Key Success Factors in Personnel Motivating Projects

DENISS KUMLANDER
Department of Informatics
Tallinn University of Technology
Raja St.15, 12617 Tallinn
ESTONIA
http://www.kumlander.eu

Abstract: - The number of projects' failures is permanently increasing despite of all modern attempts to keep projects under control. Dependable software development has become an important technique to move towards in order to improve the situation. A key success issue on this road is developing personnel into a team of highly professional, loyal and attached to the organisation people. Unfortunately the software industry is a highly technological sector with a shortness of personnel resources in many countries. There are many examples when employees do leave companies despite all common motivating approaches used in the organisation. The paper proposes establishing special, so called motivating, projects to address employees' needs to develop themselves by learning something new that can be challenging for them. At the same time it is important to understand all hidden dangers of running such projects in order to achieve the defined goal since otherwise employees could be rather unmotivated, stressed and even decide to leave the company. Fortunately some fairly simple approaches and methods described in this article allow ensuring success of the motivating projects and let focus on the true goal on the organisation evolution road.

Key-Words: - Dependable software, personnel motivating, software engineering, key success factors

1 Introduction

The number of projects failures is increasing from year to year. Those projects' costs are carried by software vendors' customers either directly or indirectly increasing the cost of developed software sufficiently. A situation with "successful" projects is not much better: only 20% of functionality (in average) is used "often" or "always" and 16% "sometimes". The remaining 64% is either never used or used just occasionally [5]. In recent years, dependability - an integrative concept comprising such criteria and sub-criteria as reliability, security, continuous development, availability, safety [1, 10] - has become to be very important. Here we arrive to another topic of this article. Nowadays increasing competition between software vendors and much more demanding markets force companies to stabilize their productivity and improve their development process in all respects [6]. It is not possible to develop really dependable, i.e. reliable and highly available software systems, without having highly professional and motivated personnel. Unfortunately the software industry is a highly technological sector [11] with a shortness of personnel resources in many countries. Therefore it is important to ensure that company employees will not leave the company. This task is not as easy as it looks like. Professionals are still migrating from one company to another despite all common motivating practices proposed so far [2, 3, 4], like good salaries, friendly working environment, social benefits etc. It happens mainly because workers are looking for something more challenging, bored to do the same work using the same tools or stacked in their self development at the current workplace. It many cases they would like to use other, modern methodologies and techniques to move themselves forward, ensure their future and try something else. Therefore, it is important to address such technological needs in addition to common motivating approaches.

The core idea of personnel motivating software development is to establish a special project (or a set of projects) that can motivate the organizational personnel by allowing them to learn something new, use new technologies and practices doing a useful work for the company [7, 9].

The article proposes to use personal motivating projects as a preparation step on the way toward developing dependable software. Here we are going to review personal motivating projects and identify key success factors for those. The article is mainly based on our recent experiences obtained from such projects. We review common mistakes and problems that often occur and describe how to avoid or address those.

The paper is organised as follows. The section 2 describes what personnel motivating projects are, why those are run and how to select participants.

The next section demonstrates common problems and mistakes manager do in motivating projects. The 4th section discusses approaches for addressing previously described problems. The last section concludes the paper.

2 Motivating Projects

This chapter provides guidelines on why and how to start motivating projects. This part of work is mostly produced using our previous articles [7, 8, 9] and serves as an already learned material, which is a starting point for the following study.

2.1 Reasons to establish

The personnel motivating projects are usually run in the following cases:

- The organization would like to encourage employees to learn something new that will be useful for the organization in the (nearest) future providing them with corresponding opportunities within a dedicated project. There can be different reasons for such organizational wish:
  1. The organization has vacancies that are easier to fill using internal resources than external. For example, if the position a worker to be moved from can be filled easier that the position he is going to be motivated for. Besides, a person having a lot of knowledge of organization processes and products can be more interesting for the company in that certain position that an outside (new) person;
  2. The organization would like to cover new areas (markets, customers) by a product using newer technologies;

- The developers encourage an organization to move to another methodology/technique etc. demonstrating a list of advantages and the organization is willing to accept this migration if those advantages will be shown during a pilot project.

It is important to notice into addition to the previously formulated list of cases that any organization reaches its operational and strategic goals mainly via their employees work. Therefore the motivating project should not be concentrated just on providing benefits for employees, but should be motivated by the organizational needs (short and long-term) otherwise the organization resources are spent improperly. Moreover, the personnel participating in the project is not really motivated as nobody is interested in accepting, reviewing and continuing such projects as it produces a feeling that the done work is just waste of time and the company doesn't have anything to work on.

One of the most valuable reasons for companies to run motivating projects is a goal to prepare personnel for implementing dependable software, i.e. highly reliable, available and supporting continuous development in the future. Such valuable and important goal requires more than just implementing software using other standards, infrastructure or methodologies. It requires highly skilled and motivated personnel.

2.2 Team

The personnel of any company are not a set of totally identical persons. Therefore they can be partitioned by their *ability to learn* something new into the following groups:

- Innovating;
- Slowly changing / slowly learning;
- Static [7, 8].

It is advisable to select members of a forming team for the pilot project among persons belonging either to the first (the best choice) or the second group. Notice that any team can be divided by functionality into a core and supporting groups. Members of the supporting sub-teams is not necessarily learning or doing something new during the project, so those can be gathered among slowly learning employees or even static. It is possible to use two additional factors considering potential participants of the core sub-team, which are presented below.

A person's *learning factor* can be calculated by multiplying his/her ability to learn something new, willingness to do that and organization wish to motivate this particular person. Notice that this requires a certain scale to measure each component of this equitation. It is also important to keep those scales in balance (having either equivalent scales or scaling by a component's importance). See for example the Saaty method [12].

The following factor should be used in case somebody is about to be motivated to move into another position – *re-positioning motivation factor*, which is calculated as a multiplication of a person's potentiality to be in this new position, willingness to deal with this position's problems and making decisions, and the organization wish to learn (motivate) this particular person.

Besides other ways and factors specific for an organization to select persons into a team can be used.

3 Common Problems Specific for Motivating Projects

Unfortunately a good project to work on and a good team don't always mean that the project will be successful. There are a lot of other factors affecting the end result and managers tend to forget about those starting the motivating project. Here we are going to present the second group of key factors that should be considered.

3.1 Stress

One common problem that occurs quite often in the motivating projects is a workers' stress. There are various reasons for that and some (main) of them is reviewed below. First of all notice that stress factors can be divided into two groups: indirect and direct, i.e. can be either realised by a worker or latent, and both groups can be presented. There is no much difference between those kinds looking on their consequences as the result is approximately the same: decreasing of efficiency and a wish to avoid such projects; although the kind of the factor can be important to address different stress types. At the same time it is important to keep those different kinds in mind to track stress and factors producing it.

The most often stress factor is an increased responsibility that workers could have in new projects. Sometimes motivating projects are replaced with an "easy to identify" study of other existing projects in order to continue those having said "your will be studying new technologies" and those projects are project that nobody would like to deal with. Fortunately this happens rarely, but the responsibility stress is very common and major reasons this type of stress occurrence are:

1. A need to meet expectations of management
    a. having a little knowledge about the project and therefore believing that this project is extremely complex;
    b. having doubts about own potentiality to continue that successfully.
2. Unwillingness (or a little of own interest) to continue that project at least during the starting phase, i.e. been not enthusiastic about it. Notice that a person could become more and more involved into the project the more he knows about it and even more the more tasks he completes in that project.

Another stress factor is usually a new work environment that is often created during the motivation projects. The change of person's environment could consist of either one or several elements listed below

1. A new work place (physical location);
2. A new team (co-workers);
3. A new tool to be used: software development language, methodology or soft-/hardware to rely on etc.;
4. A new manager or/and a customer. Notice that under the customer we normally mean somebody accepting the team work, so it could be both an outside customer and another team within the organization.

The next stress factor becomes from a need for the team (and basically each member of that) to estimate a length of work for different project tasks having no or very short understanding of how it can be done. This stress factor occurs even in projects run by developers' wish as an organisation still sees it as useful and has some expectations, so the team have a strong responsibility and wish not to fail that. Here we have experienced different employees self-protecting reactions to not fail in estimates:

1. Avoid giving estimations finding different reasons of why it cannot be done. If management will accept this behaviour then sufficient uncertainty will be introduced into the project planning resulting in impossibility to plan properly other tasks both for this team and for others (for example consider uncertainty in a date by which requirements should be finalised to allow a developers to continue development process);
2. Increase estimations sufficiently without any real reasons to do so. The following consequences of this behaviour are common:
    a. Dropping the project as its cost (basing on those estimates) will be too high;
    b. Having gaps in the work-flow if a work is done and there is no completed specification for further work or tasks synchronisation requires completeness of other tasks (read waiting for other tasks to be complete);

c. Resynchronisation with other teams. An obvious example is supporting teams like testing one. The less obvious and more important is reviewers team to identify the further development process.

Notice that the common believe that developers will under estimate the work to be done is quite a rare case in practise. Teams are usually compiled so, that team leaders do have some practise and experiences from previous projects (of course may be not using the new technology, but anyway) so they always tend to over-estimate. That is why the under-estimations' case is not considered here although could potentially produce a lot of pressure on a team having to complete their work faster than normally or fail.

In conclusion we would like to notice that a lot of managers doesn't realise that the stress factor exists in motivating projects mostly since everybody are extremely excited about the work to be done. At the same time hidden stress is much harder to control and therefore managers could face very unpleasant surprise when plans are not met after a lot of efforts have been invested into the project.

3.2 Problem connected to timing, progress and workers abilities

We are going to revise in this subchapter remaining problems that occur in motivating projects and which are mostly connected to timing, project's progress and employees' ability to move the project forward.

The most noticeable and common problem is a work progress, which is slower in many cases in compare to what was expected. It is possible to name several reasons of that. First of all it happens because of different new (unexpected) problems that are connected to the new methodology/technique used in the project. Thereafter a lot of teams tend to rebuild a product they are working on permanently. This happens because of the learning process that continues during the project (development phase). So the more team members learn the more critically they look at the already built parts and the more the like to rebuild it using recently obtained knowledge, so doing the same things twice or more. Of course sometimes the process of rebuilding older parts allows completing remaining tasks much quicker, but it is not so in many cases. The main consequence of this isn't the increased development time as it looks like at the first glance. In the end of ends it is a motivating, i.e. a learning process, so normally this increase is already calculated into estimations (a reasonable increase of time). The main problem is an increased lead-time between releases. Rare releases cut feedbacks from the team customers (internal and external as it was mentioned before) and therefore produce a feeling that nobody is interested in the work the team is doing. Customers become less involved into the process, start to forget what was discussed the last time and so forth. All this finally affect team members who start to think that the project wasn't as attractive as it looked like at the starting phase with corresponding consequences.

Besides a question of estimating employees' capabilities should also be raised. It is obvious to consider additional time needed to learn new materials, but many managers tend to forget other consequences of the learning process. A lot of new information to work with and remember normally decreases persons' productivity – they tire much quicker. Besides they have to establish an own learning environment, search for new materials etc. as usually the learning process is unsupervised. Therefore it is not possible to demand the same high level of output as before in standard projects.

4 Addressing Described Problems

This chapter is designed to provide some ideas on how to address earlier described problems. As a rule those problems should be considered (addressed) not only during the project, but even before the motivating project starts during the planning stage.

First of all skills to be used, developed and motivated within the project should be carefully selected. Theoretically, the wider experiences of employees are the better it is. In practise, this situation often means that too many efforts are spend on skills, which are either never or rarely used later. Moreover, notice stress factors produced by such learning into addition to improperly spent time. Under the circumstances, it is crucial to consider rather a "broad range of kernel (key) skills". In other words it is important to set restrictions on required skills and develop those only within that range. Fortunately this restriction will not normally mean decrease of motivation quality or fewer advantages for a company. As a matter of fact, in most cases the key goal is to make a company flexible and eliminate dependency on certain persons having exclusive knowledge and skills to continue some project. Therefore it is important to motivate employees to learn key methodologies and techniques (current and future) used within the

organisation, so natural restrictions do exist and can be easily enumerated. It is much more important to put persons to work together in different projects, obtain enough knowledge about different projects and create a collaborative work environment, so there will be a set of persons that are able to continue any project.

In addition, it is important to prepare an infrastructure to support motivating projects. The infrastructure the project needs can be divided into two major types: assisting one and relying on one. Quite often new technologies, that we are going to use in the new project, demand much more from hardware, software and communication channels. For this reason, it is important to consider

1. Do customers ready to accept a new product with those increased requirements;
2. Do current investments plan supports purchasing new equipment and software to develop using those new technologies.

Although the above infrastructure issue is crucial to start a project, an assisting infrastructure is not less important since it directly ensures success of the project. The assisting infrastructure is an infrastructure that is used during the project to report progress, bugs, provide feedbacks and so forth. Moreover the motivating project demands to include into the assisting infrastructure also sub-parts supporting the learning process. It can be for example a special courses management system containing different materials or software for web based demos and collaboration. The second important role of this infrastructure is to allow monitoring the process showing more than simple tasks' progress indicators.

Here we arrive to another crucial technique ensuring motivating projects' success – it is important to monitor such projects closely. The best way will be to have a professional (of the area/technique to be learned) that can be assigned to this task. This is necessary to:

1. Avoid spending time improperly by guiding the team;
2. Ensure that learning is really moving forward, i.e. accept the progress and ensure that everybody is giving the best;
3. Approve results and produce a report about the motivating project. For example answer – "Is the motivated person able to do the desired work or he failed";
4. Decrease stress of employees as beginners should feel that they are not alone;
5. Approve and/or correct estimates.

If there is no such person then it is possible to use a person to whom others feel a great respect. For example it can be a professional in another, similar area or a manager that can encourage others.

5 Conclusion

Personnel motivating projects are an important part in the preparation process in order to adopt dependable software development principles since highly motivated personnel is a key success factor for producing really reliable, secure systems supporting continuous development. It is important to understand all dangers and risks running motivating projects in order to achieve defined goals as otherwise employees could become rather unmotivated, stressed and even decide to leave the company. Fortunately some fairly simple approaches and methods that are described in this article allow ensuring success and let us focus on the true goals on the company evolution road.

References:

[1] A. Avizienis, J.C. Laprie and B. Randell: Fundamental Concepts of Dependability. *Research Report N01145*, LAAS-CNRS, April 2001.

[2] D. Daly, B.H. Kleiner, How to motivate problem employees, *Work Study*, Vol. 44, No. 2, 1995, pp. 5-7.

[3] B. Gerhart, How important are dispositional factors as determinants of job satisfaction? Implications for job design and other personnel programs, *Journal of Applied Psychology*, Vol. 72, No. 3, 1987, pp. 366-373.

[4] F. Herzberg, One more time: How do you motivate employees?, *Harvard Bus. Rev.*, Vol. 65, No. 5, 1987, pp. 109-120.

[5] A.A. Khan, Tale of two methodologies for web development: heavyweight vs agile, *Postgraduate Minor Research Project*, 2004, pp. 619-690

[6] D. Kumlander, Software design by uncertain requirements, *Proceedings of the IASTED International Conference on Software Engineering*, 2006, pp. 224-2296.

[7] D. Kumlander, Personnel motivating software reengineering, *Proceedings on the 10th WSEAS International Conference on Computers*, 2006, pp. 826-830.

[8] D. Kumlander, Personnel motivating projects: reasons, implementation and risks, *Proceedings of the 6th WSEAS International Conference on*

Software Engineering, Parallel and Distributed Systems (SEPADS '07), 2007, pp. 30 – 34.

[9] D. Kumlander, On using software engineering projects as an additional personnel motivating factor, *WSEAS Transactions on Business and Economics*, Vol. 3, No. 4, 2006, pp. 261-267.

[10] B. Meyer, Dependable Software, *Dependable Systems: Software, Computing, Networks*, Lecture Notes in Computer Science, Springer-Verlag, 2006.

[11] M. Rauterberg, O. Strohm, Work organisation and software development, *Annual Review of Automatic Programming*, Vol. 16, 1992, pp 121-128.

[12] T. L. Saaty, *The Analytic Hierarchy Process*, McGraw Hill, USA, 1980.