Ergonomic Aspects of Software Engineering

Andrzej Borucki

IME , Poznan University of Technology, Poznan, Poland
andrzej.borucki@put.poznan.pl

Abstract. The practice of applying rigid software development rules has killed creativity as processes and tools take precedence over technical solutions and client satisfaction. A key role in developing software is played by the intellectual resources of project teams, i.e. their knowledge. The knowledge used to produce software may be either open or hidden. In order to manage software development effectively, advantage needs to be taken of the knowledge held by each design team member. Two distinctive knowledge management strategies are available for managing software development. These are the knowledge codification strategy and the knowledge personalization strategy. The knowledge codification strategy requires the use of expensive technology to apply CASE tools. Based on two decades of experience in managing IT projects, we have grown somewhat critical of the use of CASE tools. Our experience in managing IT projects shows that a strategy of knowledge personalization in software development helps improve designer knowledge in a given business field and boosts their productivity.

Keywords: Project Management, Software Engineering, Reguirement Engineering.

1 Introduction

The prime purpose of software development ergonomics is to ensure the best possible working conditions for programmers while minimizing work onerousness and maximizing worker productivity. A study of the effectiveness displayed by our team1 on software development projects performed between 2000 and 2013 shows that, in project management practice, excessive emphasis is commonly placed on the technical aspects of project management while neglecting the innovation side. We have observed that the use of standard software development procedures does not make a project more cost efficient nor speed up the design process. What it often does, however, is keep project team members from devoting their time to forging friendly relationships with one another. Practice has shown that mutual relations among designers are pivotal for creating an environment which will unlock the intellectual potential of individual project team members and inspire the team's innovative spirit.

[1] Since 1990, during his employment in PPH Softmar, the author has led numerous deployments of management information systems. Between 1990 and 2001, he and his team developed and unrolled dozens of proprietary applications in logistics and finance.

C. Stephanidis and M. Antona (Eds.): UAHCI/HCII 2014, Part I, LNCS 8513, pp. 95–103, 2014.

Every innovation process requires positive interaction between the authors of ideas and their protectors responsible for approving a new solution and incorporating it into the project. The innovations achieved by project teams should help shift project focus from effectiveness and efficiency to innovation. Therefore, software design managers need to recognize the influence of non-environmental factors on designer productivity and select management methods and techniques which will minimize the arduousness of intellectual work and bring out the innovation potential of the design team.

The non-environmental factors which in our view are the most critical for ensuring ergonomically-sound software development and which remain within the discretion of project managers include project-specific knowledge management strategies, software development models applied in the design process and project and team management methods. By and large, the classic software development ergonomics is a study of the impact of environmental factors on the onerousness of intellectual work (including the influence of lighting, noise, screen radiation and the ergonomics of CASE tool interfaces) which downplays or fully ignores the non-environmental impacts that relate to the human factor and tie closely to the quality and innovative value of given software.

2 Ergonomics of Software Development vs. Productivity

Today's software engineering should provide efficient software design tools and define the boundary conditions to govern their use. To that end, developers need to ascertain:

- Whether a unified approach to software development does not kill designer individuality and consequently reduce designer productivity and innovation;
- Whether the things of importance for increasing the overall efficiency of project team management do not reduce the effectiveness of individual designers;
- The extent to which the stress experienced by designers during software development affects their efficiency and innovation;
- The reasons why project managers seeking to meet client expectations neglect the personal intellectual potential of individual designers.

Experience with a great number of IT projects which varied in the degree of their innovation and complexity points to the following key factors which undermine the productivity of designer teams (Borucki,2012),(Cushman,Rosenberg,1991):

1. The stress experienced by designers in developing and implementing IT projects results from:

- Their post-go-live accountability for software defects;
- Their inability to process a flood of input data supplied during software development or having to complete their assignments based on incomplete information;
- Having to cope with gaps in knowledge, especially technological, needed to complete their project,
- Changes to software requirements made in mid-project, especially by the prospective user.

2. Mismanagement of the intellectual potential of individual project team members;
3. Excessive reliance on CASE tools to aid design at the expense of pure project innovation;
4. Adopting the regime of a specific IT project management method;
5. The pressure of having to meet project deadlines and budgets after the labor intensity of a project has been underestimated;
6. Having adopted a knowledge management strategy which fails to fit project profile.

Developers frequently forget the simple fact that, as individuals, employees differ in their personalities and intellectual potential. Little is done to harness their talent for project purposes to allow staff members to derive personal satisfaction from their work (Marco ,Lisner,2002). In seeking to achieve project goals and meet deadlines (especially where clients allow too little time for document preparation), it is often believed that good and efficient management is all about pushing employees to work harder with little regard for their off time. Where the entire focus is placed squarely on the technical aspects of project management, very little time remains for finding the best way to complete a given project task, especially where standard operating procedures are followed. Having watched a number of designers at work over the last two decades, I have seen them dedicate their time overwhelmingly to performing specific project tasks while giving little attention to identifying the best possible solution. Most designers had to be "forced" by company management to read designer books or attend conferences and training. Our observations show also that, regrettably, the present design practice is geared towards getting designers to become as "efficient" as possible and to making entire projects highly effective. This makes design work extremely strenuous and often described as "the dark side of intellectual work". The reason for this lies in underestimating the importance in the design process of the human factor and, in particular, people's intellectual and psychological dimensions. Figure 1 shows three key non-environmental factors which affect the ergonomics of software development and which can be influenced by team manager.

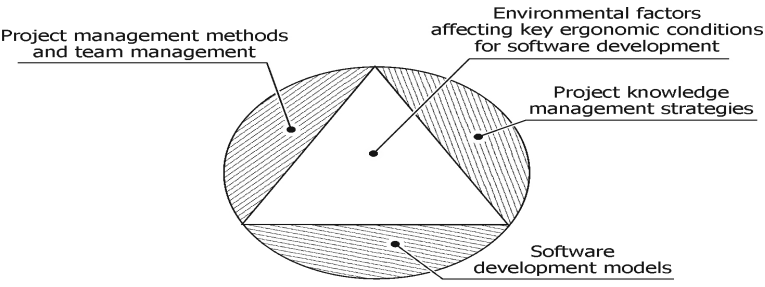

Fig. 1. Factors affecting key ergonomic conditions for software development

In searching for ways to make project teams more productive, the majority of project managers choose to use various CASE support techniques in design work and project management. Such tools are viewed as facilitators for the efficient development of complex IT systems which are more functional, reliable and efficient and allow for easier maintenance and system transferability.

The key advantages of using CASE tools to improve software design efficiency are:

- The option of modeling systems at various levels of abstraction,
- The capability to produce project documentation in a uniform format at each stage of design,
- Easier communication between project developers and the future software user,
- Facilitated two-way (vertical and horizontal) communication between designers,
- Access to knowledge from other projects for the purposes of the new undertaking,
- Reliance on standard system development methods,
- The capability to use uniform methods to evaluate project quality,
- A knowledge repository created to support future projects – the repository is a database of specifications to be used to assess the labor-intensity of projects, developers, design items, diagrams, algorithms, etc.,
- Standard formats for input and output data,
- An ability to rapidly produce a user interface and automatically generate software source codes based on diagram designs created in e.g. the UML language.

The majority of CASE functionalities are designed to help gather knowledge on the project at hand, create a knowledge-base for future reference, build a project management model and develop a knowledge and team management strategy. Our 20-years' experience in managing IT projects suggests there is a pressing need for revising the way CASE tools are being used. Despite their widespread application, such tools do little to keep developers from exceeding deadlines or overspending budgets. The key reason for that lies in poorly formulated knowledge management strategies and excessive reliance on CASE tools in software design. Such problems are particularly pronounced in innovative projects where designing is non-linear and involves a sequence of fixed steps performed up to a certain point in the process after which the developers go into feedback loops not envisioned in the initial project model. In non-linear design, the structure of a software development project may be unknown at the outset. The adverse impact of heavy reliance on CASE tools is seen in designing applications on the basis of various information technologies.

The knowledge which has been acquired tends to be used to unify software development procedures rather than create the kinds of ergonomic conditions for software development that are certain to bring out innovation.

3 Information Stress – New Challenges for Humanizing the Work of Software Designers

Contemporary software engineers employ a range of CASE tools designed primarily to create knowledge repositories for gathering and storing the information needed in software design. In designing IT systems, use is made of basically 10 areas of knowledge which are software configuration management, logic software model structure

management, software design management, software engineering infrastructure, software engineering management, testing, requirements analysis, software quality analysis, software evolution & maintenance and software engineering. Once obtained, information about current project and any relevant past projects may "come in handy" in current design work. Such information comprises powerful databanks which should be accessible to designers on an equal basis. Figure 2 illustrates a sample complex knowledge repository used in software design. Repositories of non-confidential knowledge hold information of high as well as low relevance for software design.

Many designers become overloaded with various types of data. Without proper selectiveness, developers may end up being overloaded with data and subjected to what literature refers to as "information stress". Commonly, information stress results from the inability of employees to process excessive input data supplied to them in the course of software development or from not having enough information to complete their work. At the start of the design process, it is difficult to tell which information will be of use and value. Information stress results from the discord between the volume of information provided during software development and the individual processing capabilities of the designers. Such stress can be seen to affect developers with varying intensity at all stages of software design and deployment – it affects all persons involved in creating software as well as its would-be users (who are also involved in the design process). Recent literature provides articles on intellectual work ergonomics and specifically on the impact of information overflow on the efficiency of intellectual work (Ledzińska, 2011),(Hankała, 2009). The severity of the impact of information overload depends on a personality trait which the literature refers to as "the cognitive drive" (Ledzińska, 2011). A number of factors determine an individual's sensitivity to information stress. Such factors include inquisitiveness manifesting itself as hunger for knowledge, temperament, personality and rational intelligence. Information stress may be experienced at various stages of information processing. One of the primarily responsibilities of software design management is to process information at the stages at which information is identified, analyzed and used. Excessive or insufficient information at various design stages may affect the severity of the stress experienced by designers.

Stress may also harm the working efficiency of developers who are bombarded daily with aggressive information. The resulting information glut may impair their ability to process and select information causing them to make misguided judgments. Information stress may distort cognition and result in delivering products based on false or incomplete assumptions. In extreme cases, it may contribute to project failures.

Our observations of dozens of software development projects suggests that designers experience the most severe information stress while identifying and examining functional software requirements. At this stage, conflicts abound due to excessive functional demands made by various groups of would-be users as well as functions left out by designers and the system functionalities unnoticed by future software users. Such circumstances are illustrated in Figure 3.

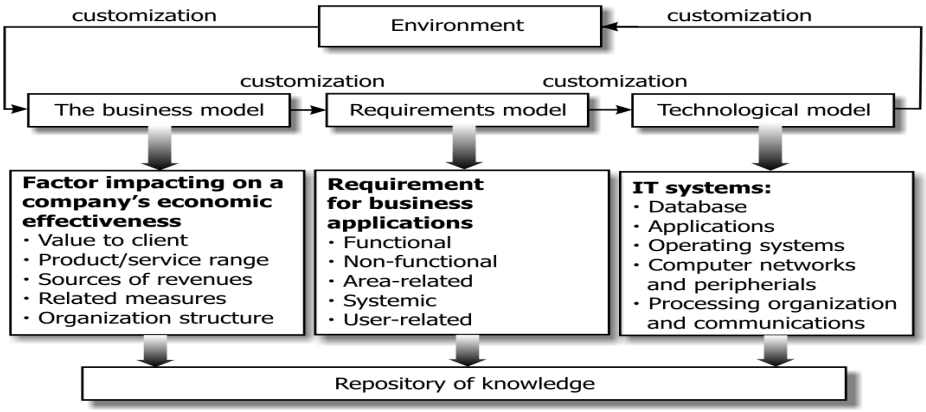

Fig. 2. Elements of knowledge gathered in software project repository

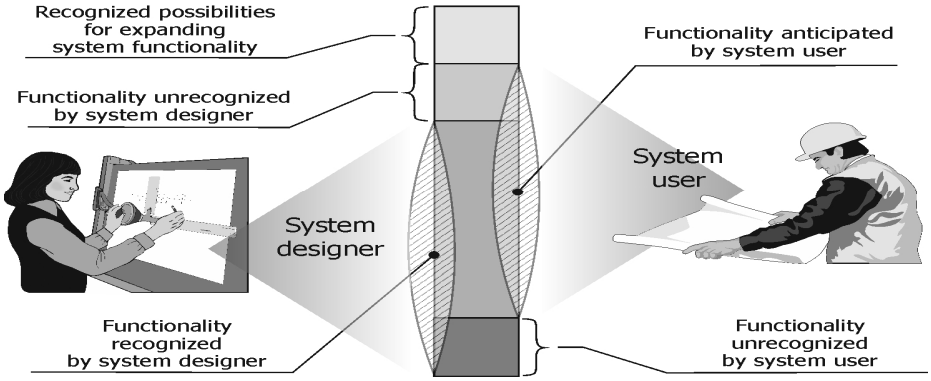

Fig. 3. Areas of knowledge about the functionalities of software in development

The simultaneous occurrence of information gluts and information deficiencies, as faced by software designers while defining software requirements, is caused additionally by the fact that prospective software users define their software functionality demands very generally or express functional requirements in their industry jargon (the so called "tribal language"). This makes it difficult to identify the precise requirements and assign priorities to the individual demands in the design process. A great number of the requirements submitted by various individuals are mutually exclusive. In designing large systems in projects which go on for months on end, requirements concerning specific functionalities may change considerably over time. This may be due to the persons responsible for selecting software functionalities changing their minds, changes in the effectiveness factors applied on the business parts of IT architecture for which a given software package is being developed and changes in the legal environment of a given enterprise.

One way to arrange knowledge into an orderly structure is to divide it into the categories of basic, advanced and innovative (Ashok, 2004). Another is to define a hierarchy of importance of the knowledge to be used in project implementation. Depending on project size and complexity and the degree of innovation involved, a layer may be set aside in the knowledge repository for exclusive knowledge with the most critical information, particularly that related to the innovative technologies to be employed in the design process. Other layers would contain knowledge generally accessible to all designers as well as knowledge on the CASE methods and programs applied in the project (Borucki,2012). Needless to say, a design firm derives most of its competitive edge from the exclusive knowledge concerning innovation. A crucial consideration in creating a knowledge repository is also how the knowledge is stored and accessed by individual project developers. My experience with various projects shows that much designer stress comes from not having the knowledge needed to solve a specific problem and being aware of gaps in one's knowledge, especially where the knowledge that is missing is essential for completing a project task.

4 Project Management Methods Mitigating Adverse Stress Impact

An overview of software development management methods shows that many traditional solutions ignore the significance of designer productivity, or the so called human factor, in its intellectual, psychological and sociological aspects. The fact of the matter, however, is that it is actually essential for project managers to recognize not only technical and economic considerations but also the soft factors. Such managers should therefore (Martin,2008),(Highsmith,2007):

1. Secure individual and collective capabilities to develop software;
2. Minimize the impact of non-environmental factors which undermine the capacity to develop software: such factors include employee stress, lack of motivation and heavy reliance on "rigid" software development methodologies;
3. Retain key personnel with critical skills on project teams;
4. Optimally select the environmental factors which enhance software development capabilities;
5. Ensure software is developed with agility, i.e. keep the designers and software users working at a swift pace;
6. Communicate osmotically to allow free circulation of information within project teams with equal access to project information ensured for all team members;
7. Ensure all staff members feel secure and free to express their opinions on project-related matters;
8. Select the right project team members with experience in the field in question to staff short- and long-term projects;
9. Flexibly apply technical support and CASE tools in software production so as not to constrain designers' individuality.

In order to satisfy the need for recognizing the human factor, new “agile” project management methods have been developed. The methods satisfy the criteria of being ergonomically sound as they rest on the assumption that project success comes from successfully utilizing the human factor and creatively engaging with clients at each stage of project implementation. The software projects which rely on agile methodologies are easier to accept for future users which in itself reduces the risk of project failure. In any enterprise, IT infrastructure is only optimal in real life when its applications can be adjusted rapidly to new business challenges and fit well into the enterprise management concept. This is made considerably easier with the arrival of agile methodologies in IT system design, especially where it is essential that the system is customized to a company's business model with as little harm as possible and without interrupting the continuity of IT processes.

5 Knowledge Personalization Strategy as a Key to Humanizing Project Work

A key role in software development is played by the knowledge resources that, if used competently, have the potential of boosting project innovativeness. Both covert and overt knowledge should be used optimally at each stage of system design. An overview of successful IT projects in recent years, particularly those relying on internet technologies, shows that a key to their success is to employ a proper knowledge management strategy which helps bring out and effectively utilize the intellectual potential of individual design team members. Interestingly, in the majority of the reviewed cases, the use of knowledge management strategies followed the collapse of markets for previously produced products.

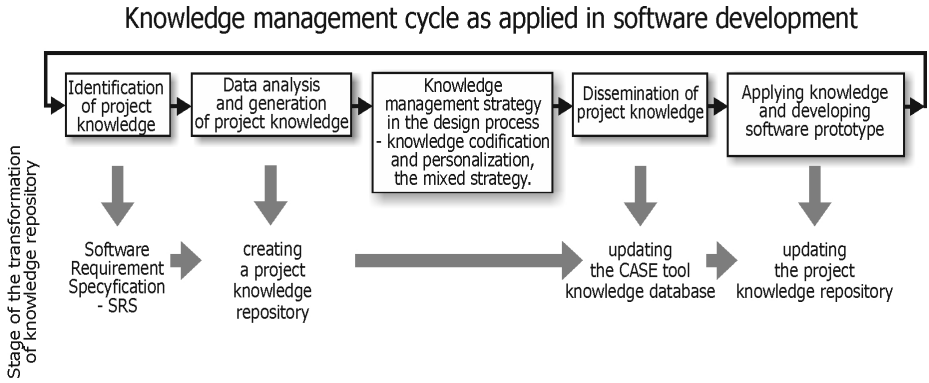

Fig. 4. Knowledge management cycle as applied in software development.

Many companies which faced such tribulations have been forced to redefine themselves on the market and adopt new competitiveness strategies to unlock their potential for product innovation. Figure 4 presents the knowledge management cycle as applied in software development.

6 Conclusions

Design firms churn out ever more sophisticated software systems which they are expected to deliver within ever shorter lead times by relying on continuous progress in software engineering. Unfortunately, the practice, at least that seen on innovative projects, is plagued by randomness and uncertainty of schedules and costs. Design firms look for effective ways to increase the productivity of their designers and entire design teams. In doing so, they usually consider the two options of either increasing process efficiency, which involves a strategy of knowledge codification, or developing innovative project solutions, possibly by means of a knowledge personalization strategy.

Our observations from numerous projects confirm that the choice of a path for knowledge personalization to support design innovation requires the use of ergonomic software development methods and properly managing the overall software design process.

References

1. Ashok, J.: Knowledge Management: An Integrated Approach. Peearson Education Limited (2004)
2. Borucki, A.: Factors Adversely Affected the Productivity of Software Designers Applying CASE Tools. In: Vink, P. (ed.) Advances Social and Organizational Factors, pp. 317–329. CRC Press Taylor & Francis Group (2012)
3. Cushman, W.H., Rosenberg, D.J.: Human Factors in Product Design. Elsevier, Amsterdam (1991)
4. Hankała, A.: Aktywność umysłu w procesach wydobywania informacji pamięciowych. Wydawnictwo Uniwersytetu Warszawskiego, Warsaw (2009)
5. Heeg, F.J., Shrader, M., Scgrader, S.: Analyse und Neugestaltung betrieblicher DV-Systeme unter besonderen Beucksichtingung Software-ergonomicher Kriterien. In: Schonplug, W. (ed.) w Software-Ergonomie 1987: Nutzen Informatiuunssytems dem Benutzer, Tanbuer, Berlin, pp. 440–453 (1987)
6. Highsmith, J.: APM: Agile Project Management, PWN (2007)
7. ISO/IEC-15939-Software engineerng-Software measurement process, Reference number
8. ISO 9241-100 Ergonomies of human system interraction - Introduction to Software ergonomics
9. ISO 9241-129 Ergonomics of human system interraction-Guidance on invidualization
10. ISO 9241-20 Ergonomic of Human system interaction-Accessibility quideline for Information communication equipment and service-General quidelines (2008)
11. Marco, T., Lister, T.: Czynnik ludzki, skuteczne przedsięwzięcie i wydajne zaspoły. WNT, Warszawa (2002)
12. Martin, R.C., Martin, M.: Agile. Programowanie zwinne. Zasady, wzorce, praktyki zwinnego wytwarzania oprogramowania w C#, Wydawnictwo Helion (2008)
13. Ledzińska, M.: Człowiek współczesny w obliczu stresu informacyjnego. Wydawnictwo Psychologii PAN, Warsaw (2008)
14. Ledzińska, M.: Nowe czasy-nowe źródła stresu w pracy. In: Juliszewski, T. (ed.) Obciążenie psychiczną pracą-nowe wyzwania dla ergonomii. Komitet Ergonomii PAN, Kraków (2011)