the study team focuses on some area particularly highlighted by the interviewee or an area requiring better understanding. In most cases, a series of questions from the defined set will be asked to ensure that each product is being evaluated from the same perspective.

The questions are delivered in a matter-of-fact manner without judgment or emotion, and are asked in such a way that the interviewee can offer as much exposition as he or she believes is necessary to complete the answer. Questions requiring a yes or no answer are avoided because they offer minimal information. For example, the question, "Do you conduct code inspections?" will yield far more information if phrased, "Can you describe the method by which you validate the code?"

Near the end of the session, a key question is always asked: "If you could have any wish granted for improving your work process for the quality of the product, what would the most important items be?" This question usually promotes a summarizing of the key problems and proposed solutions for the process stage under discussion. Finally, the interviewee will be asked if anything was overlooked that should be discussed. If so, and there is no time to complete the discussion in the allocated time slot, an off-line session will be scheduled with one or more of the study team members.

Phase III—Analysis

Team consensus on interview results. At the end of each interview day, for each process stage the study team summarizes the strengths observed as well as the items that require focus for improvement. As it is completed, each stage is evaluated by the team against the list of attributes common to all process activities.

Once this evaluation is completed by each team member, the team meets to reach a full majority consensus of the evaluation. In many cases, the team members will be in agreement, but in some cases they will have to discuss and resolve their apparent differences. For example, one member may have scored the process attribute for Unit Test as a 1, while another may have scored it a 3, and the rest of the team all scored it as a 2. Obviously they were all in the same room at the same time, listening to the same interviewee, so why the differences? Here it needs to be understood that the study team members are working under a high level of stress during a series of days, and not all of the members will be able to perform at peak levels over a sustained period. Therefore, they draw on and support each other during their evaluations. The perspective they are trying to maintain is one of complete objectivity. The team employs a modified Delphi method[8] to reach consensus and to remove the human differences in what any one individual may hear or evaluate. Thus, a more accurate representation of the information provided by the interviewees is made.

The study team completes a report that represents the initial feedback.

In the traditional Delphi procedure, the group members never meet face to face, as they do in the Site Studies. Additionally, the moderating and feedback of modified responses is done independently in the Delphi method until resolution is reached; this is done dynamically by the Site Study team after the first independent evaluations are made.

Before any evaluation is given back to the product people, the study team always reaches a consensus, with no minority reports or views. In some cases this requires the team to go back to the interviewee or to obtain additional information from other sources.

Supplemental material. All of those interviewed will bring one or more items of supplemental material to support or help explain their process activity and work products. Typical items brought are requirements documents, design specifications and other documentation, project plans, test plans, descriptions of locally developed tools, program size projections, defect tracking data, and other project control reports. The material may also include technical reports and other descriptions of new methodologies and techniques developed by the group for the product under study.

Most of the material is examined briefly and discussed during the interview session. It is then examined in more depth after the interviews are complete. The material is thoroughly evaluated by the study team while it is producing its final report.

The objectives of studying the material in depth are (1) to corroborate or clarify points discussed during the interview, and (2) to identify local methodologies and tools which might be disseminated to other locations that could benefit from them and which might be included in the corporate long-range process and tools strategies.

Phase IV—Initial feedback. After the study team has completed all interviews, examined all the material provided, and completed the evaluations for each process stage by attributes, it completes a report that represents the initial feedback to the managers responsible for the product. This feedback comprises two major sections: (1) strengths and (2) areas for focus.

While the study team members are individually and then later as a group completing the evaluations, they maintain a list of areas or items that particularly foster at least one of two aspects viewed as strengths:

A. An item which is at or is well on its way to reaching a full level of business maturity, or
B. An item which has initiated a change in the process and is evidently a best proven example that should be propagated.

Additionally, the study team maintains a list that contains weaknesses or areas requiring focus, i.e., items which are being performed or used on the project and for which there is a known better alternative that would lead to improved quality, productivity, or schedules.

These initial feedback sessions are included for two primary purposes. First, they reflect the initial findings back to management. These findings are only intended to show what is actually being done in software production for the particular product; i.e., to "hold up a mirror" to the management of a development project.

In some cases, this feedback may be challenged on one or two topics. In this case, it is used as an opportunity to obtain more information on the point of contention. The team members never debate the point; rather, they ask how they can get additional information regarding the issue raised. The interest of the team is to present the facts. If the input is in error or biased, the study can be flawed, and only additional information will clarify the differences. In all cases, a follow-up session is initiated with the technical personnel, and the topics raised as issues are discussed again with the management of the product group at the final feedback session.

The second reason for the initial feedback is to provide more immediate information, since the final report will take longer to complete.

Phase V—Final feedback. After all the information obtained through interviews, initial feedback sessions, and follow-up sessions is complete, the study team goes back to its home location to complete the final report. Preparation of this report requires a review of all of the data and documentation that were supplied. The study team in some cases then makes an analysis of the data beyond what may have been done by the group developing the product.

The main intent of the final report is to leave the development group of each product studied with a set of recommendations. These recommendations focus on areas with major potential payback from changes the study team believes the product group should consider making. In almost all cases, these recommendations are derived from the areas for focus initially shown to the managers. Each recommendation is followed by the detail needed to convince the project team that the recommendation is indeed well founded.

Finally, the maturity grids are presented, as there may have been a modification in the evaluations based on the review of the project documents and on the analysis of the project process data.

It is always suggested that management invite the entire project team to the feedback session, and at a minimum, the project interviewees should be in attendance. Additionally, the project manager can choose to invite any group or individual, because this information belongs to the manager, and the audience can be restricted or enlarged as he or she feels necessary.

Phase VI—Action plan. Within a nominal two months after the delivery of the final report, the project team is requested to share with the study team the actions it is going to pursue with respect to the recommendations.

It was decided to hold the studies on a yearly basis.

It is suggested that the project team put its action plan together from the bottom up, as this will involve all the project personnel and will more than likely expand the usefulness of the study results. The study is used to facilitate the evolution of change in the projects. Because people generally view change as threatening, they will resist change unless they are part of making it happen. They will additionally resist change if they do not perceive it as being helpful. For these reasons, the bottom-up approach is preferred.

In any case, the project team is responsible for using the action plan to develop greater process effectiveness in producing future projects in terms of quality, productivity, and schedules.

Restudy. On the basis of the results of the first set of studies from the eight programming laboratories, it was decided to hold the studies yearly. It was felt that within a one-year time frame, change would be significant enough in the programming process used by a product group to be discussed. This information in turn could add to an understanding of the rate of evolution of the consistent effectiveness of the process of a project, or, alternatively, could add to an understanding of what the inhibitors to evolution might be.

Concluding remarks

The Site Studies have served as a learning experience for all of us who were involved with them. Several items that were learned are particularly noteworthy and are recounted here. Most important, we learned once again that the most valuable resource on any project is the people who live with that project on a daily basis and who know better than anyone else what needs to be changed to achieve consistently better quality, productivity, and schedules.

We learned that especially in large organizations with multisite software development projects, it might be assumed that everyone is aware of new advances in process, methodologies, tools, and practices, but this is in fact a difficult status to maintain. Therefore, any avenue, such as the Site Programming Process Study, which opens up additional information and opportunity to the personnel will be an advantage in effecting change at a faster rate.

Not only can change in a programming culture happen, but it can be accelerated if people are not threatened by a study of their work habits. Therefore, it is most important that every effort be made to ensure that a cooperative, nonhostile, and confidential atmosphere for the study is pre-eminent in everyone's mind and is fully maintained.

The best action plans are developed from the bottom up and not from the top down. Getting all of the people involved who best know how to begin to focus the change is not only far-reaching and richer as a solution but stands a better chance of success because everyone is a part owner in the solution.

A study team must be manned with people of high technical competence who have a broad foundation in all aspects of the software business. If this had not been the case, the project and site personnel would soon have found our faults, and the study would not have been as successful as it was in serving as a catalyst for change.

A dedicated team is needed to work on such a study. The hours and stress of full-day interviews in which each team member must be constantly integrating new ideas with old, looking for relationships in information being provided, trying to remain unbiased in some situations which would try the objectivity of almost anyone, and working cooperatively with people who may not have the time or inclination to sit with the study team are some of the reasons that make this a difficult job to do.

Further, we learned how the process of the study team itself could be tuned to make the Site Studies work more effectively.

The approach described in this paper works in our environment, and we believe that it is immediately transferable to any software product environment regardless of size or number of locations.

The studies have turned up a number of excellent tools, subprocesses, methodologies, and practices that the various sites have developed and which, when added to the best available list, provide a richer programming environment with immediate potentials for improved quality, productivity, and schedules for all of the large-system programming laboratories in IBM.

Finally, we learned that use of these studies is a first step in the evolution of the software process, and it is more effective with the repetition of studies at a site or project. The maturity grid suggests that a product group has finished when it achieves a rating of "1" or "fully integrated in the business." In fact, the relationship only changes at that time, such that the highest rating is one which is always comparable to the bigger environment of the software industry, and one which is always evolving. Therefore, the programming process environment of a project will also always be changing.

The studies now occurring on a yearly basis at each site offer one of the best methods for continuously improving the quality, productivity, and schedules of the software being produced, because such studies provide a constant focus on changing the culture of the software development process.

Acknowledgments

We especially want to acknowledge Watts S. Humphrey, who asked us to put together a Programming Site Study Program. We would also like to acknowledge others who worked with us from time to time on the Site Studies, and whose involvement contributed to the success of these studies. Among these people are H. Gregorius, J. F. Hickey, J. Smith, B. Weitzenhoffer, all the process managers at the laboratories, and all the people on the projects who participated in these studies. Graham Penny is acknowledged for contributing the "Instructions for Interviewees" prior to the study beginning in Hursley, England.

Appendix. Instructions for interviewees

Your 10- to 15-minute presentation should be informal and held to one or two foils which encapsulate the process activity with which you are concerned. You need produce no other original material for the purposes of the study. You should not create any new procedures or work materials in order to satisfy the study.

Your foils should be in the form of a simple diagram showing the inputs to and outputs from your process activity, any subprocesses within the activity, involvement in other, earlier or later, process activities, reiterations of the process, etc. There is no specific format to follow; consider the foil largely as a memory aid for covering the salient characteristics of your activity.

Begin your presentation by explaining your own personal responsibilities within the activity in question.

Highlight the following items during your presentation:

- What documentation do you use/produce?
- What tools do you use?
- Do you have defined procedures?
- Do you follow them?
- How do you verify quality, accuracy, and completeness?
- What checks and balances, reviews and approvals, are involved?
- How do you track progress?
- How do you estimate resources needed (schedules and manpower)?
- How are changes controlled?

An informal discussion of about 30 to 35 minutes will follow your presentation. Please be prepared to expand on any of the items touched in your presentation. In addition, you may well be asked questions on the following topics:

- How are the necessary skills developed/maintained within your function?
- What is the level of customer awareness within your function?
- What is your function doing to improve the quality of its process?
- What are the process problems associated with doing your job?
- What can you think of that would most improve your process?
- What are the process problems associated with doing your job?
- What can you think of that would most improve your process activity? (Tools, procedures, etc.)
- Was the process used in this release different from that of the previous release?
- What changes, if any, do you intend making to the process for the next release?

Please remember that the details of this discussion will be kept confidential and that the study is above all interested in finding ways of improving the overall software development process and solving process problems.

Please note that the study is not particularly interested in

- Evaluating specific product quality parameters
- Product technical content
- Staffing and job levels
- Data processing availability/reliability problems
- How business decisions to make trade-offs between cost, schedule function, quality, etc., are made
- Interpersonal problems

Following is a checklist of the material you should bring with you:

- One copy of your foil presentation
- One copy of all workbooks and/or documented procedures in use within your activity (other than corporate or divisional standards and guides)
- Examples of documents used or produced by you as part of the process (objectives, specifications, test plans, inspection reports, performance reports, etc.)
- Examples of work material (problem reports, tracking/status reports, listings, traces, etc.)
- Copies of any memoranda that you consider in some way significant to your process, as for example, documenting a critical dependency or much-needed tool

Cited references

1. D. O'Neill, "The management of software engineering. Part II: Software engineering program," *IBM Systems Journal* **19**, No. 4, 421–431 (1980).
2. F. W. Taylor, *Principles of Scientific Management*, Harper & Brothers, New York (1947).

3. J. Aron, *The Program Development Process—The Programming Team, Part 2*, The System Programming Series, Addison-Wesley Publishing Co., Reading, MA (1983).
4. B. Boehm, "Software and its impact: A qualitative assessment," *Datamation* **19**, No. 5, 48–59 (May 1973).
5. J. R. Brown, "Modelling, measuring, and managing software cost," *Proceedings of Second Software Life Cycle Workshop* (August 1978).
6. R. A. Radice, N. K. Roth, A. C. O'Hara, Jr., and W. A. Ciarfella, "A programming process architecture," *IBM Systems Journal* **24**, No. 2, 79–90 (1985, this issue).
7. P. B. Crosby, *Quality Is Free*, McGraw-Hill Book Co., Inc., New York (1979).
8. J. F. Coates, "Some methods and techniques for comprehensive impact assessment," *Technological Forecasting and Social Change* **6** (1974).

Ronald A. Radice *IBM Information Systems and Storage Group, P.O. Box 390, Poughkeepsie, New York 12602.* Mr. Radice received his bachelor's degree from Upsala College in 1966 and joined IBM as a programmer immediately upon graduation. He held a variety of technical jobs in design, development, and testing of systems software products. In 1972, with M. E. Fagan, he began his work with software engineering which resulted in the inspection process for software design. Mr. Radice began his management career in 1974. Today he is the Information Systems and Storage Group Manager of Programming Process and is responsible for two groups that developed the Site Studies and the Process Architecture. He is a member of IEEE, ACM, and ASQC and is an Adjunct Associate Professor at Rensselaer Polytechnic Institute, where he teaches software engineering courses in the graduate program.

John T. Harding *IBM Information Systems and Storage Group, P.O. Box 390, Poughkeepsie, New York 12602.* Mr. Harding received his bachelor's degree in 1968 from Rensselaer Polytechnic Institute and an MBA in 1973 from Boston University. He joined the IBM Field Engineering Division in 1968. From 1970 to 1974, he was a customer, subsequently rejoining IBM. He has had several assignments in software development since then. At present, Mr. Harding is the manager of the Process Site Studies and has been involved with site studies since 1983.

Paul E. Munnis *IBM Information Systems and Storage Group, P.O. Box 390, Poughkeepsie, New York 12602.* Mr. Munnis attended Northeastern University in Boston. He joined IBM in 1964, and spent six years designing and implementing 1400 and 70XX emulators for the System/360. In 1973 he refined the software quality control techniques for inspection and taught them extensively to programmers. Beginning in 1974, he worked for three years as a tools planner for software and microcode. From 1977 to 1983 he was involved in the process, tools, and quality control efforts for the IBM System/38 development project. Mr. Munnis is presently a member of the Site Study team for process improvements.

Richard W. Phillips *IBM Information Systems and Storage Group, P.O. Box 390, Poughkeepsie, New York 12602.* Mr. Phillips attended the University of Wisconsin and joined IBM in 1954. He has held several technical and management posts, both in the field and at IBM development laboratories for OS and MVS software development. His technical accomplishments have included the development of programming quality measurement systems and predictive models for analyzing program quality. From 1980 to 1983, he was team leader in the Advanced Technology effort and early pilot projects that preceded and helped to found the Planning and Design Methodology described in this issue. He is currently active in a software engineering function in IBM. Mr. Phillips is a member of ACM and IEEE and is an Adjunct Associate Professor at Rensselaer Polytechnic Institute, where he teaches software engineering courses in the graduate study program.

Reprint Order No. G321-5241.