A multiobjective TPM Strategic Release Planning model and its application on a CRM project

Samia NACIRI
TIME team, ENSIAS
Mohammed V University in Rabat; Morocco
snaciri@gmail.com

Mohammed Abdou JANATI IDRISI
TIME team, ENSIAS
Mohammed V University in Rabat; Morocco
janati@ensias.ma

Abstract— A customer centric strategy leads companies to accentuate business drivers through reliable Customer Relationship Management (CRM) software, which is maintained with efficient decision making processes, especially the process of release planning (RP).

To improve the decision quality of the maintenance release planning process, we have proposed in our previous paper a S-TRPM, a Strategic RP model for Third-Party Application Maintenance (TPM) projects. This Integer linear programming model supports, in a systematic way, TPM manager decisions of when to deliver customer change requests, increases the customer satisfaction, and respects the contractual commitment in terms of service level agreements (SLAs).

Furthermore, we have validated the application of our model in a real CRM project that was managed using an ad hoc RP approach and we have produced an automatic roadmap with better SLAs compared to the ad hoc RP SLAs.

In this paper, we aim to extend our model to address efficiently change requests (CRs) with an exceeded SLA through the minimization of CR penalties. This extension involves a multi-objective approach that adds a new objective function and adapts the existing SLA constraint of our S-TRPM. Then, we apply this multi-objective model in the same CRM project by means of the NSGA-II genetic algorithm. The resulting roadmap is constructed automatically in an iterative and incremental way, based on non-dominated solutions, and then compared to the ad hoc RP ones and previous work results.

Index Term— Software Engineering; Software Maintenance; Third Party Application Maintenance; TPM; Release Planning; Next release problem; Search-based software engineering; SBSE, multiobjective optimization

I. INTRODUCTION

Information System outsourcing is a business practice in which a company contracts all or part of its information system operations to external organizations (suppliers). This practice is done to gain economic, technological, and strategic advantages [1]. Third-party application maintenance (TPM) is a particular outsourcing field referring to the delocalization of products’ maintenance. TPM allows companies to control IT costs, improve products quality and scalability, and harness external resources to maintain their software [2].

Running TPM projects raises new challenges for both companies requesting and providing such services. One of the most challenging tasks within TPM software process development is relates to the Release Planning [3].

From TPM perspective, software release planning is considered as a tedious task. It takes into account the perspectives of project stakeholders, integration problems, functional and non-functional requirements, existing technologies, anticipating coming changes, concurrency and other goals. The interdependencies between change requests are also concerned.

Deciding which change request to include in which release is a hard Next RP problem discussed largely over the Software engineering literature. In respect of customer satisfaction, cost optimization, Service Level Agreements (SLA) and other goals, TPM manager search for an optimization approach, which considers several goals (multi-objectives) and proposes optimal solutions while respecting TPM constraints.

Scheduling the requested changes and planning their delivery in upcoming releases is considered as a strategic release planning activity (called also Software Delivery Roadmap) and not operational release planning activity, reported by [4]. Strategic Release Planning involves the selection and assignment of requirements to a sequence of upcoming releases, in respect of technical and organizational constraints as well as the allocated resources. Once a strategic plan is generated, a decision is made on what features should be part of which release, while operational planning focuses only on the selection of features to be included in the next release.

Release planning (RP) is classified in literature [5] as a “Wicked problem”, it means a problem difficult to define and often, it has no clear and definitive solution. Saliu and Ruhe have categorized software RP into ad-hoc RP approach and science based RP approach. Between the two approaches exists the hybrid approach, which is considered the most effective given that it enjoys the benefits of both categories [6].

We have proposed in our previous works a Strategic TPM Release Planning model (S-TRPM) that supports TPM manager in the process of release planning [7]. This model aims to plan TPM change requests over a release roadmap, in a systematic way. The first contribution of this model over existing works is the consideration of contractual commitment in terms of service levels agreement (SLA) which were not covered fully by existing RP models. The second contribution is to produce a Strategic maintenance plan that gives customers
a whole visibility on next releases and when to deliver pending change requests. Furthermore, this generated roadmap allows TPM manager to improve planning activities of software evolution stream, this by scheduling efficiently major evolution releases while considering the maintenance roadmap.

The purpose of this paper is to present a multi-objective approach to resolve release-planning problem from TPM perspective, in particular, for non-planned change requests, which have an exceeded SLA. This work applies a known genetic algorithm to resolve the proposed model and uses a CRM real data emanated from the CRM project case that was described in our previous work [8]. Results in terms of generated roadmap and SLA will be discussed and compared with both the applied ad hoc RP approach and the previous mono-objective RP approach.

This research joins the field of software engineering, called Search-based software engineering (SBSE). SBSE seeks to reformulate software engineering problems as search-based optimization problems and applies a variety of meta-heuristics based on local and global search to solve them (such as Hill Climbing, Tabu Search, and Genetic Algorithms). Our approach uses the concept of multi-objective optimization to deal with the specific problem of TPM release planning. NSGA II [9] and SPEA-II are the most used Genetic Algorithms and have been implemented to solve several problems in the Multi-objective and Combinatorial Optimization field.

The reminder of this paper is organized as follows. Exiting TPM RP models and our previous works are addressed in section 1. Our proposed multi-objective strategic TPM RP approach and model are outlined in section 2. The application of our multi-objective S-TRPM on the CRM project using NSGA II and the corresponding parameters, are presented in section 3. Finally, section 4 discusses results of applying this multi-objective approach and compares them with the ad-hoc ones before drawing conclusions and future work.

II. RELATED WORKS

In TPM context, change requests are a particular type of requirements related to existing features of the maintained software. CR often arises when the client wants an addition, correction or alteration to the agreed-upon deliverables, and can involve a wide variety of actions, including the delivery of corrective releases. Bugs, anomalies or defects are similar terms to corrective CR [10].

A. TPM RP models

As it was reviewed and analyzed in our previous paper [11], existing works don’t propose a strategic vision through a release roadmap for TPM, they only focus on the Next Release Problem (NRP). Their proposals are often limited to one release (the next release) or two software releases without managing SLA constraint of TPM, given that in TPM context, customers require a full visibility on next releases through a maintenance roadmap. Furthermore, few existing RP were validated by case studies derived from software maintenance industry in outsourcing mode [12]. The EVOLVE models propose a comprehensive approach to address the RP problem. However, it does not lead with SLA constraints or roadmapping expectations in TPM context [13], [14], [3].

Bagnall optimization [15], [16], COVAP [17], Provotype [18], [19], [20], IFM [21] models addressed NRP but didn’t tackle the TPM problematic in terms of compliance with contractual obligation of the subcontractor, manage urgent change requests, and generate roadmap. PASM (Process for Arranging Software Maintenance Requests) advocates consolidating maintenance requests in projects to challenge ahead the benefit of grouping change requests in software releases. This last helps to save allocated costs. However, it fails to consider business factors to assess the customer satisfaction degree, or risks associated with grouping CR [22].

When Release Planning Problem involves multiple objectives, the problem is named Multiple-Objective Next Release Problem (MONRP) [23]. This last is considered more realistic since Release planning involves complex and conflicting objectives.

In the next sections of this paper, we propose to use the word change request (CR) instead of “requirement” since it is closer to the corrective maintenance context.

B. Previous works

In our previous paper [7], we have proposed a new S-TRPM approach for TPM release planning that considers the customer satisfaction and TPM constraints from a strategic perspective. Thereafter, we have studied the applicability of our model within a CRM project case that was managed with a TPM contract [8]. This CRM project, which was called “ABC project”, is a large-complex project that generates huge volume of change requests and requires at least one release per month.

Application of the ad-hoc RP approach

Our previous paper [8] was depicts this ABC project specificities and presents results of applying the ad hoc Release Planning approach to schedule CRs. This ad-hoc approach has assisted ABC maintenance team to produce manually a release roadmap and to work in each release while avoiding scope changes: Before each maintenance committee instance, the TPM manager reviews confirmed CRs that require an applicative correction, and plans them for the next release if the scope freeze date is not yet reached.

Ad hoc RP data (inputs and results) were captured from ABC project repository and tools, and then analyzed.

Table I shows the main figures of the ad hoc RP roadmap: 95 CRs were planned over the year 2011 in 16 releases. These CRs were affected 14 customers (These results will be compared later with our proposed multi-objective S-TRPM results).
CRs that having an exceeded SLA. In addition to the first S-TRPM goal, which maximizes customer satisfaction, we aim through this extension to add a second goal, which will minimize CRs penalties. Those last can be generated if we consider that each SLA is measurable and associated with a financial penalty.

Indeed, we propose in the next section, a multi-objective approach that is based on the following rules:

- Add a new objective function to the first one that allows planning CRs with exceeded SLA as soon as possible in first releases while considering the CR severity and urgency levels.
- Adapt the SLA constraint of our proposed S-TRPM model to only address CRs with non-exceeded SLA

This multi-objective approach allowed us to extend the S-TRPM model to be a multi-objective Strategic TPM RP Model.

III. A MULTIOBJECTIVE STRATEGIC TPM RP MODEL

To implement our multi-objective S-TRPM approach, we have searched in the field of multi-objective optimization problems (MOP) that helps us to identify an adequate technique to use. This section presents our proposed approach, formulation and resolution tool.

A. Multi-objective S-TRPM approach

Our proposed approach considers four main constraint categories (business constraints, resource constraints, system constraints and monitoring constraints) and is based on five stages as shown in fig. 1. A deep description of these constraints categories can be found in our preceding paper [7].

![Fig. 1. Strategic TPM release planning approach [7].](image)

I) Qualification stage: this stage which is executed by the support chain team aims to identify the corrective CRs to be planned. The team assesses the severity and urgency of each CR and collects additional details of the request sender and CR.
Table II
CR data structure (a sample)

<table>
<thead>
<tr>
<th>CR identifier</th>
<th>Description</th>
<th>Creation date</th>
<th>Customer</th>
<th>Severity</th>
<th>Urgency</th>
<th>Cost</th>
<th>CR dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>C1</td>
<td>Low</td>
<td>Major</td>
<td>5</td>
<td>Dependents C1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>01/02/2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CRs d1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C2</td>
<td>Normal</td>
<td>Normal</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>02/04/2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CRs d2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C1</td>
<td>High</td>
<td>Critical</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>03/05/2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CRs d3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C3</td>
<td>Normal</td>
<td>Major</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>04/06/2011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CRs d4</td>
</tr>
</tbody>
</table>

3) Strategic TPM RP generation stage: It concerns the CRs distribution to software releases, called also roadmapping. Based on evaluation stage results, this stage searches for an optimal combination of CRs that maximizes stakeholder satisfaction through CR business value and minimize penalties for CRs with an exceeded SLA, while respecting RP constraints. This stage includes several iterations depending on CR stock.

4) Execution stage: it’s a working stage of TPM team when the next release is built.

5) Feedback capture and analysis stage: this stage is essential to capture information concerning the quality of the delivered releases and CRs and helps to improve next releases.

B. Multi-objective S-TRPM formulation

According to the proposed multi-objective approach, the third stage (strategic TPM RP generation) involves an iterative activity; in each iteration we fix the content of a current release. The process is repeated until all pending change requests will be planned.

In [7], we have formulated this stage as an integer linear programming model where we have used and adopted the following sets and assumptions:

- D = {d1, d2, ..., dn} Change Requests to be scheduled and still present in the TPM backlog at the beginning of a given iteration. D will be updated in each iteration.
- R = {R1, R2, ..., Rm} the maintenance releases set where m corresponds to the number of iterations needed by the third stage. CRs are affected to corrective software releases named “Rk”, with Rk delivery date is the closest to the current date.
- C = {c1, c2, ..., cm} represents stakeholders of the software product. Inspired from [6], the TPM team can assign to each customer a weight or a degree of importance α ∈ {1, 3, 5, 7, 9}. Formally, α ∈ {1, 3, 5, 7, 9} may mean respectively: ignored / not important / neutral / important / very important. We will note α(cj, di) the customer weight concerned by the change request di and α(di) the customer weight if the change request has one customer.
- CR urgency: Depending on the emergency of each CR di, a CR can be of high, normal or low urgency. Considering the function urgency with urgency(di) ∈ {U1, U2, U3} where U1, U2 and U3 are predefined constant values. This function associates to each urgency level one of the mentioned constants.
- CR severity: considering Sev = {High, Medium, Low}, a severity set defined for the Software Maintenance project, each CR di depends on its severity level has a predefined constant SLA in regard to the contractual commitment. Our model considers the correction time SLA using the function TC( ). (TC(High), TC(Medium), TC(Low)) are constants that are defined in the TPM contract.
function allows to retrieve the penalty associated to a given CR.

The rest of the multi-objective model is as follows:

**Decision variables**

\( X_k \) is a vector of binary decision variables. For each release \( R_k \):

\[
X_k = (x_k(d_i))
\]

with \( i = 1 \ldots n \) and \( x_k(d_i) = 1 \) if the Change Request \( d_i \) has not been assigned to \( R_k \) with \( 1 \leq k \) and assigned to the release \( R_k \). \( x_k(d_i) = 0 \) otherwise.

As a consequence, we will have \( \sum_{i=1}^{k} x_i(d_i) \leq 1 \) for each \( d_i \) at each iteration \( k \).

**Objective function**

To satisfy customers according to their weights and Change Requests emergency degree and minimize CRs penalties, our multi-objective S-TRPM aims to select a subset of customer change requests which satisfy the following objective functions for each release \( R_k \):

Max \( \sum_{i=1}^{n} \sum_{j=1}^{l} \alpha (c_p(d_i)) \times \text{urgency}(d_i) \times x_k(d_i) \) \hspace{1cm} (1)

Min \( \left( \sum_{i=1}^{n} \text{if } \text{delivery}_d(d_i) - \text{creation}_d(d_i) \geq \text{SLA exceeded } \text{delivery}_d(d_i) - \text{creation}_d(d_i) \times \text{TC } \text{Sev}_d(d_i) \right) \times \text{Pen}(d_i) \times x_k(d_i) \) \hspace{1cm} (1')

In this multi-objective approach, we propose a first objective function (1) that aims to prioritize change requests that have both high urgency level and customer importance weight. The second objective function (1') aims to deliver sooner change requests that have an exceeded SLA, which will reduce the generated penalties.

Our multi-objective model proposes to satisfy constraints detailed below:

**Business constraints:**

When selecting CRs that to be assigned into a release \( k \), TPM managers have to ensure that SLA contractual commitmens in terms of Correction Time(TC) are satisfied.

\[
\text{delivery}_d(d_i) - \text{creation}_d(d_i) \times x_k(d_i) \leq \text{TC } \text{Sev}_d(d_i) \times x_k(d_i)
\]

\( d_i \in R_k, i = 1, \ldots n, \)

This last corresponds to the SLA constraint, with delivery_date and creation_date are input parameters defined for each CR that have a non-exceeded SLA. The TC (Sev (d)) was defined above.

**Resource constraints:**

\[
\sum_{i=1}^{k} \sum_{j=1}^{l} \text{Cost}(d_i) \times x_i(d_i) \leq \text{budget}
\]

(3) means that releases costs are under the allocated budget of the global roadmap. \( \text{Cost}(d_i) \) and \( \text{budget} \) are input parameters expressed in man-days. Budget is calculated based on working man days.

\[
\sum_{i=1}^{n} \text{effort}(d_i) \times x_k(d_i) \leq \beta (R_k)
\]

\( \beta \) corresponds to the available team effort that is defined by the TPM manager for each \( R_k \). The effort () function returns the required effort for each CR \( d_i \).

**System constraints:**

\[
x_k(d_i) = x_k(d_{i'})
\]

\( \forall (d_i, d_{i'}) \in \text{Cooupling set} \)

Coupling means that the implementation of \( d_i \) and \( d_{i'} \) must be scheduled in the same release.

\[
\sum_{l=1}^{k} x_i(d_{l'}) \geq x_k(d_i),
\]

\( \forall (d_i, d_{i'}) \in \text{Precedence set} \)

Precedence means that the implementation of \( d_i \) must precede \( d_{i'} \).

Constraints (5) and (6) express dependency constraints that can exist between change requests.

**Delivery policy constraints:**

The release delivery policy describes which factor will be considered to limit the scope of releases. Define a delivery policy, share it, and approve it with all project stakeholders (maintainer and client) is necessary for TPM management.

According to the selected maintenance delivery policy, we can add one or some of the three formulas bellow, depending on the business owner:

- For fixed time delivery policy, it concerns the release delivery frequency or distance “DIST” (a constant value) between releases. For example, the TPM client may require one release delivery per month:

\[
\text{delivery}_d(R_k) - \text{delivery}_d(R_{k-1}) = \text{DIST}
\]

Where DIST is a constant and delivery_date (\( R_i \)) is fixed.
• For fixed cost delivery policy, this policy allocates a specific budget to each release \( R_k \). It is formulated as the following:

\[
\sum_{i} \text{Cost}(d_i) \times x_k(d_i) \leq \text{budget}(R_k) \quad (8)
\]

Where \( \text{Cost}(d_i) \) and \( \text{budget}(R_k) \) are defined.

• For fixed CRs number delivery policy, it defines the fixed number "\( N \)" of CRs to deliver in each release \( R_k \). It is formulated as the following:

\[
\text{card}(R_k) \leq N \quad (9)
\]

where \( N \) is a given constant for all \( R_k \).

The given formulation of the TPM RP problem with equations (1) to (9) constitutes a multi-objective optimization problem that will require efficient optimization algorithms to be solved.

In the next sections of this paper, we will apply our proposed multi-objective S-TRPM approach to a CRM project part of Tradecom company portfolio. This large CRM TPM project has used the ad-hoc approach to produce release plans. We aim through this case study to show the benefit of our multi-objective approach in a real world release planning situations compared with the ad-hoc RP approach.

IV. APPLICATION OF THE MULTI-OBJECTIVE S-TRPM MODEL ON THE CRM ABC PROJECT

To implement our multi-objective S-TRPM approach, we searched the field of multi-objective optimization problems (MOP), which was helped us to identify an adequate technique to use.

A multi-objective optimization problem (MOP) involves two or more conflicting objective functions, which are to be either minimized or maximized [22].

Usually, it is not evident to find a single optimal solution satisfying all objectives. Instead, many good solutions “Trade-offs” may exist, that are popularly known as Pareto-optimal solutions. The concept of Pareto optimal front is used to separate dominated and non-dominated solutions. Formally, each element on this front is a candidate solution. The Pareto front thus represents a set of ‘best compromises’ between objectives that can be found by the search based algorithm.

One of the most appropriate strategies to determine a good approximate solution of an NRP multi-objective problem is a metaheuristic algorithm. This last is based on the population that evolves over the solution space and allows to find a set of non-dominated solutions.

A. Resolution Approach

Our multi-objective S-TRPM adopts an iterative approach as shown in Fig. 1. In each iteration \( k \), we will call a multi-objective S-TRPM algorithm, which is the implementation of our model to the still non-planned CRs in the iteration \( k \) to produce release \( R_k \) (planned CRs in \( R_k \)) and reject CRs that do not respect the given constraints.

The first proposed step of this approach is the definition of the delivery date of the first release and the number of days of the freeze period. Then, execute the multi-objective S-TRPM algorithm and generate a Pareto front for the two objective functions: maximizing the customer satisfaction and minimizing the rest of the non-planned CRs with an exceeded SLA.

The TPM manager may manually select one of the generated solutions (non-dominated) based on the acceptability of their values trade-off. The selected solution constitutes the first release (selected change requests). Next, the multi-objective S-TRPM algorithm is executed again to the rest of non-planned CRs until that all change requests were be planned. This approach generates a release roadmap of all existing change requests.

![Multi-objective S-TRPM resolution approach](image)

The implementation of our approach takes in input a flat file that contains existing change requests (to plan). Each Change Request is described and provides the required information to resolve the release-planning problem. The optimization algorithm is performed allowing setting the two model objectives and model constraints based on a binary variables string.

B. Implementation using NSGA-II (Elitist Non-dominated Sorting GA)

We implemented our approach using the NSGA-II multi-objective optimization algorithm part of the MOEA tool. MOEA is a free and open source Java Framework, available from http://www.moeaframework.org. MOEA provides a number of algorithms, including NSGA-II, ε-MOEAs, GDE3 and MOEA/D. In addition, the MOEA Framework provides tools necessary to rapidly design, develop, execute and statistically test optimization algorithms.
For this multi-objective S-TRPM, a simple binary GA encoding is used. Each bit codes a decision variable. For an instance, the bit string 00110 would place change request 3 and 4 inside the release, excluding the rest. Therefore, the length of a solution is equal to the number of decision variables. The tournament selection, a single-point crossover and bitwise mutation operator for binary-coded GAs are applied [9].

The main steps of the multi-objective S-TRPM algorithm are described in Table III.

| I-1 | Build the D set: D includes CR of not yet scheduled for which the creation date is less than or equal to the freeze date. Each d, is represented by a java object of structure seen in table II. |
| I-2 | - set k-1 
- set the delivery date and the freeze date of R_k, |
| I-3 | While they still exist d, which are at the same time not planned and not rejected: 
- set the R_k release Name, 
- set the R_k delivery date and the freeze date 
- do I-31 |
| I-31 | (To build objective functions inputs), for each of these d, do: 
- calculate the first objective function value : 
  \[ \alpha(c_i, d_i) * \text{urgency}(d_i) \] 
- calculate the second objective function value : 
  \[ \sum_{i=1}^{\text{d} \text{d}_{\text{i}} \text{exc}} \left( \text{delivery date}(d_i) - \text{creation date} (d_i) \right) \] 
  \[ - TC \left( \text{Sev}(d_i) \right) * \text{Pen}(d_i) \] 
- define X_k(d_i) integer type and values ranges 
- if sev(d_i) is high or critical with a workaround solution, define the TC extension time |
| I-32 | Build dependencies constraints |
| I-33 | Build resource constraints |
| I-34 | Build SLA constraints |
| I-35 | Call the library optimization functions 
Choose the first non-dominated solutions |
| I-36 | Build the R_k: read the generated bit string which constitutes the X_k vector then update each object of these d, if its corresponding bit is equal to 1 
- d, release name = release name 
- d, delivery date = release delivery date 
- d.state = ”planned” 
- go to I-3 |
| I-4 | Write d, objects in flat file (RP roadmap) |

In this paper, we propose to resolve our multi-objective S-TRPM using the non-dominated sorting genetic algorithm NSGA-II introduced by [9]. The main objective of NSGA-II algorithm is to find a set of solutions ordered by fronts, under the concept of Pareto dominance. The algorithm maintains both a population and an archive with the size of N solutions.

In this proposed resolution, we select (in each iteration), systematically, the first non-dominated solution generated by NSGA-II.

1) NSGA-II or Elitist Non-dominated Sorting GA algorithm

Referring to [25] paper, the NSGA-II algorithm is one of the popularly used evolutionary multi-objective algorithms followed by Genetic Algorithm, Integer Linear Programming, and Greedy. NSGA-II attempts to find multiple Pareto-optimal solutions in a multi-objective optimization problem and has the following three features according to [26] and [27]:

- It uses an elitist principle,
- it uses an explicit diversity preserving mechanism,
- and it emphasizes non-dominated solutions

The Pareto front offers a range of possibilities (non-dominated solution) depending on the proposed problem objectives. The best solution depends on the specific interests of the decision maker.

2) NSGA-II inputs

To apply our proposed model to the TPM CRM project “ABC”, it is necessary to start by capturing all required variables and parameters defined previously by our model [11]. These inputs were outlined in our previous paper [8].

Parameters that we set for NSGA-II algorithms are based on previous empirical researches. Settings used in this paper were mostly emanated from the MOEA framework [28] default setting (Many of the optimization algorithms that can be executed within the MOEA Framework are provided by the JMetal library).

- To parameter NSGA-II algorithm, we used the default configuration described in Table IV:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>100</td>
</tr>
<tr>
<td>Crossover type</td>
<td>Single-Point Crossover</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>0.9</td>
</tr>
<tr>
<td>Mutation Type</td>
<td>Bit-Flip Mutation</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>1</td>
</tr>
<tr>
<td>Max evaluation</td>
<td>60 000</td>
</tr>
</tbody>
</table>

- Problem encoding: Like a Knapsack problem [9], we are faced to binary variables, which take two possible values 0 or 1.
- Defining the Optimization Objectives: In this work we optimize the following objectives:
  - Maximize the Customer satisfaction
  - Minimize penalties change requests for which SLA are exceeded
- Roadmap construction: depends on data volume (change requests number). This program allows generating a release roadmap that absorbing all pending change requests while constraints respect including SLA.

This work focuses on the generation of RP roadmap that uses the NSGA-II, rather than tuning NSGA-II parameters to achieve the best performance settings, which could be explored in future works.
RESULTS AND DISCUSSION

The proposed resolution approach of the multi-objective S-TRPM model allows us to generate a release roadmap of the studied project as illustrated in Fig. 3. This last presents for each release: number of delivered CRs, cost sum, and also the two objectives values.

In the following paragraphs, we will compare results of the ad-hoc RP process and the ones obtained by our multi-objective S-TRPM model. The generated roadmaps by these two models will be compared according to two axes: CR delivery date and SLA (TC) commitment.

Regarding to axe 1:

The comparison of delivery dates obtained in roadmaps of the two models can be summarized as shown in Fig. 4.

Comparing our model roadmap using NSGA-II to the real roadmap, we note that 60% of change requests are scheduled in the same release month (32% of CR are in the same calendar month and 28% of CR are delivered in less than 30 days of difference). Those results are illustrated in Fig. 4 that presents how the NSGA-II generated-delivery dates are near or far compared with the real-delivery dates for all CRs.

Regarding to axe 2:

Our generated NSGA-II roadmap allows us to monitor easily the SLA of each CR. As defined previously, this study considers the SLA correction time (TC) which is the difference between the CR delivery date and creation date. In Table 1, we compare CRs generated TC of the two models.

Table 1: Comparing NSGA-II TC and ad-hoc TC by severity level

Fig. 3. Results of NSGA II application on real data

Fig. 4. Comparing NSGA-II delivery dates to ad-hoc ones

Table VI

Only 9% of change requests are not in the same ad-hoc delivery period (delivered after more than 60 days by our multi-objective RP model). Several rasons may explain these differences. Obviously, the extracted CR data does not track customer decision of including or delaying a CR in a release, neither urgent releases decision. In a real planning situation, customer may decide to include or exclude a CR if the business context requires.

Moreover, our resolution approach considers two objective functions within the same weights. This last consideration may be adapted in future work in order to evaluate how prioritizing the client satisfaction in favor of SLA respect or vice-versa may affect the generated roadmap. A real roadmap is based on human decisions, which remain subjective ones that take into consideration current context of CRs and customers.

The Table and Fig. 5 illustrate four indicators that present the number of CRs with best correction time (TC) and compare the NSGA-II TC with the ad-hoc ones.
• Generated TC : indicates the number of CRs for which the NSGA-II roadmap has a better TC than the ad-hoc one.
• Real TC : indicates the number of CRs for which the ad-hoc roadmap has a better TC than the NSGA-II one.
• Real TC with non-respect of freeze date: this indicator is similar to the second indicator, except that it considers CRs for which the freeze date rule is non-respected
• Real TC with non-respect of delivery policy: this indicator is the same as the second indicator, except that it considers CRs for which the constraint of delivery policy(one release per month) is non-respected

The two last indicators (42%) reflect the ad-hoc RP reality, which does not respect the TPM RP rules, in particular for critical and urgent CR without workaround solution. Our proposed multi-objective strategic TPM RP model applies the delivery policy even if there is CR to deliver sooner (urgent or critical) before the planned release.

In this multi-objective implementation, we have automatically selected (in each iteration) the first non-dominated solution generated by NSGA-II algorithm. Evolutionary algorithms (EA) have been effective in many software engineering problems. EAs do not guarantee to find the optimal solutions for a problem, however, they provide near-optimal solutions for decision making, which is perceptive for our TPM RP proposed approach. The NSGA-II used algorithm we think is appropriate to address the main research questions in this work. Through our results, we experiment that evolutionary optimization algorithms are at nearly level effective to resolve TPM RP problem.

CONCLUSION AND FUTURE WORK

TPM is a complex discipline that covers multiple activities in the software maintenance process. Some important process such as release planning which selects and prioritizes change requests can be an extremely arduous and exhaustive activity, depending on the size of the software, the application domain, interdependence between change requests, management approach and others.

In order to validate our TPM RP model, we have applied our multi-objective strategic TPM RP model to a real dataset using one of the optimization techniques that is denoted NSGA-II. Results show that the generated NSGA-II roadmap provides better SLA than the ad-hoc ones for more than 37% of the examined CR, which is a good rate. Furthermore, the NSGA-II roadmap was systematically produced within the respect of resources and policies constraints.

Our resolution approach applies the multi-objective S-TRPM model with a strict respect of delivery policy and freeze date constraints. In real situation, TPM release planning can produce unplanned (unexpected) releases if a critical CR arises without workaround solutions. This last statement can be explored in future works with new cases study.

REFERENCES


