Business process analysis by model checking

David Kuhlen[1] and Andreas Speck[2]

[1] Datenlotsen Informationssysteme GmbH, Hamburg, Germany,
david.kuhlen@datenlotsen.de, Homepage: www.datenlotsen.de
[2] Christian Albrechts University Kiel, Germany,
aspe@informatik.uni-kiel.de

Abstract. In order to check the performance of a process, key performance indicators will be necessary. Key performance indicators (KPI's) help to compare the profitability of two processes. To obtain detailed information on the performance, a procedure is needed to analyse a process by checking its model. Such a procedure helps to evaluate a process without the need of doing an expensive attempt by testing and validating the process in the reality and measure the flow parameters.
By doing process analysis, it would be possible to evaluate changes on the process model of software development. Often new or changed process models are suggested. By doing a stochastic analysis, it will be possible to evaluate the benefit a process change can offer.
This paper introduces a new algorithm which is part of *PAS*. The algorithm would be used to analyse the process of software development. The code of the algorithm, its contribution and the results are presented.

Keywords: Business process analysis, Model Checking, Key Performance Indicators

1 Introduction

In order to decide whether a redesign should be rolled out in the whole organisation or not, the management needs reliable information. Models which describe the process are the result of the design phase and could be used for their analysis [24]. The analysis could assess key performance indicators which describe the profitability of a process. Business processes often have to follow specific rules [22]. The rules are formulated in the design phase. The calculation of the key performance indicators needs to consider these rules. A procedure has to be developed which allows the analysis of a process by checking its model and support the management of software companies.

This paper describes a procedure which aims to analyse processes on the basis of its definition. It should examine the possibilities to determine key performance indicators in a process by parsing its model. It adapts a concept of embedded *Markov chains* for the analysis of a process model. A related concept was presented by Jacquier et. al. The underlying concept of using *Markov chains* for a process analysis, by simulating all paths, will be adapted in order to examine software development approaches.

The reasons for writing this paper are the challenges in the performance of the software development processes in general (e.g. [6]). Considering the case of the challenges in development processes, an analysis approach has to be investigated which supports decision makers who design a new process model. This is important, because productivity isn't at the center of attention in the field of software development [18].

The phase of business process modelling is substantial. Simple process models which arise during the design, often do not enable a basic analysis (i. e. [24]). The challenge in the field of the development processes is that the underlying process which defines how development works, often changes. Approaches like *scrum*, *waterfall model*, *kanban*, *extreme programming (xp)*, *rational unified process (rup)* or the *v-model* have influenced the process of software development like many others. Figure 1 shows one alternative of the huge variety of processes.

This is a main challenge to be solved. During the design of a process its costs will be determined. In the phase of business process modelling the only basis for the analysis is a model [24]. Often, there is little support in this phase and possibilities to analyse the profitability of a process model are not in use [25]. Without the possibility to check if the performance meets the expectations, management hasn't got the opportunity to achieve an ideal design.

To support the management of software developing companies, we are looking for an algorithm to perform this analysis of iterative process definitions. The algorithm should be the essential part of a prototype, called *PAS (Process Analysis Studio)* which has to be developed in order to assess the costs of a process. For this purpose, a redesign of the software production processes serves as an example for the validation (case).

Often, software developing companies have a hard time controlling the design of the development approach. The process is iterative, it has many exceptions and it changes often. Our objective is to solve these problems which may occur especially in the development process.

This recommendation leads to a change in the software development process. To validate the benefit offered by such a redesign, an analysis will be conducted. The result of an analysis of the classic development process could be compared with the result of the analysis of the new version of the development process.

The results, analysed by *PAS*, will be used to validate the potentials of such an analysis approach, in order to generate added value for the process management of software producers. To answer the research question, special attention is put on parameters which empower management to control the profitability of future process designs.

In order to decide about the redesign of operational processes, the analysis of a first draft of the model helps management to make the right decision. So, the prototype *PAS* should empower the management to choose the best alternative that would maximize the return [10]. In the context of validation of business process models Speck et al. explain the need to identify errors and problems in early states of the system modelling [22]. The analysis of a process will be

necessary, because a mistake in the design of a process model may lead to high throughput times, low service levels and a need for excess capacity [24].

The first part of this paper describes the state of research. Common procedures to analyse the performance of a process are described in this section. In the next part, the problems of a classic process analysis are described. In order to solve these problems, a new technique is presented in the forth section. The last section validates the technique.

2 Related Work

Different publications already observed partial aspects of the field of process cost analysis. First, process analysis aims to guide decision making. Pounds described the benefit of simulation, experimentation and process analysis to improve decision making of management [19]. Pounds also described the possibility of obtaining valid results by the application of easy decision models. Besides Pounds, various other authors investigated the benefits of simulation (e. g. [11], [27], [23], [15], [10], [3] and [5]).

The aspect of costs in the field of manufacturing processes has also been observed in different publications (e. g. [28], [9], [4], [5], [15] and [8]). The problem of cost estimation and the consideration of costs being a context-dependent metric of the performance of manufactured systems was described by Roth [9]. Roth addresses the relationship between engineering and economics.

Ways to improve the efficiency (in terms of costs) by a special process design have also beeen described differently. Charles et al. described the possibility to redesign a process which reduces its costs by eliminating unnecessary work [16]. In order to increase the profitability, the usage of ressources of a process was analysed by him.

The analysis of processes fits into the research topic of work flow management systems. Regarding to the architecture of ARIS, Scheer and Nüttgens described the possibility of cost analysis which controls and plans the business process [20]. Scheer and Nüttgens also mention the approach of activity-based costing and the determination of the best process alternative.

The approach to integrate a stochastic model in the field of process analysis was an object of investigation of Jacquier et al [12]. This concept of using a *markov chain* in the analysis was also used in this publication. Jacquier et al. presented the concept of an algorithm which computes a simulation-based estimation which leads to results with a very high degree of accuracy.

The analysis of processes in software production could be sensible. Therefore, the derivation of an algorithm, based on this preliminary work, might be helpful. This could help companies to choose the right development approach and adapt it to their business.

3 Model driven process analysis to increase the operating efficiency

To design a process, an organisation often starts with collecting information about the working steps. A business process could be defined as an ordered set of activities which become executed to attain a special purpose [6], [1]. Ultimately, the order of steps will be expressed in a process model. In the context of process design, better decision making is the key to enlarge added value [2]. To improve decision making, information needs to be obtained by calculating key performance indicators.

3.1 Business process analysis

A business process analysis aims to investigate properties of business processes [25] which express the expectable efficiency. Analysis for the purpose of obtaining financial benefits in a business process, is defined as a part in a process improvement procedure [5]. Different ways of doing an analysis will be distinguished: analysis at the design-time *(pre-analysis)* and analysis at the run-time *(post-analysis)* [24]. A *pre-analysis* will be used in the design phase and could be realized by doing a simulation [25]. Graph analysis, model checking, reduction techniques, *Petri nets* and *markov-chain* analysis can also be used for performance evaluation [24] at the design-time. In contrast, the *post-analysis* is possible after the process is used in production. Therefore, it might be less helpful in a situation when management has to decide about a process change (cf. process mining, subsection 3.2).

An automated analysis demands a formal procedure, as explained in various publications. If standards for process modelling are missing [25], an automated analysis might be aggravated.

For the validation, a model is needed which describes the business process [22]. Speck et. al explain in the context of a checking tool for validation that the process has to be described as finite automata. A process model which describes the work-flow as a finite automata enables the possibility for valid process analysis. In addition, state changes (as a result of business rules) have to be expressed in a formal notation [17]. The formulation of business rules for different paths in business processes might be possible by using CTL [21].

3.2 Distinction compared to post-analysis using the example of process mining

Process mining could deliver information about the real progress of a process, documented in logs [24]. These real progresses of multiple instances could be compared with the designed process model. This comparison reveals that reality is often quite different from the idealized models [24].

Often, a resemblance between simulation *(pre-analysis)* and work-flow models *(post-analysis)* is seen [24]. However, there is a difference between pre- and post-

analysis in regard to the objectives, the techniques and the benefits. A *post-analysis* determines key performance indicators on the basis of a real execution of the process.

A process mining technique uses the information of log files to describe the progress of each instance of the processes and compares them. This technique bases on the assumption that it is possible to collect enough work-flow logs with event data [26]. Whether these information are available or not, depends on the used information systems and the designed process. A process which involves multiple actors who cooperate without using information systems does not allow process mining. Even if information systems are used, process mining bases on *assumptions* about the completion of the log and the usefulness of the information (e. g. the assumption about a large subset of possible behaviours to be observed) [26]. If the model exhibits alternative and parallel routing, then the work-flow log will typically not contain all possible combinations, because it is not realistic that each interleaving is present in the log [26]. The application of process mining could show all alternative orders of events which happen in reality. The result shows a chaotic map of different paths. The problem is that the result shows all details, without providing a suitable abstraction [24].

During the plan of the process model, process mining techniques could not provide much benefit. If the used model is unknown, mining techniques are not very useful [26]. Naturally, log data (as a result of real past executions) is missing during the design of completely new processes.

Not all processes are qualified for the utilization of process mining. The application of mining techniques in Health-Care, for the analysis of the flow of multi-disciplinary patients [26], may be a challenge, because of many exceptions in the process. A process model which consists of a set of 'ad-hoc' activities does not fulfil the premise to describe a finite automata.

However, the model just has to be an ideal pattern for the process. The reality will be different in any case. Therefore, it is better to check the model if it really fits.

3.3 Operating efficiency in software development

Software producers have to improve the profitability of their processes [14]. To investigate ways for improving processes, techniques for the analysis of business processes can be used [25].

Different approaches are possible to conduct the redesign of operational processes. Redesigning is possible by *straight through processing* which often results in a reduction of flow time and cut of costs or by *case handling* which addresses the problem of many processes which are much too variable or to complex for being captured in a process diagram [25]. Processes for the production of software need both: they are often complex, because of creative work and they need a reduction of flow times and a cut of costs.

Process paths have different probabilities. These probabilities influence the frequency of iterations through the paths. An investment leads to a modification of the process. A modification of the process alters the probabilities of the paths.

This could be the basis to assess the modification by executing a stochastic analysis. The stochastic analysis could calculate a neighbourhood with a statistically significant total probability. The neighbourhood consists of different paths with its costs. The probable process costs could be determined by calculating the average costs of the neighbourhood. The profitability of a modification of the process could be evaluated by comparing the average costs of two process variants. The analysis of paths, weighted with probabilities, can be simplified by restricting the attention to an embedded *markov chain* [13]. Figure 1 illustrates the example of a classic process, used in the production of software development. The process consists of seven activities which belong to phases of requirements engineering, development and delivery of software. For each step, the activity quantity induced costs for an ABC-procedure are estimated. Assumptions are made too, on the probabilities of the flows.

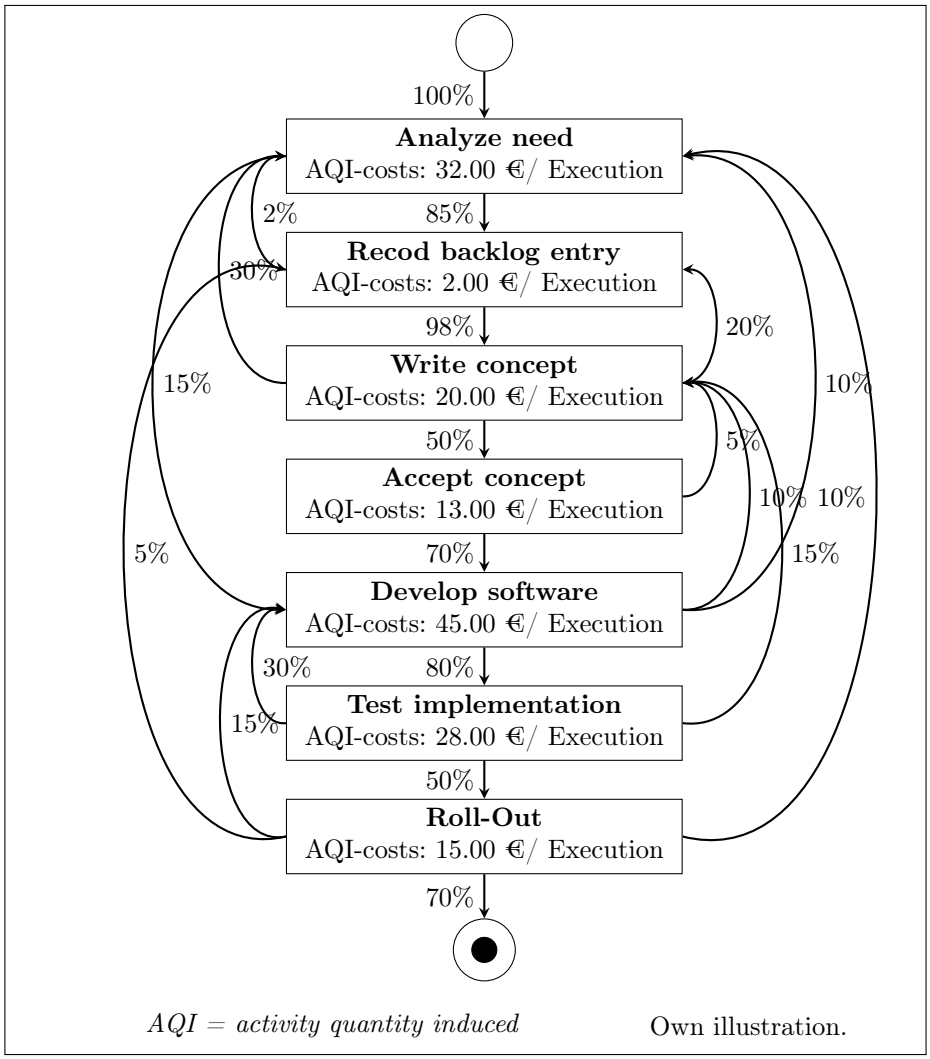

Fig. 1. Example of a process as a basis for the analysis

Figure 1 shows the development process. It includes KPI's. An analysis has to consider the process definition given by this model.

4 Automated process analysis

In software development, techniques which use the comparison of the models and the source code to assure the consistency of the source, are discussed [22]. Similar techniques are needed for an automated process analysis. The model

(source code of the process) should be compared with the emergence of process costs to assure the consistency of the business process.

In the phase of process design, typically, simulations are possible [24]. However if the simulation is restricted to displaying funny colourful bowls, running quickly through the process model, it might be less helpful. To support the construction of a capable process, the designers and the management need to control the right key performance indicators. As well as KPI's the right level of abstraction has to be chosen [24].

In general, key performance indicators empower a company to decide about process changes. The algorithm of *PAS* needs KPI's as an input (called 'parameters') to score other key performance indicators. All KPI's could be used to restructure the process during its design.

The level of abstraction has to match the need of economic KPI's: too many KPI's complicate the control of a process. Different key performance indicators may interfere. As a result, the design has to choose the optimum between competitive objectives and the consequence being a suboptimal compromise. Without the right KPI's and parameters, the management of the process could be impossible.

4.1 Parameters in the process model

Parameters help management to assume the impact of modifications on the profitability of a process during its design. The parameters need to describe the process sharply enough to enable an effective analysis. They need to provide the comparison of alternative processes.

A procedure, in order to perform the analysis of a process, needs a formal process model [25]. The empowerment of management to form a profitable process bases on a parametrized model and the possibility to estimate data for parameters [24]. By using minimal information, it could be possible to calculate all kinds of performance measures [26].

The needed parameters could be found by the attempt of formulating a process analysis mathematically. Let A be a set of n activities. Each activity has costs c_i, whereby $i = 1..n$. Let F be a set of flows, defined by $F \subseteq A \times A$ with the values $p(a, b)$, whereby $a, b \in A$. The definition of the process also determines a set of starting activities S and a set of ending activities E, whereby $S, E \subseteq A$. We are looking for the process costs C.

We can define a set P of paths as $P := \{x | x \in R(n) \wedge n \rightarrow \infty\}$. If the process definition contains a cycle, the cardinality of P aspires to infinity, so that $|P| \rightarrow \infty$. Depending on the maximum length of a path, the set of paths could be defined as follows $R(n) := \{(a_1, ..., a_n) | a_1 \in S \wedge a_n \in E \wedge a_i \in A fori = 1..n \wedge \forall (a_i, a_{i+1}) \in F\}$. The probability of each path is a transformation of $w : R(n) \rightarrow \mathbb{R}$, $w(x) := \{((a_1, ..., a_n), y) | (a_1, ..., a_n) \in R(n) \wedge y = \prod_{i=1}^{n} p(a_i, a_{i+1})\}$. The costs of each path are defined as a transformation of $c : R(n) \rightarrow \mathbb{R}, c(x) := \{((a_1, ..., a_n), y) | (a_1, ..., a_n) \in R(n) \wedge y = \sum_{i=1}^{n} c_i\}$. The costs of a path C_{path} are defined as following: $R(n) \rightarrow \mathbb{R}, C_{path}(x) := w(x) \cdot c(x)$. The overall process costs are the result of the application of an activity based costing approach. The

expectable total process costs C are defined as follows: $C : P \rightarrow \mathbb{R}, C(x) := \{(P, y)|y = \sum_{i=1}^{|P|} C_{path}(x_i) \wedge x_i \in P\}$.

In order to calculate the process costs a small number of different inputs is necessary. First of all, the sets of activities and flows have to be described by entering a classic process model. The classic model describes F, the relation of predecessor and successor. The analysis of the process costs will be possible by extending a classic process model with two attributes: the activities have to get an attribute "costs" and the flows have to get an attribute "probability". The costs are defined as c_i of each path. The probability of a flow is given by $p(a, b)$.

4.2 Analysis algorithm

An algorithm which aims to analyse a process has to make assumptions about the data structure. A process is equivalent to a digraph. Considering the parameters in subsection 4.1, the process complies with a weighted digraph. Depending on the process model, the graph could by cyclic. Considering this data structure, an algorithm to analyse a (cyclic) weighted digraph has to fulfil different requirements:

1. Only valid paths which belong to the weighted digraph should be analysed.
2. A majority of computable paths should be analysed.
3. Paths shouldn't be analysed twice.
4. The analysis should determine a result, considering all analysed paths.

Regarding to the explanations in subsection 4.1, the algorithm has to determine C. Greatest challenge in determining C is that it has to determine $R(n)$. To compute the set $R(n)$, the algorithm has to ensure that only valid paths have to be analysed $((a_1, ..., a_n), a_i \in A, i = 1...n \wedge (a_i, a_{i+1}) \in F)$. Therefore, the algorithm has to focus on the set of flows F. A process without a cycle could be analysed completely. In contrast, for a cyclic process the exact calculation of all paths is impossible, because the cardinality of the set of the paths is infinite ($|P| \rightarrow \infty$). An approximation is possible by computing the majority of paths $R'(n) \subset R(n)$ by $R'n(x) := \{(a_1, ..., a_n)|(a_1, ..., a_n) \in R(n) \wedge \sum_{i=1}^{|R'(n)|} w_i \geq \alpha\}$. The approximation bases on the definition of a maximum probability α. The third criterion could be fulfilled by the data structure, because a set never contains an element twice. However, during the analysis it would improve the performance if the algorithm didn't repeat the analysis of the same path. The analysis of alternative processes should compute a result on an ordinal scale which is comparable in terms of greater / smaller relations. By computing the expectable costs, two alternative processes could be compared.

To perform the analysis of valid paths, the algorithm has to walk through the graph. In preparation for the analysis, the process graph could be stored as adjacency list. The analysis of an unbranded chain of activities is easy straight forward. The algorithm needs a solution if the graph branches and more alternative activities are possible followers of the current activity. A branch results in multiple different paths in $R(n)$. The analysis of branched (or forked) transitions

is possible by using multiplying lists. The algorithm traverses the process. Along the way, all vertexes are stored in a list. If the algorithm reaches a treeing, it will multiply the list for each vertex following. Therefore, it creates a copy of the original list for each follower. The follower will be recorded in its copy of the original list. The use of multiplying lists is illustrated in line 19 of Code 1.1.

```
1  while(openPathExists) {
2     openPathExists = false;
3     // for all paths of the workingList which aren't finished
4     for (int i = 0; i < workingList. size(); i++) {
5        SimulatedPath workingEntry = workingList. get(i);
6        if(workingList.get(i).getProbability()<minPathProbability()) {
7           workingList. remove(i);
8        }
9        else {
10          Vertex lastPoint = workingEntry. getPath(). getLastFlow().
                getEnd();
11          if(this. getAdjacencyList().getFlowsOfEdge(lastPoint).size() >
                1) {
12             for (int j = 0; j <
                   this.getAdjacencyList().getFlowsOfEdge(lastPoint).size();
                   j++) {
13                Flow flow = this.getAdjacencyList()
14                            .getFlowsOfEdge(lastPoint).get(j);
15                // IsCycleFlow --> NumberOfVisists <= iterationlevel
16                if(!isCycleFlow(flow) || numberOfVisitsOfThisFlow(flow) <
                      iterationLevel) {
17                   // if there are more possibile followers, mulitplicit
                         the list
18                   SimulatedPath newSP = new
                         SimulatedPath(workingEntry,flow);
19                   workingList.add(newSP);
20                }
21             }
22             workingListSimulatedPaths.remove(i);
23          }
24          else if(getAdjacencyList().getFlowsOfEdge(lastPoint). size() ==
                1) {
25             Flow flow = this.getAdjacencyList().
                   getFlowsOfEdge(lastPoint). get(0);
26             if(!isCycleFlow(flow) || numberOfVisitsOfThisFlow(flow) <
                   iterationLevel) {
27                workingEntry.addFlow(flow);
28             }
29             else {
30                workingList.remove(i);
31 }}}}}
```

Code 1.1. Extract of the algorithm of *PAS* to traverse the process

Code 1.1 presents a central algorithm which is being used to simulate the process. It allows to run through the process in iterative cycles.

The advantage of Code 1.1 is its smart procedure to analyse all paths in iterative process models. Its disadvantage is the average duration.

This part of the algorithm of *PAS* has to handle cycles. First, the algorithm has to check if it should continue the analysis. In this example, the maximum limit α is realized by excluding all paths with a probability, less then the minPathProbability() (cf. line 6 - 7). Illustrated in line 16 of Code 1.1, the use of a lower bound is completed by the use of a *(iteration level)*. The restriction of visits of cycle edges *(Iteration Level)* limits the overall duration of the analysis. To cancel the analysis of paths which are unreasonable, it is useful to limit the minimal probability of each path. With this limitations, the problem of an *infinite geometric series* would be pre-empted. Irrelevant paths have to be excluded from the analysis quickly, so that they do not delay the analysis. To handle a cycle, the algorithm has to be able to recognize a cycle. The detection of a cycle is implemented in the method isCycleFlow(flow) which is used in line 16. The detection of cycle flows will be possible by a recursive algorithm which tries to end the process without repeating a transition twice. This way of proceeding has to be described in a definition of a *cycle edge*.

Definition 1 (cycle edge). *A cycle edge is an edge which leads to a vertex x_p, from where the end is reachable just by visiting the vertex x_s once more, from where it was possible to end the process without visiting x_p.*

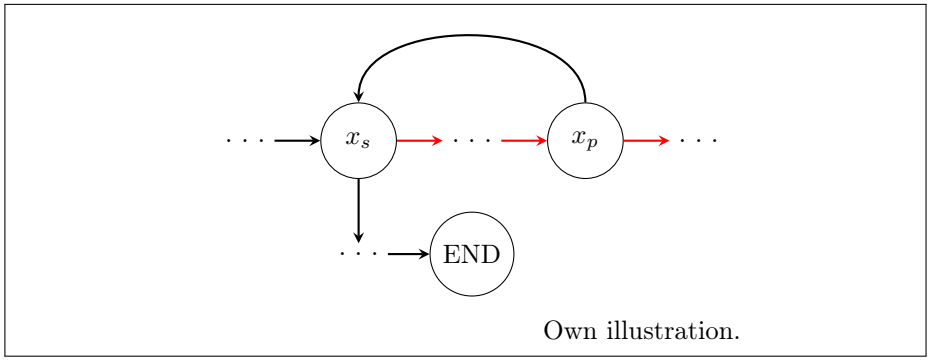

Fig. 2. Illustration of the definition of a cycle edge

Figure 2 shows the concept of a procedure, used by the algorithm in *PAS* to determine whether an edge is a cycle edge or not.

Following this definition, a *cycle edge* is a connection between two vertexes which lead the process unavoidable to a cycle. The formal description of a cycle empowers the algorithm to check whether a flow leads into a cycle. If the analysis is confronted with a cycle during the execution, it checks how often it has already

passed the *cycle edge*. The number of visits has to be less than the maximum iteration level (cf. line 16 and 27 in Code 1.1. At level 0,there won't be passed any more cycles. At level 1, each cycle wont be passed more than once, and so on.

The analysis determines the set $P(n)$. During the analysis, different elements (paths) of $P(n)$ are found. Their probability and their costs will be calculated for every path. If a cycle leads to a path which contains an activity twice, the path will get longer. Therefore, the costs of both activities increase the total costs of the path. The longer the path gets due to a cycle, the more its cost will increase. In any case, the longer the path, the more its probability will decrease. An evidence for this relation is illustrated in Figure 3. Beyond the context of a cyclic graph, this correlation does not have to be true, because a short process (with a higher probability) could also have higher costs than a long process with a minor probability.

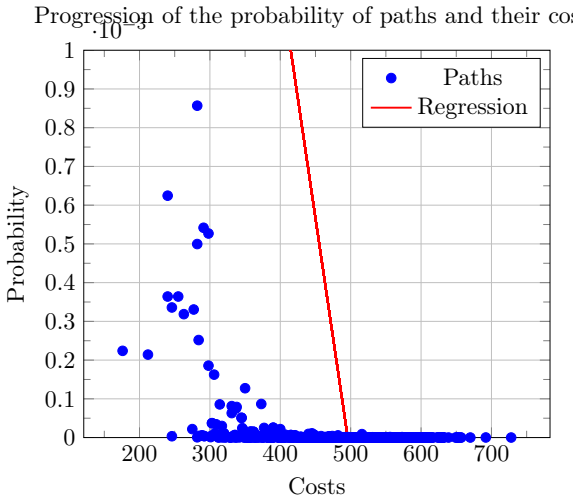

Fig. 3. Relation between probability and costs in process paths

Figure 3 shows the progression of the probability of paths and their costs. It supports the pronouncement of longer chains of activities which raise higher costs. The displayed results belong to a modified version of the process different from Figure 1. It just contains a selected set of results of the simulation.

The analysis returns a tuple of the entire probability (sum of all single probabilities of each path) and the expected process costs. Like the *expectancy value*, the expected costs could be calculated by putting weight on the costs of each path. If (because of a cycle) the entire probability is below 100%, the residual uncertainty has to be extrapolated by *PAS*.

5 Findings: Potentials of this analysis approach to guide the management of software development

The analysis approach in *PAS* could guide decision making. In this section the contribution will be elaborated.

Creative processes which develop software are hard to manage and to design. Especially in times, when scrum teams change their method of developing software depending on the current mission, a wide range of possible process alternatives exists. The application of *PAS* during the validation indicates the potentials of the analysis approach. An algorithm which assesses the profitability of a process has potentials to guide the management of software development. The investigated potentials are interesting and could help in conducting the further development of solutions to guide the management of software development.

Potential of evaluated assumptions: The design of a process has a major impact on the profitability of business operations. Managers make assumptions about the behaviour of a process. These assumptions have to be the basis for a process design. Management gets the possibility to check if a planned effect on the cost decreases of one activity really enables the decrease of overall process costs. If the assumptions do not have the planned effect on the entire process costs, management gets the possibility to determine new process changes to improve the profitability of the process.

Potential of keeping parameters in focus: The presented approach leads the attention of the management to the parameters which influence the profitability. It has the potential to change the way of thinking about a redesign of a process. Management could start looking for ways to shift the profitability of a transition between two activities more than thinking of ways to decrease the costs of an activity which would not be executed often.

Potential of iteration boundaries: Cycles of iterations have a major impact on the profitability of a process. The given algorithm allows to determine the right limit of repetitions of an activity. It could be possible to determine an upper bound, to limit the repetition of cycles. In higher iterations, the possibility of a cost explosion would increase rapidly. The limit serves as a guidance for management. If a concrete case leads to an exceeding of the limit, it would be the best to abort the treatment of the case. In practice, this could lead to a loss, however, because of a high probability that the further costs explode. This way of hindering a cost explosion will be economically reasonable.

The illustration in Figure 4 shows the cost explosion after iterative cycles (because of development cycles).

In Figure 4, the relation of a cost explosion under the terms of increasing iterations is illustrated. The calculated results belong to a new modified version of the development process which serves as an example. It makes clear that after the fourth repetition of the need analysis, the costs of the process will explode. In contrast, just after the sixth iteration of the concept, the probability of a cost explosion claims to be 100%.

On the other side, next to the potentials, the approach of using a process analysis is limited. The limitations could lead to further improvements.

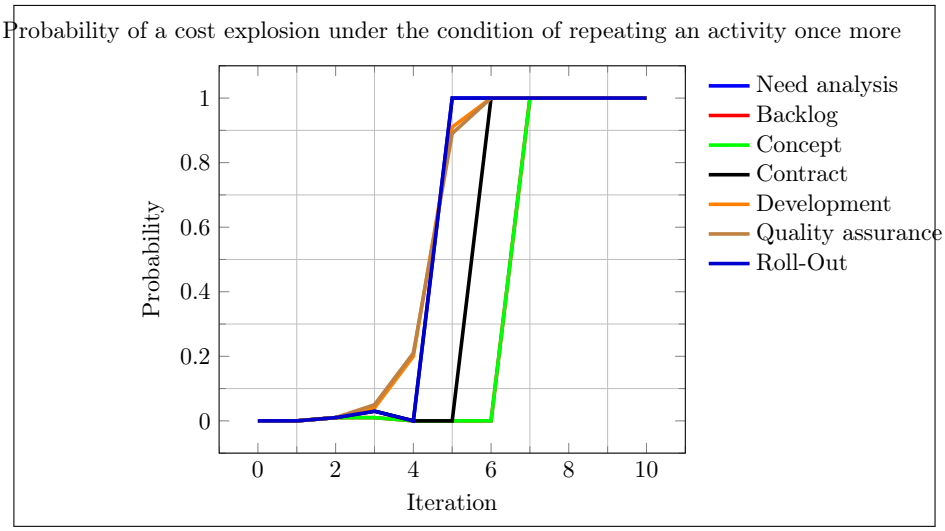

Fig. 4. Probability of a cost explosion on increasing iterations

Limit of correct information entered: In practice, it is easy to make assumptions on the probability. To obtain realistic data about the probabilities, a process mining could be executed. But the execution of a process mining approach might lead to some effort.

Limit of manipulations: It is possible to manipulate the parameters to reach the desired result. The manipulation may happen directly or subconsciously. If a person is convinced that an action has the right effect, he or she could modify the parameters until the wished results occur.

6 Conclusion

To support the abundance of an organisation which has undertaken business process re-engineering initiatives, with the aim of improving organisational performance [29], a process analysis is helpful. The proof of the developed algorithm which helps the management was successful. The prototype *PAS* empowers management to review the economic performance of a process model easily. A process could be valued in terms of its expectable costs. A process change could also be assessed. The presented approach for the analysis offers multiple potentials to support the decision making of management in daily routine.

Based on these results, a future version of *PAS* could be developed. The analysis of the capacity or the performance of a process might lead to reasonable results. In addition, it might be interesting to analyse the processing time of a process. This aspects might be sensible to consider in future analysis.

The inclusion of business planning aspects in the design of a process might empower the designer to plan a process which fits into future workloads. The

design of the process has to be able to handle the orders. The integration of future workloads will show if participants are overloaded [7]. To enable this integration, a calculation of resource usage [26] ought to be possible during the analysis. The design of a process, in combination with future workloads, allows the presentation of information about upcoming activities to the employees [7]. This empowers management to prevent the emergence of bottlenecks [7] which harm the profitability of the operations.

Beside the extension of the functionality of the analyst, a diversification of the implemented parameters could empower management to obtain more precise information on the performance of the process. In reality, the costs and the probabilities (parameters) would not have static values. It is possible that the values of the parameters change, depending on the business case in the process. It is likely that the possibility for the cycle back will decrease, the further the process proceeds. The progress (temporal and substantial) causes an increasing possibility for the exit of the process. These assumptions on the relation have to be investigated. The probability of the transition between two activities also depends on the case which is processed. In the context of the software development, cases might be requirements. The probability could be influenced by the complexity of a requirement. The same applies to the costs of an activity. In the context of the software development, the costs could depend on the effort of the requirement. In addition to the case, the costs and probabilities will be influenced by the number of the current iteration.

Bibliography

[1] Witold Abramowicz, Agata Filipowska, Monika Kaczmarek, Carlos Pedrinaci, Monika Starzecka, and Adam Walczak. Organization structure description for the needs of semantic business process management. In *3rd international Workshop on Semantic Business Process Management colocated with 5th European Semantic Web Conference*, Poznań, Poland, January 2008. ResearchGate.

[2] Barry Boehm. Value-based software engineering: Overview and agenda. 2005.

[3] Yen Cheung and Jay Bal. Process analysis techniques and tools for business improvements. *Business Process Management Journal*, 4(4):274–290, 1998.

[4] Robin Cooper and Robert S Kaplan. Measure costs right: make the right decisions. *Harvard business review*, 66(5):96–103, 1988.

[5] Thomas H Davenport. *Process Innovation - Reengineering Work through Information Technology.* Havard Business School Press, Boston, Massachusetts, 1993. Ernst & Young Center for Information Technology and Strategy.

[6] Thomas H Davenport. The coming commoditization of processes. *Harvard business review*, 83(6):100–108, 2005.

[7] Johann Eder, Horst Pichler, Wolfgang Gruber, and Michael Ninaus. Personal schedules for workflow systems. In Wil M.P. van der Aalst, Arthur ter Hofstede, Mathias Weske, Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen, editors, *Business Process Managemment*, number LNCS 2678 in Lecture Notes in Computer Science, pages pp. 216–231, Einhoven, The Netherlands, June 26-27 2003. Springer. International Conference on Business Process Management (BPM 2003).

[8] Bruce G Ferrin and Richard E Plank. Total cost of ownership models: An exploratory study. *Journal of Supply chain management*, 38(2):18–29, 2002.

[9] Frank Field, Randolph Kirchain, and Richard Roth. Process cost modeling: strategic engineering and economic evaluation of materials technologies. *Jom*, 59(10):21–32, 2007.

[10] A Gunasekaran and B Kobu. Modelling and analysis of business process reengineering. *International Journal of Production Research*, 40(11):2521–2546, 2002.

[11] Vlatka Hlupic and Stewart Robinson. Business process modelling and analysis using discrete-event simulation. In *Proceedings of the 30th conference on Winter simulation*, pages 1363–1370. IEEE Computer Society Press, 1998.

[12] Eric Jacquier, Nicholas G Polson, and Peter E Rossi. Bayesian analysis of stochastic volatility models. *Journal of Business & Economic Statistics*, 12(4):69–87, October 1994.

[13] David G Kendall. Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. *The Annals of Mathematical Statistics by the Institute of Mathematical Statistics*, Volume 24(Number 3):338–354, 1953.

[14] David Kuhlen and Andreas Speck. Wertanalyseverfahren für kundenanforderungen. In Erhard Plödereder, Lars Grunske, Eric Schneider, and Dominik Ull, editors, *INFORMATIK 2014 Big Data - Komplexität meistern*, volume P-232 of *Lecture Notes in Informatics (LNI) - Proceedings*, pages 2317 – 2322, Stuttgart, September 2014. Gesellschaft für Informatik e.V. (GI). Thanks to Prof. Dr. Andreas Speck and Prof. Dr. Hinrich Schröder.

[15] Huiping Lin, Yushun Fan, and Stephen T Newman. Manufacturing process analysis with support of workflow modelling and simulation. *International Journal of Production Research*, 47(7):1773–1790, 2009.

[16] Charles AS Marrin, Lisa C Johnson, Virginia L Beggs, and Paul B Batalden. Clinical process cost analysis. *The Annals of thoracic surgery*, 64(3):690–694, 1997.

[17] D. C. McDermid. Integrated business process management: Using state-based business rules to communicate between disparate stakeholders. In Wil M.P. van der Aalst, Arthur ter Hofstede, Mathias Weske, Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen, editors, *Business Process Managemment*, Lecture Notes in Computer Science, Einhoven, The Netherlands, 2003. Interational Conference BPM, Springer.

[18] Hans-Jürgen Plewan and Benjamin Poensgen. *Produktive Softwareentwicklung: Bewertung und Verbesserung von Produktivität und Qualität in der Praxis*, volume 1. Auflage. dpunkt. verlag, Heidelberg, Germany, Juli 2011.

[19] William F Pounds. The process of problem finding. *Library of the Massachusetts Institute of Technology*, 1965.

[20] August-Wilhelm Scheer and Markus Nüttgens. *ARIS architecture and reference models for business process management*. Springer, 2000.

[21] Andreas Speck, Sven Feja, Sören Witt, Elke Pulvermuller, and Marcel Schulz. Formalizing business process specifications. *Computer Science and Information Systems*, 8(2):427–446, 2011.

[22] Andreas Speck, Elke Pulvermüller, and Dirk Heuzeroth. Validation of business process models. In *Proceedings of ECOOP 2003 Workshop Correctness of Model-based Software Composition (CMC)*, pages 75–83, 2003.

[23] Patrik Spieß, Dinh Khoa Nguyen, Ingo Weber, Ivan Markovic, and Michael Beigl. Modelling, simulation, and performance analysis of business processes involving ubiquitous systems. In *Advanced Information Systems Engineering*, pages 579–582. Springer, 2008.

[24] Wil MP van der Aalst. Challenges in business process analysis. *LECTURE NOTES IN BUSINESS INFORMATION PROCESSING*, January November 2007.

[25] Wil MP Van Der Aalst, Arthur HM Ter Hofstede, and Mathias Weske. Business process management: A survey. In Wil M.P. van der Aalst, Arthur ter Hofstede, Mathias Weske, Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen, editors, *Business process management*, Lecture Notes in Computer Science, pages 1–12, Einhoven, The Netherlands, 2003. Interational Conference BPM, Springer.

[26] Wil MP Van der Aalst and Boudewijn F van Dongen. Discovering workflow performance models from timed logs. In *Engineering and Deployment of Cooperative Information Systems*, pages 45–63. Springer, 2002.

[27] Kostas Vergidis, Ashutosh Tiwari, and Basim Majeed. Business process analysis and optimization: Beyond reengineering. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(1):69–82, 2008.

[28] Jan Vom Brocke, Jan Recker, and Jan Mendling. Value-oriented process modeling: integrating financial perspectives into business process re-design. *Business Process Management Journal*, 16(2):333–356, 2010.

[29] Currie Weerakkody. Integrating business process reengineering with information systems development; issues & implications. In Wil M.P. van der Aalst, Arthur ter Hofstede, Mathias Weske, Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen, editors, *Business Process Managemment*, Lecture Notes in Computer Science, Einhoven, The Netherlands, 2003. Interational Conference BPM, Springer.