Demarcating the Scope of a Predelivery Maintenance Process

Ahmad Salman Khan, Mira Kajko-Mattsson
KTH School of ICT
Sweden
askhan@kth.se, mekm2@kth.se

Abstract—Predelivery maintenance is one of the processes that has not received enough attention both within the academia and industry. Still, it counts as one of the least visible lifecycle processes. In this paper, we make it more visible by demarcating its scope within software lifecycle. We do it within eleven companies. Our goal is to find out how the industry understands predelivery maintenance and how it places it within software lifecycle. Our results show that predelivery maintenance is conducted by all the organizations studied in five different contexts. These organizations however use different terms for it or they treat predelivery maintenance as part of other processes. For this reason, we conclude that predelivery maintenance is a process that cross-cuts several lifecycle processes.

Keywords—transition, handover, postdelivery maintenance, pre-launch support, product roadmap

I. INTRODUCTION

The effectiveness of software maintenance is greatly influenced by what occurs or does not occur during development phase. From the maintenance perspective, the organizations should prepare their systems for future maintenance, they should designate their future maintenance organizations early during development, and they should allow these maintenance organizations to have insight into and influence on the system being developed. In this way, the maintenance organizations would avoid many of the postdelivery maintenance problems such as poor system maintainability, lack of system knowledge, and the like.

Awareness about the importance of predelivery maintenance is still very low. Despite the fact that the predelivery maintenance standard has been around for almost one decade, it has not achieved the penetration power within the software community as expected by the standard promoters [12]. Be that as it may, the term “predelivery maintenance” is still an unknown term within the industry and its scope is not well understood and defined neither by the standard, nor the academia, nor the industry. Still, we have difficulties to identify its activities, place them within software lifecycle, identify roles involved in it and establish its relationship to other processes.

In this paper, we demarcate the scope of a predelivery maintenance process. We do it within eleven companies. Our goal is to find out how the industry understands predelivery maintenance and how it places it within software lifecycle. This paper is related to [11] dealing with a handover process. Hence, the results presented herein are strongly related to the results presented in [11].

The remainder of this paper is as follows. Section II describes the method taken within this study. Section III briefly presents predelivery maintenance. Section IV describes the criteria used for demarcating the predelivery scope. Section V reports on how the industry defines the predelivery scope. Finally, Section VI makes conclusions and suggestions for future work.

II. METHOD

In this section, we describe the research method taken in this study. We first present our research steps. We then discuss the sampling and validity of our results. Just because this paper is closely related to [11], the method presented herein is very similar to the method presented in [11].

A. Research steps

Our research method consisted of four steps. These are (1) literature study (2) questionnaire design (3) data collection and (4) data analysis. Below, we briefly describe them.

In the literature study step, we studied scientific publications and reports on predelivery maintenance. Our goal was to gain knowledge in this domain. Unfortunately, due to the scarcity of the research results within this area, we only managed to find a handful of publications [2, 5, 12, 15, 16]. Hence, our further steps are entirely based on them.

In the questionnaire design step, we created a semi-structured questionnaire. The questionnaire was developed using evaluation criteria designed for the purpose of exploring whether, how, when and by who the predelivery activities were conducted in the software industry. The evaluation criteria and the questionnaire are going to be further discussed in Section IV.

It is known that the order of questions in a questionnaire can affect the flow of the interview process, the way the interviewees respond and the way the interviewers get an understanding of the interviewees’ domain [1]. To optimize the interview results, we conducted the questionnaire design and data collection steps interchangeably in two iterations. We did so with the intention to create as optimal questionnaire as possible.

The iterative approach between the questionnaire design and data collection steps implies that we interviewed and collected data in two consecutive iterations. As outlined in Fig. 1, we first created a preliminary version of a questionnaire and using it, we made interviews within three organizations. Our goal was to (1) evaluate that the interviews went smoothly with the initial order of questions and to (2) improve it by using the experience from
the first three interviews. We did so because we used students in the final data collection step. Before allowing students to conduct interviews, we had a minor step between the questionnaire design and data collection steps during which we prepared students for the interviewing task. The students then interviewed eight companies using the final version of the questionnaire. Finally, in the data analysis step, we compiled the answers for each question and synthesized and analyzed them.

The initial three interviews conducted by us resulted in an additional question (see Question 9 in Section IV and Fig. 3). Summing up, the first version of the questionnaire included eleven questions and the final version included twelve questions. For this reason, the results presented for Question 9 will report on the status within eight organizations only.

B. Sampling and Validity

The data sampling method used in this study was convenience sampling [14]. The reason is the fact that it is difficult to make organizations show willing for an interview. This means that we did not control the choice of the organizations involved in our study.

Regarding the choice of the first three organizations, we selected those who were available. Regarding the organizations chosen by the students, we allowed the students to select just any organization (large/medium/small and/or private/ government) in any country. The only requirement was that the organizations studied should conduct predelivery maintenance, and that the interviewees had knowledge about the process. Many of our students, coming from an international master program in Sweden, chose organizations in their own countries. Hence, the countries represented in this report are Pakistan, Bangladesh, Mexico, China, India, Iran and Sweden.

Fig. 2 presents the details of the organizations studied. Due to the sensitivity of the material presented herein, we do not name these organizations. Some of them however are major well-known multinational companies.

We are aware of the fact that the validity of the results may be strongly jeopardized when one involves students. To optimize the interview results and to secure their credibility, we did the following:

• Ordered questions: We made the effort to assure that each question transitioned smoothly from previous questions. Questionnaires that jump from one unrelated topic to another feel disjointed and are not likely to result in effective interviews.

• Used figures to support the interviewing process: To improve the communication between the interviewers and interviewees, we created three lifecycle roadmaps. These are presented in Fig. 4. In this way, we secured that the two groups talked the same language.

• Prepared students for the interviews: To assure the quality of the data collection process to be performed by the students, we prepared them in several consecutive steps. We first gave a lecture on the subject. We then had three seminars during which we presented the questionnaire and explained and motivated each of its questions. To ensure that the questions were not misunderstood by the students and the interviewees, we included guidelines for asking each question.

• Required personal data from the interviewees: To ensure the traceability of the interview results and the authenticity of the collected data, we requested that the interviewees provided their contact information.

• Scrutinized student results: We scrutinized all the answers provided by the students. In case of ambiguities in the answers, the students were asked for clarifications and complimentary additions.

By coming so far in our method description, our reader must have understood that our study was of a strong explorative character. Here, our main goal was to find out as much as possible about the scope of the predelivery maintenance process. The sampling method, however, does not allow us to generalize our results. Despite this, we strongly believe that the results presented herein strongly contribute to increasing the visibility into the predelivery maintenance process.

III. PREDELIVERY PROCESS

Maintenance is considered as a postdelivery phase. The evidence may be found in many of the standards today. For instance, standards such as [3, 4] define maintenance as a pure postdelivery phase. This indicates that no maintenance whatsoever is conducted before the delivery.

Protesting voices have been raised against treating maintenance as a pure postdelivery phase [2, 12, 15, 16]. These voices advocate that the organizations should prepare the system and maintenance organizations for postdelivery maintenance. The preparation activities should be performed long before the maintenance starts. They should be of a preparatory, controlling and evaluative character [5].

During the preparatory activities, the maintainer prepares himself for future maintenance by acquiring knowledge of the system and by establishing future maintenance organization [5, 6, 8, 9, 12].
During the controlling activities, the maintainer has insight into the development process and makes sure that maintainability is being built into a system [5, 6, 8, 9, 12]. The maintainer also participates in contract negotiations to ensure that maintainability is clearly stated in the contract.

Finally, during the evaluative activities, the maintainer takes part in the evaluation of the development process by taking into account the issue of system maintainability [12].

IV. EVALUATION CRITERIA

To reach our exploration goal, we have defined four evaluation criteria. These are (1) respondents and their organizations, (2) process usage and terminology, (3) placement within lifecycle, and (4) predelivery activities and roles involved. These criteria are materialized in a questionnaire shown in Fig. 3. Below, we briefly describe and motivate them.

A. Respondents and their organizations

We asked about the interviewees’ organizations and the roles they performed within these organizations. Using Questions 1-4 in Fig. 3, our primary objective was to assure that the right roles were interviewed. Our secondary objective was to understand the domain of the organizations studied and to ensure the credibility of the answers. Therefore, we asked about the interviewees’ name, email, contact telephone number and their role within the organization. We also asked about some basic information about the organization.

B. Process usage and terminology

The most important prerequisite for including the company in our study is to make sure that it conducts predelivery maintenance. For this reason, we inquired whether the company conducted it (see Question 5 in Fig. 3). Our objective was to make sure that we interviewed the right organizations. Just because the nomenclature within software engineering strongly differs, we also asked about the term used for predelivery maintenance in the organizations studied (see Question 6).

C. Placement within lifecycle

We wished to find out where in the lifecycle the organizations placed predelivery and postdelivery maintenance. For this purpose, we presented three lifecycle roadmaps to our interviewees and asked them to place maintenance on them. Just because there are many different ways of presenting lifecycle, we presented three different roadmaps. These are presented in Fig. 4. In this way, we ensured that our interviewees at least pointed out one predelivery and one postdelivery maintenance in at least one of the roadmaps.

The roadmaps show different perspectives of software lifecycle. These are (1) product perspective illustrating product transfer between major and minor revisions (see Question 7) [6] (2) process phase perspective visualizing the transfer between the phases in the whole software lifecycle (see Question 8) [13] and (3) development cycle perspective visualizing the phases within a software development cycle (see Question 9).
D. Predelivery activities and roles

To find out who is involved in doing what in the predelivery maintenance, we asked the interviewees about which groups or organizations were involved and what they did in predelivery maintenance. We believe that this information is critical to understand the complexity of predelivery maintenance.

Regarding the groups and organizations, we primarily inquired whether developer, maintainer, and acquirer were involved. We also inquired whether any other groups were involved apart from the already-mentioned groups (see Question 10). We then continued exploring what activities were performed by the groups or organizations (see Question 11). Finally, we elicited the roles specifically dedicated to predelivery maintenance process for all the above-mentioned groups or organizations (see Question 12).

V. INTERVIEW RESULTS

In this section, we present the results of our interviews. When doing it, we follow the order as defined in our questionnaire.

A. Process usage and terminology

All the organizations studied prepare their systems for future changes and they prepare maintainers or maintenance organizations for future maintenance. Six of the organizations studied practice the “cradle-to-grave” approach implying that maintenance stays with development. Despite this, their developers still need to be trained for future maintenance. This is because the maintenance responsibilities strongly differ from the development responsibilities.

Four out of eleven organizations do not use any specific term for predelivery maintenance. According to them, predelivery maintenance activities correspond to the fulfillment of various process requirements that must be met in order to create a deliverable product. Regarding the remaining organizations, they use terms such as preventive maintenance, maintenance preparation, impact analysis, good design, and pre-launch support.

B. Placement within lifecycle

Our interviewees were requested to place predelivery and postdelivery maintenance on three lifecycle roadmaps in Fig. 4. They were not required to use all the roadmaps, only the ones that were suitable for the context within their respective organizations. However, most of our interviewees used more than one roadmap. Below, we present the interview results for each of the roadmaps.

1) Product lifecycle perspective

Regarding predelivery maintenance, all the organizations studied (eleven organizations) perform predelivery maintenance in parallel with the development of the first version. As illustrated by Roadmap 1 in Fig. 5, eight of them only conduct it during the development of the first version. In this case, predelivery maintenance ends once Version 1 is delivered. Regarding the remaining organizations (three organizations), they perform predelivery maintenance in parallel with the development of each major version.

Concerning postdelivery maintenance, six out of all the organizations studied place postdelivery maintenance within the development phase of each major release. This means that during the development of Version 2, the postdelivery maintenance of Version 1 is performed and so on. Three another organizations include even creation of minor revisions as postdelivery maintenance. Finally, one organization using RUP performs postdelivery maintenance after delivering the first iteration in the Elaboration phase.

2) Process lifecycle perspective

All the organizations studied place predelivery maintenance in the initial development phase (see Roadmap 2 in Fig. 5). However, two of these organizations consider both the development and evolution phases as part of predelivery maintenance. According to them, activities inherent in predelivery maintenance are part of good design. Hence, they should be continuously carried out throughout the whole development and evolution process.

All the organizations studied place postdelivery maintenance after initial development during the evolution phase. According to them, postdelivery maintenance starts once the first running version is approved by the acquirer. Three of these organizations call the postdelivery maintenance as the servicing phase.

3) Development cycle perspective

Six organizations place predelivery maintenance within the implementation and unit testing phase (see Fig. 5). One of these six organizations continues with predelivery maintenance during the integration and system testing.
Regarding the remaining two organizations, they consider predelivery as part of all the phases after the requirements definition phase, whereas one of these two organizations starts predelivery maintenance during system and software design and continues it during implementation and unit testing.

All the organizations studied place postdelivery maintenance after the integration and system testing phase. Postdelivery maintenance starts once the system goes through the integration and system testing phase and is delivered to the acquirer. At this point, maintenance team takes responsibility for communicating with the acquirer and for handling change requests.

C. Predelivery activities and roles

In this section, we present the roles and activities involved in predelivery maintenance on the developer, maintainer and acquirer’s side. We also list other roles and responsibilities. Our results are summarized in Fig. 6.

1) Developer’s activities and roles

The majority of the organizations studied (10 out of 11) involve developers during predelivery maintenance. Regarding the remaining organization, it has not provided any answer.

All the organizations studied have identified the activities conducted by the developer in predelivery maintenance. Fig. 6 lists these activities. They cover the whole development cycle starting from cost estimation and schedule planning to integration and system testing. The Developer defines maintainability requirements, follows coding standards, develops system documentation and deploys the system.


2) Maintainer’s activities and roles

As shown in Fig. 6, maintainer mainly designs test cases, checks development results and plans for future maintenance.

![Table of activities and roles](image)

**Figure 6. Roles involved and activities conducted in the predelivery maintenance phase.**
All the organizations identified activities performed by the maintainer during predelivery maintenance. Just as the developer, the maintainer is involved in activities spanning almost the whole development cycle starting at defining maintainability requirements to conducting stress testing and setting up acquirer environment. Moreover, the maintainer gains system knowledge and communicates with the customer. His activities are listed in Fig. 6.

The organizations studied have identified eight various roles on the maintainer side. These include (1) Maintenance Coordinator, (2) Support Engineer, (3) Software Engineer, (4) Test Engineer, (5) System Operator, (6) Quality Assurance, (7) Technical Writer and (8) Trainer.

3) Acquirer’s activities and roles

Even the Acquirer participates in predelivery maintenance in the majority of the organizations studied (8 out of 11). As shown in Fig. 6, the acquirer participates in contract signoff, design approval, acceptance testing and training sessions to understand the system. His activities range from sharing business knowledge with the developer to system acceptance testing and to attending training sessions. He provides feedback on system design and notifies missing requirements during testing and deployment phase.

The studied organizations identified four roles on the acquirer side. These include (1) Main User, (2) Product Owner, (3) Product Manager and (4) Client.

4) Other roles activities

We requested our interviewees to mention activities conducted by other roles during predelivery maintenance. The majority of the organizations studied (10 out of 11) provided feedback on this issue. Fig. 6 lists the roles and activities. The roles mentioned are (1) Architect, (2) System Team, (3) Operation Team, (4) Business Analyst, (5) Product Manager, (6) Project Manager, (7) Database Administrator, (8) Software Configuration Management, and (9) Software Quality Assurance Team.

VI. CONCLUSIONS

In this paper, we have mapped out the scope of predelivery maintenance within eleven companies. Our results show that all the companies conduct some form of predelivery maintenance. However, they do not have any specific name for it. They use other names such as preventive maintenance, maintenance preparation, impact analysis, good design, and pre-launch support.

Our study has resulted in a wide range of predelivery maintenance activities and roles performing them on the developer, maintainer and acquirer sides (see Fig. 6). This list only constitutes a subset of all predelivery maintenance roles and activities. It is definitely not exhaustive.

When analyzing the activities elicited in this study, we may clearly see that some of the roles and activities overlap. We interpret it that many of the predelivery maintenance activities are conducted by the same roles, however, in different organizational and process contexts.

By looking at the elicited activities, we conclude that predelivery maintenance is a process strongly intersecting development and handover processes. We illustrate this with Venn Diagram in Fig. 7. As pointed out by our interviewees, many of the predelivery maintenance activities are part of pure development or they overlap with the handover process [11]. Example of a clear intersection among these processes is for instance involvement in a testing process. This observation agrees with the statement as provided by Pigoski, in which he reports on similar intersections [12].

The intersection in the Venn diagram stands for all the activities that are common to development, handover and predelivery maintenance. The difference in the Venn diagram, on the other hand, includes pure predelivery maintenance activities such as development of a maintenance plan and control that maintainability is being build in into a system during development.
Our results also show that the span of predelivery process is very wide. We illustrate it by placing some of the elicited predelivery maintenance activities on Roadmap 3 in Fig. 8. As shown there, many of the predelivery activities originate in the requirement definition phase and end right after the system testing phase.

Our study has made us realize that visualizing the predelivery process on only development cycle roadmap is not enough. We should have also used another roadmap depicting business cycle perspective. However, the data elicited in this study is sufficient for illustrating the process from the business cycle perspective using the business cycle roadmap taken from [10].

The complexity of predelivery maintenance is evident from its activities spanning the whole business cycle. As illustrated in Fig 9, it originates in the identification of business opportunity phase and continues throughout tender creation, contract negotiation, implementation and ends upon delivery.

Our study reveals that a predelivery maintenance process strongly varies depending on the process and organizational context. As illustrated in Fig. 10, we have identified five different scope contexts.

- **Context 1**: Predelivery maintenance is only placed within initial development. All other lifecycle process phases correspond to postdelivery maintenance.
- **Context 2**: Development of all major versions corresponds to predelivery maintenance for future software versions. However, they also constitute postdelivery maintenance phase for earlier maintenance phases. Postdelivery maintenance also includes all minor revisions.
- **Context 3**: Development of all major versions corresponds to predelivery maintenance for all minor versions. Postdelivery maintenance corresponds to the creation of minor revisions.
- **Context 4**: Initial development and evolution including all types of development and maintenance activities correspond to predelivery maintenance. As soon as one stops enhancing the software and only services it, one steps into the postdelivery phase.
- **Context 5**: All development activities till system testing are considered to belong to predelivery maintenance. All
activities taking place after system testing are considered to belong to postdelivery maintenance.

Our study shows evidence that there are different interpretations and contexts of predelivery maintenance. This, in turn, implies that the predelivery maintenance standard must be revised from the perspective of the industrial predelivery maintenance contexts. As it is right now, it applies to only one out of five contexts as identified in this study, which is Context 1.

Finally, our study indicates that the predelivery maintenance activities intersect with the activities in the development and handover processes. Right now, the software community has a vague understanding of this intersection. We sincerely hope that by conducting this study, we have provided more visibility into predelivery maintenance and its relationship to other processes. More work, however, need to be done to increase this visibility.

REFERENCES