

Essay about managing software projects

written by Marcel D. Wegermann,
from the 16th of July to the 24th of July

Marcel@Wegermann.com
<http://www.wegermann.com>

I. Introduction

II. Main part

1. What is a software project?

- 1.1 Definition of a project
- 1.2 How to manage a rock climbing tour?
- 1.3 How to manage a software project?
- 1.4 Good Communication is important
- 1.5 A team consists of people

2. The Reality of management

3. Target concept for a successful management

- 3.1 The right people for the right tasks
- 3.2 How many people for my project
- 3.3 Motivation and increase the team spirit
- 3.4 Analysing and controlling
- 3.5 The project success is not sure
- 3.6 Risk management
- 3.7 Mistakes are human
- 3.8 Development process optimization
- 3.9 Humans are creatures of habit

III. Conclusion

I. Introduction

In the last 10 years a completely new culture of management is growing up in the software industry. For the first time software managers have a considerably career as a software developer and are able to look back to their own developing experience. In this way Tom De Marco describes the development of the last years in his novel about project management "the deadline".

From this development the modern way of software management is woken up and every day more people starts thinking and working by this way.

We can sum up this modern approaches of management as "agile" management. Until this time - hierarchical, inflexible and strong controlled management was well known in the software industry. Approaches which come from the engineering science. Engineers try to „produce“ Software instead of "creating" software.

But software developing has this own rules, creating software is more like a process of writing a piece of poetry. This essay is about the modern principles of software development and project work.

II. Main part

1. What is a software project?

Software development is a creative process. Similar of writing a piece of poetry or a nice book. That's the way Alistair Cockburn describes "agile software development" in this book.

The problem is that a software project is often too big to be managed by one programmer. How does a poem written by 20 poets look like?

That is the challenge of software development. The coordination of the poets must be managed in that way that all parts fit together. A common style of writing has to be enforced and has to follow a clear concept. Last but not least the story it is ordered by a king, in software project he is named "customer".

If the customer is used to get the ordered software to a fixed price and a given amount of time, than we are speaking about a software project.

1.1 Definition of a project

The DIN 69901 defines a project as a plan, this is indicated by the uniqueness of the conditions essentially as a whole, such as target-setting, temporal, financial, personnel and other limitations.

Definition of the PMI

The american Project Management Institute defines: *“A project is a temporary endeavor undertaken to achieve a particular aim and to which project management can be applied, regardless of the project’s size, budget, or timeline.”*

It a one time effort in a fixed amount of time. Cockburn compares software development with a “aim oriented game”, the mountaineering.

For a group climbing on a rock is it very important to get on the top of the rock in a fixed time frame hopefully the group is protected by a cord.

Mountaineering is a aim oriented cooperative game just as software development in project business.

1.2 How to manage a rock climbing tour?

As a good mountain guide you have to be able to route your team on a secure way to the top of the rock.

1.3 How to manage a software project?

The team is the most imported aspect in software development. A project manager must be able to lead his team. In this way managing means to couch people.

The basics for a good cooperation is communication. Without good ways of communication a project can not be solved successfully.

Sixty percent of all software projects are rated as a flop. Maybe the software is working as expected, but the flow of the communication to the customer was inefficiently or a training was missed to achieve a better result.

1.4 Good Communication is important

Good communication not only to the customer, but also to the other team members is the key factor of good management.

In connection with efficiency of communication it is useful to adopt the term of temperature. The hotter, the efficienter communication one each time. There exists a huge number of methods which are all aimed to increase the temperature of the communication.

Either if its about the optimization of the cooperative spirit or about learning from each other or both.

The hotter the communication is the bigger is the danger of crashing with other team members and to make the project harder than it is in reality. That is the reason for the existence of principles and values for team members. One point contains the ability to discuss and being criticised.

1.5 A team consists of people

Each human is a individual person and therefore individually. In a team the described ability to communicate is very important. Each member must be able to get the opinion and point of view of the others and have to be able to discuss things. In the future of the 21th century humans lacking that abilities will be not as successful as they could be.

But you can not expect that every team member has critical faculties. Further more it is the result of a successful collaborative team. For example the absent of security for one team member, in a simple discussion, can lead to the conclusion that all other point of use are valued as an attack.

The follow of that would be a attitude of protection and therefore the end of a successful discussion. In this way a project sociology is a very relevant topic.

Only if all team members are respecting each other there can be a base for effective communication and therefore there can be a king-size performance of the project.

2. The Reality of management

The describes ideal situation of mutual respect and being together is not often found. Much often a complete other style of working is shown by the management.

Manager are alpha animals. Especially in the top management manager are often isolated from each other. Therefore Tom De Marco describes in this book "the deadline" a ideal contrary manger image. In this book manager work together as a little high effective team.

In practice manager use threats and pressure to manage a project.

Whats not recognized is the fact that a software developer is not able to think faster if he is put under pressure.

It even getting worsen a negative re-enforcement of the pressure to the hierarchy bottom will be the effect. A top manager disobeys the critical views of his 20 project managers and skips their views. The project managers will be demotivated and give this style of working further top down to his 20 software developer. By doing this we are getting very fast 200 demotivated software developers.

Frequently the management tries to rescue a software project by regulating the project with extra work on the weekend. In addition to that Tom De Marco says: „Extra work for a short period of time could help to swear the developer into the importance of the project. For a long period of time extra work has counter-productive effects. Collaborators need time for regeneration. But in many companies more than 40 hours a week are normal.

3. Target concept for a successful management

Because of the described ideal imaginations of management and the vision of negative re-enforcement we are able to create a target concept “how to manage a software project effectively.

3.1 The right people for the right tasks

The individual particular team member should not only be very communicative of cause they have to be experts on this special subject. By doing this it works very fine to arrange a working situation with a high skilled developer and a not so much skilful team member. In this seniors-junior-concept the junior developer has the chance to learn very much from the senior and the senior gets a feeling about his own skill level.

Hiring interview

Most of the managers come to a decision and especial during hiring interviews by having a gut feeling. This is absolute correct and important but on hiring interviews do never do that alone. Two Manager are seeing, listing, feeling and smelling more than only one manager.

3.2 How many people for my project

The most successful projects are controlled by a little core team. They group is initializing the project, planning and designing it together. After this first period of time that core team members should be able to lead a own developer team for one project module. If you have got five members in the core team and ten members in each sub team the project scales up to 50 people very quickly.

3.3 Increasing the team spirit

For the core team members or rather the sub team leaders it is very easy to inspire the other developers because THEY are in the project from the first

second and the project is like here own little baby. Here main task is not only to control the sub project. They have to arbitrate the project spirit and have to motivate the team members. By Tom De Marco the most other things are not very necessary.

The aim should be that the management shows a team spirit which is positively enforced down. So the build-up of a successful team with a good community spirit is the main point of project management.

3.4 Analysing and controlling

For the controlling of a one time software project there are often no key figures which are able to measure. In this situation the manager have to compensate the figures with subjective values, like experience or feelings.

The core team should create a system model, like use-cases-diagrams, with all in- and outputs and there effects on the project. The experience of the project manager is very important here. An Example could be the following statement: *“If Mrs. Müsch have a project request like that one, normally this request will be mean to redesign the graphical module XYZ. The costs of this change are minimal two project days“*. To optimize the model system use the experience of your daily work.

3.5 The project success is not sure

All members of the team should rather clear that the success of the project is never secure. Often there is the feeling „Know the project is in dry cloths“. It is relevant that problems eventually raising up are communicated to the top management very fast.

A good idea for a big project is to install an anonymous email address where anyone is able to send an email to. So the assessment of situation can be transported to the top of the management hierarchy quickly.

3.6 Risk management

Managing a software project are never free of risks, whether it is the failure of communication or the illness of an important developer. It comes down to manage the risks of a project, too. The starting point is to make clear what the source of a risk is and what is the probability of falling into this problem.

An example could be that by building up a house in the winter, 5 per cent of the employees will be possibly ill at the same time. In software development this could result in the changing of a requirement or the customer have to follow the market challenge during the project lifetime. The effect could be to cut down the time frame of the project to get earlier on the market.

Often problems are kept back from the management. There is a big fear that the management react to a problem with restrictions and will try to fix the problem by using negative pressure or the management is not able to understand the technical problem.

A good way to detect project risks is to hire a person who is esteemed for finding and uncovering risks. Tom De Marco describes in this novel: "Release one employee from the attitude 'Yes of cause we will resolve all problems'." Depends of the project size this job will be done by the company chief or by one worker of the quality management.

3.7 Mistakes are humanly

During the process of creating software, mistakes are made all the time. With every new function a software get new mistakes. Important is that simple logical or syntax mistake are uncovered quickly. Therefore software development tools or software testing tools are helping us during the creation process all the time. Especially little human mistakes are able to be uncovered and will be fixed quickly.

In Addition to this failure- and risk management it is important that possible wrong efforts are stopped and the loss will be limited early. Also the

negative success has to be communicated to the management. So the whole company is able to learn from its failures. Tom De Marco says that a lost day on the beginning of a project is as bad as a lost day on the end of a project. There are 1000 possibilities to lose a project day but there is no possibility to win one day back.

3.8 Development process optimization

By thinking about time saving in the way of reducing the number of mistakes we will get very quickly to process optimization. The question is if it is not possible to get one project day back how can we optimize the working hours that the daily work of the project is more effective.

But please take a look on this optimization. Process optimization is changing the daily work of all team members most of the time very hard. For this change every team member is needed. More than one process optimization in a running software project is describes by Tom De Marco as an big failure. Furthermore a to high engagement of process optimization is working contra productive and so more like a “car break“ or a “time destroying machine“.

3.9 Humans are creatures of habit

If it is possible prefer to optimize a development process slowly and in a long period of time. Optimizations in a short term are named by Tom De Marco as „charlatanry“. People need time to change their way of thinking and working. Only if the developer are able to comprehends the better way of working there will be a permanent optimization.

For this modifications a good management is absolute necessary, which is able to communicate this optimizations on a humanistic way. This humanistic approach are the fundamental base of the modern “agile“ management culture.

III. Conclusion

There is a huge number of methods, principles and values to manage software projects with a humanistic approach. Some of them are presented in this essay whether it is about the optimization of the community spirit or the management of project risks. In the end you have to find out the right mixture of all principles to manage your team, like a cook has to find out the right recipe for his cake.

I wish you very much success every day,
Your Marcel D. Wegermann