Tailoring the Engineering Design Process Through Data and Process Mining

Maruster, Laura; Alblas, Alex

Published in:
IEEE Transactions on Engineering Management

DOI:
10.1109/TEM.2020.3000861

Publication date:
2022

Tailoring the Engineering Design Process Through Data and Process Mining

Laura Maruster and Alex Alblas

***Abstract*—Engineering changes (ECs) are new product development activities addressing external or internal challenges, such as market demand, governmental regulations, and competitive reasons. The corresponding EC processes, although perceived as standard, can be very complex and inefficient. There seem to be significant differences between what is the "officially" documented and the executed process. To better understand this complexity, we propose a data-driven approach, based on advanced text analytics and process and data mining techniques. Our approach sets the first steps toward an automatic analysis, extracting detailed events from an unstructured event log, which is necessary for an in-depth understanding of the EC process. The results show that the predictive accuracy associated with certain EC types is high, which assures the method applicability. The contribution of this article is threefold: 1) a detailed model representation of the actual EC process is developed, revealing problematic process steps (such as bottleneck departments); 2) homogeneous, complexity-based EC types are determined (ranging from "standard" to "complex" processes); and 3) process characteristics serving as predictors for EC types are identified (e.g., the sequence of initial process steps determines a "complex" process). The proposed approach facilitates process and product innovation, and efficient design process management in future projects.**

***Index Terms*—Classification model, clustering, engineering change (EC), innovation, process mining, process model, text analytics.**

I. Introduction

NEW product development (NPD) encompasses a variety of activities aiming to translate an idea into a new product launch. Different process models for organizing NPD activity, such as the stage-gate model [1] and the engineering change (EC) management (ECM) process model [2], improve our understanding of managing NPD sequence of activities. Such models, however, mostly focus on single actions, rather than the interactions between them [3]. Most process behaviors emerge from the interaction between activities through information flows and deliverables [3] because deliverables such as documents might require revisions and information flows may cause iteration.

Manuscript received July 29, 2019; revised December 25, 2019, March 18, 2020, and May 11, 2020; accepted May 28, 2020. Date of publication July 10, 2020; date of current version June 8, 2022. Review of this manuscript was arranged by Department Editor T. Daim. *(Corresponding author: Laura Maruster.)*

Laura Maruster is with the University of Groningen, Faculty of Economics and Business, Nettelbosje 2, 9747AE Groningen, The Netherlands (e-mail: l.maruster@rug.nl).

Alex Alblas is with the Innovation, Technology, Entrepreneurship & Marketing (ITEM) group, School of Industrial Engineering at Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (e-mail: a.a.alblas@tue.nl).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TEM.2020.3000861

In this article, we examine a specific NPD process model for ECM. ECs are caused by a glitch or a mistake in a component, or new manufacturing procedures, which might require a change to the design of that component.

To investigate the ECM process model, we compare the standard processes reported in the literature and the actual processes conducted at a company. Irrespective of company or product, the EC process is mostly perceived as a standard process, involving similar phases. However, there seem to be significant differences between the "official" documented process and the actual executed process [4]. In addition, changes can take much longer than anticipated [2], thus hindering process steps. A process analysis allows managers to disentangle the various steps and investigate the drivers and factors that influence cycle time. Large gaps between the standard and actual process may imply inefficiencies. For instance, frequent back and forth iterations between some process steps suggest *congestion*, while lots of different steps in the handling of a single EC imply *complexity*. Also, an EC request can range from very simple (changing a parameter in software) to very complex (redesigning an entire part), which will be reflected in the duration of completing an EC. Knowledge of the different types of EC is insightful because it provides information about the actual volumes of ECs and their associated key performance indicators (KPIs).

In order to gain a good understanding of the EC process, and identify or eliminate its inefficiencies, we propose a data and process mining framework for analysis. This involves raising the following research questions.

1) How can the standard and actual EC process model be automatically determined?
2) What EC types can we identify, and which process characteristics can we apply to predict them?

Prior research highlights the need for tailoring and scaling process models [3]. By investigating the process level of EC cycle time, this article aims to provide new insights. By emphasizing the interactions between activities, our proposal is a novel data-driven framework of analysis for investigating EC process models. Process mining is a current technique for automatically extracting business workflow models from event logs [5]. An event refers to a case, an activity, and a point in time. An event log can be seen as a collection of cases and a case can be seen as a trace/sequence of events [6]. Archival data are often unstructured, making it a huge challenge to derive process models [7]. This article pushes further the understanding about automatically scrutinizing the interactions at process level, also by using unstructured data that allows more in-depth analysis.

In the context of NPD and ECM, cycle time plays a vital role for the survival of many companies [7], [8]. A key aspect of cycle time research is seeking antecedents and consequences. This knowledge tells us which major project-level factors determine the cycle time. However, limited knowledge is available on *how* managers should improve cycle time at the *process level* [9]. One explanation for this lack of research is the challenges involved in analyzing the inherently sequential structure of process data. While one event can have a short cycle time in one process stage, the same event may have longer cycle times in a subsequent stage. Specifically, we are interested in whether the standard and actual processes differ a great deal, and in the potential characteristics of the fastest process. Importantly, understanding the link between process characteristics and process complexity will help managers to improve their processes. As such, our approach enables tailor-made solutions for engineering design processes by the use process mining and data science techniques. While most process models assume that activities are known *a priori*, our approach sets the first steps toward an automatic analysis. Our study also contributes to process mining research, presenting a new method to extract detailed events from an unstructured event log, which is necessary for an in-depth understanding of the EC process.

The rest of this article is organized as follows: Sections II and III review the relevant theories on ECM and process mining techniques. Section IV describes the proposed methodology, while Section V reports the results of a real-life case study. Section VI concludes this article.

II. Challenges of ECM Process Models

Process modeling is an essential activity in business process management, and is becoming increasingly popular for many purposes, such as business process redesign (BPR), simulation, total quality management (TQM), and requirement engineering [10], [11].

An EC refers to altering a product or a process after its release to production. ECM focuses on three goals—avoiding or reducing ECs, effective implementation of ECs, and learning from them [12]. The ECM process is deemed important in overall NPD cycle time, because it can become very complex, inefficient, and negatively impact planning, scheduling, and project costs [4], [13]. Moreover, ECs can propagate throughout the entire design, leading to costly rework [14]. Although the research community agrees on the causes of change [4], there is no consensus about how ECs can be effectively and efficiently executed [15]–[17]. It is thus well-established that ECM is an important process area for managing product development in general, and cycle time in particular [9]. Prior research confirms that the best way to improve ECM is by scrutinizing the ECM process in detail [2], [9]. Jarratt *et al.* highlight that innovation in product development leads to ECs and that ECs are a major source for problems in product development [4]. Organizational issues are of the primary concern in managing EC; communication about and coordination of ECs have a great influence on processing ECs effectively and the information flows are often based on incomplete and dynamic information [18]. It is also found that the EC process evolves over time [19]. This highlights the need for flexible EC processes.

The ECM literature presents various prescriptive and descriptive process overviews; see Table I. Table I shows the different formal processes reported in the literature, it highlights the diversity and novelty in approaches and perspectives in ECM. Furthermore, it highlights the need for approaches to address the informal aspects of ECM by referring to some articles. Most studies suggest that these processes have a waterfall character moving an EC through different stages, neglecting the iterative nature, and only emphasizing a few "stereotype" processes [20].

In [21], the design and development process (DDP) is characterized by involving elements of novelty, complexity, and iteration, with implications for process modeling. First, the processes involve novelty, because tasks progress and decisions are made based on incomplete information resulting in procedures to be concretized and adjusted as work proceeds. Second, DDP involves complexity as information flows are highly dynamic and nested. Third, they involve iteration caused by progress and evolving insights on the design and coordinating differences between processes. Since ECM processes inherit the three DDP challenges, the aim of the current study is to develop a process modeling approach that is more sensitive to iteration, novelty, and complexity of processes.

We derive four conclusions from Table I. First, studies examining ECM generally focus on the procedural aspects of the process, mostly ignoring the informal iterative processes that play an important role. While literature provides various prescriptive examples of procedural models, limited studies have addressed the dynamic and iterative nature of the EC management process. Second, two literature reviews highlight the diversity in ECM processes across sectors and contexts, suggesting that a prescriptive "one-size-fits-all" approach might not fit the bespoke needs in a company setting. Most studies in the field of ECM process modeling have mainly focused on the procedural aspects of managing ECs to convey best practices. Third, limited studies have investigated the actual process sequences of activities resulting in discrepancies between the informal activity and formal procedural description of the process [19], [22]. Finally, few studies focus on using advanced computerized data and process mining techniques to mimic the actual EC process. In product development, limited studies have applied computer modeling to investigate the implications of microlevel models of engineering activity [21]. The same holds for ECM process modeling. This calls for ECM process modeling approaches that are more sensitive to the specific context, that combine analytical, computational, and procedural aspects of process modeling on the microlevel [21].

In this article, we provide situation-specific insights based on representing the details of a particular ECM instance. Our work can be classified as a microlevel analytical process model that provides formalisms to assist in the modeling of design knowledge such as EC processes [21]. It is also suitable for a wider range of process areas. It also can be classified as a management science model in that it develops insights by mathematical/computational analysis of a representative case [21].

TABLE I
DESCRIPTIVE LITERATURE OVERVIEW OF ECM

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Maull (1992)[42]	Cross-sectional interviews	Cross-section of industry	Five step process	(1) filter proposal, (2) design investigation, (3) appraise design, (4) authorize change, and (5) execute change.			✓	
Lee (2006) [43]	Case study	Automobile development	Four stage model	(1) Initiating an ECR, (2) evaluating the ECR, (3) issuing engineering change orders (ECOs) to relevant participants, and (4) storing and analyzing the ECOs for management purposes.		✓	✓	
Riviere et al (2003) [44]	Conceptual	No specific sector	Three stage model	*Stage 1 EC proposal*: (1) EC initialization, (2) EC prefeasibility studies, *Stage 2 EC Investigation*: (3) EC impact and feasibility studies, (4) selection and solution definition, and *Stage 3 EC embodiment:* (5) associated document update, (6) new solution identification, (7) new solution embodiment, (8) solution implemented and notified, and documentation updated.				
Veldman & Alblas (2012) [22]	Case study	High-tech/ Gas	Compare to ECM processes	*Gas company*: (1) pre-initiation process, (2) initiation process, (3) first screening (3) second screening, (4) cost estimation, (5) safety check, (6) final decision, (7) final distribution (8) implementation. *High tech company*: (1) request, (2) initiation, (3) validation, (4) assign to project leader, (5) quick scan, (6) make business case, (7) go/no go decision, (8) realize EC, (9) final decision for implementation, (10) implementation.		✓	✓	
Loch & Terwiesch (1999)[9]	Case study	Climate control system development	Seven process steps	(1) Arrival of an EC, (2) Generation of alternatives, (3) simulation of new designs at CAD level, (4) administrative and engineering approval, (5) approval and implementation of purchasing, (6) arrival of modified parts, (7) feedback on the effectiveness of the ECO.			✓	
Terwiesch & Loch (1999) [2]	Case study	Climate control system development	Ten process steps	*Informal problem solving process:* (1) Problem detection and entry into the project management system (2) define scope of the problem and the responsible engineers, (3) meetings between engineers and function teams. Subsequent *formal process* steps same as Loch and Terwiesch (1999).		✓	✓	
Jarrat et al. (2011) [4]	Literature review	Various	Overview, Three-stage six step process	*Before approval stage*: (1) engineering change request raised, (2) identification of possible solutions to change request, (3) Risk/impact assessment of solutions; *during approval*: (4) selection and approval of solution by change control board; *after approval*: (5) implementation of solution, review of particular change process.	✓	✓		
Hamraz, Caldwell &Clarkson (2013)[12]	Literature review	Various	Overview of different perspectives	First part of formal process in line with Jarrat et al. (2004), enhanced with a last process step, review of particular change process, an important learning step proposed by Lee et al. (2006).	✓			
This study	Empirical	High-tech	Actual EC process	Tailoring Product Development Process Models through Process Mining.	✓	✓	✓	✓

(1) Study, (2) research type, (3) sector, (4) type of process, (5) process steps, (6) highlight diversity and novelty in ECM processes, (7) explicitly distinguish between formal and informal process, (8) use empirical methods to extract EC process, (9) use data mining and process mining techniques.

III. PROCESS MINING AND ANALYSIS OF PROCESSES

Process mining emerged as a new discipline linking process science with data science, thus enabling the discovery and automatic analysis of business processes based on event logs [5]. According to recent survey papers [18], [23]–[26], process mining research focuses on developing new techniques (methodologies, tools) and empirical results. As process mining research rapidly grows, and there is an increasing interest also from practitioners and commercial parties, review papers are published at regular intervals. In Table II, the "research type/purpose" (new method/tool, empirical results, review papers) is discussed for a selection of articles. Furthermore, typical process mining tasks are process discovery, conformance checking, process re-engineering, and operational support [23] (see Table II, the "PM task/perspective" column). *Process discovery* refers to learning process models from event data. *Conformance checking* focuses on detecting and diagnosing both differences and commonalities between an event log and a process model [27]. *Process re-engineering* uses both an event log and a process model as input. Here, the goal is not to diagnose differences, but to change the process model, or enrich an existing process model with additional perspectives. *Operational support* directly influences the process by providing warnings, predictions, and/or recommendations.

Next to process task, the process perspective is considered a relevant classification dimension; it refers to control flow, organizational, and case-based perspectives. The *control flow*, e.g., order of activities, and mining this perspective, means finding a good characterization of all possible paths. The *organizational* perspective focuses on the originator field, i.e., which performers are involved and how they are related. The goal is to either structure the organization by classifying people in terms of roles and organizational units or demonstrate the relationships between individual performers (build a social network). The *case* perspective focuses on the properties of cases [23].

In Table II is shown the "sector" where process mining research was executed. In a recent article [18], the authors reviewed 144 research papers where process mining is applied in various economic sectors, such as public administration, finance, healthcare, manufacturing, and education. However, to our knowledge, there has been no contribution of process mining to engineering design or NPD.

An additional dimension (see Table II) that we propose is the "data-driven method" used for research, where different data

TABLE II
DESCRIPTIVE LITERATURE OVERVIEW OF PROCESS MINING

Study	Research type / Purpose	Sector	PM task / perspective	Data-driven method
Weijters & van der Aalst (2003)[45]	New method to discover the process	Not specific	Process discovery / Control flow	Dependency-table
Song & van der Aalst (2008) [46]	New method to determine the social network	Not specific	Process discovery / Organizational	
Lau et al. (2009) [47]	New method to discover process in a network context	Supply chain	Process discovery / Control flow	Association rules
De Leoni et al. (2016) [28]	New method and methodology to include process characteristics	Service (Insurance)	Operational support (prediction) / Control flow	Clustering
Rozinat (2010) [27]	New method / tool to verify process conformance with log	Not specific	Conformance checking / Control flow	Decision Trees
Van der Aalst, (2016) [23]	Review paper to PM methods / tools	Not specific	Process discovery, Conformance checking / Control flow, Organizational, etc.	Clustering, Decision Trees
Van der Aalst (2018) [48]	Review paper to PM methods / tools	Not specific	Process discovery / Control flow	-
Bogarin et al., 2018) [26]	Review paper to PM methods / tools	Education	Not specific	-
Thiede et. al, (2018) [18]	Review paper to PM methods / tools	Various sectors	Not specific	-
Erdogan & Tarhan (2018) [25]	Review paper to PM methods / tools	Healthcare	Not specific	-
Lan et al. (2018) [49]	New method and methodology to determine design process models	Design, Education	Process discovery, Control flow, Organizational	Clustering, Text analytics
This study	New methodology, Empirical results	High-tech: ECR	Process discovery, Operational support / Control flow	Clustering, Decision Trees, Text analytics

mining and text analytics are used in conjunction with process mining techniques. It is common to propose process discovery techniques inspired by existing data mining techniques, see [28]. However, fully fledged data-driven approaches tackling EC real-world context are not so common.

Most of the process mining methods developed so far enable an objective process analysis (diagnosis), which aims to facilitate redesign and improvement. Linking to terminology from [21], these methods can be classified as providing: 1) analytical models and 2) management science/operations research (MS/OR) models. For instance, the research described in [9] proposes a methodology to develop a design process model, to be classified as an analytic model that provides situation-specific insights into design projects.

We can conclude from this table that previous research covers the considered dimensions reasonably well, but there is hardly any existing research that uses empirical research and a combination of process mining and data mining techniques (unsupervised and supervised) to obtain thorough understanding of the underlying ECR processes.

Furthermore, this research attempts to address the three characteristics of design (ECR) processes, namely, novelty, iteration, and complexity [21]. First, it addresses the *novelty* challenge, namely, it does not rely on a domain-specific underlying process architecture, and it is not affected that the process changes over time. Second, the result of the process mining is a process model which can cope with the *iteration* challenge—the resulting process model can be behaviorally and structurally verified via conformance checking [27]. Third, the proposed method aims to determine a detailed (informal) process model, which can be adjusted to the desired level of abstraction. Finally, our study proposes a fully fledged data-driven approach, where existing process mining methods and tools, clustering, decision trees, and text analytics are applied to the engineering design domain.

IV. METHODOLOGY

We use data from a manufacturer of chip-making equipment, consisting of a sample of 5000 ECs executed between 2002 and 2005. We present a framework for addressing the research questions in Fig. 1.

To answer the first research question (Step 1), we develop a novel method which combines text analytics, process mining, and data mining techniques, to identify and analyze the ECM process.

Two processes raise concerns: 1) the standard and 2) the detailed EC process. To determine the standard process, we use the *structured* information in the process log. This information covers eight typical activities—registering a new EC request (1 – New, 1 – In Process), sending for sign-off (2 – Sign-Off), request to approve change (3 – For Approval), change approved (4 – Approved), change request closed (5 – Closed), change request rejected (6 – Rejected), change request on hold (7 – On Hold),

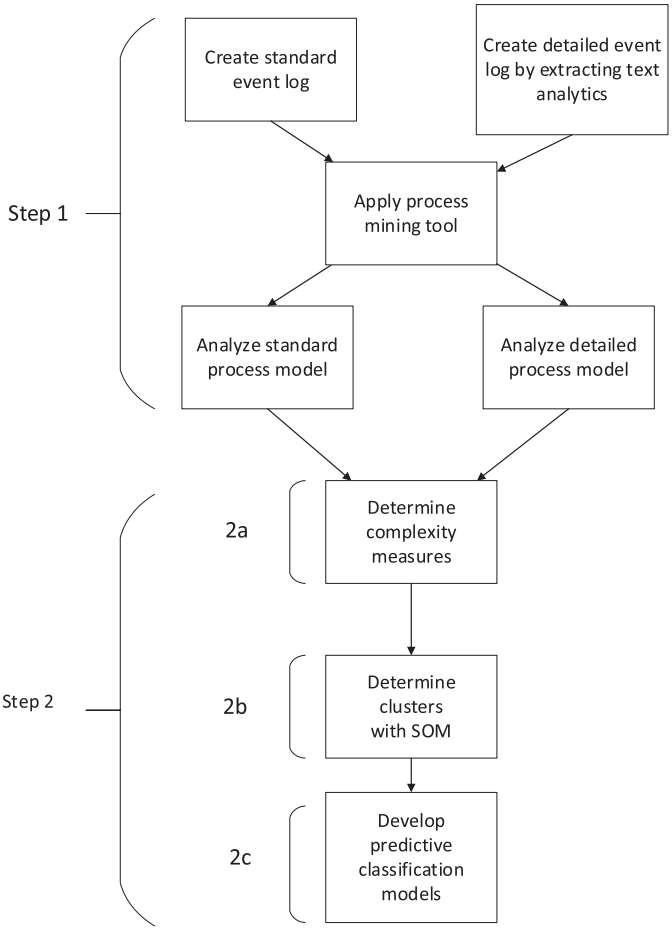

Fig. 1. Proposed framework.

and change request resumed (8 − Resumed). These activities are recorded as such in the original process log. Although this information can be useful to determine process performance, it lacks the description of the actual tasks, and the types of employees and departments involved in the change project. More importantly, this information provides only limited insights into the interactions that take place.

Thus, to gain a better understanding of the process, we investigate additional textual information. The process steps are determined by applying text analytics to *unstructured* data (texts), and subsequently, we identify the detailed EC process using the process mining technique. The process model results are then validated by discussing them with the process owner.

For the second research question (Step 2 in Fig. 1), we aim to operationalize and characterize the EC process complexity and furthermore develop a classification model. We identify the *a priori* known factors (e.g., reason for change, the actors initiating the EC), and the *a posteriori* computed factors (e.g., the frequency of events involved in an EC) which can be used to predict the EC complexity level.

To operationalize *process complexity*, we take into account that this concept can be approached from many angles. For instance, in project management, process complexity emerges due to the fact that "many different actions and states of the world parameters interact, so the effect of actions is difficult to assess" [29]. In NPD contexts, several sources of complexity are identified, including technological, market, development, marketing, organizational, and interorganizational sources [30]. In business contexts, the complexity of process models has been formalized with a variety of metrics; see review paper [31]. In [32], the control flow is considered using business process model and notation (BPMN) process models, where complexity measures refer to the total numbers of separate activities, events (including repeat activities), gateways, data objects, loops and self-loops, etc. Here, we approach process complexity from a control-flow perspective, by considering the frequency of events and lead time.

Employing these two complexity measures, the self-organizing maps (SOMs) unsupervised learning approach [33] is used to determine homogeneous clusters to define EC types. Finally, classification models are created that, based on *a priori* characteristics and steps already followed, predict future events and the approximate duration of EC processes. For developing the classification models, the decision-tree algorithm J48 (an implementation of the well-known decision-tree algorithm C4.5 [34]), and the rule-based algorithm JRip (implementation of the RIPPER algorithm [35]) are used.

The proposed data-driven methodology is novel as includes a broad set of goals, approached with techniques to detect EC process steps, identify ECM process, understand EC types, and predict complexity, all relevant to better understanding EC processes.

V. Results

A. Step 1 Results: Determining the "Standard" and "Detailed" EC Process

Step 1 retrieves the structured information, formatting it as input event log for the process mining tool—Disco [36]. The resulting process model, based on the standard EC event log, consists of eight distinct activities (and 23 454 events); see Fig. 2. The most frequent patterns (dark colored rectangles and thick lines) and also potential deviations are shown on the left side of Fig. 2. The standard, or typical path, is the sequence "New EC" -> "Sign-Off" -> "For Approval" -> "Approved" -> "Closed." You choose the level of detail in the graph by setting the percentage of activities and paths. The mean (median) duration of an activity, or of the transition between two activities, is shown on the right side of Fig. 2.

This is comparable to a "typical" EC process, reported in the literature [4], [22], which follows three main phases: "Before Approval," "During Approval," and "After Approval." These phases seem to correspond to the process phases <"New EC," "Sign-Off">, <"For Approval">, and <"Approved," "Closed">. Overall, 4288 ECs have been accepted and 667 rejected.

We observe that most rejected ECs have a long average duration (cycle time), namely, 26 weeks directly after the process starts. On the other hand, the duration of rejected ECs via the sign-off activity is only approximately 17 weeks (38.2 days + 80.6 days = 118.8 days). The approved ECs following the typical path have a substantially shorter cycle time (38.2 days + 11.1

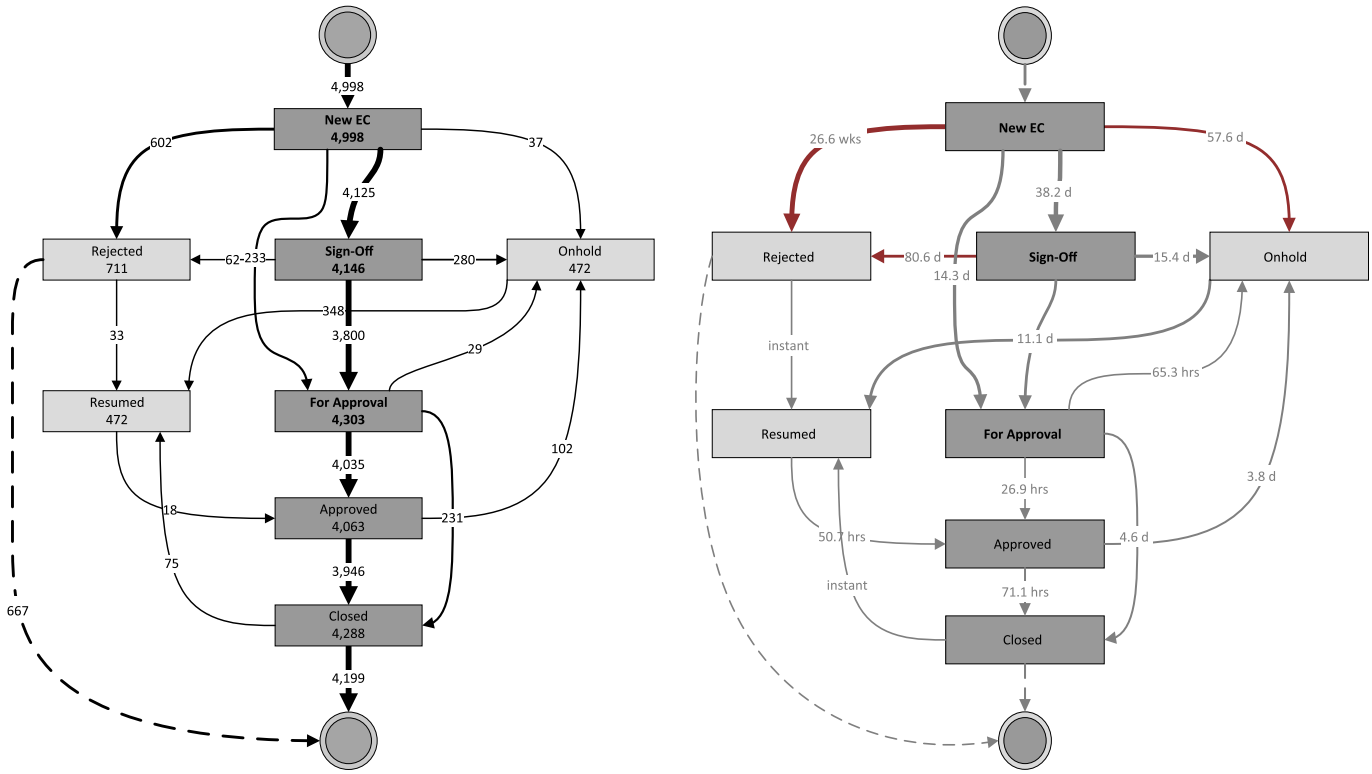

Fig. 2. "Standard" EC process (left—absolute frequency; right—mean duration); all activities and 50% of paths selected.

days + 26.9 hours + 71.1 hours = 53.38 days) than most rejected ECs (26 weeks). An interesting question is then what determines this large cycle time difference between accepted and rejected ECs.

It is worth taking a closer look at this process in order to identify unexpected results. After "Sign-Off," a new EC is usually sent "For Approval," then it can be "Rejected" or put "On Hold." Also, after a new EC arrives, it can be put "On Hold" (37 cases) or "Rejected" (602 cases).

We identify exceptional paths usually associated with long lead times; 348 EC cases are "Resumed" after the "On Hold" activity, which raises further questions about the ECs following this path. Apparently the "On Hold" activity seems to delay the resolution of 472 ECs (out of ∼5000) by almost 2 weeks on average (and if we consider the "Resume" activity, by additionally 11.6 + 12.7 days, approximately 24 days). This should be an issue of concern. If we bear in mind the 37 cases put "On Hold" for almost 2 months (57.6 days), it is definitively important to better understand the process.

Although the standard process does not seem very complex, it too raises concerns. It only allows a shallow analysis of the entire process by computing data such as frequency of activities and cycle time statistics (mean, median, total) for the overall process or each separate activity.

Thus, a more detailed approach is needed, by considering the *unstructured* log information. This information is a mix of data on the executed activities, such as date, executor, reason for change, change description, transfer to another department, as shown in Table III.

To interpret this unstructured material, the text is parsed (using an R script) and the information necessary for process mining is extracted and copied to a new, more detailed event log (in Table III, the relevant information for process mining is marked bold). The "######" separator is considered as delimiting the executed activity.

The detailed event log includes information about process instance ID, date and time, the executed activity/interaction label (or the department where the EC is sent—for example, type of EC request "ECR Status: 1 – New"), executor name ("Johnny"), EC status (Accepted/Closed or Rejected), but also the reason for the EC request ("reason for change", e.g., New Part, Functionality, Reliability, etc.), and the department ("source") that issued the EC (for example "development"). Moreover, information about the "product type" (such as "A") and "subsystem" ("system electrical layout and environmental electronics") is available. An example of an EC process instance in the detailed log is presented in Table IV. The detailed event log is cleaned by filtering irrelevant and noisy activity labels; ultimately, 89 distinct activities and 99 611 events remain. Please note that an activity can happen multiple times, thus it will result in different events. For instance, the activity "ECR summary" determines two events in the log presented in Table IV.

In Fig. 3, the detailed EC process is obtained by applying the process mining tool to the detailed event log. In order to get a readable graph, only 30% of the most frequent activities are represented in the process map.

The standard process steps can be mapped to the corresponding steps in the detailed process: "1 - New, 1 – In Process"

TABLE III
EXCERPT OF ONE EC CASE WITH SELECTIONS OF ORIGINAL UNSTRUCTURED TEXT

Johnny: 11 Dec 2002 13:30:23 ECR Summary: RELEASE CABLES AND BUNDLES FOR TR UPGRADE PIM Status: Draft Product Type: MA: 331 Other: T.A./A.M./A.P. originator: George NPL Subsystem: System Electrical Layout & Environmental Electronics) Source: Development
Johnny 11 Dec 2002 ECR Status: 1 - New Other: Reason for Change: New Part (R1, R2) Product Fam.: NM ECR type: Normal Copy Exactly (C-E): NA
Nikky: 11 Dec 2002 15:30:01 PL: 11282 (Key) ==> Mike
Ellen: 11 Dec 2002 16:12:45 ECR Status: 1 - In Process ==> Test of Solution: ==> NA
Bas: 15 Jan 2003 9:58:00 Impl. Plan NPL: Yes ==> Sign-off NPL: ==> Yes Date sign-off NPL: ==> 19 Dec 2003
Johnny: 1 Feb 2003 13:13:09 ECR Status: 2 - For Sign-off==> 3 - For Approval Sign-off PIM: ==> Yes Date sign-off
Frank: 1 Feb 2003 14:35:14 ECR Summary: RELEASE CABLES AND BUNDLES FOR TR UPGRADE ==>
Anke: 25 Mar 2003 17:00:01 ECR Status: 3 - For Approval ==>
Anne: 17 Apr 2003 14:05:07 ECR Status: 4 - Approved ==> 5 - Closed
George: 30 May 2003 15:19:47 PIM Status: Dispatch ==> PIM Archived
Ellen: 6 Jul 2003 10:02:28 END

TABLE IV
EXAMPLE OF ONE EC CASE

ID	Executor	Date/Time	Activity	Final Status	EC Reason
1	Johnny	11 Dec 2002 13:30:23	ECR Summary	Closed	New Part
1	Johnny	11 Dec 2002 13:30:23	ECR Status: 1 - New	Closed	New Part
1	Nikky	11 Dec 2002 15:30:01	Project Leader (PL)	Closed	New Part
1	Ellen	11 Dec 2002 16:12:45	ECR Status: 1 - In Process	Closed	New Part
1	Bas	15 Jan 2003 9:58:00	Implementation Plan New Product Logistics (NPL)	Closed	New Part
1	Johnny	1 Feb 2003 13:13:09	ECR Status: 2 - For Sign-off	Closed	New Part
1	Frank	1 Feb 2003 14:35:14	ECR Summary	Closed	New Part
1	Anke	25 Mar 2003 17:00:01	ECR Status: 3 - For Approval	Closed	New Part
1	Anne	17 Apr 2003 14:05:07	ECR Status: 4 - Approved	Closed	New Part
1	George	30 May 2003 15:19:47	PIM Status: Dispatch	Closed	New Part
1	Ellen	6 Jul 2002 10:02:28	END	Closed	New Part

corresponds to "ECR Status 1 – In Process," "2 – For Sign-off" to "ECR Status 2 – For Sign," "3 – For Approval" to "ECR Status 3 – For approval," "4 – Approved" to "ECR Status 4 – Approved," "5 – Closed" to END ("END" represents an artificial final event, which does not exist in the real event log).

The detailed process model (see Fig. 3) provides new insights. Namely, we can show that after a new EC enters the system, it gets the status "In Process," and a description "ECR Summary." Further, it is sent either to the production engineer "PE" (948 times), the electrical engineering department "EE" (1407 times), a project leader "PL" (1667 times), or to departments "Other1" (1073 times), "Other2" (1468 times), or explicitly via email "Other e-mail (1384 times)."

Note that activities "On Hold" and "Resume," belonging to the standard process, disappeared, and were replaced by more meaningful process steps (e.g., "In Process," "Screening," "Screened").

Analyzing the execution time of the "detailed" EC process, we see that in the initial phase (between "ECR Summary" and "PIM Status: Draft"), some departments need more time to analyze ECs than others, namely, the NPD project leader ("PL") scores the highest, with an average of 18.4 days (the absolute frequency of EC cases is also the highest—1667 cases). The next phase "Sign-Off" and "Implementation Plan" activities executed by different departments take much less time (in hours) than the activities before "PIM Status: Draft." Finally, the last phase after sending "For Approval" is streamlined. However, it is worth noting that after an EC is approved, it takes more than one week to finalize "PIM Status: Dispatched." The overall statistics in Table V show that the mean and median case duration (cycle time) of the standard and detailed EC processes differs significantly. This is because the detailed view contains more precise information about the final process steps. Based on median and mean values, it is apparent that projects take on average longer than one month to complete.

B. Step 2 Results: EC Types and Implications for EC Processes

Step 2 of the proposed framework (see Fig. 1) identifies the types of EC. An EC request can range from very simple (e.g., changing a parameter in software) to very complex (e.g., redesigning a complete part), which will be reflected in the time it takes to complete an EC. Since the data do not reveal the extent or complexity of the EC, we aim to determine different EC types from a complexity (as control flow) perspective, and investigate in which way EC types impact process duration.

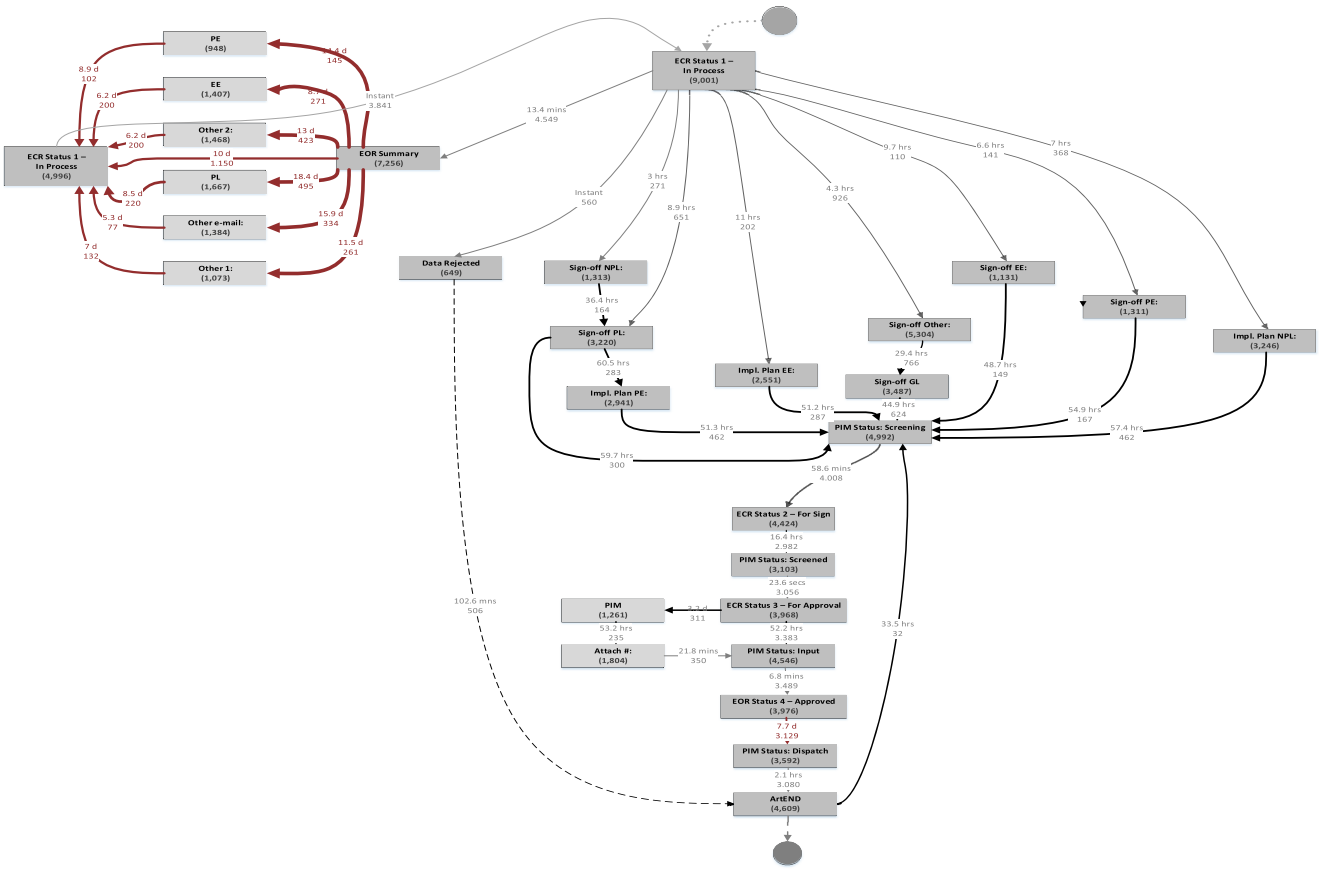

Fig. 3. "Detailed" EC process (mean duration)—only 30% of activities.

TABLE V
BASIC STATISTICS FOR THE STANDARD AND DETAILED EC PROCESS (INCLUDING ALL POSSIBLE PATHS)

Statistics	Standard process	Detailed process
No of activities	8	89
No. of events	23454	99611
Median case duration	37 days	45.1 days
Mean case duration	71.9 days	90.4 days
Number of variants	34	4593

C. Step 2a. Complexity Measures: Cycle Time, Frequency of Events, and Variants

Process duration is usually measured in terms of process cycle time (lead time or throughput time). Cycle time is related to the frequency of events in a process instance, as the latter determines the lead time. A large number of process steps may determine a short lead time, but a small number of process steps can also determine a long lead time.

Besides the numbers of events and lead time, process patterns or *variants* can reveal another dimension of complexity. A process variant is a unique path from the beginning to the very end of the process [36]. Different variants can exist in business processes. For instance, it could be useful to retrieve process variants for identifying preferred work practices [37], [38].

Here, we use the number of different process variants as an additional measure of process complexity, along with the frequency of events and lead time. Identifying process variants is a useful way to distinguish typical/nontypical or exceptional paths.

We identify 34 path variants for the standard event log. The most occurring variant—"Variant 1"—consists of the most typical process sequence, and includes five events: "New EC," "Sign-Off," "For Approval," "Approved," and "Closed," accounting for 72.51% of all variants (3625 cases). The next "Variant 2" includes two events "New EC" and "Rejected," accounting for 12.02% (601 cases). These two variants account for almost 85% of all variants, which is not so surprising; the first variant refers to the most typical path of "accepted" ECs, and the second refers to "rejected" ECs.

"Variant 3" involves seven events and includes "On Hold" and "Resumed" activities, covering 5.15% (258 cases). We provide details of the six variants that cover more than 50 cases (chosen as threshold value) in Table VI.

Determining the patterns based on the detailed event log yields 4593 variants. Unsurprisingly, this number is much larger than the standard process with only 34 variants. This detailed process view provides more insights into the process complexity. The number of variants for standard and detailed process views will be used to determine and characterize EC types.

TABLE VI
DETAILS OF THE SIX VARIANTS (EACH COVERING > 50 CASES): THE STANDARD PROCESS VIEW

Variant	Path	Cases covered	# Events
V1	"New EC", "Sign-Off", "For Approval", "Approved" and "Closed"	3625 (72.51%)	5
V2	"New EC" , "Rejected"	601 (12.02%)	2
V3	"New EC", "Sign-off", "On hold", "Resumed", "For Approval", "Approved", "Closed"	258 (5.15%)	7
V4	"New EC", "For Approval", "Closed"	220 (%)	3
V5	New EC", "Sign-off", "For Approval", "Approved", "Onhold", "Closed", "Resumed"	72 (%)	7
V6	"New EC", "Sign-off", "Rejected"	52 (%)	3

TABLE VII
CHARACTERIZATION OF CLUSTERS DETERMINED WITH THE SOM CLUSTERING METHOD

Cluster	No. instances	Events frequency (Standard)	Events frequency (Detailed)	Leadtime (Standard) (days)	Leadtime (Detailed) (days)	Characterization EC type
Cluster-0	427 (9%)	6.9904[a] (6; 7)[b] 6.993[c] 0.0836[d]	25.9521 (8; 40) 27.3138 4.9467	105.40 (1; 756) 94.41 87.50	83.53 (2; 865) 97.60 92.22	many events (S, D), medium duration (S, D) *–> complex process*
Cluster-1	1477 (30%)	4.9989 (4; 5) 4.9993 0.026	27.5484 (20; 41) 26.4807 3.6469	60.09 (0; 1465) 76.85 92.25	98.76 (3.96; 1884) 105.22 148.72	medium frequency of events (S), many events (D), medium duration (S, D) *-> moderately complex process*
Cluster-2	2214 (44%)	5.0399 (4; 7) 4.9864 0.1497	16.5513 (3; 22) 18.4693 3.2376	54.08 (0; 534) 38.47 41.72	59.88 (0; 1912) 55.62 96.41	medium frequency of events (S, D), relative short duration (S, D) *-> standard process*
Cluster-3	874 (18%)	2.0786 (2; 4) 2.3158 0.4676	13.3997 (1; 34) 8.9565 5.0414	163.47 (0; 1648) 137.6 237.22	140.46 (0; 1649) 149.97 249.69	few events (S, D), long duration (S, D) *-> long, but simple process*

[a]Central value.
[b]Minimum and maximum.
[c]Mean.
[d]Standard deviation.

D. Step 2b. Determining EC Clusters With SOMs Method

For determining EC request types, we aim to develop homogeneous and clear-cut clusters. We consider homogeneity from the process complexity perspective and the clustering variables are: 1) lead time and 2) frequency of events, for the standard (S) and detailed (D) process view, respectively. We include both standard and detailed process views because two process instances may have similar steps in the standard process view, yet be different when taking into account the detailed process view.

We do not consider the process variants as input for clustering because this complexity measure can be used later for classification purposes. Variants can be considered *a priori* characteristics of the process. For example, a still unfinished process including three steps "New EC," "Sign-Off," "For Approval" can be assigned to variants V1 or V5, but not to V2, V3, V4, or V6. Knowing *a priori* which steps have been taken, even though the process is not yet finished, may help to predict which process steps will follow, and thus, to which variant a process instance likely belongs. The complexity measures "frequency of events" and "lead time" can be seen more as *a posteriori* process characteristics (variables only known when the process instance has finished), and are therefore used for clustering.

For clustering, we apply the SOMs unsupervised learning approach. SOM is an effective method for converting complex, nonlinear statistical relationships between high-dimensional data into simple geometric relationships on a low-dimensional display. These two aspects, visualization and abstraction, can be utilized in various ways for complex tasks such as process analysis, machine perception, control, and communication [33]. For this work, we use the Weka implementation available in Knime [39]. Unlike other unsupervised clustering methods such as k-means, SOM is useful as it is not necessary to specify *a priori* the number of clusters.

Based on the analysis of EC types and implications for EC processes (Step 2), we come to the following conclusions.

1) *The method of analyzing process types enables a definition of clusters.* We obtain four clusters corresponding to four EC types, including 9%, 30%, 44%, and 18% of the EC. This clustering model typically has relatively homogeneous clusters regarding the frequency of events (for both standard and detailed views), and is relatively heterogeneous regarding duration (also standard and detailed views). Based on the characterization in Table VII, we label the four clusters; see column "characterization EC type."

2) *The clusters enable an analysis of performance differences.* It seems that ECs involving many events do not necessarily imply long duration. We also observe that most ECs belong to Cluster-2 (44% of cases), which can be labeled the "standard process" cluster. This cluster includes ECs with 5–18 events on average (for standard and detailed views, respectively). The mean duration is 38 and 56 days (for standard and detailed views, respectively). Thus, a standard process will most likely follow five events, and take between five and eight weeks. This large variation is not surprising, given that this cluster can include small but also large EC projects.
   The cluster with the smallest number of ECs is Cluster-0 (9% of cases), labeled as "complex process," which includes ECs with 7 and 27 events on average (for standard and detailed views, respectively). A complex process is likely to include on average 7 (standard view) and 27 events (detailed view). The mean duration is around 13–14 weeks (94 and 97 days, for standard and detailed views). It is also remarkable that the duration is comparable in the two views (94 and 97 weeks), despite the very different average amount of events (7 and 27).
   Cluster-1 refers to the "moderately complex process," which is almost one-third of the cases (30%). A process in this cluster will most likely involve 5 and 26 events, and takes on average between 77 and 105 days, in the standard and detailed view, respectively. This cluster seems to comprise cases which follow the five typical steps, but on closer inspection, actually involve more events. The events visible in the detailed view explain why the lead time is so long, namely, 105 days on average.
   Cluster-3 "long, but simple process," refers to the remaining 18% of cases, involving the fewest events (two and nine for standard and detailed view, respectively). This cluster groups the cases with the longest lead time (138–150 days, corresponding to 20–21 weeks). Since rejected cases usually have a long lead time, a hypothesis is that such cases will mainly be found in this cluster. Step 3 involves a separate analysis of "Rejected" ECs.
3) *The method of analysis enables prediction of future performance.* It would be useful to use the four clusters developed so far for prediction purposes. Thus, in the next section, we investigate which *a priori* characteristics (known before the actual EC process has finished), and others, such as process variants, can be used as predictors for developing classification models to classify EC types. In other words, knowing that an EC's "reason for change" is a "New Part," means it belongs to the "subsystem" "mechanical layout"; and as two events have occurred so far in the process, namely, "New EC" and "Rejected," in which cluster (of the four possible) is it likely to belong?

E. Step 3. Develop Predictive Classification Models

Classification models (also called classifiers) are based on algorithms which enable the mapping of input data to a category. Classifiers are represented as rules or trees, which can be used for decision-making and data compression. At the previous Step 2b, we developed four categories of EC. For EC data, we identify four *a priori* known EC characteristics which can be used as predictive features (nodes), namely, "reason for change," "source," "product type," and "subsystem." "Reason for change" refers to the issue which determined the EC request, and can be due to a "New Part," or a repeated "Old" EC. "Source" refers to the department that issued the EC. The "product type" refers to the product family ("NA" or "MA"), and "subsystem" to the category in which the product belongs ("electrical integration," "mechanical layout," "computer systems," or approximately 30 others). Also, the variant type is used as the fifth predictive feature, based on the standard view. For the standard view, significantly less variants were determined than for the detailed view (34 and 4593, respectively). From the 34 variants in the standard view, the six variants covering more than 50 cases are used as categories of the variant type feature (see Table VI). The class or predicted value (the "leaves" in the decision tree) is the four clusters obtained previously.

F. Development of Classification Models

To develop classification models, again we use Knime software. As there are restrictions with numerical/nominal values, not all Knime algorithms that develop classifiers can be applied. Since we aim to use categorical variables as predictors, we need to apply Knime algorithms that can handle nominal values. Since we require efficient algorithms which produce readable trees or rules for further decision-making, we find suitable candidates in the decision-tree algorithm J48 (an implementation of the well-known decision-tree algorithm C4.5 [34]), and the rule-based algorithm JRip (implementation of the RIPPER algorithm [35]). Both are optimizations of their original algorithm variant.

The first class of experiments creates a decision tree J48 model and a rule-based JRip model on 70% of the total dataset (3494 items), forming the training dataset. As usual, the training data are used to develop the classifier and the test data (the remaining 30%) to assess the classifier's performance.

The decision-tree algorithm results in a big tree (165 branches), which is hard to interpret. The rule-based algorithm JRip produces a more compact model with 19 rules, making it easier to interpret, and so we focus on this model. Table VIII shows the best rules of the JRip classifier for each cluster.

The classification model based on the JRip algorithm produces some interesting rules. For Cluster-0, the best rule involves the variable Variant 3, which applies to this cluster in 187 cases, called positive instances (there is one case where the rule fails to classify correctly, called a negative instance). Variant 3 comprises seven events ("New EC," "Sign-Off," "On Hold," "Resumed," "For Approval," "Approved," "Closed"). This rule seems to suitably characterize this cluster, as it covers more than a quarter of the entire dataset (187 positives and only one negative out of 427 items). The ECs of type "Variant 3" seem to have a complex process path (Cluster-0). When discussing the

TABLE VIII
SELECTION OF JRIP RULES FOR THE FOUR CATEGORIES OF EC TYPES

Cluster	Best rule(s)	Total number of rules	Coverage
Cluster-0 ***complex processes***	*Variant-Small = Variant3 => Cluster0*	11 rules	187 / 1
Cluster-1 ***complex, relatively standard processes***	*subsystem = Electrical-Integration => Cluster1* *subsystem = Component A-Handling => Cluster1* *subsystem = Handling-Transport => Cluster1*	3 rules	409 / 162 69 / 24 71 / 27
Cluster-2 ***standard processes***	*(no rules, majority class)*		2005 / 689
Cluster-3 ***long, but simple processes***	*Variant-Small = Variant2 => Cluster3* *Variant-Small = Variant4 => Cluster3* *Variant-Small = Variant6 => Cluster3*	4 rules	423 / 0 156 / 0 40 / 0

characterization of Cluster-0, we already see that on average, seven events (a minimum of six and maximum of seven) were involved (standard view) and 18 events (a minimum of eight and maximum of 40) in detailed view.

ECs in Cluster-1 are relatively well-predicted by the *a priori* variable "subsystem." Remarkably, the rule for this cluster only refers to the variable "subsystem" and does not include any variant type. Apparently, the "Electrical-Integration" ECs have a complex, but relatively standard process (409 positive versus 162 negative instances). The same holds for "Component A-Handling" ECs (69 positive versus 24 negative) and "Handling-Transport" (71 positive versus 27 negative). For Cluster-3, three rules which cover 619 items (out of 874) involve three variants: Variant 2 with two events "New EC" and "Rejected," Variant 4 with three events "New EC," "For Approval," "Closed," and Variant 6 with three events "New EC," "Sign-Off," "Rejected." We note that for this Cluster-3, two rules involve variants which include "Rejected" ECs. The ECs verifying the three rules associated with Cluster-3 are likely to have a long, but simple process. Also, these rules do not have negative items (i.e., zero failed classifications). Cluster-2 is the majority class referring to standard EC processes, matching many positive items (2005), but also with a high number of negative failures (689).

G. Quality of the JRip Classification Model

It is important to not only examine the number of rules and coverage ratio of positive and negative cases, but also the performance of the JRip rule-based classification model.

This can be assessed by evaluating the performance of the classification model on the remaining 30% of data (1498 − testing dataset), and by carrying out a tenfold cross validation. K-fold cross validation is a model validation technique for assessing how the results of applying a classifier will generalize for an independent dataset [40]. The usual performance measures are considered, namely, accuracy, error, precision, recall, and F-measure. Recall is the proportion of real positive cases that are correctly predicted positive, and precision denotes the proportion of predicted positive cases that are correctly real positives [41]. The F-measure is a combination of precision and recall, as it is computed as the weighted harmonic mean of precision and recall. For the testing data, 27.303% are incorrectly classified (1089 correctly and 409 incorrectly), which means an accuracy of 72.697%.

For the tenfold cross validation, the aggregated accuracy is 73.5%. Table IX presents the confusion matrix for the aggregated tenfold validation experiments, and the precision, recall, and F-measures.

The aggregated error rate of the tenfold cross validation slightly exceeds 25%. We can interpret this as modest accuracy performance. If we examine the performance for each cluster, we see that the precision is maximum for Cluster-3 (1) and very high for Cluster-0 (0.993). Recall is also very high for Cluster-3 (0.995) and Cluster-0 (0.972), and moderate for Cluster-2 (0.879). These results yield a composite F-measure of 0.988, 0.746, 0.4, and 0.982 for Cluster-3, Cluster-2, Cluster-1, and Cluster-0, respectively. Based on these results, we can be confident about the accuracy of the prediction rules for Cluster-0 and Cluster-3. Given the model's overall modest accuracy of almost 75%, we conclude that it is useful for further predictions.

Based on the predictive classification models (Step 2), scenarios *can be derived about an EC's future behavior.* In other words, for a given EC it is possible to predict future events and approximate duration based on some *a priori* characteristics or steps already followed in the process. For example, if three activities have been executed for an EC, namely, "New EC," "Sign-Off," and "On hold," that EC will probably be Variant 3. Consequently, four activities are likely to follow ("Resumed," "For Approval," "Approved," "Closed"), making this a complex process, belonging to Cluster-0. As previously described, in Step 2b, we characterize the four clusters regarding average lead time. The average resolution time in Cluster-0 is 94–97 days (∼13 weeks).

An electrical integration EC will likely end up in Cluster-1, and be a complex process with five standard steps (or 26 detailed ones), and a large spread in terms of events (standard 77 to 105 detailed days).

An EC in Cluster-3 is not so useful for predicting the next steps in the process, but can provide insights into ECs of Variants 2, 4, and 6. Variant 2 and Variant 6 refer to rejected ECs (involving 2 and 3 events, respectively), and Variant 4 refers to accepted ECs with three events ("New EC," "For Approval," "Closed"). These ECs typically have few events (between two standard and nine detailed events), but very long lead times (138–150 days, that is 20–21 weeks). It is especially striking that (423 + 40 =) 463 out of ∼690 total rejected ECs fail here. It would be interesting to investigate in depth the rejected ECs and find out why they take so long.

VI. DISCUSSION AND IMPLICATIONS

Our study proposes a new framework of analysis for investigating NPD processes. While prior research provides valuable knowledge on the main stages of the EC process, this article contributes to understanding the details of this process. Moreover, *a priori* characteristics, or initial process steps enable predictions about the next process steps, process duration, etc.

TABLE IX
CONFUSION MATRIX AND PERFORMANCE MEASURES FOR THE CLASSIFICATION MODEL, TENFOLD CROSS VALIDATION

10-fold Real class	Predicted class				Precision	Recall	F-measure
	Cluster-3	*Cluster-2*	*Cluster-1*	*Cluster-0*			
Cluster-3	870	3	1	0	1	0.995	0.998
Cluster-2	0	1937	274	3	0.648	0.879	0.746
Cluster-1	0	1049	428	0	0.621	0.295	0.4
Cluster-0	0	11	1	415	0.993	0.972	0.982

Our data-driven approach considers real-world data of a mixed structure (date/time relating to process steps) and unstructured (free text description of EC provided by engineers).

Standard and actual EC processes. The answer to our first research question, concerning automatically determining the EC process by applying the process mining technique, reveals various aspects.

First, we confirm that a "typical" or standard path exists, namely, "New EC" -> "Sign-Off" -> "For Approval" -> "Approved" -> "Closed," which takes on average 54 days. Exceptional paths also seem to exist, and if these are followed, they substantially increase the lead time. Moreover, the standard process view shows activities with an unclear meaning ("Resume," "On Hold").

Second, the standard process reveals problematic aspects, such as back and forth iterations, which can indicate congestion. Repetitive behavior may also indicate rework, which usually represents undesirable behavior in the context of the Lean Six Sigma manufacturing paradigm.

Third, the detailed process view reveals interesting information about the sequence of the process steps. For instance, we would expect to systematically see "Implementation Plan" followed by "Sign-Off" activity, and made available to the relevant departments, namely, PL (NPD project leader), NPL (new product logistics), PE (product engineering), EE (electrical engineering), and GL (NPD group leader). However, this order is not what we would expect, and its connection with the department is mixed (first, "Sign-Off PE" followed by "Implementation Plan PL").

Fourth, we note the discrepancy that sometimes occurs between the lead time in the standard view and detailed view. We observe that after an EC is approved, it takes more than one week to dispatch that EC, which is questionable. In the standard view, apparently after approval ("Approved" in Fig. 1), it takes only 71 h (three days) to close an EC. One obvious explanation is that only the most important activities are recorded in the standard view. Since the detailed view is more complete, the lead time is also longer. This illustrates that comparing the two views can provide a better understanding of the actual EC process.

Last, but not least, the detailed process view pinpoints the problematic steps in the process, such as bottleneck departments, whose processing time is longer than expected. For instance, the detailed process view in Fig. 3 shows that the PL department gets many ECs (1667) and takes the greatest associated time to receive but also to resolve a change (mean duration: 18.4 and 8.5 days). The PL department's load could therefore possibly be reconsidered.

Definition of EC types. Since ECs obviously differ in terms of complexity, we use the frequency of events and EC duration as indicators of complexity. The answer to our second research question focuses on grouping ECs in homogeneous clusters from a complexity perspective. Using the Knime implementation of the SOMs unsupervised clustering method, we obtain four clear-cut clusters corresponding to four EC types. Each cluster contains EC's with a certain frequency of events and of certain duration (with corresponding measures for means, minimum, maximum). Two types can be seen as opposite, the "standard process" (Cluster-2, 44% instances) and "complex process" (Cluster-0, 9% instances), and two types are a combination of the previous two, namely, "moderately complex process" (Cluster-1, 30% instances) and "long, but simple process" (Cluster-3, 18% instances). A general conclusion is that the average lead time within every cluster is quite long, even for a "standard process" (longer than one month). For the "complex process" cluster, the average lead time is more than 94 days.

Classification model for predicting EC type. Furthermore, we investigate which *a priori* characteristics and others, such as process variants (frequent patterns), can be used as predictors for developing classification models to classify EC types. In other words, we want to develop models that, based on *a priori* characteristics and steps already followed, predict future events and the approximate duration of EC processes.

We experiment with two types of classifiers—decision trees and classification rules (the Knime implementations of J48 and JRip algorithms). Because the decision-tree model results in a very large and hard-to-read tree, we prefer to use the rule-based model provided by the JRip algorithm. The overall classification error of this model slightly exceeds 25%, which can be interpreted as modest accuracy performance. However, if we inspect the performance measures separately for each cluster, some clusters score higher than others. Namely, the F-measure associated with rules for Cluster-0 is 98.2% and for Cluster-3 is 99.8%, which implies a very high accuracy.

Zooming into the obtained rules, if for example, three activities have been executed, namely, "New EC," "Sign-Off," and "On Hold," the considered EC is probably Variant 3. This implies that four activities are likely to follow ("Resumed," "For Approval," "Approved," "Closed"), making this a complex process, belonging to Cluster-0. If the process steps resemble Variant 2, Variant 4, and Variant 6 patterns, the EC can belong to Cluster-3. Variant 2 and Variant 6 refer to rejected ECs (involving two and three events, respectively), and Variant 4 to accepted ECs with three events ("New EC," "For Approval," "Closed"). An EC belonging to Cluster-3 is likely to end up with few events

(between two standard and nine detailed events), but a long lead time (from 138 to 150 days, that is 20–21 weeks). Concerning ECs in Cluster-2 ("standard process"), since Cluster-2 is the majority class, there are no rules associated with it. In other words, if an EC cannot be assigned to Cluster-0, Cluster-1, or Cluster-3, by default it can be associated with Cluster-2, and the characteristics associated with this cluster apply.

Furthermore, predictions can be made for ECs exhibiting certain *a priori* characteristics. For instance, an electrical integration EC will likely end up in Cluster-1 (complex, relatively standard process), be a complex process with five steps (based on the standard process view) or 26 (considering the detailed view), and have a large spread in terms of events (77 days based on standard view, and 105 days based on detailed view).

A. Research Implications

New perspective on analyzing the impact of ECs on manufacturing and service processes. While most research analyzes the effects of ECs at a design and design process level, more studies are required to investigate the impact of ECs at the manufacturing and service level [12]. Moreover, in our case study setting, we can examine not only the initiation and assessment phases of ECs, but also their implementation in a development and production setting. Implementation can be improved by analyzing the process properties of ECs, and their impact on the production and testing process, allowing for mitigation and process improvement actions. Being able to mine and analyze the EC process from initiation to final implementation provides insights about the effects of ECs on product and manufacturing process quality. While most impact assessments are done *a priori*, our methods allow for *post hoc* analysis of impacts.

New method to study design processes. The growing amount of data thanks to robotics and sensor technology offers great opportunities in the area of product and process optimization. Production processes are becoming less linear. The growth in data can be used for learning and process improvement. This requires smart methods for knowledge transfer from production and service to development. EC process mining and incident data can yield valuable knowledge. For example, who should be involved in what kind of ECs? How can development learn from this? In addition, we must carefully consider the link between the product and process architecture. Which incident should be assigned to which module? Analyzing EC, incident, and test data in a smart way can improve product and process design.

We present a new method to extract detailed process events from an unstructured event log, which enables the application of a process mining tool. To our knowledge, this is the first attempt to apply process mining in the context of NPD processes. The data-driven approach demonstrates the benefits of corroborating different standard data mining algorithms (unsupervised and supervised) and the process mining technique, on a large corporate dataset. The success of a data-driven approach does not lie so much in the choice of algorithms, but in identifying the nature of the problem and the appropriate solution type. That is, corporate data include sequential data which can be used to uncover patterns that can be interpreted by decision makers through visualizing and representing process models, and data that can be operationalized by performance aspects such as complexity, enabling the grouping and classification of similar process instances. Despite the variety of new tools and algorithms, our claim is that standard algorithms as those used in this article can produce insightful results in fields such as ECM.

In high-tech settings ECs can be caused by evolving insights on product and process technology. A glitch or a mistake in a component, or new manufacturing procedures, might require a change in the design of that component (i.e., EC). Consequently, ECs are variate, complex, uncertain, requiring bespoke processes, and thus, customization. Our method can handle these challenges and also supports process customization. As such, managers can use the method to identify a standard process with some common steps, and derivative processes that inherit the common parts, but also have specific steps.

Enabling people and process-oriented analysis to facilitate the avoidance of ECs. Engineers are better able to analyze the sequence of process steps and the various people involved in each process step. This enables the development of much needed people-oriented measures and the analysis of organizational issues related to ECM [12]. Examples are the analysis of congestion points, communication frequency, and centrality of individuals in solving issues. Our method provides a framework of analysis for these kinds of measures.

B. Implications for Management

The proposed method can help managers reconsider problematic process aspects and achieve improvements. The results of process mining can explain process optimization priorities, namely, ECs following a certain path that seems to be associated with long lead times can be thoroughly investigated. Grouping ECs in four clusters is insightful because it provides information about the actual volumes of ECs, their associated KPIs, and can shape a company's strategy. For instance, if a company aims to focus on operational excellence and customer satisfaction, it can make informed decisions on what type of EC processes to enhance. For operational excellence, it would be insightful to have a measure that assesses the performance for different types of EC. It would be interesting, for example, to focus on Cluster-3 ("long but simple process") projects and attempt to decrease the lead time. Since for the central value (median) for lead time is close or above two months for all four clusters, a shorter lead time is, of course, essential for customer satisfaction. ECs emerge both in a product development and manufacturing context. In manufacturing, the handling of ECs is particularly complex and challenging. ECs need to be released at the appropriate timing, they need to be clustered and batched where possible, they need to be planned in production and assembly in the appropriate routing slot, they include upgrades requiring new test procedures, and they even require altering warehouse locations. Especially in the area of computerized techniques, the proposed approach enables knowledge elaboration and process

innovation, facilitating the optimization of EC implementation processes and operators' training.

C. Limitations and Future Research

Our research was executed in the context of one company. Future research is required to replicate our results and validate them in other contexts. Also, unstructured data closer to natural language can lead to a more accurate classification model, and better, more useful rules. Further research will examine variables such as the various departments involved in the EC process. What is more, the proposed approach can be extended to other types of processes, not only NPD projects.

Although the focus of this article mainly was on the EC process management in the front-end and midend of the EC process, future research could address in handling ECs during the back-end manufacturing implementation process. Some challenging issues in the computerized EC management area include automatic detection of loops or iterations in the manufacturing process, impact propagation and implications for testing, operational decision support for clustering and release management, modeling of process knowledge required for EC implementation, process simulation, and so on. Another highly potential area for further research is digital twins. A digital twin is basically a digital replica of an actual product. In the same vein one could create a digital twin for a process, where the digital version is a replica of the actual process and registers changes in the process over its lifecycle. Our method provides opportunities for improvement in the organization of ECs by allowing us to compare the documented EC process and the actual process. As such, the documented process can act as a baseline, i.e., digital twin, for organizational improvement. The proposed method enables us to record the dynamics and changes over the process lifecycle, and thus, the development of digital simulation models based on the digital twin to test for potential updates and improvements. It would be interesting to examine data and process mining techniques in relation to the development and maintenance of digital twins.

ACKNOWLEDGMENT

The authors would like to thank the high-tech, industrial machinery manufacturer in Western Europe for providing the study data.

REFERENCES

[1] R. G. Cooper and E. Kleinschmidt, "Perspective: The stage-gate® idea-to-launch process—update, What's New, and NexGen systems," *J. Product Innov. Manag.*, vol. 25 no. 3, pp. 213–232, 2008.

[2] C. Terwiesch and C. H. Loch, "Managing the process of engineering change orders: The case of the climate control system in automobile development," *J. Product Innov. Manage.*, vol. 16, no. 2, pp. 160–172, Mar. 1999.

[3] T. R. Browning and R. V. Ramasesh, "A survey of activity network-based process models for managing product development projects," *Prod. Oper. Manage.*, vol. 16, no. 2, pp. 217–240, Mar./Apr. 2007.

[4] T. A. W. Jarratt, C. M. Eckert, N. H. M. Caldwell, and P. J. Clarkson, "Engineering change: An overview and perspective on the literature," *Res. Eng. Des.*, vol. 22, no. 2, pp. 103–124, Apr. 2011.

[5] V. D. Aalst, W. A. J. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, pp. 1128–1142, Sep. 2004.

[6] *ProM*. [Online]. Available: http://www.processmining.org/logs/start

[7] P. Cankurtaran, F. Langerak, and A. Griffin, "Consequences of new product development speed: A meta-analysis," *J. Product Innov. Manage.*, vol. 30, no. 3, pp. 465–486, May 2013.

[8] J. Chen, F. Damanpour, and R. Reilly, "Understanding antecedents of new product development speed: A meta-analysis," *J. Oper. Manage.*, vol. 28, no. 1, pp. 17–33, 2010.

[9] C. H. Loch and C. Terwiesch, "Accelerating the process of engineering change orders: Capacity and congestion effects," *J. Product Innov. Manage.*, vol. 16, no. 2, pp. 145–159, Mar. 1999.

[10] B. J. Hommes, "The evaluation of business process modelling techniques," Doctoral dissertation, Technische Universiteit Delft, Delft, The Netherlands, 2004.

[11] Y. Liu, H. Zhang, C. P. Li, and R. J. Jiao, "Workflow simulation for operational decision support using event graph through process mining," *Decis. Support Syst.*, vol. 52, no. 3, pp. 685–697, Feb. 2012.

[12] B. Hamraz, N. H. M. Caldwell, and P. J. Clarkson, "A holistic categorization framework for literature on engineering change management," *Syst. Eng.*, vol. 16, no. 4, pp. 473–505, Dec. 2013.

[13] C. M. Eckert, P. J. Clarkson, and W. Zanker, "Change and customisation in complex engineering domains," *Res. Eng. Des.*, vol. 15, no. 1, pp. 1–21, Mar. 2004.

[14] C. M. Eckert, R. Keller, C. Earl, and P. J. Clarkson, "Supporting change processes in design: Complexity, prediction and reliability," *Rel. Eng. Syst. Saf.*, vol. 91, no. 12, pp. 1521–1534, Dec. 2006.

[15] R. Barzizza, M. Caridi, and R. Cigolini, "Engineering change: A theoretical assessment and a case study," *Prod. Planning Control*, vol. 12, no. 7, pp. 717–726, Oct./Nov. 2001.

[16] K. R. Reddi and Y. B. Moon, "Modelling engineering change management in a new product development supply chain," *Int. J. Prod. Res.*, vol. 51, no. 17, pp. 5271–5291, Sep. 2013.

[17] A. Siddiqi, G. Bounova, O. L. de Weck, R. Keller, and B. Robinson, "A posteriori design change analysis for complex engineering projects," *J. Mech. Des.*, vol. 133, no. 10, Oct. 2011.

[18] M. Thiede, D. Fuerstenau, and A. P. B. Barquet, "How is process mining technology used by organizations? A systematic literature review of empirical studies," *Bus. Process Manage. J.*, vol. 24, no. 4, pp. 900–922, 2018.

[19] E. Subrahmanian, C. Lee, H. Granger, and N.-D. Grp, "Managing and supporting product life cycle through engineering change management for a complex product," *Res. Eng. Des.*, vol. 26, no. 3, pp. 189–217, Jul. 2015.

[20] D. C. Wynn and C. M. Eckert, "Perspectives on iteration in design and development," *Res. Eng. Des.*, vol. 28, no. 2, pp. 153–184, Apr. 2017.

[21] D. C. Wynn and P. J. Clarkson, "Process models in design and development," *Res. Eng. Des.*, vol. 29, no. 2, pp. 161–202, 2018.

[22] J. Veldman and A. Alblas, "Managing design variety, process variety and engineering change: A case study of two capital good firms," *Res. Eng. Des.*, vol. 23, no. 4, pp. 269–290, Oct. 2012.

[23] W. van der Aalst, *Process Mining: Data Science in Action*. Berlin, Germany: Springer-Verlag, 2016.

[24] E. Rojas, J. Munoz-Gama, M. Sepulveda, and D. Capurro, "Process mining in healthcare: A literature review," *J. Biomed. Inform.*, vol. 61, pp. 224–36, Jun. 2016.

[25] T. G. Erdogan and A. Tarhan, "Systematic mapping of process mining studies in healthcare," *IEEE Access*, vol. 6, pp. 24543–24567, 2018.

[26] A. Bogarin, R. Cerezo, and C. Romero, "A survey on educational process mining," *Wiley Interdisciplinary Rev. - Data Mining Knowl. Discovery*, vol. 8, no. 1, pp. 1–17, Jan./Feb. 2018.

[27] A. Rozinat, "Process mining: Conformance and extension," Ph.D. dissertation, Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2010.

[28] M. de Leoni, W. van der Aalst, and M. Dees, "A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs," *Inf. Syst.*, vol. 56, pp. 235–257, 2016.

[29] M. T. Pich, C. H. Loch, and A. de Meyer, "On uncertainty, ambiguity, and complexity in project management," *Manage. Sci.*, vol. 48, no. 8, pp. 1008–1023, Aug. 2002.

[30] J. Kim and D. Wilemon, "Sources and assessment of complexity in NPD projects," *R & D Manage.*, vol. 33, no. 1, pp. 15–30, Jan. 2003.

[31] J. Cardoso, J. Mendling, G. Neumann, and H. Reijers, "A discourse on complexity of process models," in *Proc. Bus. Process Manage. Workshop*, 2006, pp. 115–126.

[32] E. Rolón, J. Cardoso, F. García, F. Ruiz, and M. Piattini, "Analysis and validation of control-flow complexity measures with BPMN process models," in *Proc. Int. Workshop Bus. Process Model., Develop. Support Int. Conf. Exploring Model. Methods Syst. Anal. Des.*, 2009, pp. 58–70.

[33] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proc. IEEE*, vol. 84, no. 10, pp. 1358–1384, Oct. 1996.

[34] R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.

[35] W. Cohen, *Fast Effective Rule Induction*. San Mateo, CA, USA: Morgan Kaufmann, 1995, pp. 115–123.

[36] *Fluxicon*. Accessed: Jul. 10, 2019. [Online]. Available: https://fluxicon.com/blog/2012/11/how-to-understand-the-variants-in-your-process/

[37] R. Lu and S. Sadiq, "On the discovery of preferred work practice through business process variants," in *Proc. Int. Conf. Conceptual Model.*, 2007, pp. 165–180.

[38] R. P. Lu, S. Sadiq, and G. Governatori, "On managing business processes variants," *Data Knowl. Eng.*, vol. 68, no. 7, pp. 642–664, Jul. 2009.

[39] *Knime*. Accessed: Jul. 10, 2019. [Online]. Available: www.knime.com

[40] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. 14th Int. Joint Conf. Artif. Intell.*, 1995, pp. 1137–1143.

[41] D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[42] R. Maull, D. Hughes, and J. Bennett, "The role of the bill-of-materials as a CAD/CAPM interface and the key importance of engineering change control," *Comput. Control Eng. J.*, vol. 3, no. 2, pp. 63–70, 1992.

[43] H. J. Lee, H. J. Ahn, J. W. Kim, and S. J. Park, "Capturing and reusing knowledge in engineering change management: A case of automobile development," *Inf. Syst. Frontiers*, vol. 8, no. 5, pp. 375–394, Dec. 2006.

[44] A. Riviere, C. DaCunha, and M. Tollenaere, "Performances in engineering changes management," in *Recent Advances in Integrated Design and Manufacturing in Mechanical Engineering*. Dordrecht, The Netherlands: Springer, 2003, pp. 369–378.

[45] A. J. M. M. Weijters and W. M. P. van der Aalst, "Rediscovering workflow models from event-based data using little thumb," *Integr. Comput.-Aided Eng.*, vol. 10, no. 2, pp. 151–162, 2003.

[46] M. Song and W. M. P. van der Aalst, "Towards comprehensive support for organizational mining," *Decis. Support Syst.*, vol. 46, no. 1, pp. 300–317, Dec. 2008.

[47] H. C. W. Lau, G. T. S. Ho, Y. Zhao, and N. S. H. Chung, "Development of a process mining system for supporting knowledge discovery in a supply chain network," *Int. J. Prod. Econ.*, vol. 122, no. 1, pp. 176–187, Nov. 2009.

[48] W. van der Aalst, "Process discovery from event data: Relating models and logs through abstraction," *Wiley Interdisciplinary Rev. - Data Mining Knowl. Discovery*, vol. 8, no. 3, 2018, Art. no. e1244.

[49] L. J. Lan, Y. Liu, and W. F. Lu, "Learning from the past: Uncovering design process models using an enriched process mining," *J. Mech. Des.*, vol. 140, no. 4, Apr. 2018, Art. no. 041403.