

Successful Implementation of High Performing Teams in Software Development

Nicolás Martín-Vivaldi
Q-Labs AB
IDEON Research Park
SE-223 70 Lund
Sweden
vivaldi@q-labs.se

Christina Thorman
Ericsson Software Technology
Ölandgatan 1, Box 518
SE-371 23 Karlskrona
Sweden
christina.thorman@epk.ericsson.se

Abstract

Teams and teamwork has become increasingly popular within recent years, but most organisations do not achieve the full potential of the team concept. Usually organisations underestimate the need to facilitate and structure teamwork. Groups of individuals working together will not grow into high performing teams without facilitation on *what to do* and *how to do it*. Teams need to have a process to follow and also to know how to behave when working in teams.

This paper describes an experienced based work model for a team based organisation. It consists of *teamwork*, a process (the team assignment life-cycle) defining what to do when working in a team, and *team dynamics*, that covers aspects for the teams to master to be able to also manage how to handle daily situations when working in teams. The paper also describes how to initiate and sustain this model in a mature way within a software development organisation.

These concepts were introduced at Ericsson Software Technology in Karlskrona, Sweden. The results from the project where the concepts were piloted are very positive. After the introduction of the teamwork model, lead time precision and quality were extremely good, besides the developers found it much more motivating to work in a more controlled and structured way.

Keywords: team, commitment, empowerment, process, assignment, implementation, experience

1. Introduction

Many software organisations today claim that they use “teams” in their development. Often these “teams” are rather a group of people working together without explicit common goals and without joint responsibility. We have identified five aspects that characterises an organisation with well functioning teams:

- **Commitment:** within the organisation projects tasks are treated as agreements which implies higher responsibility and job satisfaction.
- **Empowerment:** teams are empowered to find and implement creative solutions to the problems they encounter.
- **Goal driven:** a clear and explicitly stated goal enables participation and common understanding.
- **Quality:** early fault detection is a key to reach higher quality.
- **Humanity:** the teamwork process ensure that we focus not only on technical aspects but on the human aspects such as personal goals, self-esteem, values, trust and feedback.

During 1994, parts of Cleanroom Software Engineering [1] were successfully introduced at Ericsson (ETO) in Arendal, Norway. Further information about those experiences is described in [2]. A second improvement programme at ETO implied coaching of a team at another part of Ericsson (EPK) in Karlskrona, Sweden working as subcontractor for ETO.

It was decided to introduce the successful parts of Cleanroom from the first project at ETO. One of the selected parts were teamwork (which is more an enabler than a core Cleanroom concept) and team dynamics. The status at EPK before the improvement project was generally good. They already worked in teams, but not as formal as described in this paper and followed their development process. The problem was that there were not many experienced designers. So one of the most important goals was to create a sound environment for learning, technology transfer, and also to weld this group of people together.

2. Problems

Several organisations claim to use teamwork, but in fact most of these organisations just have individuals working together. The following points are the usual symptoms found in today’s industry and how the team based organisation address these problems:

- Employees do not know the goals of the project they are working in
 - Deeper understanding, through better communication and clear role definitions
- Missed deadlines, which usually are realised very late
 - Increased delivery precision, by clear assignments, better estimates and higher commitment
 - Improved leadtime, by creating effective teams and reducing re-work
- Low quality and slow learning, same mistakes repeated within the same project
 - Increased quality, by early fault detection through team reviews and shared responsibility
 - Higher competence, by more time for improving and learning from other team members and creating a culture that enables constructive feedback
- The skills of the employees are not utilized properly
 - Higher motivation, by higher job satisfaction, team spirit and higher commitment
 - Better utilization of resources, by creating teams of complementing team members

3. Team Based Development

3.1. Effective teams

What makes some teams outstanding? Knowledge and experience is the usual answer. But there is more than just these obvious aspects. In high performing teams the team members know how to cooperate so that synergi is achieved. This synergi effect means that a team can deliver result that exceed the result that the team members would have delivered working individually (In popular speech $1+1=3$). There are two basic concepts that a team needs to become high performing [3]:

- an external process describing *what* to do, that is a process that describes how the daily work should be done, activities to perform, documents to write, etc. The external process is in this paper represented by teamwork including a team definition, a process to follow and also descriptions of different kinds of teams described in chapter 3.2.
- an internal process describing *how* the work must be performed. Which are the ground rules that the team should follow? How should decisions be made? How should meetings be performed? These are some of the questions the internal process should answer. Team dynamics represents the internal process, described in the chapter 3.3.

3.2. Teamwork

3.2.1. Definition of a team

There exist several team definitions in the literature, most of them stress the importance of a small team, with a common goal and shared responsibility. The team definition presented here, is tailored to fit the software development environment:

A team is a group of preferably 3-5 people that work together interdependently as a whole over a period of time to achieve a common goal, where the team members are jointly responsible for the result, and strives to maximise performance through innovative methods.

There are some parts in the definition that need to be elaborated:

3-5 people. To create synergi there need to be more than 2 people. The upper limit from a communication perspective is 8 people. However to keep the joint responsibility with the complex products of software development the upper limit is only 5.

A period of time. To keep the focus it is important to have assignments that are defined in time. Team members must know how long they are to work together, so that their effort can be focused.

Common goal. The goal of the organisation and the project must be synchronised with the goals of the team. The goals of the individuals must also be taken under consideration.

Jointly responsibility. To create an environment with commitment the key is joint responsibility. This can seem easy when the project is running smoothly but can be much harder when there is time pressure and missed deadlines. Joint responsibility is achieved by having the team members review each other's products (documents or code) weekly. The concept is called frequent reviews and also one of the most powerful ways for the teams to deliver with high quality.

Maximise performance. It is the responsibility of the whole team to find short-cuts and improvements of the work processes. If a committed and empowered team is established this is usually just happening.

There are three different players in the teamwork concept, the *project manager* responsible for the project, the *line manager* responsible for the resources and the *team members* performing the work. The main roles inside the team are:

- **Team member.** All persons within a team are team members. To achieve the joint responsibility it is necessary that the allocation of each member is clear and is at least 70% within the team. The rule of thumb is therefore “one person -one team”. This is however something we have seen violated several times resulting in lower commitment
- **Team leader.** The team leader should be seen as a role and not a person. The main tasks that this role contain are:
 - Act as the team co-ordinator, the administrative part.
 - Is the formal external interface, reports to management.
 - Assures decision making. (Not making the decision but *assuring* that a decision is made)
 This role should as soon as possible be split between the team members to increase the participation and joint responsibility within the team.
- **External resource.** An external resource is an external person that adds competence within a specific area in order for the team to perform a certain task. This person is considered external since he/she is not responsible for the teams deliverables. He/she assists with the specific knowledge that is need and then steps out of the team. The reason for handling them as externals is to keep the joint responsibility within the team. If there is a team member that has low allocation the team will spend too much time updating this person. The person can be a technical, process or quality person, e.g.: system expert, tester or review attendee.

3.2.2. Team Assignment Life-cycle

Central in the teamwork concept¹ is the Team Assignment Life-cycle (see figure 1), describing the different phases in a teamwork environment.

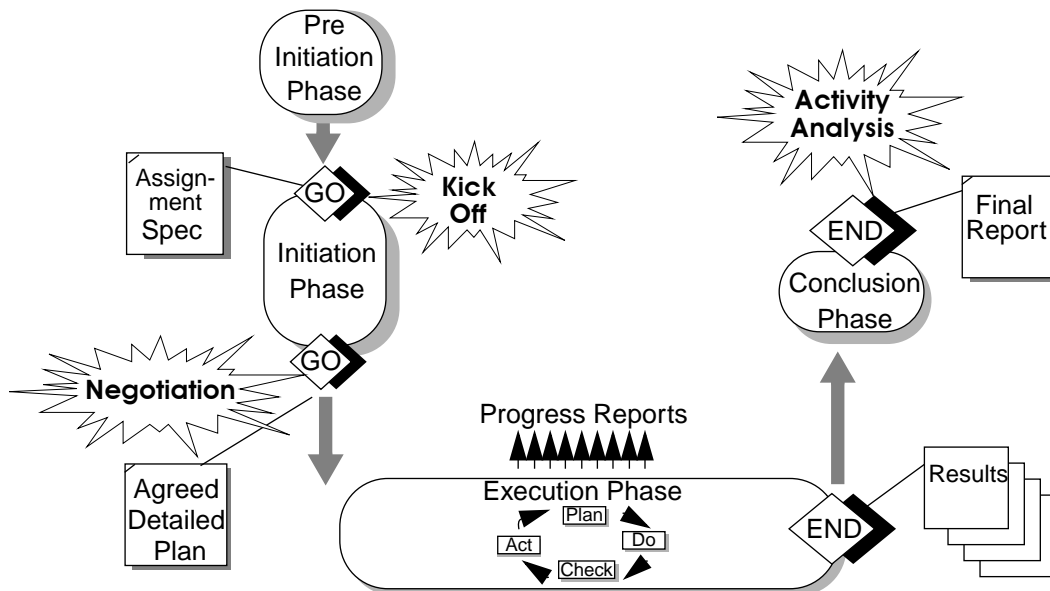


Figure 1. The Team Assignment Life Cycle, with phases, documents and activities.

1. Developed by Ericsson in Montreal, Canada and Q-Labs, for further details see [4]

The different phases are:

Pre-initiation phase. In this phase the project manager and the line manager break down the project into assignments suitable for a team (leadtime of 6-16 weeks) and create teams with the right competence for the assignments. This phase takes about two to three months depending on the size and complexity of the project. For each assignment the following three phases should be performed.

Initiation phase. It is started by a kick-off where the team composition is presented and the team assignments are handed out to the teams. The teams then create their detailed plan describing how they want to perform the task described in the team assignment specification. This will make the teams much more committed since the team members have given their own input to the assignment they are going to perform.

The Initiation phase is ended by a negotiation of the detailed plan where all the three players must participate. The negotiation is really one of the key activities. The goal is to achieve a win-win situation where all three parties feel satisfied. The factors that are subject for negotiation are:

- Leadtime, can the team get more or have less leadtime?
- Resources, can the team get another member?
- Functionality, is it possible to add/cut away functions?
- Quality, if none of the above parameters are changed (when it is needed) the quality will be effected. Either deliberately as a conscious decision or it will turn up at the end and then there is more difficult to handle the situation.

This phase takes about 1-2 weeks.

Execution phase. In this phase, the teams perform the assignments according to the agreed detailed plans. The cycle for this phase should be weekly. The week should be organised by the Plan, Do, Check and Act cycle. The teams plan for a week, work for a week, follow up and take actions (if needed) and plan for the next week. This phase should be between 6-16 weeks. 6 weeks is the lower level since it is uneconomical to plan two weeks for six weeks of work. The upper limit of 16 weeks is due to several aspects, to keep up the focus the assignment should not be too long, the teams must reach conclusions phase repeatedly so that they can reflect and learn and finally if the team should be able to write a plan that they feel committed to it can not be too long.

To maintain the empowered and committed teams, the management must trust the teams and not interfere with their business. The management should instead coordinate the teams since if you manage to create committed teams they will end up self-centred and only care about their assignment. Management must secure that interfaces between the teams are handled. But to keep management out of the team's work the teams need the management to feel that they have the control, that they know what is going on. Therefore progress reports should be written and given both to project and line manager on a weekly basis. These reports do not need to be formal, they could be in mail format, handwritten, web page or a document dependent what the teams and management negotiated, but should at least state how the team is progressing compared to the detailed plan.

If the prerequisites in the assignment changes, e.g. one team member needs to be transferred to a new project with higher priority or that the effort of a task is wrongly estimated, a renegotiation must take place. The renegotiation does not need to be as formal as the negotiation but management need to show that they take the changed prerequisites seriously and agree what the team can do, when taking the changes into account. This is important for the teams commitment since if the renegotiation is not done the teams will not put the needed effort in the detailed planning next time, because they know that as soon as something happens the plans will not be updated and become impossible to meet.

Conclusion phase. When the assignment is completed the conclusion phase starts. It is about 1/2-1 day where the main purpose is to gather improvement suggestions. The teams write their final reports including what went well/wrong and improvement suggestions, both regarding process and product. The Final report is then subject for discussion in the Activity Analysis where the three players; team members, line and project manager, meet to discuss the team's findings. The outcome of this meeting is to define the responsible for each improvement proposal.

It is obvious that the conclusion phase will increase the learning in the organisation, but it also has a psychological effect. When reaching the goal it is important and to take a breath, if the focus and commitment shall be maintained. It can imply to celebrate if the assignment was a success or to regain confidence if it was failure

3.2.3. Different kinds of teams within software development

The teamwork model was primarily developed for development teams but can easily be applied for other kind of teams [5]. This chapter describes different kinds of teams within software development and how they relate to the teamwork model.

- Development teams. Can be refined into two cases that both easily can adopt the teamwork model:
 - Phase Oriented, where all team members have the same skills.
 - Cross functional, where the team consist of team members with different skills, e.g. testers and designers in the same team or designers with different specialities.
- Maintenance teams. These teams do not have clear assignments of 6-16 weeks. Their work can be more as support day by day and can be seen as eternity work. This kind of team needs to specify specific goals for a specific time period and evaluate themselves towards those goals at the termination of the set time. This can be seen as creating fake *assignments* but is really important for the teams to get a chance to reach the goal and to reflect over what has happened to the teams abilities.
- Coordination teams. The team members are allocated on a lower basis (e.g 20%). The work in this kind of team is sometimes close to nothing and sometimes very hectic. The members in the group do usually not have joint responsibility since they have other activities where they spend the greater part of their time, e.g participates in a development team. Here it is important with short assignments that must be treated as much as possible as an ordinary assignment otherwise it is easily delayed since it not the top priority for the team members.

3.3. Team Dynamics

Team dynamics explain different aspects of what happens when people work together in teams. If teams know about these aspects they will mature faster into high performing teams. This paper will not go into depth in this area but will describe some important aspects and how they relate to teams.

- **Process-Content model.** How does a team perform a task? It is not only WHAT to do, it is HOW to it. Technical people tend to rush into the problem without thinking of how to go about.
 - The “what” part is the *Content*, what is discussed, what problems are there to solve.
 - The “how” part is the human *Process*, how the team go about it.

Compared with what an effective team must deal with (see section 3.1), they must know what to do; teamwork, and how to do it; team dynamics.

- **Perception.** It easy to perceive something and draw own conclusions based on one's experience. The teams must have an understanding of what happens, from when we expire, to when we make own opinions and act according to those opinions.
- **Norms.** Within the teams there are usually different norms. The norms are the unwritten rules within a team and it is according to these norms that a team member is judged. It is important that the team agree on common norms so that the team members know what they expect from each other.
- **Personal Goals.** When working in team it is easy to forget that the team consists of team members with different personal goals. Those are very important to share and discuss because they are the reason for many of the individuals' actions.
- **Team roles.** People have different preferred roles when working in teams [6]. The team members should have a basic idea of the different roles, what their strength and weaknesses are. This will help to understand different behaviour in the group and that all roles are equally important and necessary.
- **Communication.** Communication is difficult to master. It not easy to make the receiver understand exactly what one is thinking. Active listening and paraphrasing (repeating in your own words what the speaker just said) are two ways of increasing the understanding. One of the main challenges for a team is to start talking about the aspects that usually is considered to be personal. Of course there must be a part that is kept by the individual but there is more to discuss than the actual work, e.g. norms, personal goals and how one is affected by the others. This must be done through openness, to have good communication within the group.
- **Team Life Cycle.** A team goes through different phases [7] from when:
 - they first meet and everyone is polite to each other and the members question if they belong to the group,
 - they start to question each others roles
 - they have learned enough about each other to work efficiently together

It is important to understand this model to speed up the process to reach the stage where people can make use of each other.

- **Decision Making.** Many problems arise in the teams when decisions are to be made. It is important to understand that there are different ways of making decisions from unanimous to dictator and that each way has its occasion. The main parameters are the time to come to a decision and the quality of it (the commitment). When working in teams consensus is one of the best ways to come to a decision since nobody must be ignored.
- **Feedback.** The best way to improve is to receive both positive and negative feedback. Most organisations try to give positive and negative feedback is hard and it often ends with a tap on the shoulder, "Good work John" (How much will John be able to learn from that). Some important aspects when giving feedback:
 - Be specific rather than general. Try to point out specific examples so that the one receiving the feedback can better understand.
 - Determine the appropriate time and place. Feedback should be given in close connection with the activity and usually one to one.
 - Address changeable behaviour. Feedback on behaviour that the receiver can not change will not result in an improvement.
 - Include yourself. Describe how the behaviour affect you.
- **Conflict handling.** Conflicts are often considered to be something negative. Conflicts are

often based on difference in opinion that makes them to go deep into the trenches. This difference in opinion should be seen as an opportunity to grow. Even though every conflict can not be solved, one can come to a point of mutual understanding. One way of handling is to perform the following steps:

- Check own values, are your own inputs correct?
- Check the opponents values, have you understood the other person's thoughts correctly. This could be done by paraphrasing and active listening.
- Look on the situation from an outside perspective, what is the worst thing that could happen? Is it worth while all the energy?

If this does not help it could be necessary to bring in a negotiator. But do not do this too early. Conflicts are energy and if treated properly, coming to an understanding of each others standpoint a strong and very committed decision can be the outcome.

These aspects can be seen as key issues for team success since a team mastering these aspects will get much more return from using the teamwork concept.

4. Implementation process

This chapter presents a general implementation model for organisations that wants to introduce team based development. Experience of using this process is described in chapter 5.

1. Train Management

The first activity is to train management in the team based development ideas. A general problem is that too many decisions on organisational improvements are based upon very vague knowledge. It is important that management understands the concepts of both teamwork and team dynamics, the possible benefits, the risks and the effort required to implement and maintain the ideas.

2. Strategy

When management is committed to the concept they often rush into implementation of the concept before setting a strategy for the implementation. One way to solve this is to run a vision seminar [8] where some questions must be answered:

- Depth. Should the concept be introduced in a pilot project or applied on all projects?
- Responsible. Which project(s) should apply the concepts? It is important that the project manager is committed and not forced to apply since this will highly affect the result of the implementation.
- External assistance. Is there enough knowledge within the organisation or is there need for external assistance? Using external assistance has the positive effect that they are not biased and that the organisation will take on the ideas more willingly (a prophet is hungry in his own country). The problem with external assistance is not to become too dependent. It is important to clear out how the concept should be maintained in the organisation after the external assistance has left.
- What to introduce. What parts of the concept should be introduced? All parts of the concept might not fit the organisation and it is important to determine which parts are most important.
- Time span. How long do we run the pilot(s)? The advantages of a short pilot is that it gives quick answers as to the success or failure of the concept. However, it is time con-

suming to learn to work with new concepts, therefore it might take time for possible benefits to appear.

- Goal setting. What is the main purpose of introduction? To be able to conclude whether the concept was successful or not it is important to set SMART (Specific, Measurable, Accepted, Realistic, Time-based [9]) goals so that the concept can be evaluated when the pilot(s) is terminated.

3. Tailor model

The general model and documents must be analysed to best fit the project/organisation. In connection it must be analysed what of the old way of working that is to be replaced to avoid unnecessary work.

4. Train Management in tailored version and team dynamics

Since management is going to be the role model for the new concept it is crucial that they know the role they are going to play. There is hardly anything more motivating for the team members than a manager that actively shows that he/she believes in the new ideas. Now it is also time to train management in team dynamics. Usually they already have some training in this area, since it is often part in several manager courses, but this time it should be more in connection with the problems with introducing team based development.

5. Train Team members

Team members must have training in the teamwork concept since that is their new way of working. They must also be trained in team dynamics so that easier can grow into an effective team easier.

Team members should receive their training in connection with project start up. This way they will be more motivated to implement the ideas. The training could also take place in advance, the benefit will be that the team members will have some time to think through the concept and even test some parts in advance. The drawback is that the enthusiasm will be lower and that knowledge is perishable so training must be repeated (in smaller scale though).

6. Introduction Kick Off

In the project kick off it is important to motivate the new concept and explain why this new concept is used. Part of the strategy should be stressed, especially the main goals of the concept.

7. Coaching

During the work it is important that the users of the new concept are coached.

- Managers (Project and Line), could use coaching in: refining the project into suitable assignments, composing the teams, writing team assignments, handling the negotiation, etc.
- Team members, could use coaching in: writing their detailed plan (especially in planning and estimating), handling the team leader role, running a frequent review, etc.

8. Follow up

After some time (about six weeks) it is time to assess the usage. This is mainly done to tailor the focused reminder of the teamwork concepts, but also to get an understanding of how the implementation of the new concept is working.

9. Evaluate/Package

When the pilot(s) is terminated it is important to evaluate if the concept achieved the goals set. Try to analyse the outcome and the connection to the new concept. These conclusions must be packaged so that the organisation does not lose this information.

5. Experience

In this chapter the experience of implementing a team based development at EPK is described.

The project where the teamwork concept was introduced was a rather small development project based on Ericsson 's AXE 10 technology for mobile telephony. The organisation consisted of a project manager a line manager and four development teams (cross functional, with designers and testers in the same team). the project was divided into three releases (A, B and C).

In the first release there where no specific focus on teamwork. The project suffered from having too many new employees with no or little experience. The second release started and Q-Labs, working as external assistance, gave some shorter presentations to the teams about the concepts of teamwork and especially frequent reviews. Then the requirements got changed rather dramatically and the planned effort to introduce the new concepts was postponed. After the second release, it was decided to introduce the teamwork concept together with some team building activities to get the spirit up after the two tough releases. A two day course was provided with the following contents:

Day 1: General ideas of team dynamics regarding social competence, personal skills, communication, meeting culture, team conflicts, resistance to change, and several team building exercises. The day was concluded with a cooking exercise where several of the theories were put into practice.

Day 2: The teamwork concept were presented. The team assignment life-cycle was described and the teams had several exercises high lightening common problems when applying the teamwork concepts.

A month later Q-Labs came back and assessed how the teamwork concepts were used and gave a refreshment of the teamwork concept. The outcome of the assessment was very interesting, all teams had adopted most of the teamwork concept. The teams used it slightly different depending on own interest and understanding but they felt supported by the process.

Regarding the model it self there where some findings:

- No formal negotiation between the team and project manager was held but there had been informal discussions between the project manager and the team leaders. This was found not so well since the teams felt that the team leader was the "owner" of the plans.
- The assignments had been well defined and described in the detailed plans. They were so specified that they could be used for weekly schedules as well. The teams found them very useful. One reason for this is perhaps that the assignments were pretty small, about 8 weeks and that teams also were small. Milestone plan have been written according to the teams' detailed plan. These plans turned out to be very accurate.
- Progress has been followed up each week in status meetings and documented in monthly progress reports.
- Final report (on project level) will be written by Quality Assurance responsible with input from the different teams.

In spite of that the third increment package perhaps was the most difficult one, quality was kept high. One of the reasons for that were frequent reviews, according to the developers. A problem with the frequent reviews was that experienced people sometimes got overloaded due to that they where frequently requested to participate in the reviews.

5.1. Project results

The result of the project is presented in the three releases A, B and C. For each release there are data; size of software, how long lead time, number of trouble reports found in function test and number of faults found during six month in operation. The functionality in the different releases was very much the same.

The extremely good results in release C can of course be explained by more than that the teamwork concept was introduced. The large number of new employees had now come up to speed, a more stable project and the motivation was higher.

Table 1: Results in the three releases

Release	Modified of total Size (KLOC)	Lead time (month)	Trouble Reports in Function Test	Trouble Reports in Function Test/KLOC	Faults found during six month in operation	Faults found during six month in operation/ KLOC
A	0.44 of 11	4	9	0.81	5	0.45
B	1.28 of 24	5	9	0.37	3	0.12
C	5.1 of 39.4	7	24	0.61	0	0

6. Conclusions

In short the team concept was appreciated. EPK found that the knowledge was transferred in the group. Especially the concept of cross functional teams was found good. To have a tester involved early and the contrary to have a developer involved in the testing was found most useful. The possibility to discuss ideas and solutions within the team was also good. The main benefits are that the project participant had much more responsible and understanding during the third release. The lead time precision was extremely good during the third release.

The teams feel that the development has improved with each release. The result also points in that direction. "In the first release we did not know what to do, in the second we started to learn and in the third release we were performing" is one statement from a member in the project. One reason for this is could be the huge number of inexperienced designers that during the project learned a lot. Another reason is the lack of experienced support, handling of requirements, i.e. a system group.

The introduced concepts worked out well. In the second release where the requirements were changed, the spirit was a bit nagged due to all extra work. But in the third increment package the spirit went up again. The work was stable and the knowledge in the new techniques spread. This third release was perhaps the most difficult, it could be that the teams saw that as a challenge resulting that they worked very well and achieved high quality. One problem with teamwork concept was that the experienced people sometimes had to spend too much time transferring knowledge. But that on the other hand is an investment for the future.

One reason for the success of improvement is that the project participants were willing to adopt the new ideas, frequent reviews were used, some teams adopted teamwork concepts in a very good manner. The explanation we found to the smooth adoption is three folded:

- The organisation was very young and it was easy for them to adopt the new ideas since they were not stuck to a specific way working.
- They were already practising parts of the teamwork model. The training gave them the overview and better understanding which resulted in better performance.
- The two first two releases were tough to accomplish, the first due to experience and the second due to the shifting requirements, which resulted in a firm commitment for the concept that described the way they wanted to work.

7. Further work

We have identified the following areas as possible future work:

- Team composition, how should the teams be composed? One way is to analyse the different team roles, to create the right mixture within the team. But there need to be more parameters taken into account like knowledge, experience, culture, personal situations, etc.
- Follow up on team plans, a model for the organisation to be able to feedback the experience of comparing the detailed plans compared with the actual outcome. This is really critical since the ability to write an accurate detailed plan is often influencing the success of the execution.

8. References

- [1] "Cleanroom Process Model", R. Linger, IEEE Software, pp. 50-58, March 1994.
- [2] "Experiences from using Cleanroom at Ericsson in Arendal Norway", Øyvind Johansen, Even-André Karlsson, Proceedings from EISCSE 2, Berlin 1995.
- [3] "Creating TOP flight Teams", Hilarie Owen, London, Kogan Page, 1996.
- [4] "Cleanroom Software Engineering Practices", Shirley Becker, Harrisburg, Idea group publishing, 1997. "The Fifth Discipline Fieldbook",
- [5] "Designing Team-Based Organizations", Susan Mohrman, Susan Cohen, Allan Mohrman Jr, San Francisco, Jossey-Bass, 1995.
- [6] "Management Teams: why they succeed or fail", M. Belbin, London, Heinemann, 1981.
- [7] "Leaders of Schools: Firo Theory Applied to Administrators", Will Schutz, Univ. Associates, US, 1977
- [8] "The Fifth Discipline Fieldbook", Peter Senge et al., Doubleday, 1994
- [9] "The Tao of Coaching", Max Landesberg, Santa Monica, Knowledge Exchange, 1997