# PLAYING THE GAME

## Managing Computer Game Development

**INTERNATIONAL EDITION**

Version 1.1

**Written by Simon Larsen**

# About the author

Born in the 70s in Thisted, in a very cold and remote northern part of Denmark. Later move to a suburb just outside Copenhagen.

Didn't do much else than play games on my Commodore 128 D throughout the 80s. In the first part of the 90s I watched a whole lot a movies, and decided that I wanted to be a film producer.

To begin with I aided in the start-up of a local cinema, which still stands this day, and then went on to writing movie reviews for the highly reclaimed movie magazine "Levende Billeder" (Living Images).

Later I graduated from the European Film College as a producer, and in the summer of 1998 founded Focus Film in Aarhus with three friends, and made a couple of short films before we sold it about a year later with a staggering profit of 750 dollars in total.

Moved back to Copenhagen, and co-founded the game development company ViaMetropol in the spring of 1999. Our aim was to create the greatest MMORPG of all time, no less. But the lack of interest from investors with deep pockets forced us to terminate the company in the summer of 2000.

In the summer of 2001 I graduated as a New Media Manager from the Business Academy of Information and Communication Technology in Copenhagen, Denmark.

Since then I've been employed as a web manager / developer at the Danish Board of Technology. Currently I'm working part-time while I'm trying the get a BSc in Business Administration and Computer Science at the Copenhagen Business School.

I my spare time I do little else than play computer games.

# Table of Contents

# Foreword

Hello,
First of all thank you for downloading "Playing the Game". What you're holding in your hands (or maybe reading on your screen) is a project long overdue.

The history of this report goes back a long way… all the way back to February 2001. That was when I started writing on my final thesis that later became the Danish version of "Playing the Game".
I made this as part of my graduation project about computer game development and production. This was the final project period of my education as a New Media Manager from the Business Academy of Information and Communication Technology in Copenhagen, Denmark.
I graduated with flying colours in June 2001, and went looking for a vacant project manager position in the local game industry. But the industry jobs where hard to come by, and instead I settled for a job as a web developer / manager.

The idea for this report came after I posted a message in the newsgroup comp.games.development.industry with the headline "What is wrong with game development today?". This message and a follow-up message got around 300 replies put together.

In August 2001 I published a portion of this report in English on my former website (www.allsimon.net/postmortem) and within three days it had been read by more than 3,800 unique visitors, mainly because GameDev.net featured it on their front page[1].
Since then I've received around a hundred e-mails from people all over the world requesting (and sometimes demanding) an English version of the entire report.

So there you have it. A year later than the Danish version, I bring you the International Edition.

The aim of this report is more to summarize than to revolutionize. You are very likely to run into something you've read somewhere before.

Thanks to all the people how visited my website in the summer of 2001, and a special thanks to all the very nice e-mails I got. I really made me want to take this further and translate the Danish version.

Simon Larsen,
Copenhagen, Denmark
September 2002

[1] GameDev.net still has the original news post at their website.

# 1. - Introduction

Every time a computer games is released it's accompanied by endless reports and stories about how disorganized and chaotic the production had been.
It's a wonder to me that an industry that a yearly growth of about 15%[2] still is plagued with bad planning.

Even though the industry make a ton of money each year, it's a know fact[3] that many project end up with red numbers in the end.
This is partly due to the very big economical investments that are normal in the industry. It's common that large productions have a productions budget of around 3-4 million dollars. That's just for production.
You don't have to be a mathematical genius the figure out that you have to sell a lot of copies to make a return on your investment. The profit is a long way down the road when a single copy only brings in a couple of dollars in real profit.
And many games never sell over 500,000 copies worldwide in their entire lifespan[4].

To examine if the bad production planning really is the very root of the problem, I analyzed 43 different Postmortem articles from Gamasutra.com[5]. The result of the analysis can be read in appendix A.
The analysis both confirmed and changed my view of the way professional game development is being carried out in the development houses around the world.

Even though the analysis shows that some productions did indeed have good planning, and a solid pre-production, the majority of the projects did not. They where trouble with either bad or no planning, or bad project managers. Some even have all of the above. The articles all stated this as one of the main problems.

One other thing that's quite remarkable is that almost all of these games went on to receive huge critical and/or gamer acclaim. So it's not so much a question of finding a qualified workforce. The problem leis more in the planning and management parts of the projects.

Another thing that all the productions share is that none seems willing to learn form former mistakes. The articles used in the analysis have all been printed in the period between the September 1999 and August 2002. So even though the information was there, none of the projects show any wish for change in the development form.

The lack of management, or bad management if you will, wears down all the people involved in the development and does not create the desired production level. More time than necessary is used correcting errors and not with what everyone really wants: to invent, develop and refine unique ideas for games.

Many mention the loose structure and the lack of control as a necessity for creative inspiration to thrive. In this report I will try to argument against this and many more opinions.

---

[2] Source: Interactive Digital Software Association (IDSA): *"2000 – 2001 State if the Industry Report"*.
[3] Source: Clayborn, Johnathan*: "Why PC Gaming Will Not Die"*. The Adrenaline Vault. March 27th 2001.
[4] Source: Wardell, Brad: *"PC Gaming As An Industry, Part I"*. The Adrenaline Vault. February 7th 2001.
[5] A Postmortem article from Gamasutra.com is usually written by one involved in the entire project. Be that the project manager, the game designer or some other high ranking project worker. Common for the articles is that they are all are usually very honest and revealing.

I will try to show that you as a game developer can save a huge amount of time in the game development process with a proper and through planning coupled together with a type of management that gives all the people involved time and space for creative work.
With the saved time comes saved money. Time is, as we all know, money.

## 1.1 – Hypothesis

Game development projects are generally badly planed and badly managed. This results in delayed productions and exceeded budgets.
The process with creating computer games is not effective enough and a huge amount of time and human resources are wasted because of this. It seems that the tendency is to "invent the wheel" every time around.

## 1.2 – Problem

**How can the process of creating computer games be more effective *without* the loss of any creative force?**
**This effective adaptation shall be seen though project management, planning and organizational reflections.**

## 1.3 – Audience for This Report

The target group for this report is everyone working, or planning to start working, in the computer game development industry.
That's a rather broad statement but it's my opinion that everyone can draw from the hopefully constructive solutions and suggestions written herein.

## 1.4 – Notes

Throughout the report I use the term "computer game", which here is meant as a technological aided game, be that either a console game, PC game, handheld game or what have you.

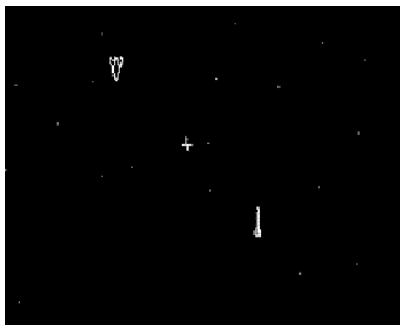
Secondly, all the people (gamers, programmers, producers, etc.) mention in this report are referred to by using the pronoun "he". This is because the writer of this report is a "he" and the majority of the people working in with computer games development are male. Why *that* is, is a longer and much more elaborate discussion that lies beyond the aims of this report.

Some of the books refereed to are only available in Danish. The ones I've been able to find in English have been listed in the reference list.

# 2. – Background

*"In the century to come, the medium producing the most dynamic, vital, and exciting new art will be…video games. Games will be more entertaining than movies, more profitable than movies, and, yes, more moving than movies. For where the moving image was cinema's bold new advantage over previous media, video games boast interactivity, an even better way to engage the emotions of the audience."*
**- Seth Stevenson, Newsweek**[6]

## 2.1 – A Brief History of Computer Games

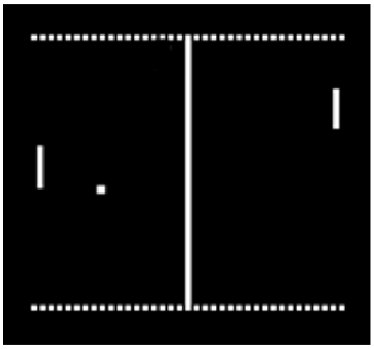


*The world first computer games: "Spacewar" (approximately 1962)*

"Spacewar" from approximately 1962 is considered to be the first real computer game[7]. It was made by some students from Harvard (with Stephen Russell leading the pack). They thought that the institutes' giant mainframe computers could be used for some other and much more "constructive" than number crunching. And from that moment computer games was born.

I the beginning of the 70s several companies started to see a commercial value in these games and began producing arcade machines that were placed in arcade halls and burger joints all over the world.

More games saw the light of day. Especially "Pong" and "Space Invaders", and later "Pac Man" were a big determining factor in bringing games to the public.




*"Pong" (left) was one of the first games to get foothold inside the family homes. For many years "Space Invaders" provided some of the lighting in the dark corners of burger joints.*

[6] Quote taken from: Interactive Digital Software Association (IDSA): *"2000 - 2001 State of the Industry Report"*
[7] Source: Game-Research.com (http://www.game-research.com/history.asp)

In the 90s the publishers and developers alike started buying each other. Those with money in the bank bought competitors left and right. The industry consolidated it self throughout the decade in gigantic corporations. Worldwide the industry now only consists of about 10-12 different publishing houses[8]. All the other companies either worked for or were owned by these.

I the mid 90s the economic value of interactive entertainment became bigger than the film industry[9]. But games have never been surrounded with the same prestige and broad cultural interest.

The games industry has grown to an enormous size. According to The Interactive Digital Software Association (IDSA) the sales growth in 2000 was 14.9%, and at the same time generated revenues of approximately 6 billion dollars[10].
The industry is the second largest exporting industry in the United States, surpass only by the weapons industry[11].

## 2.2 – The Grown-up Child

The industry has matured no doubt about that. From small garage-like code shops it's become a full-fledged industry with all the pros and cons following.

But one place where the industry not has matured as much is in the way the projects are managed and the type of control that is applied. This is quite remarkable since the productions' budgets and size have grown as much as the popularity of the games. A common production costs around 4 million dollars from start to finish (see appendix A).
It's not because the industry is home to incompetent people that can't find employment elsewhere, but the projects all show sings of rushed and/or unintelligible decisions.

The computer games industry has developed parallel with the traditional software industry. A possible reason that the traditional software business has had more success in controlling the productions might be found in the large financial investment the business has seen through the years. These kinds of investments are rarely seen in the games business.

The following chapters seek to provide an optimal platform for future game development projects, and hopefully contribute to bringing the escalading development cost down, or at least stabilize them.

## 2.3 – The PTO Pyramid

PTO stands for Project, Team and Organization. These elements are inextricably bound together. All the elements must be present at a given time to carry out a project.

[8] Among those: Infograms, Electronic Arts, Activision, Microsoft, Eidos, THQ, Sega, Nintendo, Vivendi Universal, Ubi Soft.
[9] Some say it became bigger, others still insist that the film industry is the top dog. Source: Tapscott, Don: *"The power of electronic play"*. Computerworld, May 1999.
[10] Source: Interactive Digital Software Association (IDSA): *"2000 - 2001 State of the Industry Report"* (2000) and *"Essential Facts about the Computer and Video Game Industry"* (2002).
[11] Source: Tapscott, Don: *"The power of electronic play"*. Computerworld, May 1999.

The team is in it self “just” people without any specific aim or goal. Organization can be described in very detailed and minute forms but if doesn’t contain any people no work will be done. The project obviously can’t be carried out without the team. Every element binds closely together in what will here after be referred to as The PTO Pyramid.


*Figure 1. The PTO Pyramid: Shows the mutual dependency that exists between a Project, the Team and the Organization as the foundation. (Source: own production).*

The structure of the pyramid shall not be seen as a list of how important the individual elements are. All the elements are equally important. The structure shall be seen as the inextricably dependency as described above.
It’s more important to look at if the elements fit each other. The organization must fit the team and the team must fit the project. The project must fit the team and the team must fit the organization. The dependency is unavoidable. It must be a solid pyramid.

In other words, it’s no good to hire experienced team members if you stuff them into an organization where they can’t to their work properly. Furthermore it must the right team with the right skills to produce an end product with a satisfying result. Everything must form a synthesis.

Each element will be treated and debated separately in the following chapter in the order of Project, Team and lastly Organization.
It will be description of the problems when they emerge but also a journey through the entire structure and foundation of the company back to what often causes the problems, the team and the Organization.

The element “economy” has deliberately been left out of The PTO Pyramid, because this element is vital to all elements. Nothing can be done without money.

# 3. – The Project

*"The one who figures on victory at headquarters before even doing battle is the one who has the most strategic factors on his side. The one figures on inability to prevail at headquarters before doing battle is the one who has the least strategic factors on his side. The one with many strategic factors in his favor wins, the one with few strategic factors in his favor loses – how much the more so for one for one with no strategic factors in his favor. Observing the matter in this way, I can see who will win and who will lose."*
**- Sun Tzu: "The Art of War"**[12]

You can't open a book about project management without running into *"Think Big – Start Small"* and *"If you're failing to plan, you're planning to fail"*. It is on page 1 in every modern book on the subject. They are so elementary and simple that you'd expect them to be memorized by every project manager. But it's exactly these sentences that's forgotten when the project is rolling.

There, now I've properly insulted every project manager in the computer games industry. Of course this is a generalizing and in real projects there is used a lot of time on the planning. But the purpose of this chapter is to give examples on how to plan and guide the project from start to finish and try to take care of all the possible problems that might arise. Hopefully you would be able to prevent making too many temporary solutions.

## 3.1 – The Role of the Project Manager

What kind of project is it that we're making? The aim of the pre production phase is to define what propose and goal the project has. What is it that needs to be solved, what requirements must be met and how will we do it.

In the beginning it's important to determine whether the goal of the project is to verify or to validate the final product. There's a proportional big difference between to two, without either one excluding each other. You should always strive to verify your project, to make the final product right. But it's equally important to validate, to make the right product. See the difference?
There's nothing gained in making the *technically* most advanced games in the world with groundbreaking graphics if the game in the end is no fun to play.

It the project manager's role is to ensure the first, the verification, but other than that he plays an instrumental role in allocating time and resources to the continual validation of the product.

## 3.2 – Project Uncertainty

As a project manager you should always be aware of the fact that projects change. It's often said that projects gets a life of its own. If you're not attentive to the fact that projects change when more information become available for the project, you could risk grid locking yourself in a type of management that entirely useless and uncalled for in a given situation.

[12] Quote taken from "The Illustrated Art of War" (1998).

### 3.2.1 – Contextual Uncertainty

Christensen & Kreiner (1991, p. 41) has described what they refer to as the contextual uncertainty. Contextual uncertainty derives to the dilemma that in the starting phase you have little or none knowledge (experience) and information availed to make the decisions you're more or less forced to make. Towards the end of the project the collective knowledge and information are considerable larger and the importance of your decisions so much the smaller.

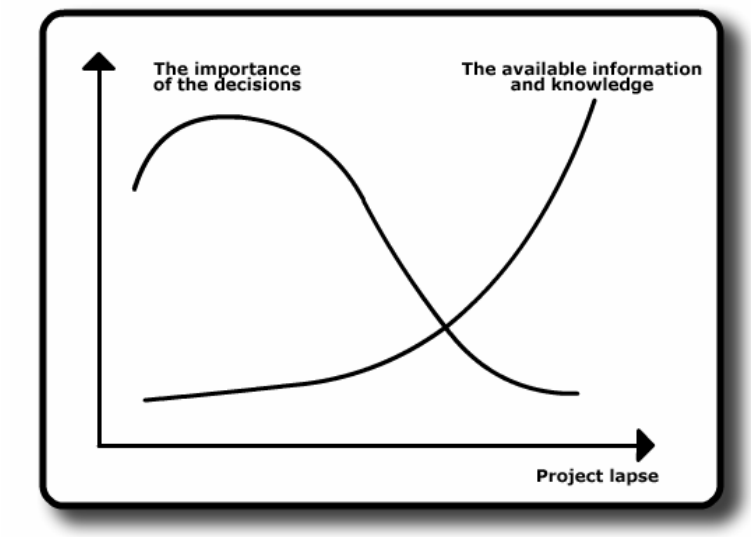



*Figure 2. The contextual uncertainty. The importance of the decisions compared to the available knowledge and information. (Source: Christensen & Kreiner 1991, p. 41)*

It's precisely because of the contextual uncertainty that it's necessary conduct a midway evaluation of the projects aims and goals. You could call it a midway validation. Otherwise you might end up with a product that has solely verified the project goal. You could work with blindfolds on for years and in the end create a product that's technically obsolete[13].

Decisions made in the beginning of the project can whether to produce the final game for the PC or the Gameboy. To entire different projects, and this decision will have great impact on the project. Later, towards the end of the project, the decisions will be more along the line of how big the buttons in the game menu shall be.

It's not just the project manager's decisions that are marked by the contextual uncertainty. It's the entire project team's. The team's knowledge and information will also expand considerably throughout the project.

The contextual uncertainty is retrospective, which means that it refers to an already passed period. It is therefore not possible to look at the contextual uncertainty in the beginning of a project. Changes in knowledge and information are not possible to analyze before later on.
This is also one of the reason some projects planning slips. If the project manager and/or the management in general insist in keeping the original project plan and ignore the newfound information, it can result in a project that's completed on a wrong foundation.

[13] Take the now closed Ion Storm as an example. They relied too much on John Romeros' legacy in creating the game "Diakatana". The game was delayed by months and when it finally came out, it was an utter disappointment, both in terms of technical execution and game play.

Neglecting the importance of the contextual uncertainty is properly one of the main reasons that so many of the projects in Appendix A where delayed.

### 3.2.2 – Operational uncertainty

A projects operational uncertainty is the uncertainty you encounter when launching a project. The knowledge and information that's available to you to effectively do your job. It's logical that the more knowledge and information you have about a given task the more effectively you can solve it[14]. You must therefore do as much research and preparation and possible before embarking on your task. It's through this preparation that you can bring the operational uncertainty down.
Many of the analyzed cases from appendix A stated that they never looked at the operational uncertainty, and didn't have an overview of the knowledge and information at hand. For some it resulted in surprises during the project and delayed the entire production until a useful solution was found.

Many think that to minimize the operational uncertainty you must invest in the best solutions money can buy. Figure 3 shows a graph of how you can view the effectiveness of a programmer in an economical perspective. To the left we have the very new programmer with no project experience at all. He's cheap, no doubt about that but he'll take longer to complete his tasks and the end result might not live up to the standard of the project. On the other side we have the top programmer with years of industry experience. He's extremely expensive to hire but always delivers top quality work.
Does it make sense to hire this top programmer just to do a simple menu system or is it a waste of money? Are the current project members capable in doing this job if they're just allocated some extra time? You have to strike the golden mean. It's not always worth the money.

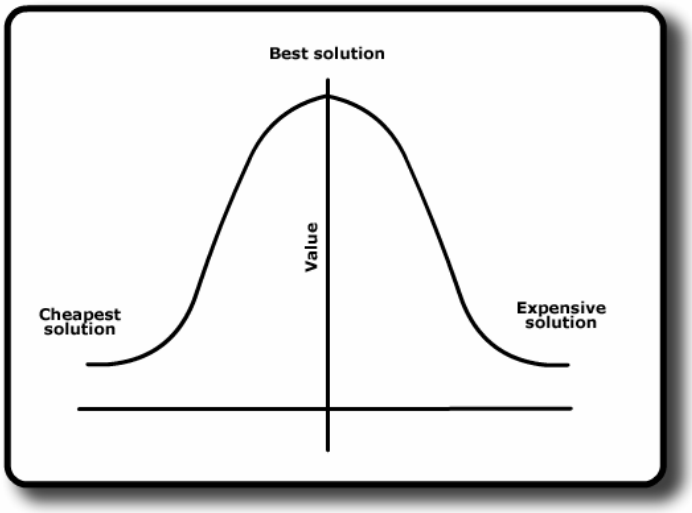



*Figure 3. The effectiveness curve. The most expensive solution is not better by default. (Source: Ahearn, Luke: ""Budgeting and Scheduling Your Game". Gamastura.com, May 4 2001).*

The only time where it's always worth your money in when you're investing in development hardware. Here the most economical effective solution will in most cases be the most expensive. There's absolutely nothing gained in buying old or used equipment that crashes

[14] Please note that I here talk about effectiveness and not about quality. It is two different things, that shouldn't be interchanged.

or can't run the production programs properly. Everyone in the team works faster and more effectively on newer hardware. It even turns out happy team members (everyone likes new hardware) and happy team members are effective team members. Another gained benefit.

## 3.3 – Risk Management

To determine what type of control the project manager must use to complete a project with the fewest possible errors, John K. Christensen's elaboration of F. Warren McFarlan's theories about risk dimensions come to good use (Melander (ed.) 1993, p. 148-154). Table 1 shows projects divided by size, structure and whether it's known technology[15] or not.

| | *Very structured* | *Less structured* |
|---|---|---|
| **Known technology:** *Good knowledge and experience with the work method and work form.* | **Large project** - *low risk* | **Large project** - *low risk* (but easily influence but weak management) |
| | **Smaller project** - *very low risk* | **Smaller project** - *very low risk* (but easily influence but weak management) |
| **New technology:** *No or little knowledge and experience with the work method and work form* | **Large project** - *medium risk* | **Large project** - *very high risk* |
| | **Smaller project** - *medium / low risk* | **Smaller project** - *large risk* |

*Table 1. The eight project types according to McFarlan and John K. Christensen (Melander (ed.) 1993, p. 154).*

### 3.3.1 – Technology

Through a closer review of what technology (knowledge, work methods, etc.) you have at hand you should be able to position your project within one of the frames.

Production of computer games fits somewhere between new and known technology. It's not the first time someone is trying to make a computer game, but there will inevitably appear situations that require innovative solutions in areas where few have treaded.

One way to make a huge reduction in risk is to hire experienced team members. Members with either experience with the production hardware or experience with the type of project. By doing such a simple selection the projects risk is brought down from *very high risk* all the way to *low risk*.
This will not be possible in practice because the financial resources that limit the management in exclusively hiring top professional team members. It wouldn't be financial feasible.

[15] Here the word technology is used in a very broad sense. It covers knowledge, information, hardware, work methods, etc.

But as it shows from Christensen's model even projects with known technology is very easily influenced by a weak management if the structure also is loose. Risk is an element that casts its shadow over every part of a project. Finding the best work place, buying the best equipment and hiring every top skilled developer in business is not going help much if the management (both the company's and the project's) is incompetent or weak. Weak meant as not visible or conflict shy and therefore looses control with the project.

### 3.3.2 – Size

There's no question about that computer game projects belong in the *large project* category because of the sheer size of the production budget, number of team members and the length of the production time. Just look at the graphs in appendix A, the projects *are* large. Dividing the project into smaller sub-projects can reduce the size of a project. This modular form of working with the project has the distinct advantage that the different sub-projects become very manageable.

But as always, advantages don't come free. It's very difficult, if not impossible, to iterate between sub-projects once their finished. If you do, it'll delay the project notably.

You can work around that by buying finished or half finished tools or elements from outhouse sources. The development company id Software[16] has great success with licensing parts of their game engine to other companies, which then builds their own games "on top" of this engine.

### 3.3.3 – Structure

You should as project manager at all times try to create as much structure as possible and by doing so reduce the risk involved with the project work. If you take a rational approach to the project and seek to minimize the uncertainties listed below you're on your way.

From this you can work towards a form where the uncertainties are reduced significantly. Structuring and making the following uncertainties clear will reduce the risk:

- A clear definition of goals.
- Clear and synonymous formal organizational structures.
- Support for the entire project from the management and the board of the company.
- Detailed and thoroughly planning and control (follow-up and evaluation).
- Information and communication routines.
- Qualified project managers and team members.

(Adapted from McFarlan, Melander (ed.) 1993, p. 142)

They seem relatively simple and self-explanatory but each one of them can in this absence tip the project in the wrong direction. At the same time they can be guidance to making the project a success. You should make curtain that all the stated points are fulfilled and the project are frequently evaluated to make sure if there is need for adjusting the work form.

By analyzing your project's risk you can ask yourself if the project is *worth* the risk. If you're a new founded company and concluded through your analysis that your coming

[16] id Software is known for creating games like the DOOM and Quake series. Especially the engine used in Quake III has been licensed to numerous companies. Games like "Star Trek Voyager: Elite Force", "Jedi Knight II: Jedi Outcast", "Soldier of Fortune II – Double Helix", "Medal of Honor: Allied Assault", "Return to Castle Wolfenstein" have all been made using the Quake III engine. Read more about id Software at www.idsoftware.com.

project falls into the very large risk category, it might be time to reassess if you with other means can bring down the risk. Otherwise the decision might be to postpone the project until the team members and project managers alike have enough experience to embark on the project.
Until then there's more sense, business wise, in doing something more tangible and smaller to get the experience necessary.

It's here that the risk management bursts through. There are always things that you can't guard yourself against. There is no point in waiting a year before the local top programmer gets time off to stop by. In these cases you should "take what you can" and just be aware of the increased risk that's involved with doing so.

Some might argue that the theory of the contextual uncertainty is inconsistent with the points just mentioned here. But if the "new" and changed goals you get from looking at the contextual uncertainty are made a visible as the original goals where, there's absolutely no increase in the risk because of this. On the contrary the project now has a more solid and realistic foundation, that'll lower the risk in the longer run.

## 3.4 – Managing tools

When working with projects with high risks and uncertainties there are different arrays of managing tools that you can apply to make everything flow easier, so to speak.

There are the external and the internal integrations tools and there are the formal planning tools and lastly the formal control and evaluation tools (Melander (ed.) 1993, s. 156).

### 3.4.1 – External Integration Tools

The external integration tools comes into use in when it's time to include the end user in the game production to make sure that the end product can be used properly. That happens with the use of formal and informal tests of the alpha and beta version of the game[17].

One tool that is getting a lot of focus at the moment is online communities. Communities where coming gamers (users) can express their wises for features they like to see in the final game. It has the obvious advantage that you get bombarded with ideas and suggestions, some better than others, but some that you as a developer might not have thought of. The disadvantage is the same as the advantage. Because of this bombardment of ideas it demands quite a lot of work from your side to get the community to function and prosper. Some of the larger online communities have over 30.000 active members[18].
The integration happens when you reward the users with the best suggestions with for example putting their names into the credit list in the final product. Imaging the responds you would get from a teenager when he sees his names roll over the screen with a "Special Thanks To…" right in front of it. It most certainly would make his day.

[17] Alpha version: an unfinished game that only works in the development environment. Shows the games basic structure but is mostly seen as a technology test to make sure that the different parts of the games can operate together. Beta version: A nearly finished version that works outside the development environment. Beta tests try to uncover bugs with the technical aspects of the game but more so with the game playing aspects.
[18] See for example the Unreal Tournament forum at www.forumplanet.com/planetunreal

### 3.4.2 – Internal Integration Tools

Because computer game development are, as defined earlier, high risk projects with a high amount of contextual uncertainty it is exactly the internal integration tools that can make or break a project.

As mentioned earlier it is very important that the goals for the project are being communicated to everyone. Everyone doesn't necessary have to same understanding of what the goals is but everyone should be aware of them. An art director might see beautiful graphics in the goal "The best strategy game", whereas a programmer might see bug free code in the very same goal. It is the project manager's role to bring this vision to everyone.

It is even more important to hire a project manager with experience within the field when you are dealing with high-risk projects. Not that he has to have experience with big scale computer game development, but experience with working with the mentioned uncertainties and risks.

Communication is the keyword when taking about internal integration tools. Status meetings and daily contact with every team member all contribute in creating a working environment where information flows freely and is available for everybody.
For such a purpose it will be an advantage to create a company intranet. On the intranet all information about the project should reside (Project plans, vision documents, meetings reports, design documents, staff lists with phone numbers and e-mails, etc). As little as possible should be kept secret for the production team.

The intranet can also be the place where individual team members can suggest solutions and changes to areas outside their own. It is important that this kind of communication is as informal as possible. You can always have formal meetings. It is hard to find it relevant to hold a big management meeting for something that could have been solve over a cup of coffee.

When planning a type of project as describe within this report, it is close to impossible to do so without the involvement of the different team groups (programming, art direction, sound, etc.). When the design paper nears completion you gather up the entire production team and in cooperation create a production schedule with milestone and deadlines. This is a way to avoid unrealistic deadlines created by a sole project manager.

There is no better at setting deadlines than the ones working with the subject daily. Everyone is interested in getting the project finished and working professionally. After that it is the project manager's job to tie the strings together and make the overall planning for the entire project.

### 3.4.3 – Formal Planning Tools

The literature is packed with books about formal planning tools. These tools come from the traditional form of project management, and it is these tools that are often seen as being stiff and rigid.
They *are* stiff and rigid but that is their advantage. Where we before talked about open communication and the softer values in terms of the integration tools we here have some more hard tools that can be used to keep the project on the right track.

A very classic planning tool is the Gantt chart. Everybody can track their progress with a Gantt at hand and see what milestones that lie ahead.
It can be use to see who have been made responsible for a given part of the project (ex. creating a menu system) and it will list the tasks that are dependant on other tasks.
It is no idea xeroxing your Gantt chart and handing it out to everyone saying, "Now you'll know what to do the next two years". For that, the Gantt chart is much to abstract. It has to be visible to every team member but it should never be considered as a to-do list. You create a Gantt chart to create an overview.

What you do instead is divide every "beam" in the Gantt chart into many smaller and more manageable parts. This method is called Work Breakdown Structure or WBS for short. For example: "Graphics to main characters" will be divided into the different main characters and further down to sketching, drawing, colouring, animation etc. You keep on dividing the work areas until you have reached a working schedule with tasks no longer than a couple of hours work.

The advantage for each team member is that they get an excellent view over what things that needs to be done. It also has a great psychological effect that the entire team experiences the progress in the project when you during a single working day can cross out 4-6 things on your to-do list.

Please note that this advantage could loss its effect if every team member is handed a list of tasks for the next one or two years. The lists should not contain more work than for 5-6 working days. Every thing else is simply too overwhelming and also very stiff to work with if (or when) changes occurs.

Besides the WBSs you should supplement your Gantt chart with milestones. The traditional literature describes milestones as the Holy Grail of project management. And it *is* a very important tool; just try not to make it into anything more than it is: project bearings.

But create them nonetheless, and it is vital to make them as tangible as possible. A milestone with the describing "Level ready for testing" might seem concrete, but consider who is working in out team. Later we will look into the enormous diversity in skills and background the people working in the game development has. It's what defines the industry.

You cannot expect an art director and programmer to read this milestone the same way. The programmer might read the milestone as a code test that will take him 7-10 hours to prepare to. Whereas the art director might see it as an aesthetical test of the graphical elements, something he will be around a week to prepare for.

The milestones shall be as concrete and unambiguous as possible. If that's not possible it should be made clear whom the milestone is for. Whether it is the programmers, art directors, designers or the sound crew.

Now our good friend the contextual uncertainty shines his head again. The formal planning tools help bring down the operational uncertainty. The contextual uncertainty is as mention retrospective and therefore it is necessary to adjust and follow up on you production plans.

It is also much better to go to the management and tell that you're going to be a week over schedule a couple of times than it is to come to the very last moment and tell that you are

running nine months behind. The total delay might very well be exactly the same, but it gives a clear sign of your overview and acknowledgement of the organic development of a project.

This shall of course not be seen as loophole for drawing the project out in eternity, until everyone is 110% satisfied with their work. The perfect game does not exist. But the good, though out and polished games does.

Take a look a figure 4 below. It is The Impossible Triangle. The element “quality” refers to the quality of the final game, but also the quality of the size and scope of the project. “Resources” refers to the resources you have both in terms of money and workforce. The element “Time” is self-explanatory.

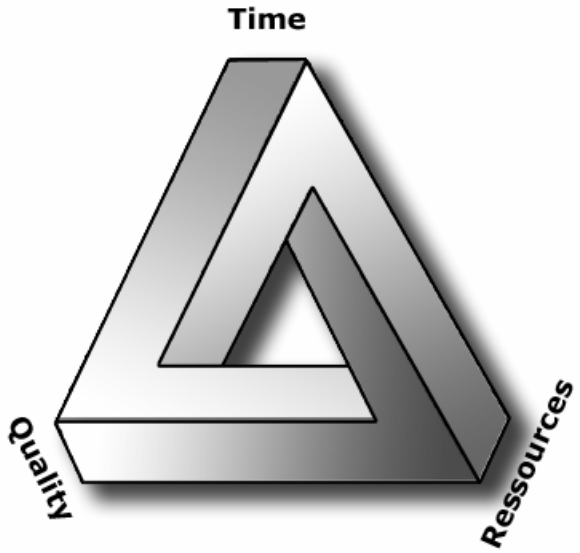



*Figure 4. The Impossible Triangle. You cannot change one of the elements without affecting at least one of the others. (Adapted from Rollings & Morris 1999, p. 306).*

If you don’t adjust your time schedule it will affect the quality of the team’s work. If you do not want that to happen you would have to ad more resources in terms of overtime payments, hiring more staff or outsourcing tasks to outside companies.
Overtime isn’t always a good solution. You can always question if a team member does the same amount of work in a 16-hour workday as a team member that dividends the same work into two days.

If you have a fixated deadline, for example a Christmas game, it can be necessary to lower the scope of the games (the size quality). It is by al means better to make a smaller quality game that a bug ridden game that’ll be returned to the stores in January.

### 3.4.4 – Formal Control and Evaluation Tools

The formal control and evaluation tools also come from the realm of classic project management tools, but they are not quite as stiff as the formal planning tools have a tendency of.
The designation covers all the tools that are used to make sure that the project is still rooted in the overall organization.

The different sub groups in the project hands in frequent status reports showing the progress in the project work. These status reports can also lead to so-called presentation meetings where every sub group show what they have been working on since the last meeting.
Through such meetings every team member, and the management, gets a clear indication of how far you are in the project. The meetings can even become a moral catalyst for the entire team as you experience the forming of the final game.

To keep the relevance of the overall productions plan and prevent it from becoming a still picture of how things *used* to be, it's important that all the team members have the option to tell the project manager about changes. It is, as mentioned, better to tell about changes in the in plan continually than reveal at the end that the plan by no means is realistic.
It is important to have an informal and open channel for such reports otherwise you risk not getting them.

On the other hand it is at least as important that these reports don't go "unpunished". If it is the team member's own fault that the schedule can't be kept? If other things, such as illness etc., caused the slip you would have to change to plans to deal with the new circumstances.

That leads to another and important factor; the exchange of knowledge or knowledge sharing, as it also called. As mention earlier it a necessity to have an intranet where every team member can find information about the project.

There exist a couple of programs that can assist a project team with keeping track of all the different elements a game consists of. It is no uncommon that you have created over 15.000 different files during production. Needless to say that such an amount could create chaos if not handled systematically. It is important with a central system that keeps track of who is working on what and when.

Microsoft has made program called Visual SourceSafe[19] and NxN Software has made a similar program called Alienbrain[20]. They are two of the more well known but there are a wide variety of other similar programs available. Which you prefer is more a matter of taste.
Many have the misconception that these programs only apply to the programmers' work. This is a *big* misconception. Everyone, from project manager to art director and testers, benefits from using programs like these.

They each have their advantages and disadvantages but common for all of them is that you can store graphic files, text documents and code fragments side by side. It a huge advantage for the project team that you at all times can see who is working on what and at the same time prevent that you accidentally overwrites or deletes someone else's files.
They even have the advantage that others can continue to work on others' tasks if the project is struck by illness among the team members.

A thing all the projects analyzed in appendix A list as a problem is undocumented code. When a programmer writes code it unfortunately often happens that no notes are written to

[19] Read more about Microsoft's program Visual SourceSafe at msdn.microsoft.com/ssafe
[20] Read more about NxN Software and their program Alienbrain at www.nxn-software.com

accompany the code. That makes the code practically unreadable by anyone else than the originator.
This is a situation that you should go great lengths to avoid. Not that the programmers need to write manual to their code, but it is important that others can approach the work in case of illness, time pressure, etc.

## 3.5 – Chapter Conclusion

What must a project manager do to bring the project in on time and not too much over budget? With two words: work dynamic. He should be aware of the contextual uncertainty and at the same time bring the operational uncertainty down by using the formal planning tools.

He should surround himself with talented people. This is an important issue that will be elaborated more in the next chapter. Mature and skillful team members that dare and understand when to take responsible, even if not given directly.

He should be a good judge of character. He should be a communicator that brings the team together without using a whip to do so. He should be aware of all his tools and know when to use them. When to be bossy and when to support the team.

He should be aware of The Impossible Triangle, but not see is at a problem but as a challenge. Not just for him but for the entire team.

Lastly, but not least, he should have a basic knowledge of all the aspects of game development. Not that he necessary have to be able to program in C++ or animate in Maya, but he should be aware of the way the different team members work. This is an invariable rule for project managers working with game development. Knowledge and respect for the different trades.

It also makes it a lot easier for the project manager to adjust the schedules if he is aware of the different working routines. He can even harness respect by not looking like he just fell from the moon when an animator proudly claims that he just pushed 100 more polygons into the main character without lowering the FPS.

# 4. – The Team

*"No one is bigger than the band. Making a game involves an entire team of talented and dedicated individuals. No one person makes any product; much like a Super Bowl-winning football team is not all about the quarterback, a game is not about a programmer or designer. A good team works well together, hangs out occasionally, and acts like a well-oiled machine."*
**- Cliff Bleszinski, Epic Games**[21]

Now we are at the T in the PTO Pyramid: The Team. In this chapter we will try and work toward a foundation, where both the team members and the project thrive, without restriction the creative development.

## 4.1 – Passion for Games

It goes without mentioning that people working with games have to be creative. That goes for everybody, from art directors to programmers.
But creativity won't always be sufficient. It will to a high degree also be required to have a passion for games. The optimal team consist primary of people that love computer games. Really *love* them. Knows what game play elements that work and what don't.

The development of computer games attracts a very diverse crowd. If there is anything the game development industry is recognizable by, it is the enormous diversity that is in personalities, talent and knowledge. Everything for a shy nerd without a shred of social skills to a rule fixated Excel-type project manager. They are all there, even though is usually people in "the middle" you come by.

If the project manager does not posses this passion himself, he will not have a good chance in making the project flow, because interests and visions often will go in opposite directions. Fancy budgets and tight schedules won't do it. The passion *must* be the driving force, even for the project manager.

How else will he be able to understand that the development of a feature that allows players to check up on their online characters in your new MMORPG from their mobile phones could benefit the game's sales potential? Seen from the gamers' point of view it would be a cool feature that will enhance the game experience considerably. Seen from a project's financial point of view the feature would be of less importance since it doesn't directly concern the games as a whole. To the contrary it would demand long and extensive development phase before it can be implemented successfully in the game. It is two-edged sword the project manager here is holding, and unless he has a "background" as gamer it can be very hard (if not impossible) to see the difference in such a circumstances.

## 4.2 – The Communication Process

Something that unfortunately has plagued the game development business in almost all of its lifetime is that the project team often consists of "ten kings and an emperor". Meant that the team members' egos sometimes are bigger than the project itself. The pride of your trade and the ignorance of the same is often the cause of broken or bad communication.

[21] Quote taken from Saltzman, Marc (ed.): *"Game Design: Secrets of the Sages (2nd edition)"*, p. 17.

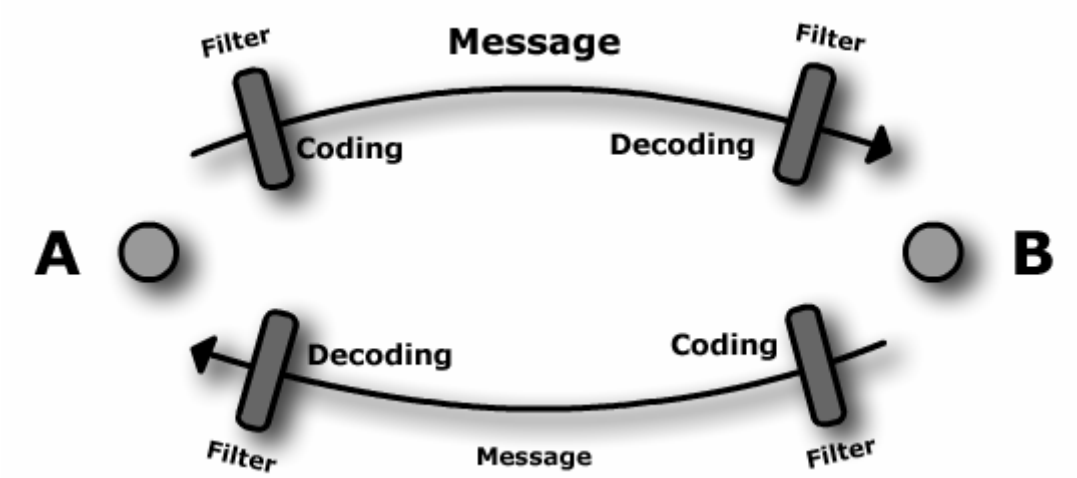



*Figure 5. The Communication Process. You should be aware of the different filters people have to code and decode their communication. (Adapted from communication expert Elisabeth Plum).*

It is important to know each other's filters. We perceive everything differently. And we perceive everything very differently when we are wearing our professional glasses.

An example from figure 5 could be that, A, a programmer asks B, a graphics artist, to make some small files that he can implement in his newly developed menu system. When the programmer (A) sends his message he codes it by saying "small" files. The graphics artist (B) decodes it and figures that his big 200 MB files should be reduced to about a 1/20, to approximately 10 MB. The graphics artist says he will be done in a minute. Half a day later he returns with the files to the programmer that immediately tears his hair out. He asked for "small files" expected to get files with a size of around 40-60 KB. But the graphics artist heard "small files" with his filter on, which meant around 10-15 MB.

Whose fault is it that the communication went wrong here? None of them, and both of them. They could both have saved the graphics artist the wasted work and the programmer's sudden lack of hair by being aware of each other's professional filters. Had the programmer clearly stated that "small files" was files with a size below 100 KB, or had the graphics artist known what programmers really mean when they say "small files" the problem had been avoided.

If it is an entirely green project team, some bigger decisions and messages can pass though the project manager and by that give him the responsibility to communicate in onwards. Professional or more experienced team members will possess a better overview and are often aware of the different professional filters that exist in a project group.

The communication expert Elisabeth Plum[22] describes it as going from high context to low context communication. Which means going from a form of communication where body language, (professional) knowledge and culture have a great role in understanding what is being said, down to a form of communication where the word themselves carry the meaning.

[22] Read more about Plum at www.plum.co.dk. As of now (August 2002) the website is unfortunately in Danish only.

## 4.3 – Maturity

Another thing that can slow the process down is the lack of maturity. Not just among the team member but also with the project manager. It is regrettably something that is characteristic for the industry.

The programmers frequently get blamed for being immature and it is often so because of their work methods they tend to put aside their social skills. But the phenomenon isn't confined to the programmers. Far from. Every branch of game development houses these bad elements.

But what do you do about them? You identify them and take them of the project. Preferably as early in the process as possible. It only takes *one* team member to bring down the entire project.

Rollings & Morris describes in their book "Game Architecture and Design" (1999, p. 171-176) some arch types you should try to avoid having in your projects.

That includes the great individualist that might possess great talent and knowledge but is the exact opposite of a team player. It is seldom worth your time and money to hire such a person on to the project. The individualist will always try to defend his workspace as long as possible and nobody can take over his job. Which mean in the end that it can delay the project considerably if the entire team has to wait until he is finished with an important element.

The Prima Donna is even worse. His is an individualist and he knows he's the best. He loathes critic and always reacts with excessive force and aggressively when his work is being criticized.

In the other end of the scale we find the shy person. They do not represent any direct threat to the project as a whole, because they never say (or do for that matter) anything. But it's a waste of time and money to hire a person that never participates in the work related discussions. As we all know, more minds think better than one, but if the knowledge that the shy person possess never is revealed the effort is wasted.

Then there is the sleeper. He might seem like the ideal team member for the project management but as soon as they turn their backs at him he starts racking everyone and everything down. There isn't anything in the world that will please the sleeper. It is very hard to identified types like him because you cannot and shall not expect the remaining team to rat on him. But it is important to remove such types from the team. They only bring bad energy with them at it can become very contagious.

Lastly we have the jack-of-all-trades. He is not necessary a menace for the team but you should always try to have as few as possible of this type, because they are doer of many but master of none. You should instead move these types to an assistant kind of role, where their talent will be of more use. Most jacks might find it hard to realize that their talent is more appreciated here, so you will need some heavy persuasions to make to them agree.

The types outlined here are of course stereotypes but it is important for the project manager to know them so he will be able to identify them quickly. All these types bear clear marks of immaturity.
Not that you should always try making a project team consist of "grey" people with 2.3 kids and a Volvo, but the types listed above are all disasters for a project and its morale.

### 4.3.1 – Teamwork

It is call "a team" and "teamwork", but there is little *team*work being done in a group of individualist covering their own work.

You should as a project manager constantly call upon the project team to do as mush teamwork as possible. Who knows, one of the art directors might come up with a solution for a problem that causing headache for the programmers, just because he is seeing thing from a different perspective.
It's simple synergy. The collective knowledge of the team and the ability to solve problems and carry out tasks is greater than that of the single team members put together.

It is chaos theory. A butterfly flaps its wings in Tokyo and a snow blizzard hits New York. A small idea that might seem useless or even ridicules at first might spark a train of thoughts in the minds of the other team members, and three months later you have come up with the killer app that will distance your final game for the competition. What if the little useless idea wasn't put forth in the beginning?

It is extremely important that there is always room for suggestions and ideas. They can all contribute in creating a unique product.

By appreciating the performance of each team member everyone will get the felling of indispensability. That the project can't live without them. Just don't over do it. You might provoke the Prima Donna effect in some.

## 4.4 – The Common Goal

As mentioned earlier it important to give the project team a common, tangible and believable goal with the entire project.
It is about giving the team a Hitchcock MacGuffin[23] as the driving force in the production. Something everybody can see as the vision and the goal. Something to put their dreams and visions into.

Of course it's difficult. A method that can be used to ensure that everyone can see their dreams in the project is to do a mind map together. Rip a day out of the calendar and use it to brainstorm over what kind of a project that it is the team is starting on.

### 4.4.1 – Cooperation Agreement

Such a brainstorm should, among other things, end up with a cooperation agreement. A written contract of all the expectation the management (here the project manager and the

[23] The famous film director Alfred Hitchcock invented what he called a MacGuffin. He used it in most of his movies as the driving force for the story. It wasn't necessary anything but it was the turning point of the entire film. The mistaken spy in "North by Northwest", the hatbox in "Rear Window". Other directors like Quentin Tarantino have also use the MacGuffin. In "Pulp Fiction" Vincent Vega and Jules Winnfield are throughout the movie searching for a briefcase, but we never find out what is in it and it doesn't matter. What matters is that it drives the story forward.

producer) has to the project and the team, but also the expectations each team member have for the work and each other.

Besides that, the cooperation agreement should list what actions must be taken if the agreement if broken. That could be that you must give a round of beer at the end of the week if your repeatedly is late on Monday mornings.

Every team member and the management sign the cooperation agreement. The purpose of the agreement is to nurture common responsibility for the project. It can help rise the morale of the project group.

## 4.5 – Chapter Conclusion

Most of the tools described in this chapter might not at first seem as tools to make the project work flow more effectively. And they do not in the short term. By in the *long* run they cause huge effectiveness. Think of all the time and resources that will be saved form ensuring that your colleagues are feeling good and therefore decides to stay on the project and see it completed.

It this kind of indirect influence that these tools give, that is a determinate factor for the entire project's well-being. If the team is thriving and has a solid work environment they will, all things considered, produce a good piece of craftsmanship.

Where the keyword for project management was "dynamic", the keyword here is: maturity. To have a mature look at your co-workers, both as a project manager and as a team member. You should dare, be capable of and willing to take and give responsibility.

Not trusting anyone else than yourself to carry out a certain task is a direct sign of immaturity. And it is destructive for the work morale of the team if they are constantly being watched and controlled. If you have hired *adult* people for your team you wouldn't have to.

You should as Christensen & Kreiner describes is; as a project manager be a gardener that seed the seeds and waters them, take away the weed and just watch the plants grow and blossom. (1991, p. 103)

# 5. – Organization

*"Organization is key. I can't stress this enough"*
**– Jay Wilson, Monolith Productions**[24]

Last but not least we shall look at the O in the PTO Pyramid: Organization. The bottom of the pyramid. The glue that keeps the project tied with the team.

## 5.1 – Organize: The Key to Freedom

There is properly someone that will insist that this headline is a contradiction. Because you have to part with some of your personal freedom to fully participate in an organization. But by optimizing the organization of the company all those working inside the organization will experience bigger *work* freedom.

## 5.2 – The Theories

The theories concerning organizations and their structure has nurtured much research and inquires in the last 100 years or so. The research and implementation of new organizational forms have put its mark on how the work methods and work forms have evolved and vice versa.

There isn't a company today that hasn't at some point addressed the issues of their organizational form.
Many of the theories have been formed with very specific industries in mind and others have tried to come up with general suggestions to how a company could and should be run. Of all the theories two of the oldest ones are: The Function Organization and The Product Organization.

### 5.2.1 – The Function Organization

Here the organization is divided into the functions that must be carried out before at product reaches the market. Divisions such as Purchase, Production, Sales, Marketing and Administration are created.
This organizational form has the very apparent advantage that it is very clear because of the well-arranged layout. There is no confusion about who is doing what.

The theorist Mintzberg calls it for a very "simple structure". The expertise is centered in the different divisions. The administration is done in the Administration division, production in the Production division and so forth. Clear and simple (Bakke & Fivelsdal 1998, p. 44-46).

Unfortunately this organizational form sometimes creates sparks between the different division in form of pride of trade and denial of responsibility.
Examples: *"Those desk clerks from Admin are always so arrogant…"* and *"… at least it's not our fault. The Purchase Division had brought some really bad raw materials"*.

[24] Quote taken from: Saltzman, Marc (ed.): *"Game Design: Secrets of the Sages (2nd edition)"*, p. 32.

### 5.2.2 – The Product Organization

Contrary to the function organization the organization here is divided by product. Every product is handled in smaller function organization that operates inside the main organization. Every product division manages a single product or market. Staff directly under the management takes care of the overall planning and administration. (Bakke & Fivelsdal 1998, p. 47)

The disadvantage is the huge amount of double work this organization gives. The large-scale operations are not used to the same extent as in the function organization.

## 5.3 – Dynamics and Improvement

In the pure form these organizational forms have little use in connection with game development. Even though that are many different processes, such as animation, programming, marketing and planning the projects are of such a feasible size that there would not be any advantage in dividing them into divisions. That would result in divisions with no more than 1-2 people working in them. Each division would on top of their daily work also do their own administration. No doubt this would quickly end in nothing but paper work.

The product organization doesn't have some relevance either, since it is (almost) always the same type[25] of product you produce.

These organizations are not wrong or bad. Defiantly not. But they belong to the group of companies that runs some kind of industrial production. Even though you properly couldn't get the CEO of Ford or Coca-Cola to claim anything else than that all their products are unique, then they are not unique in the sense used here.
They are all copies or recreations of the same product and they are all produced under predictable and for the most parts uncomplicated circumstances.

It's a long time since owning a factory was a sure way to financial success. The big money is now being made in the knowledge-based industries, where big physical production machines have been replaced with smart heads. But that also makes it much easier for the competition to copy your company's knowledge and products.

From project to project there happen great changes to the organization and its form. The organization must be dynamic and organic. Two fancy words used to describe the company's ability to change and grow with the tasks. Something that of course is important for all industries but it's the very key to success in game development. The development in the industry is going so fast that the companies have to keep on changing the organization with every project in order to survive.

For this purpose the decentralized matrix organization, or the project organization as it's also called, is excellent. The matrix organization demand constant change.
The matrix organization is a bit of a mixture of many different organization theories. You can see elements from the previous mentioned Function and Product organization.
Work on many projects can be carried at the same time and the company as a whole can through that benefit from making every team members work more effective. The same team member can with out problems work on several projects at a time.

[25] With "type" I mean type in its purest form. In this case: computer games.

# 5.4 – The Matrix Organization

*"It is not the strongest of the species that survive, nor the most intelligent, but the most responsive to change".*
**- Charles Darwin**

The quote above might be a bit far fetching, but it illustrates where the strength of a project must be: in the ability to adapt and change.

Just as there in the Darwinian theories are no master species, there are no master organizations. The perfect organization does not exist, but organizations with the ability to dynamically change the work form and the decisions processes will have a much higher chance of survival than stiff hierarchical organizations. You could call it organizational Darwinism.

Everywhere in the business life and at academic institutions there is being researched and experimented with different organizational forms. And the general saying is that the organizations must been flatten; there must be a shorter way from top to bottom. The information must flow freely in the entire organization; everyone must have access to the same information. The decisions must be made transparent; it must clear to everyone how and why decisions are made.

This of course also applies to the gaming industry. It is crucial for the success of every project that the open communications channels and the flat structure are present. The decisions must be made quickly and it lowers the effectiveness if the organization contains too many joints.

## 5.4.1 – The Hybrid Organization

Theories are as we know theories that sometimes tend to be a bit black and white. So a hybrid of theories with the best from both worlds might be more relevant. A hybrid where the projects have space to grow and change.
You could divide the organization into a basis organization and a project organization.

The basis organization would consist of all the administration, including general management, marketing, bookkeeping etc. It is in the basis organization that the daily management of the company takes place. A textbook function organization.

The project organization is the "attached" to the basis organization and all work with the projects take place in there. The projects are divided so that the workforce can flow evenly within the different departments. A programmer can therefore both work in the Tool Group and on Project A at the same time. The only confining factor would be time, which of course is limited.

This is also a method to make sure that the team members don't become fed up with a project. There will always be someone who doesn't mind changing projects once in a while. It can also give the team members the needed experience for future projects.

At the same time you should be aware of the "starters" and the "closers". Some are great in a start-up phase of a project and some are great when the end is nearing.

It is an advantage when managing your staff to move people around but it is also an organizational advantage.

This kind of organization is how a medium to large sized game production company could be structured. Smaller companies with only one project at a time could wait with this dividing of the company and instead gather all the functions in the one project, and just attach an administrational department. The smaller company can then have this organizational form in mind when growing so the entire company is geared and ready for a smooth changeover to this organizational form.

## 5.5 – Changes

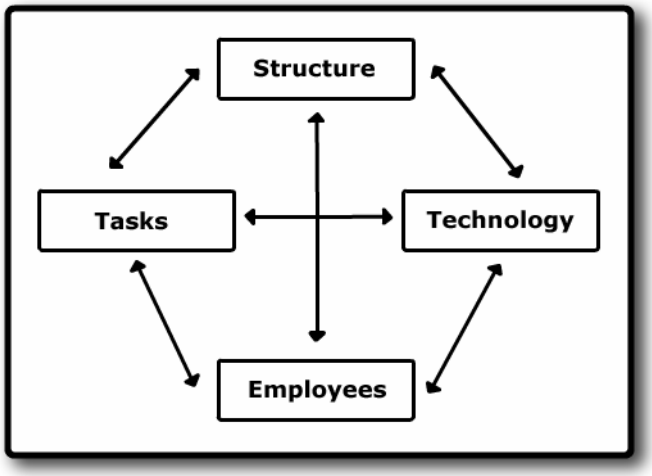



*Figure 6. Leavitt's Open System Model. (Adapted from Bakka & Fivelsdal 1998, p. 274)*

H. J. Leavitt's Open System Model from 1965 (Bakka & Fivelsdal 1998, p. 274) is an excellent tool to map out what effects internal and external changes will have on a given organization. The model consists of four elements that are closely intertwined.

Tasks are the different work carried out in the company. That will be the daily work with everything to bookkeeping to project work. The element Tasks changes when the company takes upon more and/or different projects.

The term Structure shows who does what at in what department are they located. You can also talk about structure when describing the communication pattern in the company. That could be the work routines.

Technology is every kind of hardware and software located in the company. Including offices, furniture, computers, network, physical surroundings, etc. The Technology can be influenced by replacing, removing or adding new technology.

Without Employees there couldn't be any organization, meant as if you have no workers then there wouldn't be anything to organize. The current work form in any given company will change when more employees is added.

### 5.5.1 – Influencing the Elements

You can deliberately influence one ore more of the elements in Leavitt's Open System Model. But it is impossible to totally control the process, because of the elements intertwines.

Leavitt's Open System Model should be seen as an organic coherent model. It is not possible to control it or to control the outcome when changing an element. If something is changed a chain reaction is started influencing every element in the model like rings in the water.

Even though you to some extent can try to form your organization as you wish, you will in many cases influence unintentionally. If you hire more programmers (employees) if brings more knowledge to the company and increases the need for new computers (more technology) and everyone in the organization now have more people to communicate with (change in the structure), but it also gives you the ability to do more tasks. The same example can be used if the company gets more tasks and therefore have use for more technology and/or employees.

## 5.6 – Evaluation the Form

It is important with evaluation process in organization and the decisions processes. After each ended project everyone that participated evaluates the lapse of the project and states the good and bad aspects. This way you can optimize future organization and work forms and not continue making the same mistakes.

By adding new and hopefully improved elements and work forms to the organization you can start a spiral that can give the organization automatic propulsion. (Kelly 1999, p. 25).


*Figure 7. When improving the elements during the evaluation process it can give the organization a higher value since the work forms and methods are now better. In return it gives new improvements that also heighten the value of the organization. If the spiral is mended proper it will never end. (Adapted from Kelly 1999, p. 25)*

The same principle is used in connection with the organizational theory The Learning Organization. Here it is called *metacognition* (Hauen 1998, p. 200); to reflect over why we think as we think. The figure below shows how you experience new things in the teaching

circle that gives room for improvements of the current form. Both the organizational form and the work form.

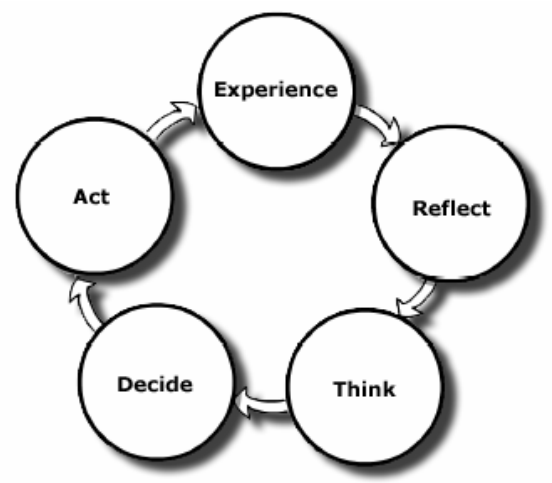



*Figure 8. The Teaching Circle. By constantly reflection over the experiences your future actions are greatly improved. (Source: Hauen 1998, s. 178)*

## 5.7 – Chapter Conclusion

Now we have the keywords dynamic from the project management chapter and maturity from the last chapter. The keyword for this chapter must be change.
The ability to change and improve the organization is one of the key elements that will make the work flow more effective. Not only for the project team but also for the whole company.

Furthermore it is important for the survival of the company that it is capable to follow the development of its surroundings. Companies in the game development industry that are producing games on yesterday's knowledge have small changes of survival.
Leavitt's Open System Model shows perfectly how a change in one element creates rings in the water that influence the entire organization.

The hybrid organization as described in this chapter makes room for the dynamic form the project management must have to constantly improve upon the team's effectiveness.

The organization is as mention in the introduction to this chapter, the glue that binds the project with the team. It because of this that it is important with the ability to change. People change and so do projects.

# 6. – Conclusion and Perspective

## 6.1 – Conclusion

The keywords are dynamic, maturity and change. And it is exactly these keywords that will help computer games development become more effective without losing any creative drive.

The industry is plagued by immaturity. Not just on a personal level, but also in the type of project management that is being used. And often it takes place in organizations that isn't geared for development and change. So the projects are not given the space required.

This is of course a generalization, but not many see what computer game projects really are: Projects in movement.
They change constantly. If the team members aren't professional and sincerely engaged in the work it can become difficult for everyone, from testers to project managers, to get an overview of the project. You have to be professional mature as personal mature. You should always try to get people in your team that dares and understand to take responsibility for their own work and at the same time lift the common responsibility for the entire project team.

It goes without saying that you have to be apple to cooperate with many different types of people. Not just different professional types, but also different personal types.
It is the paradox of the industry that the sum of all the different professional knowledge and cultural background is a huge factor in creating unique experiences for gamers all over the world is also the source of so many conflicts.
The most prominent ability a project manager can posses is judge of character. You can always learn theoretical control tools and how to make good-looking project plans.

The organization must be a solid foundation for the organic form that projects often assume. The must be room for cooperation and the professional barriers must be downplayed as much a possible or remove completely. Job titles are great for business cards, but during project work, a more low-key attitude towards the work is required. Everyone must be open to criticism from unexpected sides. It is about creating synergy where everyone contributes to the foundation of the project.
The projects team's primary goal must always be to create the best possible project. Personal goals, such as promotion and higher salary, must always take second place. The personal goals are also mush easier achieved if you have actively participated in creating a quality product.

The wellbeing of the team members will in the longer run ensure the effectiveness of the entire project group. By ensuring that the team members stay on the project and fell their work appreciated, the members will also work to their best.
The respect for and the knowledge of each others trade will also help the project to run more conflict free. All the time and all the resources you can save by doing so can not be measured in money.

It would be wrong to imply that the suggestions in this report will lead to perfect project conditions. The perfect organization does not exist. The perfect team member does not exist. The error free project does not exist. But the *mature* organization and team member

does, and if combined with dynamic project management they can create a quality product that every one can be proud of.

It *can* be done better.

## 6.2 – Perspective

… and it *must* be done better.

The competition in the gaming industry today is so though that it is only a fraction of the productions that faces a successful future.
Around 2000+ games are released worldwide every year and many of them are in direct competition.

The gaming industry has experience and is right now experiencing a transformation not far from the one the movie industry went through some decades ago.
The big computer games corporations rarely produce their games themselves. This work is made in the smaller development houses around the world. It more or less goes without saying that when the corporations needs to pick a new development house for the next installment in their multimillion dollar gaming license, that they pick the team most capable of competing the project successfully.
The demands for the production and how it is manage and controled is therefore very high. It's a matter of getting a quality product on the marked fast.

Creating computer games is an enormously complex process, and unless you continually generate new and improved work forms, you will lose out.

At the same time as the normal IT business has experienced some of its most turbulent times ever, the computer games industry has gone through this more or less unharmed. The great demand for computer games and other kinds of interactive entertainment is unchanged.
Computer games are here to stay, and the industry will as a whole will experience a rising development in the decade to come.

Without doubt.

# Reference

## Danish books

Melander, Preben (red.): *"Projektstyringens problemer og værktøjer"*. Denmark, Jurist- og Økonomforbundets Forlag, 1993. ISBN: 87-574-5805-7

Christensen, Søren; Kreiner, Kristian: *"Projektledelse i løst koblede systemer"*. Denmark, Jurist- og Økonomforbundets Forlag, 1991. ISBN: 87-574-5930-4

Bakka, Jørgen Frode; Fivelsdal, Egil: *"Organisationsteori. Struktur, kultur, processor"*. Denmark, Handelshøjskolens Forlag, 1998. ISBN: 87-16-13394-3

Hauen, Finn van; Strandgaard, Vagn; Kastberg, Bjarne: *"Den Lærende Organisation"*. Denmark, Peter Asschenfeldts Nye Forlag, 1998. ISBN: 87-600-0054-6

## English books

Bates, Bob: *"Game Design: The Art & Business of Creating Games"*. USA, Prima Publishing, 2001. ISBN: 0-7615-3165-3

Saltzman, Marc (red.): *"Game Design: Secrets of the Sages (2nd edition)"*. USA, Brady Publishing, 2000. ISBN: 1-56686-987-0

Rollings, Andrew; Morris, Dave: *"Game Architecture and Design"*. USA, Coriolis Group, 2000. ISBN: 1-57610-425-7

Kelly, Kevin: *"New Rules for the New Economy. 10 Radical Strategies for a Connected World"*. USA, Peguin Books, 1999. ISBN: 0-670-88111-2

Tzu, Sun: *"The Illustrated Art of War"*. Translated by Thomas Cleary. England, Shambhala, 1998. ISBN: 1-57062-422-4

## English articles

Wardell, Brad: *"PC Gaming As An Industry"*. The Adrenaline Vault, 7th February, 2001. (http://www.avault.com/developer/getarticle.asp?name=bwardell1)

Wardell, Brad*: "PC Gaming As An Industry, Part II: The Industry At This Moment"*. The Adrenaline Vault, 6th March 2001.
(http://www.avault.com/developer/getarticle.asp?name=bwardell2)

Wardell, Brad: *"PC Gaming As An Industry, Part III: Ideas on what's coming next…"*. The Adrenaline Vault, 9th April 2001.
(http://www.avault.com/developer/getarticle.asp?name=bwardell3)

Wardell, Brad: *"PC Gaming As An Industry, Part IV: Destroying Myths"*. The Adrenaline Vault, 7th. may 2001.

(http://www.avault.com/developer/getarticle.asp?name=bwardell4)

Clayborn, Johnathan*: "Why PC Gaming Will Not Die"*. The Adrenaline Vault, 27th March 2001.
(http://www.avault.com/developer/getarticle.asp?name=jclayborn)

Ahearn, Luke: *"Budgeting and Scheduling Your Game"*. Gamastura.com, 4th May 2001.
(http://www.gamasutra.com/features/20010504/ahearn_01.htm)

Tapscott, Don: " The power of electronic play". Computerworld.com, 24th May 1999.
(http://www.computerworld.com/cwi/story/0,1199,NAV47_STO35777,00.html)

Interactive Digital Software Association (IDSA): *"2000 - 2001 State of the Industry Report"*
(http://www.idsa.com/releases/SOTI2001.pdf)

## Postmortem articles from Gamasutra.com

*Postmortem: Irrational Games' Freedom Force*
Printed in Game Developer – May 2002 issue

*Postmortem: 2015's Medal of Honor: Allied Assault*
Printed in Game Developer – June 2002 issue

*Postmortem: Surreal Software's Drakan: The Ancients' Gate (Drakan 2)*
Printed in Game Developer – July 2002 issue

*Postmortem: Treyarch's Spider-Man*
Printed in Game Developer – August 2002 issue

*Postmortem: Gas Powered Games' Dungeon Siege*
Printed in Game Developer – September 2002 issue

*Postmortem: Raven Software's Star Trek: Voyager - Elite Force*
(http://www.gamasutra.com/features/20010207/pellertier_01.htm)

*Postmortem: Looking Glass's Thief: The Dark Project*
(http://www.gamasutra.com/features/19990709/thief_01.htm)

*Postmortem: Raven Software's Soldier of Fortune*
(http://www.gamasutra.com/features/20000927/biessman_01.htm)

*Postmortem: Ensemble Studios' Age of Empires II: The Age of Kings*
(http://www.gamasutra.com/features/20000307/pritchard_01.htm)

*Postmortem: Epic Games' Unreal Tournament*
(http://www.gamasutra.com/features/20000609/reinhart_01.htm)

*Postmortem: Zombie's SpecOps: Rangers Lead the Way*
(http://www.gamasutra.com/features/20000201/ridgeway_01.htm)

*Postmortem: Surreal Software's Drakan: Order of the Flame*
(http://www.gamasutra.com/features/20000418/denman_01.htm)

*Postmortem: Ion Storm's Deus Ex*
(http://www.gamasutra.com/features/20001206/spector_01.htm)

*Postmortem: Turbine Entertainment's Asheron's Call*
(http://www.gamasutra.com/features/20000525/ragaini_01.htm)

*Postmortem: Westwood Studios' Command & Conquer: Tiberian Sun*
(http://www.gamasutra.com/features/20000404/tiberiansun_01.htm)

*Postmortem: Sierra Studios' Gabriel Knight 3*
(http://www.gamasutra.com/features/20001011/bilas_01.htm)

*Postmortem: Sierra's SWAT3: Close Quarters Battle*
(http://www.gamasutra.com/features/20000306/napier_01.htm)

*Postmortem: Ritual Entertainment's Heavy Metal: F.A.K.K. 2*
(http://www.gamasutra.com/features/20001220/alden_01.htm)

*Postmortem: Irrational Games' System Shock 2*
(http://www.gamasutra.com/features/19991207/chey_01.htm)

*Postmortem: Activision's Heavy Gear 2*
(http://www.gamasutra.com/features/19991208/imislund_01.htm)

*Postmortem: The X-Files*
(http://www.gamasutra.com/features/19991203/xfiles_postmortem_01.htm)

*Postmortem: Star Trek: Hidden Evil*
(http://www.gamasutra.com/features/19991119/startrekpostmortem_01.htm)

*Postmortem: Micro Forte' s Fallout Tactics*
(http://www.gamasutra.com/features/20010420/oakden_01.htm)

*Postmortem: Blizzard Entertainment's Diablo II*
(http://www.gamasutra.com/features/20001025/schaefer_01.htm)

*Postmortem: Red Storm's Rainbow Six*
(http://www.gamasutra.com/features/20000121/upton_01.htm)

*Postmortem of Nihilistic Software's Vampire: The Masquerade – Redemption*
(http://www.gamasutra.com/features/20000802/huebner_01.htm)

*Postmortem: Minds Eye's Sheep*
(http://www.gamasutra.com/features/20010413/evans_01.htm)

*Postmortem: Naughty Dog's Jak and Daxter: the Precursor Legacy*
(http://www.gamasutra.com/features/20020710/white_01.htm)

*Postmortem: Tony Hawk's Pro Skater (Dreamcast)*
(http://www.gamasutra.com/features/20000628a/fristrom_01.htm)

*Postmortem: Shiny Entertainment's Wild 9*
(http://www.gamasutra.com/features/20000107/wild9_01.htm)

*Postmortem: Monolith's No One Lives Forever*
(http://www.gamasutra.com/features/20010608/hubbard_01.htm)

*Postmortem: Treyarch's Draconus*
(http://www.gamasutra.com/features/20000814/fristrom_01.htm)

*Postmortem: Angel Studios' Resident Evil 2 for the N64*
(http://www.gamasutra.com/features/20000728/meynink_01.htm)

*Postmortem: Startopia*
(http://www.gamasutra.com/features/20011027/imlach_01.htm)

*Postmortem: Poptop Software's Tropico*
(http://www.gamasutra.com/features/20011010/smith_01.htm)

*Postmortem: Lucas Arts' Star Wars Starfighter*
(http://www.gamasutra.com/features/20010801/corry_01.htm)

Postmortem: Lionhead Studios' Black & White
*(http://www.gamasutra.com/features/20010613/molyneux_01.htm)*

Postmortem: Multitude's Fireteam
*(http://www.gamasutra.com/features/20000105/fireteam_01.htm)*

*Postmortem: Factor 5's Star Wars Rogue Leader: Rogue Squadron II*
(http://www.gamasutra.com/features/20020501/engel_01.htm)

*Postmortem: Pseudo Interactive's Cel Damage*
(http://www.gamasutra.com/features/20020227/wu_01.htm)

*Postmortem: Mythic's Dark Age of Camelot*
(http://www.gamasutra.com/features/20020213/firor_01.htm)

*Postmortem: Frog City's Trade Empires*
(http://www.gamasutra.com/features/20020125/bernstein_01.htm)

*Postmortem: Bohemia Interactive Studios' Operation Flashpoint*
(http://www.gamasutra.com/features/20011219/spanel_01.htm)

# Appendix A: The Blackwood Benchmark™

Or the analysis formerly known as The Gamasutra.com Postmortem Analysis

This is the result of an analysis I made from studying 43 different articles in the ongoing postmortem series published at Gamasutra.com.

The articles I've looked at have all been published in the period from den September 1999 to August 2002.

The idea for this analysis came after I posted a message in the newsgroup comp.games.development.industry with the headline **"What is wrong with game development today?".** This message and a follow-up message got around 300 replies put together.

I've split up all my findings into positive and negative subjects.

The titles in the brackets are referring to the article where the subject is mentioned. For example "(Deus Ex)" means that I found the following statement/subject written in the article: "Postmortem: Ion Storm's Deus Ex" by Warren Spector.
This is more or less just a sum up of all the "What went right" and "What went wrong" (... if you're familiar with the postmortems articles).

Please note that this is my own subjective view of what I've read in the articles. You may or may not agree with all of the interpretations of the text that I've made.

It's meant as a guideline or as a checklist for future productions.

# Planning and game design issues

| Positive | Negative |
|---|---|
| Good planning of the project.<br><br>(Soldier of Fortune, Jak & Daxter, No One Lives Forever, Resident Evil 2, Star Wars: Rogue Squadron II, Cel Damage, Trade Empires, Spider-Man). | Bad or no planning. Weighted the different sub-projects wrong.<br><br>(Thief, The X-Files, System Shock 2, Heavy Gear 2, SWAT 3, Drakan, Asheron's Call, Vampire, Soldier of Fortune, Deus Ex, Heavy Metal: FAKK 2, Sheep, Dracounus, Drakan 2). |
| | Didn't have a proper risk estimate.<br><br>(Deus Ex). |
| Clear description of the goal and mission statement.<br><br>(No One Lives Forever, Resident Evil 2). | |
| Good balance between the different sub-projects. Realistic expectations.<br><br>(C&C: Tiberian Sun, Soldier of Fortune, Diablo 2, No One Lives Forever). | Unrealistic plan. Bad plan.<br><br>(The X-Files, Heavy Gear 2, C&C: Tiberian Sun, Drakan, Gabriel Knight 3, Fallout Tactics, Black & White, Medal of Honor: AA, Dungeon Siege). |
| Frequent milestones. Effective milestones.<br><br>(SWAT 3, Diablo 2, Deus Ex, Fallout Tactics, Resident Evil 2, Medal of Honor: AA, Spider-Man). | Didn't have a final deadline. Didn't have milestones.<br><br>(Soldier of Fortune, Operation Flashpoint, Drakan 2). |
| Realised that a game project is an organic process where it often is necessary to go back and reconsider decisions.<br><br>(Deus Ex, Trade Empires, Operation Flashpoint). | Locked the game design to early in the process which made it very difficult to change it later on.<br><br>(C&C: Tiberian Sun, Asheron's Call). |
| Good pre-production of the project. Long analysis- and design phase. Intensive work on the storyline.<br><br>(Deus Ex, Star Trek: Elite Force). | Badly executed storyline.<br><br>(Drakan, Black & White). |
| | Didn't do much pre-planning.<br><br>(No One Lives Forever). |
| Focus on the project. Concentration and a more précis specification of the end goals and visions with the project. And a will to keep them.<br><br>(Thief, Rainbow Six, SWAT 3, Asheron's Call, Vampire, Deus Ex, Heavy Metal: FAKK 2, Star Trek: Elite Force, Startopia, Tropico). | Too ambitious project. Unrealistic expectations.<br><br>(The X-Files, Rainbow Six, SWAT 3, C&C: Tiberian Sun, Asheron's Call, SpecOps, Vampire, Deus Ex, Fallout Tactics, Freedom Force, Dungeon Siege). |
| | Didn't have a complete overall design of the entire game. Didn't have focus on the design of the entire game.<br><br>(Rainbow Six, Unreal Tournament, Soldier of Fortune, Fallout Tactics, Sheep, Jak & Daxter, Tropico, Star Wars: Starfigher, Cel Damage, Drakan 2, Spider-Man). |
| Good central description of the process of the production.<br><br>(Age of Empires 2, Medal of Honor: AA). | Loss of staff. Can create a certain amount of chaos if there are no one else who are capable of continuing the work of the departured staff member.<br><br>(Thief, Heavy Gear 2, Rainbow Six, SpecOps, Gabriel Knight 3, Heavy Metal: FAKK 2). |
| | Changes in the design where difficult to trace which made it very hard to go back and recreate elements of the game. Lacked documentation of the projects process. |

| | |
|---|---|
| | (SWAT 3, Asheron's Call, Fallout Tactics, Wild 9, Cel Damage). |
| Thorough preparation of the game as a whole.<br><br>(Thief, Heavy Gear 2, SWAT 3, C&C: Tiberian Sun, Drakan, Soldier of Fortune, Gabriel Knight 3, Deus Ex, Star Trek: Elite Force). | Bad weighting of the gameplay. Some things where way to hard and some too easy. Some gameplay elements lacked.<br><br>(The X-Files, Heavy Metal: FAKK 2, Trade Empires). |
| Professional attitude toward the project. God project managers.<br><br>(Rainbow Six, SWAT 3, C&C: Tiberian Sun, Gabriel Knight 3, Fallout Tactics, Dracounus, Resident Evil 2, Star Wars: Starfigher, Dark Age of Camelot, Freedom Force, Drakan 2, Spider-Man, Dungeon Siege). | Bad or lacking project managers.<br><br>(The X-Files, Heavy Metal: FAKK 2, Sheep, Medal of Honor: AA). |
| | Did not have the formal project process' written down (on some key issues).<br><br>(Age of Empires 2, Deus Ex, Tropico). |
| | Almost loosing focus and concentrating on other thing (other projects, personal projects, etc.)<br><br>(Wild 9). |

# The project group and organization

| Positive | Negative |
|---|---|
| Use of professional freelancers, when or if the project group did not contain the expertise needed.<br><br>(Vampire, Soldier of Fortune, Wild 9, Cel Damage). | Lacked experienced staff members in certain areas of the project.<br><br>(System Shock 2, Rainbow Six, SWAT 3, Asheron's Call, Unreal Tournament, SpecOps, Gabriel Knight 3, Star Trek: Elite Force, Sheep, No One Lives Forever, Star Wars: Starfigher, Fireteam, Cel Damage, Spider-Man). |
| The whole project team was located at the same physical place.<br><br>(Vampire, Tony Hawk, Wild 9, Resident Evil 2). | Bad communication.<br><br>(Asheron's Call, Soldier of Fortune, Dracounus). |
| Good group dynamic. Good communication within the team.<br><br>(The X-Files, System Shock 2, Rainbow Six, SWAT 3, C&C: Tiberian Sun, Gabriel Knight 3, Deus Ex, Fallout Tactics, Sheep, Tony Hawk, No One Lives Forever, Resident Evil 2, Startopia, Tropico, Star Wars: Starfigher, Black & White, Dark Age of Camelot, Operation Flashpoint, Freedom Force, Drakan 2, Spider-Man, Dungeon Siege). | The whole project team was not located at the same physical place.<br><br>(Unreal Tournament, Freedom Force). |
| Everyone came forward with ideas, suggestions and criticism.<br><br>(System Shock 2, Drakan, Unreal Tournament, Vampire, Deus Ex, Tony Hawk, Tropico, Dungeon Siege). | |
| Good prototypes = Gave the project members a sense of progress.<br><br>(Heavy Gear 2, Asheron's Call, Deus Ex, Fallout Tactics, Tony Hawk). | Moral problems. For example delay in the project, or changes in the market, which meant that there might not be a market for the final game.<br><br>(Gabriel Knight 3, Fireteam, Freedom Force) |
| Trust from the management. Project members where given responsibilities and where expected to live up to it.<br><br>(Heavy Gear 2). | |
| Other team members taking over positions needed.<br><br>(Wild 9, Drakan 2). | Need a dividing of the company. Some key personnel members where both designing the game and building up the company.<br><br>(System Shock 2, Deus Ex, Sheep, Black & White, Freedom Force). |
| | Loss of staff members meant break of moral.<br><br>(Wild 9). |

# Technology (programming, art and audio)

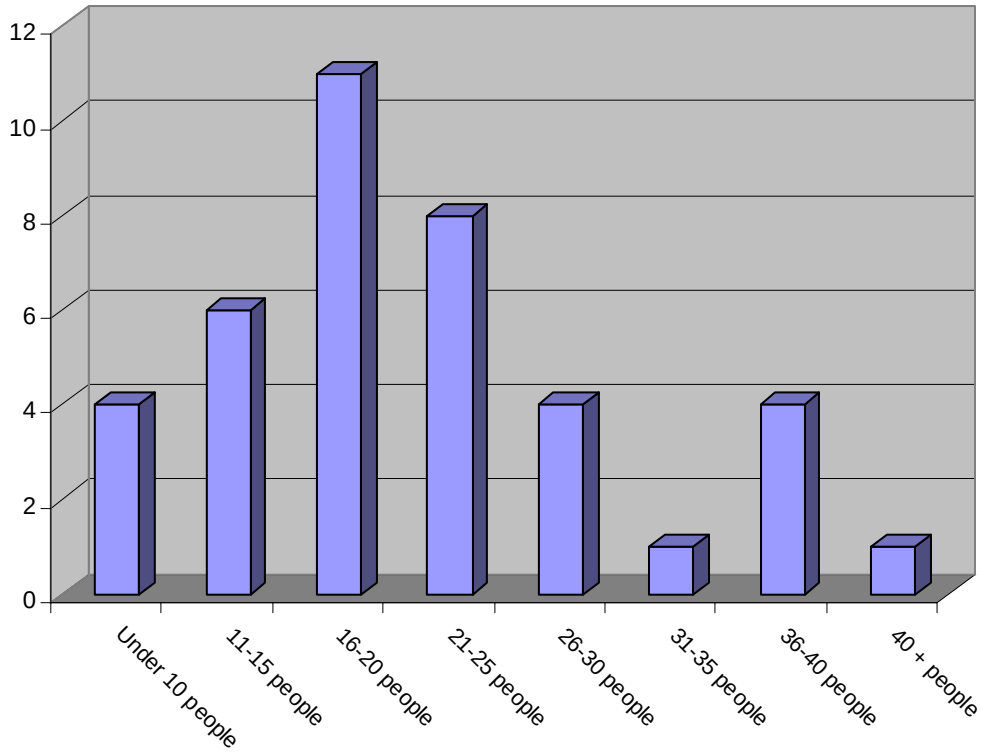
| Positive | Negative |
|---|---|
| Developed own tools for the non-programmers, so that almost everyone could add and change the game in progress.<br><br>(Thief, Star Trek: Hidden Evil, Heavy Gear 2, Rainbow Six, Drakan, Unreal Tournament, Age of Empires 2, Soldier of Fortune, Gabriel Knight 3, Deus Ex, Heavy Metal: FAKK 2, Star Trek: Elite Force, Fallout Tactics, Sheep, Jak & Daxter, Wild 9, No One Lives Forever, Tropico, Fireteam, Star Wars: Rogue Squadron II, Cel Damage, Dark Age of Camelot, Trade Empires, Operation Flashpoint, Medal of Honor: AA, Drakan 2, Spider-Man). | Almost broke their back in trying to make all the tools themselves, and not using any of the standard tools available.<br><br>(Thief, Rainbow Six, Unreal Tournament, Diablo 2, Jak & Daxter, Dark Age of Camelot, Medal of Honor: AA, Spider-Man, Dungeon Siege) |
| | Did not have manuals for the "home-grown" tools, which meant that the programmers used a lot of time answering questions that could have been in a manual.<br><br>(Thief, System Shock 2, Age of Empires 2, Jak & Daxter, Tony Hawk, Star Wars: Rogue Squadron II, Dungeon Siege). |
| Use of standard tools. Use of tools developed by other game development companies.<br><br>(The X-Files, SWAT 3, Vampire, Deus Ex, Heavy Metal: FAKK 2, Star Trek: Elite Force, Star Wars: Starfigher, Star Wars: Rogue Squadron II, Dark Age of Camelot, Trade Empires, Freedom Force, Dungeon Siege). | Used only standard tools. Didn't have "home-grown" tool at all. The standard tools locked the game.<br><br>(Startopia, Medal of Honor: AA). |
| Familiar with the technology.<br><br>(Soldier of Fortune, Heavy Metal: FAKK 2, Tropico, Star Wars: Starfigher, Cel Damage, Dark Age of Camelot, Trade Empires, Drakan 2, Dungeon Siege). | Technical problems. Missed or overlooked technical difficulties.<br><br>(Gabriel Knight 3, Diablo 2, Heavy Metal: FAKK 2, Star Trek: Elite Force, Tony Hawk, No One Lives Forever, Dracounus, Resident Evil 2, Startopia, Star Wars: Starfigher, Black & White, Fireteam, Star Wars: Rogue Squadron II, Dark Age of Camelot, Trade Empires, Operation Flashpoint, Medal of Honor: AA, Drakan 2). |
| | Lacked hardware. Specific development hardware, like a console development station or a testing server.<br><br>(Dracounus, Dark Age of Camelot). |
| Started to design the technical aspects of the game, and the got the artistic elements to fit in.<br><br>(System Shock 2, SpecOps, Gabriel Knight 3). | Worked only the with graphics, and almost forgot the technical aspects.<br><br>(Startopia). |
| | Fixation on the technology, which in the end delayed the project.<br><br>(SpecOps, Soldier of Fortune, Tony Hawk). |
| Reuse of code from different projects = shorter development time. And plan to use the current code in future projects.<br><br>(Thief, The X-Files, System Shock 2, SWAT 3, Asheron's Call, Unreal Tournament, Age of Empires 2, Gabriel Knight 3, Diablo 2, Dracounus, Resident Evil 2, Spider-Man). | Reuse of bad code. Planed to reuse a lot of code but spent almost more time trimming the old code and making new.<br><br>(Tropico, Drakan 2). |
| Enhancement of gameplay elements that have proven their "strength". No real reason for invention new elements when those you got already works fine.<br><br>(System Shock 2, C&C: Tiberian Sun, Age of Empires 2, Diablo 2, Drakan 2). | |

| | |
|---|---|
| Great documentation of the code.<br><br>(Jak & Daxter). | No or bad documentation of the code.<br><br>(Wild 9, Startopia, Operation Flashpoint). |
| Worked intensively with the art<br><br>(Dracounus, Black & White, Star Wars: Rogue Squadron II). | Concentrated on different thing than the art before it was almost too late.<br><br>(Trade Empires). |
| Worked intensively with the sound and/or voice action.<br><br>(Thief, Gabriel Knight 3, Heavy Metal: FAKK 2, Star Trek: Elite Force, Startopia, Star Wars: Rogue Squadron II). | Audio problems. Didn't work as much as need with the audio.<br><br>(Jak & Daxter, Resident Evil 2, Spider-Man). |
| Scaled down the project's size.<br><br>(Star Trek: Hidden Evil, Tony Hawk, Fireteam). | The project was too big.<br><br>(Black & White). |
| | Added too many unnecessary features to the game.<br><br>(C&C: Tiberian Sun, Diablo 2). |

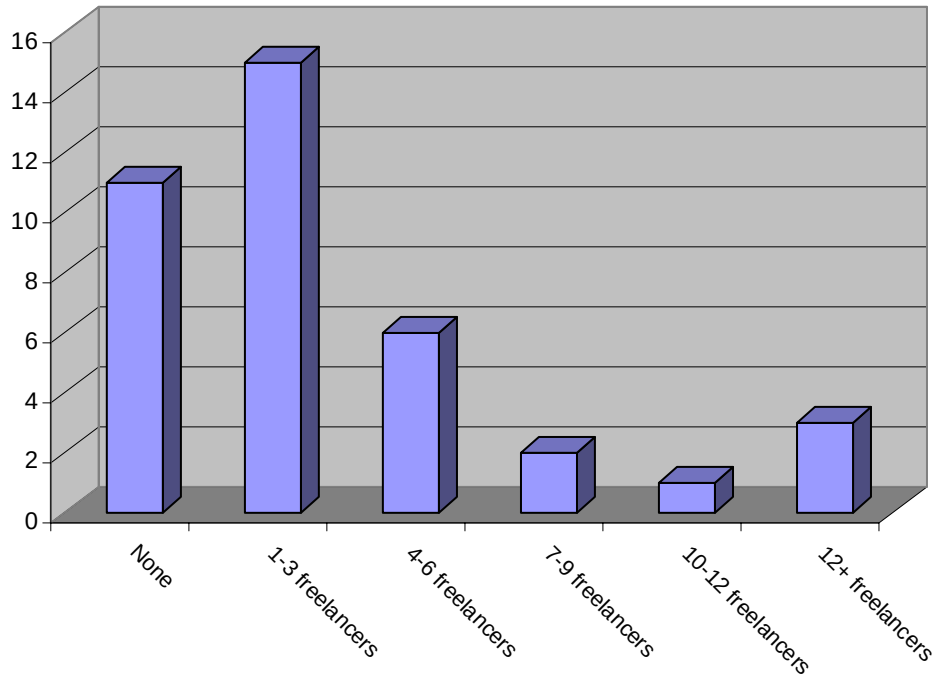
# Test and publishing (+ misc.)

| Positive | Negative |
|---|---|
| Get partners. Either companies with money to throw around or publishers who'll invest in the project to get the distributing rights for the final game. Use the knowledge these companies possess.<br><br>(Asheron's Call, Fallout Tactics, Dark Age of Camelot, Medal of Honor: AA). | Bad communication with the publisher. Didn't understand the communication processes of the publisher.<br><br>(Tony Hawk, Dracounus, Resident Evil 2, Cel Damage, Operation Flashpoint). |
| | Worked a long time without a publisher.<br><br>(No One Lives Forever, Operation Flashpoint). |
| Thorough preparation and execution of the test phase, which in the end ensured a product of higher quality.<br><br>(Heavy Gear 2, Asheron's Call, Age of Empires 2, Diablo 2, Fallout Tactics, Tony Hawk, Startopia, Star Wars: Starfigher, Cel Damage). | Didn't have time to test the final game thoroughly for bugs. Finished the project too early.<br><br>(Rainbow Six, SWAT 3, The X-Files, Drakan, Resident Evil 2, Black & White, Trade Empires, Operation Flashpoint). |
| | Didn't document the test results or didn't use the test results constructively.<br><br>(Tony Hawk, Resident Evil 2, Spider-Man). |
| Good launch of the game. Good marketing up to the launch.<br><br>(Unreal Tournament, Soldier of Fortune, Diablo 2, Tony Hawk, Dark Age of Camelot). | Not all publicity is good publicity.<br><br>(Deus Ex, Black & White, Dark Age of Camelot). |
| | Didn't have a cohesive marketing plan.<br><br>(Dark Age of Camelot, Trade Empires). |
| Direct contact with the gamers and the gaming community. Received a lot of suggestions from gamers all over the world.<br><br>(Unreal Tournament, Soldier of Fortune, Fireteam, Dark Age of Camelot, Operation Flashpoint). | Lacked press coverage.<br><br>(Startopia). |
| | Temporary solutions.<br><br>(Heavy Gear 2, Diablo 2, Cel Damage). |

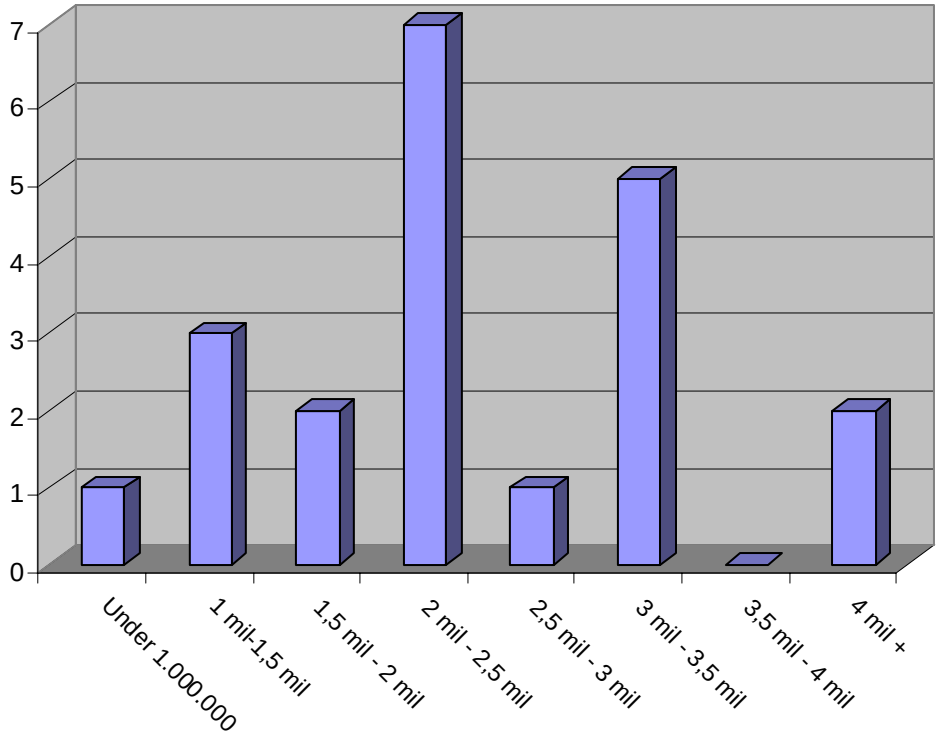
## Team size (full time developers)




## Number of contractors (freelancers, contractors)



# Production budget (in million dollars)




# Production time