Impact of Methodology and Software Tool Selection on Economical Aspects of Software Development Process

Željka Požgaj
Faculty of Economics and Business
University of Zagreb
Trg J.F.Kennedy-a 6, 10000 Zagreb, Croatia
zpozgaj@efzg.hr

Danijela Munić
RIS d.o.o.
Pilepčić 10, 51215 Kastav, Croatia
danijela.munic@gmail.com

Abstract. *Regardless of the development process (generic or agreed product), two key moments in the software development process are: adequate methodology selection and software tool selection suitable for the software development process. During the election process, time required for software development and costs that arise from that process play important roles in the final selection.*

Starting from software production, we will analyze the impact of methodology and software tool selection on economic aspects of the software development process. The study will be based on MIRIS, the first Croatian methodology applied on the information systems development; and on Clarion, the 4GL and code generator. The paper will present the results of an empirical research based on the case studies. We will discuss the time needed for the development of particular software, the results of cost-benefit analyses and SWOT analyses.

Keywords. MIRIS methodology, Clarion code generator

1 Introduction

The end of the twentieth century has brought wider implementation of business information systems in companies in order to achieve greater competitiveness i.e. offer high quality service and information to final users. Considering the trends, it is not unusual that due to the specific way a business is conducted some companies decide to develop fixed business applications (tailor-made applications) and do that in cooperation with companies that specialise in business application development. Such business cooperation is carried out to the satisfaction of both parties, but at the same time initially set deadlines are often extended and costs increased. Because of that it is important that the company developing software carefully chooses which application development methodology and which program code generator they will use in order to facilitate the development process itself.

After the introduction, in the second chapter we shall deal with basic features of information system design. MIRIS methodology is dealt with in the third chapter, and the fourth chapter presents features of the software tool Clarion. Fifth chapter deals with the research data from the studies that tried to explore the following: (1) the impact of the programmer's knowledge of methodology on the time needed for application development; (2) the impact of the knowledge of methodology and the code generator implementation on the cost-effectiveness of the business application development (cost-benefit analysis); (3) SWOT analysis of the situation when a company developing business applications decides to introduce development methodology and application generator. Sixth chapter is the conclusion.

2 Information systems design

Each information system is developed on the principles of a real (business) system. It represents computer-assisted real business and it has to be in accordance with company's business processes.

Scientific definition of methodology is too broad to be applied on specific fields but we can narrow it to the field of information sciences or even better – information system design. In that way a term specialised methodology of information system development is coined that

represents recommended line of steps and procedures (methods) that are followed and carried out during information system development [1]. Further in this article this specialised methodology will be referred to as only methodology in order to facilitate the reading of text.

Methodology of information system design determines the way in which it is possible to analyse a real system, realise the models on following the analysis and in the end implement a final, computer-assisted, information system. The methodology was developed in order to simplify all information system planning, control and development processes, but also in order to formalise the whole process and make it temporally measurable. A great number of commercially available methodologies exist in the field of information system design.

Implementation of one of the methodologies results in a model design that represents a theoretical creation demonstrating an object, process or system using a set of variables and set of connections existing between these variables. The model is created through process of abstracting reality and recording components crucial for a particular model, and the result of the process is a visual representation of reality: plan, chart or any other set of information. The process of creating a model is called modelling.

2.1. Process model

Dealings of each business entity can be divided into a series of activities that employees have to carry out in order to initiate, perform and terminate a business activity. Logical and temporal sequence of activities represents a business process. Integrating all business processes gives us an outline of overall dealings. The most intuitive way of outlining business processes is by using graphic methods, which enables better process formalisation and specification.

2.2. Data model

Each realistic system in its dealings receives processes and distributes information crucial for its existence and further development. Data modelling includes data object analysis, object connection identification, data dissection, making a plan for the future database but also a plan for object-oriented programming.

3.1. Resource model

Resource model is a technological basis of the information system that describes all aspects not dealt with in the process and data model: cadres, organisational units and informatics equipment. Methods for resource modelling are not strictly defined and there are various ways of defining this model: textual descriptions, organisational maps, network and other charts.

3 MIRIS methodology

MIRIS (**M**etod**I**ka za **R**azvoj **I**nformacijskog **S**ustava) methodology is a collection of methods and instructions describing information system design and development. The development of this specialised methodology began in 1984 and was first published in 1995 in the book called *Razvoj informacijskih sustava* [1]. Since then it has continually been upgraded with new knowledge from the development field of information sciences, and the current valid version is number 3. In February 2006 MIRIS methodology version 2 [5] was given a *Croatian Creation* (in original *Izvorno hrvatsko*) label by Croatian Chamber of Economy. The methodology is continuously perfected and following the changes in the field of information sciences (objective approach, active methods...)

MIRIS methodology includes entity relationship method and structured system analysis method, which are supplemented with concepts and rules derived from years of designing and developing information systems. Improvements can be seen in the following areas: process of data and process modelling, correlation of methods, preparations for interviewing users, expanding methods for data modelling and simplification of the transfer of data model into database relational scheme.

Basic thesis of MIRIS methodology is founded on the decomposition system. Business processes and data flows present in the system are decomposed. Decomposed processes and data flows are later at the software production stage organised into modules whose programming is performed according to the principle of iterative development – system is divided into smaller functional units which are then at a later development stage united into final product.

MIRIS covers the field of information system design and business application development.

Activities are divided into those that refer to logical formation (information system strategic planning, main and executional project) and physical formation (software production, introduction, implementation and maintenance).

The methodology does not depend on the chosen database and software tools. It can be used in information system construction using 3GL language for files, 4GL language for relational databases, in application generator development or combined with CASE tools.

4 Business application development with generator implementation

Program code generators are not a new concept in business application development, but their implementation is becoming ever more significant. The reason for that lies in the need for an application to be implemented in the shortest time possible and at the lowest cost possible in order to become competitive on the market. From the perspective of business entities whose primary activity is business application development, the implementation of generators enables considerable savings in that it shortens the time needed for the development of routine operations (updating and searching data, reporting and other), and automatised operations are less (or not at all) subject to errors so their maintenance costs are reduced.

Program code generator works on the principle of reading metadata (e.g. database model or data dictionary) and creating completed and well-structured program code for performing routine operations – everything that is repeated can be automatised, and a programmer is left with the task of filling in the blanks with controls and program code usually used in business process and application [8].

Considering that the market offers a great variety of program code generators, what needs to be considered when deciding on the generator is the type of business application being developed and the features of each generator as it is of greatest importance that the generator speeds up the programming process and justifies time savings in programming application interfaces and routine operations. Along with what has been stated, one should also take into consideration advantages and disadvantages of a generator (Table 1 and Table 2), especially the possibility of modification and extension of the generated program code [3].

Table 1. Advantages of program code generators

ADVANTAGES
Consistent and stable code not subject to errors (machine-generated)
Enables fast application development
Adjustable
Installed support for the modification of users' requests and database structure
Programmers can concentrate on problems typical of business process
Programmers can learn from the generated code

Table 2. Disadvantages of program code generators

DISADVANTAGES
Has to be written before the application programming begins
Applicable only under certain conditions
Certain code parts have to be reprogrammed (or overridden) manually
Database structure has to be well defined and normalised
It can generate a great amount of unnecessary program code and increase application size

4.1. Clarion software tool – program code generator

Clarion Software Corporation was founded by Bruce Barrington in 1982. In April 1986 the first version and in 1988 the second version of the Clarion program tool was introduced. In the late 1980s the tool was considered to be one of the leading tools for application and database development under DOS operating system along with dBASE, R:Base and Paradox. Although in the early 1990 Microsoft developed new solutions that reduced the implementation of this tool, Clarion managed to maintain a market share and continued to develop despite the new situation considering it was the first program code generator that used templates and one of the DOS program languages which included 4GL tools for display design, report writing and access to databases. In 2000 the development of Clarion software tool goes into the hands of the company SoftVelocity. In 2007 the newest version 7.0 that supports Microsoft .NET technologies is being tested which shows that the

company is following current trends in software tools development [2], [8].

As opposed to most software tools, Clarion was designed exclusively for work with databases and it supports a wide range of databases that are a foundation of the graphic user interface design. The main feature of this software tool is a programme language based on the use of templates, high visual quality and the promptness of application design. Clarion is open for adding templates that can, depending on the demand, be purchased as separate products from SoftVelocity or other independent manufacturers, and it even allows the programmers themselves to make a particular solution for an application and install it into the tool. There is even an option of installing templates that will generate applications in other program languages.

According to the report published in the late 1990s by TopSpeed Corporation [8], Clarion was compared to three then current program languages used in application design: Delphi, Visual Basic and PowerBuilder. Development promptness, maintenance efforts and application size were compared. Programmers had to work twice as much on application development if they used other software tools and not Clarion. Maintenance efforts and efforts of the implementation of application modifications decreased fourfold in favour of Clarion and the application size, which is crucial for its promptness and transportability, was five or six times smaller than in other applications. All that has just been stated is a direct consequence of the program code generating and its optimisation.

5 Impact of methodology and tools on application development

The goal of each business application manufacturer is to develop a new application in shortest time possible and at the lowest costs possible in order for the company to achieve profit in a reasonable time period. Implementation of a well-structured methodology in designing affects the time period needed for finding a solution to the problem a project team is faced with in that it shortens it, and the design of a logical and stable model (process and data) ensures the development of a business application that is operationally stable, prompt and easy to use.

Case 1: knowledge of methodology, time needed for application development

Positive outcome of the MIRIS methodology as a basis for information system design was confirmed by an empirical research based on the case study of a business application that processes 25 different documents and whose design took 9 days to complete (2 days for process modelling and 7 days for data modelling) [6]. The assumption that the system analyst is not acquainted with any methodology lead to conclusion and assessment that the system analysis would take 25 days, i.e. 1 day per document that needs to be processed in the system. This data is approximative and it only refers to the analysis of data contained in the document, but not the correlation of data.

Research was carried out based on the time a programmer spent designing program support for processing 7 documents. The time was then put in proportion with the number of relation attributes (entities, tables) and a constant K was derived (expressed in hours), which represents average time spent per attribute. The knowledge of methodology assumes that the number of relation attributes will be smaller than the real amount of document data due to the implementation of normalisation rules and that it will be possible to generalise certain document groups and join them into one relation.

The assumption that the analyst does not possess knowledge of any methodology leads to conclusion that the analysis and programming will include only document data but not the correlation of data. Thus we can conclude that K hours will be spent on each piece of the data contained in the document (it is assumed that the programmer considers all document data as relation attributes).

The results of this empirical research show that in the case of non-existing knowledge of methodology the number of relation attributes increases by 36.67%, and since the number of attributes (data) is multiplied by the constant K, the time needed for programming application software is also be prolonged.

Case 2: Cost-benefit analysis and new application development

Many business decisions today are made solely on the ground of company's cost analysis, i.e. on the management's estimate of whether a new

product will increase or decrease company's profit in short, i.e. long time period. In both cases competitiveness of companies on the market plays a significant role in the profit being increased or decreased by a certain business application development. In business application development costs differ depending on the application type:

- generic applications – developed for the free market, i.e. anonymous user
- fixed application – tailor-made for a familiar user

Data from the price-list of Croatian Association of Business Advisors was used in the cost–benefit analysis, and the price projection for 2006 was made according to data about price growth index released by Croatian Central Bureau of Statistics (Statistical Chronicle for 2006 and 2007) [4]. The data refer to the prices of industrial products.

According to the price-list, total development price is reached after taking into account the number of hours needed for the development, costs per hour (236.07 monetary units), structural contents coefficient (1.8 – informatics and communication projects) and company size coefficient (0.8 – small company) [7].

Hardware supply was not taken into account in the cost-benefit analysis because the platform for database location already existed, and client computers stand for direct costs of the final users. The calculation is based on the following gross development costs per hour (Table 3):

Table 3. Gross development costs per hour

	Costs	Profit
Expert	120	340
Employee	100	290

The use of methodology at the stage of documentation gathering and planning reduced the work period to a total of 10 days, and the use of a program code generator meant that the final product was ready in 2 months. From the data stated one can conclude that a completely functional product can be available on the market 3 months after the deal is made.

During the first year, manufacturers' costs refer mainly to business application design and development (costs to the amount of monthly salaries of the people working on the project, marketing costs). During the remaining three years dealt with in the analysis, costs are manifested through additional involvement of staff entrusted with the application maintenance due to both legal modifications as well as modifications requested by the users.

Income from the business application sales in the first year amounts only to a flat-rate sum agreed upon with the initial user which covers a little less than two thirds of development and advertising costs. Market analysis has shown that there are 250 potential users, but considering that there are similar products present on the market, the manufacturer has set a goal to take over 20% of the market share in the period of three years which resulted in number of 16 users a year.

By comparing costs and profits it is clear that in the first year the project is operating at a loss, but with increase in the number of users a business application becomes cost-effective during the second year. If the application is taken as a generic product without the contract with the initial user, the cost-effectiveness is only achieved during the third year.

Case 3: SWOT analysis of methodology and generator implementation in application development

Cost analysis for business application design and income prediction for the future are a basis for carrying out a SWOT analysis that completes total analysis in cases when development methodology and application generator are just about to be introduced into the project. SWOT analysis is a tool for strategic planning that is used for assessing strengths, weaknesses, opportunities and threats in a project.

Fig. 1 shows SWOT analysis matrices in development methodology implementation and Fig. 2 shows matrices in application generator implementation.

Cost-benefit analysis based on the case study and SWOT analysis of methodology and application generator implementation lead to conclusion that development methodology and application generator affect the reduction in time necessary for business application design and development, but at the same time it can be concluded that it is best that each company decides on a constant implementation of the same development methodology and application generator for all future projects in order to cover the costs of methodology training courses and supply of the development tools.

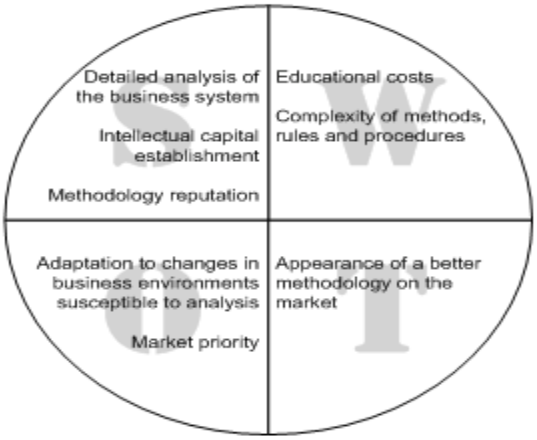

Figure 1. SWOT analysis of methodology

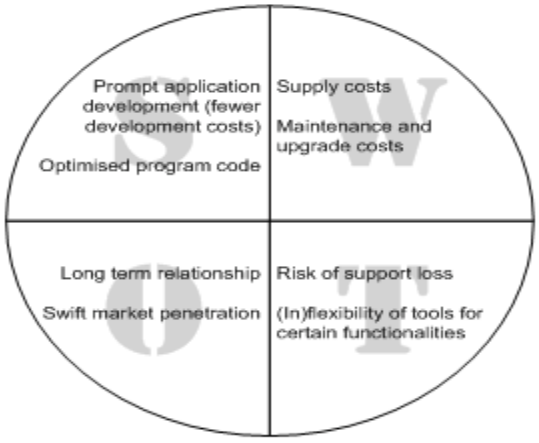

Figure 2. SWOT analysis application generator implementation

6 Conclusion

The key initiator of all social segments in information society is prompt access to information. Trends in information technologies enable the manufacturers of business applications to establish themselves on the market and offer their products in all segments of business, but at the same time their solutions are expected to be in line with the information technologies development and society's needs, and that means prompt development, stability and application sustainability at a reasonable price. Low costs of business application design and development are a result of knowing and using development methodology and adequate software tool such as program code generators with ready-made solutions for basic functionalities in the application performance. Development methodology enables the design of project that will provide a well-structured and stable basis for programming with minimum interventions that refer to the modification of project during the development (programming) stage. A wide range of software tools (code generators) will allow for prompt and relatively simple application development. Using data from study analysis this article tries to show the advantages of implementing adequate methodology and code generators in the process of business application development.

References

[1] Pavlić, M: **Razvoj informacijskih sustava,** Znak, Zagreb, 1996.

[[2] Spitzer, T: **Catching Up with Clarion**, available at http://www.dbmsmag.com/9606d18.html accessed: 6th January 2008.

[3] Stephens, M. **Automated Code Generation,** available at http://www.softwarereality.com/programming/code_generation.isp, accessed: 16th December 2007.

[4] Državni zavod za statistiku RH, **Statistički ljetopis 2006**, available at http://www.dzs.hr, accessed: 25th February 2008.

[5] **Metodologija MIRIS verzija 2**, available at http://www.ris.hr , accessed: 20th May 2008.

[6] **Programsko rješenje za ordinacije opće medicine,** available at RIS d.o.o. Pilepčić 10, 51215 Kastav, Croatia

[7] **Programsko rješenje za ustanove za zdravstvenu njegu u kući,** available at RIS d.o.o. Pilepčić 10, 51215 Kastav, Croatia

[8] **The TopSpeed Legend**, available at http://www.artofprogramming.net/dev_clarion_legend.html , accessed: 6th January 2008.