# EM$^3$: Software Retirement Process Model

Mira Kajko-Mattsson
School of Communication and Information Technology
KTH Royal Institute of Technology
Sweden
mekm2@kth.se

Anna Hauzenberger, Ralf Fredriksson
Department of Computer and Systems Sciences
Stockholm University
Sweden
anna-hau@fc.dsv.su.se, rafr1976@student.su.se

***Abstract*—The software community has been so much focused on creating and improving development and evolution processes, so that it has completely forgotten retirement. Today, there are no retirement process models whatsoever despite the fact that many software organizations desperately need guidelines for retiring their software systems. In this paper, we elicit theory about software retirement process and put it into a software retirement process model, which we call *EM$^3$: Software Retirement Process Model*. The elicitation has been done within "If…", a Nordic insurance company. The model is based on two comprehensive case studies conducted within two real-life retirement projects.**

***Keywords-case studies; software lifecycle; software migration; software phaseout; software closedown; software disposal.***

## I. INTRODUCTION

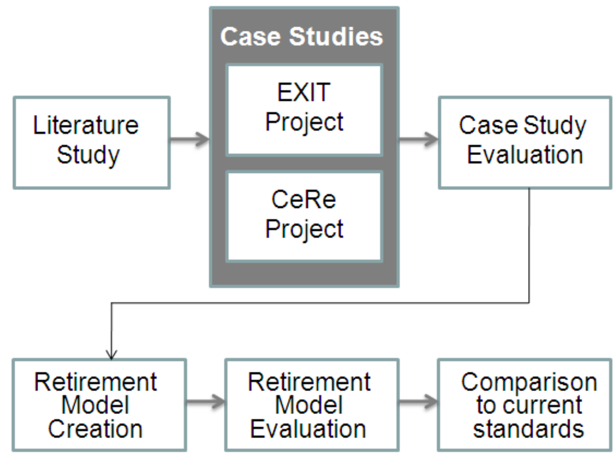
Research on software lifecycle process models has not been well balanced so far. Most of the attention has been paid to software development. Less focus has been put on software maintenance. No research been made on software retirement whatsoever.

Retirement is the disposal process whose aim is to end the existence of a software system [11]. It consists of the actual software system phaseout, removal of it from a regular usage, migration of its still relevant parts to some other system(s), closedown, and the archiving of it [1].

There are plenty of reasons why a system needs to be retired. Some of them are the system age and complexity, removal of its software and/or hardware platform, rules embodied by the external environments, and the like. Irrespective of the underlying reasons, retirement is an extremely complex and difficult process. Hence, it must be carefully planned and performed.

Right now, the concept of retirement is not well established within software engineering [5]. Neither are there any process models describing it. There are only very few standards and these standards are not based on any real-life studies [2][3]. Their contents has been mainly chosen in ballots; hence, they are very general. At its most, they cover a whole retirement process model within only a few pages. Hence, they do not provide sufficient guidelines for the organizations in their complex retirement work.

In this paper, we outline a retirement process model, called *EM$^3$: Software Retirement Process Model*. The model is part of *EM$^3$* standing for ***E****volution and* ***M****aintenance* ***M****anagement* ***M****odel*. The model has been created within "*If…*", one of the leading property and casualty insurance companies in the Nordic region. This company has recently undergone nine retirement projects. We have studied two of them: (1) the EXIT project performed in Sweden [8] and (2) the CeRe project performed in Finland [9]. Our goal is to provide a basis for creating theory in the domain of software retirement, to evaluate current process standards and provide feedback for their extension.




In the second case study, we explored Steps taken in our research process.

The two projects studied, differed in their prerequisites and process designs. For this reason, we made two separate case studies and put them into their respective process models [8, 9]. In this paper, we first present the two models and consolidate these them into one general model which we then evaluate within "*If…*".

The remainder of this paper is as follows. Section 2 describes our research method. Sections 3 and 4 present the EXIT and CeRe projects. Section 6 evaluates and compares the consolidated process model to the existing retirement standards. Finally, Section 7 makes conclusions and suggestions for future work

## II. RESEACH METHOD

Our study was a typical design research [10]. Its goal was to explore and model the domain of retirement by identifying all its relevant process constituents and the relationships among them. As illustrated in Figure 1, our exploration work consisted of several consecutive phases.

 ISBN: 978-1-61208-367-4 

TABLE I. THE EVALUATION QUESTIONNAIRE

1. Have you participated in other retirement projects, other then EXIT and CeRe?
1. What was your role within the project(s)?
3. What problems have been encountered within these projects?
4. What is your opinion about our retirement process model?
   a) Does it include the right phases?
   b) Is there any phase missing?
5. For each of the phases:
   a) Are the activities appropriate for the phase?
   b) Are there any activities missing?
   c) Do the right roles perform the activities?
   d) Can you see any problems with this phase and/or its activities?
6. How can the model solve the problems you have encountered?
7. What is good and what is bad about the model?
8. Do you have any suggestions for improvements?

In the first phase, the *Literature Study* phase, we made an extensive and comprehensive literature study. We went through various articles and standard process models touching on retirement. None of them, however, provided us with detailed information about the process. Only [2][3] outlined very general models. Due to their very coarse-grained nature, they did not provide any sufficient platform for starting our work. Hence, we may claim that our results are entirely elicited from scratch using the industrial support.

In the second and third study phases, the *Case Studies* and *Case Study Evaluation* phases, we studied both the EXIT and CeRe projects and evaluated them within the company [8][9].

Regarding the EXIT project, we studied it by first scrutinizing all the relevant project documentation. This documentation included about 100 different documents describing the retirement project, project plans, status reports, activity lists, system overviews, reports from various meetings such as steering groups, reference groups, and the like.

In the study of the CeRe project, our first step was to interview the CeRe project leader who presented the overall retirement process to us. We then continued to scrutinize relevant project documentation. This documentation included about 30 various documents.

Due to the fact that CeRe was a Finnish project, all the documentation was written in either English or Finnish. The documents written in Finnish were translated to us to Swedish by the CeRe project leader, either orally or in written.

In both the EXIT and CeRe projects, the documents studied did not fully describe the whole retirement project. Hence, we had to complement our explorative study with a series of interviews with the project leader and one operation manager.

Based on the understanding gained, we created two preliminary retirement process models for each of the retirement project studied [8, 9]. These models outlined a set of process activities in the EXIT and CeRe projects, structured these activities into process phases and identified roles involved in them. They were then presented to the project managers. The goal was to evaluate their credibility

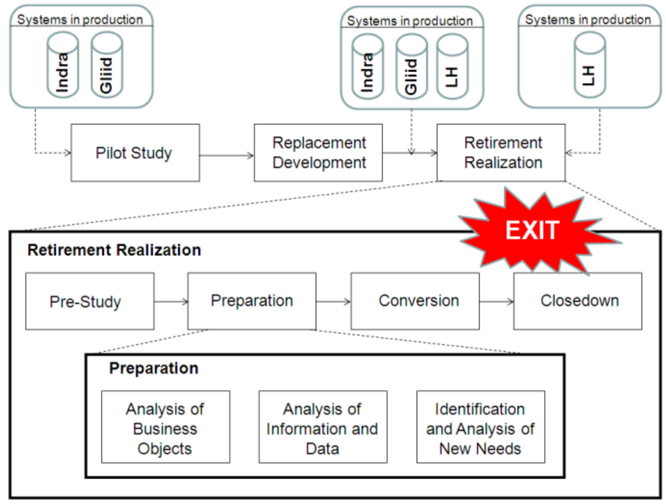



Figure 1. Phases in the EXIT project.

and adherence to the EXIT and CeRe projects, respectively. The evaluation step resulted in some minor modifications to the process models. These modifications are presented in Section 6.

The process models of the EXIT and CeRe projects covered various aspects of retirement. Hence, they differed in their prerequisites and designs. As a next step, we consolidated them into one general process model, which we call *$EM^3$: Software Retirement Process Model*. We then evaluated the $EM^3$ model within the company using tête-à-tête interviews. The questionnaire used for the evaluation purpose is presented in Table I.

Six people were involved in evaluating our retirement process model. Two of them were retirement project managers, one decision maker, one system analyst and maintainer, one developer and one business manager. These people were involved in at least one retirement project.

Finally, we compared our model to the standard models [2][3]. To enable the comparison, we created a set of comparison criteria. These criteria are listed in Table V. Due to the fact that the standard process models studied are very general, we could only define our comparison criteria on a very general level.

## III. THE EXIT PROJECT, CASE STUDY 1

In this section, we present the EXIT project. We first present its context in Section 3.1. We then describe the project in Section 3.2.

### A. *Context of the EXIT Project*

Two legacy systems, *Indra* and *Gliid*, were going to be retired and replaced with a system called *LH*. As illustrated in Figure 2, the overall retirement process consisted of three phases. These were (1) *Pilot Study*, (2) *Replacement Implementation*, and (3) *Retirement Realization*.

In the first phase lasting for one year, *"If..."* made a pilot study during which they examined *Indra* and *Gliid* and decided that a replacement system, *LH*, would be developed and *Indra* and *Gliid* would be retired.


ISBN: 978-1-61208-367-4

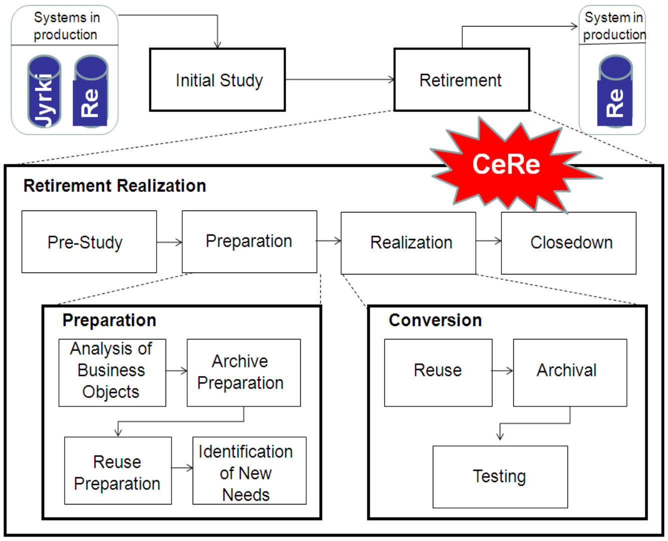



Figure 2. Phases in the CeRe project.

During the second phase, lasting for two years, "*If…*" was in the process of developing *LH*. In the next-coming two years, it started the retirement of *Indra* and *Gliid*. In the fifth year, both *Indra* and *Gliid* were closed down and only *LH* has been used since then. The star banner in Figure 2 marks the scope of the EXIT project.

## B. *The EXIT Project*

The EXIT project consisted of four phases. They are (1) *Pre-Study* (2) *Preparation*, (3) *Conversion*, and (4) *Closedown*. Below, we briefly describe them.

*1) Pre-Study:* The goal of the *Pre-study* phase was to investigate the systems to be retired, determine which of their parts should be migrated and disposed off, identify appropriate archiving and migration strategies, define a retirement project and plan for it.

In this phase, one first investigated the types and volume of business objects to be retired and migrated. One then determined the archiving and migration needs to be further used for identifying the appropriate migration and archiving strategies. As a next step, one determined the project scope. When doing it, one first analyzed *Indra* and *Gliid*'s overall architecture and design and then identified dependencies to other interfacing systems. Here, one considered other systems and their users that were dependent on the retiring systems.

Identification of the interfacing systems affected by the closure of *Indra* and *Gliid* led to the identification of the additional activities required for managing the retirement project. In our case, one recognized a need (1) for analyzing the migration and archiving strategies, and (2) for making deeper analysis of adjacent systems and their connections to the systems to be retired.

Finally, one defined a retirement project. The project definition included risk management and creation of a retirement plan. Risk management concerned risks such as access to resources required, staff illness and various technical risks [13]. The retirement plan, on the other hand, covered most of the rudimentary project planning activities.

*2) Preparation:* The goal of the *Preparation* phase was to further analyze the systems to be retired, make a decision on archiving and migration strategies, determine changes to be made in the adjacent systems and in the replacing system.

As a first step, one studied the business objects to be migrated. The goal was to identify active objects and to attend to the inconsistencies in them. An example of an active business object is a car insurance.

For all the active business objects, one analyzed their individual data fields in order to determine whether they should be migrated to the new system. One also analyzed special cases. An example of a special case is when one and the same business object is administered by both systems, namely, the retiring and the replacing systems.

For the data fields to be migrated, one created a conversion table and a conversion testing plan. Testing implied that one chose a specific numeric field, summed it for all the business object instances to be migrated and compared their sum to the corresponding sum in the new system.

*3) Conversion:* As a first step in the Conversion phase, one developed the automatic conversion method including scripts and the automation process. This method was then tested. The purpose was to estimate conversion time and to assure a problem free conversion. When the tests were successfully passed, one conducted both the automatic and manual conversion. The conversion results were finally tested to verify that the conversion was successful.

*4) Closedown:* Finally, in the *Closedown* phase, one closed down the *Indra* and *Gliid* systems and removed their dependencies to the adjacent systems.

# IV. The CeRe Project, Case Study 2

In this section, we present the CeRe project. In Section 4.1, we first present its context. In Section 4.2, we describe the CeRe project itself.

## A. *Context*

In the second case study, we explored the process of retiring a system called Jyrki. Jyrki was internally developed to be used for managing debts and credits. It had about 35 users. At the beginning of the CeRe project, it was 18 years old. Together with eight other systems, it was installed on HP 3000. In the first year, HP announced that HP 3000 would be phased-out in the future five years. For this reason, "*If…*" decided to retire all the eight systems installed on this platform.

Out of the eight systems, we focused our study on Jyrki due to the following reasons: (1) it was the last system on the HP 3000 platform to retire; hence, the project results were fairly fresh, (2) having it as the last retirement project, "*If…*" had matured with respect to its retirement management; hence, the project provided us with feedback on a matured retirement process; (3) almost all the project documentation

TABLE II. RETIREMENT CHECKLIST

1. Identify the business criticality of the system and the need to use it.
2. Analyze system structure and its size.
3. Identify system users and their needs.
4. Estimate the system lifecycle lenght.
5. Identify adjacent systems and assess impact on them.
6. Identify all contracts applying to the retiring system.
7. Identify other systems possessing similar functionality and/or overlapping functionality.
8. Identify and study candidate environemtns to/in which the retiring system might be migrated/reused.
9. Identify the laws and rules to be considered.
10. Identify the need to reuse the system and/or its parts.
    1. Identify the business objects/functionality/data to be reused.
    2. Determine their volume.
11. Identify the need to destroy the system and/or its parts.
    1. Identify the business objects/functionality/data to be removed.
    2. Determine their volume.
12. Identify the need to archive the system and/or its parts.
    1. Identify the business objects/functionality/data to be archived.
    2. Determine their volume.
    3. Determine the archive's accessability.
    4. Determine the archive's size.

was in English; hence, we could easily follow it, and (4) many of the people involved in the retirement of Jyrki were still available; hence, they could help us in this study.

In the second year, "*If…*" decided that relevant parts of Jyrki would be migrated to an existing system called RE. RE was a standard bought-in system installed on another platform. The retirement work itself lasted for exactly one year after the decision was made to retire it.

### B. *The CeRe Retirement project*

As illustrated in Figure 3, the CeRe project consisted of two main phases. They are (1) *Initial Study* and (2) *Retirement*.

*1) Initial Study:* The Initial Study phase took place in the second year. Here, one made an inventory of all the systems installed on HP 3000 and evaluated each of them. Using the checklist presented in Table II, one identified each system's criticality, analyzed its structure, users, contracts, and adjacent systems. One then studied the laws and rules to be obeyed in the process.

The analysis conducted in this phase was very general. Its goal was to provide a basis for planning future retirement work and for determining the order of retiring the systems. Later on, this analysis would be repeated for each of the retiring system.

*2) Retirement*

Regarding the second phase, the *Retirement* phase, it consisted of four sub-phases: (1) *Pre-Study* (2) *Preparation*, (3) *Realization*, and (4) *Closedown*.

*a) Pre-Study:* The goal of the *Pre-study* phase was to investigate the system to be retired, to determine which of its parts should be migrated, disposed off and archived, to identify appropriate archiving and reuse strategies, to define a retirement project and plan for it.

When investigating the CeRe project, one used the same checklist as in the *Initial Study* phase (see Table II). The goal was to find out whether its results were still relevant. This investigation was then complemented with an additional study, this time focused on archiving problems. It was led by a series of questions that are listed in Table III.

TABLE III. QUESTIONS DEALING WITH ARCHIVING PROBLEMS

1. How critical is the retiring system to the business?
2. How many users does the retiring system still have?
3. Are there any needs to migrate the retiring system's data to other systems?
4. What types of business objects are managed by the retiring system now?
5. Who generates reports and statistics in the organization and for what purpose?
6. Should the statistics and reports contain historical information?
7. How far back in time should the historical accounts stretch themselves?
8. What coupling does the retiring system have to the adjacent systems?
9. Does the retiring system need an archive?
10. What laws and requirements generate the need for an archive?
11. What are the accessibility requirements of an archive?
12. How does the retiring system influence the adjacent systems?

The *Pre-study* phase resulted in an updated plan of the continued work. The plan covered (1) the specification of the roles and activities required for conducting the work, (2) specification of the business objects to be considered, and (3) the identification of the overall strategies required for reusing and archiving the business objects. In addition to the basic strategic issues, the reuse strategy focused on confirming that RE still constituted an appropriate platform for migrating some parts from *Jyrki*. The archiving strategy, on the other hand, focused on designating the technical solution of the future archive. It was decided that Microsoft Access would be used.

*b) Preparation:* The *Preparation* phase encompassed a number of analyses on various levels, from business objects down to the data field level. The goal was to determine which business objects should be reused and archived and to determine the migration impact on the RE system.

As illustrated in Figure 3, the *Preparation* phase consisted of four sub-phases: (1) *Analysis of Business Objects*, (2) *Archive Preparation*, (3) *Reuse Preparation,* and (4) *Identification of New Needs.*

In the *Analysis of Business Objects* phase, one analyzed which of the business objects should be reused and archived. Here, one decided that only active objects, such as unpaid invoices, were to be reused. The objects needed for future retrieval should be archived. The rest should be disposed off. One also decided that all the reused instances should be easily traceable both in the archive and in the *RE* system. With this, one expected to have control over the migrated business objects.

For each type of a business object, one then analyzed its instances to make sure that the right ones got migrated to the *RE* system. Here, one generated lists of all the active business object instances. One then analyzed them to confirm that they had the right status. Finally, one flagged all the reused instances to make them traceable.

In the *Archive Preparation* phase, one specified requirements on the archive, identified business objects to be archived and procedures for migrating data to the archive. As a first step, one analyzed the objects on a data field level to determine which of the fields should be archived and how they should be retrieved.

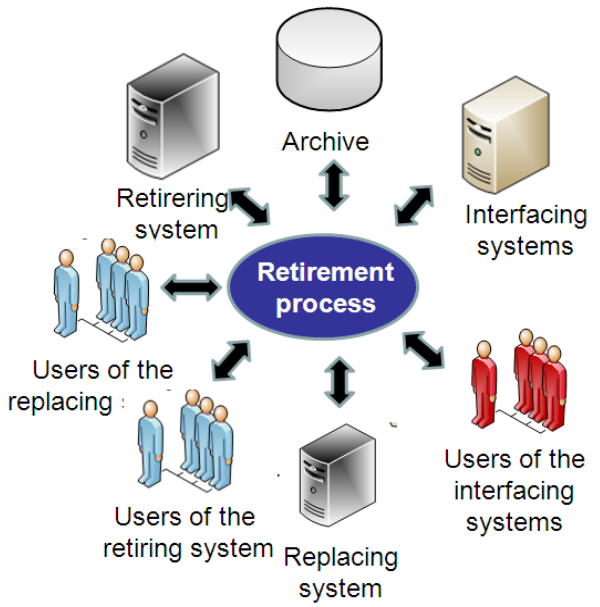



Figure 3. Components in *EM*$^3$*: Software Retirement Process Model*.

As a next step, one developed a simple archive prototype. The purpose was to verify that the final archive would fulfill the organizational requirements. One then tested it and solved all the problems encountered in it. Finally, one made a decision on how to test the final archive after it got implemented.

The goal with the *Reuse Preparation* phase was to further detail the reuse strategy, specify the conversion process, revise the conversion requirements, determine testing procedures, and the like. As a first step, one established which types of business objects should be reused. For each type, one analyzed its individual fields and decided on whether they should be reused or not.

It is not easy however to reuse data fields in another system. The declarations may substantially vary. To ensure the quality of the reuse, one mapped the data fields in *Jyrki* to the data fields in *RE*. For each of them, one then determined a conversion approach, either manual or automatic.

As a next step, one specified the order in which the business objects should be converted. The order was influenced by the dependencies among the objects. For instance, customer objects should be converted first before converting their insurances. One then determined the conversion testing method. The method implied that one chose a specific numeric field, summed it for all the instances in Jyrki and compared their sums to the corresponding sum in RE.

In the *Identification of New Needs* phase, one studied whether the data migrated from *Jyrki* would affect *RE*. For this purpose, one investigated whether the working routines would have to be changed. This investigation resulted in the identification of new requests for changes to be made in *RE*. These changes were then implemented and tested. Finally, one updated the *Retirement Plan*.

*3) Realization:* The Realization phase consisted of the following sub-phases: *Reuse*, *Archival*, and *Testing*.

In the *Reuse* sub-phase, one first defined a conversion method. One did it for both the manual and automatic conversions. The goal was to secure that the conversion would be conducted in the right order and that nothing would be forgotten.

Regarding the manual conversion, one created a crib supporting the manual work. For the automatic conversion, one created a list identifying the automatic procedures, specifying the data to be converted and their order.

As a next step, one implemented the automatic procedures. Due to the fact that the *RE* system already implemented the automatic conversion procedure, one did not need to implement it. What one only needed was to implement procedures accessing *Jyrki's* data.

One then implemented and tested the conversion method. When testing the manual method, one converted some instances following instructions as specified in the crib. One then controlled the results. Possible problems in the manual conversion procedures were then attended to and tested anew.

When testing the automatic conversion method, one downloaded the data into the *RE*'s testing environment. One then verified the results. In case of problems, one solved them and tested the automatic procedures anew. Before starting the conversion, however, one made sure that all the preparations had been made correctly. For instance, one checked whether all the required changes had been done to *RE*. Finally, one migrated (converted) data to the RE system. One conducted the manual conversion first. Both conversions were then tested and approved.

The goal of the *Archival* phase was to create an archive, migrate data to it and test. Using the stated requirements, one started the development of the archive. One then developed the automatic procedures to transfer data from *Jyrki* to the archive. To be able to present the data in the archive, one needed reports. About ten reports corresponding to the most frequent searches in *Jyrki* were developed.

The migration of data to the new archive was tested using a sample data first. While doing it, one created a user manual and educational material. One then educated and trained its users. Finally, one conducted the entire migration to the archive. The migration was entirely automatic.

After the migration was fulfilled, one tested its results in the last *Testing* sub-phase. One did it to secure the migration correctness by comparing the data in *Jyrki* and the new archive, using similar tests as in the *Reuse* sub-phase.

*a) Close down:* Before one conducted the final conversion, one removed the opportunities to update *Jyrki*. However, one waited for two months before disposing off *Jyrki* and its hardware platform. This time period was a security measure during which the users could attend to the inconsistencies observed in *Jyrki*, *RE*, and the archive.


ISBN: 978-1-61208-367-4

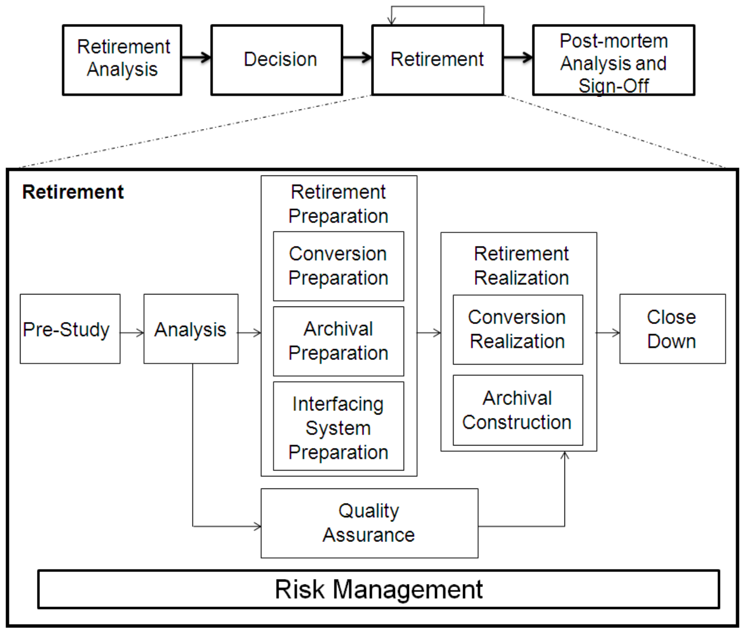



Figure 4. Design of $EM^3$: Software Retirement Process Model.

## V. RETIREMENT PROCESS MODEL

In this section, we outline *$EM^3$: Software Retirement Process Model*. We first provide its overview in Section V.A. We then describe the retirement phases and roles involved in them in Sections V.A and V.B, respectively.

### A. Process Model Overview

As illustrated in Figure 4, *$EM^3$: Software Retirement Process Model* manages components such as (1) *Retiring system(s)*, (2) *Replacing system(s)*, (3) *Interfacing system(s)*, (4) *Users of the retiring system(s)*, (5) *Users of the replacing system(s)*, (6) *Users of the interfacing system(s)*, and (7) *Archive*. Our suggestion for a retirement process model is depicted in Figure 5. It consists of four main phases:

1. *Retirement Analysis*: In this phase, one analyzes the retiring system using either the checklist presented in Table II or a decision matrix [4]. This activity is usually initiated due to many reasons. Some of them are (1) high maintenance cost, (2) removal of the software or hardware platform of the retiring system, (3) duplicated functionality in several systems [1].
2. *Decision*: In this phase, one decides whether the system should continue to provide service or whether it should be disposed off.
3. *Retirement*: If the decision has been made that the system is to be retired, then the system undergoes a retirement process.
4. *Post-mortem Analysis and Sign-Off*: After the retirement has been realized, one analyzes the process, assures that all the planned activities have been performed as planned and that all the goals have been achieved, one collects lessons learned, and finally, one signs off the retirement process.

### B. Retirement Process Roles

The *$EM^3$: Software Retirement Process Model* retirement process involves the following roles:

- *Decision Maker (DM):* set of managerial roles responsible for planning and managing the retirement process.
- *Maintenance Organization (MO):* organization responsible for maintaining the archive.
- *Operations Expert (OE):* role possessing expert knowledge of the system to be retired and the retirement process to be conducted.
- *System Manager (SM): role* responsible for the operation and maintenance of the system.
- *System Analyst (SA):* a role responsible for planning and analyzing the system to be retired.
- *Project Leader (PL):* role responsible for the retirement project.
- *System Architect (SAR):* role is responsible for knowing the overall architecture of the systems to be retired. This is a new role added to our model after the industrial evaluation step.
- *User (U):* role using the system to be retired.
- *Developer (D):* role involved in the implementation of the retirement process.
- *Support Technician (ST):* role responsible for operation and support of the system to be retired.

### C. Retirement Process Phases

The retirement process consists of five phases (1) *Pre-Study*, (2) *Analysis*, (3) *Retirement Preparation*, (4) *Retirement Realization*, and (5) *Close down*. Below, we describe each of them. As can be seen in Table IV, the total

TABLE IV. PHASES AND ACTIVITIES IN *EM³: SOFTWARE RETIREMENT PROCESS MODEL*. THE UNDERLINED ACTIVITIES WRITTEN IN BOLD WERE ADDED AFTER THE MODEL EVALUATON. TE ABBREVIATIONS IN THE PARANTHESES IDENTIFY THE ROLES PERFORMING THEM

### Pre-Study

1. Analyze the system to be retired (SA, OE)
2. Study the analysis results (PL, DM, OE)
3. Identify the dependencies between the retiring system and its environment (interfacing systems)
4. Create a retirement plan (OE, SA)
   1. Identify/determine stakeholders to be involved in the retirement project (PL, DM)
   2. Identify/determine roles required for managing and executing the retirement process (PL)
   3. Determine the competence required for managing the retirement (DM, OE)
4. Plan for how archiving, reuse and closure (SA, OE)
5. **Manage risks**
   1. **Identify risks (PL, OE, DM, SA)**
   2. **Analyze risks (PL, OE, DM, SA)**
   3. **Make a decision on how to manage risks (PL, OE, DM, SA)**
6. Create a retirement stratey (DM, OE, SA)
7. Create an archiving strategy (DM, OE, SA)
8. Determine mistones and dates (DM, PL)
9. **Determine budget (PL, DM)**
10. ….

### Analysis

1. Analyze the business objects to be migrated/archived. For each type:
   1. Identify the business objects to be migrated/archived (U)
   2. Identify the business objects to be quality assured (DM, OE, SA)
2. Analyze the business objects to be migrated
   1. Identify special cases of business objects (OE, SA)
   2. For each type of business object
      1. Study existing conversion techniques (automatic/manual) (DM, OE, SA)
      2. Analyze its individual data fields in the retiring system (OE, SA)
      3. Analyze its individual data fields in the replacing system (OE, SA)
      4. **Create conversion rules (OE, SA)**
      5. Create a conversion table between the retiring and replacing system (OE, SA)
      6. Create a conversion testing plan (OE)
      7. **Determine the order of converting the business objects (OE, SA)**
5. Analyze the business objects to be archived
   1. **Identify special cases of business objects (OE, SA)**
   2. **Study the archiving method (SA, D, MO)**
3. For each type of business object
   1. Analyze its individual data fields in the retiring system (OE, SA)
   2. Analyze its individual data fields in the archive (OE, SA)
   3. **Create conversion rules (OE, SA)**
   4. Create a conversion table between the retiring system and the archive (OE, SA)
   5. Create a conversion testing plan (OE)
   6. Analyze how the information is to be presented in the archive (OE, SA)
   7. **Determine the terminology to be used in the archive (OE)**
4. Develop the archive's prototype (SA, D)
5. Test the prototype (OE)
6. **Determine how the business object which are going to be neither migrated nor archived should be managed (DM, OE)**
7. Modify the dates in the plan, if needed (DM, OE, MO)
   1. Date when the retiring system is to be closed for update
   2. Date when the business objects should be migrated to the replacing system
   3. Date when the retiring system system should be retired

### Quality Assurance

1. Analyze the business objects and their instances.
   1. For each type:
      1. Determine how the business objects should be analyzed (OE, SA)
1. Analyze the instances of the business objects
   1. Assure that the instances are flagged in the archive (OE)
   2. Make necessary changed to the instances, i.e., update their status (U)

### Retirement Preparation

1. Prepare for the conversion
   1. Develop conversion process/procedures
      1. Describe how the conversion will be performed (both the manual and automatic conversion) (D, OE)
      2. Develop the extraction and loading processes/procedures for the automatic conversion (D)
   2. Test
      1. Test the manual conversion (OE)
      2. Test the extraction and loading processes (D)
      3. **Conduct functionality tests in the replacing system using the converted data (OE)**
2. Prepare for the archiving activity
   1. Develop the archive and the archiving process
      1. Develop the automatic procedures/procedures for accessing and storying data (D)
      2. Adapt the archive to the rules specified (D)
   2. **Determine the maintenance organization which will take over the archive (DM)**
   3. Test (D, OE)
      1. Test the automatic procedures
      2. Test the archive
   4. Create a user manual (OE)
   5. Educate and train
      1. Create educational material (OE)
      2. Conduct the education and training (OE, U)
3. Prepare the environment (interfacing systems)
   1. Analyze dependencies between the retiring system and the environment (SF, SA)
   2. Analyze the manual working routines (OE, SA)
   3. Analyze the consequences in the replacing system and the interfacing systems (OE, SA)
   4. Issue change requests and make changes in the replacing and interfacing systems, if required (DM, OE)
   5. Test the changes (DM, OE).

### Retirement Realization

1. Perform conversion
   1. Conduct the manual conversion (U)
   2. Conduct extraction and loading processes (D)
   3. Make sample tests in the retiring and the replacing systems (U, OE)
2. Perform archiving
   1. Conduct the archiving work (D)
   2. Make sample tests in the retiring system and the archiOE (U, OE)
   3. **Transfer the archive to the maintenance organization (PL)**

### Close Down

1. Remove the opportunity to use the retiring system (MO)
2. Close down and remove the automatic procedures (MO, DT)
3. Assure that all the planned activities have been conducted (PL, OE, DM)
4. Dispose off the retiring system (MO, DT)

process is structured into phases and activities. The goal is to create a reference framework mapping out what activities are relevant in what phase. However, the order of the activities as listed in our model does not impose any specific order of conducting them. Depending on the context at hand, these activities may or may not be selected. If selected, then they may be implemented in the order that is suitable for the context at hand.

**5.2.1. Pre-Study.** The *Pre-Study* phase starts only after one has made a decision that the system of concern is going to be retired. Here, one makes a comprehensive and detailed analysis of the retiring system. When doing it, one may use a checklist as presented in Table II, the same checklist that has been used in the more general *Retirement Analysis* phases. The goal is to get an understanding of the retiring system and to create an overall retirement plan.

**5.2.2. Analysis.** In this phase, one performs deeper analysis of the retiring system in order to get an understanding of the forthcoming retirement process. Here, one identifies the business objects to be managed and their underlying functionality. One then decides how they should be handled and one designates retirement project dates. One also decides on the quality levels for securing the management of the business objects.

**5.2.3. Quality Assurance.** This phase starts after one has determined which business objects should be quality assured. It runs in parallel with *Retirement Preparation* and partly with *Retirement Realization*. Here, one determines the rules for quality assurance and conducts the quality assurance.

**5.2.4. Retirement Preparation.** In this phase, one (1) prepares the system parts to be reused in the replacing system, (2) one prepares the system parts to be archived, and (3) one studies the impact of the conversion and retirement processes on the interfacing systems.

**5.2.5. Retirement Realization.** In this phase, one conducts the actual conversion and archival of the business objects and their underlying functionality.

**5.2.6. Close Down.** In this phase, the retiring system gets closed down and disposed off. Its data may be accessed only in its archive.

## VI. EVALUATION RESULTS

This section presents the results of evaluating our retirement model. Section 6.1 first presents the evaluation results of the EXIT and CeRe projects. Section 6.2 describes the evaluation results of our model.

### A. Evaluation of the EXIT and CeRe Projects

In the third phase of our study (see *Case Study Evaluation* in Figure 2), the models of the EXIT and CeRe processes were presented to the project managers responsible for the respective retirement project. According to them, our models were realistic and they fully reflected their retirement processes. They had, however, some minor deficiencies. These concerned lack of three important activities: *Activity 5 (Manage Risks)*, *Activity 9 (Determine budget)* in the *Pre-Study* phase, and *Activity 2.2.7 (Determine the order of converting the business objects)* in the *Analysis* phase. They also concerned lack of the role of *System Architect*.

According to our interviewees, risk management constitutes an essential activity within retirement. Not doing it implies a critical business risk by itself. Risk management should be run continuously throughout the whole retirement project. Due to the difficulties of integrating its activities with our retirement model, we only mark their start in the *Pre-Study* phase. However, in Figure 5, we place risk management as a parallel phase to the entire *Retirement* phase.

Regarding the second activity, the activity concerning the determination of retirement project budget, our interviewee from the EXIT project claimed that due to the project criticality, it is very important to assign substantial resources to the retirement project. Otherwise, one runs the risk of underestimating the project scope and thereby fails with its completion.

We admit that this planning activity is very important. When creating process models of the individual processes, we were mainly focused on identifying pure retirement activities. On purpose, we left out many activities typical of a traditional project planning. To remedy this, we have expressed the need for more project planning activities with three dots in *Activity 10*.

The third activity, *Activity 2.2.7 (Determine the order of converting the business objects)* in the *Analysis* phase, concerned the specification of the order of converting business objects. Some objects, should be converted first before converting the other objects. For instance, client objects should be converted before their insurance objects. As a response, we have added this activity to our model.

One role was claimed to be missing within the first evaluation phase. It concerned *System Architect*. According to both project managers, this role is indispensible in all the retirement projects. Not only does this role know the system to be retired but also all its architectural flaws and deficiencies that should not be migrated to the new system.

### B. Evaluation of $EM^3$: Software Retirement Process Model

In the fifth phase of our research (see *Retirement Model Evaluation* in Figure 2), the model was presented to six software professionals within *"If…"*. As already mentioned in Section 2, all of them were involved in at least one retirement project.

According to the "If…."'s software professionals, our retirement process model is realistic and appropriately mirrors the retirement process. They have however had some comments and suggestions for its improvement. Some of them have been attended to by complementing the model with additional activities. Those which could not be attended to immediately constitute our suggestions for future work.

The activities that have been added are:

- *Activity 6* in the *Analysis* phase (*Determine how the business objects which are going to be neither migrated nor archived should be managed*). It is important to analyze and make decisions on all the objects in the

 ISBN: 978-1-61208-367-4 

TABLE V. OUR COMPARISON RESULTS

| Activities | IEEE Std. 10741991 | ISO/IEC 15288 | EM³: Sw Retirement |
|---|---|---|---|
| - System Analysis | — | — | + |
| - Archiving Strategy | — | + | + |
| - Migration Strategy | — | — | + |
| - Mngt of Adjacent Systems | — | + | + |
| - Retirement Planning | + | + | + |
| - Risk Management | — | — | + |
| - Conduct Archival | + | + | + |
| - Designate Maintenance Org. | — | — | + |

retiring system. It is only then one may make sure that one has not omitted any business object.

- *Activity 2* in the *Retirement Preparation* phase *(Determine the maintenance organization which will take over the archive) and Activity 2.3* in the *Retirement Realization* phase *(Transfer the archive to the maintenance organization):* These two activities are very important. Our interviewees claim that the maintenance organization should be designated as soon as possible and that it should play the driving role within the retirement project.
- *Activity 3* in the *Close Down* phase *(Assure that all the planned activities have been conducted):* According to our interviewees, one needs an additional activity in the *Close Down* phase to assure that all the planned activities have been successfully implemented.

During the interviews, one issue was raised. It concerned information dissemination and documentation. Information dissemination has been regarded as a very important process activity. If not properly performed, it may lead to many problems. Regarding documentation, it is important that it is pervasive throughout the whole process. Our interviewees claim that all the process phases and activities should be thoroughly documented to assure that retirement gets implemented in a proper way. It is especially important that the conversion and archiving processes, and archive manuals are documented.

Our interviewees have also identified some problems within their respective retirement projects. The problems are:

- *Too little effort has been put into the analysis of the retiring system:* This has prolonged the retirement process due to the fact that additional work was required for repeating the analysis steps.
- *Lack of resources:* It is difficult to estimate the resources required for retiring the system. This is due to lack of retiring experience and too little effort put into the analysis activities.
- *Difficulties to man the retirement projects:* It is difficult to find individuals possessing the right competence for retiring software systems.
- *Retirement projects are not high priority projects:* Retirement projects are less prioritized than other projects. This in turn prolongs their duration.
- *Weak decision making*: Retirement is a very complex activity during which many important decisions are taken. They concern decisions whether to migrate, archive, or dispose off. Hence, key individuals must be assigned clear responsibilities to make decisions. They must also be members in the project and display interest and engagement in the retirement work. Lack of it may lead to the overall project delay.

Finally, our interviewees made a suggestion that one should wait with the physical disposal of the system for a while. In this way, one makes sure that no important activity or decision has been missed. If, for some reason, defects have been injected, one may still attend to them before it is too late.

### *C. Comparison to Standards*

In this section, we compare our retirement process model with the standard process models as described in IEEE STD 10741991 [2] and ISO/IEC 15288 [3]. When doing this, we follow the comparison criteria listed in Table V. Except for the criteria concerning the roles, all the comparison results are listed in Table V.

None of the standard process models suggests any roles to be involved in the retirement process. Only the *IEEE* model mentions a user role, who should be notified about the closure of the system. Our model however has identified ten different roles. These are listed and described in Section 5.2.

The broad portfolio of the roles identified in our model indicates that the retirement project involves the majority of the organizational roles ranging from user to various analyst and design roles, to managerial roles and even to front-end support roles [6]. This, in turn, indicates how complex and comprehensive the retirement process model is.

As illustrated in Table V, none of the standard process models include the activities during which one analyzes the retiring and the replacing systems. In accordance with the opinion of our interviewees, we believe that these are one of the most important activities within the retirement process. They could be compared to the requirements specification activities. It is a common knowledge that a non-recognition of the requirements, irrespective of what type of a project it concerns, does not lead to successful project results. For this reason, we claim that lack of analysis activities is a series deficiency in the standard process models studied.

Only the *ISO/IEC 15288* standard suggests identification of archiving strategies. None of the standards proposes migration strategy. In our opinion, identification of both these strategies is very important. Identification of the retirement strategy is a must. However, the identification of the migration strategy should be an option. This is due to the fact that not all retiring systems undergo migration. We believe, however, that the inclusion of this strategy in the retirement process model indicates that the retirement process does not exist in a vacuum. Many times, parts of the retiring systems have to be migrated to other new replacing systems or other new archiving systems.

Only the *ISO 15288* standard briefly mentions that the interfaces to the adjacent systems should be considered. None of the standard models suggests how the interfacing systems and their users should be handled. In our opinion, this is a serious omission. Improper management of the adjacent systems may lead to big inconsistencies and problems in their future operation. Hence, we suggest that the interfacing systems and their handling should be highly prioritized in a retirement process.

Both the standard process models studied include the planning activities. However, they only recognize the need for planning. They have not provided any suggestions specific to the retirement planning process.

None of the standard process models studied included risk management. We did not include it either in our preliminary process model outline. Even if risk management is a separate process, we strongly believe that it definitely should be integrated with the retirement process. Retirement and replacement imply many serious business risks. Not considering them may jeopardize the whole retirement process, and thereby, the organization's future business opportunities.

All the standard process models included the archival activity. This activity however was only briefly mentioned, even in our process model. We suspect that this activity is quite complex. Hence, it should be further scrutinized in the future.

Finally, none of the standards designates the maintenance organization responsible for driving the retirement project and for maintaining the archival. We believe that including maintenance organization right from the beginning helps avoid many future maintenance problems.

## VII. Final Remarks

In this paper, we have outlined a retirement process model. The model is called *$EM^3$: Software Retirement Process Model* and it is part of *$EM^3$*. It has been designed and evaluated within "*If…*", a company that has recently undergone nine retirement projects.

Except for a very few standards, there are no retirement process models whatsoever. Hence, we dare claim that our work is unique and innovative. Our results are entirely designed from scratch using the industrial support. Hence, this paper is one of the first reports on this very complex process. More work however needs to be made to both validate and elaborate on our process model. We, therefore, cordially invite the software community to help us with this very exciting project.

## References


[1] S. W. Ambler, M. J.Vizdos, and J. Nalbone, The Enterprise Unified Process :extending the Rational Unified Process Upper Saddle River, N.J. :: Prentice Hall PTR, 2005.

[2] IEEE Standard for Developing Software Life Cycle Processes, IEEE Std 10741991. 1991. The Institute of Electrical and Electronics Engineers, Inc.345 East 47th Street, New York, NY 10017-2394, USA, 1991.

[3] ISO/IEC 15288, Systems and Software Engineering – System life cycle processes, IEEE Std 15288-2008, 2002.

[4] I. Jacobson, F. Lindström, Reengineering of old systems to an object-oriented architecture. *SIGPLAN Not. 26*(11), 340–350, 1991.

[5] M. Kajko-Mattsson, "Common Concept Apparatus within Corrective Software Maintenance" International Conference on Software Maintenance, IEEE Computer Society Press: Los Alamitos, CA, Sep 1999, pp. 287-297, ISBN: 0-7695-0016-1, doi: 10.1109/ICSM.1999.792626.

[6] M. Kajko-Mattsson, L.-O.Tjerngren, A. Andersson, "$CM^3$: Up-front Maintenance", Conference on Software Engineering and Knowledge Engineering, Knowledge Systems Institute, 3420 Main Street, Skokie, IL, 60076, USA, 2001, pp. 371-378.

[7] M. Kajko-Mattsson, "Corrective Maintenance Maturity Model: Problem Management", PhD thesis, Department of Computer and Systems Sciences (DSV), Stockholm University and Royal Institute of Technology (KTH), 2001, ISBN Nr 91-7265-311-6, ISSN 1101-8526, ISRN SU-KTH/DSV/R--01/15.

[8] M. Kajko-Mattsson, R. Fredriksson, A. Hauzenberger, "Elicting a Retirement Process Model: Case Study 1" International Conference on Computer Science and Software Engineering, IEEE, 2008, doi: 10.1109/CSSE.2008.1364 .

[9] M. Kajko-Mattsson, R. Fredriksson, A. Hauzenberger, "Elicting a Retirement Process Model: Case Study 2", International Conference on Innovation in Software Engineering, IEEE, 2008, doi: 10.1109/CIMCA.2008.94.

[10] B. Laurel, Design Research: Methods and Perspectives, the MIT Press, 2003.

[11] V. T. Rajlich, K. H. Bennett, A staged model for the software life cycle. *Computer 33*(7) 2000.

[12] I. Sommerville, *Software Engineering* (8th ed.). Addison-Wesley, 2007.

[13] M. Kajko-Mattsson and J. Nyfjord, "State of Software Risk Management Practice", International Journal of Computer Science, IAENG, vol. 35, iss. 4, pp. 451-462, 2008.