Applying of Smart Technologies: Evaluation of Effectiveness

Zane Bičevska

Applying of Smart Technologies: Evaluation of Effectiveness

Zane Bičevska

University of Latvia, 19 Raina Blvd., Riga, Latvia
Zane.Bicevska@di.lv

Abstract. This paper is dedicated to various aspects and perspectives of applying the smart technologies in software solutions. Based on experience acquired during software development projects, the author proposes criteria to evaluate the effectiveness of enhancing of existing products with smart technologies. The evaluation of effectiveness can be used to support decision making on implementing the smart technologies in existing products or products in development.
Keywords: Smart Technologies, Evaluation of Effectiveness

1 Introduction

The main goal in development of an information system is to ensure conformity of the software with the initial system requirements. But it also should build a solid base for further information system's development, improvement and maintenance. One of the possibilities to establish long-term reliable software is to include additional support mechanisms for better usage, maintenance and enhancement already during the initial design and development phases of the information system.

The idea of smart technologies [1] bases on the vision about "clever" software that like living beings is able to "self-management". Hence the smart software is able to handle unpredictable events in unknown environments. Unlike external solutions providing specific supporting features from outside a smart technology conform software is able to react on external and internal events adequately by itself. The typical external events are changes in the infrastructure or changes in external systems; the typical internal events are changes in database structures or new versions of source code.

Smart technology oriented approach implies that the core functionality of software solution should be enhanced and integrated with several additional features supporting the usage, maintenance and further development of software.

The identified components of smart technologies are following:

- Business model-based system's functionality – the functionality of information system changes according to business processes described in the configuration.
- Built-in version management and data synchronization – automatic updating of software versions from the central server, including the conversion of data structures and data.
- Built-in self-testing – ability to check the internal integrity by automatic execution of test cases in the productive environment and to inform users and developers about detected inconsistencies.

- Environment testing – ability to check the external environment (options of operating system, data base management system etc.), to adapt itself to the specific environment and/ or to inform developers about detected inconsistencies.
- Data quality monitoring – ability to check the completeness and integrity of data stored in the database.
- Availability testing – monitoring of system availability using agent technologies; ability to inform developers remote about the status of the software and additional components needed for a correct functioning.
- Security and load testing – monitoring of system security using agent technologies; ability to provide monitoring of performance and load balancing.

The smart technologies include already practically verified components like self-testing [2], external environment testing [3], intelligent version management [4] but also other features that are topicality for further research.

The proposed concept of smart technologies is related to the concept of autonomous systems announced 2001 by IBM [5,6,7]. Both concepts set as a goal the increasing of software "intellect" to achieve a wide range of non-functional features – an ability to adapt itself to variable external events, self-recovering, self-optimizing etc. –, but the proposed technical solutions are different.

The decision about applying of smart technologies can be made not only during the initial design and development of an information system but also in later phases of the development or maintenance, e.g. within the new release of the software or as an separate project. But there should be a SWOT-analysis made to settle on the applying of smart technologies.

A number of criteria could be evaluated within the decision making to prove or to disapprove the usage of smart technologies in the concrete case. The goal of this paper is to present empirically indentified criteria for evaluation the effectiveness of applying smart technologies.

2 Criteria for Evaluation of Effectiveness

2.1 Effectiveness vs. Efficiency

The notion of effectiveness is explained as a "power to be effective; the quality of being able to bring about an effect" [8] or "the extent to which a program or service is meeting its stated goals and objectives" [9].

In contradiction to efficiency that describes the degree of achievement of definite parameter values and in fact is close to the notion of productivity, the effectiveness is an ability of an object or a process to fulfill predefined goals under definite conditions.

The evaluation is described as a "systematic determination of merit, worth, and significance of something or someone using criteria against a set of standards" [10].

The most precise form of evaluation is measurement, but it is not the only way to evaluate something. The measurement is "the process of assigning a number to an attribute (or phenomenon) according to a rule or set of rules" [11].

So there can be identified the following preconditions to evaluate the effectiveness:

- Choice of criteria characteristic for effectiveness;

- Identifying of optimal values for criteria (etalon values);
- Description of evaluation (measurement) process.

The next chapters outline a proposal for different criteria inferred from the practical evidence collected in real software development projects. Identifying of optimal values for criteria and description of an exact evaluation process to have a basis for decision making for or against usage of smart technologies in information systems are a subject for further research.

2.2 Economical Criterion: Return on Investment

The prime task in every commercial project is to maximize the potential return on investment. Every rational market participant will be ready to invest in a project only in that case if the investments in closer or further future could bring a profit [12].

Hence the first criterion proposed for the evaluation of effectiveness is a prospective return on investment (ROI). An estimation of ROI gives an approach to measure the investment potential by comparing the expected gains to investment costs. The value of this variable can be calculated on the basis of an investment return plan containing a comparison of an initial investment and discounted future cash flows: ROI is the ratio of average prospective cash flow over the period of the project divided by the initial investment outlay.

The commonly used effort, cost and schedule estimating methods (e.g. COCOMO [13]) can be employed for the estimation of an initial investment. But the following adjustments should be taken into account:

- Previous experience and efforts in implementing of smart technologies can reduce the necessary initial investments; it is rather possible that already once developed smart technology components will be adaptable for similar tasks than if the components are to be developed from scratch.
- Modularity of the product to be prepared with smart technologies: the more modular is the software product the easier is to add to it “just” one more module.
- The available time schedule should be comfortable enough to implement and/or integrate smart technology features besides the basic functionality desired by the customer explicitly.
- Feedback and level of user tolerance show whether the concrete users will be able and ready to tolerate short-term inconveniences if they would arise and take an active part in the improvement of the product features. Not all customers are happy to be involved in “experiments”. Doubtless more open for innovations in their projects are long-standing customers with high allegiance to the supplier.

For estimation of prospective cash flows can be used both bottom-up and top-down approaches. The bottom-up approach estimates the minimal savings of costs/ expenses (could be achieved by enhancing the software product (total projected savings). The typical examples are transport and personnel costs that were spent for travelling to end users as well as installation materials, hardcopy of documentation, partly user support resources etc. It is some kind of “production-oriented” way to find the break-even-point where the initial investment becomes profitable.

The top-down approach is based on the estimation of the market opportunities, i.e. the achievable prices and sales volume is evaluated to have a view of feasible gains of smart technologies.

The simultaneous usage of both approaches returns the gap where the prospective cash flows should fit in to be realistic.

When both components – the initial investment and prospective cash flows – are obtained, they should be made comparable. The usual way to do it is a discounting of future cash flows with the rate of interest.

The discounted present value (for one cash flow in one future period) is expressed as:

$$DPV = \frac{FV}{(1+i)^n} = FV(1-d)^n\,, \tag{1}$$

where

- DPV is the discounted present value of the future cash flow (FV), or FV adjusted for the delay in receipt;
- FV is the nominal value of a cash flow amount in a future period;
- i is the interest rate, which reflects the cost of tying up capital and may also allow for the risk that the payment may not be received in full;
- d is the discount rate, which is $i/(1+i)$, ie the interest rate expressed as a deduction at the beginning of the year instead of an addition at the end of the year;
- n is the time in years before the future cash flow occurs. [14]

Incorporation of smart technologies into software products is beneficial if the initial investment is less than the present net value of prospective cash flows.

But there should never be forgotten that the calculations are based on the estimated (not real!) values. In particular the effort estimation methods can be very inaccurate in some cases [15].

2.3 Quality Criteria

The quality of a product or service is intrinsically linked with reputation and hence is crucial to organisation's sustainability. Considerations on quality assurance could enhance integration of smart technologies into strategically important products, even if economic returns on such investments can be expected only in long-term.

The quality achieved by implementing smart technologies can hardly be measured. Prevailing practice in quality evaluation is based on experience values acquired during similar projects.

One of the possible quality criteria is an efficiency of customer support. This variable shows in what extent the level of client support could be improved if the software product is equipped with smart technologies by the same number of support staff. The efficiency of customer support can be qualitatively measured by interviewing clients and analyzing selected users' messages, etc. The quantitative indicators can include number of supported users (number of processed cases) in a certain period of time or average proceeding time of problem processing: Either more clients should receive support, or service quality for existing clients should be increased by the given number of support resources.

Alternatively there could be estimated if the existing customer support level can be assured by reduced amount of support staff. If implementation of smart technologies has not resulted in increased number of clients, still the effectiveness is positive due to fewer resources used to provide more qualitative support to the same number of clients.

Another possible criterion for evaluation of the effectiveness of smart technologies is efficiency of maintenance. As the developers consider software maintenance as a very significant area for applying smart technologies this criterion would show if the quality of software equipped with smart technologies is increased. Quick problem detecting, short problem solving time, low number of user reported errors – all these are quantifiable variables indicating high quality of maintenance and resulting in clients' satisfaction.

2.4 Organizational and Marketing Criteria

Unlike the previously described criteria targeted at profit and reputation, the organizational criteria deal with developers' organizational and technical ability to carry out the development and the necessary integration of smart technologies into end-products.

The development of smart technologies is not possible without highly professional system architect and skilful (able) developers. On the one hand, the development of smart technologies requires good academic knowledge and professional background, but on the other hand, such development also requires profound understanding of product's architecture and functionality. During the development issues such as work-discipline, rigorous following the guidelines and avoiding the use of specific techniques that could cumber maintenance should be considered.

These criteria are hardly quantifiable; therefore evaluation of these criteria is preferably done by experienced architects and developers.

Since implementation of smart technologies contribute to significant changes in software construction, potential error probability is not excluded; therefore a well-designed crisis management plan should exist from organizational point of view. Strict and prudent managers possessing power of persuasion are the best support in such cases. The developed end-product should undergo scrupulous testing, therefore appropriate technical support including virtual machines and powerful central server should be available.

The main goal of applying smart technologies is maximizing of profits expected in long-term. Reduced expenses or improved quality of software – the customer should have a clear concept of an added value. The task of marketing activities is to promote the emotional and practical adjustment of the customer to the improved product, to motivate the user to be familiar with the new features of the product. So, the decision about (for or against) smart technologies means also the answering the question if there is a marketing power in the enterprise that is able to "make money" from the technological innovation.

3 Empirical Data Sources

Principles of smart technologies are partially implemented and approved in software development and maintenance projects. Although, in neither of software products smart

technologies were implemented to the full extent, the gathered experience shows that usage of even only several features or narrowed version of features can bring an additional effectiveness and economical benefit.

The criteria for evaluation of effectiveness briefly described in the previous chapter are all identified in concrete development projects where particular smart technology features were added to existing information systems supplementary.

One of the future research tasks is the gathering of statistics to demonstrate whether and how the usage of smart technologies improves the quality and maintainability of software. The development projects serving as a source for the actual research are shortly described in the next chapters.

3.1 Version Management and Environment Checking Components

The Project A was started 2002, and the main task of it was to create a finance management information system according to system requirements formulated by a ministry in Latvia. The supplier had to develop software for structured gathering of financial information on several hierarchal levels.

Already shortly after the first release of the solution new customers accrued from the importance of the solution's functionality in the public sector. The number of users grew up from ca 100 in January 2003 to ca 500 in January 2004. In the face of continuous increasing demand the information system was transferred to a standardised software product purchasable in three different versions (lite, standard, advanced) and two separately usable functional models.

At the end of 2006 the solution in its different completions was used by ca. 1500 users. The system was supposed to run in over than 600 public offices located throughout the territory of Latvia ensuring regular gathering of information by different time periods. Furthermore, the specific requirements dictated that in certain periods the most users will use the system simultaneously. Since it was required that the system should be able to run not only in offices provided with internet connection but also in offices with irregular and instable internet connection or even in offices without it, the system architecture was based on client-server application principles with central data base and local desktop applications that were able to run either in online mode if internet was available, or offline in cases of internet absence.

In view of a large amount of users and their geographical location as well as very restrictive agreements with customers and limited resources allocated for installation and maintenance services, developers made a decision to enhance the software product with one of the smart technology features – the automated version management including both installing and updating. In approximately a year the developers added to the software product one more smart technology component – the automated environment checking including a comprehensive messaging mechanism.

As a result an improved version of the software product was obtained ensuring completely automated remote installing and updating of software versions in different target environments (different computer systems with various operating systems, office packages etc.).

The automated feature for checking of external environment was developed as an independent but configurable module providing the "spying" (pre-checking of target

environment) and informing functions necessary for detecting the adequacy of the target environment's parameters to the minimal technical values.

The acquired result exceeded the expectancy: The features of automated version updating and automated environment checking have been significantly lightened the processes of software installation and maintenance as well as reduced the consultant resources necessary for the user support.

The following conclusions were captured within the project:

1. Adding of smart technologies to the software after the development is useful though requires more resources than including smart technology already in the software architecture design phase.
2. Implementation of smart technology principle in software takes fewer resources than full-range configuration support. In the same time smart technology places fewer constraints on the acceptable means of expression.
3. Smart technologies allow reducing the efforts (time, resources) for software testing and setting up, thus increasing the client service level significantly.
4. Smart technologies assist to provide software performance in a changing environment and environment containing heterogeneous platforms and infrastructure. Nevertheless mechanisms of smart technologies need regular adaptation to the environment changes.
5. It is very important to provide in-depth messaging mechanism to inform the developers about indicated problems in time.
6. Even the mechanisms making the users work easier should be appropriate explained to end users. The absence of a human in the set-up process of software is in discrepancy with users' former experience.

3.2 Version Management and Data Synchronisation Components

The Project B was started 2008, and the main task was to create a completely new version of an over 10 years old software product (to rewrite it). As the product had over 50 installations in various places of Latvia, there was already during the design phase decided to use the successful experience from the Project A and to include features of smart technologies into the new version of the product.

As enhancement was automated version management planed. During the detailed analysis there was an additional important smart technology feature – advanced data synchronising – identified. This feature was necessary for the effective distribution of configuring and classifying information from the central database to local bases and for collecting of data entered by users from local data bases to the central base.

The modernisation, including the complete recoding of the core functionality, took approximately eight man-months. About 50% of the effort was spent for adaptation of data synchronisation module (a sketch of the feature was already there) to the specific requirements. The product was successfully completed, and it is being offered in the market now.

The implemented solution of automated initial installation of the product was not as successful as originally desired. Due to specifics of the used commercial database

management system there was quite often assistance by the developers necessary to ensure, the initial installation will be performed successfully.

However the automated updating of version including the data synchronisation worked outstanding; it saved resources spent for user support and reduced effort necessary for software maintenance indeed.

The project results analysis passed after the project's end showed the main reason for the waste of resources – the chosen technical solution of data synchronisation.

The version installing and updating in the Project A was managed according to a specific configuration file containing statements for controls and setup and the software solution was able to interpret these statements adequately. The Project B had another technical strategy for data synchronization management - the control flow statements were stored in the database.

The following conclusions were captured within the project:

1. The moment of decision-making about enhancing the information system with the features of smart technology is less important than the concrete technical solution.
2. Implementing the functions of smart technologies it is advisable to use proven and robust architectural and technical solutions – it will reduce the number of problems to be solved simultaneously and improve the controllability of the development process.
3. Although the clients approve opportunities provided by the smart technologies, usually, they are not willing to provide additional financial means to ensure them.
4. The smart technologies are certainly cost-effective and should pay-off in long-term business projects and long-term cooperation with notable number of users and differently configured users' workstations.

4 Conclusions

The following conclusions result from the practical experience described above:

- The prospective effect from the applying of smart technologies is considerable; it ranges over client service quality up to increasing of staff qualification.
- Even usage of single or several smart technology features can be profitable.
- Economical aspects are the main barrier for wider spread of smart technologies.
- Only very qualified and skilled suppliers are able to fulfil the set of preconditions necessary to ensure the successful usage of smart technologies in commercial projects.
- Other hard evaluable factors like reputation of enterprise, marketing/sales capacity, user support level etc. also impact the final decision about the applying of smart technologies.

References

1. Bičevska, Z., Bičevskis, J.: Smart Technologies in Software Life Cycle. In: Münch J., Abrahamsson P. (eds.) Product-Focused Software Process Improvement. 8th International Conference, PROFES 2007. LNCS, vol. 4589, pp. 262–272. Springer, Heidelberg (2007)

2. Bičevska, Z., Bičevskis, J.: Self-testing: A New Testing Approach. In: Haav, H.-M., Kalja, A. (eds.) Proceedings of the 8th International Baltic Conference on Databases and Information Systems (Baltic DB&IS'2008), Tallin, pp. 179–189. (2008)
3. Rauhvargers, K., Bicevskis, J.: Environment Testing Enabled Software – a Step Towards Execution Context Awareness. In: Haav, H.-M., Kalja, A. (eds.): Databases and Information Systems, Selected Papers from the 8th International Baltic Conference, IOS Press, vol. 187, pp. 169–179 (2009)
4. Bičevska, Z., Bičevskis, J.: Applying of Smart Technologies in Software Development: Automated Version Updating. In: Scientific Papers University of Latvia. Computer Science and Information Technologies, vol. 733, pp. 24–37 (2008)
5. Ganek, A.G., Corbi, T.A.: The Dawning of the Autonomic Computing Era, IBM Systems Journal 42(1), 5–18 (2003)
6. Sterritt, R., Bustard, D.: Towards an Autonomic Computing Environment. In: Proceedings of the 14th International Workshop on Database and Expert Systems Applications, pp. 694–698 (2003)
7. Lightstone, S.: Foundations of Autonomic Computing Development. In: Proceedings of the 4th IEEE international Workshop on Engineering of Autonomic and Autonomous Systems, pp. 163-171 (2007)
8. The Free Dictionary – Effectiveness, http://www.thefreedictionary.com/effectiveness, accessed August 25, 2009
9. http://www.scoea.bc.ca/glossary2001.htm, accessed August 25, 2009
10. Wikipedia, http://en.wikipedia.org/wiki/Evaluation, accessed August 25, 2009
11. http://www.absoluteastronomy.com/topics/Quantitative, accessed August 25, 2009
12. Sullivan, A., Sheffrin, S.M.: Economics: Principles in action, Pearson Prentice Hall (2003)
13. Boehm, B. et al.: Software Cost Estimation with COCOMO II (with CD-ROM). Englewood Cliffs, NJ: Prentice-Hall (2000)
14. Wikipedia, http://en.wikipedia.org/wiki/Discounted_Cash_Flow, accessed August 26, 2009
15. Schach, S.R.: Object-oriented and Classical Software Engineering. McGraw-Hill Professional (2004)