Summer 6-27-2016

Towards a Business Intelligence Systems Development Methodology: Drawing on Decision Support and Executive Information Systems

Ehsanur Rahman Safwan
Monash University, ehsan.safwan@monash.edu

Rob Meredith
Monash University, rob.meredith@monash.edu

Frada Burstein
Monash University, frada.burstein@monash.edu

TOWARDS A BUSINESS INTELLIGENCE SYSTEMS DEVELOPMENT METHODOLOGY: DRAWING ON DECISION SUPPORT AND EXECUTIVE INFORMATION SYSTEMS

Ehsanur Rahman Safwan, Faculty of Information Technology, Monash University, Caulfield East, Australia, Ehsan.Safwan@monash.edu

Rob Meredith*, Faculty of Information Technology, Monash University, Caulfield East, Australia, Rob.Meredith@monash.edu

Frada Burstein, Faculty of Information Technology, Monash University, Caulfield East, Australia, Frada.Burstein@monash.edu

Abstract

Business Intelligence (BI) systems are important IT platforms providing decision support in many enterprises, but there is a lack of independent BI-specific systems development methodologies. Further, while BI has been acknowledged as a successor to Executive Information Systems (EIS), there is little empirical evidence underpinning this view. This paper presents a case study of a BI systems development project in a large Australian healthcare organisation to argue that the development challenges faced by BI systems developers are largely similar to the challenges addressed by EIS development methodologies, and that an EIS development methodology is a useful starting point for designing a development methodology specifically for BI systems.

Keywords: Business Intelligence, Data Warehousing, Development Methodologies, Healthcare.

---

* Corresponding author.

1 INTRODUCTION

Business Intelligence (BI) systems are an integral part of the IT service platform in many enterprises. BI systems provide reporting and analytical capabilities to enterprises through integrating data collected from a range of internal and external sources, and providing BI system users with data visualisation tools to improve organisational decision-making. Negash and Gray (2008) define BI as "a system that combines data gathering, data storage, and knowledge management with analysis to evaluate complex corporate and competitive information for presentation to planners and decision makers, with the objective of improving the timeliness and the quality of the input to the decision process" (p. 176).

IT industry analyst firms such as Gartner Inc. have consistently reported BI and business analytics as a top priority for enterprise CIOs since the early 2000s (Arnott & Pervan, 2014). Gartner's (2014) most recent survey of over 2500 CIOs shows that BI and business analytics is the number one priority technology investment for 2015. This is mirrored in the *Society for Information Management's* survey of IT trends and issues (Kappelman, Johnson, McLean & Torres, 2016), where BI and analytics are reported as the largest IT investment by surveyed organisations every year since 2009. Arnott and Pervan (2014) suggest that global spending on BI platforms was approximately US$13 billion in 2012.

In addition to their industry-wide importance, BI systems projects often have a high visibility within individual organisations. As large-scale (sometimes enterprise-scale) systems, BI projects impact organisations by breaking down organisational information barriers and influencing decisions that direct the course of the enterprise. Although success is important for any IT initiative, the visibility and priority of BI projects means that success is especially important.

However, Gartner suggests suggests that 70% of corporate BI projects fail (Gartner, 2012). While there are many possible reasons for this high rate of failure, one key factor may be the inherent difference between analytics-oriented IT such as BI and other, more traditional types of information systems. Marchand and Peppard (2013) argue that this difference is a contributor to issues with big data and business analytics projects, and the arguments they put forward apply equally to business intelligence. In the past, one approach to addressing this difference is the adoption of development methodologies tailored to specific types of systems, including personal DSS (Keen, 1980; Sprague, 1980) and executive information systems (EIS) (Watson, Rainer & Koh, 1991).

Although the concept of BI has existed at least since the term was popularised by Howard Dresner in 1989 (Arnott & Pervan, 2014), there are few well-researched, peer-reviewed, independently-developed methodologies specifically for BI systems. The methodologies that do exist have largely been produced by consultants and vendors who have a commercial interest in advocating a specific development approach influenced by their particular product and service offerings. Consequently, these methods have not seen widespread adoption in projects that use different vendor technologies or consultancy services. General IS development approaches based on traditional waterfall lifecycle models (Royce, 1970) or Agile variants dominate. Given the high reported failure rate of BI projects and the success of tailored methodologies for personal DSS and EIS, there is a prima facie case for the development of an independent, academically rigorous, BI-specific systems development methodology.

The development of such a method is a significant research effort. The purpose of the project reported in this paper is to lay the groundwork for such an effort by looking to the literature on DSS and EIS development methodologies. Arnott and Pervan (2005; 2014) argue that BI is the current successor to EIS. Arnott and Pervan (2014) distinguish BI from these earlier EIS, though, arguing that BI is intended for wider adoption throughout an organisation than just the executive suite which was the focus of EIS. They highlight differences in technology as well, including web-based interfaces, dashboard-style reporting and the use of business performance measurement techniques such as Balanced Scorecard.

Arnott and Pervan's (2014) argument that BI is an evolution of EIS suggests that an EIS development methodology may be suitable for guiding BI development. However, although convincing, their argument is not based on empirical investigation and the differences between the two kinds of systems that they cite could be such that simply adopting an EIS method may not be appropriate for a BI project.

Our objective for this paper, therefore, is two-fold. The first is that we seek to add empirical evidence to the question of whether BI is a successor to EIS and other data-driven DSS (Power, 2009). The second is to establish the efficacy or otherwise of using an EIS development methodology as the basis for a BI-specific methodology. We do this by conducting a case study of a BI systems development project and using Arnott, O'Donnell & Jirachiefpattana's (1996) EIS development method, and Arnott's (2004) framework of DSS evolution as theoretical lenses. Our intent is not to prove efficacy or optimality of Arnott et al.'s (1996) methodology for BI, but rather to establish the face-validity of the idea of using this as a starting point for a broader research agenda to develop such a methodology.

We have chosen these two lenses primarily because these theories are independent and vendor neutral. The commonality of Arnott as an author suggests conceptual consistency between the two theories (and a reading of the two papers confirms this), and together they represent a coherent approach to solving the challenges of EIS development.

The paper is organised as follows. The next section provides a brief review of the literature on business intelligence systems development methodologies, the need for the evolutionary development of BI and the EIS development methodology of Arnott, et al. (1996). The subsequent section presents the case of BI development at a large Australian healthcare organisation, followed by discussion and conclusions for BI systems development practice and research.

2 EXISTING APPROACHES TO BI DEVELOPMENT

The literature on BI-specific development methodologies is limited. We conducted a literature search of several online databases including Google Scholar, ProQuest, EBSCHost, the ACM digital library, the AIS electronic library and Cengage Gale. Search terms used included "business intelligence development", as well as numerous variations including "business intelligence systems development" and "business intelligence systems development methodology", including replacing business intelligence with "BI".

The results included several independent, BI-specific development methodologies described below. Despite their focus on BI, each of these methods has several drawbacks compared to Arnott et al's (1996) EIS methodology. We address each of these, followed by a discussion on evolution in DSS and the Arnott et al EIS method.

2.1 Life Cycle Development of BI System (Gangadharan & Swami, 2004)

The Life Cycle Development of BI Systems (Gangadharan & Swami, 2004) is based on several phases arranged in a cyclical development process (see Figure 1). Despite its cyclical nature, the model suggests a unidirectional flow, and so the evolution that may take place between stages.

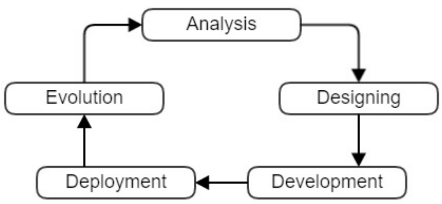

Figure 1. Life Cycle Development of BI System (Gangadharan & Swami, 2004)

Although the model explicitly addresses the concept of evolution, a key principle from DSS development, Gangadharan and Swami's discussion of the concept suggests that they actually intend iteration rather than evolution. That is, development of further applications rather than adaptation of BI applications developed during the project. Adaptation (Keen, 1980) depends on a close user-developer relationship, and this methodology doesn't explicitly address this requirement. The lack of expected user input into the development process in this methodology, and the weak representation of actions required to follow up related changes does not effectively capture the evolutionary nature of BI that is required for successfully managing BI development.

2.2 BI SDLC (Bara, Botha, Diaconita, Lungu, Velicanu & Velicanu, 2009)

Bara et al. (2009) propose a lifecycle model that is similar to that of Gangadharan and Swami (2004), but more closely tied to the traditional waterfall lifecycle model. The development process is divided into 6 stages forming a cycle: assessing the needs of the business, planning, business analysis, design, construction, and implementation. In addition to user involvement in the planning stage, it is also required in the development stage to capture the changes in the BI system and the evolutionary development it leads to. Again, while cyclical in nature, as with Gangadharan and Swami, the cycles are iterative rather than evolutionary in nature.

2.3 Simplified BI Development Methodology (Olszak & Ziemba, 2007)

Olszak and Ziemba (2007) propose a simplified BI development method involving two primary stages: BI creation and use. The BI creation stage consists of defining, designing, developing and implementing the BI system. The lifecycle iterates between creation and use, and while accommodating evolution as well as iteration, does not describe the management of the evolution in requirements in detail.

2.4 Lessons from the Past: DSS Evolution and EIS Development

2.4.1 DSS Evolution

One of the unique characteristics of decision support tools is the need for adaptation during the development process (Keen, 1980). Although writing about personal DSS, Keen's argument applies to any technology intended to support semi- or un-structured decision tasks. While the methodologies described above incorporate some aspects of evolution, they do so in a limited fashion.

Arnott (2004) provides a detailed framework for understanding the nature of DSS evolution. He argues that DSS evolution can occur on two levels: within an application as new features and functionality evolve, and between applications as new tasks require a major development effort. This is also reflected in the EIS method of Arnott et al (1996), described below. Arnott's framework describes various evolutionary "tempos", and categorises evolutionary causal factors as cognitive (changes in understanding about the decision task) and environmental (changes in the development environment). The framework is shown below in Figure 2.

	Within Application	Between Application
Cognitive Causal Factor	Continuous Evolution	Punctuated Equilibrium
Environmental Causal Factor	Punctuated Equilibrium	Punctuated Equillibrium Quantum Evolution

Figure 2. DSS Evolution Framework (Arnott, 2004)

The causal factors that Arnott (2004) suggests trigger DSS evolution are shown below in Table 1.

Cognitive Causal Factors	Environmental Causal Factors
System use	Technology change
Analyst interaction	Personnel change
Peer interaction	Internal organizational change
Consultant interaction	Merger and Acquisition
Training course	Industry changes
'Idle' thought	Coevolution

Table 1. Causal Factors of DSS Evolution (Arnott, 2004)

2.4.2 The Monash EIS Development Method (Arnott, et al, 1996)

Arnott et al.'s (1996) EIS development methodology draws on prior DSS research to inform its design. The methodology incorporates both iteration and evolution, and addresses the larger-scale projects typical of EIS and BI projects, compared to the earlier personal DSS addressed by Keen (1980) and Sprague (1980). The lifecycle model provides a detailed description of three high-level phases, each broken down into several tasks (see Figure 3 below). Tasks can be cycled within each phase, while the dual-level evolution described later by Arnott (2004) is depicted as the high-level cycling of phases (between-application evolution) and between system delivery and use (within-application evolution). Arnott et al (1996) and Arnott (2004) are therefore complementary. Together, the two frameworks provide a detailed understanding for managing the evolution of large-scale decision support applications.

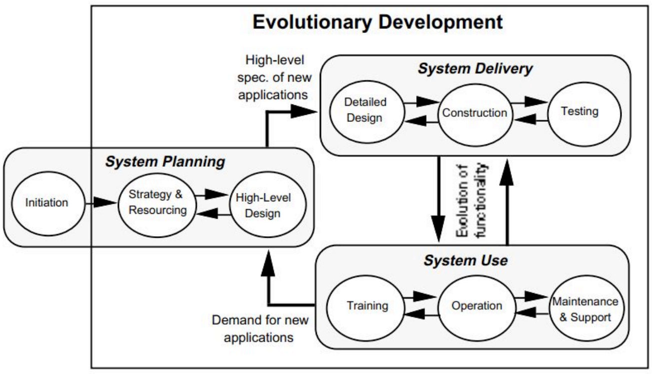

Figure 3. The EIS Development Lifecycle Model of Arnott et al. (1996).

2.4.3 Summary

The purpose of this section of the paper has to been to conduct a literature review of current BI development methodologies, and to present the two theoretical lenses that will be used to inform the design and analysis of the case study presented in section 4. The literature review shows that there are few BI-specific, vendor- or consultant-independent methodologies available, and of those reviewed, the critical aspect of evolution is not well-addressed. The two theoretical lenses that we intend to adopt, however, provide a rich understanding of the nature of evolutionary development in the context of DSS (Arnott, 2004) and a detailed lifecycle model for the development of EIS (Arnott et al., 1996).

3 RESEARCH DESIGN

As stated in section 1, the objective of this paper is two-fold: the first is to provide empirical support (or otherwise) to the conjecture that BI is the successor technology to EIS; the second is to establish the face-validity of the conjecture that the EIS development methodology of Arnott et al. (1996) is a useful starting point for the development of a BI-specific development methodology.

To achieve these objectives, we have adopted an exploratory, interpretivist case study approach (Myers, 1997). Since we are not testing theory, but laying the groundwork for theory development, an interpretivist case study allowed us to collect rich, qualitative information to inform both this study, as well as provide insights to further work in producing a BI development methodology.

The research design involves observation of a BI development project and applying Arnott et al's (1996) EIS development methodology, and Arnott's (2004) evolutionary framework, as lenses to describe the case. While Arnott (2004) is a descriptive theory, Arnott et al's (1996) methodology is a prescriptive theory. To allow us to use the prescriptive theory in a descriptive manner, we have distilled three key theoretical constructs from the methodology, with the final construct also being informed by Arnott's (2004) evolutionary framework. These are:

- a staged approach to development;
- iteration between development stages;
- dual-level evolution of the system: within-application and between-application

Our argument for the paper is that in being able to successfully apply these constructs as lenses to the observed project, we are therefore able to add empirical weight to the argument that the key issues that are addressed by EIS development methods in an EIS context are the same issues prevalent in a BI context, thereby addressing the two research objectives.

The unit of analysis for the case study is a single BI development project undertaken at a large Australian teaching hospital. Selection of the case was opportunistic and as a result of a pre-existing professional relationship between the project sponsor and one of the authors. A side-benefit of this case is that there is little extant literature on the development of BI systems in a healthcare context.

Six people from the hospital BI development team accepted our invitation to participate in the project. These included a department head, an information architect, a business solution designer, a database administrator/app developer and the acting director of knowledge management (AKM). Most had tertiary qualifications in IT and had from one to five years professional experience in the IT industry. An interview protocol was developed to guide semi-structured, one-on-one interviews with each participant[1], making use of Canter, Brown and Groat's (1985) visual card sorting technique to assist with concept explanation.

Analysis of the interview transcripts used deductive thematic analysis (Daly, Kellehear, & Gliksman, 1997). Analytical themes were derived from the three theoretical constructs derived above and applied to the transcript text as per Fereday and Muir-Cochrane (2006).

4 BI DEVELOPMENT IN A HEALTHCARE ORGANISATION

The BI development project took place over the period 2011 to 2015 in a large Australian teaching hospital. Two partner healthcare institutions were also involved. All three organisations have numerous departments with data located throughout in several thousand databases. There were no established data governance strategies in place to ensure the safe-guarding of critical institutional and medical data. One of the original aims of the project, therefore, was to secure this critical data against loss and corruption.

[1] The protocol can be downloaded from http://dsslab.infotech.monash.edu/downloads/Healthcare_BI_Interview_Protocol.pdf

A significant challenge for the organisations developing a BI platform was the complexity presented by the very high number of databases holding the source data.

The BI system was developed to support several decision domains: hospital management, clinical auditing, clinical decision support, and clinical research. The project timeline and key stages are presented below in system underwent a number of evolutionary cycles, and its changing structure is portrayed through the timeline presented in Figure 4 and Table 2, below.

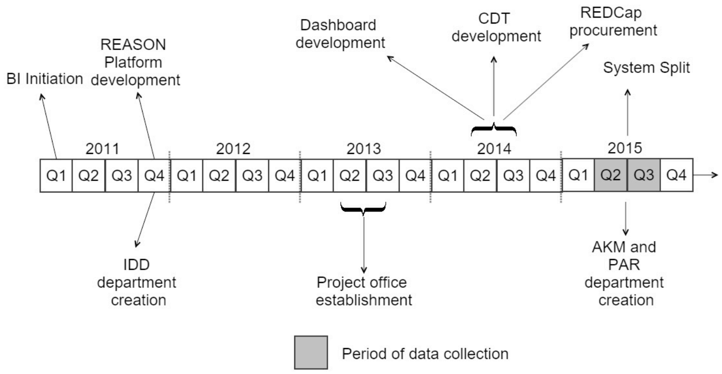

Figure 4. *BI Project Timeline*

Stage	Brief Description
1. BI Initiation	The first stage when the development of the BI system was first undertaken by the development team under the finance department
2. REASON Platform development	This stage started when the Research Analytics and Operations (REASON) platform project was undertaken for development by the Information Development Division (IDD)
3. Project Office establishment	The start of the collaborations between the development team and the project office. The project office had a continuous influence on change due to the different projects being initiated at the organisation.
4. Dashboard development	The fourth stage of the development was when interactive dashboard capability was developed for hospital executive use.
5. Cohort Discovery Tool (CDT) development	The CDT is a knowledge discovery tool added to the BI system portfolio for identifying cohorts among hospital patients.
6. REDCap procurement	Research Electronic Data Capture (REDCap) application was added as a backend data collection tool for the BI system. REDCap is a software solution designed for rapid development of data collection tools for use in clinical and research data collection (Harris et al., 2009).
7. System Split	This was a major split of the system, with one split focused on healthcare management and the other on clinical decision support (including research and clinical auditing).

Table 2. *Identified Stages of BI development at the healthcare organisation*

In the following sections, we apply the major phases of Arnott et al.'s (1996) methodology to demonstrate a fit with the case in sections 4.1 through to 4.3. We then turn to Arnott (2004) to discuss system evolution in section 4.4.

4.1 System Planning

The initial planning of the BI system involved selecting an executive sponsor and development team, setting the objectives of the system, and determining user requirements. The executive sponsor during stage one of the project was the chief financial officer, with the system initially addressing financial reporting. The organisational drive to integrate hospital data to a single platform in stage two of development increased the scope of the system, changing its objectives, and an executive sponsor from the information management department was given responsibility.

The changing of the executive sponsor was seen by the development team as bringing new opportunities to the BI system. One of the participants recalled, "*…he was a clinical, he was a doctor and he knows more about the clinical platform, and he was from U.S. so I think he involved very well in the planning and designing, like in a direction where…*". Having an executive sponsor with domain knowledge led to improved focus for the BI system.

The development team selection was also a part of the planning process, and this activity was initiated at stage one, stage two, and stage seven of the development due to the changes occuring in the team. The core of the team was made up of the operating sponsor, and several of analysts and developers who took charge of the development. The operating sponsor was found to be the most important member of the development team, as the operating sponsor led the development including planning, procuring tools, and managing resources. The operating sponsor had a strong grasp of the development, which was aided by the balancing of both clinical and technical skills.

One of the analysts mentioned, "*whole actions and design everything was from Mr. X* (operating sponsor)*, I will say it is all Mr. X's baby, because he planned, designed, and purchased everything.*" This showed that the development was heavily driven by the operating sponsor. Having a great deal of dependency on one individual may create instability in the development environment if the operating sponsor was to change. This was something witnessed in stage seven of the development, as the future direction was temporarily unclear and required a strong guidance from the executive sponsor to determine the scope of the system, and the actions required for the development. To tackle this, the development team felt the need for better documentation of past efforts to track the development.

Most of the development activities were performed in-house except for some of the construction activities that required external contractors. External contractors were required as the development team had other work responsibilities and lacked the required technical skills. The insourcing of these skills occurred in stage four of the project when a dashboard tool was developed.

Identification of data sources was prioritised based on the complexity of the data source, relevance to hospital strategies, and their financial importance. The operating sponsor mentioned, "*…the simple stuff gets done first, and the financially important stuff gets done first.*" Some of the more complex data sources, such as the clinical information system were tackled much cautiously as it was being hosted on another state, and outside stakeholders such as the vendor was also involved. The operating sponsor had remarked, "*Probably the hardest source to get acquisition to, was the clinical information system, because it is hosted in another state and it is big. We had to push quite hard to get the data from that system, and it took probably a year of pushing and negotiation but we eventually got access to it.*"

The development team faced difficulties in getting access to the different data sources as they were individually managed and had their own data managers. Identifying data sources was complex due to the large number of systems, and integrating these disparate sources was a challenge. Further, the data managers were initially protective of the data and considered the efforts of the development team to be a threat to their job security, creating information 'silos.' To address this, the source system data managers were given access to the data in the BI platform and were included in planning and testing.

Requirements elicitation involved the active participation of system users at different stages. The majority of the techniques in Watson and Frolick's (1993) list of EIS requirements gathering techniques were observed. Some techniques were more effective for gathering requirements for hospital management use, while others were more suitable for clinical decision-making and research.

A challenge for requirements elicitation in a BI context that was observed was the complexity of eliciting requirements from disparate groups of users at a variety of organisational levels. Users of the BI system include data managers, medical researchers, clinicians, nurse managers, nurses and other hospital staff, as well as senior executive. EIS, by contrast, are intended to focus just on the senior executive group meaning a smaller number of users from which requirements need to be gathered.

Several user groups of the BI system were involved in the information gathering process for some stages and others were not. As an example, requirements for the dashboard development (stage four) were gathered from senior executives as they were the prospective users of the dashboard interface. On the other hand, the cohort discovery tool (CDT) development involved clinicians, data managers and researchers. Clinicians were also heavily involved in stage seven with the development of a clinical decision support application. Multiple techniques were used including seminars, surveys and planning meetings.

Requirements gathering techniques for the initial stage of development differed from the ongoing stages. In the initial development stage, identification of critical success factors, examination of other organisations' BI systems and prototyping were used. In later stages, discussions with support personnel, attendance at meetings, examination of the strategic plan, volunteered information and formal change requests were used. Throughout the project, though, other techniques such as discussion with executives, examination of existing information systems, participation in strategic planning sessions, examination of non-computer generated information, and BI system demonstrations were used. There was some ambiguity over whether ongoing observation of usage patterns was carried out, with one development team member suggesting that this occurred, while other team members thought that it did not.

Requirements gathering techniques also differed for hospital management applications and project-based changes, versus research and clinical applications. For example, volunteered information and prototyping were more effective for research and clinical use, while more formal techniques such as meetings were more effective for hospital management.

Through system planning two main deliverables were produced. The first was a draft specification for the design and construction of the system. The second was an overall plan of development. The plan was loosely, rather than strictly, followed as most of the development was driven by the direction of the operating sponsor's vision. The specifications also changed to accommodate the practicality of development. This aligns with comments from Arnott et al (1996) who suggest the EIS planning phase should be flexible and in some cases informal.

Arnott et al. (1996) state that the draft specification of the requirements be "signed off" by relevant stakeholders. This was observed in the case where the specification was reviewed by domain experts. Data managers played an important role in verifying the draft specifications for stage two and stage five of the development, as they had expert knowledge on the data of the different departments. Also, reviewing the specifications with clinicians had helped to make certain that the draft specification was achievable for development.

4.2 System Delivery

Prior to the system being constructed, the project went through a design stage, with data models and user interfaces being developed. Construction was followed by testing of each component before further development progressed. These three primary activities, detailed designing, construction, and testing are also incorporated in the system delivery module of Arnott et al. (1996). The delivery phase activities occurred from stage two to stage six of the development, and were triggered by either new data source acquisitions, draft specifications of new applications, or changes required through system use.

The first activity in the sytem delivery phase of Arnott et al. (1996) is detailed design. In the case, detailed design The BI development design for the dashboard was also user driven as the operating sponsor recalled, "*So some people just want summary lines for certain things, and others want summary lines and detailed data behind it that you can drill in, cross references and things.*". The team designed the dashboards depending on the individual needs and styles of the executive users.

The data structure of the system did not follow the popular multidimensional data warehouse approach (Kimball & Ross, 2002). Most of the databases supporting the BI system were exact copies of the source databases stored in an operational data store. Data was transformed to the desired format in a staging area before generating the reports for presenting in dashboards and interfaces. This approach is known as 'extract, load and transform' (ELT), as opposed to the more common 'extract, transform and load' (ETL) approach, where transformation to the data is done after loading to a data store (Cohen, Dolan, Dunlap, Hellerstein, & Welton, 2009). This approach is more common in 'big data' environments (Priebe & Markus, 2015).

Having such an approach can bring flexibility to the data presentation. The operating sponsor said, "*You want some more flexible approaches to cleaning and manipulating data and pushing it out in different ways depending on the topic or the need. So, that's probably the approach that should be used going forwards.*" Such flexibility could be useful in clinical research as clinicians and researchers might use the same data in different formats. An example shared by the operating sponsor was that of 'age' where one research project may require it to be in five-year brackets, while another project may require a three-year bracket and so on.

In terms of construction, the development team did not follow any specific software development methodology. One team member stated, *"It certainly wasn't a classic waterfall model, or a waterfall with iteration model. It was much more agile and iterative than that. But always with the sense of where we are trying to go to is probably the best way to explain it."* One of the development team members noted that through using a more systematic approach, the development process could have gone smoother than it did. Prototyping was a technique used in most stages of the development that involved new tools or creating new applications. While not all of the development tasks required prototyping – data structure construction in stage two of the project primarily involved integrating data sources, for example – prototyping was used during stages three to six. This aligns with Arnott et al.'s (1996) recommendation to use prototyping during system delivery.

Testing was performed in collaboration with the source system data managers. One of the developers noted, "*That's right, and they benefit too, so none of us are clinically trained, that support this platform. So what we do is that we use their expertise clinically, 'cause they can objectively look at the data and say "hey, that's not right, it shouldn't be like that, you've named it wrong, or the data doesn't make sense, or the key is not right". So we use that information to improve the quality of the platform as well.*"

Testing helped identify errors in data capture, and made the data in the BI system more accurate and reliable. Testing was effectively performed when external stakeholders and domain experts were involved for validation. As medical data are critical and sensitive in nature, the main focus of testing was directed towards validating the data and ensuring its reliability. Testing other components of the system were performed simultaneously with the construction activities.

4.3 System Use

The system use phase of Arnott et al.'s (1996) EIS development lifecycle model consists of three activities: training, operation and maintenance and support. In the case study, training was focused on non-executive users. The operating sponsor stated, "*Tend to not do too much training to executives, 'cause they don't want to be trained. They just want the other people to get the information. They just want the obvious report.*" Training was preceded by user engagement activities. These activities were important because of the large number of users from different user groups and it was necessary to inform them about the capabilities of the system prior to providing any training.

Consistent with Arnott et al. (1996), as well as the broader DSS literature (for example Arnott, 2004 and Keen, 1980), system use in the case led to demand for change by system users. A particularly interesting observation that was noted in the case was that the development team was unable to respond to every request for change. The large number of users and heterogeneous requirements meant that change requests had to be triaged, with different classes of user receive different levels of support. One developmetn team member said, "*It depends on who requested the change. If it was an executive or a director, then yes. They would have the power. If it is a nurse, then forget it. It might help, but it is not necessary."* In an EIS environment with a smaller group of organisationally powerful users, it is unlikely that this attitude would prevail.

Evolution in requirements was driven by enthusiastic take up of the system encouraged by a supportive development team. One team member observed, *"Absolutely, once they realised what was available, it was like oh look! Something shiny. So they would say "right, so how do we get access to that" "How do I use this" "What does this data mean". So you constantly fill the enquiries about the information in the platform and helping them to be able to integrate it with their systems."*

To tackle the issue of an overwhelming number of requests resulting from a large number of users, the development team chose to address 'themes' of requests instead of individual ones. One of the reasons for not fulfilling all the requests was that the development team did not want the system to deviate from the original design objectives. Change requests were assessed against these objectives to ensure that the functionality of the system remained within the original scope. Despite this check on system evolution, as will be discussed below in section 4.4, the case exhibited evolutionary patterns consistent with both Arnott et al. (1996) and Arnott (2004).

4.4 System Evolution

Arnott's (2004) framework for DSS evolution describes two categories of causal factors for system evolution (cognitive and environmental factors), and contrasts these with two types of evolution (within- and between-application evolution). While the case exhibited both kinds of evolution over the duration of the project, there was no direct observation of cognitive triggers for between-application evolution (cognitive factors were noted, however, for within-application evolution).

Table 3 summarises the evolution observed in the case using Arnott's framework:

	Within Application		**Between Application**	
	Evolution	*Cause*	*Evolution*	*Cause*
Cognitive Causal Factor	Data Architecture	Idle Thought	None observed.	
	Number of Screens	System Use Training Course Analyst Interaction		
	Dashboard	System Use Analyst Interaction		
	CDT	Peer Interaction Idle Thought		
Environmental Causal Factor	Data Architecture	Technology Change Internal Organisational Change	REASON Platform	Industry Changes Internal Organisational Change
	Data Source	Merger and Acquisition Internal Organisational Change	REDCap	Technology Changes Industry Changes
	Dashboard	Technology Change	System Split	Internal Organisational Change

Table 3. Evolution in the BI system

The within-application evolution during the project was caused by both cognitive and environmental factors. In addition to the users demanding changes resulting from system use, the development team also played an active role in contributing to the system evolution through idle thought and peer interaction. Some of the evolution was affected by both cognitive and environmental triggers such as the data architecture and the acquiring of a dashboard tool. Within-application evolution was observed as complex and driven by multiple causal factors.

In contrast, between-application evolution in the case was primarily caused by environmental factors. Global, industry-level trends towards the use of technology to support clinical decision making (Van De Graaff & Cameron, 2013) created indirect pressure that led to the expansion of the system scope to support this application area in stage seven. Further, technology changes brought new opportunities such as REDCap that increased the scope of the BI system as it was being used for developing tools for clinical auditing.

One interesting finding was that support for hospital management and clinical decision support could not be catered for with a single BI system, leading to the stage seven evolution of two separate systems. This showed that supporting both decision types through the same system was complex, and could not be handled by a single development team. The system split was also driven by internal organisational change as the development team was split, and a new department was created to handle decision support for hospital management, while the existing development team focused on clinical decision support. The complexity of both of these decision types was also apparent during requirements gathering, where both required different techniques.

5 DISCUSSION AND CONCLUSION

The purpose of this paper was two-fold: to add empirical weight to the argument that BI systems development projects face similar challenges to those faced by EIS projects, and that in developing an independent, theoretically informed BI development methodology, the EIS and DSS literature offers a sound starting point. To that end, we have presented a case study of a BI development project in a large health organisation, and applied Arnott et al's (1996) EIS development methodology and Arnott's (2004) DSS evolution framework as analytical lenses.

The BI development in this case study closely followed the identified theoretical constructs in both lenses, and demonstrated similarities between EIS and BI development processes. The development was staged, following the phases described in Arnott et al (1996)'s methodology. Further, the development was iterative and evolutionary. The evolutionary changes observed were triggered by both cognitive and environmental causal factors as described by Arnott's (Arnott, 2004) framework. Table 4 summarises these findings.

Theoretical Lens	**Theoretical construct**	**Case Study Observations**
Arnott et al.'s (1996) EIS Development Methodology	A staged approach to development	Each phase and the activities noted in the EIS methodology were also observed in the case.
	Iteration between development stages	The flow of stages and tasks in the case align with those in the EIS methodology. Repetition of stages was observed as scope and functionality expanded.
Arnott's (2004) Framework for DSS Evolution	Dual-level evolution of the system: within-application and between-application	Both levels of evolution were observed, although Arnott's cognitive causal factors were not observed in this case for between-application evolution.

Table 4. Summary of case study findings

One difference observed between the case and Arnott et al. (1996) is reflective of a difference between EIS and contemporary BI: that is, the larger and less homogenous user-base common for BI systems, and the impact this has on requirements gathering and managing evolutionary pressure. The development team needed to utilise different requirements elicitation techniques for different user groups, and triaged (and in some cases ignored) requests for system adaptation. The other main discrepancy between the theoretical lenses used and the case study observations was the lack of identified cognitive causal factors for between-application evolution.

Despite these differences between observation and theory, we believe that the case study findings support our conclusion that there is a prima-facie case for using the EIS and DSS literature as starting points for the development of a BI-specific systems development methodology. While the lack of observed cognitive factors leading to between-application evolution may be a consequence of the small number and homogeneous nature of the study participants (no users were interviewed, for example), the challenges posed by a wide and heterogeneous user-base for BI systems is something that such a methodology would need to address. We also believe that this case lends empirical support to the conjecture of Arnott and Pervan (2005; 2014) that BI is the modern successor to EIS.

References

Arnott, D. (2004). Decision Support Systems Evolution: Framework, Case Study and Research Agenda. *European Journal of Information Systems, 13*(4), 247-259.

Arnott, D., O'Donnell, P. A., & Jirachiefpattana, W. (1996). *EIS Development Guidelines*. Retrieved from Victoria, Australia: Monash University

Arnott, D., & Pervan, G. (2014). A Critical Analysis of Decision Support Systems Research Revisited: The Rise of Design Science. *Journal of Information Technology, 29*(4), 269-293.

Bara, A., Botha, I., Diaconita, V., Lungu, I., Velicanu, A., & Velicanu, M. (2009). A Model for Business Intelligence Systems' Development. *Informatica Economica, 13*(4), 99-108.

Boyatzis, R. (1998). *Transforming Qualitative Information: Thematic Analysis and Code Development*. Thousand Oaks, CA: Sage.

Brooks, P., El-Gayar, O., & Sarnikar, S. (2015). A Framework for Developing a Domain Specific Business Intelligence Maturity Model: Application to Healthcare. *International Journal of Information Management, 35*, 337-345.

Canter, D., Brown, J., & Groat, L. (1985). A Multiple Sorting Procedure for Studying Conceptual Systems. In M. Brenner, J. Brown, & D. Canter (Eds.), *The Research Interview: Uses and Approaches* (pp. 79-114). London: Academic Press.

Cohen, J., Dolan, B., Dunlap, M., Hellerstein, J. M., & Welton, C. (2009). *MAD Skills: New Analysis Practices for Big Data.* Paper presented at the VLDB Endowment, Lyon, France.

Crabtree, B. F., & Miller, W. L. (1999). *Doing Qualitative Research* (2nd ed.). Thousand Oaks, CA: Sage Publications.

Daly, J., Kellehear, A., & Gliksman, M. (1997). *The Public Health Researcher: A Methodological Approach*. Melbourne, Australia: Oxford University Press.

Fereday, J., & Muir-Cochrane, E. (2006). Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development. *International Journal of Qualitative Methods, 5*(1), 80-92.

Gangadharan, G. R., & Swami, S. N. (2004). *Business Intelligence Systems: Design and Implementation Strategies.* Paper presented at the 26th International Conference on Information Technology Interfaces.

Gartner (2012). *Predicts 2012: Business intelligence still subject to non-technical challenges report.* Gartner Predicts Special Report, Gartner Inc.

Gartner (2014). *Flipping to Digital Leadership: Insights from the 2015 Gartner CIO Agenda Report*. Gartner Inc.

Harris, P. A., Taylor, R., Thielke, R., Payne, J., Gonzalez, N., & Conde, J. G. (2009). Research Electronic Data Capture (REDCap)- A Metadata-Driven Methodology and Workflow Process for Providing Translational Research Informatics Support. *Journal of Biomedical Informatics, 42*(2), 377-381.

Keen, P. G. W. (1980). Adaptive Design for Decision Support Systems. *SIGMIS Database, 12*(1-2), 15-25.

Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* (2nd ed.). U.S.A: John Wiley & Sons.

Marchand, D. A., & Peppard, J. (2013). Why IT Fumbles Analytics. *Harvard Business Review, 91*(1/2), 104-112.

Myers, M. D. (1997). Qualitative Research in Information Systems. *MIS Quarterly, 21*, 241-242.

Negash, S., & Gray, P. (2008). Business Intelligence. In F. Burstein & C. W. Holsapple (Eds.), *Handbook on Decision Support Systems 2* (Vol. VII, pp. 175-193). Berlin: Springer.

Olszak, C. M., & Ziemba, E. (2007). Approach to Buidling and Implementing Business Intelligence Systems. *Interdisciplinary Journal of Information, Knowledge, and Management, 2*(134-148).

Poon, P., & Wagner, C. (2001). Critical Success Factors Revisited: Success and Failure Cases of Information Systems for Senior Executives. *Decision Support Systems, 30*(4), 393-418.

Priebe, T., & Markus, S. (2015). *Business Information Modelling: A Methodology for Data-Intensive Projects, Data Science and Big Data Governance*. Paper presented at the IEEE International Conference on Big Data (Big Data), Santa Clara, CA.

Royce, W. W. (1970). *Managing the Development of Large Software Systems: Concepts and Techniques.* Paper presented at the IEEE WESCON, Western Electric Show and Convention, Los Angeles.

Sprague Jr, Ralph H. A framework for the development of decision support systems. *MIS Quarterly 4(4)*, 1-26.

Van De Graaff, J., & Cameron, A. (2013). Quest for Business Intelligence in Health Care. *Healthcare financial management: journal of the Healthcare Financial Management Association, 67*(2), 44-48.

Watson, H. J., & Frolick, M. N. (1993). Determining Information Requirements for an EIS. *MIS Quarterly, 17*(3), 255-269.

Watson, H. J., Rainer Jr, R. K., & Koh, C. E. (1991). Executive Information Systems: A Framework for Development and a Survey of Current Practices. *MIS Quarterly, 15*(1), 13-30.