

Theo van Stratum  
ID UD2428SIT5955

**Why is it that Internal Project Management mostly fails?**

A Final Thesis Presented to  
The Academic Department  
Of the School of Science and Engineering  
In Partial fulfilment of academic requirements for the  
Degree of Doctor of Science in Information Technology

ATLANTIC INTERNATIONAL UNIVERSITY

*This page is intentionally left blank*

# Index

Table of Figures .....	v
Table of Tables .....	vii
Acknowledgements .....	1
Abstract.....	3
Chapter 1: General Introduction.....	5
Research restrictions.....	6
Chapter 2: Definition of the Investigation .....	8
Areas of project management .....	8
Triangle of Constrain .....	11
Ownership.....	12
Company politics / strategy.....	12
Other projects .....	12
Office politics .....	13
Internal Project influences .....	14
Chapter 3: Dynamics of the anticipated Solution .....	17
The goal and Objective of this research .....	17
Methodology used .....	18
Questionnaire .....	18
Interviews.....	20
Chapter 4: Overall Outcomes .....	21
Years in project management.....	21
Number of project done in the last 2 years .....	22
Methodology used .....	24
Projects .....	26
Exit plan.....	27
Post Mortem Sessions.....	27
Involving Users .....	28
Testing.....	28
Elapse Time versus Working Time.....	29
Chapter 5: Analysis.....	31
Experience .....	31
Risk Management.....	32
Change Management .....	33
Estimating and Scheduling.....	34
Intercommunication .....	35
Algorithmic cost modelling .....	35
COCOMO Model .....	36
Methodology.....	37
Project description and Goal.....	37
Waterfall method.....	38
Usage of the waterfall model.....	39
Arguments for the waterfall model.....	40
Criticism of the waterfall model .....	43
Agile Method.....	45
The Agile Manifesto .....	47
Comparison with other types of methodologies.....	48
When to use agile methods.....	49

Boehm and Turner's risk-based approach .....	49
Criticism on the Agile Method.....	50
What choice to make? .....	50
Chapter 6: Conclusion .....	51
Communication lines .....	54
How to avoid the failing of internal projects? (Recommendations) .....	55
Final statement: .....	57
References .....	59

## Table of Figures

Figure 1: Failing percentage Internal versus External PM. ....	5
Figure 2: Areas of influence (drawn by author) .....	8
Figure 3: Triangle of Constrain .....	11
Figure 4: Years in Project management.....	21
Figure 5: Number of projects .....	22
Figure 6: Projects successful .....	23
Figure 7: Methodology used .....	24
Figure 8: Change Management .....	25
Figure 9: External versus Internal Projects .....	26
Figure 10: Success rate versus experience .....	31
Figure 11: Risk Management versus Experience .....	32
Figure 12: Change management versus experience .....	33
Figure 13: Waterfall method.....	39
Figure 14: Communication Internal PM.....	54
Figure 15 Communications External PM.....	54

*This page is intentionally left blank*

## Table of Tables

Table 1: Project Budgets.....	26
Table 2: Number of Resources .....	26
Table 3: Project time .....	27
Table 4: Cause of failing .....	27
Table 5: Testing time .....	28

*This page is intentionally left blank*



Why is it that Internal Project Management mostly fails?

## **Acknowledgements**

I wish to thank:

Mr. Iain Clark from Albanet Limited for the support he gave me in believing that I could do this study and in the way he created the opportunity for me to do this.

Dr. Franklin Valcin for his encouragement and advice during my time with the Atlantic International University.

Last but not least my wife Maureen van Stratum for the unlimited support during the time it took me to reach this point.

Inverness, 2 March 2006

Why is it that Internal Project Management mostly fails?

*This page is intentionally left blank*

Why is it that Internal Project Management mostly fails?

## **Abstract**

This paper discusses the reason why there is a more likelihood that a project, led by an internal project manager, fails the led by an external project manager.

This research is done by interviewing project managers in software development, internal and external, to see if there is a common ground for this and if there is a way to follow and avoid this failing. Although the research is done in this area there is enough evidence to believe that the reasons found are similar in other project management areas.

The paper describes the general differences between the work systems, experience, and methods used by the two groups of managers.

In general we see that the external project managers have more experience, and with that the authority to handle projects and their conflicts in a way that the project keeps on track. The internal project manager lacks this authority by his colleagues and managers and they could interfere with the project in more then one way and endanger the project with those actions.

Further is the lack of the use of a formal method to lead and register the project one of the biggest causes of failure. Things like: User requirements, Feature creeping, Change and risk management are more likely to be in place with an external project manager then an internal project manager. This is sometime not even by choice but forced due to costs. This is especially the case within the triangle of constrain where we see the three main control places of a project: Cost, Time, and Quality. No single person or group should be allowed to be in control of all three of them, but in the case of internal project management we see that this is mostly the case. If there is an alteration in one of the three points the project manager must be allowed to change the others accordingly and this is not always done and brings the project in danger.

As recommendation we see that proper training of the internal project manager in a formal method is a must, and if this is combined with a mentor during the first project, we will increase the success rate of internal project manager drastically.

Why is it that Internal Project Management mostly fails?

*This page is intentionally left blank*

Why is it that Internal Project Management mostly fails?

## Chapter 1: General Introduction

In the many projects I have done over the years I came across a phenomenon that gave me the idea for this research. It appears that projects that are carried out by internal project managers have a higher failing rate than the same type of projects led by external project managers. In a small investigation amongst my peers I came to the conclusion that this problem is broader than expected and with an average failing rate of 65% with internal project management or more against 28% with an external project manager, a research into this phenomenon seems appropriate.

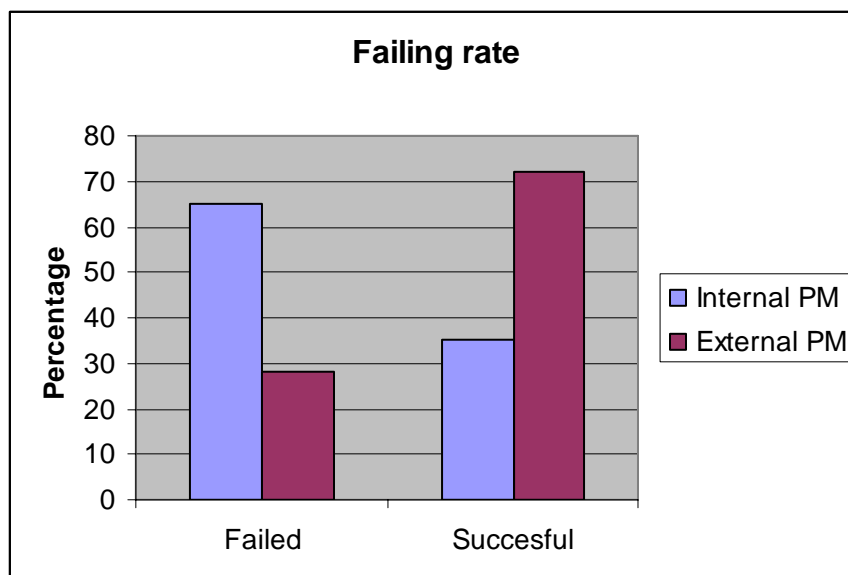


Figure 1: Failing percentage Internal versus External PM.

In figure 1 we see that the failing percentage of internal project management versus external project management is too high to be coincidental. These figures are taken over 593 projects, equally divided over internal and external project management, over the last two years (2004, 2005).

Companies are losing millions per year due to this phenomenon still, it seems that there is no learning curve within those companies to stop this from happening again and losses are taken on face-value. This is especially true with organizations that are not driven by profit like Government and semi-Government. As an example we can take the construction of the Government

Why is it that Internal Project Management mostly fails?

building in Scotland (Edinburgh), which went 10 times over its original budget and took 2.5 years more than planned. There were external project leaders in place to do the actual work on the floor but the overall project management was done by an internal project bureau. Although there was an enquiry, real action to solve this wasn't taken.

If there is a common approach or cause, then the knowledge of such could prevent project managers to make the same mistake over and over again and save the economy millions. The intention of this research is to find such a common cause.

During the research it was necessary to redefine the common definition of failing of a project. Amongst the enquired project managers we noticed that the definition about failing a project was very diverse: from abandoning a project to failing to deliver on time. The definition in this document about failing a project is as follows:

*“A project appears to be failed if the outcome of the project is not within time, cost, or quality, defined at the start of the project or after changes made through approved change requests.”*

The parameters: Time, Cost, and Quality are parts of the triangle of Constraints, which plays, as we will later see, an important role in project management. From this new perspective we can see that a project can be finished with the desired product or service but still is failed within the definition of failing. The project could have cost more, took longer, or not all the requirements were implemented.

### ***Research restrictions***

The researched projects were limited to software development projects only. These are the best documented and traceable in historic events during implementation. The enquired project managers were in the range of

Why is it that Internal Project Management mostly fails?

professional project managers to the occasional, temporary, project manager. The reason for this differentiation was to get an insight in the approach of managing and failing of the projects.

Why is it that Internal Project Management mostly fails?

## Chapter 2: Definition of the Investigation

As described in the previous chapter this research is restricted to the area of software development. The problems encountered are more general and not restricted too software development only but the area is take due to the fact that the processes used are more approachable for research purposes.

### Areas of project management

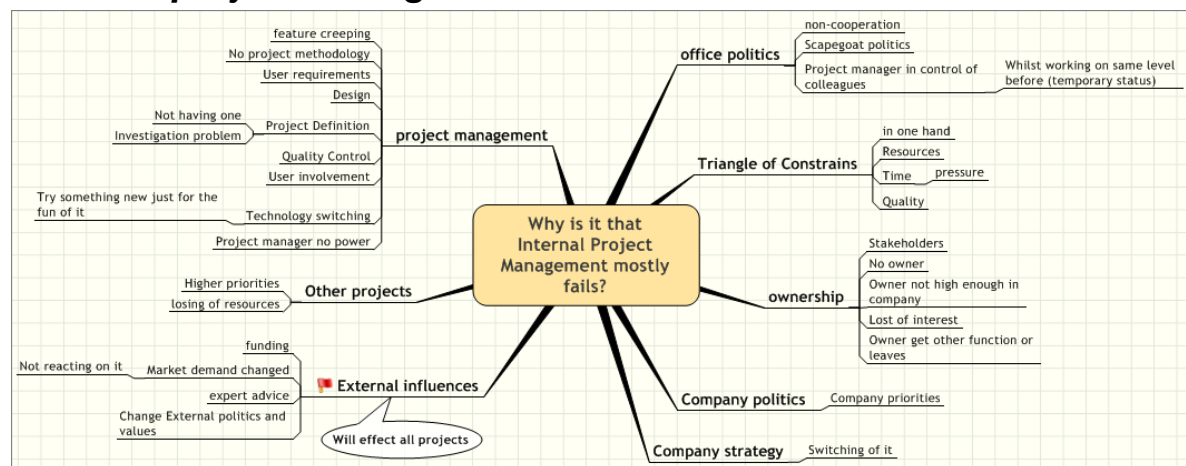


Figure 2: Areas of influence (drawn by author)

In figure 1 we find the areas around a project that are of influence of the outcome of that project.

As Richard Newton (2005) mentioned; “A Project Manager needs to look over the boundaries of his or her project”. What he means by this is: that beside the problems a project manager encounters in his or her project, there are external influences that could have a bigger effect on the outcome of the project then at first evaluated. There are influences were the project manager, nor the company, is in control off. A few of those external influences are, but is not exclusive to this list:

#### External Influences:

- Funding,
- Market demand changes,
- Change of external politics and values,
- Technology.



Why is it that Internal Project Management mostly fails?

### **Funding**

In the case of losing external funding, investors are rejecting their funding to the company or project, the project is doomed to fail unless the company could raise other funds. The project manager is not in control of this but needs to investigate (as far as this is known in this point in time), before starting the project, if adequate funding is available and/or reserved for the project. In principal it means that an investigation into the solvency and cash flow of the company needs to be done. In the case of the project managers interviewed for this research I found that this was never done and the responsibility of the funding towards the projects was placed by the responsible directors.

Although this is a sound approach we must remember that in the case of an internal project manager this is very difficult to do as the directors involved could be offended and it could be on influence of the career path of the persons involved, but on the other hand we will have the same problem when the project fails and the finger is pointed towards the project manager as cause of failing.

### **Market demand changes**

This is happening when the product or service developed in the project is overtaken by new demands and or developments in the market. Most of the time caused by the fact that the time to market for the product or service is to long and competitors have the time to develop a new, better, product before the project is finished. Again not something the project manager is in control off but adequate market monitoring could indicate early enough that a demand change is on the merge and the project manager can react on this before the event occurs. Although market monitoring should be an active task for the directors of a company, in the case of a project of which the outcome is to fulfil a market demand the project manager shares the responsibility in this.

Why is it that Internal Project Management mostly fails?

### **Change of external politics and values**

When Ethical and/or Political values are changing in a way that it has an influence on the project outcome (the product that is developed is not ethical or politically correct anymore) means that the project is obsolete and needs to be abandon or changed in a way that the product or service falls within the new legislation or ethical values. Again monitoring this should be a company/directors task but a project manager could be active in this as well.

### **Technology**

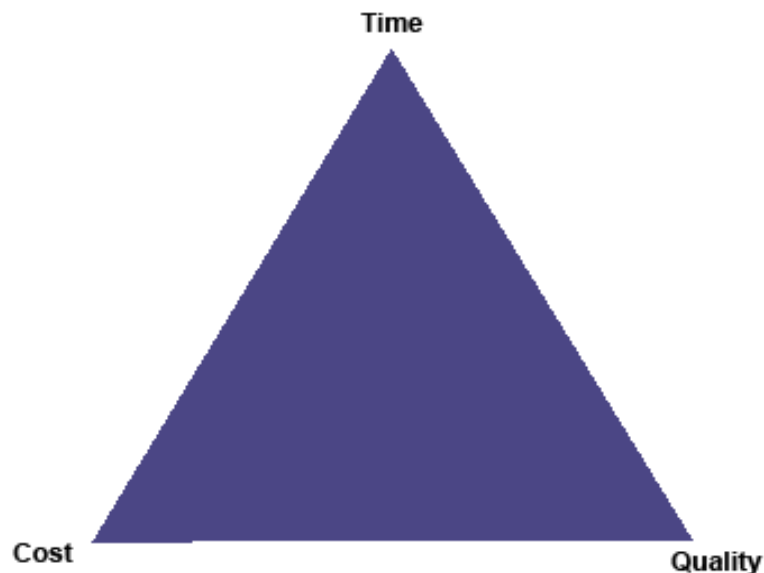
When technologies changes during the project, and this is especially valid in long running projects, a choice needs to be made if the project is still valid to finish or if technologically adjustments need to be included. As usual this is a question of funding and market demands and should be made by the directors of the company. As an example we can look at the new development of the operation systems of Microsoft. In the last few year this is changed for development advantages with the new dot.net Framework. If you are running a development project you will face the question: are we adopting the dot.net framework or not. Of course there will be a lot of parameters included in this decision like: how far are we from completing? What is the cost involved in changing: time, training, and so on. Is it better to finish now on older technology and adapt to dot.net in a later version?

All of the external influences are outside the scope of control of the project manager but close monitoring of the external environment can give clues early enough for the project manager to react proactive.

Why is it that Internal Project Management mostly fails?

### Triangle of Constrain

If there is one thing a project manager should know about engineering something then it is the Triangle of Constrain (see Figure 3)



**Figure 3: Triangle of Constrain**

*Cost* is the measure of how many people, engineers and overhead like secretaries and so no, are working for the project, *Time* is the measure of how long the project has to finish, and *Quality* is the measure of how mane features (user requirements) and testing (Quality Control) will be allowed within the project scope,

Controlling time, cost, and quality are all important goals which need to be under control but *nobody should be in total control of all three of them*. If the customer, or manager, arbitrarily specifies all three corners of the Triangle of Constrain, then the project is doomed to fail. Any change to one of the corners must be balanced off by changes in the other corners. The moral is that if the project is to be successful, the project manager must be permitted to make the necessary adjustments to at least one of the goals.

Why is it that Internal Project Management mostly fails?

In the 1990s, NASA briefly adopted the slogan 'Faster, cheaper, better.' This was followed by a series of unsuccessful projects – and then they abandoned the slogan. It is important to realize that, pushed to the limit, the 'Faster, cheaper, better' slogan is impossible to satisfy. It is absurd a statement as 'I can fly'. There is a saying among software engineers that *corrects* the statement of NASA's praiseworthy but impossible goal: 'Faster, cheaper, better: pick two out of three.'

### **Ownership**

Ownership of a project needs to lie by a director, or somebody from senior management, is a must for a project to succeed. The reason for this is that it gives a project status and priority with in the company from a higher management perspective. When resources and funding needs to been allocated to several projects the owner is mostly in the position to make decisions and could influence that the projects he owns, is responsible for, will get the needed funding.

### **Company politics / strategy**

Company politics are very important and are tied with company strategy. They could influence the future of a project. If the companies' strategy is changes it could mean that project can becoming obsolete. This doesn't mean that a project fail in the sense of this research but as priorities are changing with strategy it could mean that projects are becoming lower in importance and with that the change of failure becomes more realistic.

### **Other projects**

As always there are only limited resources available within a company and several projects will compete for those resources. As explained in the Triangle of Constrain: if there is a loss of resources then alterations need to be made in the time and/or quality of the project. This is a threat in a running project especially where a resources is just lend for a small task to another project.

Why is it that Internal Project Management mostly fails?

We must make sure that action is taken against such activity and to adjust the project planning accordingly.

### **Office politics**

This is a grey area within project management and working for a company in general. Office politics is about power play between colleagues and can be of an enormous influence on the project. If the project manager is respected and the project has a high priority within the company that this influence will be low or even be an advantage. But if the project manager is temporary chosen from the staff and bad feelings about this are amongst the colleagues then this phenomenon could be the cause of failure on its own.

All the above described 'external' influences on projects are mostly not in the power-reach of the project manager and are difficult to manage, but if the project manager wants to finish his project successfully then monitoring of those external influences is a necessity at all times.

Why is it that Internal Project Management mostly fails?

### **Internal Project influences**

The following issues in project management are common and everybody is taking them on face-value, but a lot of projects are failing just because of that. Keep in mind that according to the interviewed project managers most of the problems within the project are forced and with that not in their own control anymore.

### **Project definition**

Although very logical, you must have a project definition (a goal) you are working towards; it seems that this is a mostly forgotten or underestimated part of the project. In the questionnaire, which we will discuss in the next chapter, we had a few questions about Project definitions and user requirements and we saw that in the case of a project definition only 25% of internal projects had one. This means that in 75% of the internal projects the teams didn't really know what they were making! In the case of software projects we see that the common line from management is: "You know what I want, don't you?"

As V.A. Vyssotsky (The Mythical Man-Month (2004) pp 142), of Bell Telephone Laboratories' Safeguard Project says, "The crucial task is to get the product defined. Many, many failures concern exactly those aspects that were never quite specified"

### **User requirements**

As the project definition gives us a general framework of what the customer wants, the user requirements should be very specific and non-ambiguous. It tells us what it is that the outcome of the project should have or not have. If we take a software building project then the user requirements should tell us exactly the business-rules for the product. Appropriate time should be set aside in the project planning to get all the user requirements on paper. Although we must be aware that in very long running projects it is not always the right way to go by defining all the user requirements as an initial step. We have seen in the paragraph above that projects are changing over time and with that the

Why is it that Internal Project Management mostly fails?

user requirements. We will see later in the document how we can handle this in an appropriate way.

### **Feature creeping**

Feature creeping is one of the biggest causes of project failure, in the research done we saw that more than 95% of projects are suffering from this. A feature if nothing more than an enhancement of the final product, it can go from a nice to have till a must have and every project manager has to deal with them. They are un-avoidable in real-life. It is only the way they are dealt with. If we are looking at the triangle of constrain we that on the right lower corner we have quality as a measure point. Qualities are the features in a product so according to the triangle of constrain if we add features then we must accordingly alter one or all of the other measure points, time and costs. With in feature creeping this is normally not done. Under the saying: "O, this is only a small thing to adjust or to add" a feature is added to the project and the project start to run out of cost and or time before we know it.

### **Quality Control**

Quality control is where the quality of the project and product is monitored. It ensures us that all the user requirements are implemented and tested that they work in the way it is described in the requirements. This begins long before we are even starting to build anything. As Brooks mentioned (2004): "Specifications, user requirements, need to be tested on their completeness and clarity." and needs to be handled by an outside testing group. This is especially true in software engineering projects, as Vyssotsky says; the developers themselves cannot do this: "They won't tell you they don't understand it; they will happily invent their way through the gaps and obscurities".

Why is it that Internal Project Management mostly fails?

### **Project management methodology**

There are several approaches that can be taken to project management, including phased, incremental, and iterative approaches.

The "traditional" approach identifies a sequence of steps to be completed. This contrasts with the agile software development approach in which the project is seen as relatively small tasks rather than a complete process. The objective of this approach is to impose as little overhead as possible in the form of rationale, justification, documentation, reporting, meetings, and permission. This approach may also be called the "spiral" approach, since completion of one of the small tasks leads to the beginning of the next. Advanced approaches to agile project management, applicable not only to software development but to any area, utilize the principles of human interaction management to deal with the complexities of human collaboration.

### **Technology switching**

Technology switching during a project is expensive and a dangerous thing to do. We see this type of actions mostly in software projects where designers, programmers, and so on come across a new language (like C# in the last years) and start to redesign the project in such a new language. Of course this is deadly from a project management point of view, although it is not always possible to avoid this. Especially in the case of Company politic to always follows the latest technology.

All the subjects that are described above are points of failure when not proper handled or monitored according to the interviewed project managers during the research.



Why is it that Internal Project Management mostly fails?

## **Chapter 3: Dynamics of the anticipated Solution**

### ***The goal and Objective of this research***

As described in the question of this research: “Why is it that Internal Project Management mostly fails” I try to find the reasons behind the phenomenon that over 65% of projects that are led by internal project managers fail. If there is a specific reason for this then knowledge about this can lead to a solution or methodology to increase the success rate of future projects.

The questions that came up where:

- Is this happening due to inexperience in project management?
- Is it more likely to happen in projects that are in an unknown area for the project manager?
- Which methodology is used, if any?
- Is it market specific, in other words: is it specific to a type of project? (software engineering, design and so on)

In general literature about project management we find everything about methodology and techniques, how to report and on what, but what they don't try to teach is the fact that project management in the first place is “managing of people”. When I was researching this and interviewing project managers about the way they work and what they were thinking the failure was, then all the points from chapter 2 where mentioned but nobody was talking about the people behind this failure points. Because of this and my feeling that one of the most difficult task a project manager has is the human interaction, if it is the man on the work floor or the director in the boardroom, the project manager must communicate on their level but not only that, he must also translate between those levels to make sure that they understand each other. Which brings me to the second goal of this research; “Is communication the biggest point of failure within project management?”

Why is it that Internal Project Management mostly fails?

### ***Methodology used***

For the main part of information gathering I used quantitative research via a questionnaire. The reason for this type of data gathering is that is cheaper and not that labour intensive. The questionnaire was email based and sent to over 500 project managers over the world. Because the questionnaire was on personal application the response rate was very high, 76% of the questionnaires sent, returned. As said before the projects in this research came all from software engineering. It is however my strongest believe that the results will be the same in every other form of project management due to the fact that people will find the same problems and solutions everywhere.

### **Questionnaire**

The questionnaire was built in 4 groups of questions:

1. Experience
  - a. Years in project management,
  - b. Professional certification,
  - c. Number of projects done,
    - i. How many successful,
    - ii. How many failed,
    - iii. How many abandon,
2. Methodology normally used,
  - a. Which one, if any
    - i. Officially trained in it,
    - ii. Waterfall –method
    - iii. Agile method
  - b. If any followed completely,
  - c. If any, was it helpful,
  - d. How much time spend in following the rules
  - e. If not, why,
  - f. Was risk management in place,
  - g. Was change management in place,
3. Projects

Why is it that Internal Project Management mostly fails?

- a. How many External
  - i. On average:
    1. Project budget
    2. Number of resources
    3. Estimated project time
- b. How many Internal
  - i. On average:
    1. Project budget
    2. Number of resources
    3. Estimated project time
- c. Success rate External
- d. Success rate Internal
- e. Looking at the failed projects, if any, what would be the area of failing,
  - i. Time (project took longer then estimated,
  - ii. Cost (project was more expensive then estimated)
  - iii. Quality (not all the user requirements where implemented),
- f. What was, according to you, the main cause of failing (average over the failed projects),
- g. Do you have an exit plan in place?
- h. Do you hold post-mortem sessions?
- i. Where users constantly involved.
- j. How much time was taken for testing, in percent of total project time,

After the questionnaire 20 project managers where selected, to get a broad range as possible, for personal interviews to get a deeper understanding on the success and failing rate in this area. The selection criteria for those project managers was that they all had to have done internal and external projects whilst one half of the group needed to have more failures in the internal projects and the other half needed to be more successful overall.

Why is it that Internal Project Management mostly fails?

## **Interviews**

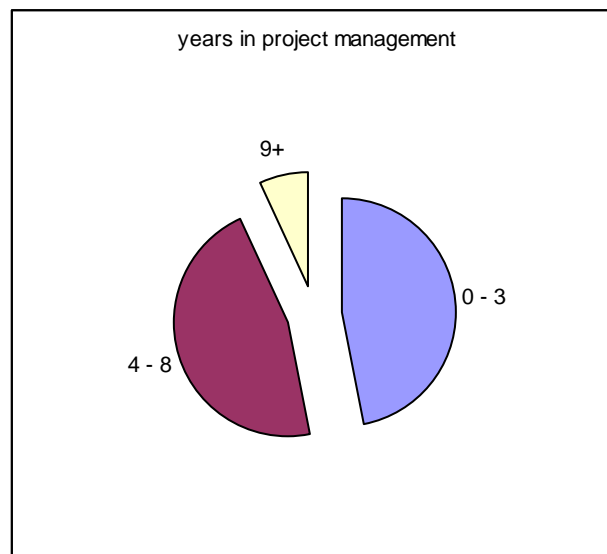
The interviews were done in person, over the phone, or by direct internet communication (chatting). All the questions were open ended and none of the project managers were given the questions in advance. Because of this we ended in most cases in an open discussion. This gave me a good insight into the way project managers are thinking. I found in general that first-time project managers are optimists and believe in a happy ending.

Why is it that Internal Project Management mostly fails?

## Chapter 4: Overall Outcomes

From the 500 questionnaires sent, 382 were returned which gives a return percentage of 76.4%. In total 593 projects were taken by those project managers in a time period of 2 years.

### *Years in project management*



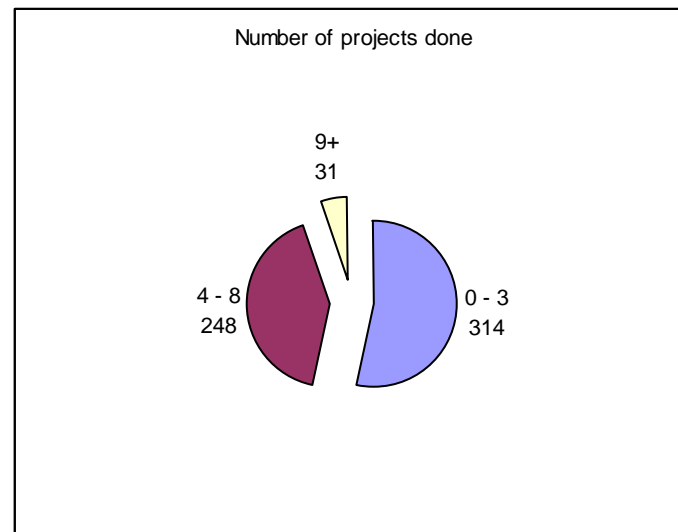
**Figure 4: Years in Project management**

On the question: “How many years experience of project management do you have we see that the division between the group from 0 – 3 and 4 – 8 is equal, respectively: 176 and 179, whilst only a small group (27) has more than 8 years of experience. The group from 0 – 3 years of experience was asked the question if this was their first project and 83 answered positive on this (46.36%).

Only 13.35%, 51, of the project managers had a professional, all PRINCE II, certificate and from those 51 managers none of them was in the 0 – 3 experience group.

Why is it that Internal Project Management mostly fails?

### ***Number of project done in the last 2 years***



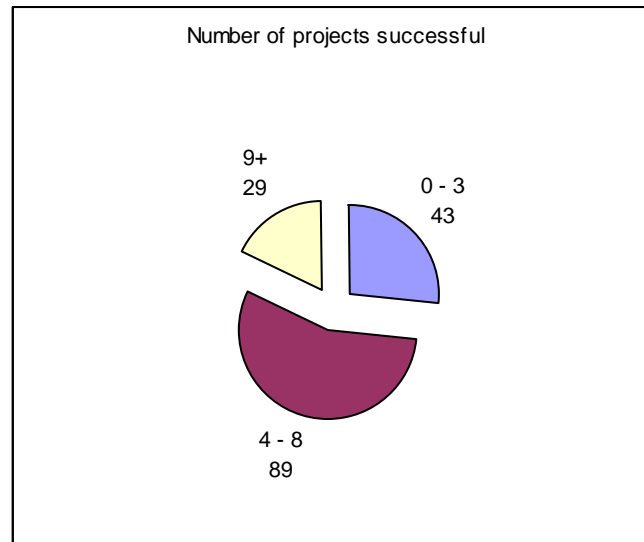
**Figure 5: Number of projects**

On the question how many projects are done in the last two years we see that in the experience group 0 – 3 the most projects are done, 176 project managers did 314 projects, This is 1.8 project per project manager, whilst with the more experienced project managers this is lower, 1.4 in the group 4 – 8 years, and 1.1 in the 9+ group.

On the sub-question: How many projects do you run at the same time? We see that in the group with the less experience the most projects are done in the same time or overlapping. With the average of 2, whilst in the group with the highest experience all of the project managers are doing only 1 project at the time.

Why is it that Internal Project Management mostly fails?

On the question: “How many were successful, according to this research success definition?”



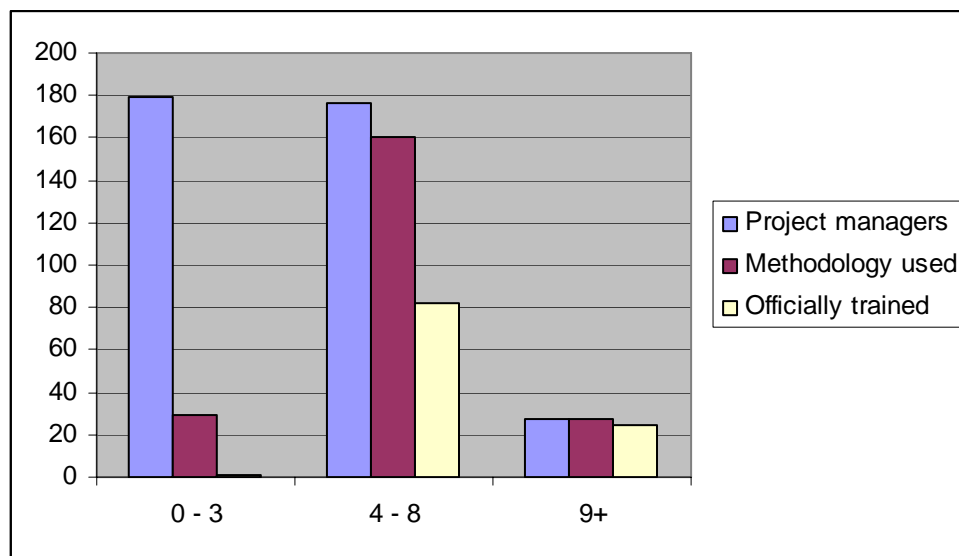
**Figure 6: Projects successful**

In the group 0 – 3 we see that only 13.7% of the projects done are successful, that is: within time, costs, and quality expected. The group 4 – 8 is slightly better: 35.9%, whilst in the 9+ group 93.5% of the projects were successful.

In total only 27.2% of the projects were successful if measured again the research parameters, which is worrying for the quality of project management in software engineering.

Why is it that Internal Project Management mostly fails?

### ***Methodology used***



**Figure 7: Methodology used**

On the question if a methodology is used we see that the more experience a project manager has the more a project methodology is used and officially trained in the method.

On the question which methodology the answer was consistently PRINCE II, which is not surprising in software engineering where PRINCE II is the main methodology. Within project management we see that the agile project management method is rising but when the question about this was asked we saw that only 7% was using this, not including the project managers that were not using a methodology and the rest, 93%, uses still a form of waterfall method.

On the question if the method was followed completely through none of the project managers did this and all were using some form of adjustment in the workflow. Most of the ones that were using a method found this useful and giving them peace of mind.

The project managers that didn't follow an official methodology were asked why did was and 87% didn't know how to but would be willing to learn whilst the other 13% did see the need in them due to the fact that their projects



Why is it that Internal Project Management mostly fails?

where very small and they thought that the hustle in maintaining the documentation around the projects would be too costly in time.

In the case of risk management, what are the risks in the project that could be the cause for failing, we see that only 45% of the project managers actively looks at the risks and register them against the task involved. The idea of risk management is that contingency plans are developed in the case the event occurs.

Change management was in place in over 55% of the projects.

Divided over the experience groups give us the following overview

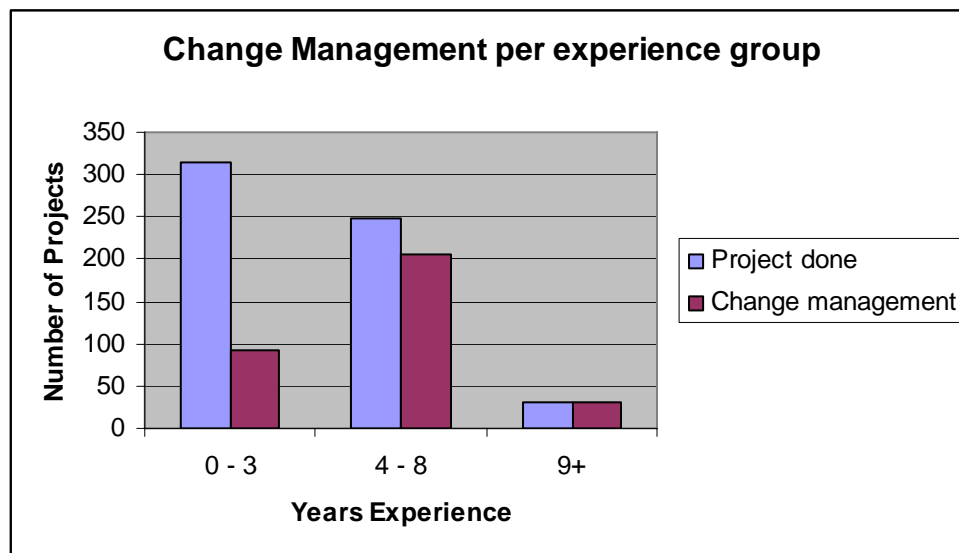
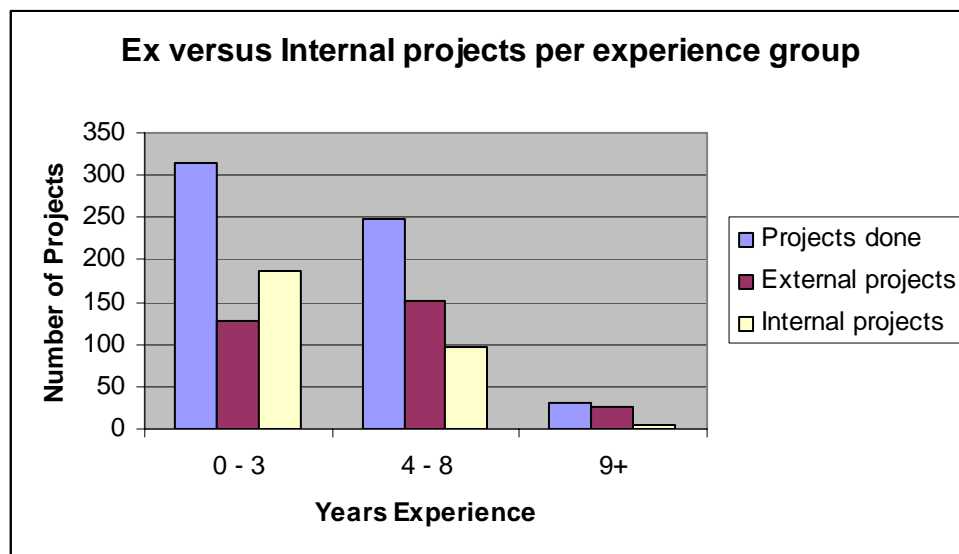


Figure 8: Change Management

Why is it that Internal Project Management mostly fails?

## Projects



**Figure 9: External versus Internal Projects**

On the question how many external versus internal projects are done with in the experience group we see that there are more internal project done by the less experience groups than the higher experienced ones. 60% in 0 – 3 group, 39% in the 4 – 8 group, and only 13% in 9+ group.

The average project budget was

**Table 1: Project Budgets**

Experience group	Internal Projects	External Projects
0 – 3	\$ 5.000 - \$ 12.500	\$ 10.000 - \$ 26.000
4 – 8	\$ 4.500 - \$ 23.500	\$ 17.000 - \$ 150.000
9+	\$ 8.000 - \$ 50.000	\$ 25.000 - \$ 1.250.000

The budgets are recalculated to dollars

The average number of resources in projects

**Table 2: Number of Resources**

Experience group	Internal Projects	External Projects
0 – 3	3 – 5	3 – 5
4 – 8	2 – 7	5 – 12
9+	2 – 8	8 – 25

Why is it that Internal Project Management mostly fails?

The average project time (elapse time)

**Table 3: Project time**

Experience group	Internal Projects	External Projects
0 – 3	13 – 15 weeks	12 – 14 weeks
4 – 8	12 – 17 weeks	13 – 20 weeks
9+	13 – 30 weeks	17 – 50 weeks

On the question: “Looking at the failed projects, what would be the area of failing?” This question is very subjective and the interviewed project manager could give more than one answer. The idea behind the question was to get a feeling of what the project manager him self though what the biggest problem was. The figures are in percentage average over the projects

**Table 4: Cause of failing**

Experience group	Time	Cost	Quality
0 – 3	75%	95%	35%
4 – 8	87%	81%	40%
9+	56%	25%	10%

### **Exit plan**

An exit plan is planning for failure. If we are thinking of project management then most project managers are thinking of success but if we see the failing rate then we should think about failure as well. What are we going do to if the project fails, and need to be abandoned? This is described in an “exit plan”. When the question was asked only 1 project manager had used an exit plan.

### **Post Mortem Sessions**

To learn from previous projects it is always good to have a post mortem session after finishing a project, if it was a success or failure you can learn from it and repeat the good things and try to avoid the bad ones. From the project managers interviewed none had used a post mortem at all. The main argument was that there was no time and budget reserved for this.

Why is it that Internal Project Management mostly fails?

### **Involving Users**

This was to see if users are actively involved in the project. The effect of involving users can be good, if the project manager handles this correct. In the agile project and development methods it is common that the users are part of the project team. This way they can make sure that the customer gets what he has paid for.

Only 36% of the projects done had a close relationship with their users and this was the same in all the experience groups.

### **Testing**

Due to the fact that the project managers interviewed came out of the software development area we had a question about testing, which is in every software development project a hot item. On the question: "How much time was taken for testing, in percent of the total project time" we got the following figures:

**Table 5: Testing time**

Experience group	internal	external
0 – 3	5%	7%
4 – 8	12%	17%
9+	8%	21%

The interviews were more to get a global overview about how a project manager thinks when working on his project. There are not direct measurable figure but the outcome from those interviews are incorporated into the analysis of this research.

Why is it that Internal Project Management mostly fails?

### ***Elapse Time versus Working Time***

A common problem within project management is the difference between working time, man-days, and elapse time, project time. The problem is that when project managers estimate the task project times they take a man-day for an eight-hour working day and this is unrealistic. Nobody is working 100% of its time per day. For my own projects I did an investigation into this and used my own group of programmers to write their time down, for a period of 8 weeks, in everything they did. We came to the conclusion that the disturbance of telephone, meetings, questions, getting and drinking tea and coffee, and having sanitary stops is taking a lot of time away from the working hours. Average over the group and eight-weeks we saw that between 60 and 75% effective working time was left for planning. What does this means in planning? When the project times are estimated we must first calculate the man-days (man-hours) and value them at the percentage we take for effective programming time for that programmer, this can be different for each of the programmers involved, and then recalculate them for project elapse time. If we say that a task cost 10 hours and we have an effeteness of 70% then the elapse time for that task is 14 hours.

Why is it that Internal Project Management mostly fails?

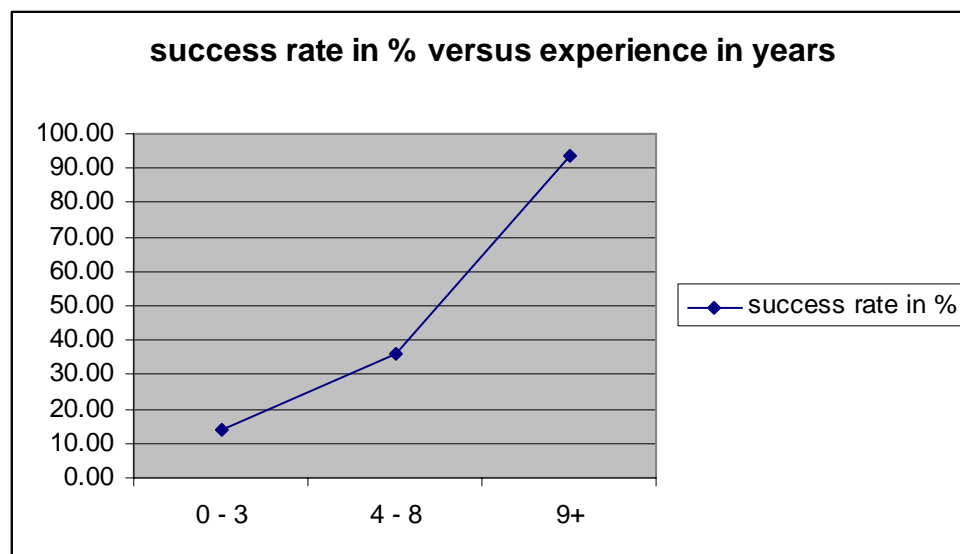
*This page is intentionally left blank*

Why is it that Internal Project Management mostly fails?

## Chapter 5: Analysis

### *Experience*

Although I couldn't find a lot of project managers with 9+ years of project management experience we can see that the success of bringing a project to a successful end is related to experience,



**Figure 10: Success rate versus experience**

We see that the project managers with more experience are using a methodology to manage their projects in a structured way. In the interviews with the project managers it was clear that when they had an official training in a method the chance in success was higher. What was clear however, was that when a method was used within several projects, the manager was changing the way the method was used from a rigid following towards a “use what you need approach”. The latter was according to the managers more useful. With the implementation of a methodology we see the implementation of two techniques, which are a must for successful project management.

Why is it that Internal Project Management mostly fails?

## Risk Management

Generally, Risk Management is the process of measuring, or assessing risk and then developing strategies to manage the risk. In general, the strategies employed include transferring the risk to another party, avoiding the risk, reducing the negative effect of the risk, and accepting some or all of the consequences of a particular risk. Traditional risk management, which is discussed here, focuses on risks stemming from physical or legal causes (e.g. natural disasters or fires, accidents, death, and lawsuits). Financial risk management, on the other hand, focuses on risks that can be managed using traded financial instruments. Regardless of the type of risk management, all large corporations have risk management teams

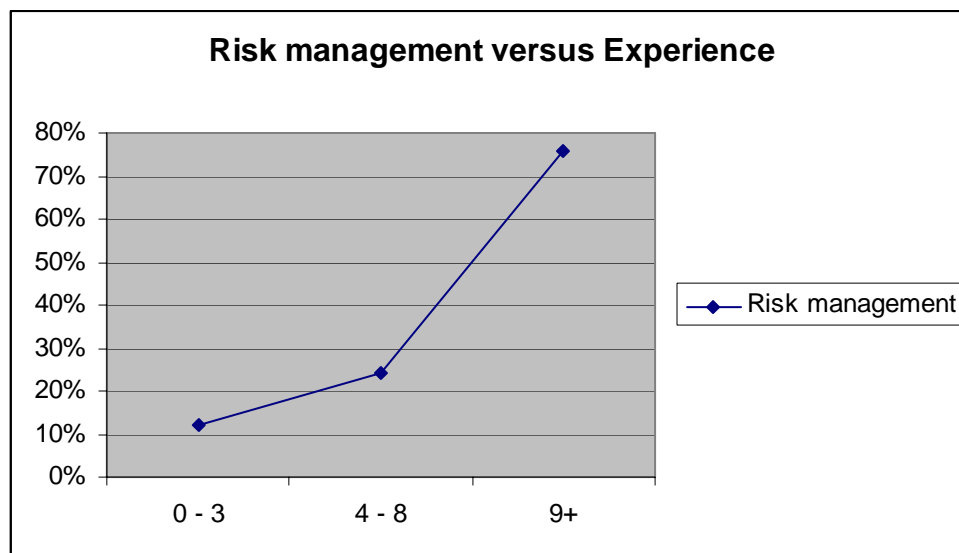


Figure 11: Risk Management versus Experience

As we can see in figure 11, the implementation of risk management follows the same line as in the implementation of a project management methodology although there is a steeper line after the 4 – 8 years experience which let us to believe that the insight in the risks of a project and the need to manage them in a structured way come with more experience over the years. What was disappointed however is that even with the most experienced of the interviewed project managers not all of them saw the need to implement.



Why is it that Internal Project Management mostly fails?

### Limitations of risk management

If risks are improperly assessed and prioritised, time can be wasted in dealing with risk of losses that are not likely to occur. Spending too much time assessing and managing unlikely risks can divert resources that could be used more profitably. Unlikely events do occur, but if the risk is unlikely enough to occur, it may be better to simply retain the risk, and deal with the result if the risk does in fact occur.

Prioritizing too highly the Risk management processes itself, could potentially keep an organization from ever completing a project or even getting started. This is especially true if other work is suspended until the risk management process is considered complete.

### Change Management

One of the biggest challenges of project management is how to cope with feature creeping. Feature creeping is the phenomenon that feature, quality, is added to the project without adjusting the resources and/or time components of the triangle of constrain, see figure 3.

The rules of the triangle of constrain tells us that if we alter one of the corners; costs, quality, and time, we must alter at least one of the other corners to compensate for the change.

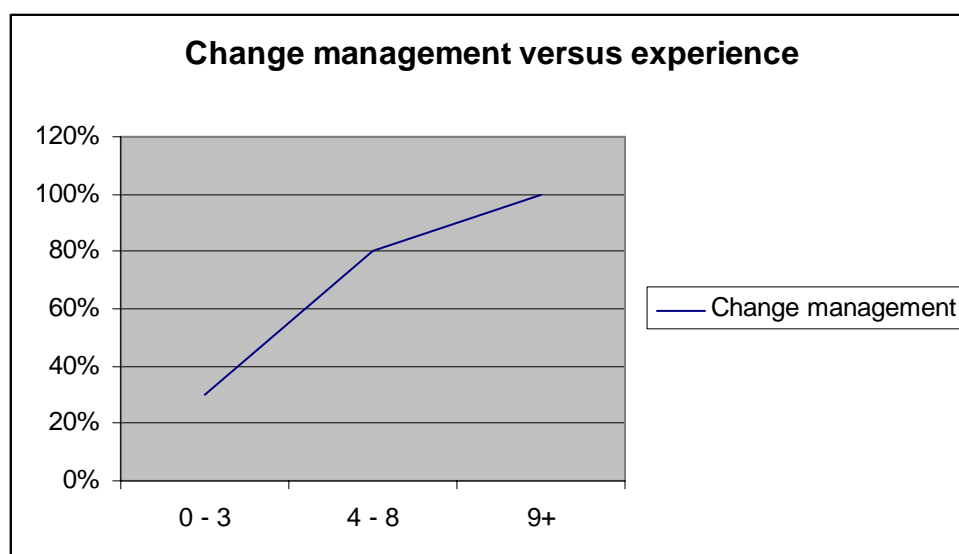


Figure 12: Change management versus experience

Why is it that Internal Project Management mostly fails?

In figure 12 we see that the curve of the implementation of change management versus experience is the opposite of the curve of implementation of risk management.

The reason for this is clear in the sense that added features have a direct noticeable effect of the project progress and the need to protect the project team is high. Although in internal, and in somewhat external, project management we see that feature creeping is done in a subtle way like, “this will only take a few minutes”, or “adding this will give us time later because...”. Most of the time this is done by the sales team with the backup of directors and the project manager is forced into a move that can cause the project to fail. The rules of the Triangle of constraints are not followed and that is always a bad thing.

### ***Estimating and Scheduling***

In the interviews came across that the estimating of cost and time necessary for the project on hand is not always straight forward. Especially beginning project managers have problems with this. In general estimating how much time a task will cost is based on historical data, previous projects done. For the beginning project manager, this is not available other than projects done by other project managers. This means that he must rely on the assumed correctness of this data. Another way of doing the estimate is to take mathematical models as described below.

In the book “The Mythical Man-Month”, Brooks describes the problem of the man-month. The time unit used to estimate effort versus time used. Brooks mentioned: “Cost varies as the product of the number of men and the number of months. Progress does not.” It is very dangerous to use a man – month (week or day) to measure the task on hand. It implies that men and months are interchangeable. This is only true if the task can be partitioned amongst many workers with no communication amongst them. This is true of picking strawberries or cotton; but in any task where communication between the

Why is it that Internal Project Management mostly fails?

team members is a necessity this fails. When a task cannot be partitioned because of sequential constraints, and then adding more men has no effect on the schedule. In software engineering this is common due to the sequential nature of debugging.

In tasks that can be partitioned but which require communication amongst the subtasks, the effort of communication must be added to the amount of work to be done. Therefore the best that can be done is somewhat poorer than a trade of men for months.

### **Intercommunication**

Intercommunication between team members increases the effort needed to complete the task. If each part of the task must be separately coordinated with each other, the effort increases with  $n(n-1)/2$ . Three workers require three times as much pair-wise intercommunication as two; four requires six times as two. If, moreover, there need to be conferences among three, four, etc. workers to resolve things jointly, matters get worse yet.

### **Algorithmic cost modelling**

The most systematic, although not necessarily the most accurate, approach to calculate the estimate in man-months in software engineering is algorithmic cost estimation. An algorithmic cost model can be built by analysing the costs and attributes of successful completed projects. A mathematical formula is used to predict costs based on estimates of projects size, number of programmers and other process and project factors. In its most general form, an algorithmic cost estimate for software cost can be expressed as:

$$\text{Effort} = A \times \text{Size}^b \times M$$

A is a constant factor which depends on local organisational practices and the type of software that is developed, Size may be either an assessment of the code size of the software or a functionality estimate expressed in function or

Why is it that Internal Project Management mostly fails?

object points. The value of exponent B usually lies between 1.0 and 1.5. It reflects the disproportionate effort required for large projects. M is the multiplier made up by combining different process, product and development attributes.

All algorithmic models suffer from the same basic difficulties:

- It is often difficult to estimate Size at an early stage in a project where only a specification is available. Function point and Object point estimate are easier to produce than estimates of code but may still be inaccurate.
- The estimates of the factors contributing to B and M are subjective. Estimates vary from one person to another depending on their background and experience.

The number of line of source code in the finished system is the basic metric used in most algorithmic cost models.

There are a number of algorithmic models that have been proposed as a basic for estimating the effort, schedule and costs of a software project. These are conceptually similar but use different parameter values.

### **COCOMO Model**

COCOMO is a model designed by Barry Boehm to give an estimate of the number of programmer-months it will take to develop a software product. This "COntstructive COst MOdel" is based on a study of about sixty projects at TRW, a Californian automotive and IT company, acquired by Northrop Grumman in late 2002. The programmes examined ranged in size from 2000 to 100,000 lines of code, and programming languages used ranged from assembly to PL/I.

COCOMO consists of a hierarchy of three increasingly detailed and accurate forms.

Why is it that Internal Project Management mostly fails?

- Basic COCOMO - is a static single-valued model that computes software development effort (and cost) as a function of program size expressed in estimated lines of code.
- Intermediate COCOMO - computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes.
- Advanced COCOMO - incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc.) of the software engineering process.

One of the most important observations in the model is that personnel motivation overwhelms all other parameters. This would suggest that leadership and team man ship are the most important skills of all, but this point was largely ignored. Researchers would rather create tools.

Personnel motivation is not part of the model. The single most important driver is software complexity, followed by personnel attributes (capability and experience, not motivation).

### ***Methodology***

In the research we saw that the more experience a project manager is the incline to use a project management methodology. This is, however, less in internal projects. From the interview we can deduct that by pressure from management there is mostly no time taken for implementation of the formal method. The deadlines are set before the project description is written.

### **Project description and Goal**

No project manager should start a project without a proper project description, we would say that this is obvious but during the interviews it became clear that most of the less experienced project managers had problems to get a proper description of what to build from the customer. Especially in the case of internal projects we see that statements like: "we want to monitor servers

Why is it that Internal Project Management mostly fails?

on the network, you know what I mean” as starting point for a project are common. Those statements are then normally followed with; “how long do you think it will take?” expecting to get a deadline for this project. The biggest problem the project manager faces is that management is expecting that he comes with a real estimate for this project but realistically he cannot even start the user requirements gathering because of no goal at all. Christopher Duncan describes in his book “The Career Programmer! (2002) how a beginner in project management can deal with this behaviour, and I feel that it is a must to read if you are coming from a programmer’s career into project management.

The first thing a project manager should do if faced with this problem is buy himself some time to write a project description as he understand it and use this as a project proposal. In reality he is leading the project as project owner. This project proposal is presented to management and on which they have to sign-off. The document will go through a number of iterations before everybody agree on the content and the time that is put in to this is not lost, although management will think this, but time invested in this will pay itself back. If consensus about the project description is reach the project manager needs to make a choice in project method; waterfall or agile.

### **Waterfall method**

In the research we saw that the waterfall method is still the most used, but is it the best for the project underhand?

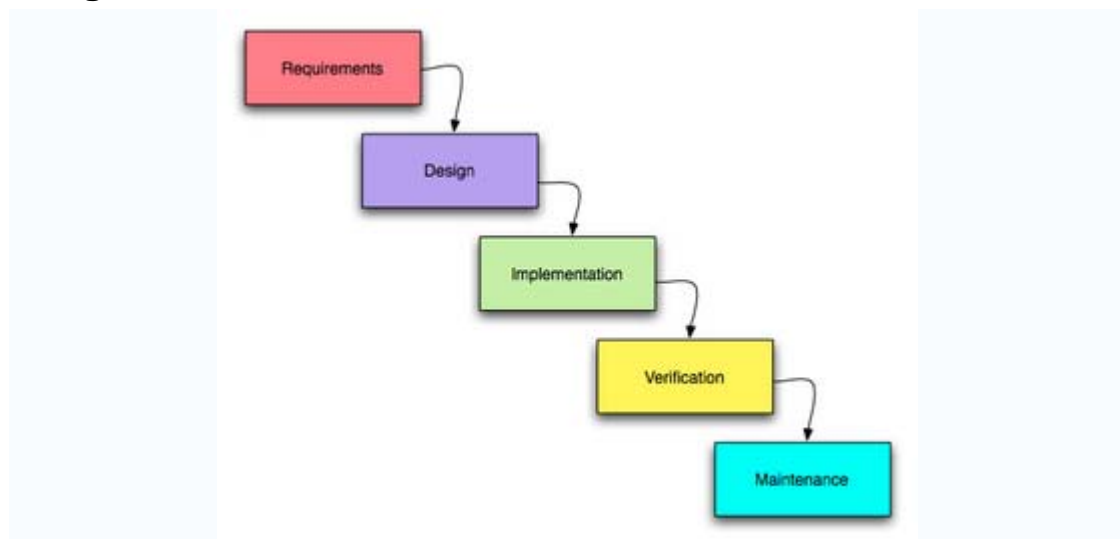
In his 1970 paper, Royce proposed what is now popularly referred to as the waterfall model as an initial concept, a model which he argued was flawed. His paper then explored how the initial model could be developed into an iterative model, with feedback from each phase influencing previous phases, similar to many methods used widely and highly regarded by many today. Ironically, it is only the initial model that received notice; his own criticism of this initial model has been largely ignored. The "waterfall model" quickly came to refer not to Royce's final, iterative design, but rather to his purely

Why is it that Internal Project Management mostly fails?

sequentially ordered model. *This paper will use this popular meaning of the phrase waterfall model.*

Despite Royce's intentions for the waterfall model to be modified into an iterative model, use of the "waterfall model" as a purely sequential process is still popular, and, for some, the phrase "waterfall model" has since come to refer to any approach to software creation which is seen as inflexible and non-iterative. Those who use the phrase waterfall model as a general insult for non-iterative models that they dislike usually see the waterfall model itself as naive and unsuitable for a "real world" process.

## Usage of the waterfall model



**Figure 13: Waterfall method**

The unmodified "waterfall model". Progress flows from the top to the bottom, like a waterfall. In Royce's original waterfall model, the following phases are followed perfectly in order:

- Requirements specification
- Design
- Construction (aka: implementation or coding)
- Integration
- Testing and debugging (aka: verification)
- Installation
- Maintenance

Why is it that Internal Project Management mostly fails?

To follow the waterfall model, one proceeds from one phase to the next in a purely sequential manner. For example, one first completes "requirements specification" — they set in stone the requirements of the software. When and only when the requirements are fully completed, one proceeds to design. The software in question is designed and a "blueprint" is drawn for implementers (coders) to follow — this design should be a plan for implementing the requirements given. When and only when the design is fully completed, an implementation of that design is made by coders. Towards the later stages of this implementation phase, disparate software components produced by different teams are integrated. After the implementation and integration phases are complete, the software product is tested and debugged; any faults introduced in earlier phases are removed here. Then the software product is installed, and later maintained to introduce new functionality and remove bugs.

Thus the waterfall model maintains that one should move to a phase only when it's preceding phase is completed and perfected. Phases of development in the waterfall model are thus discrete, and there is no jumping back and forth or overlap between them.

*However*, there are various modified waterfall models (including Royce's final model) that may include slight or major variations upon this process.

## **Arguments for the waterfall model**

Time spent early on in software production can lead to greater economy later on in the software lifecycle; that is, it has been shown many times that a bug found in the early stages of the production lifecycle (such as requirements specification or design) is more economical (cheaper in terms of money, effort and time) to fix than the same bug found later on in the process. This should be obvious to some people; if a program design is impossible to implement, it is easier to fix the design at the design stage than to realize months down the track when program components are being integrated that all the work done so far has to be scrapped because of a broken design.



Why is it that Internal Project Management mostly fails?

This is the central idea behind Big Design Up Front (BDUF) and the waterfall model - time spent early on making sure that requirements and design are absolutely correct is very useful in economic terms (it will save you much time and effort later). Thus, the thinking of those who follow the waterfall process goes, one should make sure that each phase is 100% complete and absolutely correct before proceeding to the next phase of program creation. Program requirements should be set in stone before design is started (otherwise work put into a design based on "incorrect" requirements is wasted); the program's design should be perfect before people begin work on implementing the design (otherwise they are implementing the "wrong" design and their work is wasted), etcetera.

A further argument for the waterfall model is that it places emphasis on documentation (such as requirements documents and design documents) as well as source code. More "agile" methodologies can de-emphasize documentation in favor of producing working code - documentation however can be useful as a "partial deliverable" should a project not run far enough to produce any substantial amounts of source code (allowing the project to be resumed at a later date). An argument against agile development methods, and thus partly in favor of the waterfall model, is that in agile methods project knowledge is stored mentally by team members. Should team members leave, this knowledge is lost, and substantial loss of project knowledge may be difficult for a project to recover from. Should a fully working design document be present (as is the intent of Big Design Up Front and the waterfall model) new team members or even entirely new teams should *theoretically* be able to bring themselves "up to speed" by reading the documents themselves.

However it should be noted that agile methods do attempt to compensate for this: for example, extreme programming (XP) advises that project team members should be "rotated" through sections of work in order to familiarize all members with all sections of the project (allowing individual members to leave without carrying important knowledge with them).

Why is it that Internal Project Management mostly fails?

As well as the above, some prefer the waterfall model for its simple and arguably more disciplined approach. Rather than what the waterfall adherent sees as "chaos" the waterfall model provides a structured approach; the model itself progresses linearly through discrete, easily understandable and explainable "phases" and is thus easy to understand; it also provides easily mark-able "milestones" in the development process. It is perhaps for this reason that the waterfall model is used as a beginning example of a development model in many software engineering texts and courses. It is argued that the waterfall model and Big Design Up Front in general can be suited to software projects which are stable (especially with unchanging requirements) and where it is possible and likely that designers will be able to fully predict problem areas of the system and produce a *correct* design before implementation is started. The waterfall model also requires that implementers follow the well made, complete design accurately, ensuring that the integration of the system proceeds smoothly.

The waterfall model is widely used, including by such large software development houses as those employed by the US air force, the US Department of Defense and NASA, and upon many large government projects

Steve McConnell sees the two big advantages of the pure waterfall model as producing a "highly reliable system" and one with a "large growth envelope", but rates it as poor on all other fronts. On the other hand, he views any of several modified waterfall models (described below) as preserving these advantages while also rating as "fair to excellent" on "working with poorly understood requirements" or "poorly understood architecture" and "providing management with progress visibility", and rating as "fair" on "managing risks", being able to "be constrained to a predefined schedule", "allowing for midcourse corrections", and "providing customer with progress visibility". The only criterion on which he rates a modified waterfall as poor is that it requires sophistication from management and developers. (*Rapid Development*, 156)

Why is it that Internal Project Management mostly fails?

## **Criticism of the waterfall model**

The waterfall model however is argued by many to be a bad idea in practice, mainly because of their belief that it is impossible to get one phase of a software product's lifecycle "perfected" before moving on to the next phases and learning from them (or at least, the belief that this is impossible for any non-trivial program). For example clients may not be aware of exactly what requirements they want before they see a working prototype and can comment upon it - they may change their requirements constantly, and program designers and implementers may have little control over this. If clients change their requirements after a design is finished, that design must be modified to accommodate the new requirements, invalidating quite a good deal of effort if overly large amounts of time have been invested into "Big Design Up Front". (Thus methods opposed to the naive waterfall model, such as those used in Agile software development advocate less reliance on a fixed, static requirements document or design document). Designers may not (or more likely, can not) be aware of future implementation difficulties when writing a design for an unimplemented software product. That is, it may become clear in the implementation phase that a particular area of program functionality is extraordinarily difficult to implement. If this is the case, it is better to revise the design than to persist in using a design that was made based on faulty predictions and which does not account for the newly discovered problem areas.

Steve McConnell in *Code Complete* (a book which criticizes the widespread use of the waterfall model) refers to design as a "wicked problem" - a problem whose requirements and limitations cannot be entirely known before completion. The implication is that it is impossible to get one phase of software development "perfected" before time is spent in "reconnaissance" working out exactly where and what the big problems are.

"Many of the [systems] details only become known to us as we progress in the [systems] implementation. Some of the things that we learn invalidate our design and we must backtrack." McConnell, Steve (2004)

Why is it that Internal Project Management mostly fails?

The idea behind the waterfall model may be "measure twice; cut once", and those opposed to the waterfall model argue that this idea tends to fall apart when the problem being measured is constantly changing due to requirement modifications and new realizations about the problem itself. The idea behind those who object to the waterfall model may be "time spent in reconnaissance is seldom wasted".

In summary, the criticisms of a non-iterative development approach (such as the waterfall model) are as follows:

Many software projects must be open to change due to external factors; the majority of software is written as part of a contract with a client, and clients are notorious for changing their stated requirements. Thus the software project must be adaptable, and spending considerable effort in design and implementation based on the idea that requirements will never change is neither adaptable nor realistic in these cases.

Unless those who specify requirements and those who design the software system in question are highly competent, it is difficult to know exactly what is needed in each phase of the software process before some time is spent in the phase "following" it. That is, feedback from following phases is needed to complete "preceding" phases satisfactorily. For example, the design phase may need feedback from the implementation phase to identify problem design areas. The counter-argument for the waterfall model is that experienced designers may have worked on similar systems before, and so may be able to accurately predict problem areas without time spent prototyping and implementing.

Constant testing from the design, implementation and verification phases is required to validate the phases preceding them. Constant "prototype design" work is needed to ensure that requirements are non-contradictory and possible to fulfill; constant implementation is needed to find problem areas and inform the design process; constant integration and verification of the

Why is it that Internal Project Management mostly fails?

implemented code is necessary to ensure that implementation remains on track. The counter-argument for the waterfall model here is that constant implementation and testing to validate the design and requirements is only needed if the introduction of bugs is likely to be a problem. Users of the waterfall model may argue that if designers (etcetera) follow a disciplined process and do not make mistakes that there is no need for constant work in subsequent phases to validate the preceding phases.

Frequent incremental builds (following the "release early, release often" philosophy) are often needed to build confidence for a software production team and their client.

It is difficult to estimate time and cost for each phase of the development process without doing some "recon" work in that phase, unless those estimating time and cost are highly experienced with the type of software product in question.

The waterfall model brings no formal means of exercising management control over a project and planning control and risk management are not covered within the model itself.

Only a certain amount of team members will be qualified for each phase; thus to have "code monkeys" who are only useful for implementation work do nothing while designers "perfect" the design is a waste of resources. A counter-argument to this is that "multi-skilled" software engineers should be hired over "specialized" staff anyway.

## **Agile Method**

Most agile methods attempt to minimize risk by developing software in short time boxes, called iterations, which typically last one to four weeks. Each iteration is like a miniature software project of its own, and includes all the tasks necessary to release the mini-increment of new functionality: planning, requirements analysis, design, coding, testing, and documentation. While

Why is it that Internal Project Management mostly fails?

iteration may not add enough functionality to warrant releasing the product, an agile software project intends to be capable of releasing new software at the end of every iteration. At the end of each iteration, the team reevaluates project priorities.

Agile methods emphasize real-time communication, preferably face-to-face, over written documents. Most agile teams are located in a bullpen and include all the people necessary to finish software. At a minimum, this includes programmers and their "customers." (Customers are the people who define the product. They may be product managers, business analysts, or actual customers.) The bullpen may also include testers, interaction designers, technical writers, and managers.

Agile methods also emphasize working software as the primary measure of progress. Combined with the preference for face-to-face communication, agile methods produce very little written documentation relative to other methods. This has resulted in criticism of agile methods as being undisciplined hacking (a.k.a. Cowboy coding).

Why is it that Internal Project Management mostly fails?

## The Agile Manifesto

Agile methodologies are a family of methodologies, not a single approach to software development. In 2001, 17 prominent figures in the field of agile development (then called "light-weight methodologies") came together at the Snowbird ski resort in Utah to discuss the unifying theme of their methodologies. They created the Agile Manifesto, widely regarded as the canonical definition of agile development.

“Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

**Individuals and interactions** over processes and tools

**Working software** over comprehensive documentation

**Customer collaboration** over contract negotiation

**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.”

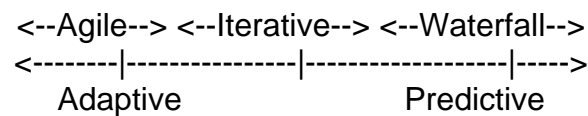
Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas, © 2001, the above authors  
this declaration may be freely copied in any form, but only in its entirety through this notice.

The Agile Manifesto is accompanied by the Principles behind the Agile Manifesto, a complete list of agile principles.

Why is it that Internal Project Management mostly fails?

## Comparison with other types of methodologies

Agile methods are often characterized as being at the opposite end of a spectrum from "plan-driven" or "disciplined" methodologies. This distinction is misleading, as it implies that agile methods are "unplanned" or "undisciplined." A more accurate distinction is to say that methods exist on a continuum from "adaptive" to "predictive." Agile methods exist on the "adaptive" side of this continuum.



Adaptive methods focus on adapting quickly to changing realities. When the needs of a project change, an adaptive team changes as well. An adaptive team will have difficulty describing exactly what will happen in the future. The further away a date is, the vaguer an adaptive method will be about what will happen on that date. An adaptive team can report exactly what tasks are being done next week, but only which features are planned for next month. When asked about a release six months from now, an adaptive team may only be able to report the mission statement for the release, or a statement of expected value vs. cost.

Predictive methods, in contrast, focus on planning the future in detail. A predictive team can report exactly what features and tasks are planned for the entire length of the development process. Predictive teams have difficulty changing direction. The plan is typically optimized for the original destination and changing direction can cause completed work to be thrown away and done over differently. Predictive teams will often institute a change control board to ensure that only the most valuable changes are considered.



Why is it that Internal Project Management mostly fails?

### **When to use agile methods**

Agile development has been widely documented as working well for small (<10 developers) collocated teams. Agile development is particularly indicated for teams facing unpredictable or rapidly changing requirements. While there are experience reports of teams succeeding with agile development outside of these parameters, there are too few experiences reported as of April 2005 to draw firm conclusions.

Agile development's applicability to the following scenarios is open to question:

- Large scale development efforts (>20 developers)
- Distributed development efforts (non-collocated teams)
- Mission- and life-critical efforts
- Command-and-control company cultures

### **Boehm and Turner's risk-based approach**

Barry Boehm and Richard Turner suggest that risk analysis be used to choose between adaptive ("agile") and predictive ("plan-driven") methods. The authors suggest that each side of the continuum has its own home ground:

Agile home ground:

- Low criticality
- Senior developers
- High requirements change
- Small number of developers
- Culture that thrives on chaos

Plan-driven home ground:

- High criticality
- Junior developers
- Low requirements change
- Large number of developers
- Culture that demands order

Why is it that Internal Project Management mostly fails?

By analyzing the project against these home grounds, the risk of using an agile or plan-driven method can be determined.

## **Criticism on the Agile Method**

Agile development is sometimes criticized as cowboy coding. Extreme Programming initial buzz and controversial tenets, such as pair programming and continuous design, have attracted particular criticism, such as McBreen and Boehm and Turner.

In particular, Extreme Programming is reviewed and critiqued by Matt Stephens' Extreme Programming Refactored.

Criticisms include charges that agile development:

- fails to provide an adequate level of structure and necessary documentation
- only works with senior-level developers
- incorporates insufficient software design
- requires too much cultural change to adopt

## **What choice to make?**

This is depending on team that is led. If the project manager has a programmers back-ground then the Agile method would be more appropriate, as I feel that each software engineering project should be run, but if the project manager doesn't have the right background that this could be difficult.

Why is it that Internal Project Management mostly fails?

## Chapter 6: Conclusion

Looking at the overall outcome of the research then we are noticing that there is no single solution this problem. In general the problem lies in experience of the project manager. We saw that the more experience project manager is using project management methods to help him steering and managing his project, the stakeholders, and project team; whilst the less experience project manager the project leads without these tools.

Due to the fact that most of the less experience project managers is not using a proper project manager method could lead to:

- Not properly estimating the work on hand,
- Feature creeping,
- Failing to make deadlines.

When a project starts the project manager needs to assess the work on hand in a way that time, cost, and quality could be correctly estimated. This is especially a problem by internal project management. The triangle of constraints tells us that the three parameters of a project: Time, Costs, and Quality cannot be in control of one person (team). Especially first time project managers fall in this trap. Instead of being able to do a proper assessment they are told what the project needs to accomplish (quality), when it needs to be ready (time) and how much they can spend (cost). In other words he is in no control whatsoever. If they are lucky then there is a functional description of the project goal, but more often that is not the case.

With this comes the second problem: Feature creeping:

During the project user expectations change and that is normal, if it is done under controlled circumstances (change management) and forces within the triangle of constraints are managed. But more often the project manager gets

Why is it that Internal Project Management mostly fails?

changes, with the comment: “cost only a few hours to do”, and none of the parameters are changed with it. This makes that a project fails.

The more experience project manager recognises this and will handle accordingly to get the project in control by introducing a proper functional specification, signed-off by the stakeholders, and change management to handle the changes on the functional specification.

This last part is more of a problem if the project is handled according the “Waterfall” method as it is by the “Agile” method. Remember that feature creeping isn’t a big thing in the “Agile” method due to the fact that the end-users, stake-holders and project members are involved in the each stage and decide together what are the features included. The time line is not altered but features will.

What we further discovered was that in internal projects the project manager is, in general, not in control of the resources assigned to the project. Most of the times they are taken from other work within the company and have still work in their “old” departments. This generates a conflict of interest in the sense that their loyalty lies with their workplace, after the project they need to go back. As soon as there are “problems” or colleagues asking for help they could abandon their work on the project, or in worse cases their manager could take them of the project, just for a few days. This endangers the project in a way that is not traceable and reflects on the project manager.

So why does it seems to be that those problems not, or less, occur in projects led by an external project managers?

It seems that this is due to two things:

1. Experience
2. Authority

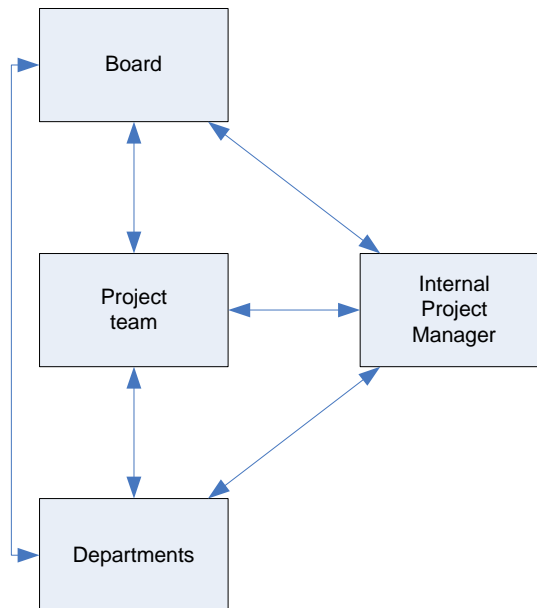
Why is it that Internal Project Management mostly fails?

As discussed before: the external project manager is “normally” longer in his profession and has more experience in leading bigger projects, they “see” problems arising before they are happening and react pro-active towards them. But besides that there seems to be a more important “skill” that an external project manager has: authority.

Generally spoken: the external project manager gets, as a specialist, the responsibility and authority to handle all things concern the project, including the hire of resources. Because the project manager is seen as an expert he has the ear of the board of directors and much of the problems, the internal project manager has, disappear. Most of the time the project, done by external project managers has the highest priority in the company, whilst projects done by internal project leader are lower on the list of priorities. As I have noticed during the interviews, the internal project manager gets the responsibility but not the authority, which makes it difficult when decisions need to be made and office politics starts to kick in.

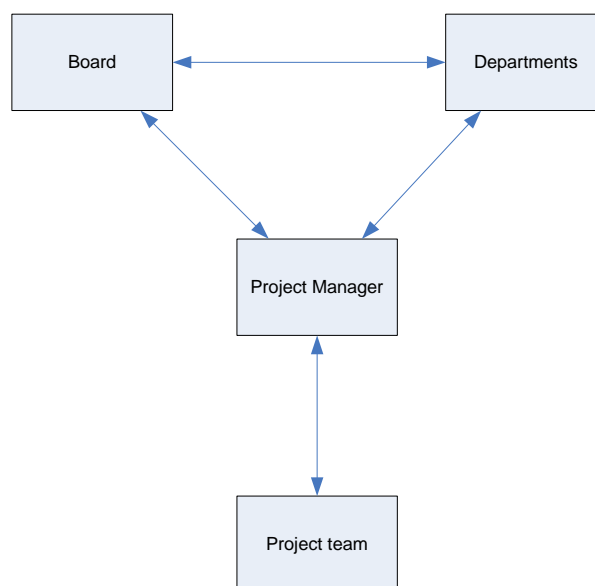
Why is it that Internal Project Management mostly fails?

## Communication lines



**Figure 14: Communication Internal PM**

Figure 14 shows the communication lines a internal project manager has to deal with and we see here that this should be of great concern. Not only flows the communication between the project manager, the Board, project team, and departments but also between the individual objects themselves with the biggest problem that this bypasses the project manager who starts loosing control.



**Figure 15 Communications External PM**

Why is it that Internal Project Management mostly fails?

In figure 15 we see the more structural project management approach, which is normally implemented with external project management with the project manager in the middle. This doesn't imply that the Board and Department are not talking with the project team member but there are no decisions made without the project manager.

### **How to avoid the failing of internal projects? (Recommendations)**

To avoid the failing of internal projects due to the causes found in the research is not easy. It demands a cultural change in the managing of companies, which start with training for the internal project manager. The problem with this is that making the choice to go for an internal project manager most of the time is made to save costs, the salary of the project manager is already a cost for the company (and most of the time NOT calculated in the project costs). This means that we must get a budget for training, not only the project manager but also the project team, if only in training in new techniques for the project. This is something that is mostly forgotten or not in the budget. We must however keep in mind that those cost will result in a better change in finishing the project and the trained project manager could be used in other projects what will spread the cost.

Further needs "normal" management hand over the authority over the project resources to the project manager and not interfere by taken resources back, even for a short time without adjusting the project deliverables (triangle of constrain). Management needs to tread the internal project manager, as was he external, with all the responsibilities and authority.

Especially for fresh starting project managers coaching would be a good alternative. An experience project manager coached the junior project manager through the project, learning on the job. This could be a hired or internal project manager, but we must take in consideration that the coach needs to have a minimal 6 years of project management experience in the area the project. Especially in the start-up of the project this would be valuable

Why is it that Internal Project Management mostly fails?

in the sense that the coach forces the junior project manager in the use of a methodology and helps with writing the first drafts of the project goal document and functional specification. After that first period a bi-weekly meeting between the coach and project manager would be enough, depending on the project size.

Choosing the right methodology is crucial, especially in software development projects. As stated before we have two main lines: Waterfall and Agile development methods. With all the pro and cons regarding the two methodologies I tend to favour for the Agile methodology especially when the project manager is from the development field, which with most of the starting project manager is the case. The research revealed that more than 90% of the internal project managers are the more senior developer promoted to do this job. Waterfall could still be used in smaller, short term projects. The main reason for this is, as discussed, that in long term projects, functional specifications and user requirements could become out of date due to several changes, external and internal.

Get from day one change management in place! This is not only for development task but for everything that has to do with the project: Documentation, User Requirements, Functional, and Technical specifications. Make sure that everything done is traceable. By making this a habit the project manager can make sure that there are no questions during the project.

Last but not least we have the point of quality. To measure quality we must have a non-ambiguous User Requirements document with the applied change documents. This together is what the goal for the project is. User Acceptance Testing can prove that what is build is in the documentation and on correct the working of it. This means that the project manager needs to make sure that the appropriate test reports are made before the testing begins. Those reports are a combined User Requirements, Change documentation and users manual. In the last document are the operating steps noted, and needs to be used as test document.



Why is it that Internal Project Management mostly fails?

**Final statement:**

Project Management is a hectic profession and needs to be managed from the start: Time, Costs, and Quality are the basic parameters that measure a project on success and failing. By applying the recommendations the project manager makes sure that he or she is in control of those parameters and not the other way around. This makes his way to a successful project possible.

Why is it that Internal Project Management mostly fails?

*This page is intentionally left blank*

Why is it that Internal Project Management mostly fails?

## References

- Adler, P. A. & Adler, P. (1987). Membership roles in field research. Sage, Newbury Park, CA:.
- Beck, Kent.(1999) Extreme Programming Explained: Embrace Change. Addison-Wesley, Boston.
- Boehm, B. and Turner, R. (2004), Balancing Agility and Discipline: A Guide for the Perplexed, Addison-Wesley, Boston.
- Brooks, Frederick P. (2004), The Mythical Man-Month, Addison-Wesley, Boston, USA
- Bruce, Andy et al (2000), Project Management, Dorling Kindersley, London
- DeWalt, K. M. & DeWalt, B. R. (2002). Participant observation. AltaMira Press , Walnut Creek, CA:.
- Duncan, Christopher. (2002), The Career Programmer, Après, Berkeley, CA
- Fowler, Martin.( 2001) Is Design Dead?. (Appeared in *Extreme Programming Explained*), G. Succi and M. Marchesi, ed., Addison-Wesley, Boston..
- Giddens, A. (1990). The consequences of modernity. Stanford University Press, Stanford, CA:.
- Malinowski, B. (1922/1961). Argonauts of the Western Pacific. E. P. Dutton, New York:.
- McConnell, Steve (2004). Code Complete, 2nd edition, Microsoft Press.
- McConnell, Steve (1996). Rapid Development: Taming Wild Software Schedules, Microsoft Press
- Newton, Richard. (2005), The Project Manager: Mastering the Art of Delivery, Financial Times Prentice Hall, Indiana, USA
- Stephens M., Rosenberg D..( 2003) Extreme Programming Refactored: The Case Against XP. Apress L.P., Berkeley, California.
- Wolcott, H. F. (1995). The art of fieldwork. AltaMira Press, Walnut Creek, CA:.
- Wolcott, H. F. (1999). Ethnography: A way of seeing. AltaMira Press, Walnut Creek, CA:.