Information Systems 34 (2009) 87–107

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Toward data mining engineering: A software engineering approach

Oscar Marbán *, Javier Segovia, Ernestina Menasalvas, Covadonga Fernández-Baizán

Facultad de Informática, Universidad Politécnica de Madrid (U.P.M.), Spain

ARTICLE INFO

Article history:
Received 19 February 2008
Received in revised form
22 April 2008
Accepted 22 April 2008
Recommended by: D. Shasha

Keywords:
Data mining
Software engineering
Knowledge engineering

ABSTRACT

The number, variety and complexity of projects involving data mining or knowledge discovery in databases activities have increased just lately at such a pace that aspects related to their development process need to be standardized for results to be integrated, reused and interchanged in the future. Data mining projects are quickly becoming engineering projects, and current standard processes, like CRISP-DM, need to be revisited to incorporate this engineering viewpoint. This is the central motivation of this paper that makes the point that experience gained about the software development process over almost 40 years could be reused and integrated to improve data mining processes. Consequently, this paper proposes to reuse ideas and concepts underlying the IEEE Std 1074 and ISO 12207 software engineering model processes to redefine and add to the CRISP-DM process and make it a data mining engineering standard.

1. Introduction

In its early days, software development focused on creating programming languages and algorithms that were capable of solving almost any problem type. The evolution of hardware, continuous project planning delays, low productivity, heavy maintenance expenses and failure to meet user expectations had led by 1968 to the stagnation of software development, causing what came to be known as the *software crisis*, the term coined at the first NATO conference on software development [1]. This crisis was caused by the fact that there were no formal methods and methodologies, support tools or proper development project management, all of which were standard techniques used in projects developed in other classical branches of engineering. The software community realized what the problem was and decided to borrow ideas from other fields of engineering, which it incorporated into software project development. This was the origin of software engineering (SE). As of then process models and methodologies for developing software projects began to materialize.

Software process models describe the tasks to be performed to develop a software system, whereas development methodologies schedule the tasks and specify what methods to use to do the tasks [2]. Software development improved considerably as a result of the new methodologies. This solved some of its earlier problems, and little by little software development grew to be a branch of engineering. This shift means that project management and quality assurance problems are being solved. Additionally, it is helping to increase productivity and improve software maintenance. This is one of the major problems in software development, as it can amount to up to two-thirds of costs throughout the software system's lifetime [2].

The history of knowledge discovery in databases (KDD), now known as data mining (DM), is not much different, at least so far. In the early 1990s, when the KDD processing term was first coined [3], there was a rush to develop DM algorithms that were capable of solving all a company's problems of searching for knowledge in large

* Corresponding author at: DLSIIS, Facultad de Informática, Universidad Politécnica de Madrid (U.P.M.), Campus de Montegancedo s/n, 28660 Boadilla del Monte, Madrid, Spain. Tel.: +34 913367388; fax: +34 913367393.

E-mail addresses: omarban@fi.upm.es (O. Marbán), fsegovia@fi.upm.es (J. Segovia), emenasalvas@fi.upm.es (E. Menasalvas), cfbaizan@fi.upm.es (C. Fernández-Baizán).

0306-4379/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.is.2008.04.003

volumes of data. Apart from developing algorithms, tools (Clementine [4–6], IBM Intelligent Miner [7,8], Weka [9], DBMiner [10]) were also developed to simplify the application of DM algorithms and provide some sort of support for all the activities involved in the KDD process.

From the viewpoint of DM process models, the year 2000 marked the most important milestone, as this was when the first standard and tool-independent DM process model was published. This standard is known as CRISP-DM (*CRoss-Industry Standard Process for DM*) [11,12].

The number of applied projects in the DM area is expanding rapidly [13]. This growth is confirmed by reports by the Gartner Group [14,15] and Forrester Research [16]. The Gartner Group estimates [14] that there will be an upsurge of DM projects over the next decade (over 300%) to improve customer relationships and help companies listen to customers. Another Gartner Group report [15] claims that enterprises in the DM area grew by 4.8% from 2005 to 2006, and DM is now the area in which companies are investing most. While it is true that a lot of DM projects are being developed, neither all the project results are in use [17–19] nor do all projects end successfully [20,21]. The failure rate is actually as high as 60% [22]. Deployed by about 50% of respondents, CRISP-DM is the most commonly used methodology for developing DM projects [23–25]. However, its use is not becoming any more widespread due to rivalry with other, in-house methodologies developed by work teams, which account for another, almost 30%.

All the above goes to show that while CRISP-DM was an improvement on the earlier state of affairs, the process model is not perhaps yet mature enough to deal with the complexity of the problems it has to address. And this detracts from the effectiveness of its deployment, as it does not produce the expected results.

Are we at the same point as SE was in 1968? Certainly not, but we do not appear to be on a par yet either, DM cannot be considered a mature field as SE [26]. Table 1 compares DM's history with SE's past. Looking at the KDD process and how it has progressed, we find that there is some parallelism with the advancement of software. From this viewpoint, DM project development is at stage 4, and is defining development methodologies to be able to cope with the new project types, domains and applications that organizations have to come to terms with. SE has reached stage 5, where development processes pay special attention to organizational, management or other parallel activities not directly related to development, such as project completeness and quality assurance. CRISP-DM has not yet been sized for these tasks, as it is very much focused on pure development activities and tasks.

This paper is moved by the idea that DM problems are taking on the dimensions of an engineering problem. Therefore, the processes to be applied should include all the activities and tasks required in an engineering process, tasks that CRISP-DM might not cover. Our proposal is to enhance CRISP-DM by embedding other current standards, as suggested in [27], inspired by the work done recently in SE derived from other branches of engineering and from developer experience.

2. DM process models

There is some confusion about the terminology different authors use to refer to process and methodology. Below we describe the definitions of standard SE terminology. These terms are used with the aim of unifying criteria, because they are better established and backed by the International Organization for Standardization (ISO) or the Institute of Electrical and Electronics Engineers (IEEE).

Table 1
Parallelism between DM and SE

SE phase	DM phase	DM characteristics	SE characteristics
Phase 1 (1945–1955)	Phase 1 (…–1990)	Gathering knowledge hidden in data was a hard thing to do Statistical techniques Machine learning	Programming was a hard thing to do Use of machine and assembly language
Phase 2 (1955–1965)	Phase 2 (1990–1995)	DM and a lot of algorithms appeared. DM tools appeared All sorts of things could be done	A host of languages appeared All sorts of things could be done
Phase 3 (1965–1970)	Phase 3 (1995–1999)	DM projects went unfinished Errors, continuous changes, unpredictable costs Nothing could be done DM environments	Program development went unfinished Inefficiency, errors, unpredictable cost Nothing could be done
Phase 4 (1970–1980)	Phase 4 (1999–…)	Process models: CRISP-DM DM methodologies: SEMMA, 5A's	Programming fundaments Design methodologies Program verification
Phase 5 (1980–…)	Phase 5 (?–?)	Unknown	Programming environments Formal specification Automated programming Software quality Human resources management

A process model can be defined as the set of tasks to be performed to develop a particular element, as well as the elements that are produced in each task (outputs) and the elements that are necessary to do a task (inputs) [2]. The ultimate goal of a process model is to make the process repeatable, manageable and measurable (to be able to get metrics). A good process model should be [28,29]:

- Effective: An effective process should help us produce the right product.
- Maintainable: So we can quickly and easily find and remedy faults or work out where to make changes.
- Predictable: Any new product development needs to be planned, and those plans are used as the basis for allocating resources: both time and people. A good process will help us do this. The process helps lay out the steps of development.
- Repeatable: If a process is found to work, it should be replicated in future projects. Ad hoc processes are rarely replicable unless the same team is working on the new project. Even with the same team, it is difficult to keep things exactly the same.
- Quality: Quality in this case may be defined as the product's fitness for its purpose.
- Improvable: No one would expect their process to reach perfection and need no further improvement itself. Even if we were as good as we could be now, both development environments and requested products are changing so quickly that our processes will always be running to catch up.
- Traceable: A defined process should allow the project staff to follow the status of a project.

Methodology can be defined as the instance of a process model that both lists tasks, inputs and outputs and specifies how to do the tasks [2]. Tasks are performed using techniques that stipulate how they should be done. After selecting a technique to do the specified tasks, tools can be used to improve task performance. These tools implement the techniques and improve task performance. Briefly, process models denote *what to do*, whereas methodologies indicate *how to do it*.

Finally, the life cycle determines the order in which each activity is to be done [30]. A life cycle model is the description of the different ways of developing a project. A life cycle's primary functions are:

- To determine the order of the project development stages and processes.
- To establish the transition criteria for moving from one stage to the next (intermediate products). All this includes criteria for verifying that the current stage has been completed and criteria for selecting and starting the next stage.

Life cycles provide guidance on the order (stages, activities, prototypes, validations, etc.) in which the key project activities should be performed. A project's success will depend on the life cycle selected to develop it, as the life cycle can help to assure that each step taken leads to the achievement of the goal. A poorly selected life cycle can lead to continual delays and unnecessary rework. Life cycle selection depends on many variables that the project manager should consider, such as how much time you have until the customer wants to see results, how well specified the requirements are or the size of the project.

From the viewpoint of the above definitions, what do we have in DM? Does DM have process models and/or methodologies [27]? The KDD process [31] has a process model component because it establishes all the steps to be taken (what to do) to develop a DM project, but it is not a methodology because its definition does not set out how to do each of the proposed tasks. It is also a life cycle, specifically a waterfall life cycle plus feedback, as developers can go back to the last stage to put right any error detected in any of the stages. It is not an iterative life cycle, because there are no iterations or small planned advances in project development. The project is developed as a whole. Like the KDD process, Two Crows [32,33] is a process model and waterfall life cycle. At no point does it set out how to do the established DM project development tasks. SEMMA [34,35] is the methodology that SAS proposed for developing DM products. Although it is a methodology, it is based on the technical part of the project only, i.e. its aim is to solve the DM part and it does not take into account all the management side. Like the above approaches, SEMMA also sets out a waterfall life cycle, as the project is developed through to the end. If the solution is not interesting, developers go backwards through the stages. The 5 A's [36] is a process model that proposes the tasks that should be performed to develop a DM project and was one of CRISP-DM's forerunners. Therefore, their philosophy is the same: it proposes the tasks but at no point suggests how they should be performed. The life cycle is similar to the one proposed in CRISP-DM. The 6-σ [37,38] is in principle a development paradigm for projects of any type. It focuses on the quality of the project results. Specializing 6-σ to the DM environment, it becomes a process model and a life cycle similar to the one proposed by 5 A's. CRM catalyst [39] is a methodology for developing CRM systems. CRM systems are divided into three main parts: collaborative CRM, operating CRM and analytical CRM. The last part uses DM. CRM catalyst defines the tasks to be performed to develop a CRM system, but also defines how they should be performed. Therefore, it is methodology incorporating a life cycle. In this case, the life cycle is iterative, as the CRM system is built as small increments, not as a at one go. DM industrial engineering [40] is a methodology because it specifies how to perform the tasks to develop a DM project in the field of industrial engineering. It is an instance of CRISP-DM, which makes it a methodology, and it shares CRISP-DM's associated life cycle. Market Consulteks [41] integrates the technological part of DM into the RUP software development methodology [42], which also defines the requirements-driven iterative and incremental life cycle. Finally, CRISP-DM [11,43,44] states which tasks have to be carried out to successfully complete a DM project, making it a process model. It is also a waterfall life cycle, as it suggests no more than one iteration on requirements and tackles the problem as a

whole or at most divides it into different problems. This way DM models are built for each identified subproblem. This increases the number of DM models output by the project. CRISP-DM also has a methodological component, as it gives recommendations on how to do some tasks. However, it just proposes other tasks, giving no guidance about how to do them. Therefore, we class CRISP-DM as a process model.

In the next section, we analyze SE process models for comparison with CRISP-DM.

3. SE process models

The SE panorama is quite a lot clearer, and there are two well-established process models: IEEE Standard 1074 [45] and ISO 12207 [46]. In the following, we will analyze both processes in some detail and propose a generic joint process model. This joint model will then be used for comparison with and, if necessary, to expand the CRISP-DM.

3.1. IEEE STD 1074

The IEEE Std 1074 [45] specifies the software life cycle processes for developing and maintaining software. It determines a non-time-ordered set of essential activities that should be part of developing a software product. The life cycle that should be followed to develop the product is selected and established by the project manager for each project. IEEE Std 1074 neither defines nor prescribes a particular life cycle. Each organization using the standard should instantiate the activities specified in the standard within its own development process.

Fig. 1(a) shows the key processes defined in this process model. The *software life cycle selection process* identifies and selects a life cycle for the software under construction. Possible life cycle models are identified and analyzed for a project based on the type of software product under development and the project requirements, and a model that properly supports the project is then selected. The *project management processes* are the set of processes that establish the project structure, and coordinate and manage project resources throughout the software life cycle. *Development-oriented processes* start with the identification of a need for automation. It may take a new application or a change of all or part of an existing application to satisfy this need. With the support of the integral process activities and under the project management plan, the development processes produce software (code and documentation) from the statement of the need. Finally, the activities for installing, operating, supporting, maintaining and retiring the software product should be performed. *Integral processes* are necessary to successfully complete the software project activities. They are enacted at the same time as the software development-oriented activities and include activities that are not related to development. They are used to assure the completeness and quality of the project functions.

3.2. ISO 12207

ISO 12207 divides the activities that can be carried out during the software life cycle into five primary processes (*primary life cycle processes*), eight supporting processes (*supporting life cycle processes*) and four organizational processes (*organizational life cycle processes*), as shown in Fig. 1(b). Each life cycle process is divided into a set of activities, and these activities are further divided into a set of tasks.

The *primary* processes are a compendium of five processes that serve the primary parties throughout the software life cycle. A primary party is the party that starts or enacts software development, operation or maintenance. The primary parties are the acquirer, supplier, planner, developer, operator and maintainer of a software system or product. The activities and tasks of a primary process are the responsibility of the organization that starts and enacts this process.

The *supporting* processes support other processes as an integral part with a distinct purpose and contribute to the success and quality of the software project. The supporting processes are divided into eight subprocesses, any of

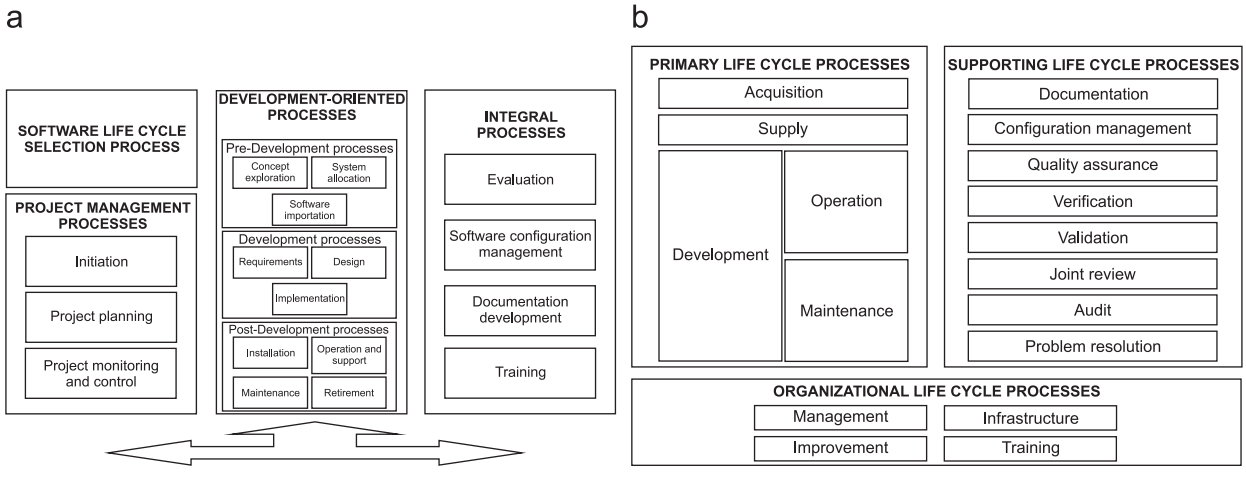

Fig. 1. Software process models. (a) IEEE 1074 and (b) ISO 12207.

which can be used in the acquisition, supply, strategic planning, development, operation or maintenance processes or any other supporting processes. The supporting processes are used at several points of the life cycle and can be enacted by the organization that uses them, by a separate organization as a service or by a customer as a planned or contracted part of the project. The supporting process activities and tasks are the responsibility of the organization that uses and enacts the process in question. The organization that uses and enacts a supporting process manages that process at project level as per the management process, establishes an infrastructure for the process as per the infrastructure process and drives the process at the organizational level as per the improvement process.

The *organizational* processes are used by an organization to perform organizational functions, such as management, personnel training or process improvement. These processes help to establish, implement and improve software process, achieving a more effective organization. They tend to be enacted at the corporate level and are outside the scope of specific projects and contracts. However, the lessons learned from projects and contracts contribute to improving the organization. The organizational process activities and tasks are the responsibility of the organization using the process.

3.3. Unification of IEEE STD 1074 and ISO 12207

Having reviewed IEEE Std 1074 and ISO 12207, the goal is to build a joint process model that is as generic as possible to then try to use it as a basis for defining a process model against which to compare CRISP-DM.

Fig. 2 shows the correspondence between IEEE Std 1074 and ISO 12207. Clearly, most of the processes proposed in IEEE Std 1074 match up with ISO 12207 processes and vice versa. To get a joint process model we have merged IEEE Std 1074 and ISO 12207 processes. The process selection criterion was to select the most thoroughly defined IEEE Std 1074 and ISO 12207 processes and try not to merge processes from different groups in different process models. According to this criterion, we selected IEEE Std 1074 as a basis, as its processes are more detailed. Additionally, we added the ISO 12207 *acquisition* and *supply processes*, because IEEE Std 1074 states that ISO 12207 acquisition and supply processes should be used [45] if it is necessary to acquire or supply software.

Fig. 3 shows the joint process model developed after studying IEEE Std 1074 and ISO 12207 according to the above criteria.

Fig. 3 also shows the details of the major process groups, the activities they each involve according to the selected standard for that process group.

In the next section we will analyze which of the above activities CRISP-DM includes and which it does not in order to try to build a process model for DM projects that is as complete as possible and that organizes activities logically.

4. CRISP-DM

Analyzing the problems of DM projects, a group of highlighted enterprises (Teradata, SPSS (ISL), Daimler-Chrysler and OHRA) that develop DM projects, proposed a reference guide to develop DM projects. This guide is called CRISP-DM [11]. CRISP-DM is vendor-independent so it can be used with any DM tool and it can be applied to solve any DM problem.

CRISP-DM defines the phases that we have to do in a DM project. CRISP-DM also defines for each phase the tasks and the deliverables for each task. CRISP-DM is divided in six phases (see Fig. 4). The phases are described in the following.

- Business understanding: This phase focuses on understanding the project objectives and requirements from a business perspective, then converting this knowledge into a DM problem definition and a preliminary plan designed to achieve the objectives.
- Data understanding: The data understanding phase starts with an initial data collection and proceeds with activities in order to get familiar with the data, to

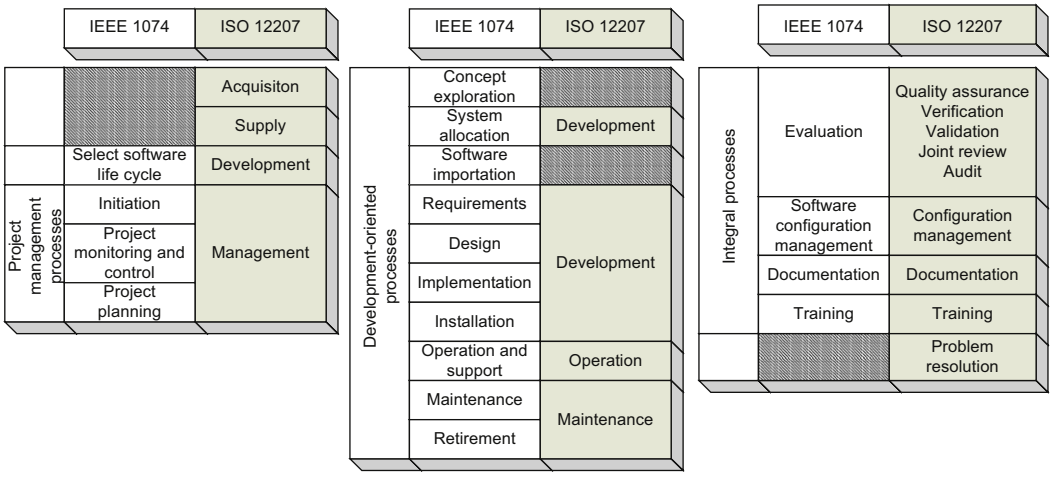

Fig. 2. Mapping ISO 12207 to IEEE Std 1074.

PROCESS	ACTIVITY
Acquisition	
Supply	
Software life cycle selection	Identify available software life cycles
	Select software life cycle
Project management processes	
Initiation	Create software life cycle process
	Allocate project resources
	Perform estimations
	Define metrics
Project monitoring and control	Manage risks
	Manage the project
	Retain records
	Identify software life cycle process improvement needs
	Collect and analyze metric data
Project planning	Plan evaluations
	Plan configuration management
	Plan system transition
	Plan installation
	Plan documentation
	Plan training
	Plan project management
	Plan integration
Development-oriented processes	
Pre-development	
Concept exploration	Identify ideas or needs
	Formulate potential approaches
	Conduct feasibility studies
	Refine and finalize the idea or need
System allocation	Analyze functions
	Decompose system requirements
	Develop system architecture
Software importation	Identify imported software requirements
	Evaluate software import sources
	Define software import method
	Import software
Development	
Requirements	Define and develop software requirements
	Define interface requirements
	Prioritize and integrate software requirements
Design	Perform architectural design
	Design data base
	Design interface
	Perform detailed design
Implementation	Create executable code
	Create operating documentation
	Perform integration
Post-Development	
Installation	Distribute software
	Install software
	Accept software in operational environment
Operation and support	Operate the system
	Provide technical asístanse and consulting
	Maintain support request log
Maintenance	Identify software improvement needs
	Implement problem reporting method
	Maintenance support request log
Retirement	Notify user
	Conduct parallel operations
	Retire system
Integral processes	
Evaluation	Conduct reviews
	Create traceability matrix
	Conduct audits
	Develop test procedures
	Create test data
	Execute test
	Report evaluation results
Software configuration management	Develop configuration identification
	Perform configuration control
	Perform status accounting
Documentation development	Implement documentation
	Produce and distribute documentation
Training	Develop training materials
	Validate the training program
	Implement the training program

Fig. 3. Joint process model.

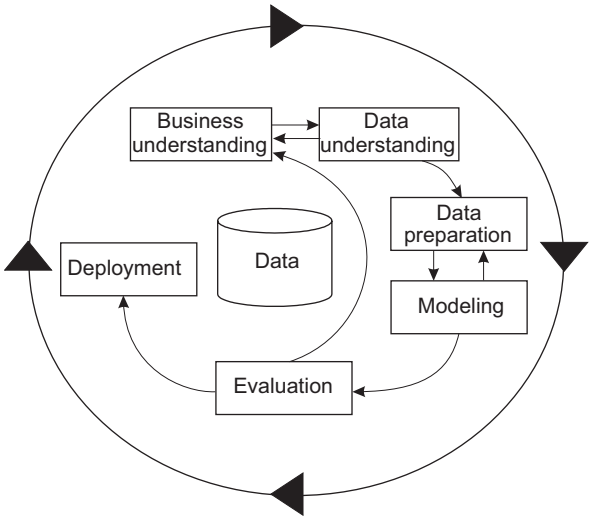

Fig. 4. CRISP-DM process model [11].

identify data quality problems, to discover first insights into the data or to detect interesting subsets to form hypotheses for hidden information.

- Data preparation: The data preparation phase covers all activities to construct the final data set from the initial raw data. Data preparation tasks are likely to be performed multiple times and not in any prescribed order.
- Modeling: In this phase, various modeling techniques are selected and applied and their parameters are calibrated to optimal values. Typically, there are several techniques for the same DM problem type. Some techniques have specific requirements on the form of data. Therefore, stepping back to the data preparation phase is often necessary.
- Evaluation: At this stage of the project a model (or models) will have been built that are of seemingly high quality from a data analysis perspective. Before proceeding to final model deployment, it is important to evaluate the model more thoroughly and review the steps taken to construct the model to be certain that it properly achieves the business objectives. At the end of this phase, a decision on the use of the DM results should be reached.
- Deployment: Creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that the customer can use it.

Table 2 presents an outline of phases and generic tasks that CRISP-DM proposes to develop a DM project.

We have chosen CRISP-DM because it is the "facto standard" to develop DM projects. In addition, CRISP-DM

Table 2
CRISP-DM phases and tasks

Business understanding	Data understanding	Data preparation	Modeling	Evaluation	Deployment
Determine business objectives	Collect initial data	Select data	Select modeling techniques	Evaluate results	Plan deployment
Assess situation	Describe data	Clean data	Generate test design	Review process	Plan monitoring and maintenance
Determine DM objectives	Explore data	Construct data	Build model	Determine next steps	Produce final report
Produce project plan	Verify data quality	Integrate data Format data	Assess model		Review project

SE Process Model		CRISP-DM	
Process	**Task**	**Process**	**Task**
Acquisition			
Supply			
Software life cycle selection	Identify available software life cycles		
	Select software life cycle		

Fig. 5. Comparison of life cycle selection processes.

is the most commonly used methodology to develop DM projects [24,23].

5. SE process model vs. CRISP-DM

This section presents a comparison between CRISP-DM and the joint process model discussed in Section 3.3. This comparison should identify what SE model elements (activities, tasks) are applicable to DM projects and are not covered by CRISP-DM. This way it will be possible to build a process model for DM projects based on fairly mature SE process models.

Note that the correspondence between CRISP-DM and SE process model elements is not exact. In some cases, the elements are equivalent, but the techniques are different. In other cases, the elements have the same goal but are implemented completely differently. This obviously depends on the project type. In one case the project aim is to develop software and in the other it is to gather knowledge from data.

5.1. Acquisition, supply and life cycle selection process

The purpose of the set of processes[1] for selecting the life cycle (*Life cycle selection*) is in software projects (Fig. 5) to *identify* and *select a life cycle* for the software project that is to be developed. Possible life cycle models are identified and analyzed based on the type of software product to be developed and the project requirements. Then a model that provides proper support for the project is selected. This set of processes also extends to third party software *acquisition* and *supply*. These two processes cover all the tasks related to supply or acquisition management.

CRISP-DM does not include any of the *acquisition* or *supply* processes at all. Author's own experience in DM project development suggests that acquisition and supply processes may be considered necessary and third parties engaged to develop or create DM models for projects of some size or complexity. Their management should therefore be specified as processes.

Developers undertaking a DM project also need to *select a life cycle*. The life cycle depends on the type of project to be developed. Life cycle models are used for software development because not all projects are equal, and not all developers and clients have the same needs. Developing a more or less everyday piece of software (e.g., a common management application) has nothing to do with building a totally unknown piece of software (e.g., control software for a nuclear power plant). This also applies to DM projects. A typical client segmentation, which is the first thing any data miner learns how to do, is quite a different kettle of fish from predicting aircraft faults, where everything is new, there are a huge number of variables, and all the possible techniques have to be tested. Therefore, even if the activities are similar, the order and way in which they are performed will not be the same.

Life cycle selection is not an easy task, as it involves weighing up the project type in terms of complexity, experience in the problem domain, knowledge of the data that are being analyzed, variability and data expiration. The life cycle selection process would then be useful for

[1] "Software" has been removed from the title because DM does not produce software.

DM projects. However, DM project life cycles will have to be defined, as no thorough studies have yet been conducted on possible life cycles for use or the variables or criteria that distinguish one life cycle from another.

5.2. Project management processes

The set of processes defined here (Fig. 6) establish the project structure, and coordinate and manage project resources throughout the project life cycle. The project *initiation* process defines the activities for creating and updating the project development or maintenance infrastructure throughout the life cycle. *Project planning* covers all the processes related to planning project management activities, including contingency planning. The *project monitoring and control process* analyses technical, economic, operational, support and scheduling risks. It aims to identify potential problems, determine the likelihood of their occurrence and their impact and establish the steps for their management. Additionally, it also covers subprocesses related to project metric management.

Project management processes are evidently also necessary when a DM project is undertaken. DM projects are high risk. Consequently, the tasks that are to be performed need to be planned, and a contingency plan is necessary. Also it is necessary to analyze project costs, benefits and ROI. Looking at the tasks covered by the CRISP-DM stages; however, only the *business understanding* (*BU*) phase includes any project management-related tasks. The *identify major iterations* task is comparable to mapping activities for the selected life cycle, except that the DM project iterations are only roughly outlined as there are no defined DM life cycles. Additionally, the philosophy behind the *experience documentation* task is the same as the *identify software life cycle process improvement needs* task, i.e. it aims to document and improve how the project is developed after CRISP-DM is deployed in the organization.

CRISP-DM's *inventory of resources* task accounts for resources allocation, although it tends to identify what resources are available rather than allocating resources across the project. CRISP-DM does not cover this issue.

The other tasks proposed by CRISP-DM directly match up with the SE process model tasks. And all the tasks that do not appear in CRISP-DM are considered necessary in a DM project.

However, CRISP-DM's biggest snag in terms of project management is related to metrics (*define metrics*, *retain records*, *collect and analyze metrics*). For the most part, this can be attributed to the field's immaturity. There is a need to define DM metrics to establish costs and deviations throughout project execution. The other major omission is the evaluation component (*plan evaluations*). CRISP-DM does have a results evaluation stage, but this component refers to process evaluation as a whole. Tasks need to be devised in order to evaluate each completed stage transversally across all stages.

Configuration management (*plan configuration management*) aims to manage versions, changes and modifications of each project element. CRISP-DM does not cover DM project configuration management, but, because of the size of current projects and the teams of human resources working together on such projects, we believe that it should. Different people generate multiple versions of models, initial data sets, documents, etc., in a project. Unless they are well located and managed, it is very difficult to go back to earlier versions, should the current versions not be valid. Also there is a risk of models, data and documentation for different versions getting mixed up.

	SE Process Model		CRISP-DM	
	Process	**Task**	**Process**	**Task**
Project Management Processes	**Initiation**	Create software life cycle process	BU: Produce project plan	Identify major iterations
		Allocate Project resources	BU: Assess situation	Inventory of resources
		Perform estimations	BU: Assess situation	Cost and benefits
			BU: Produce project plan	Project plan
		Define metrics		
	Project monitoring and control	Manage risks	BU: Assess situation	Risks and contingencies
		Manage the project	BU: Produce project plan	Produce project plan
		Retain records		
		Identify software life cycle process improvement needs	D: Review project	Experience documentation
		Collect and analyze metrics		
	Project planning	Plan evaluations		
		Plan configuration management		
		Plan system transition		
		Plan installation	D: Plan deploymnet	Plan deployment
		Plan documentation		
		Plan training		
		Plan project management		
		Plan integration	D: Plan monitoring and maintenance	Monitoring and maintenance plan

BU: Business Understanding
D: Deployment

Fig. 6. Comparison of management processes.

SE Process Model			CRISP-DM	
	Process	Task	Process	Task
Pre-Development Processes	Concept exploration	* Identify ideas or needs * Formulate potential approaches * Conduct feasibility studies * Refine and finalize the idea or need	BU: Determine business objectives	Background
				Business objectives
				Business success criteria
			BU: Assess situation	Inventory of resources
				Requirements, assumptions and constraints
			BU: Assess situation	Terminology
	System allocation	* Analyze functions * Develop system architecture *Decompose system requirements	BU: Determine business objectives	Background
				Business objectives
				Business success criteria
			BU: Produce project plan	Project plan
				Initial assessment of tools and techniques
	Software importation	Identify imported software requirements		
		Evaluate software import sources		
		Define software import method		
		Import software		

BU: Business Understanding

Fig. 7. Comparison of pre-development processes.

Additionally, any DM project should include tasks for managing the transfer and use of the results (*plan system transition*, *plan installation*), tasks that CRISP-DM does not cover either.

Finally, the other major oversight, fruit of process immaturity, is the documentation task (*plan documentation*). Reports are generated in all stages. However, there is no task aimed at planning what form this documentation should take if it is to conform to thorough standards. This would improve documentation evaluation and review and facilitate work on process improvement.

5.3. Development-oriented processes

Software development-oriented processes start with the identification of a need to automate some tasks to be performed by a computer (*identify ideas or needs*). A new application or a change of all or part of an existing application could be needed to satisfy this need. With the support of the *integral process* activities and subject to the project management plan (*plan project management*), the *development processes* produce the software (code and documentation) as of the statement of need. Finally, activities for installing (*installation*), operating (*operation and support*), supporting (*operation and support*), maintaining (*maintenance*) and retiring (*retirement*) the software product should be performed. These processes are grouped as *pre-development*, *development* and *post-development* processes.

DM projects start with the need to gather knowledge from an organization's data to help with business decision-making. This knowledge can be used directly or can be integrated into the organization's systems. In DM, this is the most mature set of processes at present, as all the existing "methodologies" for DM project development focus primarily on this part. Development starts with the need to gather knowledge, and the development processes produce this knowledge and its documentation. As in SE, these processes can also be divided into *pre-development*, *development* and *post-development* stages.

5.3.1. Pre-development

The *pre-development processes* (Fig. 7) are related to everything that has to be done before starting to build the software system, such as *concept exploration* or *system allocation requirements*. The *concept exploration* process includes identifying an idea or need (*identify ideas or needs*) for the system to be developed, and the formulation (*formulate potential approaches*), evaluation (*conduct feasibility studies*) and refinement of potential solutions at system level (*refine and finalize the idea or need*). Once the system limits have been established, a statement of need is generated for the system to be developed. This statement of need starts up the *system allocation* process and/or *requirements* process and feeds the *project management* processes. It is the document upon which all the later engineering work is based.

The statement of need is just as necessary in DM projects as in any other engineering project. It is a starting point for project development as it provides an understanding of the problem to be solved and establishes the supposed requirements for and constraints on the project to be developed. Because of its importance, CRISP-DM already accounts for this process, as shown in Fig. 7. However, it is spread across different tasks and always in the *BU* process at the start of the project. The *software importation* process is related to the reuse of existing software. This software can belong to the developers or be purchased from third parties. In the case of software, this process provides the means required to identify what requirements imported software can satisfy and assess the software to be used. Software does not, in principle, not need to be imported in a DM project, because a DM project gathers knowledge and does not develop software.

SE Process Model			CRISP-DM	
	Process	**Task**	**Process**	**Task**
Development Processes	**Requirements**	* Define and develop software requirements * Define interface requirements * Prioritize and integrate software requirements	BU: Assess situation	Requirements, assumptions and constraints
			BU: Determine Data Mining goals	Data Mining goals
	Design	Perform architectural design		
		Design data base	DU: Describe data	Data description report
			DU: Collect initial data	Initial data collection report
		Design interfaces		
		Perform detailed design		
	Implementation	* Create executable code * Create operating documentation * Perform integration	DU: Collect initial data	Initial data collection report
			DU: Describe data	Data description report
			DU: Explore data	Data exploration report
			DU: Verify data quality	Data quality report
			DP: Select data	Rationale for inclusion/exclusion
			DP: Clean data	Data cleaning report
			DP: Construct data	Derived attributes
			DP: Construct data	Generated records
			DP: Integrate data	Merged data
			DP: Format data	Reformatted data
			M: Select modeling technique	Modeling technique
			M: Select modeling technique	Modeling assumptions
			M: Build model	Parameter settings
			M: Build model	Model
			M: Assess model	Model assessment
			M: Assess model	Revised parameter settings
			E: Evaluate results	Assessment of data mining results with respect to business success criteria
			E: Determine next steps	List of possible actions
			E: Determine next steps	Decision

BU: Business Understanding DU: Data Understanding DP: Data Preparation M:Modeling E:Evaluation

Fig. 8. Comparison of development processes.

Its equivalent in a DM project would be to import existing DM models that are useful for the current project. For example, one common practice is to have a client clustering and use that clustering in the ongoing project to classify clients. To do this, you need to have imported the earlier model. Therefore, a process is also required to manage the importation of DM models for use in the ongoing projects.

5.3.2. Development

This set of processes (Fig. 8) is responsible for building the software or gathering knowledge in the case of DM projects. These processes are divided into three phases: requirements specification (*requirements*), problem analysis and solution proposal (*design*) and development of the solution in practical terms (*implementation*).

There is no exact match between the development processes in DM projects and SE projects, as the ends are completely different. DM projects aim to gather knowledge, whereas SE projects target software construction. Even so, they do share the same phases (i.e. the same phases as any engineering project): *requirements* definition, solution *design* and solution development (*implementation*).

The requirements stage bears most resemblance, as its aim is to gather the client needs and describe these needs in practical terms for the designers and/or implementers (*assess situation and determine DM goals*).

As in software design, DM's design stage has to design the software support for data, since the data will ultimately be analyzed on the software support. However, the key SE design task, which is "*perform architectural design*", has no exact equivalent in DM. As already mentioned, the goal of SE design is to translate specifications and requirements into a preliminary design of the solution (e.g., using object-oriented design or structured modular design). Therefore, perhaps the best thing would be to equate this task to the early decision made on what DM paradigms (clustering, classification, dependency modeling, deviation detection, sequence analysis, etc.) are to be explored to achieve the *DM goals* specified in the *requirements* process. This would fit in with a later

	SE Process Model		CRISP-DM	
	Process	**Task**	**Process**	**Task**
Post-Development Processes	**Installation**	* Distribute software * Install software * Accept software in operational environment	D: Plan deployment	Plan deployment
	Operation and support	Operate the system		
		Provide technical assistance and consulting	D: Plan monitoring and maintenance	Monitoring and maintenance plan
		Maintain support request log		
	Maintenance	Identify software improvements needs		
		Reapply software life cycle		
		Implement problem reporting method		
	Retirement	Notify user	D: Produce final report	Final report
		Conduct parallel operation		
		Retire system		

D: Deployment

Fig. 9. Comparison of post-development processes.

implementation phase, where the right modeling technique for the preferred paradigm for each goal would be selected (*select modeling technique*).

There is no direct mapping between the implementation stages, as they pursue different goals. This is the best researched stage of DM, on which all the proposed "methodologies" focus. The implementation stage would be equivalent to gathering and analyzing the data available for the project, the creation of new data from what are already available, tailoring for DM algorithms and the creation of DM models. CRISP-DM covers all of these activities.

5.3.3. Post-development

Post-development processes (Fig. 9) are the processes that are enacted after the software has been built and are applicable during the later life cycle stages. The *installation* process involves the transportation and installation of a software system from the development environment to the operating environment. The *operation and support* process involves system operation by the user and user support. Support includes technical assistance, user queries and support request entry in the support request log. This process can start up the *maintenance* process that provides feedback information to the software life cycle and leads to changes in the software. Finally, the retirement process is the *retirement* of an existing system by withdrawing it from operation and support.

The knowledge gathered in DM projects should be passed on to the user and either installed as pure knowledge or integrated into the client organization's software system for use. The *operation and support* process is necessary to validate the results and how they are interpreted by the client in a real environment. The *maintenance* process is equally important for updating models or discovering which of the gathered knowledge is erroneous or invalid when new data are entered. This can lead to backtracking in the global process or life cycle in order to select new attributes or techniques not considered before.[2] As regards *retirement*, DM models also have a period of validity: if the data profiles change, the models will also change and will no longer be valid. As Fig. 9 shows, CRISP-DM neither satisfactorily nor completely covers any of the above processes, despite their importance.

5.4. Integral processes

Integral processes (Fig. 10) are necessary to successfully complete the project activities. They are enacted simultaneously to development-oriented processes and include activities that are unrelated to development. They are used to assure the completeness and quality of the project functions.

The *evaluation* processes are used to discover defects in the product or in the process used to develop the project. This process covers the performance of all the verification tasks, including verification tests, reviews and audits, and all the validation tasks, including validation tests, run throughout the life cycle to assure that all the requirements are satisfied. This process is applied to each life cycle process and product.

The *software configuration management* process identifies the structure of a system at a given time in the life cycle. This structure is termed system configuration. Its goal is to control system changes and maintain system coherence and traceability to be able to conduct audits of the evolution of configurations. On the other hand, the *documentation development* process is the set of activities that design, implement, edit, produce, distribute and maintain the documents developers and users require. Finally, the *training* process includes the development, validation and implementation of training programs for developers, technical support personnel and clients and the preparation of proper training materials.

[2] Maintenance is a key task in SE because it has a big impact on project development. It is so important that there are specialized maintenance journals and conferences.

	SE Process Model		CRISP-DM	
	Process	**Task**	**Process**	**Task**
Integral Processes	**Evaluation**	Conduct reviews	E: Review process	Review of process
			D: Review project	Experience documentation
		Create traceability matrix		
		Conduct audits		
		Develop test procedures	BU: Determine Data Mining goals	Data Mining success criteria
			M: Generate test design	Test design
		Create test data		
		Execute test	M: Assess model	Model assessment
		Report evaluation report	E: Evaluate results	Assessment of data mining results with respect to business success criteria
	Software configuration management	Develop configuration identification		
		Perform configuration control		
		Perform status accounting		
	Documentation development	Implement documentation	D: Produce final report	Final report
			D: Produce final report	Final presentation
		Produce and distribute documentation	M: Build model	Model description
			E: Evaluate results	Assessment of data mining results with respect to business success criteria
	Training	Develop training materials		
		Validate the training program		
		Implement the training program		

BU: Business Understanding M:Modeling E:Evaluation D: Deployment

Fig. 10. Comparison of integral processes.

The *documentation development process* will be almost the same for DM projects as it is for SE, but changes should be made to how the *evaluation* process is done. As mentioned early, the *configuration management* (see footnote 1) process is a key CRISP-DM omission. This process is vital, because one or more people developing a DM project generate a great many versions of input data, models and documents, etc. Unless these versions are properly organized through configuration management, it is very difficult to return to early models if necessary.

We believe that any new DM process model should account for the *training* process. Here a distinction is made between data miner training and user training. To be able to repeat the enacted process or properly interpret the project results in the light of new data, users sometimes need to be trained in DM, as well as in the knowledge gathered from data.

6. A process model for DM engineering

From the comparison of CRISP-DM with a SE process model, we found that many of the processes defined in SE that are very important for developing any type of DM engineering project are missing from CRISP-DM. This could be the reason why CRISP-DM is not as effective as it should be. What we propose here is to take CRISP-DM tasks and processes and organize them by processes as has been done in SE (see Figs. 5–10). Also we propose adding what we onsider to be key development activities. The activities missing from CRISP-DM are primarily *project management* processes, *integral* processes (that assure project function completeness and quality) and *organizational* processes (that help to achieve a more effective organization). Fig. 11 shows an overview of the proposed process model, including the key processes. The *KDD process* is the project *development* core. Figs. 5–10 illustrate the details of which DM activities and tasks are part of each process.

In the following we describe the processes shown in Fig. 11. We also explain why we think they are necessary in a DM project and describe how they can be developed.

To help with this description, the processes defined in Fig. 11 are classified in Table 3 as follows.

- Type I: the process already exists in CRISP-DM or in another KDD process and can be used as it is in a DM project.
- Type II: the process exists partially in CRISP-DM or in another KDD process, but it must be added to and adapted to cover all process tasks.
- Type III: the process exists somehow in CRISP-DM or in another KDD process, but must be redefined to fit into the new general process model.
- Type IV: the process does not exist in CRISP-DM or in any other KDD process, but could be extracted and adapted from SE.
- Type V: the process does not exist in CRISP-DM or in any other KDD process and must be defined from

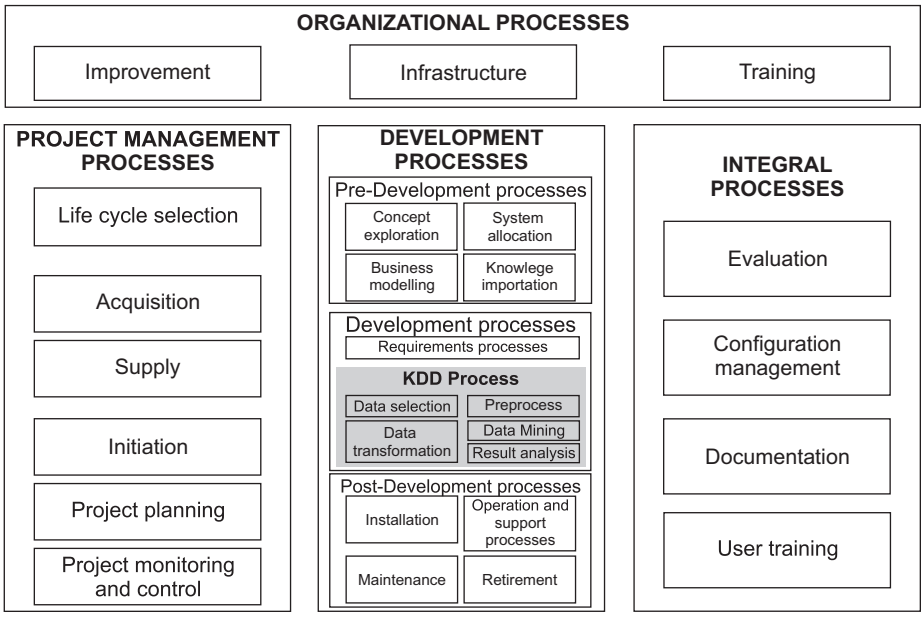

Fig. 11. Data mining engineering process model.

scratch or adapted from other engineerings different from other branches of engineering other than SE.

6.1. Organizational processes

This set of processes helps to achieve a more effective organization. They also set the organization's business goals and improve the organization's process, product and resources. Neither the IEEE Std 1074 nor the ISO 12207 SE process models include these processes. They were introduced in ISO 15504 or SPICE [47]. These processes affect the entire organization, not just one project.

This group includes the following processes (see Fig. 11):

- Improvement: This activity broadcasts the best practices, methods and tools that are available in one part of the organization to the rest of the organization.
- Infrastructure: This task builds the best environment in the organization for developing DM projects.
- Training: This activity is related to training the staff participating in current or ongoing DM projects.

No DM methodology consider any of these activities. We think that they could be adapted from the SPICE standard because they are all general-purpose tasks common to any kind of project.

6.2. Project management processes

This set of processes establishes the project structure, and coordinates and manages project resources throughout the project life cycle. We define six main processes in the project management area. Existing DM methodologies or process models (such as CRISP-DM) take into account only a small part of project management, i.e. the project plan. The project plan is confined to defining project deadlines and milestones. All projects need other management activities to control time, budget and resources, and the project management processes are concerned with controlling these matters.

6.2.1. Life cycle selection

This process defines the life cycle to be used in the DM project [2]. Until now, all DM methodologies have had a similar life cycle to CRISP-DM: waterfall life cycle with backtracking. However, there is a fair chance of new life cycles being developed to meet the needs of the different projects, as happens in SE from where they could be adapted.

6.2.2. Acquisition

The acquisition process is related to the activities and tasks of the acquirer (who outsources work). Model building is one possible example of outsourcing. In the case of outsourcing, the acquirer must define the acquisition management, starting from the proposal and ending with the acceptance of the outsourced product. Not now considered in DM processes, this process must be included because DM projects now developed at non-specialized companies are often outsourced [48].

The acquisition process could be an adaptation of the process proposed in the ISO 12207 standard. It defines software development outsourcing management from requirements to software (this depends on which part is outsourced).

6.2.3. Supply

The supply process concerns the activities and tasks that the supplier has to carry out if the company acts as

Table 3
Process type

	Type I	Type II	Type III	Type IV	Type V
Improvement				×	
Infrastructure				×	
Training				×	
Life cycle selection				×	
Acquisition				×	
Supply				×	
Initiation—Create DM life cycle	×				
Initiation—Allocate project resources	×				
Initiation—Perform estimations			×		
Initiation—Define metrics					×
Project planning—Plan evaluation				×	
Project planning—Plan configuration management				×	
Project planning—Plan system transition				×	
Project planning—Plan installation	×				
Project planning—Plan documentation				×	
Project planning—Plan training			×		
Project planning—Plan project management				×	
Project planning—Plan integration	×				
Project monitoring and control—Manage risks	×				
Project monitoring and control—Manage the project				×	
Project monitoring and control—Retain records				×	
Project monitoring and control—Identify life cycle process improvement needs	×				
Project monitoring and control—Collect and analyze metrics				×	
Pre-development processes—Concept exploration			×		
Pre-development processes—System allocation	×				
Pre-development processes—Business modeling				×	
Pre-developmentprocesses Knowledge importation—					×
Development processes—Requirements processes				×	
Development processes—KDD process	×				
Post-development processes—Installation		×			
Post-development processes— Operation and support			×		
Post-development processes—Maintenance					×
Post-development processes—Retirement				×	
Evaluation—Conduct reviews	×				
Evaluation—Create traceability elements					×
Evaluation—Develop test procedures	×				
Evaluation—Create test data				×	
Evaluation—Execute test	×				
Evaluation—Report evaluation	×				
Configuration management—Develop configuration identification				×	
Configuration management—Perform configuration control				×	
Configuration management—Perform status accounting				×	
Documentation	×				
User training				×	

the developer of an outsourcing project. This process defines the tasks the supplier has to perform to interact with the outsourcing company. It also defines the interaction management tasks.

As above, this process can be adapted from ISO Std 12207.

6.2.4. Initiation

The initiation process establishes project structure, and coordinates and manages project resources throughout the project life cycle. This process could be divided into the following activities:

- Create DM life cycle: This activity maps the generic life cycle phases into real life cycle phases. Once a life cycle is selected to meet the needs of the DM project, life cycle phases have to be created. This activity extends across the following CRISP-DM tasks: *BU—produce project plan—identify major iterations*.
- Allocate project resources: This activity allocates project resources. CRISP-DM includes this activity as *Inventory of resources* in the *produce project plan task* of the *BU* process.
- Perform estimations: This process is related to estimating the human resources, budget and effort necessary to develop the DM project. CRISP-DM considers that a budget must be proposed in terms of cost and benefits, but it makes no suggestion as to how it should be estimated nor does it state what effort should go into this. Some papers [49,50] have dealt with the estimation of DM projects through an adaptation of the COCOMO software development estimation model [51].

- Define metrics: Although some DM metrics has been defined (ROI, accuracy, space/time, usefulness, etc.) [52–55], they are not, as far as we know, being used in ongoing DM developments. Consequently, metric definition and application to the estimations are not considered as a CRISP-DM task. In our view, the task should be included, highlighting the need for further research in this area to develop more useful metrics.

6.2.5. Project planning

The project planning process covers all the tasks related to planning project management, including the contingencies plan. This process does not receive much attention in DM. DM methodologies focus primarily on technical tasks, and they overlook most of the project management activities. The set of activities considered in this process are:

- Plan evaluation: In DM projects, evaluations are carried out but they are not planned. This activity could be adapted from SE standards [56,57].
- Plan configuration management: No DM methodologies or processes consider this task at all, even though it is very important because DM projects generate a lot of different information, which requires some organization. The configuration management planning could be adapted from SE standards [56,58]. All that would be required is to change and adapt configuration items to DM instead of software.
- Plan system transition: This process concerns planning the transition from the old to the new system, which includes the knowledge acquired from the DM project. DM methodologies do not consider this kind of plan, but it can be adapted from SE standards [56,59].
- Plan installation: This activity is considered in CRISP-DM by the *deployment—plan deployment—plan deployment* task.
- Plan documentation: Although documentation is developed in DM projects, its development is not planned or organized. This planning can be adapted from SE standards [60–63].
- Plan training: CRISP-DM states that the user should be trained, but this training is not planned. This planning can be adapted from SE standards [64,65].
- Plan project management: This activity details the project organization and assigns responsibilities. It specifies standards, methodologies, and tools for configuration management, quality, evaluation, training, documentation and development. This activity apportions the project budget and staffing, and defines schedules. This activity includes planning for support, problem reporting, risk management and retirement. No DM methodologies or processes formally consider this activity, but it can be adapted from SE standards [66–68].
- Plan integration: Plan integration process designs how to embed the knowledge in an information system if the user is to use such knowledge. This activity is considered in CRISP-DM through the *Deployment—Plan monitoring and maintenance—Monitoring and maintenance* task.

6.2.6. Project monitoring and control

The project monitoring and control process covers all tasks related to project risk and project metric management. CRISP-DM considers almost all the activities in this process to a different extent and depth. The activities that are considered within this process are as follows:

- Manage risks: CRISP-DM considers this activity through the *BU—asess situation—risks and contingencies* task.
- Manage the project: This activity manages the execution of all activities in the life cycle according to the plans designed in the project planning process. The progress of the project is reviewed and measured against the established estimates and plans (e.g., estimated vs. actual cost, estimated vs. actual effort, and planned vs. actual progress). This activity is new in DM because CRISP-DM considers project planning but not plan control. This activity could be adapted from SE's IEEE Std 1074.
- Retain records: This activity is related to recording data for metrics. This is a new activity, as metrics are not used in current DM projects, although this general model proposes their use. Initially, this activity can be adapted from SE standards [69,70].
- Identify life cycle process improvement needs: This process is designed to adapt the life cycle to enterprise needs, because different companies and projects may not use the same life cycle in the same way in. Although no different life cycles have been defined yet for DM projects, it is not unreasonable to think that any life cycle could be reviewed and improved. In fact CRISP-DM considers this task similarly through *development—review project—experience documentation* (learning from development experience).
- Collect and analyze metrics: Current DM methodologies do not collect metrics, but this task can be adapted from SE standards [71,72].

6.3. Development processes

These are the most highly developed processes in DM. All DM methodologies focus on these processes. This is due to the fact that development processes are more related to technical matters. Consequently, they were developed at the same time as the techniques were created and started to be applied. These processes are divided into three groups: pre-development, development, and post-development processes.

The development process is the original KDD process defined in [73]. The pre- and post-development processes are the ones that require a greater effort.

6.3.1. Pre-development processes

These processes are related to everything that you have to do before the project kicks off. In a DM project this set of processes is divided in the following processes.

6.3.1.1. Concept exploration. The concept exploration process identifies the idea or need behind the project. This

process formulates potential approaches to the solution of the identified problem, and it conducts feasibility studies.

In CRISP-DM these activities are considered across different tasks such as *BU—determine business objectives*, *assess situation*. However, these tasks do not completely cover the activities because they focus primarily on the terminology and background of the problem to be solved. We conclude that some tasks adapted from SE standards, such as *identify ideas or needs* and *conduct feasibility studies*, need to be added to optimize project development. This process should allocate project resources.

6.3.1.2. System allocation. This activity defines the business success criteria and the business goals and makes an initial assessment of tools and techniques that are available for the project. These tasks are considered in CRISP-DM through the *BU* phase: *determine business objectives—business objectives and business success criteria* and *produce project plan—initial assessment of tools and techniques*.

6.3.1.3. Business modeling. This is a completely new activity. The CRISP-DM business model is described in the *BU* phase, but there are no business modeling procedures or formal tools and methods as there are in SE [74–77]. Some further work and research should be carried out in the DM area on business modeling and its mapping to specific DM goals and models. We believe that a great amount of the research done in SE in this direction can be adapted to DM.

6.3.1.4. Knowledge importation. This process is related to the reuse of existing knowledge or DM models from other or previous projects, something which is very common in DM.

CRISP-DM does not consider this process at all, and its SE counterpart is related to software and cannot be easily adapted. Consequently, the process must be created from scratch.

- Identify the knowledge to be imported.
- Define the method for importing the knowledge, e.g., if we had an association model, we could import the rules or the model.
- Import knowledge. Use the selected method to incorporate previous knowledge into the project.

6.3.2. Development processes

This is the most developed phase in DM methodologies, because it has been researched since late 1980s. CRISP-DM phases include all these processes in one way or another. In SE the process is divided into requirements, analysis, design and implementation phases. For DM projects we can easily map the requirements, design and implementation phases. The design and implementation phases match the KDD process, and we will stick with this process and its name.

6.3.2.1. Requirements processes. CRISP-DM does consider this set of processes but they are incomplete. The requirements are developed in the *Assess situation* (*requirements*, *assumptions and constraints* and in *determine DM goals* tasks of CRISP-DM's *BU* phase

In CRISP-DM this task produces a list of requirements, but the CRISP-DM user guide does not specify or describe any procedure or any formal notation, tool or technique to obtain the requirements from the business models. Neither does it specify or describe how to translate requirements into DM goals and models for proper use in the subsequent design and implementation phases: the KDD process. We believe that requirements can be described formally like they are in SE [78–80]. For example, something like use-case models could be adapted to specify the project requirements. Further work and research, possibly inspired by SE best practices, should be put into developing a core of formal methods and tools adapted to this task in the DM area.

6.3.2.2. KDD process. The KDD process matches the design and implementation phases of a software development project. This set of processes is responsible for acquiring the knowledge for the DM project. KDD includes the following tasks: data selection, preprocessing, data transformation, DM, result analysis, as shown in the outlined version of KDD in Fig. 12. Fig. 12 also shows how CRISP-DM covers the KDD process.

The phases of the KDD process are:

- Data selection: Data sources and data set that are needed in the project must be identified.
- Preprocessing: Once data have been identified, they must be studied to be understood and to detect integration errors or outliers. Data must be cleaned and adjusted to the DM project.
- Data transformation: After the selection of DM algorithms and their parameters, data must be translated to the algorithm input format.
- Data mining: In this phase, knowledge is gathered through the application of selected DM algorithms.
- Result analysis: Once the knowledge has been gathered, it must be analyzed to evaluate whether it is correct, valid and useful.

6.3.3. Post-development processes

Post-development processes are the processes that are carried out after the knowledge is gathered. They are applicable during the later life cycle stages.

6.3.3.1. Installation. This process is commended with transferring the knowledge extracted from the DM results to the users. The knowledge can be used as it is, i.e. to help managers to make decisions about a future marketing campaign, or it could need some software development, i.e. to improve a web-based recommender system that already exists. CRISP-DM considers planning for deploying the knowledge at the client site, but it does not regard the development of software installed and accepted in an operational environment as part of this deployment.

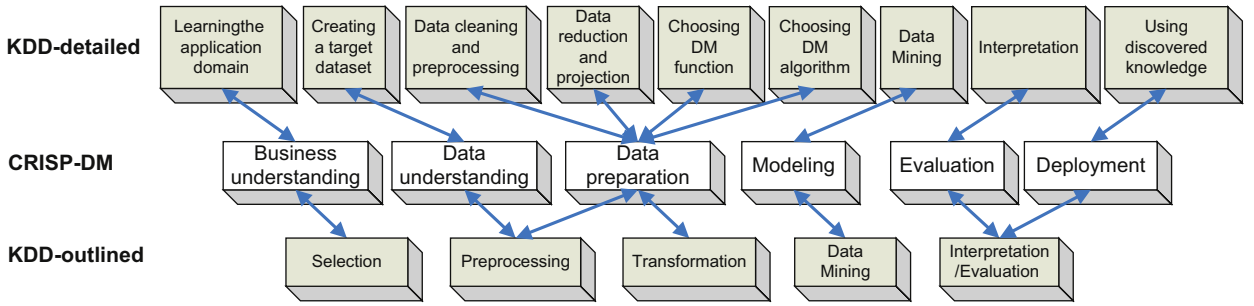

Fig. 12. Comparison of KDD (outlined and detailed) and CRISP-DM.

This task could be adapted to consider all these aspects from SE standards [58,56].

6.3.3.2. Operation and support process. This process is necessary to validate the results and how they are interpreted by the client, and, if software is developed, to provide the client with technical assistance.

CRISP-DM only includes results monitoring. In addition, we propose tasks to validate the results (this is a new task) and to provide technical assistance if necessary (this task could be directly incorporated from IEEE Std 1074).

6.3.3.3. Maintenance. The maintenance process has two different paths. On the one hand, if knowledge is embedded in software, this process will provide feedback information to the software life cycle and lead to changes in the software. For this path, the task can be adapted from the IEEE Std 1074 maintenance process. On the other hand, CRISP-DM does not include a task for knowledge used as it is, and this needs to be developed from scratch.

6.3.3.4. Retirement. The knowledge gathered from data is not valid forever, and this task is in charge of the retirement of obsolete knowledge from the system. CRISP-DM does not cover this process, but it can be adapted from IEEE Std 1074.

6.4. Integral processes

Integral processes are necessary to successfully complete the project activities. These processes assure project function completeness and quality. They are carried out together with development processes to assure the quality of development deliverables. The integral processes group the four processes described in the following.

6.4.1. Evaluation

This process is used to discover defects in the product or in the process used to develop the DM project. CRISP-DM more or less covers the evaluation process through evaluation activities spread across different phases: *evaluation*, *deployment*, *BU and modeling*. But we think the organization of the SE process is more appropriate and covers more aspects. The activities proposed in this process are:

- Conduct reviews: CRISP-DM considers this activity through *evaluation—review process—review of process*, *deployment—review project—experience documentation*. CRISP-DM defines four types of reviews: in-process, management, process improvement, and post-development reviews. Each review has a different goal and is carried out at a different time in the life cycle.
- Create traceability elements: These items are not specified in CRISP-DM and can be adapted from SE standards [81–83].
- Develop test procedures: Test procedures are developed to refine the test approach from the planned evaluation. The test procedures define what type of tests are to be conducted, what is to be tested, the data to be used in testing, the expected results, the test environment, and the procedures to be followed in testing. In DM methodologies, CRISP-DM divides this activity into different phases: *BU—Determine DM goals—DM success criteria*, *Modeling—Generate test design—Model assessment*. However, the test procedures described in CRISP-DM focus on DM goals because it does not define business goals. The test procedures for DM need to be reviewed in depth to evaluate their results in terms of business success instead of DM success, as in SE. For instance, a client clustering may pass a statistical test on accuracy, but be inappropriate for use in an application deployed in a future marketing campaign, where new data should be considered for the development of new clusters. The clustering accomplished the DM goal but not the business goal. This review, involving further work and research, is relaxed to the ones already suggested above concerning the business modeling and requirements processes.
- Create test data: This is not specified in CRISP-DM and can be adapted from SE standards [84,85].
- Execute test: This activity manages the execution of tests. CRISP-DM considers this activity: *modeling—assess model—model assessment*.
- Report evaluation report: The results evaluation report is also considered in CRISP-DM: *Evaluation—evaluate results—assessment of DM w.r.t. business success criteria*.

6.4.2. Configuration management

This process is designed to control system changes and maintain system coherence and traceability to be able to conduct audits of the evolution of configurations. We consider this to be a key process in a DM project because

of the amount of information and models generated throughout the project.

Surprisingly, DM methodologies do not account for this process at all. We consider that SE standards [86,87] could be a good starting point. This process includes the following activities.

- Develop configuration identification: This activity defines the DM configuration identification, including project baseline definition, titling, labeling, and numbering to reflect the structure of the product for tracking. This process identifies those configuration items that are to be addressed by the configuration identification. The configuration identification also defines the documentation that is required to record the functional and physical characteristics of each configuration item.
- Perform configuration control: This activity controls the configuration of models according to the configuration identification. Changes to controlled models are tracked to ensure that the configuration of the model is known at all times. All items specified in the configuration identification are subject to this change management discipline. Changes to controlled items are allowed only with the approval of the responsible authority. This can result in the establishment of a formal DM configuration control board. Controlled items are maintained in a library.
- Perform status accounting: This activity receives the configuration identification and change status and creates and updates the status reported information to reflect the status and history of controlled items. The history of changes to each controlled item is maintained. Status reported information may include such data as the number of changes to date for the project, the number of releases, and the latest version and revision identifiers.

6.4.3. Documentation

This process is related to designing, implementing, editing, producing, distributing and maintaining the documentation of the project. CRISP-DM considers this process across different phases: *Deployment—produce final report—final report*, *deployment—produce final report—final presentation*, *modeling—build model—model description*, *evaluation—evaluate results—assessment of DM results w.r.t business success criteria*.

6.4.4. User training

Current DM methodologies do not consider user training at all, but it can be adapted from SE standards [65,88]. This process is related to training inexperienced users in the use and interpretation of the results of the DM project. The activities considered in this group of processes are as follows:

- Develop training materials.
- Validate training program.
- Implement the training program.

Note that the user training plan is created in the project planning process. On the other hand, staff training is part of the organizational processes.

7. Discussion

The proposed model primarily aims to improve CRISP-DM to make it comparable to an engineering process such as is described in IEEE 1074 and ISO 12207. Section 5 "SE process model vs. CRISP-DM" compared the process models, highlighted the parallelism between SE and DM, and pointed out that the DM process standard does not bear comparison with a mature engineering process standard such as is used in other areas like SE. It indicates that the DM standard is poorly organized and has gaps, uncovered or poorly covered activities. Section 6 "A process model for DM engineering" proposes a standard or model that includes all the activities in a well-organized manner, describing the proposed process phase by phase, and discussed whether or not CRISP-DM acceptably covers each phase. If not, it was indicated whether the phase could be covered by improving CRISP-DM stipulations or whether the respective phase could be adapted from SE standards or any other source.

To what extend SE standards are appropriate to CRISP-DM? It can be argued that the nature of the DM process is different from the SE process since it seems highly iterative, and a less rigorous approach would be fine. But while this could be true for projects concentrated on the development of new and improved DM algorithms, which has been the research trend in the area up to now [27], introducing a DM application into an organization is not essentially very different from any other software application project [89]. Industry-oriented DM exhibits many characteristics similar to SE [27], and DM projects typically have a higher complexity than most other software projects [89]. The CRISP-DM model is very industry-oriented [27], and its definition was inspired and based in SE models [90–92], but it is not complete [27,44,89,91–95]. Several authors have already detected many pitfalls and reported problems in CRISP-DM due to the lack of an overall definition and integration of the entire process. They have pointed out problems in activities not defined in CRISP-DM but existing in IEEE 1074 and ISO 12207 related to project planning [89,44], project estimation [89], metrics [89,92], management and control [27,44,89,94,96,97], acquisition [98], supply [89,98], business processes [93,95,97,99,100], requirements [94,97,101], deployment [93], evaluation and quality assurance[44,90,100], installation [102], configuration management [93,96], documentation [44,94,96], and team and user training [93]. All of these activities exist in a DM industry-oriented project and must be considered and included in the definition of a complete DM process. Our model extracted from the SE standards covers these activities, confirming the assumption that the application of the SE principles to the DM process is appropriate at this level.

There are other works exploring possible links between DM and SE. As in this paper, some authors propose the use of models derived from SE processes to cover some

activities of the DM process. The new version of CRISP-DM to be developed looks for inspiration into software development, quality and project management [93]. Other authors suggest the adoption of SE life cycles [92]. In [103] the author is more precise and proposed the ISO/IEC 12207 standard as a source of ideas and a framework for a DM life cycle. And in [101] a requirements phase similar to the one in SE is introduced in the process. There are other examples of interaction between SE techniques and DM: in [104] a Services Oriented Architecture (SOA) and a Model-View-Controller pattern are used to support a DM project, and in [105–108] UML profiles are used to model DM algorithms.

The proposal indicates that there is still work to be done before a model like the one we are putting forward can be applied. For example, CRISP-DM is missing both the Evaluation Plan and Configuration Management activities, and they need to be created and defined. We claim that the contents of these SE activities can be considered as a good starting point. However, we think that it is beyond the scope of this paper to make specific proposals for these and other activities that are missing from or incomplete in CRISP-DM.

Finally, there is another argument to take into consideration regarding the fact that in SE not all tasks and recommendations described in IEEE 1074 and ISO 12207 are followed in all environments and conditions. Extreme Programming or Agile software development are good examples of this fact. What it is proposed in this paper is a general organizational framework for DM processes that should be adapted to the needs of the different projects, and presumably the "Extreme" or "Agile" DM counterparts will be derived and created. From this point of view, CRISP-DM can be viewed as an adaptation of the proposed model to certain environments with some tasks and activities incomplete or missing.

8. Conclusions

The premise of this paper is that SE's maturity would mean that its standard processes, better tailored to the large and complex projects that are now being developed in the field of DM, would account for aspects not covered in DM's current development standard: CRISP-DM. After analyzing SE standards, IEEE Std 1074 or ISO 12207, we developed a joint model that we used to compare SE and DM procedures process by process and activity by activity. This comparison highlighted that CRISP-DM fails to address many tasks related to project management, organization and quality in enough detail to be able to deal with the complexity of projects now under development, if at all. These projects tend to involve not only the study of large volumes of data but also the management and organization of large interdisciplinary human teams. As a result, we proposed a process model for DM engineering that covers such aspects, making a distinction between what is a process model and what is a methodology and life cycle.

The proposed process model is a correct and adequate organizational framework for DM project development activities, in which it is also specified which activities are already being carried out correctly (albeit organized differently) and which need to be improved or created from scratch. It includes all the activities covered in CRISP-DM, but spread across process groups according to more comprehensive and advanced standards of a better established branch of engineering with over 40 years of experience: SE. The validity or benefit of the proposed framework would not need to be demonstrated experimentally, because it follows from its validity and benefit when applied in other engineering projects, like SE projects.

The model is not complete, as this paper merely states the need for the processes and especially the activities set out in IEEE Std 1074 or ISO 12207 but missing in CRISP-DM. The adaptation and detailed specification of these processes is outside the scope of this paper.

This overview is the basis for further research. First, the processes that are missing or only partially covered by CRISP-DM need to be specified and tailored from their IEEE Std 1074 or ISO 12207 counterpart. Second, possible types of life cycle for a DM project need to be examined and specified. Some existing SE life cycles, like the waterfall, incremental or iterative life cycles, perhaps already exist in DM, but have not been identified as such; others will be exclusive to DM. Third, the process model specifies what to do, but not how to do it. This is denoted by the methodology used, meaning that the different methodologies that are being used for each process (like the methodology proposed in DM industrial engineering or CRM catalyst) would have to be examined and tailored to the model. And, finally, any methodology has a number of associated techniques and tools. Many such techniques and tools have already been developed in DM (such as Clementine or the neural networks technique), but others have not. As they are well established in SE (e.g., configuration management or business process modeling formal specification, techniques and tools), it would be worthwhile looking at how they could be adapted for DM processes.

Acknowledgments

We would like to thank Prof. Natalia Juristo for helping us to understand the IEEE Std 1074 or ISO 12207 models and building the joint model that served as a basis for comparison to CRISP-DM.

This work was conducted as part of the CYCIT-funded Project no. TIN2004-05873.

References

[1] P. Naur, B. Randell, Software engineering: report on a conference sponsored by the NATO science committee, 1969.
[2] R.S. Pressman, Software Engineering: A Practitioner's Approach, sixth ed., McGraw-Hill Science, New York, 2005.
[3] G. Piatetsky-Shaphiro, W. Frawley, Knowledge Discovery in Databases, AAAI/MIT Press, MA, 1991.
[4] ISL. Clementine User Guide, volume Version 5, ISL, Integral Solutions Limited, July 1995.
[5] T. Khabaza, C. Shearer, Data mining with clementine, February 1995.

[6] C. Shearer, User driven data mining, 1996, Unicom Data Mining Conference, London.
[7] D.S. Tkach, Information mining with the ibm intelligent miner family, February 1998, IBM Software Solutions White Paper.
[8] IBM, Application programming interface and utility reference. IBM DB2 Intelligent Miner for Data, IBM, September 1999.
[9] E. Frank, I.H. Witten, Data Mining: Practical Machine Learning Tools with Java Implementations, second ed., Morgan Kaufmann, Los Altos, CA, 2005.
[10] The Data Mining Research Group, DBMiner User Manual. Intelligent Database Systems Laboratory, Simnon Fraser University, December 1997.
[11] P. Chapman (NCR), J. Clinton (SPSS), R. Kerber (NCR), T. Khabaza (SPSS), T. Reinartz (DaimlerChrysler), C. Shearer (SPSS), R. Wirth (DaimlerChrysler). CRISP-DM 1.0 step-by-step data mining guide, Technical Report, CRISP-DM, 2000.
[12] G. Piatetsky-Shapiro, Knowledge discovery in databases: 10 years after, SIGKDD Explor. Newsl. 1 (2) (2000) 59–61.
[13] KdNuggets.Com. Data mining activity in 2007 vs 2006 ⟨http://www.kdnuggets.com/polls/2007/data_mining_2007_vs_2006.htm⟩, October 2007.
[14] L. DiLauro, What's next in monitoring technology? Data mining finds a calling in call centers, May 2000.
[15] M.P. McDonald, T. Jaffarian, L. Mok, S. Stevens, Growing its contribution: then 2006 cio agenda. Gartner group ⟨www.gartner.com⟩, 2006.
[16] B. Chatham, B.D. Temkin, K.M. Gardiner, T. Nakashima, CRM's future: humble growth through 2007, July 2002.
[17] B. Eisenfeld, E. Kolsky, T. Topolinski, 42 percent of CRM software goes unused. ⟨www.gartner.com⟩, February 2003.
[18] B. Eisenfeld, E. Kolsky, T. Topolinski, D. Hagemeyer, J. Grigg, Unused CRM software increases TCO and decreases ROI. ⟨www.gartner.com⟩, Febrero 2003.
[19] A. Zornes, The top 5 global 3000 data mining trends for 2003/04, META Group Research-Delta Summary, 2061, March 2003.
[20] H.A. Edelstein, H.C. Edelstein, Building, Using, and Managing the Data Warehouse. Data Warehousing Institute, first ed., Prentice-Hall PTR, Englewood Cliffs, NJ, 1997.
[21] M. Strand, The Business Value of Data Warehouses—Opportunities, Pitfalls and Future Directions. Ph.D. Thesis, Department of Computer Science, University of Skövde, December 2000.
[22] J.E. Gondar. Metodología Del Data Mining. No. 84-96272-21-4. Data Mining Institute, S.L., 2005.
[23] KdNuggets.Com. ⟨http://www.kdnuggets.com/polls⟩, 2002.
[24] KdNuggets.Com. ⟨http://www.kdnuggets.com/polls⟩, 2004.
[25] KdNuggets.Com. ⟨http://www.kdnuggets.com/polls/2007/data_mining_methodology.htm⟩, 2007.
[26] KdNuggets.Com. Is data mining a mature field? ⟨http://www.kdnuggets.com/polls/2007/data_mining_mature_field.htm⟩, September 2007.
[27] L.A. Kurgan, P. Musilek, A survey of knowledge discovery and data mining process models, Knowl. Eng. Rev. 21 (1) (2006) 1–24.
[28] P. Richards J. McCall, G. Walters, Factors in software quality, NTIS AD-A049-014, 015(055), November 1977.
[29] S. Tyrrell, The many dimensions of the software process. ACM Crossroads, 6.4, 2000.
[30] J. Moore, Software Engineering Standards: A User's Road Map, IEEE Computer Society, Los Alamitos, CA, 1998.
[31] U. Fayyad, G. Piatetsky-Shapiro, P. Smith, R. Uthurusamy, Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, MA, 1996.
[32] Two Crows Corporation, Introduction to Data Mining and Knowledge Discovery, second ed., Two Crows Corporation, 1998. ISBN 892095-00-0.
[33] Two Crows Corporation, Introduction to Data Mining and Knowledge Discovery, third ed., Two Crows Corporation, 1999, ISBN: 1-892095-02-5.
[34] SAS Institute Inc., From Data to Business Advantage: Data Mining, SEMMA Methodology and the SAS System (White Paper). SAS Institute Inc., 1997.
[35] SAS Institute, SEMMA data mining methodology ⟨http://www.sas.com/technologies/analytics/datamining/miner/semma.html⟩, 2005.
[36] F.J. Martínez de Pisón Ascacibar. Optimización Mediante Técnicas de Minería de Datos Del Ciclo de Recocido de Una Línea de Galvanizado. Ph.D. Thesis, Universidad de La Rioja, 2003.
[37] T. Pyzdek, The Six Sigma Handbook, second ed., McGraw-Hill, New York, 2003.
[38] M. Harry, R. Schroeder, Six Sigma, The Breakthrough Management Strategy Revolutionizing The World's Top Corporations, Currency, 1999.
[39] J. Dyché, The CRM Handbook: A Business Guide to Customer Relationship Management, first ed., Addison-Wesley Pub Co., Reading, MA, 2001.
[40] Jose Solarte. A proposed data mining methodoloy and its aplication to industrial engineering, Master's Thesis, University of Tennessee, Knoxville, 2002.
[41] Market ConsulTek ⟨http://www.marketconsulteks.com⟩, 2005.
[42] I. Jacobson, G. Booch, J. Rumbaugh, The Unified Software Development Process, Addison-Wesley Longman Inc., 1999.
[43] C. Shearer, The CRISP-DM model: The new blueprint for data mining, J. Data Warehousing 15 (4) (2000) 13–19.
[44] R. Wirth, J. Hipp, CRISP-DM: Towards a standard process model for data mining, in: Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining, Manchester, UK, 2000, pp. 29–39.
[45] IEEE, Standard for Developing Software Life Cycle Processes. IEEE Std. 1074-1997. IEEE Computer Society, Nueva York (EE.UU.), 1991.
[46] ISO, ISO/IEC Standard 12207:1995, Software Life Cycle Processes, Int. Organization for Standarization, Ginebra, Suiza, 1995.
[47] ISO, ISO/IEC Standard 15504:2004, Software Process Improvement and Capability dEtermination. International Organization for Standarization, Ginebra, Suiza, 2004.
[48] KdNuggets.Com. Outsourcing data mining ⟨http://www.kdnuggets.com/polls/2007/outsourcing_data_mining.htm⟩, February 2007.
[49] O. Marbán. Modelo Matemático Paramétrico de Estimación Para Proyectos de Data Mining (DMCoMo). Ph.D. Thesis, Facultad de Informática, Universidad Politécnica de Madrid, June 2003.
[50] C. Fernandez-Baizan O. Marban, E. Menasalvas. A cost model to estimate the effort of data mining projects (DMCoMo), Inf. Syst., 2007.
[51] B.W. Boehm, C. Abts, A.W. Brown, S. Chulani, B.K. Clark, E. Horowitz, R. Madachy, D. Reifer, B. Steece, Software Cost Estimation with COCOMO II, Prentice-Hall, Englewood Cliffs, NJ, 2000.
[52] W.C. Pai, Hierarchical analysis for discovering knowledge in large databases, Inf. Syst. Manage. 21 (2004) 81–88.
[53] P.D. Scott, E. Wilkins, Evaluating data mining procedures: Techniques for generating artificial data sets, Inf. Software Technol. 41 (1999) 579–587.
[54] K.-E. Biebler, M. Wodny, B. Jager, Data mining and metrics on data sets, in: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol-1 (CIMCA-IAWTIC'06), CIMCA '05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 638–641.
[55] M. Smith, A. Khotanzad, Quality metrics for object-based data mining applications. in: Proceedings of the International Conference on Information Technology, ITNG '07: IEEE Computer Society, Washington, DC, USA, 2007, pp. 388–392.
[56] D.J. Reifer, Software Management, seventh ed., Wiley-IEEE Computer Society Press, 2006.
[57] S. Biffl, D. Winkler, Value-based empirical research plan evaluation. In: First International Symposium on Empirical Software Engineering and Measurement, 2007, IEEE, 2007, pp. 494–494.
[58] Software Engineering NASA LaRC and Analysis Lab, Software Engineering Process Guidebook, Software Configuration Management Planning, NASA, December 1995.
[59] Joint Financial Management Improvement Program, White paper: Parallel operation of software—is it a desirable software transition technique?, 2001.
[60] J.T. Hackos, Managing your documentation projects, Wiley, New York, NY, USA, 1994.
[61] C.J. Metschke, Secrets of successful documentation projects, Intercom 43 (3) (1996).
[62] ERIC, Guideline for Software Documentation Management. ERIC, Springfield, VA 22161, 1984.
[63] N.J. Haneef, Software documentation and readability: a proposed process improvement, SIGSOFT Software Eng. Notes 23 (3) (1998) 75–77.
[64] C. McNamara, Complete guidelines to design your training plan. ⟨http://www.managementhelp.org/trng_dev/gen_plan.htm⟩, 2007.
[65] K. Kraiger, Creating, Implementing, & Managing Effective Training & Development: State-of-the-Art Lessons for Practice, first ed., Pfeiffer, 2001.

[66] B. Hughes, M. Cotterell, Software Project Management, fourth ed., McGraw-Hill Education Europe, 2005.
[67] N. Jenkins, Project Management Primer, Creative Commons (Attribution-NonCommercial- ShareAlike) 2.5 License, ⟨http://www.nickjenkins.net/prose/projectPrimer.pdf⟩, 2006.
[68] J. Westland, The Project Management Life Cycle A Complete Step-by-Step Methodology for Initiating, Planning, Executing and Closing the Project Successfully, Kogan, 2006.
[69] N.E. Fenton, S. Lawrence Pfleeger, Software Metrics: A Rigorous and Practical Approach, second ed., Course Technology, 1998.
[70] S.H. Kan, Metrics and Models in Software Quality Engineering, Addison-Wesley, Reading, MA, 1995.
[71] R.J. Offen, R. Jeffery, Establishing software measurement programs, IEEE Software 14 (1997) 45–53.
[72] M. Shepperd, Foundations of Software Measurement, Prentice-Hall, Englewood Cliffs, NJ, 1996.
[73] G. Pistesky-Shaphiro, An overview of knowledge discovery in databases: recent progress and challenges, Rouhg Sets, Fuzzy Sets and Knowledge Discovery, 1994, pp. 1–11.
[74] E.H. Eriksson, Business Modeling with UML, Business Patterns at Work, Wiley Computer Publishing, 2000.
[75] J. Gordijn, H. Akkermans, H. van Vliet. Business modelling is not process modelling. in: ER Workshops, 2000, pp. 40–51.
[76] J. Gordijn B. Van der Raadt, E. Yu. Exploring web services from a business value perspective, in: 13th IEEE International Conference on Requirements Engineering, IEEE, Paris, 2005, pp. 53–62.
[77] Y. Pigneur, A. Osterwalder, Y. Tucci, Clarifying business models: Origins, present, and future of the concept, Commun. AIS 15 (2005) 1–40.
[78] G. Kotonya, I. Sommerville, Requirements Engineering. Processes and Techniques, Wiley, USA, 1998.
[79] S. Robertson, J. Robertson, Mastering the Requirements Process, second ed., Addison-Wesley Professional, 2006.
[80] K.E. Wiegers, Software Requirements, second ed., Microsoft Press, 2003.
[81] V. Henderson, E. Bersoff, S. Siegel, Software Configuration Management, Prentice-Hall, Englewood Cliffs, NJ, 1980.
[82] W. Babich, Software Configuration Management, Addison-Wesley, Reading, MA, 1986.
[83] F. Buckley, Configuration Management: Hardware, Software and Firmware, IEEE Computer Society Press, USA, 1992.
[84] ANSI/IEEE, IEEE Standard for Software Unit Testing, ANSI/IEEE Std 1008, ANSI/IEEE, 1987.
[85] J. Laski, Testing in the program development cycle, Software Eng. J. 4 (2) (1989) 95–106.
[86] IEEE/ANSI, IEEE Guide to Software Configuration Management. IEEE/ANSI Standard 1042-1987, IEEE/ANSI, 1987.
[87] IEEE/ANSI, IEEE Standard for Software Configuration Management Plans. IEEE/ANSI Standard 828, IEEE/ANSI, 1990.
[88] Harold Stolovitch, Telling Ain't Training, first ed., ASTD, 2002.
[89] N. Lavrac, H. Motoda, T. Fawcett, R. Holte, P. Langley, P. Adriaans, Lessons learned from data mining applications and collaborative problem solving, Mach. Learn. 57 (2004) 13–34.
[90] A. Geyer-Schulz, M. Hahsler, Comparing two recommender algorithms with the help of recommendations by peers, in: O.R. Zaiane, J. Srivastava, M. Spiliopoulou, B. Masand, (Eds.), WEBKDD 2002—Mining Web Data for Discovering Usage Patterns and Profiles, 4th International Workshop, Edmonton, Canada, July 2002, Revised Papers, Lecture Notes in Computer Science, Lecture Notes in Artificial Intelligence, vol. 2703, Springer, Berlin, 2003, pp. 137–158.
[91] K. Rennolls, An intelligent framework (o-ss-e) for data mining, knowledge discovery and business intelligence, in: First International Workshop on Philosophies and Methodologies for Knowledge Discovery in 16th International Workshop on Database and Expert Systems Applications (DEXA 2005), August 2005, pp. 715–719.
[92] C. Clifton, B. Thuraisingham, Emerging standards for data mining, Comput. Stand. Interfaces 23 (3) (2001) 187–193.
[93] CRISP-DM Consortium, Crisp-2.0: Updating the methodology ⟨http://www.crisp-dm.org/new.htm⟩.
[94] Q. Yang, X. Wu, 10 challenging problems in data mining research, Int. J. Inf. Technol. Decision Making 5 (4) (2006) 597–604.
[95] H. Stoyan O. Hogl, M. Müller, The knowledge discovery assistant: making data mining available for business users, in: ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2000, pp. 106–114.
[96] K. Becker, C. Ghedini, A documentation infrastructure for the management of data mining projects, Inf. Software Technol. 47 (2) (2005) 95–111.
[97] Françoise Fogelman Soulié, Data mining in the real world. what do we need and what do we have ? in: Workshop on Data Mining for Business Applications. Held in conjunction with The 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2006.
[98] E. Bendoly, Theory and support for process frameworks of knowledge discovery and data mining from erp systems, Inf. Manage. 40 (7) (2003) 639–647.
[99] K. Rennolls, Visualization and bayesian nets to link business aims through kdd to deployment, in: 17th International Conference on Database and Expert Systems Applications (DEXA'06), IEEE Computer Society, Los Alamitos, CA, USA, 2006, pp. 603–607.
[100] E. Koutsofios, T. Dasu, J.R. Wright, Zen and the art of data mining, in: Workshop on Data Mining for Business Applications. Held in conjunction with The 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2006.
[101] V. Cannella, G. Russo, D. Peri, R. Pirrone, E. Ardizzone, Towards MKDA: a knowledge discovery assistant for researches in medicine, in: AI*IA 2007: Artificial Intelligence and Human-Oriented Computing, Lecture Notes in Computer Science, vol. 4733, Springer, Berlin, 2007, pp. 773–780.
[102] A. Feelders, H. Daniels, M. Holsheimer, Methodological and practical aspects of data mining, Inf. Manage. 37 (5) (2000) 271–281.
[103] M. Hofmann, The development of a generic data mining life cycle (DMLC), Ph.D. Thesis, Dublin Institute of Technology, 2003.
[104] M. Castellano, G. Mastronardi, F. Fiorino, G. Bellone de Grecis, F. Arcieri, V. Summo, E-Service Intelligence, in: Orchestrating the Knowledge Discovery Process, Springer, Berlin, 2007, pp. 477–496.
[105] C.C. Alves Edilberto Silva, B.A.C. Pereira, T.G. Viott, Metodologia para desenvolvimento de sistemas de suporte a decis ao crisp-dm utilizando a notaç ao uml, in: Uma Abordagem Aplicada à Gerência de Atendimento Hospitalar (outubro/2007). In: SBIC—I Simpósio Brasileiro de Inteligência Computacional, Universidade Federal de Santa Catarina, Florianópolis, October 2007.
[106] J. Zubcoff, J. Trujillo, A uml 2.0 profile to design association rule mining models in the multidimensional conceptual modeling of data warehouses, Data Knowl. Eng. 63 (1) (2007) 44–62.
[107] J. Pardillo, J. Zubcoff, J. Trujillo, Integrating clustering data mining into the multidimensional modeling of data warehouses with uml profiles, in: Data Warehousing and Knowledge Discovery, Springer, Berlin, 2007, pp. 199–208.
[108] J. Zubcoff, J. Trujillo, Conceptual modeling for classification mining in data warehouses. in: Data Warehousing and Knowledge Discovery, Lecture Notes in Computer Science, Vol. 4081, Springer, Berlin, 2006, pp. 566–575.