Journal of Computing and Information Technology - CIT 18, 2010, 4, 377–384
doi:10.2498/cit.1001920

A Framework for Collecting and Defining Requirements for Data Warehousing Projects

Nenad Jukic and John Nicholas
School of Business Administration, Loyola University Chicago, Illinois, USA

In this paper we address the issue of failure of data warehousing projects due to inadequate requirements collection and definition process. We describe a framework that can help accomplish the objective of developing a business-driven, actionable set of requirements. The framework consists of a series of steps to facilitate collection and definition of requirements in data warehousing projects.

The paper also addresses the difficult yet common issue of end-users' limited availability during the requirements collection process and proposes an approach that can serve to alleviate this problem.

Keywords: data warehouse, SDLC, requirements collection and definition

1. Introduction

Data warehousing has become a standard practice for most large companies worldwide [8]. The focus of this paper is on the data warehouse requirements collection and definition process. This process has been examined by a number of publications, e.g., [8,11,12], usually as one step in the larger data warehouse system development process. In this paper we focus on the data warehouse requirements collection and definition process itself and provide a detailed prescriptive framework.

Usual approaches to requirements gathering are classified as data-driven and requirements-driven [6,14]. The data-driven (also known as supply-driven) approach focuses on the analysis of the underlying operational data sources as a basis for establishing the scope and functionality of a future data warehouse. The method starts with analysis of the data sources with the designer selecting which portions of the data available from the corporate databases are relevant to decision makers.

The requirements-driven (also called demand-driven) method starts by determining the information requirements of the users. Potential problems of matching the requirements to the available data sources are faced *a posteriori* [14,15].

Contrasting the methods, the data-driven requirements method simplifies designing the ETL, but gives a subordinate role to user requirements; the requirements-driven approach gives priority to user requirements, but requires more effort in designing the ETL [6].

Our approach is requirements-driven and proposes a chronologically ordered set of steps for collecting and defining data warehouse requirements that we believe would serve to minimize the possibility of data warehouse project failure due to inadequate requirements definition. As in most large-scale IT projects, data warehousing projects carry a risk of failure [9], the causes of which have been addressed in studies, e.g., [4,7,12,14,16,17]. These studies identified numerous causes for failure from throughout the organization, including "poor objectives", "poorly managed expectations", "failure to understand why the warehouse exists", "incorrect assumptions", "improper planning" and "no relation between business and IT benefits". Many of the identified reasons for failure can effectively be traced to inadequate requirements definition.

2. Data Warehouse Requirements Collection and Definition Process within the SDLC

Like most information system development processes, data warehousing projects typically follow some form of a System Development Life Cycle (SDLC). SDLC is the overall process of developing information systems through a multi-step process that includes planning, analysis, design, and implementation [5]. One popular data warehouse-focused variation of the SDLC is the Data Warehousing Lifecycle [11] illustrated in Figure 1. Certain steps of the SDLC (such as product selection, project initiation, etc.) are omitted from the figure for brevity. The steps depicted are common to any data warehousing project; they include: data warehouse requirements collection and definition, data warehouse modeling and design, ETL design and development, front-end application specifications (requirements collection and definition), front-end design and development, deployment, and use/maintenance/growth.

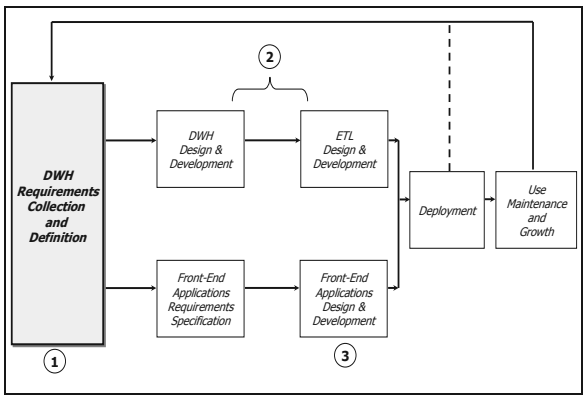

Figure 1. Abbreviated data warehouse system development lifecycle.

As Figure 1 illustrates, the actual design and development of the data warehouse, ETL infrastructure, and front-end applications all follow the collection and definition of the data warehouse requirements. The dashed line in Figure 1 represents the options of alpha (within the development team) and beta (outside the development team) test releases prior to the deployment of the actual working-system. These releases are designed to provide for testing and feedback collection that can result in modifications of the requirements and changes in the system before the actual deployment takes place.

As highlighted in Figure 1, the focus of our paper is on the data warehouse requirements collection stage (marked by number 1). We will however also discuss the effects of the data warehouse requirements collection stage on the data warehouse and ETL design and development process (number 2) and on the front-end application design and development process (number 3).

3. Steps of the Data Warehouse Requirements Collection and Definition Process

Figure 2 illustrates our framework for the chronologically ordered steps that should be taken during a typical data warehousing requirements collection and definition stage. In particular, it illustrates the data warehousing requirements collection and definition stage where the box labeled "DWH Requirements" represents its outcome. Arrows in Figure 2 represent steps that take place *within* the data warehouse requirements collection and definition phase (Figure 1, number 1). In Figure 2, the solid arrows indicate the process of *requirements collection* and the dashed arrows indicate the process of *considering the implications* of the collected requirements.

Before we describe the framework in detail, we emphasize its *iterative* nature, i.e. the four steps (arrows 1.1, 1.2, 1.3, 1.4) will repeat more than once before the actual quality set of requirements is composed.

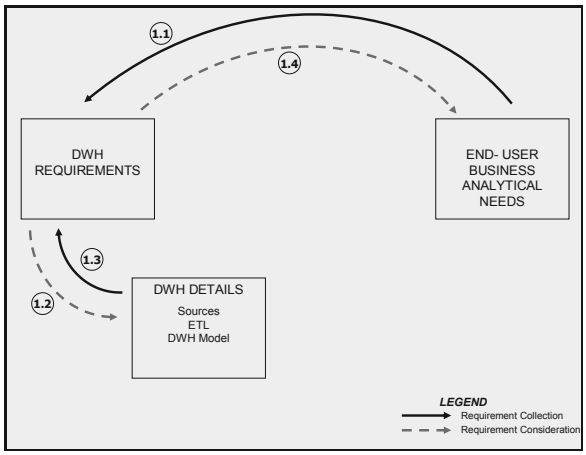

Figure 2. Data warehouse requirements collection and definition process.

As illustrated by arrow 1.1 in Figure 2, the first step in the requirements collection process is to establish (in the first iteration at a high-level and in subsequent iterations at levels of increasing detail) the end-user business analytical needs that the future data warehouse should satisfy. In other words, we should determine how the data warehouse will be used by the analysts and other users. In its first iteration, this step provides a set of initial high-level requirements containing the themes, opportunities, topics, guiding principles, and vision for the data warehouse. These needs should be prioritized, e.g., into categories such as "must have," "want," and "wish to have," to allow flexibility later on in the development process to better enable the system to meet cost or technical limitations.

The next step, indicated by the arrow 1.2, is to consider what implications the initial set of requirements will have on details of the future data warehouse. Available operational data sources are recognized, and the initial details of the data warehouse data model and the ETL process are identified. As illustrated by the Figure 3, during this process certain parts of the initial set of requirements may be eliminated due to specific reasons that emerge while considering the details of the data warehouse. For example, certain initial analytical needs stated in the initial set of requirements produced by step 1.1 may not be supported by any available operational or external sources. Figure 3 also indicates the possibility of new requirements being added to the initial set of requirements after considering the data warehouse details. For example, new possibilities for analysis (not identified during the step 1.1) may emerge while studying the operational (and other possible) sources.

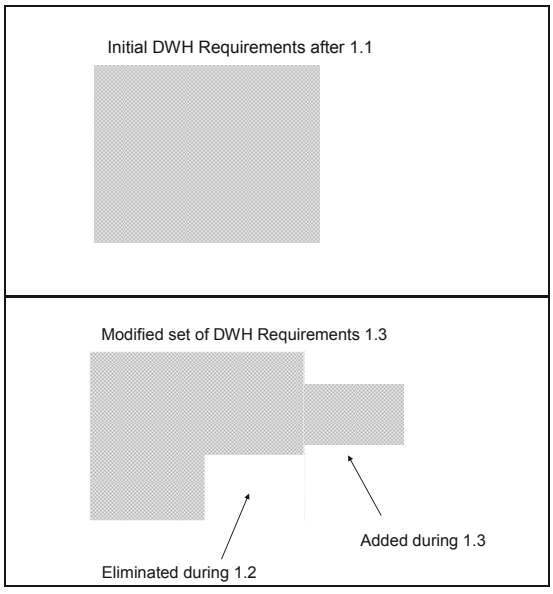

Figure 3. Modifying the requirements – example.

Step 1.3 in Figure 2 indicates the creation of the modified set of requirements from the initial set of requirement (step 1.1) after considering the data warehouse details.

Step 1.4 in Figure 2 represents a sensitive and vital step in the requirement collection phase. During this step the set of requirements that has been modified from the DHW details perspective is now considered *again* from the end-user business analytical needs perspective. This step is necessary for two critical reasons. One reason is to ensure that whatever was added during step 1.3 has analytical and/or reporting uses and adds value to future front-end applications. The other reason is that consideration of the requirements in step 1.4 will lead into another iteration of step 1.1.

In the second and subsequent iterations of step 1.1, the end-user business analytical needs will move from high-level ideas and themes to low-level details about particular uses.

As mentioned above, it is important to note the iterative nature of the process and that the cycle of steps 1.1, 1.2, 1.3, and 1.4 will repeat multiple times. Also important to note is that the final iteration of the entire process should end with step 1.4 in order to ensure that the set of requirements to be used for actual implementation is considered and approved from the end-user business analytical needs point of view. This is illustrated by Figure 4 showing the sequence of steps in the data warehouse requirement collection and definition process based on the framework presented in Figure 2.

The cycle of steps 1.1, 1.2, 1.3, and 1.4 is repeated multiple times to address high-level business needs and the needs of individual and groups of users (through iterations of 1.1 and 1.4, addressing requirements referred to as *business requirements* and *user requirements*, respectively [15]), while also acknowledging the technical requirements of the data warehouse system (through iterations of 1.2 and 1.3, addressing the technical or *system requirements*).

The process illustrated by Figure 3 describes *only* the requirements collection and definition phase (Figure 1, number 1). The *implementation* of the various portions of the actual data warehousing system (Figure 1, numbers 2 and 3) *follows* this stage. The one type of development that can occur during the requirements collection and definition phase is creation of simple demo-prototypes (e.g. using such simple tools as MS Access) with sample data for requirements verification, communication and clarification during the 1.1-1.4 cycle. Note that this is not the same process as the step indicated by the dashed line in Figure 2. The difference is that the former is a quick and simple demo-type development done during the requirements collection and definition stage, whereas the latter is a part of the actual working system creation, an examination of a version of the actual system. In addition to simple demo-prototypes, other verification, communication, and clarification methods potentially used during the requirements collection and definition cycle include creation and examination of written documents, ER models, dimensional models (star-schemas), and so on.

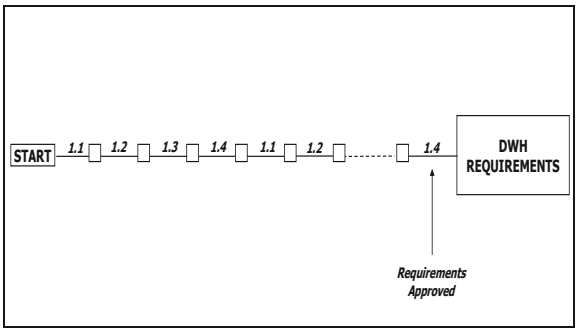

Figure 4. Example sequence of steps in a data warehouse requirements collection and definition process.

4. User Involvement

Step 1.4 and follow-up step 1.1 are often neither intuitive nor simple, and therefore, as numerous industry examples demonstrate, are easily neglected or completely omitted. The consequences of this omission are usually dire. Although the definition of high-level business requirements often begins with high-level managers and other stakeholders, essential to the success of the process is the involvement of the actual end-users of the data warehouse.

Identifying the right users and involving them in the requirements collection and definition process can be problematic. Sometimes the users can be listed by name, although often it is better to start with a list of constituencies or categories of users [4]. The list can be populated by brainstorming, but actually attaching names to the list to get the right people can be difficult. One way to get the names is to *broadcast* the project during the requirements collection and definition stage, i.e. to [4]:

Post an announcement on the company newsletter, on bulletin boards, on email, or whatever way the organization uses to broadcast information. The announcement tells what is going on, and solicits contributions.

However, including the users in the process is not necessarily easy, nor is it a panacea for removing problems from requirements collection and definition. End-user involvement can diminish too soon (e.g. after the first iteration of step 1.1), and selecting the right blend of end users can be a challenge, e.g.: [1]

The most vocal users are usually the ones to show up in the requirements-gathering workshop. This is known as the process of "squeaky wheel end-user selection". Another problem is when the most technical subset of users is selected to participate in business drivers.

Even if we are able to ensure participation of a perfect combination of users for the entire duration of the requirement collection and definition process, stated input from the end-users alone cannot be the sole determinant of the requirements. That input has to be carefully assessed and managed. The following quote from [12] illustrates this point:

Never ask the end user "What do you want in your data warehouse?" That puts them in the position of designing the system. That's your job. Besides, there is only one right answer to this question: "Everything." Instead, ask questions that help you learn what the end user does, and then translate this into what needs to go into the system. A question such as "How do you know when you have done a great job?" can get you started in the right direction.

In other words, end-user provided input has to be thoughtfully extracted and then converted into system requirements by data warehouse team members (DWH team) whose focus is on the functionality of the system. Through careful iterative interaction with the intended clients of the data warehouse using demo-prototypes, sample data, sample data models, etc, a quality set of on-target system requirements will emerge.

One of the difficult obstacles in this process is the amount of available access to the end-users by the DWH team during the data warehouse requirements collection and definition phase. Often, due to reasons such as organizational problems in the company, political and power-struggle issues, low level business-end sponsorship for the data warehousing project, logistical problems of meeting with geographically dispersed stakeholders, etc., access to critical end-users is limited and in some cases not possible at all. It is easy in those cases for DWH development team members to simply shrug their shoulders and, after getting their initial high-level (and usually vague) marching orders in the first (and, in that case, only) iteration of 1.1, perform the entire data warehouse requirements collection and definition phase by the series of DWH *team-internal* iterations of steps 1.2-1.3. That, however, is usually a recipe for failure. The result is often a competently designed data warehousing system that is technically correct, but neither fulfills the existing needs of the end-users nor clearly indicates to users its capability to create and satisfy new and useful analytical requests. Consequently, the system fails due to the lack of use.

One way for the DWH team to deal with the difficult, but not uncommon, issue of insufficient available access to the end-users during the data warehouse requirements collection and definition phase is to divide the requirements collection and definition team into two groups with separate roles. One role, which we call the *DWH analytical requirements role*, would be in charge of steps 1.1 (usually not the first iteration of it, which is often a group effort between the leaders of the DWH team and business sponsors, but the subsequent ones) and 1.4. The other role, which we call the *DWH details role*, would be in charge of steps 1.2 and 1.3.

The task of the DWH analytical requirements role is to be the gatekeeper of the requirements, to keep the *end usage* of the data warehouse in the forefront. The mandate of this role is to employ all means available to deduce, as much as possible, requirements that are most likely to result in creation of an attractive and intuitive system for future end-users. The team in charge of the DWH analytical requirements role can rely on several means to deduce user requirements that, in addition to interviews, include [3][5]:

- Review of available records.
- Review of past interviews
- Focus groups
- Questionnaires surveys to all or a sample of users
- Comments about existing data warehouses
- Observation to see what users really do, what they use, what they need.

At the end of the data warehouse requirements collection and definition phase, the team in charge of the DWH analytical requirements role should be responsible for final approval and verification to the requirements.

While this approach of dividing roles cannot guarantee the success of the data warehouse requirements collection and definition phase (and, subsequently, to the entire data warehousing project) when the company environment limits its access to end-users, it can reduce the risk of complete project failure.

On the other hand, the "shrug-the-shoulders approach" will almost certainly guarantee failure. We call such an approach *the ETL driven data warehousing project*. It occurs when, due to the lack of clearly defined requirements, the entire focus of the DWH team is driven by the structure and technical details of the operational sources and the process of their integration. In this scenario, the goal of the team is to *first* integrate all available promising sources and develop a detailed ETL infrastructure to facilitate the integration. The assumption is that once a large and comprehensive store of fully integrated data is available, its uses will become apparent.

The problem with this scenario is that it does not begin with a business context and therefore can very easily miss the target. Of course, achieving high-quality data integration and ETL infrastructure is critical and necessary for the success

of any data warehousing project, but data integration and ETL alone should not become (even initially) a self-contained and self-serving process. It must be directed at fulfilling the business needs of end user needs.

Although user requirements customarily address user functional needs, i.e., so-called *functional requirements*, they must also address *non-functional* requirements such as usability, maintainability, expandability, etc. Though these obviously directly affect the users, oftentimes users fail to state them explicitly. User requirements should also address constraints imposed on the data warehousing project (e.g., policies, time, cost) as well as on the data warehousing system (e.g., interface with procedures or other systems) [13]. Accounting for all these requirements expands the list of participants in the requirements collection and definition stage to stakeholders beyond the obvious "users."

As the requirements collection and definition process moves iteratively through steps 1.1-1.4, the requirements should be listed on a master user requirements document [2]. This list will expand in breadth and level of detail and become the definitive document to which the DWH team refers as it moves through later phases of the DWH project. It is also the document to which everyone refers at the end of the project, to assess the success of the deployed DWH system.

Before the project moves beyond the requirements collection and definition stage, senior managers responsible for approving or funding the DWH project should review and endorse the user requirements document; there is little point in proceeding with a group of requirements that satisfy the end-users, but exceed or conflict with the goals or objectives of management [2].

Although the user requirements document is substantially completed at the end of requirements collection and definition stage, some requirements will likely be modified or added after the stage has been completed.

Beyond the scope of this paper, but crucial to project success, is a formal procedure the DWH team follows to review and approve suggested changes or additions to the requirements. This procedure, called "change control" in project management, is essential to delimiting "scope creep" and keeping the project on target [13].

5. Implementation Steps Following the Data Warehouse Requirements Collection and Definition Process

The output of the data warehouse requirements collection and definition process shown in Figure 2 is a set of data warehouse requirements. These requirements are used to implement the details of a data warehouse. The process of design and implementation of the data model for the data warehouse and of the associated ETL infrastructure connecting the data warehouse with the operational and other sources *follows* the 1.1-1.4 iterative cycle of requirement collection and definition.

Once the details of the data warehouse (i.e. implemented DWH Model and ETL) are created, front-end applications that will retrieve information from the data warehouse can be implemented. Each front-end application has its own set of requirements. Collecting and defining front-end applications requirements (illustrated in Figure 1 by the box "Front-end applications requirements specification") is the process that determines the navigation style, intended look and feel, and the analytical needs that each particular application will address.

Front-end application requirements can be assembled prior to the implementation of the actual data warehouse, but the *implementation* of the actual front-end applications (such as OLAP/BI queries and reports) can occur only *after* an actual data warehouse is in place. Any front-end application can do only what the data warehouse permits. That is why it is crucial during the data warehouse requirements collection and definition phase to consider end-user analytical needs: these become the basis for developing front-end applications.

In summary, the framework in Figure 2, properly applied, ensures that the end-user business analytical needs are considered early enough in the project so that the subsequently developed data warehouse provides a fitting foundation for the development of front-end applications.

6. Conclusions

The requirements collection and definition process for traditional operational information systems is a fundamentally different process from

the requirements collection and definition process for data warehouses. The goal of a typical operational information system application is to facilitate (and sometimes improve or even change) an existing business process. Though a business process can be highly complex, it is essentially describable. Consequently, the requirements for such a system can be accurately defined. The failure of such an operational information system due to improper or inadequate requirements collection and definition can often be summarized as "the system is not doing what it is supposed to do". In contrast, the failure of a data warehousing system can often be summarized as "there was never an agreement about what the system was supposed to do in the first place, and whatever was decided was certainly not it".

The exact mission and purpose of a data warehouse is rarely (if ever) clear at the onset, which creates a challenge for the data warehouse requirement collection and definition process. In this paper we illustrated and addressed the issues and problems that hinder the data warehouse requirements collection and definition process.

We introduce a new framework for developing business driven, actionable set of DWH requirements that minimize the probability of DWH project failure. The introduced framework acknowledges and fills the clearly described gap in current practices in a logical manner based on the established best project management practices.

The framework presented in this paper (Figure 2) contains steps that are similar to those taken during most data warehouse requirements collection and definition efforts. However, our framework differs in that it prescribes how to effectively structure and chronologically use the required steps and, subsequently, obtain a quality set of requirements. The framework determines business and end-user needs and defines a set of data warehouse requirements that satisfies those needs and conforms to technical limitations of the data sources

References

[1] J. DYCHE, *e-Data, Turning Data into Information with Data Warehousing*. Addison-Wesley, 2000.

[2] R. FAULCONBRIDGE, J. RYAN, *Managing Complex Technical Projects*. Artech House, Boston, 2003.

[3] K. FORSBERG, H. MOOZ, *Visualizing Project Management*. John Wiley, New York, 1996, pp. 110–111.

[4] S. R. GARDNER, Building the Data Warehouse. *Communications of the ACM*, 4, 9, (September 1998), pp. 52–60.

[5] D. GAUSE, G. WEINBERG, *Exploring Requirements: Quality Before Design*. Dorset House, New York, 1989.

[6] P. GIORGINI, S. RIZZI, M. GARZATTI, Goal-oriented requirements analysis for data warehouse design. *Proceedings, 8th ACM International Workshop on Data Warehousing and OLAP*, (2005). Also in GRAnD: Goal-oriented approach to requirement analysis in data warehouses. *Decision Support Systems*, 45 (2008), pp. 4–21.

[7] P. GRAY, *Manager's Guide to Making Decisions about Information Systems*. Wiley, 2006.

[8] W. INMON, *Building the Data Warehouse*. 3rd Edition, Wiley, 2002.

[9] N. JUKIC, Data Modeling Strategies and Alternatives for Data Warehousing Projects. *Communications of the ACM*, 49, 4, (2006), pp. 83–88.

[10] N. JUKIC, M. VELASCO, Data Warehousing Requirements Collection and Definition – nalysis of a Failure. *International Journal of Business Intelligence Research*, forthcoming.

[11] R. KIMBALL, L. REEVES, M. ROSS, W. THORNTHWAITE, *The Data Warehouse Lifecycle Toolkit*. Wiley, 1998.

[12] J. MUNDY, W. THORNTHWAITE, R. KIMBALL, *The Microsoft Data Warehouse Toolkit*, Wiley, 2006.

[13] J. NICHOLAS, H. STEYN, *Project Management for Business, Engineering and Technology*. Butterworth-Heinemann, Amsterdam, 2008.

[14] N. PRAKASH, A. GOSAIN, Requirements driven data warehouse development. In *CAiSE Short Paper Proceedings*, (2003).

[15] J. SCHIEFER, B. LIST, R. BRUCKNER, A Holistic Approach for Managing Requirements of Data Warehouse Systems. *Proceedings, 8th Americas Conference on Information Systems*, (2002), pp. 77–87.

[16] R. STACKOWIAK, When Bad Data Warehouses Happen to Good People. *The Journal of Data Warehousing*, 2(2) (1997), pp. 33–36.

[17] B. H. WIXOM, H. J. WATSON, An Empirical Investigation of the Factors Affecting Data Warehousing Success. *MIS Quarterly*, 25, 1, (March 2001), pp. 17–24.

Received: June, 2010
Accepted: November, 2010

Contact addresses:
Nenad Jukic
Information Systems and Operations Management Department
School of Business Administration
Loyola University Chicago
1 E Pearson
Chicago, IL 60137
USA
e-mail: njukic@luc.edu

John Nicholas
Information Systems and Operations Management Department
School of Business Administration
Loyola University Chicago
1 E Pearson
Chicago, IL 60137
USA
e-mail: jnichol@luc.edu

NENAD JUKIC PhD, is a professor of information systems and the Director of Graduate Certificate Program in Data Warehousing and Business Intelligence in the School of Business Administration at Loyola University Chicago. Dr. Jukic conducts active research in various information technology related areas, including data warehousing/business intelligence, database management, e-business, IT strategy, and data mining. His work was published in a number of management information systems and computer science academic journals, conference publications, and books. In addition to his academic work, his engagements include providing expertise to a range of data management, data warehousing, and business intelligence projects for U.S. military and government agencies, as well as for corporations that vary from startups to Fortune 500 companies.

JOHN NICHOLAS PhD, is a professor of operations management in the School of Business Administration at Loyola University Chicago. His research and publication interests are in project management and lean production, about which he has authored numerous articles and several textbooks, most recently *Project Management for Business, Engineering, and Technology*, 2008 (with Herman Steyn) and *Lean Production for Competitive Advantage* (2011).