

The human factor: predictable or unpredictable?

Kris Oosting

Abstract

This paper describes some issues we actually know, but almost never considered. Why? Because they are mostly human and difficult to understand and predict. Although project managers are managing technology projects, they should not forget human factors like: stress, TIA, re-use, fire fighting, etc. Humans make it work. So actually a project manager is managing humans. These humans have a big influence on project control. But are they really so unpredictable that we are not able to control our projects?

1. Current factors

When moving to a new technology, we want to understand the new issues and factors that influence our projects. But is there a difference or do we think there is a difference. Technology is often perceived as a silver bullet. We've just didn't find the wolf. Whatever silver bullet we use, some factors remain the same.

1.1. Management wants a demo

When a project follows a method correctly, it will take a while before something can be shown on screen. In practise we discovered that most managers can keep themselves together for about 1 – 2 months. If nothing is demonstrated by then, they will ask for a demo of the product.

We know that this behaviour often is devastating for projects. The only problem is that we have a feeling, but not always the metrics to support it.

This example shows the power of metrics on a real project. Project size is 12 persons, working on the product for almost a year, C++, UNIX environment, Object-oriented method used, and incremental delivery.

For no apparent reason the top manager wants a demo of the product and forces this decision with a date the demo should be given. He also tells what he wants to see in terms of functionality.

What do you do as developer? Well, let everything fall out of your hands and code as quickly as you can in order to demonstrate as much functionality as you can. In this case 80% of their time was coding. What happens during the demo? Of course ... it crashes! Then most managers will politely inform you that what they expected is true: "Told you so. Methods, CASE tools, and measurements don't work!"

Not in our case. The developers entered comment statements in the source code for the parts that were not supported by good design. In this case they used // Hack.

The time keeping showed an interesting 'human factor'. Soon after the main coding job ended (high stress), people reported ill, or took days off.

See figure 1 below. This shows time registration of 12 persons.

Okay, what about the metrics. After the demo all //Hack comments were redesigned in a proper way and coded again. This little effort took an extra 30 man-weeks! The manager was confronted for the first time with the price of his decision. At this company, from that moment, metrics are used to support decision making. This will make people less nervous, more motivated, and it will assist project control and communication.

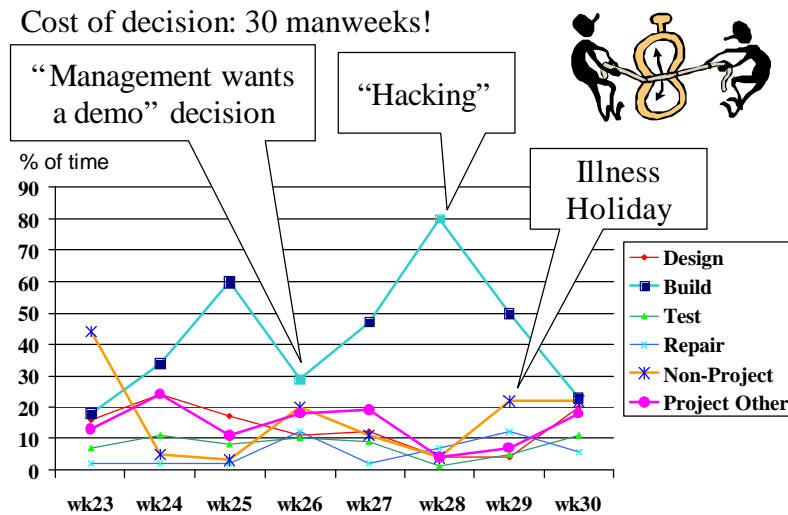


Figure 1

1.2. Adding more ...

Frederick Brooks wrote in his book ‘The Mythical Man-month’: “adding more people to a late project makes it only later”. Guess what. We still do it. By adding more people to a running project, a project manager influences a ‘human factor’ of the project – productivity. One new person on a project will keep about two experienced project members off their work for the next 2 to 3 weeks. Not because the person is not smart, but just because he/she has to learn and understand. Most projects do not always have good documentation, you know.

This in turn will increase irritation, stress, and decrease productivity. Solution you might ask? Well, most project managers will add more people ...

And in CBD? You’ve probably guessed it. Adding more components to a late project ...

Solution? This often depends on the project manager’s skills and nerves.

1.3. Resistance is futile – we plan for you

Another interesting behavioural pattern can be discovered when we observe steering committees. Many developers and project managers do not really understand how the steering committee determined the budget and the end-date of the project. Especially if the end-date is already determined before the project manager could even report what the project is all about.

Again the use of software metrics as a means of communication can help. As well as steering committees as project managers often do not have sufficient data to support their decisions. That’s why they make decisions they cannot support. This in turn makes it that projects are very difficult to control. “Why control my project if the end-date is determined by others?”

Unfortunately, the project has to be finished in time anyway, so project managers often reach for their toolkit – fire fight equipment. But is this bad?

1.4. Fire fighting pays very well!

Well, we all know them. People that run around with lots of papers under their arm telling everybody how hard they work solving all those problems. The positive thing for the employee is: fire fighting creates visibility, visibility creates the idea that something is really happening, so one gets more salary for the hard work.

With all those quality improvement programs around, we have to ask ourselves: “which problems are solved? Problems they’ve inserted themselves by using a bad and not measured process?”

Unfortunately, many companies still pay their employees mainly for doing things, not for thinking about it.

We all know: “we always have time to do the project over again, but never time to do it the first time right.”

Fire fighting is a human reaction to a ‘bad’ process and ‘bad’ planning. This reaction causes stress, demotivation, defects that have to be repaired, so less time ... more fire ...

1.5. A game called estimation

A challenge for most project managers. First we had the waterfall model. But this was not sufficient, so we used the radical model of planning. Again, not fast enough, so we moved to evolutionary planning ... time boxing ... puzzle. Making a plan is difficult. New technology, new functionality, and ... human factors! The last one is not so obvious, that we often don’t even consider it. And why should we. This time we’re using the best C++ programmer in our company. Unfortunately, most project managers learned it the hard way. There are many factors that can influence even the best programmers, and there are many ways to understand these intangible factors. Estimation still is a game, but you’ll have more fun playing it till the end.

1.6. Only time for the real work!

“Process? Metrics? Methods? UML? – come on, I only have time for the real work!”. Often heard by people involved in quality programs. The point is, what is the real work. For many developers this still is, sorry, hacking a product together. And we can’t blame them. Most companies still pay their employees for doing things, not for thinking about it. The end product, the software running on a computer, has such an important place in our minds, that we do everything possible in order to show something on screen.

It also has to do with how we perceive the phases in our development life cycle. If we perceive building a system as solving a problem, then we could say:

- ?? Problem Definition, or the requirements phase, defines the problem (or the challenge) as well as possible. This can be text, use case models, etc.
- ?? Problem Specification, or the analysis phase, specifies the problem in a way that software developers can do something with it. Models can be class models, sequence diagrams, etc.
- ?? Problem Solution, or the design phase, delivers a solution to the specified problem. A lot of creativity is required here, while the amount of human interaction is lower than during analysis. When one is creative, one can make mistakes, called defects—although making mistakes doesn’t completely rely on creativity. It also relies on how the problem was specified, of course. Another factor is how often one was interrupted during the work.
- ?? Problem Implementation, or the coding phase, delivers the implementation of the found solution. As you have probably guessed by now, it is the least interesting phase of all. Unfortunately we have to translate the designs to something the poor computer understands, namely code. In this phase we “measured” low levels of human interaction and creativity. Everybody was caught within the “walls” of the programming language.

In order to assist people with this process of change we give a different type of training. Training time should be as much as possible translated to working time. This means that the exercise in the course should be a real project from the company. So after the training

developers just continue, instead of starting all over again. Problem is however, that many trainers know their ATM example very well, but are afraid to face a real project with real, unpredicted, problems. Don't be afraid, dear trainers, the issue is not to demonstrate that one knows the exercise, but how to apply the theory.

Only then this human factor pattern of, "I-only-have-time-for-real-work" can be changed. But be careful ... You never get a second chance to make a first impression. This is especially true for measurement programs. Introduce these step-by-step with just a few metrics to start with. And ... do it goal based. If there is not a goal, one should not measure.

1.7. Reuse – what's in it for me?

This turns out to be a very interesting Human Factor. When objects came to the world, reuse was big. Now components come to the world and yes, you've guessed it. Reuse!

What we see about 7 out of 10 times is, that discussions about reuse are not about "how do we manage all those components", but "how do we reward people, so they really reuse the stuff".

So we're human after all – what's in it for me? The question is, does this really work on the long run? Reuse is possible, but only with good component management tools, someone responsible for it, and the right development process. This also means that in phases like analysis, architecture, and design there has to be a mechanism in place to support reuse.

But there is another issue that plays a role. Projects that develop software that has to be reused take longer than projects that build software without reuse in mind.

Estimation tools like *Optimize* for estimating OO/CBD projects make sure you think about these issues in an early stage of estimation. This assists the project manager to focus and decision making.

2. New factors

This section describes some new factors that should be part of a project manager's skill set.

2.1. Metrics enable communication and early warning

Metrics can be used for understanding what is going on, for controlling projects, etc. But a powerful applications of metrics, if the right ones are used, is that metrics enable communication between different levels in the organisation.

Further metrics can function as part of an early warning system for projects. This means that metrics are collected during the project and reported for feedback. When something tends to go wrong, the project manager can act before it's too late. In other words, prevent fire fighting and save a lot of spilled hours. This will lower the stress on developers and they produce better software. The metrics are not controlling them, but warn them in time if something tends to go in the wrong direction.

As was said before, metrics should serve a goal. True, but on different levels in the organisation there are different goals. Often these goals do not match or support each other and this will make people insecure or even irritated (figure 2).

The Goal-Question-Metrics method works fine as long as goals fit together and just-enough of the proper metrics are collected. Every company has different goals. Maybe not the main goal, but the translation in sub-goals is. This often leads to different metrics and makes it difficult to compare companies. Another issue is that companies not often reveal the internals of their quality and measurement programs to the outside world.

Our experience is that metrics are great for internal communication and play an important

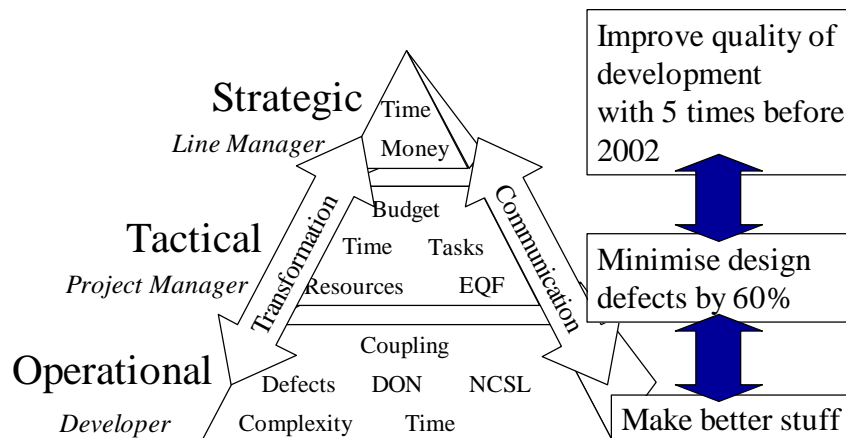


Figure 2

role in early warning.

2.2. Stress

There are many different types of stress, but here we deal with project stress.

In this article we do not deal with stress coming from the personal private situation. However, it should not be under-estimated. It often can be a cause of dysfunction in a project and can make people "uncontrollable".

A certain amount of stress is necessary for us in order to get things done. A stress reaction causes an increase in certain hormones and this in turn changes the state of our body. This is great when one has to run fast while the house is on fire. After a stress reaction one needs to recover from it. During stressful projects the time for recovery is often not available. This in turn increases stress. Not only bad for project control, but medical science proved that it also damages your health. No wonder we have more people with burn-out nowadays.

2.2.1. Impact on quality

When under high stress, one starts to make mistakes. These mistakes cause more work, more stress, and more mistakes. Quality will suffer, and often quality control is abandoned in order to 'save' the project.

Another measurement we have used is the Dirty Word Scan. This little program parses source code looking for 'dirty words' like: never, bug, wrong, fix, problem, crash, hack, illegal, maybe, later, future, etc. It also includes the words one can not write down here. With this we located the most frustrated programmer...

2.2.2. Impact on productivity

Although a person under high stress works faster initially, he/she will be less efficient in thinking, finding solutions, etc. More stuff is produced, but it often is not the right stuff. On many projects we've seen this. It looks that more is done, but suddenly it becomes clear that the implementation of the real functionality is still behind.

2.2.3. Impact on predictability

When stress will hit the project is actually not predictable. However we can see the symptoms indicating that stress is arriving on your project and will soon hit with force.

We all know that $\text{effort} * \text{time} = \text{constant}$, and that when one has much time left, one will not invest much effort. It also means that when the time we have to finish the project is almost zero, the effort will increase enormously. And this is often not planned, of course.

How can stress be made predictable? Next to indicating the complexity, and re-use level for specific parts of a system, you can also give an indication of expected stress and how well the team can cope with the stress. Further, you should determine the impact of stress on the overall project. Actually, it is just risk management. Stress is a risk factor.

2.2.4. Impact on project control

When stress hits the project, and it will of course be in a way that we not notice, project control is often more difficult. The above paragraphs explained why.

2.3. Technology Induced Anger

Technology Induced Anger (TIA) is a term we derived from Technology Related Anger. What does it mean? Dead-lines are set, stress is at its maximum, the stuff is almost finished ... beep ... the system hangs for no apparent reason, the network goes down, your word processor crashes, your internet browser disables your system, documents are lost, the CASE tool does not support the notation in the right way (believe me, the tool exists), etc. Your blood pressure rises, irritation increases, you can't keep the dead-line. Looks familiar, doesn't it? Unfortunately it happens all too often. Our development projects rely on technology, our estimates are in a way that we do not account for bad working systems.

What about the human factor? Irritation, stress, and anger are bad for your health, but also influences quality and productivity. We cannot see things as clear anymore and make mistakes.

Literally hitting the computer and call it names are symptoms. High blood pressure and even burn-out can follow.

2.4. RSI

RSI, or Repetitive Strain Injury, is a growing problem nowadays. Due to constant stress and TIA, we increase the probability of being one of the persons with this 'mouse arm'. We work hard and we work long, not noticing the first symptoms. After a while it becomes a chronic disease and it takes years to heal, and sometimes one has to change jobs because it will never heal again. This human factor influences projects of course. Project managers should make sure that people that are 'mousing' too long should take a break. Although discussing all details of RSI is beyond the scope of this paper, a short list of symptoms is given: wrong posture under high stress for a long period of time, arms are often tired, arms feel heavy, hands are stiff and often cold, tingling in the hands, difficulty to keep things up (e.g. brushing teeth, shaving).

2.5. Office environment

Office Environment has a big influencing on the human factor. Open offices have a lower productivity than closed offices. We see an increase in interior design for project environments as well. The results are promising. Large companies start to use more and more concepts of Feng Shui – the Chinese art of placement.

3. Models

In order to understand how people react in certain situations, we start to look for alternative models. The problem with ‘techie’ people is that ‘when a computer states it, it must be true’. Even when the program is wrong. Understanding means learning and looking for alternatives. This is what western medicine did when it incorporated acupuncture, for example.

The following alternative methods gave interesting results on real projects:

?? Team behaviour model: how does the team behave in normal conditions and under stress?

?? Person relation model: how do people relate to each other. Can they work together?

?? Intuition model: experience and people skills.

3.1. Team behaviour model

This model gives an indication of how the team and its members will react in normal and stressful project conditions. Stressful here means real project stress over a longer period of time. Unfortunately this paper does not allow to go into detail, but instead an overview is given.

This model recognises twelve types of persons. These are: Allocator, Motivator, Supporter, Planner, Decision-Maker, Consolidator, Inspirer, Evaluator, Producer, Sorter, Co-ordinator, and Communicator. These types are divided into nine categories.

How does this apply on real projects? See figure 3 below. These concern a project of 18 persons. This model indicates what will happen when stress appears on the project. Well, no decisions are made and not a lot will be produced. Note that even when this team does not experience stress, they still have difficulties with making decisions. Nice to see that stress did hit the project indeed not a lot happened except from great brainstorm.

This again shows this usefulness of these models for providing us with more information about the project team.

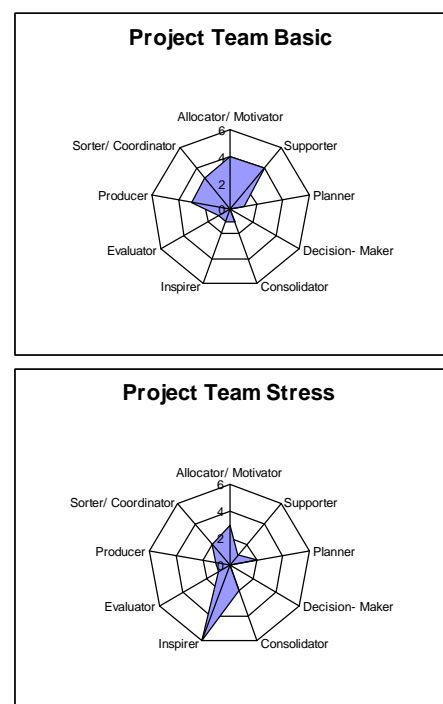


Figure 3

3.2. Person relation model

This model is very simple and gives good results. On medium large project we were not sure if three people could really work close together on a sub-project for building a very important part of the system. By using this model we predicted that they could not. And by ‘not’ we meant NOT. The project manager was not a great believer in the ‘soft-approach’ and disregarded the advice. One month later the first sign of problems showed itself and two months later they started to fight. Not only was the relationship damaged, but the whole project suffered on a motivational level. Also the believe in the project manager was damaged. Result? An uncontrollable project. A lot of work had to be done to restore people faith in this project.

What was the method used? Simple. Chinese astrology. Probably normal astrology would do as well, but we had more experience with the Chinese variant and only in relationship issues, not predicting the future.

The problem above was the typical so called rooster-rabbit situation. Soft? Maybe, but the manifestation was very tangible and visible.

3.3. Intuition model

Actually intuition should be a basic skill set of every project manager. It is based on experience and people skills. This means: how well can you detect interpersonal relation problems and how well do you understand people and how they behave for what reason. Are you product driven only, people driven only, or both. Do you understand stress and what it can do to your project?

4. Laws and metrics

There are some general laws that apply on these situations. These are:

- ?? “Always do one thing less than you think you can do” (B.M. Baruch)
- ?? “We don’t have time to think about the project, only about doing it”
- ?? “We never have time to do the project right the first time. We always have time to do it over again”
- ?? “Nobody can work more than 40 hours a week, at least not continuously and on a level that is needed for creative work”
- ?? “People under time pressure don’t work better; they work faster”
- ?? “People under extreme time pressure will start to sacrifice quality”
- ?? “Quality beyond the demands of the end-user is a way to higher productivity”
- ?? “Organisational work has the tendency to grow until it fills a working day” (Parkinson’s Law)
- ?? “The function of a manager is not to put people to work, but to make it possible for them to do their work”
- ?? “There are a million ways to lose a working day, but none to get one back”
- ?? “You’ll need 15 – 20 minutes to get back in concentration after an interrupt”
- ?? “100 gr of application is worth 1000 kg of thinking” (Booker’s Law)
- ?? Efficiency-factor = (# uninterrupted hours worked) / (# total hours present)

5. Conclusion

The human factor is predictable and unpredictable. But with more ‘soft’ techniques and models available we are able to understand more about it. By having these models and opening our minds for different approaches, we better understand the impact of stress, burn-out, Technology Induced Anger, RSI, and office environment. The better we understand it, the better we can deal with it in our risk management plans. Understanding opens the road to controlling and that in turn opens the road to managing a project. Often the human factor is not as unpredictable as we like to think.

6. Checklist

This checklist can be helpful in planning the Human Factor.

- ?? Do people have to perform too many different tasks?
- ?? Is it clear to people what is expected from them?
- ?? Is responsibility and authority in balance?
- ?? Is the team consisting of ‘career hunters’ or people that want to get the job done?
- ?? Is there a balance between novice and expert team members? (note: if they are all experts, the project will take longer!)

- ?? Do people feel involved?
- ?? Does the steering committee determine the estimate or does the project manager?
- ?? Are there clear goals on management and project level?
- ?? Do these goals support each other?
- ?? Are these goals realistic?
- ?? Do you know how these are measured and how the results are presented? And to whom?
- ?? Do you know how you will deal with conflicts? Do you understand the limits of your abilities?
- ?? Is the project a challenge for the members? Or is it just another project no one wants to do?
- ?? Do you know how to motivate yourself and others in stressful times?
- ?? Are the metrics collected during the project support decisions or do they measure people?
- ?? Do you know how you will reward people if a deadline/milestone is reached?
- ?? Do you know how to delegate work?
- ?? Do you often work long hours? Others in your team maybe?
- ?? Are metrics in place to support communication and early warning of problems?
- ?? Do people know exactly what is expected of them?
- ?? Do you recognise signs of burn-out? (signs are for example: disappointment, tiredness, not interested, no motivation, quickly irritated, negative behaviour, careless, sloppy, can't concentrate, can't make decisions, often headache, lower back pain, angry, aggressive)
- ?? Is the project manager planned on the critical path?
- ?? Do you understand the behaviour of the team under high stress? Is this planned in your risk management plan?
- ?? Make sure you listen to people when they express emotions. As a project manager you can learn from it and it will warn you.
- ?? When working, are your hands, or one hand, often cold?
- ?? Is one of your arms feeling 'heavy' or tired during work?
- ?? Do you take enough breaks? Do you take a short walk after lunch or do you eat behind the computer?
- ?? Do you often work more than 40 hours a week?
- ?? Do you often eat a warm meal after 8.00 p.m.?
- ?? Do you often have headaches? Especially if you have a difficult task to perform?
- ?? Do you often perspire a lot during the night when sleeping?
- ?? Do phones often ring in your working area?
- ?? Can you work undisturbed for more than an hour?
- ?? How do you feel, honest, when working in the project area or your office?
- ?? Do you often feel restless when working in your office or work environment?
- ?? Do you have a checklist with emergency procedures that tells you what to do when things go wrong on your project?
- ?? Do you enjoy to go to work every morning?
- ?? Do you feel happy and has life a meaning for you?
- ?? Do you do time management, and still have no time?
- ?? Did you attend this presentation?

7. Recommended reading

There are some interesting books you just have to read. These are: The soul of the new machine by Tracy Kidder, PeopleWare by Tom DeMarco and Tim Lister, Books on Mind Mapping and Speed Reading of Tony Buzan.

