Business-driven information system development

HAMK
HÄMEEN AMMATTIKORKEAKOULU

Bachelor's thesis

Degree Programme in Business Information Technology

Hämeenlinna, Visamäki, . .2012

Matti Kärnä

HAMK
HÄMEEN AMMATTIKORKEAKOULU

ABSTRACT

VISAMÄKI
Degree Programme in Business Information Technology
Systems work

Author Matti Kärnä **Year** 2012

Subject of Bachelor's thesis Business-driven information system development

ABSTRACT

The purpose of this thesis was to study how a company's information system could support the business better. The aim was to find gaps in the system's functionality using a business-driven approach. In addition, the possibility of upgrading the Visual Basic 6 based application to use the .NET technology was examined.

In the theoretical part a business-driven approach to information technology and the methods of upgrading Visual Basic 6 applications were introduced. These were unified under the concept of information technology strategy. The methods of aligning and impacting business strategy with information technology were examined.

The research was conducted as an action research based on developer and user observations. The implementations of three key business processes in the system were analyzed with the intention of reducing the company's working capital. The different Visual Basic 6 upgrade options and the business value they add were compared.

Big gaps in functionality were found especially in the sales and stock processes. The needed changes were listed and prioritized based on their business impact. For the upgrade option, interoperating with the .NET technology was chosen for its zero costs and simplicity. The methods presented here can also be used for other goals. With the business-driven approach the results are also accessible to areas outside of information technology. Implementing the suggested changes can bring substantial financial benefits to the company.

Keywords Information Technology strategy, Business-driven IT management, Visual Basic 6

Pages 34 p.

TIIVISTELMÄ

HAMK
UNIVERSITY OF APPLIED SCIENCES

VISAMÄKI
Tietojenkäsittelyn koulutusohjelma
Systeemityö

Tekijä Matti Kärnä **Vuosi** 2012

Työn nimi Liiketoimintalähtöinen tietojärjestelmäkehitys

TIIVISTELMÄ

Tämän opinnäytetyön tarkoituksena oli tutkia, miten yrityksen tietojärjestelmä voisi tukea liiketoimintaa paremmin. Tavoitteena oli löytää puutteita järjestelmän toiminallisuudesta käyttäen liiketoimintalähtöistä lähestymistapaa. Lisäksi tutkittiin mahdollisuuttaa päivittää Visual Basic 6 -pohjainen ohjelma käyttämään .NET-teknologiaa.

Teoriaosassa esiteltiin liiketoimintalähtöinen lähestymistapa informaatioteknologiaan sekä Visual Basic 6 -ohjelmien päivittämiskeinot. Nämä yhdistettiin informaatioteknologiastrategia-käsitteen alle. Työssä tutustuttiin tapoihin, joilla informaatioteknologiaa voidaan sovittaa tai käyttää vaikuttamaan liiketoimintastrategiaan.

Tutkimus toteutettiin toimintatutkimuksena perustuen kehittäjän ja käyttäjien havaintoihin. Kolmen avainliiketoimintaprosessin toteutukset analysoitiin ja tarkoituksena oli vähentää yrityksen käyttöpääomaa. Työssä vertailtiin myös Visual Basic 6 -päivitysvaihtoehtoja ja niistä tarjoutuvaa lisäarvoa liiketoiminnalle.

Etenkin myynti- ja varastoprosesseissa löytyi isoja puutteita toiminallisuudessa. Tarvittavat muutokset listattiin ja priorisoitiin niiden liiketoimintavaikutuksen perusteella. Päivitysvaihtoehdoksi valittiin yhteensovittaminen .NET-teknologian kanssa sen yksinkertaisuuden ja ilmaisuuden takia. Tässä työssä esitettyjä menetelmiä voidaan käyttää myös muihin tavoitteisiin. Liiketoimintalähtöisen lähestymisen ansiosta tulokset ovat lähestyttävissä myös informaatioteknologian ulkopuolelta. Ehdotettujen muutosten toteuttaminen voi tuoda yritykselle merkittäviä taloudellisia hyötyjä.

Avainsanat informaatioteknologiastrategia, liiketoimintalähtöinen it-hallinta, Visual Basic 6

Sivut 34 s.

GLOSSARY

BDIM = Business-driven IT management. A set of tools, models and techniques to evaluate the impact IT has on business.

COGS = Cost of Goods Sold. The inventory costs of sold products.

COM = Component Object Model. An interface technology created by Microsoft to provide a language-neutral way to link software components.

COM interop = Allows COM and .NET objects to interact with each other.

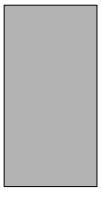
LINQ = Language Integrated Query. Component that allows data extraction and processing from a wide variety of sources using query expressions.

RegFree-COM = Registration-free COM. Allows the deployment of COM components without the need to register them.

ROI = Return on Investment. A comparison between the benefit and the cost of an investment.

XCOPY deployment = Extended copy deployment. An application installation by copying files.

TABLE OF CONTENTS



1 INTRODUCTION

The client of this Bachelor's thesis is the Baltic operations of a group operating in 11 countries in Europe. The client wishes to stay anonymous and is referred simply as 'the company'.

For its information system the company uses a tailored application written in Visual Basic 6 called Bizman. Bizman handles the invoicing process for the company. As the company has grown, Bizman has started to lag behind on its support for some business processes. This thesis aims to find the features and changes needed to make Bizman support three key business processes better. In addition, the possibility of upgrading Bizman to a newer technology and the business value it could offer are examined.

This thesis aims to answer two research questions: ”How can a business-driven approach be used to identify gaps in a business application?” and ”What are the upgrade possibilities and methods of Visual Basic 6 applications?” These both questions are examined under the concept of information technology strategy. A business-driven approach is presented, where application functionality is linked with business metrics through business processes. The different upgrade scenarios for Visual Basic 6 are presented and their possible use cases are examined.

The research questions are answered using an action research that examines the current situation and results in recommendations for future actions. The use of the business-driven approach is limited to only showcase the link between a business metric and application functionality. In addition, only one business metric and the related business processes are chosen for examination.

The results of this thesis comprise of the changes and new features suggested for Bizman and the recommended upgrade approach. The changes are given a priority based on their impact on the business. For the more critical changes, a suggested implementation is described. The most suitable upgrade scenario is chosen based on costs and impact on business. A small proof-of-concept is created with the chosen upgrade approach. The company can use these results to plan and prioritize Bizman development. The approach used can also be applied to other business processes, helping develop Bizman in other areas as well.

2 INFORMATION TECHNOLOGY STRATEGY

Bocij, Chaffey, Greasley and Hickie (2006, 580) define corporate strategy as ”Definition of the future direction and actions of a company defined as approaches to achieving specific objectives”. Strategy itself is broken down to four areas. A vision creates an easily remembered image of the company's future direction. A mission states the goals of the company. Strategies and policies guide how the strategy is carried out.

Usually substrategies that support the business strategy are organized hierarchically under it. For example marketing and business information strategies are developed under the business strategy and they can in turn have their own hierarchy of strategies underneath them. (Bocij et al. 2006, 580.)

The most important concept in strategy development is that it is a continuous process. This process can be broken down into five elements. In strategic analysis the company's strengths, weaknesses, opportunities and threats are assessed. From this assessment, strategic objectives are formulated. These objectives are then evaluated and the strategic options to reach the objectives are generated and selected. The selected options are then implemented. Finally, a control procedure is set up to detect problems and adjust the strategy if needed. Although these elements are usually executed in a sequence, they might need multiple iterations or going back to previous steps. (Bocij et al. 2006, 587–588.)

2.1 Information Technology strategy

Information technology (IT) strategy is created to define the organization's technological infrastructure. It aims to standardize software, hardware and suppliers used in the company. Its main goal is to ensure cost-effective and the most efficient support to business users. The IT strategy has to cover four main layers of the infrastructure. The most visible layer to the business users is the applications layer. This contains the company's business applications, like Customer Relationship Management (CRM) or Enterprise Resource Planning (ERP). IT strategy should cover the technological aspect of this layer, for example by answering questions about how are the applications deployed and maintained. The management of the optimal selection of applications, called the applications portfolio, is also the role of IT.

The second layer is the systems software layer. A company has a lot of other systems besides business applications. The software for these vital systems, like web servers, is defined in this layer. The third layer is the network layer where the physical network is defined. In the final layer called the storage layer, actual physical storage devices and backup procedures are defined. (Bocij

et al. 2006, 582.)

Besides software and hardware, IT also manages suppliers and external service providers. No company can produce all of their IT in-house so external resources are needed. IT must develop a strategy to manage these resources. The quality of these services is defined in a service level agreement. The agreement covers for example the rights and responsibilities of both sides, the availability level and the expected response times to support requests. (Jordan & Silcock 2006, 192–194.) As one service provider can rarely fulfill every need, companies must select the best solution for each need for the right price. This approach is called the best-of-breed. Similar services can be formed into IT service clusters. The technologies and activities that a particular service or groups of services are comprised of form the cluster, that is then offered for outsourcing. Some parts of the cluster can also be left out of the offer and kept in-house. (Jordan & Silcock 2006, 219–220.)

2.2 Information Technology risks

Jordan and Silcock (2006, 58) define an IT risk as "An IT risk is something where information technology can fail and have a negative impact on business". They continue to divide IT risks into 7 categories. First, there are IT projects that are aimed at implementing new or extending existing systems. These can miss the goals of schedule, quality or scope. Second risk category is the continuity of IT services. If an IT service is down, the business processes depending on it can be fully incapacitated. The third category is managing suppliers and service providers. The provider's inability to fulfill the service level agreement can cause major malfunctions in applications or services it offers. The fourth category is infrastructure or hardware. Hardware malfunctions or breakdowns can cause downtime or data loss. In addition, as hardware and systems develop incompatibilities can be formed between them. The three others, namely intellectual property, applications and IT and business strategy mismatch, are looked at more closely below.

Most applications contain valuable information assets. As these assets can be business critical, the possibility of the data vanishing, getting corrupted or being misused is a serious risk. Proper security and backup procedures help to fight against these problems. Also limiting the application user's data access to only what is required is essential. Some outsourcing arrangements, like telemarketing or customer service, can force a company to share an application or data with an external service provider. The procedures of how this data is exchanged and used have to be defined and controlled.

Applications themselves are also a risk category. Modern business applications are expected to run around the clock and are usually a part of a group of integrated solutions. If one application goes down it can cause large domino effects across these application groups, even across different companies. Applications also have long term risks. Some systems are not easily maintainable or they are hard to extend. These are usually cases that were initially built quickly to solve a specific need. Over time they have become larger and even business critical. As new features are added without restructuring the application, these features can break another part of the system. Systems are also vulnerable to staff changes. Without proper documentation the developer can take the knowledge with him and make the job of taking over harder for a new developer.

The third risk category comes into play when the company's IT and business strategy do not align. This category's effects are gradual but can lead to a situation where IT is blocking the path for business strategy. This can happen when the application uses an older technology that does not support a newer standard. This can lead into a dead end from a development point of view when the new business need can not be fulfilled. When organisation follows the developments in IT they find new possibilities to take advantage of. The strategic advantages gained from IT spring from a great IT strategy. (Jordan & Silcock 2006, 60–64.)

2.3 Information Systems strategy

The Information Systems (IS) strategy is deployed to ensure that IT supports the business strategy. IS and IT strategies do overlap between some elements, like selecting the appropriate application portfolio. IT focuses more on the technological side while IS is concerned with the information and business requirements. Because of this overlap they are sometimes combined and referred as IS/IT strategy. (Bocij et al. 2006, 582.)

One of the big problems in information systems planning is what is called the strategic mismatch. At a point when it is noticed that the IS/IT capabilities cannot meet the needs of the organization, plans are created to develop them. However when these new changes are implemented the company's business strategy has evolved again, possibly creating another mismatch of capabilities and needs. Also at the other end it possible overemphasize the possible gains from IS/IT development alone thus neglecting business strategy development.

Bocij and others (2006, 600–601) present two different approaches to integrating IS and business strategies. Companies can try to either align or impact. When a company chooses to align its IS strategy, it aims to support an existing business

strategy. The IS strategy is derived from the business strategy by using techniques such as Critical Success Factor (CSF) analysis. CSFs exist in every area and level of the business. They are the things that must succeed in order for that process or function to be efficient. For a CSF to succeed a number of key decisions have to be made. To support these decisions, information is needed. This is where IS steps in the picture. What data is needed and how is it supplied? This kind of analysis will help to identify the gaps in IS support to the business.

A business-impacting IS strategy aims to create new opportunities to develop business strategy directly. This impact is achieved with the adoption of new technologies or techniques. These new opportunities can be found through the use of value chain analysis. (Bocij et al. 2006, 600.) This method organises the company's key activities, like inbound logistics and operations, in a chain from conception to delivery to customer. Each activity is analysed to see how and how much they add value to the product or service down the chain. From an IS point of view this gives the possibility to look at how IS could help to increase the efficiency of an activity. But maybe more importantly it also allows to analyse the links between the activities from an IS perspective. IS allows the bridge the gaps between the activities by creating linkages between the different systems used. (Bocij et al. 2006, 596.) The biggest examples of this are Enterprise Resource Planning (ERP) solutions. ERP aims to integrate the entire value chain under one single solution. It links the different functions through the use of a central database. This removes the need of data transfers between applications of different activities. (Bocij et al. 2006, 55.)

3 BUSINESS-DRIVEN IT MANAGEMENT

3.1 Evolution of IT management

As IT started to become important to businesses in the 1980s a discipline of IT infrastructure management was birthed. It aimed to unify the views of infrastructure control in an enterprise. The IT department's main goal was to minimize business disruption. As such the IT departments were seen as the enterprises' technology providers.

Later IT started to shift its focus from infrastructure to the user. The early phases of IT Service Management (ITSM) were an opposite of the infrastructure centered view. Its main idea was to provide services to users. A service, such as corporate email, uses infrastructure objects to aid users and customers to do business. IT becomes a service provider and its performance is measured in Quality of Service. This is reached by fulfilling the Service Level Agreement objectives. IT was still seen as a separate function from business and the main concern was to align IT and business.

As ITSM has matured, it has focused more on the business as a whole rather than individual users. Infrastructure, services, staff, projects, risk management and strategy alignment are all a part of this corporate approach to IT governance. IT has evolved into a business partner. (Moura, Sauvé & Bartolini 2008.)

ITSM decision-making is still based mostly on technical measures, like availability and response time. In their research Moura, Sauvé & Bartollini (Moura et al. 2008) suggest that ITSM has to change to a more business-centric view. Decision making should be steered by business-oriented measures, like revenue or personnel utilization. This is called Business-driven IT management (BDIM). BDIM's main goal is to help the business. In order to achieve this it offers a set of models, tools and techniques to evaluate the impact IT solutions have on business performance. Using that evaluation it is possible to improve the IT solutions and the related business results. Although a still research in progress, it promises a large benefit to the business.

3.2 Implementing BDIM

Sauvé, Moura, Sampaio, Jornada and Radziuk (2006, 2) propose that BDIM is deployed using the iterative PDCA cycle shown in Figure 1. PDCA comes from words Plan, Check, Do, Act and it was initially introduced in the 1930s by W. Shewhart of Bell Labs. In BDIM context, the Plan step comprises of formalizing the problem and setting goals to align IT with business. These

planned changes are implemented in the Do step. In the Check stage, the changes are analyzed and compared against the goals that were set. In the Act step the solution is used to improve IT's quality of service. It also triggers a new cycle if the results deviated from the goals. Below, the two main techniques of the planning stage are described in more detail.

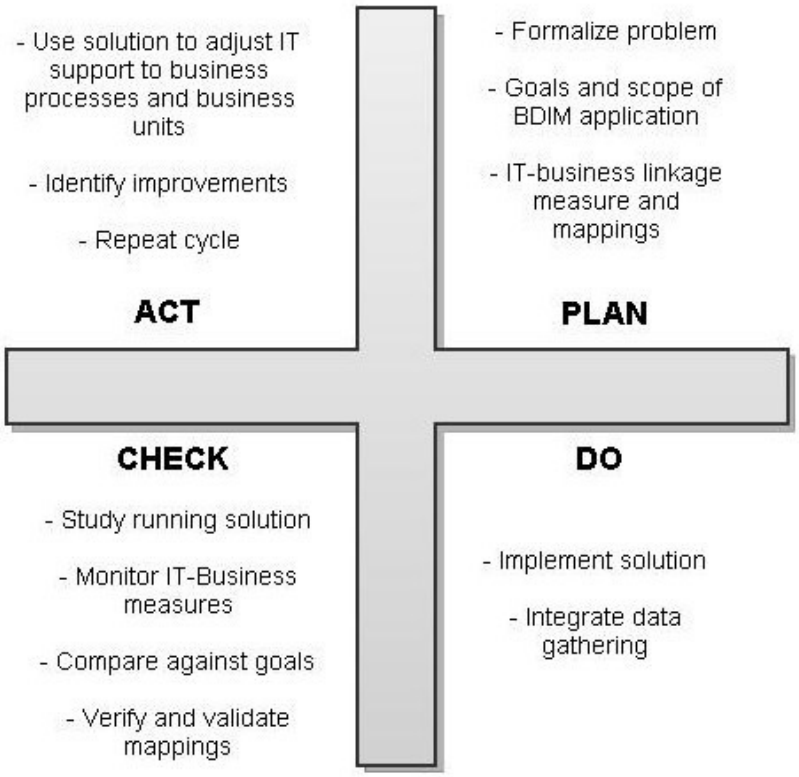


Figure 1: The PDCA steps

3.2.1 Identify business metrics

As BDIM is a business-centered approach, the first step is to select the business measures of interest. These are stated as business objectives and inferred from key performance indicators (KPIs). A KPI is a company's way to measure a certain activity's performance. If a company's business objective is to get 1000 new customers inside the next three months, a natural selection for a KPI is the number of new customers. (Moura et al. 2008.) Financial measures, although sometimes difficult to determine, are desired. They are easily understandable in areas outside IT and are a driving force in executive decision-making. Non-financial measures can still be used and even those can lead to a monetary value indirectly. This is called monetizing the impact. It is used to estimate the potential financial gain or loss. (Sauvé et al. 2006, 4.)

3.2.2 BDIM model

In the next step, the selected business metrics are mapped to their dependencies on IT performance. This is done through a BDIM model. Figure 2 shows the basic three-layer model built on top of the IT infrastructure. At the top, the business layer calculates the business metrics and feeds business inputs, such as regulations and strategies, down the model. The Business Processes (BP) layer has those processes that utilize IT services. This layer helps to link IT behaviour with business results by capturing how IT metrics or events cause changes in business metrics. The IT services layer takes into account the infrastructure's services and the way users behave while using those services. Each layer can feed their measures down to support the lower layer's decisions. The accuracy of these metrics may not be critical if the decision-making is based on picking an option and the metric is used as a comparison tool. In these cases, it just needs to provide with the right order for the options. (Moura, Sauvé & Bartolini 2007.)

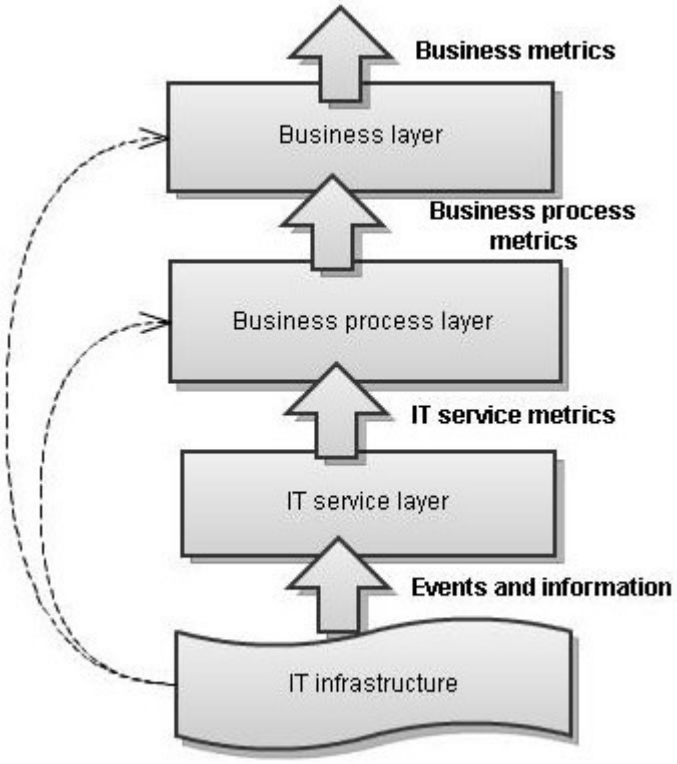


Figure 2. The BDIM model

Figure 2 has two dashed lines on the left that provide the possibility to simplify the model. This enables the retrieval of the IT metrics directly from the IT infrastructure without the need to analyze the effects on the IT service or BP layers. If the

company wants to measure the revenue loss from an unavailable database server, it can be linked straight to the processes it affects without considering the intermediate effects it has on the IT services it uses. (Moura et al. 2007.)

The BDIM model's solution is essentially a relationship between business and IT measures. This can be achieved through mathematical analysis, measurements or simulation. (Moura et al. 2007.) In some cases, it is possible to measure this result exactly but more frequently it will be an estimation. The hierarchical layers allow the use of different solution techniques across the layers. One layer can be solved mathematically and another by simulation. The model also shows how widely it is possible to apply BDIM. More often it is used to adjust IT infrastructure but it also makes adjusting business processes a possibility. (Sauvé et al. 2006, 6.)

4 VISUAL BASIC 6 UPGRADE SCENARIOS

Visual Basic 6 (VB6) is a programming language developed by Microsoft to develop Windows applications. The first version, Visual Basic 1, was released in 1991 and it revolutionized how people developed for the Windows platform. VB enabled the Rapid Application Development (RAD) approach to building user interfaces. Instead of building windows and controls from code, VB relied on a set of ready-made controls that the developer dragged on the design surface. VB also handled the plumbing code, like event handling, behind the controls. (Robinson, Bond & Oliver 2001, 4)

During the years developer needs changed. Web development was becoming more and more popular and although VB6 had some Web development capabilities it was still very much a Windows development tool. Microsoft sought after building a better architecture for the new development needs and a unified framework for application development. This led to the development of the .NET framework. A lot changes would have to be made to the language to make it a part of the unified framework. In 1999, Microsoft made the decision to break compatibility between VB6 and the .NET framework and decided to offer the possibility to upgrade VB6 projects to the new VB.NET. (Robinson et al. 2001, 5–9)

4.1 Why upgrade?

The most important factor to examine first is if there really is a business need to move to .NET. The most common reason to move is to web-enable the application. This usually means creating or consuming XML Web Services or creating web interfaces using ASP.NET. (Artinsoft 2008a.)

After the decision to upgrade has been made it is time to choose the approriate approach. Depending on the application's quality and business value there are four main options shown in Figure 3. Some options also have different variations. The most common approaches are covered in the chapters below.

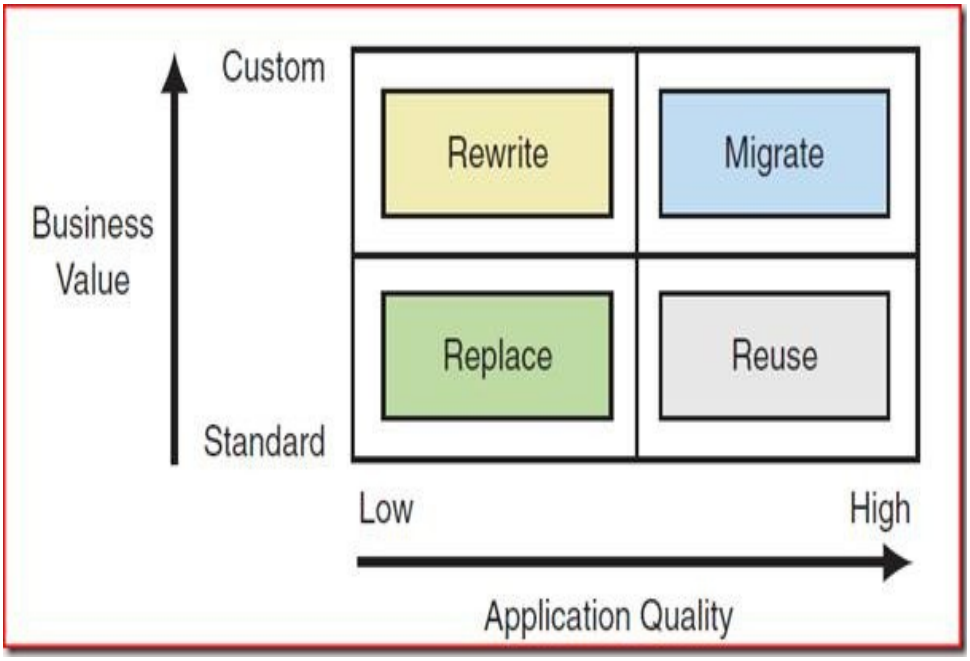


Figure 3. The Visual Basic 6 upgrade options (Nelson 2008)

4.2 The 'do nothing' approach

By far the easiest upgrade option is to leave the application to Visual Basic 6. It is important to remember that upgrading to .NET is optional. For some applications there is simply no need for an upgrade as the application is already running perfectly. It all comes down to business requirements. If there is no need to leverage .NET features the upgrade can be postponed or just skipped altogether. (Robinson et al. 2001, 45–46.)

Applications will still run on a supported platform now and in the future as the runtime components of VB6 are fully supported and shipped in Windows Vista, 7 and 8. Although the developer environment has been out of support since 2008 it is still tested for compatibility in the newer versions of Windows. (Microsoft 2012.)

4.3 Partial upgrade

The partial upgrade is ideal for n-tier applications that consist of two or more distinct projects. This enables upgrading one or more components of an application to .NET. In a three-tier application, the client-side can be upgraded while the often complex business logic and data access code can be left in VB6. The Component Object Model (COM) interop layer handles the communication between the .NET and COM components. A partial upgrade is a good stepping stone towards a full upgrade.

The upgraded components can benefit from the improved performance and scalability without the effort of fully upgrading the application. The interop between the components does add overhead, so the performance benefit is slightly smaller compared to a fully upgraded application. Also deploying applications that still have COM dependencies requires object registration and versioning that .NET objects do not. (Robinson et al. 2001, 46–47)

4.4 Phased migration and interop

In this scenario, the most important tool is the Interop Forms Toolkit made by Windows. The toolkit provides the capability to create .NET forms and controls and use them in an existing VB6 application. This allows a phased iterative migration where the application is migrated one form at a time. More importantly it allows leveraging single .NET capabilities, that are difficult or impossible to execute in VB6, like multithreading using the .NET's BackgroundWorker component. The toolkit also simplifies deployment of these hybrid applications by allowing the use of RegFree-COM. This removes the need for COM object registration and allows for a simple XCOPY deployment. (Massi 2007.)

4.5 Tool assisted full migration

In a full migration scenario the whole application is changed to utilize .NET code and technologies. From a developer point of view this seems like the most attractive option although, excluding rewriting the whole application from scratch, a full migration requires the most effort. (Robinson et al. 2001, 47)

Microsoft and its partners have published a few tools to help in the migration. The Code Advisor for Visual Basic 6.0 and the Visual Basic 6.0 to Visual Basic .NET Upgrade Assessment Tool help in preparing the code for upgrade. The Assessment Tool analyzes VB6 source code and generates a report of issues that need manual work. It also gives an estimate of the time and developer skill needed required to fix the issue. The Code Advisor is a VB6 add-in that reviews code and suggests improvements to meet coding standards based on Microsoft's best practises.(Microsoft 2005; Microsoft 2006.)

The .NET Upgrade Wizard shipped with Visual Studio 2003, 2005 and 2008. It was designed to assist in the migration by refactoring code and pointing out the issues that needed fixing. Microsoft's partners have developed tools that make this process more automatic and provide powerful features such as the possibility to generate C# code. (Artinsoft 2008b, 1.)

The migration tools have a proven track record of saving money and time. It is a very fast approach compared to a full rewrite. It also preserves all the business logic that has been built into the system over the years. (Artinsoft 2009.)

Also more modern migration tools allow for a tool-assisted rewrite where the existing code is iteratively translated to .NET where the translation rules are constantly enhanced, producing better code after each cycle. (Great Migrations n.d.)

4.6 Rewrite or replace

A full rewrite is a very tempting option as it offers the possibility to start from scratch and armed with hindsight creates a better application. This allows changing everything from programming language to architecture. Rewrite projects are very risky as all the bug fixes and business logic are basically thrown away and replaced with new code. Rewriting allows adding new features that could have been hard to implement earlier. This can also be seen as a negative as it adds to the project workload. This approach is ideal for applications where the code quality is fairly low, so migration with tools is difficult, but the level of customization is high so it has some value to the business. Low customization and low quality applications can be usually replaced by transferring their functionality to another application or using an off-the-shelf solution. (Artinsoft 2008a; Nelson 2008)

5 RESEARCH GOALS AND PROBLEMS

5.1 The starting point

For its Baltic operations the company uses a system written in VB6 named Bizman. Bizman handles the invoicing process for the three Baltic countries. It is tailor-made for the company to support its subscription based business model. It was programmed by an external developer who based it on a core of another program and MySQL database for the quick and cheap solution. This solution was ideal, as the expected lifetime of the system was relatively low as the adoption of the company's ERP was due relatively soon. But due to financial reasons the adoption was postponed and after four years the Baltics continue to use Bizman. This background is important as it explains many of the problems and risks that plague the system.

In 2012, the company wanted to develop the Baltic markets. One of the biggest identified risks was Bizman.The development and maintenance was reliant on an external developer. Also, the company's most important resource, the customer database, was hosted on an external server. To eliminate these risks the company decided to purchase Bizman and after the takeover plan to relocate the customer database to its own servers. To achieve this, the company hired the writer as an in-house Bizman developer and key user. This move was very important from a strategic viewpoint. It aimed to protect the company's most valuable resource and in the same time guarantee that the development and maintenance of this business critical system would happen inside the company. It was also noted that Bizman had not grown with the business successfully. Especially its financial capabilities were lacking and did not support all the needed business processes.

5.2 Goals

This thesis has two goals. The first one is to find out how Bizman could support the business better. This is achieved by using BDIM modeling to identify the needed improvements. The business objective chosen for the model was to lower the company's working capital. This method can then be applied later to other objectives, such as improving the follow-up of marketing activities. The second goal was to examine does Bizman need to be upgraded to .NET and what new features .NET could provide. This is done by analyzing the application's current capabilities and future needs.

5.3 What is working capital?

Working capital is the funds a business needs to pay its short-term debts and expenses. A working capital deficit means a company is not able to pay its debts while a working capital surplus affects profits negatively. (Dash & Ravipati 2009.) Working capital management covers three key areas: Inventory, accounts receivable and accounts payable. These are handled in business processes called Forecast-to-Fulfill, Customer-to-Cash and Purchase-to-Pay respectively. (J.P. Morgan 2010.)

5.4 Research methods

The main research method for this thesis is action research. It is a qualitative approach that emphasizes the interaction between theory and practice. It has two primary goals of developing and influencing. Action research is heavily based on observation and reflection. Based on the observations the current situation is assessed and goals are set. (Action research n.d.)

In the research part of this thesis the observations of a developer and users are used to assess the business process implementations. Developer observations are used when considering the upgrade options. The main goal of this research is to point out the strong link between business and information technology.

6 BUSINESS PROCESSES IN BIZMAN

For the purposes of this thesis, the BDIM model is used just to identify the changes needed in Bizman and possibly in business processes. The model is not actually mathematically solved but instead used as an illustration as it is easily approachable for areas outside of IT as well. The model is also simplified a little by leaving the IT services layer out and linking the business processes straight to Bizman capabilities.

In the heart of the model are the business measures of interest. For this model, working capital was chosen as the metric. The main reason for choosing this objective was the company's alarmingly high receivables. The second reason was to improve inventory management and make it more reliable. The three business processes linked to working capital are Customer-to-Cash, Purchase-To-Pay and Forecast-to-Fulfill. First, the way each process is done currently in Bizman is described. The missing features and current problem points are discussed. The results comprise of suggestions how to implement new features and changes.

6.1 Customer-to-Cash process

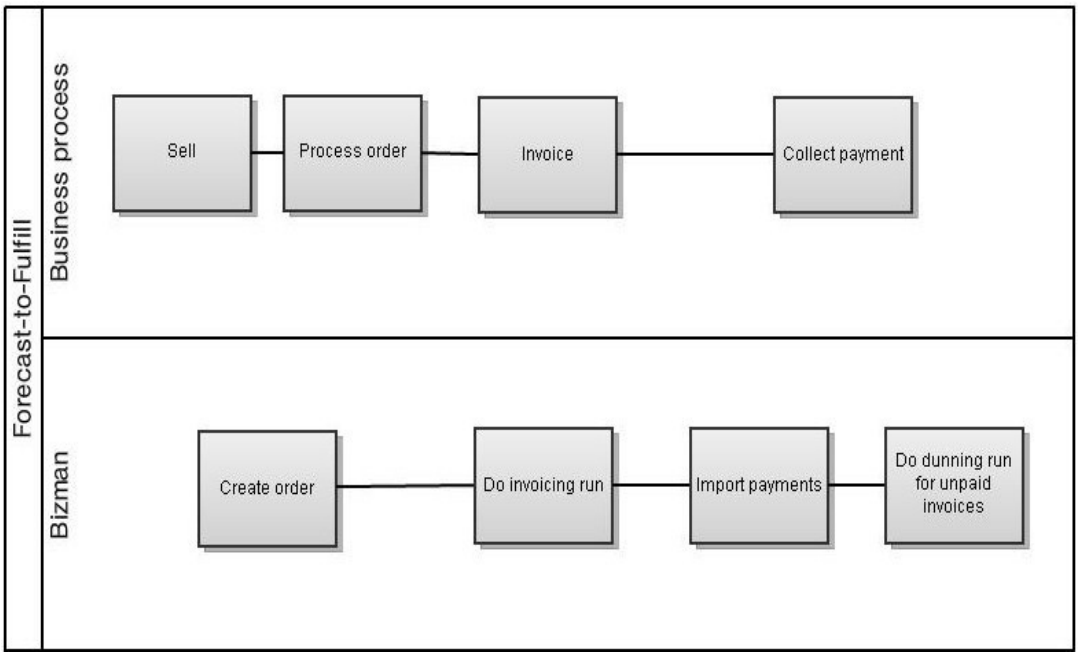


Figure 4. The Customer-to-Cash business process

Figure 4 shows the main points of the sales process. This is the business process Bizman was made for. It was a pure invoicing application at first, everything else has been built around it after that. The company's products are sold either as singles or

subscriptions. A single is one product for one order while in a subscription a new product is sent every month while the order is active. This makes the invoicing process a bit more demanding and requires a lot of extra business logic in Bizman to ensure that the maximum number of customers is invoiced in their subscriptions and are shipped a product they have not received in previous shipments.

The process starts by creating an order in Bizman. This can be made by a Bizman user, for example a telemarketing agent, or by importing orders created in campaign websites or webshops. When importing orders, Bizman tries to match the buyers to existing customers using a matching code created from some of the customer's information. Eliminating duplicates is important for credit control that is an important factor in the next step of this process. Right now, the matching only looks at one combination and if that does not match exactly it creates a new customer. It is also not making a clear enough distinction between different customers living in the same address.

Invoicing in Bizman is done through three different runs. The single run invoices all the possible single orders, the starter run ships the first product of a subscription and the regular run invoices the subsequent subscription shipments. Each run has some unique business logic but the basic idea is the same. All active orders are selected and the stock levels for the products are checked. For subscription shipments, the credit limits are enforced. Invoices are printed and sent to the warehouse for shipment. This step of the process has two big gaps in functionality. First, there is no integration between the shipment and stock transactions. It is almost impossible to tell accurately what the stock value of the shipment is. This also affects returns where the product should be returned with the same stock value it was shipped with. Second, the credit limits are calculated on a subscription basis so bad credit in one does not stop shipments in another one. This has the benefit of not stopping all shipments for good customers that have many subscriptions, because of one missed payment. It does however open the possibility for fraud. Even more alarming is that the single run does not have credit limit checking, although the customer's overall credit is usually checked during order creation by the sales agent it should be automatic. The credit limit should be formed by the total sales and payment history of the customer.

As customers pay their invoices these payments are registered into Bizman by importing accounting files sent by the payment providers. The invoice number is used as the reference to match the payment to the open invoice and mark the invoice as paid. A dunning run creates payment reminders to customers that have not paid their invoices by due date. After three reminders, the

invoice is sent to an external debt collection agency for collection. At the time of writing, this has not been implemented in Bizman yet as the decision to start debt collection is fairly new. Implementing debt collection also requires adding two additional features unsupported at this time. Sometimes the customer cannot pay the whole invoice amount so Bizman needs to allow partial payments and a possibility to write-off invoices also.This step is major contributor to the high working capital. Many unpaid invoices are two to four years old and classified as bad debt, and without debt collection it keeps accumulating. The dunning run is not without problems either. From a coding point of view it is one of the hardest procedures to understand.. It is full of old business rules and assumptions. This causes some invoice types to be dunned only once or not at all. Also, every subscription needs a separate dunning run which takes a lot of time and unnecessary manual work. Regular invoices are dunned three times. Payment statistics show that there is a significant drop in payment percentage between the second and third dunning. The third dunning could be eliminated, essentially cutting down the time an invoice can be sent for debt collection by at least one month. This requires some changes in the dunning letters as well because only the third letter mentions the possibility of debt collection.

6.2 Purchase-to-Pay process

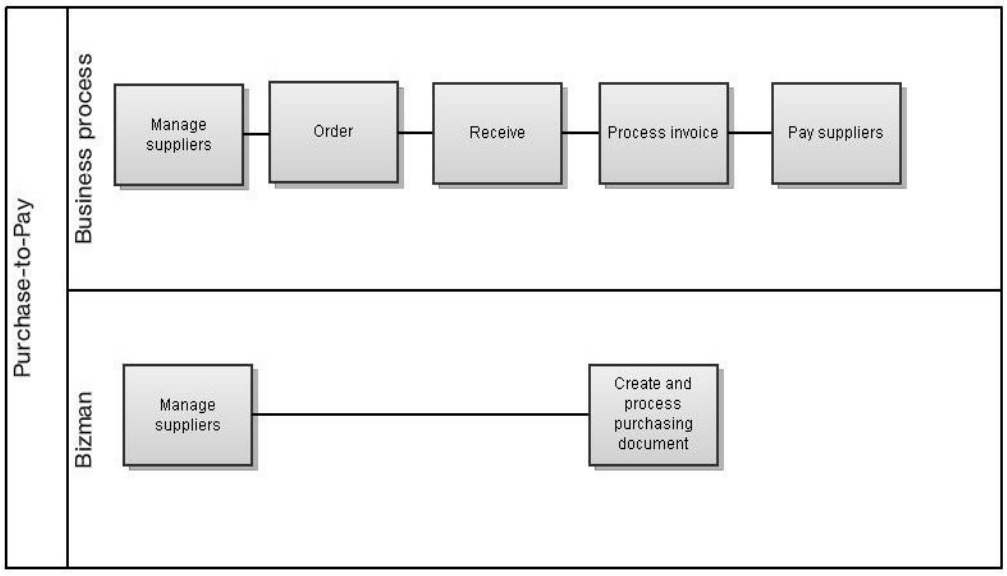


Figure 5. The Purchase-to-Pay business process

Figure 5 shows the main points of the purchasing process. The purchasing process plays an important role in Bizman. A product's stock quantity cannot be manually manipulated in Bizman, it has to always involve a stock transaction and in most

cases a purchasing document. There are three main sources to purchase from. Suppliers are external companies that either sell or produce the products and their information is managed in Bizman. The company can also make intercompany sales between the Baltic countries. This is used to create stock transfers for products that are sold in more than one country. The company can also make internal stock transfers for one product. This happens when the product is sold as a single and as a part of a subscription and the individual stock amounts need to be controlled.

Bizman's current implementation simplifies this process a lot. It does not involve creating purchasing orders or payments. It also combines goods receiving into invoice processing. External supplier and intercompany purchases have their own invoices that when processed create the stock transactions for the purchased products. The stock transactions create a batch for the product with the given purchase price and change the stock quantity. Internal stock movements do not have an actual implementation in Bizman. These cases are very rare and at the moment their transactions are done manually straight into the database. Implementing the other steps of the process would not directly affect the working capital but could offer the possibility to automate a part of the purchasing process. This could be achieved by integrating the purchasing process with the planning and invoicing processes. In certain scenarios, based on current stock quantity and the forecasted need, Bizman could automatically create a purchase order. This is a very advanced scenario and requires some changes in the other processes as well but is an example of what could be achieved by making Bizman a more integral part of the business.

6.3 Forecast-to-Fulfill process

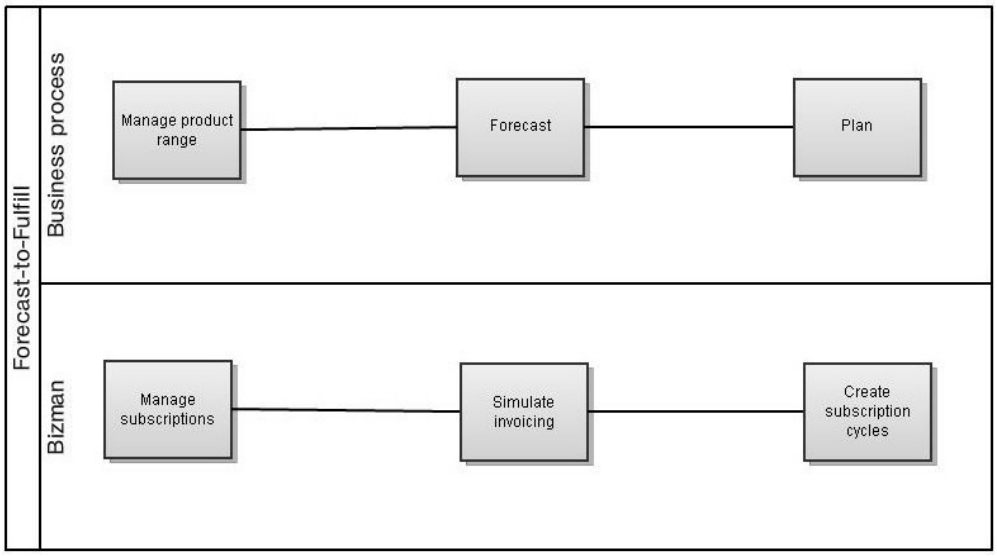


Figure 6. The Forecast-to-Fulfill business process

Figure 6 shows the inventory management process. Reliable inventory management is important in order to effectively plan and forecast the purchases needed to fulfill the customers' orders. This process acts like a messenger between the Customer-to-Cash and Purchase-to-Pay processes. This process also creates the product master data needed for the other processes. Everything starts by creating a subscription, even single products are part of a subscription. The products belonging to the subscription are added and now inherit some characteristics from the subscription. These include, unless separately defined on the product's information, the price and the default invoice type. The subscription also includes the settings for credit and return limits. These are used to block shipment to customers with bad credit and to stop the order when a customer has returned too many products from the subscription. With this basic setup done, purchases and customer orders for the subscription and its products can be done.

The company uses an external warehouse in Poland. The warehouse uses their own system to manage the stock. The challenge is to have the same information in both systems. With stock quantities this works but the stock values differ greatly. This mainly comes down to the way the systems handle batches and the accounting technique they use. The warehouse uses the first-in, first-out (FIFO) principle that marks the oldest items sold first. Bizman tries to use FIFO but fails in some key areas. First, it does not enforce the correct use of batches. This means that when shipping, Bizman does require that a batch exists but it does not check that the total quantity in the product's batches

correspond to the actual stock quantity. Few bugs in the batch determination for intercompany sales have also corrupted the batch data for some products. As the batches are pretty much invisible to the user, accessible only from the database, this has gone unnoticed for a long time. Second, when reporting the stock value of sent items at the end of each month it is not calculated using FIFO. Bizman instead calculates an average stock price of all batches without taking into account the remaining quantity. This causes old batches that have already been sold to still affect the calculated price. As this price is used for accounting purposes, it has created a mismatch between the reported and the actual stock value. As the stock prices have mainly risen, the old prices drag the reported cost of goods sold (COGS) down. This results in a higher total stock value in accounting which in turn raises working capital.

Forecasting is used to optimize purchasing and inventory levels. In a subscription-based model this means managing the next invoicing in order to maximize the number of subscribers that will be invoiced. In order to achieve this a lot variables have to be accounted for. The subscriber amount and the items they have not yet received affect how many of a certain product can be sent. This information can then be used to purchase the correct amounts of products. Right now Bizman does not really provide information required to make these decisions. Reports are needed to provide active subscribers and the items they have already received. This forecasting process can be simulated by doing a shipment run beforehand but not finishing the invoicing. This approach still needs some kind of an overview option to see the results easily.

The forecast is used to plan the subscription invoicing. In Bizman this means creating a cycle for the subscription. A cycle determines what items are offered to customers and in what order. This is done by assigning priority numbers for the items in the cycle. During a run the highest priority item is offered to the customer first. This is repeated until an item is found that can be sent to the customer or all the items have been cycled through.

7 BIZMAN UPGRADE OPTIONS

Bizman has become a business critical system and its risks have gone unmanaged for a long time. Taking over of the development and maintenance gave the company control of their key system. Windows' announcement that VB will be supported in Windows 8 removed the risk of running into a dead-end anytime soon as Bizman will continue to work even if the company switches to Windows 8. Plans to relocate the database to the company's internal servers have been made. This does require changing how Bizman is distributed to external users, like customer service agents. With these risks eliminated it is time to look at the future of Bizman. The group the company belongs to is adopting a new version of their ERP software starting in 2013. The Baltics are pretty far down in the priority list as larger countries are served first. Bizman will stay for a minimum of one to two years because of this alone. The ERP is a huge investment and with the current turnover of the Baltics return on investment (ROI) would be hard to achieve. The focus then should be maximizing Bizman's effectiveness and make it help the business grow. In the previous chapter a business-aligning method was used to analyze the current functionalities and how they could be improved. This chapter focuses on impacting business by examining the possibilities of rewriting, migration to .NET and COM interop, and how they could provide added value.

7.1 Rewriting Bizman

Starting from scratch with the newest technologies and tools sounds the best option, from a developer perspective at least. It would enable fixing the three big problems that make new development time consuming and risky. As Bizman has grown with new features they have been added without refactoring the old code. This has resulted in hard to maintain code that has duplicate procedures and undocumented bug fixes. The second problem is the tightly coupled 2-tier architecture. As the user interface and the business logic are not separated, changing them independently is not possible. Every change also requires creating a new version of the client and distributing it to the users. The third problem is the data structure. It is a cross between good and bad database design. The good tables are normalized and have relationships between primary and foreign keys. However, these relationships are not strictly defined in the database itself, so the database engine cannot enforce referential integrity when entering, updating or deleting records. Some important relationships, like between invoices and shipments, rely on joins between multiple fields that are not primary keys. This creates a possibility for duplicates. Over time the business

logic has injected itself into the data layer as well, taking over its job of enforcing integrity. One example is the creation of invoice numbers. It is created by combining the customer number and an incrementing three digit number. This is done in the business logic layer and its uniqueness is not checked in the data layer. A recently fixed bug was causing duplicate invoice numbers in certain circumstances. This makes matching payments, shipments and returns hard and causes the forementioned duplicates during joins. Rewriting would enable re-designing Bizman not just as an invoicing application but as an ERP-like solution. This would mean adding the support for accounting and bridging the gaps between the different business processes.

Rewrite is a risky investment as it stops the current development. Bizman contains 100 000 lines of code, rewriting it and the new functionality is going to take a lot of time. With the current resources of one developer it is not viable. Hiring extra resources in turn raises the costs of the rewrite. Bizman is also missing some key features that would give immediate ROI, like debt collection, and rewriting would delay their implementation further.

In a larger scale, as a group-wide investment, rewriting Bizman does offer an interesting possibility. If the group starts its operations in a new country it needs a way to invoice the customers. Rewriting Bizman to this role, as a smaller scale ERP, that would be quick to implement would be a valuable asset. As the standard business model would be already built-in, the implementation would consist of creating the country specific information and localizing the invoices and other documents.

7.2 Migration to .NET

Using migration tools would be the quickest way to newer technology. It also does not force to throw away the old codebase. Sadly .NET as a whole really does not have much to give in terms of impacting the company's business. Leveraging single .NET features can already be achieved through COM interop discussed in the next chapter. Migration does not fix the fundamental architecture problems presented in the previous chapter. Using commercial migration tools would also be a big investment for the Baltics with no immediate ROI in sight. Like in the rewrite scenario, migration also provides the possibility for a group-wide investment to transform Bizman into a group asset by using tool-assisted rewrite. It would require an investment into the tool itself and consulting to help in the process.

7.3 Bizman with COM interop

Using the Interop Forms Toolkit is probably the easiest option to justify from a business perspective, as it is free of charge. It also does not need any changes in the existing codebase, just the deployment changes. As there is no need to stop VB6 development there are no indirect costs caused by waiting either. Interop is a precision tool making certain features more viable in Bizman. One example where interop would be a very a elegant solution is retrieving currency exchange rates that are needed in some Bizman transactions. When doing intercompany sales the company has to report the VAT amount in Polish currency. Now, this is done manually by the accountants during the end-of-month reporting. They have to retrieve the exchange rate for each intercompany invoice based on its invoice date. However, the Polish bank offers this information in XML format as well. Using interop and .NET's powerful XML tools, like XDocument and LINQ, this information could be retrieved automatically during invoice processing. These XML tools also have some other uses, most notably integrating the company's webshop with Bizman's stock. The upcoming new webshop platform has the possibility to exchange information using XML. This could also be used to export orders from it.

A lot of the use cases of the toolkit are small improvements that eliminate manual work. But it does have potential for much more. It gives the possibility to do all new development with .NET and then just adding that to Bizman. For the developer this provides the very powerful and simple data development tools and a chance to learn developing with newer technologies. The downside of this approach is that it really does not give an opportunity to fix the big problems, like the architecture and data structure.

8 RESULTS

8.1 Reducing working capital

The BDIM model was used to illustrate the links between working capital, business processes and Bizman features. Going through the implementations of the business processes in Bizman revealed some key missing features and points of improvement. These can be divided into three categories based on their effect on working capital and overall business criticality. The categories can be used to prioritize the implementation order.

8.1.1 Must implement

This category holds the most important changes and should be implemented as soon as possible. They have a direct effect on working capital and offer almost immediate ROI. Implementing debt collection and adopting the use of FIFO fall into this category.

Debt collection revolves around co-operation with a debt collection agency. This in turn requires exchanging information. The first step is to select the invoices to be sent for debt collection. The invoice has to meet certain criteria. First, it has to have been dunned three times as the debt collection is mentioned in the third dunning letter. It also has to be past the due date of the third dunning and the original invoice date has to be in 2011 or later. A file is generated of the invoices selected and the customer data linked to it and sent to the agency. At the time of writing almost 10% of the company's receivables would be eligible for debt collection.

When a customer pays there are two options. Either the payment is made to the agency or straight to the company. The agency collects their payments into accounting files that can be imported into Bizman like regular payments. If the customer pays to the company it is also imported among the other regular payments but a small trigger has to make an extra entry for that invoice in another table. These entries, called adjustments, have to be sent to the agency to reduce the capital the customer owes.

Two other unsupported features have to be implemented as well. If the customer cannot pay the whole invoice amount, a partial payment should be registered. Bizman's data structure does allow this without no changes in the database. Implementing this requires only a new payment procedure that checks that the invoice will not be overpaid. Some other areas of Bizman check that the invoice is paid by checking the existence of a payment. When implemented, this payment can be a partial one so these

checks need to be changed to compare the paid amount with the invoice amount. The other feature needed is write-off. Some debts can be judged as uncollectable. These need to be cleared from the accounts receivable and booked as credit loss. The write-off procedure has to just check the invoice's unpaid amount and create a write-off entry for it in a dedicated table. These entries are then reported for accounting at the end of each month.

In order to make inventory management reliable, Bizman needs to use the same FIFO principle as the Polish warehouse. In order for this to work some key changes have to made to how Bizman handles batches. The possibility to inspect a product's batches and total stock value has to be created to the user interface. The use of batches has to be made mandatory in shipping by enforcing batch quantities and adding batches into shipment lines. The former is achieved by not allowing shipping done for items that have a mismatch between stock and batch total quantity. This check needs to be added to the start of every shipping routine and inform the user if the check fails. The latter is a little more complicated. This has to be done to link warehouse transactions and shipments reliably. When the shipment has the batch it was shipped from, it simplifies the calculation of the COGS of shipments. It is also required to process returns correctly by returning the item to the same batch it was shipped from. This functionality requires some database changes as the batch field has to be added. The batch determination procedures have to be changed into functions that return the batch number to the calling procedure. The returned value can then be inserted into the shipment details.

As the stock value calculation currently does not conform with FIFO, it needs to be changed to calculate the value based on the remaining quantity of a batch and its price. The most important change locations are the end-of-month reports that are sent to the accountants. Bizman also has a report inside the user interface for total stock value that needs to be fixed.

In order for FIFO to actually work, Bizman needs to be updated with the correct batch quantities and prices. These figures can be obtained from the warehouse. The old batch and transaction data can be transferred to history tables just in case. The new data is uploaded into empty tables and a version containing all the FIFO changes can be published. This change will fix the stock value mismatch in accounting but more importantly it makes Bizman's stock values reliable and should prevent the mismatch occuring again.

8.1.2 Should implement

This category holds the features and changes that should be implemented after debt collection and FIFO. This category has an effect on working capital and provide ROI inside a short timeframe. Changing the dunning run, changing the credit limit check, adding new customer matching options and improving forecasting fall into this category.

The dunning run requires a couple of changes to improve its coverage and efficiency. First, all invoice types should be dunned three times. This is easily achieved by eliminating one check in the run. Second, preparations for eliminating the third dunning should be made. This means changing the second dunning letter to mention debt collection. The debt collection run can then be changed to include invoices that have been dunned with the new second dunning letter. This effectively cuts the time it takes to send an invoice to debt collection by a minimum of one month. Third, the need to do the dunning run separately for each subscription needs to be eliminated. One run should cover the whole country making dunning an easier routine. This requires some refactoring of the run code and a partial rewrite.

Credit limits should be added for single orders as well. This requires changing the credit limit to function at the customer level instead of the subscription level. The group uses an implementation like this in its ERP that could possibly be emulated in Bizman. This would also benefit from a better customer matching mechanism. It would require some additional checks for duplicates where the current method fails. These should check for combinations of address, telephone and name. This will prevent Bizman from creating a new customer with a new credit limit and help stop shipments to bad payers.

The forecasting tools are mostly already in place but they still need to be improved, checked for correctness and made more user-friendly. An invoicing run can be simulated but it is missing a proper overview of the results. Also it is somewhat time consuming and blocks other work from being done. This could be possibly run in the background, maybe with the help of .NET, or timed to run outside office hours.

8.1.3 Could implement

The last category is reserved for the more vague ideas and visions. The priority for these are low and they offer no direct effect on working capital. Adding more features to the purchasing process and some possibilities provided by .NET fall into this category.

Although the purchasing process is working fine as it is, more of it could be implemented in Bizman. This would mean that purchase orders and supplier payments would be tracked in Bizman as well, making it a more documented process. Implementing these features is not a huge task but the purchasing process only starts to provide ROI if it is integrated with the other processes. This possibility is unexplored but would require a lot of extra business logic in order to work.

The possibility to leverage the power of the .NET framework does offer some interesting scenarios. This could enable true integration with other applications or services with Web Services. These could be the company webshop or postal services like address checking or shipment tracking. These and other visions are more realistic thanks to .NET**.**

8.2 Bizman and .NET

No investment need or additional commitments make COM interop the best option for Bizman to utilize .NET. With its relative simplicity and the ability to target individual .NET features, makes it a powerful asset to use in Bizman development. In this chapter a small proof of concept is created where exchange rates are retrieved to Bizman with the help of the interop toolkit and utilizing XML and LINQ.

First, the .NET parts are constructed. A simple interop forms toolkit user control project is created. The user control will comprise of a simple Datagrid, a LINQ query that parses the exchange rates from the XDocument and a button that fires these actions and loads the results to the Datagrid (Figure 7). The interop toolkit generates all the necessary plumbing code required to make the interop work.

```
Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click

    Dim x = XDocument.Load("http://nbp.pl/kursy/xml/lasta.xml")

    Dim items = From row In x.<tabela_kursow>.<pozycja>
                Select Currency = row.<kod_waluty>.Value,
                Amount = row.<przelicznik>.Value, ExchangeRate = row.<kurs_sredni>.Value

    DataGridView1.DataSource = items.ToArray

End Sub
```

Figure 7. Code for the exchange rate retrieval

The interop project can then be referenced in the Bizman project and the user control be added to a VB6 form. This makes all the methods and events of the user control accessible from VB6.

Compared to a normal Bizman deployment, the interop adds two new requirements. The file containing the user control has to be copied with the normal Bizman executable. In addition, an XML manifest file has to be created linking these files (Figure 8).

Bizman.exe.manifest - Notepad

File Edit Format View Help

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
    <assemblyIdentity type="win32" name="Bizman" version="4.2.0.1" processorArchitecture="MSIL" />
        <dependency>
            <dependentAssembly>
                <assemblyIdentity type="win32" name="FirstInterOpTest" version="1.0.0.0" />
            </dependentAssembly>
        </dependency>
</assembly>
```

Figure 8. The XML manifest

The Bizman executable, the user control file and the manifest are then ready to be copied to the installation folder and offer a seamless user experience in a production environment (Figure 9).

Currency	Amount	ExchangeRate
THB	1	0,1058
USD	1	3,2503
AUD	1	3,3735
HKD	1	0,4193
CAD	1	3,2525
NZD	1	2,6469
SGD	1	2,6523
EUR	1	4,1521
HUF	100	1,4615
CHF	1	3,4452
GBP	1	5,1682
UAH	1	0,3958
JPY	100	3,9993
CZK	1	0,1628
DKK	1	0,5567
ISK	100	2,5599
NOK	1	0,5627

Button1

Figure 9. Exchange rates retrieved in Bizman

Although this was a very simple example, it does show how easily .NET can be utilized from VB6. With this possibility tested, it is easy to implement more complex features in a production environment in the future.

9 CONCLUSIONS

The purpose of this thesis was to find out the changes and new features needed to help Bizman support the business better. In addition, the possibilities and the added business value of upgrading Bizman to .NET were researched.

The two quite distinct goals were unified with the concept of IT strategy. This was the main part of the theory. IS strategy introduced the aligning and impacting strategy approaches that were both used later in the research. For aligning Bizman with the business, BDIM was introduced. With the BDIM's model the business metric of working capital was linked to Bizman's functionality through the business process layer. For impacting the business, the upgrading of Bizman to .NET was considered. The different upgrading scenarios and methods were introduced in the theory part.

With the help of the BDIM model, the ways the sales, purchase and inventory management processes were done in Bizman were described. The process implementations were examined with the goal of reducing working capital. A big gap in functionality in the sales process was the lack of debt collection. Another big fault was the inventory management's incorrect batch and accounting implementation. Other changes and new features that would help reduce working capital were introduced as well.

For upgrading Bizman three different scenarios were considered. These were a full rewrite, assisted migration and COM interop. Examining these scenarios also brought light to the problems in the way Bizman is built. Serious problems in the data structure and the tight coupling of the business logic and user interface were recognized. When taken to a larger scope, upgrading also offered the possibility to utilize Bizman as a group-wide information system for smaller countries.

The results provide changes and new features divided into three priority groups based on their effect on working capital. They also contain a suggested implementation method for the more critical changes. Based on the zero costs and simplicity of the COM interop, it was chosen as the upgrade method of choice. It was also recognized that a complete .NET migration would not offer enough business value. With COM interop, single features of .NET can be leveraged relatively easily.

The company now has a development plan for Bizman from a working capital point of view. The methods presented here can be used to create a similar plan for other business processes as well. A business-driven approach is more accessible to areas outside of IT, making collaboration and justifying investments easier. With the changes suggested here implemented, the

company can expect substantial financial benefits.

10 SOURCES

Action research n.d. Tutkimus- ja kehitystyön menetelmät. Accessed 10th November 2012. http://www.elearningcentre.hamk.fi/tko/menetelmat/toimintatutkimus.html

Artinsoft 2008a. Realities of upgrading large real world applications. Accessed 16th September 2012. http://www.artinsoft.com/whitepapers/Realities%20Of%20Upgrading%20Large%20Real%20World%20Applications.pdf

Artinsoft 2008b. Visual Basic Upgrade Companion and UpgradeWizard: Comparative Summary. Accessed 16th September 2012. http://www.artinsoft.com/whitepapers/VBUC%20vs%20UW.pdf

Artinsoft 2009. 5 myth-busting reasons for choosing an automatic migration vs. A manual rewrite. Accessed 11th September 2012. http://aisstorage001.blob.core.windows.net/aiscom/whitepapers/5%20Reasons%20for%20choosing%20migration%20vs%20rewrite%20-%20v2.pdf

Bocij, P., Chaffey, D., Greasley, A. & Hickie, S. 2006.Business information systems. 3rd edition. Harlow: Pearson Education Limited.

Dash, M., Ravipati, R. 2009. A liquidity-profitability trade-off model for working capital management. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1408722. PDF file.

Great Migrations n.d. Great Migrations methodology. Accessed 19th September 2012. http://www.greatmigrations.com/services_methodology.aspx

Jordan, E. & Silcock, L. 2006. Strateginen IT-riskien hallinta. Finnish: Kaskas Design. Helsinki: Edita.

J.P. Morgan 2010. Steps to successful working capital optimization. Accessed 19th September 2012. www.jpmorgan.com/cm/BlobServer?blobkey=id&blobnocache=true&blobwhere=1158593576665&blobheader=application%2Fpdf&blobcol=urldata&blobtable=MungoBlobs&blobheader

name1=Content-disposition&blobheadervalue1=attachment;filename=Working_Capital_Optimization_Presentation_June_2010.pdf

Massi, B. 2007. Interop forms toolkit – your new best friend. Accessed 10th September 2012. http://blogs.msdn.com/b/bethmassi/archive/2007/06/13/interop-forms-toolkit-your-new-best-friend.aspx

Microsoft 2005. Upgrading Visual Basic 6.0 Applications to Visual Basic .NET and Visual Basic 2005. Accessed 11th September 2012. http://msdn.microsoft.com/en-us/library/aa480541.aspx

Microsoft 2006. Code Advisor for Visual Basic 6.0. Accessed 11th September 2012. http://www.microsoft.com/en-us/download/details.aspx?id=1222

Microsoft 2012. Support Statement for Visual Basic 6.0 on Windows Vista, Windows Server 2008, Windows 7, and Windows 8. Accessed 10th September 2012. http://msdn.microsoft.com/nb-no/vbrun/ms788708%28en-us%29.aspx

Moura, A., Bartolini, C. & Sauvé, J. 2008. Business-driven IT management: Upping the Ante of IT. IEEE Communications Magazine, vol. 46, no. 10, pp.148–153. Accessed 5th October 2012. http://www.sites.google.com/site/jpsauve/COMMAG-BDIM-UppingAnteIT-accepted_ve.pdf

Moura, A., Sauvé, J. & Bartolini, C. 2007. Research challenges of business-driven IT management. 2nd IEEE international workshop on business-driven IT management. Munich, Germany. 21 May 2007. IEEE. Submitted paper. Accessed 3rd October 2012 http://www.hpl.hp.com/personal/Claudio_Bartolini/download/BDIM%2007%20papers/27156.pdf

Nelson, E. Reuse vs Migrate vs Reuse vs Replace. Accessed 19th September 2012. http://blogs.msdn.com/b/goto100/archive/2008/11/03/rewrite-vs-migrate-vs-reuse-vs-replace.aspx

Robinson, E., Bond, MJ. & Oliver, I. 2001. Upgrading Microsoft Visual Basic 6.0 to Microsoft Visual Basic .NET. Microsoft Press. Accessed 16th September 2012. http://msdn.microsoft.com/en-us/vstudio/ms788236

Sauvé, J., Moura, A., Sampaio, M., Jornada, J. & Radziuk, E. 2006. An introductory overview and survey of business-driven IT management. Accessed 1st October 2012. http://www.dsc.ufcg.edu.br/~sampaio/Artigos/anintroductoryoverviewandsurveyofbusinessdrivenitmanagement.pdf