Abstract — Current study uses the event process logs of a custom software development process used internally in the IT department of a large automotive company. The target is to extract hidden information about the process, difficult to identify due to the wideness and complexity of the input data. The log file contains all incident records tracked during one release of a software product. This analysis was performed using process mining techniques and tools included in ProM Framework, an academic project of the Eindhoven Technical University. The paper describes the steps followed to extract the working process model, organizational network and statistical information. Based on the obtained results, an action plan is put in practice with the scope of improving the development process and the release result.

Keywords—process mining, process model, event log, heuristic miner

I. INTRODUCTION

Process mining [5] represents a specific area of data mining domain focused on event process logs recorded by an information management system during the lifecycle of a software project. Process mining analyzes process logs with the scope of discovering real process models, enhancing, correcting and improving the performance of processes. Besides process discovery and extension, process mining is able to provide different perspectives of the organization such as the social perspective. [1]

Large software organizations with complex software development processes are the perfect candidate for applying process mining techniques. In such organizations it can be difficult to identify the real bottlenecks in the working processes or to measure the global performance of the process. Process mining fills the gap between “assumed” processes and the real ones. The output is a window of reality and not a theoretical process model introduced but hardly followed. Such information is valuable for the development of organizations. Better processes lead to better results and better business.

Process mining could be split into 3 major phases [2] resumed in Fig. 1.

In the first phase, data preparation, the input for process mining algorithms is prepared and the context is defined: work methodology, expectations and targets for the results of the mining analysis.

In the second phase, pattern discovery, the process mining algorithms are applied on the prepared dataset. In the third phase an analysis of the results obtained is performed. This includes checking the statistical results for calculating performance indicators, verifying the conformance of the resulted process model, comparing theory with reality, retrieving other relevant information internal to organizations, identifying bottlenecks.

Process mining techniques include:

- Extracting process models from event logs
- Identifying deviations by comparing the output resulted model with the input theoretical model
- Updating and extending the initial process model
- Verifying the conformance of the new process models obtained
• Extracting statistical, performance information

Process mining becomes an important tool for enhancing organizations that must manage and improve non-trivial processes. It’s still a new and challenging area that requires expertise for obtaining good results. One of the main limitations represents the quality of the event log data which can be incomplete, inconsistent, coming from different sources, with a different level of granularity [1]. Another challenge represents overcoming the human factor in putting in practice the process: activities handled differently due to different assumptions and premises, dependent on the level of knowledge, expertise and approach of the owner.

Event logs are the start point for mining from three different perspectives [2]:

• process perspective with the scope of identifying all possible paths an item could follow to reach a close state
• organizational perspective for checking which are the resources involved in the activities and how they interact
• case perspective focused on analyzing specific behavior due to specific values of the attributes

II. SOFTWARE DEVELOPMENT PROCESS

With the scope of increasing the control in the software development methodology, a large software product is split into software elements (SE). The entity in charge of managing structure elements is called Software Configuration Management (SCM). A component of the SCM in charge of handling changes required by users of different types (change requests, feature requests, problem reports and information requests) is called Change Control Board (CCB) [4].

![Change Control Board model](image)

The CCB model as presented in Fig. 2 is a theoretical model, resilient to the diversity of the changes submitted (level of details, complexity, quality of the requirements, solution), human and external factors (organizational constraints).

In the next chapters the real process model is identified and compared with the theoretical one.

III. DATA PREPARATION

Data preparation represents the first step in the process mining methodology. The information available from providers such as IMS (Information Management Systems), ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), and SCM (Supply Chain Management) is analyzed and transformed in a standard format.

The current analysis is performed using ProM 6.3 Toolkit [6], an open-source framework for implementing process mining.

The current study relies on the event logs recorded during one release of a software product. The development was outsourced to an external company. The process followed was defined by the external customer as a project initial requirement but the company responsible with the implementation had to customize it due to internal organizational constraints. The custom implementation of the process model inside the outsourcing provider company could cause deviations from the initial source model. Such deviations are tracked using process mining techniques.

Any process mining analysis session should start with concrete questions. Without a target, extracting meaningful data becomes very difficult. In the current situation the following questions represent the start for applying process mining techniques on the current dataset:

• What is the real process model and what are the differences between real and theoretical models
• Could an enhancement be identified in the process definition with the scope of avoiding rework of items (change of requirements, problems reported after release) which cause overhead, additional costs and risks
• What is the quality of the requirements (how many items are rejected in the first line, before analysis, due to conflicting specification, user error)

The event logs recorded are exported from the internal tool tracking all the changes, IMS, in csv format:

• users.csv – all users having permissions to access and update the items in IMS
• items.csv – all requirements (change requests, feature requests, problem reports, information requests)
• events.csv - all operations performed by the users to update the state of the items

An item represents one requirement submitted by the support team on customer side. It could be of the following types:

• problem report – a regression or defect found in a maintained release
• change request – the existing feature should be changed due to one or more reasons such as usability, technical impact, architecture, implementation
• feature request – a new functionality is requested
• information request – the current behavior is checked.
An item is defined by several other mandatory fields: title, description. These will represent attributes in the resulted event log.

The items are operated by users which belong to a group (support, management, and engineer) and have a role (engineering, project team).

Events are new states for items triggered by users. An event is defined by several attributes:

- eventName - the ID of the item operated
- userID – the ID of the user who triggers the event
- timestamp – identify when the event occurred
- details – other information
- eventType – the new state of the item.

Below are listed the possible values:

- Create – the item is created by the support team
- Open – the item is opened by the engineering team
- Analysis – the item is analyzed either by the engineering team (check impact, implementation, plan and estimate) or by the support team (check the specification proposed or the resulted implementation)
- Specification – the item is specified by the engineering team
- Implementation – the item is under implementation and scheduled in an intermediate or final release
- Solved – the item is implemented
- In Tests – the item is under validation by the test team part of the engineering team
- Complete – the item is completed and included in a release

Below samples from events.csv, items.csv and users.csv are extracted for exemplification:

<table>
<thead>
<tr>
<th>OrderID</th>
<th>EventName</th>
<th>UserID</th>
<th>EventType</th>
<th>Timestamp</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TEP-SWCN-65</td>
<td>1</td>
<td>Create</td>
<td>8-1-2013 10:00</td>
<td>Please see if this could be scheduled in the next release</td>
</tr>
<tr>
<td>2</td>
<td>TEP-SWCN-65</td>
<td>2</td>
<td>Open</td>
<td>8-10-2013 10:30</td>
<td>Notification received, item will be analyzed</td>
</tr>
</tbody>
</table>

ITEM_ID;TYPE;PRIORITY;TITLE;DESCRIPTION

TEP-SWCN-65;FeatureRequest;2;Brace highlighting:Implement brace highlighting feature as it is implemented in the default C editor of eclipse

TEP-RCN-8;FeatureRequest;4;Enhance printing:Implement print selection of structures

userID;userName;userGroup;userRole

1;Alexander;Support;Engineering

2;Laura;ProjectManager;ProjectTeam

In practice event logs could have all kinds of formats. A common accepted format was introduced as a base for process mining processing. This format is called XES – Extensible Event Stream, an XML-based standard for process event logs [7].

As the exported event logs of the current analysis are in csv format, OpenXES open-source library was used to transform the csv data in an XES event log. An XES file is composed of the log instance of one process. The log is composed of traces representing the execution instances. A trace is composed of events. In the current study the trace contains all the activities logged for one item. Each event is defined by a set of attributes.

Traces have one global attribute concept:name mandatory for all items which in this case is represented by the item identifier.

```xml
<event>
  <string key="concept:name" value="TEP-SWCN-65"/>
  <string key="org:group" value="ProjectTeam"/>
  <string key="org:resource" value="Laura"/>
  <date key="time:timestamp" value="2013-10-30T10:00:00+03:00"/>
  <string key="lifecycle:transition" value="complete"/>
  <string key="concept:name" value="release"/>
  <string key="direction" value="fwd"/>
  <string key="details" value="Item completed included in 6.0.0 M1"/>
</event>
```

IV. PATTERN DISCOVERY

The pattern discovery is the phase of extracting a process model [1] from the event log data in XES format. During this
phase, the log could also be filtered with the scope of simplifying and increasing the visibility of process patterns. [3]

For the current study, the full event log data was used to discover the real process model, and filtered afterwards after attribute direction. This filtering was performed with the scope of analyzing the problem that lead to the second question: what is a common characteristic of the items reworked and how could loops in process model be avoided.

In the first step the event data log is mined using the Heuristic Miner algorithm. This is one of the most commonly used process mining algorithms, stable and suitable for real-life data. The well-known alpha algorithm [1], is recommended more for checking properties of events and extracting patterns based on properties. The fuzzy miner algorithm represents the new generation of process mining algorithms and is mostly used for dataset with major inconsistencies. The result is the process model that was followed collaborative by the engineering, support and management teams.

![Real process model](image)

As a brief description of the process model, after one item is created, it can be opened or transferred to the backlog, cancelled. Once an item is opened, it is either analyzed or implemented. Analysis could be followed by define step, case in which a new analysis is performed afterwards. Besides define as a next step, analysis could be followed by implementation. The next activity after implementation is always test. Following test, an item is released. A released item could be reopened, case in which the complete workflow is re-executed.

V. ANALYSIS PHASE

This analysis phase is conducted targeting the identification of the differences between the process model obtained after pattern discovery phase execution and the theoretical model.

The following conclusions are drawn after comparing the theoretical process model Fig. 2 with the real process model Fig. 3:

- in the real process model the analyze step could be missing (some items are moved from open activity directly to implementation)
- in the real process model if the analyze step is not missing, it could be split into 3 activities: open – the project team is notified about the submitted change; analyze – the project team identifies the root cause; define – a detailed specification is prepared by the project team and proposed to the customer
- in the real process model the customer evaluation of the analysis submitted by the project team could be missing
- in the real process model the solve activity is split into implement and test and handled as separate activities by different resources
- in the real process model no evaluate activity follows the resolution of an item, before release activity
- in the real process model an item could be cancelled either when it was submitted (problem is not reproducible) or after analysis/define is performed (if the solution for the item breaks the design, product specification, or the cost is too high for the value added)

The next step is to check the performance of the real process model by analyzing the obtained statistical results:

- 48.28% of items are opened and analyzed after create
- 31.03% of items are cancelled after submission
- 20.69% of items are cancelled after open (user error, contradictions with product specifications, incompatibilities with current design)
- only 23% of items opened are actually released
- 56% of items are implemented without an analysis in advance
- 50% of items are also defined (specified) after analysis
- all items are tested after implementation
- all items implemented and tested are released
- 23% of items are re-opened after release

Surprisingly the most time consuming activity proves to be define when it is actually present in the lifecycle of an item. This could be caused by some lacks in the technical writing area in the project team.
After analyzing the results, some general conclusions concerning the performance of the process model are drawn and listed below.

A large number of items are cancelled and a lot of time is spent for cancelling them (opening, checking with the user, analyzing). As there is no value added for such items, this rate must be decreased for increasing the global process performance.

A very small number of items are actually released. The rest of them are cancelled in different process phases. This rate must be increased.

A large number of items are implemented without an analysis in advance. This could lead to misinterpretation of the requirements, incorrect assessment of the impact over the product, incorrect/incompatible solution, limitations. Such an approach is error prone. The analysis step should follow the creation of an item and should be customized by the type of the item, the difficulty, the severity, priority.

A good number of items are re-worked. This is even more problematic as rework might not be detected for all items which were actually reopened due to problems found or incorrect definition of the requirements. Bugs could be tracked as new items and not handled as being caused by one previously handled item. This could even be continuous, from one release to another. This is one known limitation of the dataset caused by a project management approach. Reworked items that cause loops in the process model are analyzed in the next chapter.

VI. FILTERING

The scope of customizing the dataset and filter only a limited set of items which have the value for the attribute direction “back” is to identify a common pattern for items reopened.

The real process model obtained on the filtered dataset is presented in Fig. 5.

The filtered dataset represents the input for the social network plugin embedded in ProM Framework. The scope is to identify the communication structure and dependencies among involved people for the case of items reworked. The obtained social network is shown in Fig. 6:

The filtered dataset analysis leads to the following results:

- Reworked items are created mainly by 3 users from support team as seen in Fig. 6. In order to avoid rework, these users could receive additional training and support for logging items with a good level of information.
- The real process model for the filtered dataset does not include any analysis activity performed by the project team. 50% of items for which an analysis is not performed are reworked.
- Reopening of items could be avoided if this analysis activity is introduced as mandatory and handled accordingly: check requirement, define specification, define solution, check the impact, and define testing strategy.
- The real process model does not include any analysis phase of the support team for verifying if the project team understood the context, the details of the requirement.
VII. CONCLUSIONS

Some answers are obtained by applying process mining techniques on an exported, standardized and filtered dataset created during the software development process execution. Some statements still remain deeply hidden in the dataset. Some cannot be found with the current level of information of the dataset and limitations of the applied process mining algorithms. But in general, process mining is able to provide a real perspective, closer to the reality, over any process followed inside an organization. This information could be used for defining a strategy on how to increase the productivity and the performance of an organization.

Following the current study, an action plan is put in practice inside the outsourcing software company, for handling future releases of the same product as the one analyzed:

• The process is adapted to support continuous integration of the work to avoid rework and integration difficulties.

• The analysis step is introduced as mandatory. This activity must be executed by the project team when an item is submitted (check the impact, identify possible solutions, check limitations) and by the customer support team responsible with creating the items (check if the requirement was correctly understood, check if the resulted solution is the expected one).

• A clear checklist is defined for submitting the items (scenario, level of information).

Applying process mining on a dataset is not a straightforward activity. It requires a deep understanding of the data and a lot of expertise in interpreting the results and finding solutions to enhance the process. The difficulties are also fueled by the level of usability of the existing process mining tools which are academic, not commercial. But in spite of the difficulties, process mining relies on data that already exists, it’s not dependent on any of the external factor and could provide valuable information that should increase the level of productivity inside an organization.

VIII. FUTURE WORK

For organizations with custom dynamic processes as in the outsourcing business, a backward mining analysis could bring some improvements to future processes but as usually work methodologies are defined with the customer, specific to the collaboration, the analysis results are limited to the work with that customer on a specific product line. If the collaboration is a long term one, than the process mining output could lead to better products and higher efficiency. But in the software development area, the organizations have to keep a high level of dynamism in order to adapt to the market. Adapting to new customers, new methodologies, and development processes is considered a survivor kit. The process mining analysis results are not very useful for such a dynamic business area. Due to this, adding real time capabilities is one of the goals we have in mind for future research.

Process mining could fill the gap between business management trends (continuous process improvement, six sigma, and total quality management) and production, another reason why it’s a high potential area.

The quality of the process mining output from the current academic and commercial implementations are tightly dependent on the event logs input. Logs may contain data of different level of granularity; the information could be inconsistent, coming from different sources. The impact over the obtained results is too high. Introducing a preprocessing step of the event log data is another direction we plan to follow in order to increase the usability of process mining and quality of process models.

Quality in process mining is definitely a domain which needs enhancement. Validating process models results in a controlled manner would increase the confidence in this area.

Process mining is used currently inside one organization. This working model is not feasible nowadays when most of the software is produced in collaborations in between organizations. Enhancing process mining to produce results in the multi-organizational context is a challenging future work direction.

References


