Business Process Measurement in small enterprises after the installation of an ERP software.

Stefano Siccardi and Claudia Sebastiani

CQ Creativiquadrati snc,
via Tadino 60, Milano, Italy
http://www.creativiquadrati.it

Abstract. We report the observation of the first six months of operation after the installation of an ERP software in a group of small Italian enterprises (some dealers of various products and one manufacturer). Before the ERP, no explicit process descriptions existed within the companies: the operations were manually performed, using office automation software or legacy programs that were not process oriented. The new ERP is equipped with a workflow engine, a number of standard processes that should be followed by the users, and a tracking system that logs the main steps of the processes. We use process mining tools to analyze the events logged by the ERP during the sales, the purchases and the manufacture cycles. Our aim is to 1) compare the ideal processes suggested by the ERP with the real paths followed by the users 2) describe the eventual adaptation of these paths, as the users became acquainted with the ERP 3) highlight critical segments in terms of time spent, iterations, etc. 4) compare the processes of different companies that are in similar business areas. The final goal is to get a better understanding of the processes and a rationalization of the operations. It must be stressed that both the ERP and the main tools used are open source, so that the process measurement is affordable even for very small (micro) enterprises.

Keywords: Business process mining, small enterprises, open source software, enterprise resource planning

1 Introduction

This report deals with the business process measurement in a sample of small enterprises in Italy. All the companies in the sample have recently adopted a new Enterprise Resource Planning (ERP) system, that is Odoo (previously OpenERP), available at `www.odoo.com`.

The introduction of the ERP was motivated by several reasons, mainly the necessity of replacing obsolete programs, the aim of integrating the software and the need to gain better control of operations. One of the main motivations of the choice of Odoo was its being an open source software.

The process control was not a primary goal at this stage: process design and measurement are usually activities too expensive, both in terms of time of the internal resources and of investment in consultancy, to be affordable for micro enterprises. So they were simply not considered by the entrepreneurs in the set of the realistic goals. On the other hand, processes in small enterprises are by no means simpler than in bigger ones, and a correct process understanding and design can be a major factor of success.

Another point that we were interested in was the companies' adaptation to the new ERP: although we customized the software in order to fit the users' business habits, it was unavoidable that they had somehow to change their ways of operating. So, in the first time of operation, we observed that they tried some different workflows before finding the one that best fit their needs. The changes in the processes during the first six months of operation, if any, can describe this adaptation phase.

As the Odoo system is intrinsically based on workflows, and automatically records many steps of the business process, it was evident that a very interesting byproduct of the ERP usage would have been the possibility to represent the processes as they are actually run. This can offer a number of interesting ideas to the companies' management.

Of course, the process representation would not have been possible without a process mining tool; we chose PROM rel. 6.4 (available at `www.processmining.org`), as it has all the capabilities we needed, and conforms to the open source philosophy of our projects. The basic ideas upon which PROM has been designed can be found e.g. in [1]; techniques to compute alignment between models and event logs are included in the software, and have been used in our analysis (see e.g. [2]).

For the preliminary data preparation we used Xesame ([3]).

The activity of process measurement aims to determine the performance of business processes, while the process monitoring is the activity to check them in order to discover some significant aspects (see e.g. [4]). In what follows, we concentrate on the first, and adopt a six months period of observation. There are also formal methods to study process adaptation and drift (see e.g. [5], [6]), however when considering the processes adaptation we limited ourselves to a qualitative approach.

1.1 The main process

We focused on the process of *selling - delivering goods or services - invoicing*, as it is the main one for all the companies of the sample. It is also a process that can be very complex, as customers can change their mind at several stages (before and after confirming an order, receiving goods, receiving invoices, etc.). Moreover it can be stopped and rescheduled several times if the goods to be delivered are not available, or if they arrive later at the companies' warehouses.

Three kinds of documents are involved in the process: a Sale Order (SO), containing the list of products and services that a customer wants to buy, with prices and payment terms; one or more delivery orders (OUT), generated by the

SO, with a list of products that must be taken from the warehouse and sent to the customer; one or more invoices (INV) that are fiscal documents specifying amounts to be paid with taxes and that can be created either from the SO or from the OUT(s).

When only services are sold no OUT is created, therefore the process is simpler.

We also consider the purchase process, consisting of the activity of ordering goods to suppliers and receiving them. Two documents are created: the Purchase Order (PO), with the list of products or services to buy; and one or more lists of items to receive at the warehouse (IN). Finally, for one company of the sample we also consider the manufacturing process, that is concerned with the Manufacture Order (MO), the list of products to produce. We limit ourselves to purchases and manufactures that are triggered by an SO, that is products are bought or manufactured just when a customer places an order for them.

The following eventrs have been tracked for the documents described above:

SOcreate: a sales order is created by a user and is in the draft state. This is the starting point of the whole process.

SOmail: a sale order is sent to the customer

SOconfirm: a sale order is validated and reaches the confirmed status

OUTcreate: a list of goods to deliver is created in draft status

OUTconfirm: a list of goods to deliver has been checked and confirmed by the user

OUTwaiting: a list of goods to deliver is put in a waiting status, as something is not available

OUTready: a list of goods to deliver is ready to be sent

OUTsent: a list of goods to deliver has been sent to the customer

OUTinvoice: a list of goods to deliver has been invoiced

OUTcancel: a list of goods to deliver has been cancelled by the user

OUTbackord: a list of goods to deliver has been partially delivered and a back order has been created for the remaining items

OUTscrap: some items in list of goods to deliver have been marked as scrapped

OUTnot2invoice: a list of goods to deliver has been marked not to be invoiced (e.g. because it replaces something under warranty)

INVcreate: an invoice has been created in draft status

INVproforma: an invoice has been transformed in a pro-forma invoice

INVvalidate: an invoice has been confirmed by the user

INVcancel: an invoice has been cancelled

INVpay: an invoice has been paid

For the purchase process the events are:

POcreate: a purchase order is created in draft status

POconfirm: a purchase order has been confirmed by the user

INcreate: a list of goods to receive is created in draft status

INreceive: a list of goods has been received

INcancel: a list of goods to receive has been cancelled

For the manufacturing process the events are:
MOcreate: a manufacturing order is created in draft status
MOwaiting: a manufacturing order is waiting raw material
MOready: a manufacturing order is ready to produce
MOstart: a manufacturing order production started
MOdone: a manufacturing order has been produced
All the above events are defined by the ERP systems, that automatically keeps track of their changes. The processes involving them have been mined and are described by the Petri nets in the next sessions.

The analysis method is the following:

1 - first we load in PROM the event log extracted by the Odoo database and preprocessed with Xesame

2 - we then take note of the main statistics of the complete log

3 - then we filter the cases in order to obtain cases that can be considered complete by a business point of view, e.g. cases that have been invoiced and paid, or, if not, that have been closed in some other reasonable way due to an agreement with the customer or else that can be analyzed as they have completed a definite section of the process

4 - next we build a model in Petri net format of the process as the company intends it

5 - at this stage we compare the filtered log to the model and draw considerations about its conformance and the general performance of the system

6 - we also analyze the filtered log to find users' behavioural patterns, e.g. if specific users are concerned with specific actions, or if some users tend to pass activities to others

7 - we again filter the log keeping only the cases started in the first month of operation, then those started in the last two months considered and repeat steps 5 and 6 on both sets

2 The cases

2.1 The sample companies

Our sample consists of six small Italian enterprises, that were observed during the first six months of their operations after the ERP implementation. We will use two digit codes to label them and we will group them according to their main business models:

1) Sale of goods that are kept in stock. Sample 01 (professional equipment and spare parts) and in part sample 06 (electromechanical devices) belong to this group.

2) Repair of customers' devices. Samples 02 and 03 belong to this group, both in the building industry, even if they sometimes sell devices and spare parts to their customers.

3) Sale of goods that are bought on the customer order and of maintenance services. Samples 04 and 05 belong to this group (electronic devices like PCs and telecommunication apparatus, software, etc.).

4) Manufacture to customer order. Sample 06 produces electromechanical components to fulfil customer orders, and sometimes to make stock, so in part belongs to this group and in part to group 1.

The event logs of sample enterprises are actually small, especially of samples 02 and 03; this is in part due to the fact that we chose the six months test period and the sample enterprises before knowing exactly how they would use the software and how many transactions they would process. Probably samples 02 and 03 are so small that it is even questionable if an ERP fits their needs.

However, as these kinds of enterprises are presently adopting ERP systems (as open source ERP systems are available), that forces them to follow structured processes, it is interesting to understand how they face this task.

2.2 General data

All the tracks of all the samples start with the SOcreate event, as expected. Some general data are summarized in table 1. The first lines refer to the total log as it is extracted from Odoo.

The second set of values refers to the log filtered keeping all the cases that end with INVpay, INVvalidate, OUTinvoice or OUTsent. This means that the process has been followed at least until some goods have been sent to the customer.

Table 1. General data of the samples

	01	02	03	04	05	06
Total Log						
Total tracks	2151	131	35	253	330	2105
Event classes	18	11	11	22	16	26
Events in the shortest track	1	1	2	1	1	1
Events in the longest track	61	20	12	64	50	66
Average events per track	11	9	8	9	9	13
Ending with INVpay	1064	73	11	45	125	845
Ending with INVvalidate	452	38	15	55	-	611
Ending with OUTsent	376	-	-	-	-	105
Log of completed cases						
Total tracks	1908	115	27	112	207	1579
Events in the shortest track	4	7	6	5	5	6
Events in the longest track	61	20	12	64	50	66
Average events per track	11	10	10	16	12	15
Min. event classes per track	4	10	6	4	5	6
Max event classes per track	14	11	11	17	15	19
Avg event classes per track	9	10	10	9	9	11

2.3 The 01 business sample

The process model is represented by the Petri net in figure 1. Sale orders always refer to products, even if some services are sold together, so at least one OUT is expected.

We note that:

- the first steps are related to the SO; when it is confirmed, the OUT is created
- when the OUT becomes ready, an invoice is created
- after invoice creation there may be a loop involving invoice cancelling and creating anew
- the process of sending goods may be repeated
- the last steps are validating the invoice and receiving payments
- no purchase orders are triggered by the sale orders, because the user sells goods that are kept in stock

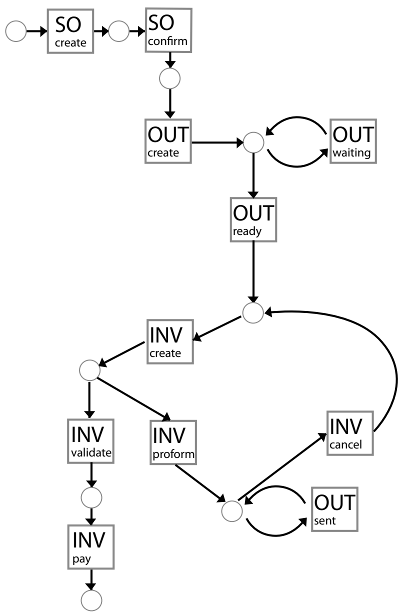

Fig. 1. The SO-OUT-INV process model for sample 01.

Applying the 1903 completed cases log onto this model, we found deviations mainly related to the events SOconfirm, OUTcreate, OUTwaiting, OUTready, INVcreate, INVvalidate, INVcancel and, to a lesser extent, INVproforma. The average throughput time is 1 month. From a time perspective, INVvalidate and INVcancel are critical; moreover, when a proforma invoice is issued, it is found a time lag between this event and the actual invoice. Applying only cases ending with INVpay (1064 items), we have similar results.

We then filtered the log for cases starting in the first month of the period, and found 244 tracks; and for events starting in the last two, and ending with one of the four events already considered by the end of the period, and found 178 tracks. These filtered logs, once applied onto the model, gave similar results. The only remarkable difference is the reduction in the average throughput time from 2 months to about 12 days. This is undoubtedly related to a better ability in the use of the new system and to the fact that in the first time of operation the users were still using their old software in parallel.

The PROM software can highlight actual cases that do not follow the model, this is useful to discuss with the users if the model has to be modified or if any unwanted actions are being performed. For instance, 255 cases follow the path:

SOcreate - SOconfirm - INVcreate - OUTinvoice - OUTsent - INVvalidate - INVpay

that is the invoice is created starting from the SO instead of the OUT. This is an alternative way of operating that can be included in the model.

On the other hand, 105 cases follow the path:

SOcreate - SOconfirm - OUTcreate - OUTwaiting - OUTwaiting - OUTready - INVcreate - OUTwaiting - OUTready - OUTsent - INVvalidate - INVpay

probably meaning that sale orders are modified later than expected, when an OUT has already been created. There are also much more complex examples, involving multiple sequences of OUTcreate - OUTconfirm - OUTcancel - OUTnot2invoice - OUTcreate etc.

This, in turn, can suggest the opportunity of clearer communication with the customers, so that they can place their orders correctly at the first time.

Finally, selective filtering of the log shows that all the users are involved in all the events, that is there is no user dedicated to a specific process section.

2.4 The 02 and 03 business samples

As 03 is a spin off of 02, these two samples are quite similar. Both the companies are very small and almost all the transactions were recorded by a single user, so a general uniformity of behaviour is found in the data.

The process model for both 02 and 03 is represented by the Petri net in figure 2. Sale orders refer always to products, even if some services are sold together, so at least one OUT is expected.

We note that in contrast with the 01 process, no loops are considered in this model.

Applying the log of completed cases of sample 02 (115 cases) and of sample 03 (27 cases) onto this model, we found deviations mainly related to the event INVcreate only. The average throughput time is 2.3 months for sample 02 and 1.35 for sample 03. From a time perspective, the main critical event for both samples is INVpay; this probably reflects a commercial issue and is not related to the process complexity. For sample 03 also OUTready is critical, whose average throughput time is 20 days; this probably reflects the fact that the company business mainly consists of fixing customers' devices, so when something arrives

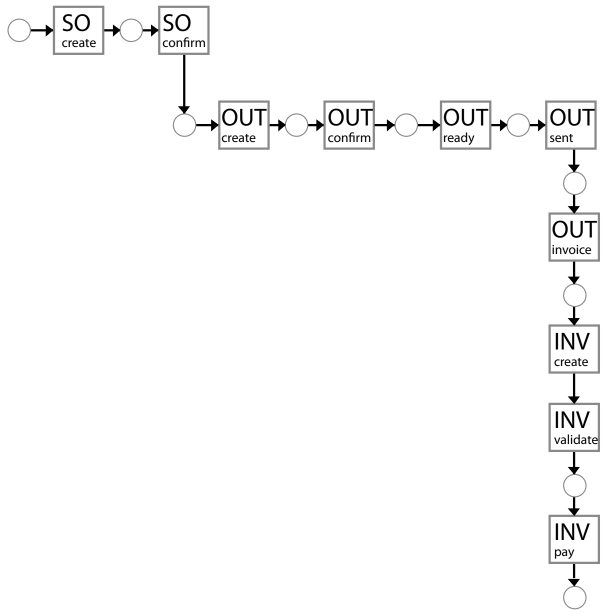

Fig. 2. The SO-OUT-INV process model for samples 02 and 03.

at their warehouse, time is needed to repair it before it can be sent back to the customer.

The filtered logs of sample 02 for cases starting in the first month and in the last two months, once applied onto the model, gave similar results.

As the cases are very few for sample 03 we did not compare the first months to the last ones.

Although the model is quite linear, the analysis of tracks highlighted some cases comprising loops and some degrees of complexity. Their number, however, was so low that they can be considered exceptional cases more than an indication of a more complex process. An example with multiple OUT and INV processing is shown in figure 3. Here an OUT can be cancelled after being confirmed and declared ready, and even when the invoicing process has started, by a manual action that has been represented as an "invisible" task (see e.g. [7]), so that it must be created again. Multiple invoices are created, one just after the SO confirmation, and others after sending goods.

2.5 The 04 and 05 business samples

In both samples, sales are of two different kinds: sometimes the user sells devices and equipment, that originate physical movement of goods (OUTs) and purchase orders; other times they sell services, e.g. assistance contracts, that do not imply any OUT. Moreover they manage assistance agreements with their customers, that have a single sale order and several invoices to be paid e.g. quarterly, so it

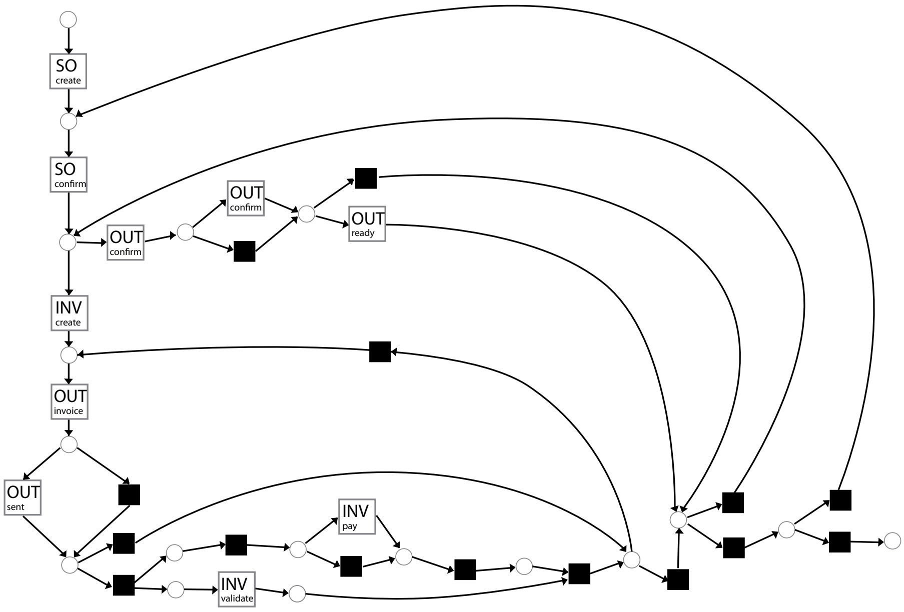

Fig. 3. A complex case of sample 02.

can happen that payment is not found as the last event in many recorded cases. The users usually buy products when a SO is received, so the PO process is triggered.

For both samples two business process models are found: for sample 04, 124 cases involve some physical products and 129 services only, for sample 05, 177 cases involve some physical products and 153 services only.

For 04 the filtered log of complete cases contains 58 cases involving products and 54 service cases.

For 05 the filtered log of complete cases contains 92 cases involving products and 115 service cases. Moreover 91 cases end with the SOmail event, that is they are quotations that the customers did not accept.

The first process is represented by the Petri net in figure 4 and the second in figure 5.

We note that for the first process:

- the first steps are related to the SO; when it is confirmed a PO is created

- when the PO is confirmed, an IN is created waiting for the purchased products; when products are received the OUT becomes ready

- after sending the OUT the invoice is created, validated and paid

- there can be change requests before invoice validation, that restart the process modifying the OUT

For the second process:

- the first steps are related to the SO; when it is confirmed an INV is created

- after invoice creation there is a loop involving invoice cancelling and creating anew, or receiving a payment and creating a new invoice

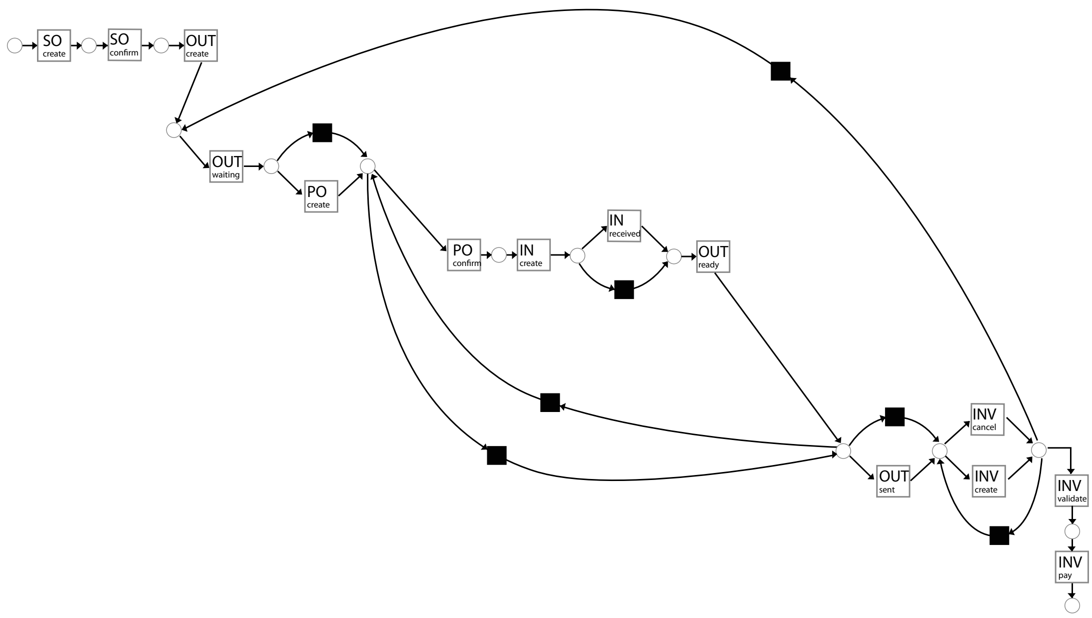

Fig. 4. The SO-PO-OUT-INV process model for products of samples 04 and 05.

Applying the cases log onto these models, we did not find significant deviations. For the product process of sample 04 the average throughput time is 1.4 months and for sample 05 it is 1.8 months. From a time perspective, no specific events are critical.

For the service process of sample 04 the average throughput time is 2.7 months. From a time perspective, INVcreate is critical; however, as these kind of sales involve long periods this could be a characteristic of the business (an invoice being issued quarterly in most cases).

For the service process of sample 05 the average throughput time is 1.7 months. From a time perspective, no specific events are critical. The logs are in good agreement with the model.

As the cases of sample 04 are very few, we did not compare the first month to the last ones.

For sample 05 comparing cases of first month to the last two, we found that in the first month the average throughput time was higher (2.45 months instead of 11 days for products and 2.5 months instead of 10 days for services).

Finally, selective filtering of the log shows that in sample 04 all the users are involved in all the events, moreover a single user was involved in about two thirds of the cases.

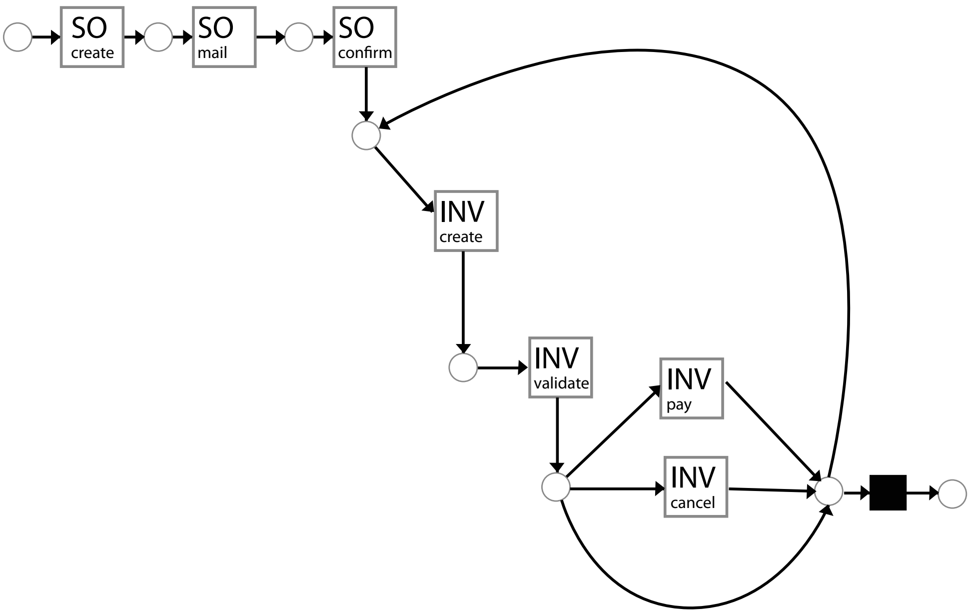

Fig. 5. The SO-INV process model for services of samples 04 and 05.

In sample 05 all the users are involved in the SO and OUT processing, but only 3 in invoicing and 2 in purchasing.

2.6 The 06 business sample

The user manufactures products both for keeping stock and to fulfil specific sale orders, so the MO process is triggered by an SO for some products. Most sales involve products, although 179 service SO have been found.

The filtered log of complete cases contains 441 cases involving products specifically manufactured for the SO and 1138 cases involving products taken from the warehouse.

The first process is represented by the Petri net in figure 6, while the second is similar to the one already analyzed in figure 1.

We note that for the first process:

- the first steps are related to the SO; when it is confirmed a MO is created
- when the MO is done, the OUT becomes ready
- after sending the OUT the invoice follows its usual process

Applying the cases log onto these models, we did not find significant deviations. For the first process the average throughput time is 3.1 months. This long time is probably due to customers placing their orders even months in advance. From a time perspective, the only event that needs further analysis is the MOready as it is sometimes delayed for lack of raw materials, or sometimes due to an agreement with the customer about the delivery date. The main deviations from

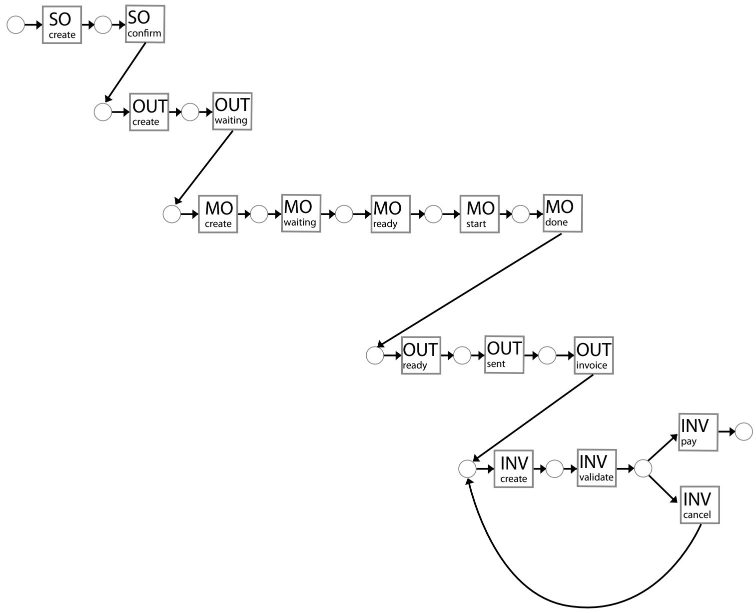

Fig. 6. The SO-MO-OUT-INV process model for sample 06.

the model are cases where products are delivered in two or more shipments, and backorders are created.

For the second process the average throughput time is 2.5 months. From a time perspective, no specific events are critical. Some deviations from the model are found for the events SOconfirm and OUTconfirm, probably due to late changes to the SO requested by the customers.

Comparing the first month of operation with the last two, we found that the main difference is a decrease in throughput time and in case length, that probably means that the users gradually become acquainted with the system and do not need to cancel and redo operations if not in exceptional cases.

Finally, selective filtering of the log shows that although 9 users recorded some transactions, sales orders and invoices were usually managed by 3 users, manufacturing by 2 others and delivering by all of them.

3 Discussion

The results of our analysis show that, in general, the processes follow the models that the users have in mind for their "normal" business cases. This is somewhat in contrast with the subjective impression that the users themselves have of their own business.

In the preliminary meetings that we had, to check if the ERP would actually fit the companys' needs, the users described many complex situations that could arise, as if they were very common. Looking at the data, it turned out that these are actually exceptions; it is true that they can be very involved, and can generate time loss and dissatisfaction, but they are rarer than expected.

Although the data are too few to get exact statistics, the indication is that criticality is related to exceptional cases and not to the average ones.

We discussed specific cases, like that of fig. 3, with the users, and found that the main source of complexity, for all samples, consists of the customers' requests to change their agreements at a late stage. That is, some customers want to change their SO after confirming them, or even after they have received the products; or they want to change the shipment or payment conditions, etc. This means that documents must be cancelled and processed again, often with consequences on other documents.

It is interesting that we found two different reactions to the above remarks. Some managers told us that their goal is to be flexible, and that to fulfill their customers requests is exactly their mission. They think that complexity is inherent to their job, and used the cases logs to ask new implementations to our ERP to fit the way they run their business.

Other managers, on the contrary, realized that simpler processes are more efficient, and decided to follow the ERP standard processes for as long as possible. They defined some new internal rules for this, e.g. a standard time to wait before confirming the main documents (e.g. SO). If a customer wants to change something after, he will have to pay an extra fee for the reprocessing of the order.

4 Conclusions

We applied process mining techniques to a sample of micro enterprises, that would not be able to consider their own processes in other ways. The main goal of this activity is, in our opinion, to bring process design to the managers' attention.

About our business sample, we observe that:

- processes of similar business are very similar, for instance those of sample 04 and 05, and in part also of sample 01 and 06
- in general the actual processes follow the models, and specific deviations can be discussed with the managers to suggest new ideas; anyway they are more often exceptional cases than variants of the models
- we did not observe process modification or adaptation in time; the main difference between the first and last months is an increase in performance. As it is not realistic that the business itself has quickened so much in six months, this is probably due to an increase in the users' familiarity with the system
- especially in the smaller samples, there are no fixed correlations between tasks and people (everybody does everything), while bigger companies tend to assign specific roles to the users

Finally, we note that in this analysis we used the standard events traced by Odoo, but it is possible to trace any events that the users think could be relevant for their business.

References

1. Weijters, A.J.M.M., van der Aalst, W.M.P.: Workflow Mining: Discovering Workflow Models from Event-Based Data, in C. Dousson, F. Happner, and R. Quiniou, editors, Proceedings of the ECAI Workshop on Knowledge Discovery and Spatial Data, 78–84, (2002)
2. Adriansyah, A.: Aligning observed and modeled behavior, PhD Thesis. Technische Universiteit Eindhoven, Eindhoven, The Netherlands, (2014)
3. Buijs, J.C.A.M.: Mapping Data Sources to XES in a Generic Way, Master Thesis, Technische Uviversiteit Eindhoven, Eindhoven, The Netherlands, (2010)
4. Kueng, P., Hagen, C., Rodel, M., Seifert, S.: Business Process Monitoring and Measurement in a Large Bank: Challenges and selected Approaches, In: Proceedings of the 16th International Workshop on Database and Expert Systems Applications (DEXA2005), IEEE Press, New York, 955–961, (2005)
5. Hallerbach, A.,Bauer, T., Reichert, M.: Capturing Variability in Business Process Models: The Provop Approach, Journal of Software Maintenance and Evolution: Research and Practice, 519–546, (2010)
6. Bose, R.P.J.C., van der Aalst, W.M.P., Zliobaite, I., Pechenizkiy, M.:Dealing With Concept Drifts in Process Mining, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, (2014)
7. Wen, L., Wang, J., Van Der Aalst, W.M.P., Huang, B., Sun, J., Mining process models with prime invisible tasks, Data and Knowledge Engineering, vol. 69, 999–1021, (2010).