Why is software late? : an empirical study of reasons for delay in software development

Why is Software Late? An Empirical Study of Reasons For Delay in Software Development

Michiel van Genuchten

Abstract—**This paper describes a study of the reasons for delay in software development that was carried out in 1988 and 1989 in a Software Engineering Department. The aim of the study was to gain an insight into the reasons for differences between plans and reality in development activities in order to be able to take actions for improvement. A classification was used to determine the reasons. One hundred and sixty activities, comprising over 15 000 hours of work, have been analyzed. Actions have been taken in the Department as a result of the study. These actions should enable future projects to follow the plan more closely. The actions for improvement include the introduction of maintenance weeks. Similar studies in other software development departments have shown that the reasons varied widely from one department to another. It is recommended that every department should gain an insight into its reasons for delay in software development so as to be able to take appropriate actions for improvement.**

Index Terms— **Analysis of software development, empirical study, improvement, measurement, metrics, project management, reasons for delay, software development, software engineering management.**

I. Introduction

There is frequently a difference between the planned and actual progress of a software project. Why projects do not run according to plan is less clear. This paper describes a study which was carried out in 1988 and 1989 in a Software Development Department. The aim of the study was to obtain information about the differences between plans and reality in software development. The study led to actions for improvement in the department concerned, which should enable future projects to follow the plan more closely.

The aim of this paper is to add to the present knowledge concerning the reasons for overrun and delays in software development. The paper consists of the following sections: Section II describes a number of surveys on delays and reasons for delays, as described in the literature. Section III explains the definition and planning of the study. The results of the study are given in Section IV. The results were interpreted by the project leaders participating in the study. The interpretation of the results is described in Section V. Finally, the conclusions of the study are presented.

Manuscript received February 15, 1990; revised January 10, 1991. Recommended by M. S. Deutsch.

The author is with the Department of Management Information Systems and Automation, Faculty of Industrial Engineering, Paviljoen D3, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.

IEEE Log Number 9144265.

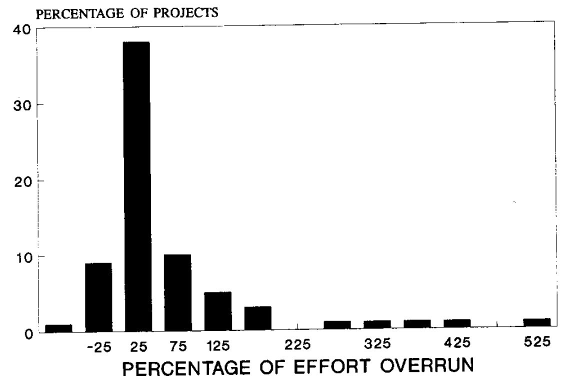

Fig. 1. Distribution of relative effort overruns [7].

II. Surveys on the Overrun of Development Projects

Three empirical studies concerning the overrun of development projects will be discussed in this section. These studies will be referred to as "surveys." The definition of the surveys and their results will then be compared with the study described in Sections III through V of this paper.

A. Survey by Jenkins, Naumann, and Whetherbe [7]

Jenkins *et al.* [7] interviewed the developers of 72 information system development projects in 23 major U.S. corporations. The aim of the survey was to collect empirical data on the systems development process in organizations. The average duration of the projects was 10.5 months. Over 70% of the projects took less than 1000 person days to finish. The users of the developed systems stated that they were "satisfied" to "very satisfied" with the result in 72% of the projects. The relative effort overruns are given in Fig. 1.

The average effort overrun was 36%. Fig. 1 shows that 38% of the projects had an overrun of between 0 and 50%. Nine percent of the projects had an underrun of between 0 and 50 %. The relative schedule overruns are given in Fig. 2.

The average schedule overrun was 22%. Fig. 2 shows that 40% of the projects had an overrun of between 0 and 50%. One conclusion of Jenkins *et al.* was that the cost and schedule overruns seem to be uniformly distributed among large, medium, and small projects. They did not look into the reasons for delays and overruns.

B. Survey by Phan, Vogel, and Nunamaker [9], [10]

Researchers at the University of Arizona attempted to

0098-5589/91/0600-0582$01.00

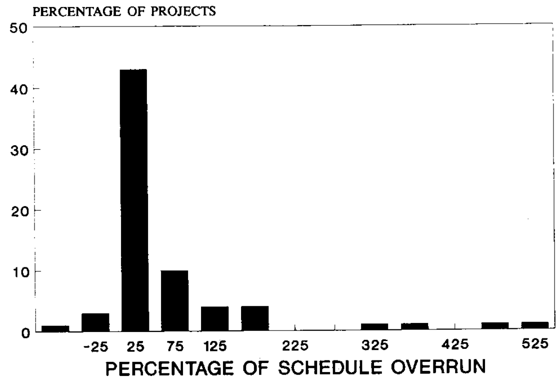

Fig. 2. Distribution of relative schedule overruns [7].

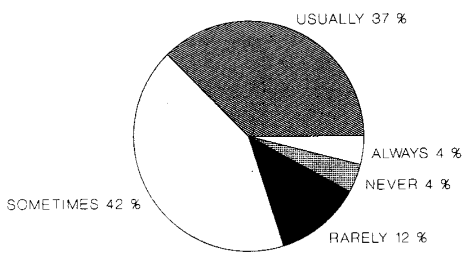

Fig. 3. Prevalence of cost overruns [9].

determine why the planned lead times and costs of information system development projects were overrun [9], [10]. Questionnaires were sent to 827 members of the American Institution of Certification of Computer Professionals. The survey yielded 191 responses. The respondents were involved in projects with an average duration of 102 person months. On average, the lead time was 14 months and 17 people worked on a project. The average cost overrun was 33%, similar to the 36% overrun reported by Jenkins *et al.* [7].

The survey was comprised of 100 questions. In 72 of these the respondents were asked to recall the frequency with which the events occurred as: (a) always; (b) usually; (c) sometimes; (d) seldom/rarely; or (e) never. Over 70% of the respondents claimed that user requirements and expectations were usually met. Fig. 3 shows the prevalence of cost overruns.

Only 16% of the respondents answered that they never or rarely had cost overruns. Cost overruns were usual for 37% of them. Fig. 4 shows the prevalence of schedule overruns.

Fig. 4 shows that more than 80% of the respondents stated that their projects were sometimes or usually late. The survey also addressed the reasons for cost overruns and late deliveries. According to 51% of the respondents, over-optimistic estimation was usually a reason for a cost overrun. Almost 50% stated that frequent changes in design and implementation were usually a reason for a cost overrun. Nine percent stated that these were always a reason. The survey also investigated why the product lead times were overrun. Over-optimistic planning was a reason to which 44% usually attribute the delay. Minor and major changes were usually a reason for 33 and 36% of the respondents, respectively. The lack of software development tools was only mentioned by 17% as a usual reason.

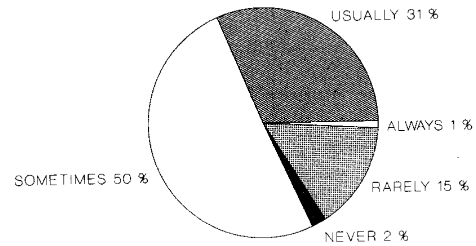

Fig. 4. Prevalence of late deliveries [10].

The four actions most frequently taken to regain control over delayed projects were:

1) Upgrading the priority of the project
2) Shifting part of the responsibility and obligations to other groups
3) Renegotiating the plan and schedule
4) Postponing features and upgrades to the next version.

C. Survey by Thambain and Wilemon [12]

The aim of a field study by Thambain and Wilemon [12] was to investigate the practices of project managers regarding their project control experiences. The scope of the survey was not confined to software engineering projects; the leaders of electronics, petrochemical, construction, and pharmaceutical projects were interviewed. Data was collected from 304 participants in project management workshops or seminars. Those questioned had an average of five years' experience in technical project management. The average lead time for the projects was one year, and on average eight people worked on a project.

Among other things, the survey investigated what the project leaders and their superiors (such as senior functional managers or general managers) believed to be the reasons for cost and lead time overruns. The reasons for overruns were arranged in order of importance by project leaders and general managers. The results are given in Table I.

It is striking to note that the project leaders and general managers do not agree on the importance of 9 of the 15 reasons. According to the researchers, the "practical implication of this finding is that senior management expects proper project planning, organization, and tracking from project leaders. They further believe that the external criteria, such as customer changes and project complexities, impact project performance only if the project had not been defined properly and sound management practices were ignored. On the other hand, management thinks that some of the subtle problems, such as sinking team spirit, priority shifts, and staffing are of lesser importance" [12].

The researchers also investigated the reasons which caused the problems referenced in Table I. These less obvious reasons were called "subtle reasons," which can be classified into five categories:

TABLE I
DIRECTLY OBSERVED REASONS FOR SCHEDULE SLIPS AND COST OVERRUNS

RANK BY			Agreement between general and project management
General managers	Project managers	PROBLEM	
1	10	Insufficient front-end planning	Disagree
2	3	Unrealistic project plan	Strongly agree
3	8	Project scope underestimated	Disagree
4	1	Customer/management changes	Disagree
5	14	Insufficient contingency planning	Disagree
6	13	Inability to track progress	Disagree
7	5	Inability to track problems early	Agree
8	9	Insufficient number of checkpoints	Agree
9	4	Staffing problems	Disagree
10	2	Technical complexity	Disagree
11	6	Priority shifts	Disagree
12	10	No commitment by personnel to plan	Agree
13	12	Uncooperative support groups	Agree
14	7	Sinking team spirit	Disagree
15	15	Unqualified project personnel	Agree

- Problems with organizing the project team
- Weak project leadership
- Communication problems
- Conflict and confusion
- Insufficient upper management involvement.

Obviously, the subtle reasons cited by the project leaders and general managers were not technical reasons, but related to organizational, managerial, and human aspects.

III. DEFINITION AND PLANNING OF THE STUDY

A. Definition of the Study

The framework of experimentation, as proposed by Basili *et al.* [1] will be used to define the study which is described in this paper. According to this framework, a definition consists of six parts: motivation, object, purpose, perspective, domain, and scope. The motivation of this study was to gain an insight into the reasons for delay in order to be able to improve the control of future development projects. This new insight should lead to actions for improvement designed to enable future projects to follow their plan more closely. The object of the study was defined as the primary entity examined [1]. The object in this case was software development activities. Projects can be analyzed on various levels of detail; namely, as a whole (as done by Jenkins *et al.* [7]), at phase level, or at activity level. Data was collected and analyzed at the activity level in this study, because experience has shown that a project generally does not overrun because of one or two main problems, but rather because of a large number of minor problems. According to Brooks [3]: "How does a project get one year late? One day at a time." These small problems could almost certainly be overlooked if data were collected at the project level. In this study an activity was defined as a unit of work that is identified in a plan and can be tracked during its execution. A typical activity may be the specification of a subsystem, the design of a module, or the integration of some modules.

TABLE II
THE DEFINITION OF THE STUDY

Motivation	To increase insight into the reasons for dela
Object	Software engineering activities
Purpose	To evaluate reasons for delay
Perspective	Project leader
Domain	Project
Scope	Six projects in one development department

The purpose of the study was to evaluate the reasons for delay. This was done from the perspective of the project leader. The domain studied was software projects. The scope of the study covered six development projects in one software development department. The definition of the study is summarized in Table II.

B. Planning the Study

The motivation of the study was to gain an insight into the reasons for delay in software development. The kind of questions the study aimed to answer were:

- What are the predominant reasons for delay?
- What is the distribution of the reasons for delay?
- How is the delay distributed over the phases of a project?
- Which actions for improvement can prevent delay in future projects?

The following basic principles were used for data collection:

1) The control of a project refers to the control of quality, effort, and lead time. The study was based on the assumption that an activity is only completed when the (sub)product developed fulfills the specifications. In other words, if the quality of the product developed is adequate. In the department concerned this was monitored by reviews and testing. This assumption allowed attention to be focused on the collection of data relating to time and effort.

TABLE III
DATA DETERMINED FOR EACH ACTIVITY

	PLANNED	ACTUAL	DIFFERENCE	REASON
EFFORT	—	—	—	—
STARTING DATE	—	—	—	—
ENDING DATE	—	—	—	
DURATION	—	—	—	—

2) Data collection focused on the differences between a plan and reality. All planning data were obtained from the most recently approved plan. If a project was officially replanned, the new plan was taken as the starting point for the comparison between the plan and reality. The consequences of a replan will therefore not show up in the measurements. Six projects were studied: one of them was not replanned, four were replanned once, and one was replanned twice during the study. It might be argued that the differences between plan and reality were greater than the measurements will show.
3) The third principle was that data collection should not take the project leaders much time. This was a condition stated by the development department.

The definition of the study and the above principles resulted in a one-page data collection form. This consisted of a table with the data to be collected for each activity and a classification of reasons for delays. The table is shown in Table III.

The planned and actual efforts were expressed in hours. The starting and ending dates were given in weeks. The duration of an activity was defined as the calendar period between the starting and ending dates. All planning data were obtained from the most recently approved plan. The difference column indicated if there was any difference between the plan and reality. The reasons for the three types of differences were distinguished in the final column:

- The reason for a difference between the planned and actual effort
- The reason for a difference between the planned and actual starting date
- The reason for a difference between the planned and actual duration.

A reason for the difference between the planned and actual ending date was not mentioned, because this difference can be explained by the difference in the starting date and the difference in duration.

Obviously, much of the data in Table III was not only kept for the purpose of this study: the planned and actual hours and duration were also required for normal project control purposes. A recent survey [11] showed that in practice, data of this kind are not kept as a matter of course; as many as 50% of the respondents claimed that they did not record progress data during the course of their projects. In this study the project plans provided the planned effort, starting date, and ending date. The clerical office provided the actual data, which were collected on the basis of time sheets. The actual data were validated in interviews with the participating project leaders every other week.

TABLE IV
GROUPS OF REASONS

Group of reasons	Description Reason relating to
capacity-related	the availability of the developers
personnel-related	the experience of the developers
input-related	conditions which must be fulfilled
product-related	the software product to be developed
organization-related	the organization in which the development takes place
tools-related	the tools used to develop the software
other	none of the previous categories

The final column was filled in specially for this study. This was performed by the project leader who, in consultation with the researcher, determined the reasons for differences between planning and reality. A classification was used, for two purposes, to determine a reason: first, the classification gave structure to the reasons identified and allowed results to be compared; and secondly, the classification saved time for thinking up reasons. Six groups of possible reasons for differences were identified in the classification. The division into six groups was based on a discussion with the project leaders concerned and on a previous study [6]. The groups are listed in Table IV.

The division into six groups has proved to be valid for several (software) development departments. In fact, similar studies using the same groups of reasons were applied in a number of departments. About 30 reasons for delay were found within the groups. A first classification of reasons was identified after a discussion with the participating project leaders. A definite classification of reasons was identified after a pilot study. Similar studies in other departments showed that the reasons were specific to the engineering environment in question because of differences among the software engineers, the type of software developed, and the organization of the department. This confirms the measurement principle, which states that metrics must be tailored to their environment, as formulated in [2]. The classification of reasons, as used in this study, is displayed in Table V.

A reason labeled "other" was included in each category, because it was not exactly clear at the start of the study what reasons could be expected. During the study, however, it was found that the reason "other" only needed to be used rarely.

TABLE V
THE CLASSIFICATION OF REASONS AS USED IN THIS STUDY

CAPACITY-RELATED REASONS
11 capacity not available because of overrun in previous activity
12 capacity not available because of overrun in other activity
13 capacity not available because of unplanned maintenance
14 capacity not available because of unplanned demonstration
15 capacity not available because of other unplanned activities
16 capacity not available because of other causes
19 other

PERSONNEL-RELATED REASONS
21 too little experience with development environment
22 more inexperienced people in team than expected
29 other

INPUT-REQUIREMENTS NOT FULFILLED
31 requirements late
32 requirements of insufficient quality
33 (specs of) delivered software late
34 (specs of) delivered software of insufficient quality
35 (specs of) hardware late
36 (specs of) delivered hardware of insufficient quality
39 other

PRODUCT-RELATED REASONS
41 changing requirements during activity
42 changing of the interfaces during the activity
43 complexity of application underestimated
44 more problems than expected with performance requirements or memory constraints
45 product of insufficient quality developed (redesign necessary)
49 other

ORGANIZATON-RELATED REASONS
51 less continuity in project staffing than expected
52 more interruptions than expected
53 influence of software Quality Assurance
54 bureaucracy
59 other

TOOLS-RELATED REASONS
61 development tools too late or inadequately available
62 test tools too late or inadequately available
69 other

OTHER
71–79

If the actual hours, starting dates, and ending dates were recorded, little time was needed to determine the reason for any difference. In practice, determining the actual hours and starting and ending dates was found in practice to take a great deal more time than determining the reasons. This was done in an interview once every other week with the project leader in question. It was important to analyze the data during the project, because it would have been difficult to collect accurate data after the project had finished, and validating the data would have been almost impossible. Several reasons could be given for each difference, with a maximum of four; in practice, it was found that the difference could usually be ascribed to one reason.

C. Comparison of the Study and Surveys

The study definition which was just described will be compared with the surveys, as in Section II. They will be compared with respect to their motivation, object, scope, and the data collection technique used. The motivation of the survey by Jenkins *et al.* [7] was to conduct empirical research on the information systems development process in organizations. The survey by Phan *et al.* [10] aimed to collect factual data with regard to the management and control of software projects. Thambain and Wilemon [12] investigated the practices of project managers in relation to their project control experience. The motivation of the study described in this paper was to gain an insight into reasons for delay.

The object of the three surveys was projects—Jenkins *et al.* [7] and Phan *et al.* [10] took information systems development projects as their object, while Thambain and Wilemon's [12] survey was concerned with engineering projects. The object of the study described in this paper is the activities performed within a project. The scope of the surveys covered multiple projects in multiple organizations. This study is limited to six development projects within one department. The last and most obvious difference between the surveys and the study described in this paper is the data collection technique. Jenkins *et al.* conducted interviews on 72 completed projects. Phan *et al.* sent out a questionnaire and received 143 qualified responses. Thambain and Wilemon collected questionnaires from 304 participants in workshops and seminars. In the study described here, data were collected and validated during the execution of the projects on the basis of a number of interviews with the project leaders and the available project data. Because of the differences mentioned, the study and surveys were complementary, rather than similar.

IV. Results

The study took place in a Software Development Department in the second half of 1988 through the first half of 1989. The Department was concerned with the development and integration of system software in the operating system and data communication fields. The Department employed 175 software engineers and covered a range of 300 products. Six representative projects in the Department were selected for the study. A total of 160 activities in the projects were studied. The data in Table III were determined for each activity; these were the planned and actual hours and the starting and ending dates. The average duration of an activity was 4 weeks, and the average effort was close to 100 person hours.

When determining the actual effort and actual starting and ending dates, the existing registration was found to be of limited value, because some of the data on the actual implementation of the project were not available in a usable form. Recording starting and ending dates was no problem, because management emphasized the control of duration. Starting and ending dates were reported at the progress meetings. The number of hours spent on each activity was difficult to determine in the first part of the study for two reasons: first, the lack of reliability of the recorded hours. The validation of the data by project leaders showed that the difference between the recorded hours and the impression of the project leader was sometimes too large to be credible. Second, it was found that the numbering of the activities by the project leaders was found not to be unique in every case. This meant that the hours recorded could not be related to activities. The actual hours were not recorded if the effort could not be related to activities or the validation indicated that something was wrong. As a result, the planned and actual efforts could only be compared for 97 of the 160 activities.

The most important results of the study are presented in the form of four figures. Fig. 5 shows the frequency distribution of the difference between the planned and actual durations of the activities.

Fig. 5 shows that over 30% of the activities were finished according to plan. Nine percent show a one week underrun; 17% show a one-week overrun. Fig. 6 shows the relative difference between the planned and actual efforts for 97 activities. This figure relates to only 97 activities due to the problems which occurred in the recording of hours for each activity.

Fig. 6 shows that about 50% of the activities overran their plan by more than 10%. About 30% underran their plan by more than 10%. The comparison of the planned and actual figures yielded some useful insights. It showed, for instance, that the relative differences between planned and actual efforts increased for the subsequent phases of the project; the delays and overruns increased toward the end of the project. The same result has been found in other engineering environments [4]. This fact makes it possible to discourage the idea that delays can be overcome as the project progresses.

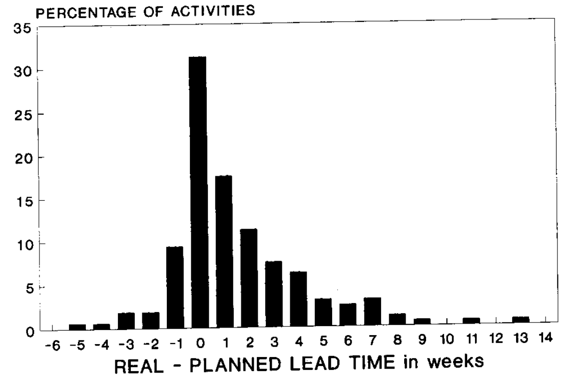

Fig. 5. Frequency distribution of the difference between the planned and actual durations (N=160).

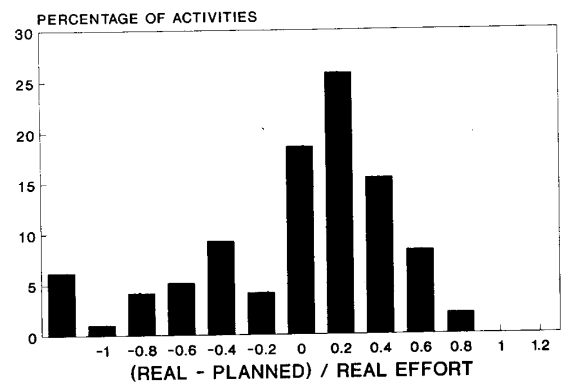

Fig. 6. Frequency distribution of the relative difference between the planned and actual efforts (N=97).

Figs. 7 and 8 present the reasons for the delays and overruns. During the study it was found that many activities started too late. Fig. 7 shows the distribution of the reasons for activities which start too late. These were divided into the groups identified in Section III. Note that when an activity started too late because of a delay in a previous activity, it was recorded as reason 11, a capacity-related reason (see Table V). This explains the large capacity section in Fig. 7.

The input-related reasons had to do with the late delivery of hardware components developed in parallel with the software. The start of the software development activities was also delayed because of this. The reasons for the differences between the planned and actual duration are listed in Fig. 8.

Within the groups identified it was found that the most frequent reasons for differences between the planned and

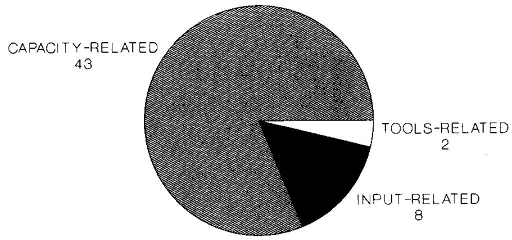

Fig. 7. Distribution of reasons for differences between the actual and planned starting date ($N = 53$).

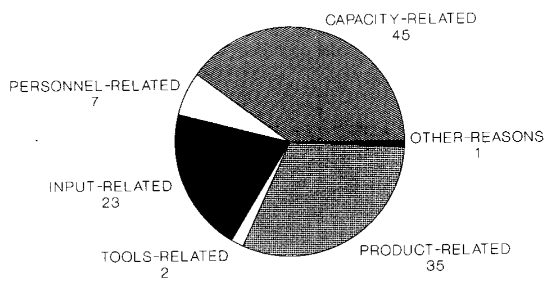

Fig. 8. Distribution of the reasons for differences between the actual and planned durations ($N = 113$).

actual durations were: reasons 12 to 16: "more time spent on other work than planned" (these reasons were named in 27% of the cases). Reason 43: "complexity of application underestimated." Some outsiders blame all the software delays on underestimation. In this case, underestimation was given as an explanation in about 20% of the cases.

V. Interpretation of the Results

The results were interpreted during a meeting attended by the project leaders taking part, the department manager, and the researcher. In the researcher's opinion, data of this kind should, in the first place, be analyzed together with the people involved in data collection. Six reasons for this are given: first, it is the engineer's, project leader's, and manager's job to control software development. They should be supported with all the available data. Second, those involved represent the knowledge of software development in the department concerned; this knowledge is needed to interpret the results. Third, those involved can assess the feasibility of any actions for improvement. Fourth, actions which are decided on by members of the organization concerned will be accepted more easily, and thus be implemented more quickly, than actions recommended by an outsider. Fifth, interpretation of the results shows the people involved that the data is being used for their benefit. This should motivate them to participate in future analyses. Finally, a meeting like this can contribute to creating a common understanding among project leaders and general managers regarding problems within the department. Collective interpretation of the results can help to prevent different perceptions of the problems, as was reported by Thambain and Wilemon [12] (see Section II).

During the meeting it was found that the results of the study confirmed and quantified a number of existing impressions of project leaders and the manager. For some of those present, the results provided new information. For instance, it was not clear to everyone that the amount of other work had such a significant effect on duration.

The following are examples of the possible actions for improvement that were discussed at the meeting.

It was found that the amount of "other work" in the projects studied was underestimated. During the meeting it was shown that the other work consisted mainly of maintenance. Those present decided that in future projects, more time and capacity should be set aside for "other work."

During the meeting it was shown that the maintenance activities in particular constantly interrupted development. A number of possible ways of separating development and maintenance was discussed. The possibility of setting up a separate maintenance group was discussed and rejected. It was decided to schedule the maintenance work as far as possible in maintenance weeks, and to include two maintenance weeks in each quarter. It was obvious that not all maintenance can be delayed for a number of weeks. Any defect that affected the customer's operation was resolved immediately, irrespective of the maintenance weeks. Defects of this kind were only a small fraction of the defects, and correcting them involved only a small fraction of the maintenance effort. The vast majority of defects was found in products before they were released to customers. By carrying out most of the maintenance during maintenance weeks, it was hoped that development could proceed more quickly and with fewer interruptions during the other weeks. This suggestion was implemented by the department within one month after the meeting.

The department wanted to gain more insight into the origin of maintenance. Another analysis study started. Its aim was to gain an insight into the origin of maintenance in order to be able to take improvement measures which could reduce future maintenance effort.

At the end of the meeting it was concluded that the study had yielded sufficient results for those involved. A considerable contribution was the fact that ongoing discussions could now be supported by facts.

Comparable studies have been carried out in a number of other software development departments. The result of one of those studies is given for the sake of comparison, and also to discourage unjustified generalizations of the results given so far. Fig. 9 shows the reasons for differences between planned and actual durations, which are given for 80 activities, carried out in a development department which develops systems software and CAM software [8]. The groups of reasons distinguished in Section III were again used here.

The differences between the distribution of reasons given in Fig. 8 should be obvious. Based on ongoing measurements in a number of departments, the author concludes that the distribution of causes varies strongly for each department. Every department should therefore gain an insight into its

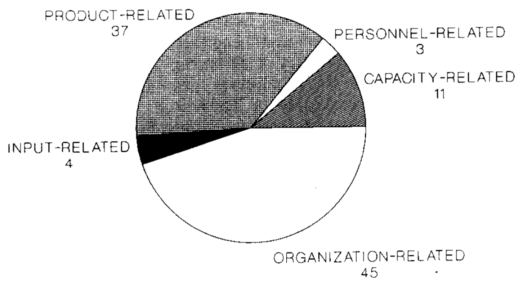

Fig. 9. Distribution of the causes of differences between the actual and planned durations in another department [8].

reasons for delay in software development projects in order to be able to take appropriate actions for improvement.

VI. Conclusions

The conclusions below consist of two parts: first, the study and its results are compared with the surveys discussed in Section II of this paper, after which the main conclusions of the study will be restated.

Three surveys concerned with the investigation of delays and the reasons for delays were presented in Section II of this paper. A comparison of the definition of the surveys and the study presented in this paper was given in Section III. The comparison showed that the surveys on the one hand, and the study on the other, were complementary rather than similar. The comparison of the results of the present study with the surveys described in the literature provides the following information:

The average overruns found in the present study approximated the overruns found by Phan *et al.* [9], [10] and Jenkins *et al.* [7]. However, in the present study the relative lead time overrun was greater than the relative effort overrun. Jenkins *et al.* found the opposite result.

Over-optimistic planning was cited as a probable cause in all the studies which examined reasons for delay. Phan *et al.* found that 44% of the respondents named over-optimistic planning as a reason. An unrealistic project plan and underestimation of the scope were named as major reasons in Thambain and Wilemon's [12] survey. The study described in this paper recorded underestimation of the complexity as a reason in 20% of the cases.

Thambain and Wilemon's investigation of the subtle reasons for delay indicate that the reasons were not technical in nature, but were related to organizational, managerial, and human aspects. The present study shows a similar result. The product- and tools-related reasons represent most of the technical reasons. They comprise only one-third of the reasons mentioned.

It must still be noted that relatively few studies on delays and their reasons have been described in the literature. Moreover, this statement is generally true for empirical studies of the control of software development.

An empirical study of the control of software projects was presented in this paper. An important advantage of the study definition selected was that, in spite of the limited effort required from the project leaders taking part, results were achieved fairly quickly. The cooperation of the developers and project leaders was vital in carrying out the study. One of the conditions for the cooperation of the project leaders was that it was made clear in advance for what the data collected would and would not be used.

Insight into the predominant reasons for delay enabled actions for improvement to be taken in the department concerned. An important conclusion was that the distribution of reasons for delay varied widely from one department to another. The author recommends that every engineering department should gain an insight into its reasons for delay in order to be able to take adequate actions for improvement.

This study targeted the activities within a project. If a project plan is regarded as a set of agreements concerning the work to be done, it might be said that the study investigated to what extent agreements within projects were fulfilled. External entities also have an influence on the execution of a project plan. One example is the fact that departmental management does not provide the planned resources. Another example is failure of a marketing department to deliver clearly defined requirements on time. The author concludes that to some extent, a project cannot be executed according to plan, because external entities do not fulfil their agreements. Software engineers should continue to investigate how agreements are fulfilled within projects. The author would also recommend a comparable study on the fulfilment of those agreements which influence the execution of a project plan, but are not controlled by the project team.

Acknowledgment

The author would like to thank the following people for their cooperation: E. Buijs, M. Fierst van Wijnandsbergen, E. der Veen, H. Keizers, G. Scholten, L. Hulstman, and E. de Vries, and also T. Bemelmans, F. Heemstra, M. Howard, R. Kusters, and again, M. Fierst van Wijnandsbergen, for their contributions to and comments on an earlier version (see [5]) of this paper.

References

[1] V. R. Basili, R. W. Selby, and D. H. Hutchens, "Experimentation in software engineering," *IEEE Trans. Software Eng.*, vol. SE-12, pp. 733–743, July 1986.
[2] V. R. Basili and H. D. Rombach, "The TAME project: toward improvement-oriented software environments," *IEEE Trans. Software Eng.*, vol. SE-14, pp. 758–773, June 1988.
[3] F. B. Brooks, *The Mythical Man-Month: Essays on Software Engineering*. London: Addison–Wesley, 1975.
[4] M. J. I. M. van Genuchten, "Towards a software factory," Ph.D. thesis, Eindhoven Univ. Technology, The Netherlands, 1991.
[5] M. J. I. M. van Genuchten and M. Fierst van Wijnandsbergen, "An empirical study on the control of software development," in *Proc. Conf. Organization and Information Syst.* (Bled, Yugoslavia), Sept. 13–15, 1989, pp. 705–718.
[6] F. J. Heemstra, "Estimation and control of software development projects," Ph.D. thesis, Eindhoven Univ. Technology, Kluwer, Deventer, The Netherlands, 1989.
[7] A. M. Jenkins, J. D. Naumann, and J. C., Wetherbe, "Empirical investigation of systems development practices and results," *Inform. Manage.*, vol. 7, pp. 73–82, 1984.
[8] F. L. G. van Lierop and R. S. A. Volkers, "Controlling software projects: a matter of measurement," Masters thesis, Faculty of Industrial

Eng., Eindhoven Univ. Technology, Kluwer, Deventer, The Netherlands, 1989.

[9] D. Phan, "Information systems project management: an integrated resource planning perspective model," Ph.D. thesis, Dept. Management Inform. Syst., Univ. Arizona, Tucson, 1990.

[10] D. Phan, D. Vogel, and J. Nunamaker, "The search for perfect project management," *Computerworld*, pp. 95–100, Sept. 1988.

[11] W. J. A. M. Siskens, F. J. Heemstra, and H. van der Stelt, "Cost control of automation projects: an empirical study" (in Dutch), *Informatie*, vol. 31, pp. 34–43, Jan. 1989.

[12] H. J. Thambain and D. L. Wilemon, "Criteria for controlling projects according to plan," *Project Management J.*, pp. 75–81, June 1986.

Michiel van Genuchten received the M.Sc. degree in industrial engineering from the Eindhoven University of Technology, The Netherlands.

He is currently employed by the Eindhoven University of Technology, Department of Industrial Engineering, Section Management Information Systems and Automation, and Philips Electronics (Lighthouse Consulting Group). His research interests include the control and analysis of software engineering, the application of production control concepts to software engineering, and the reuse of software.