

The Software Prince

Allan Baktoft Jakobsen

Abstract

In the year 1514, Niccolo Machiavelli wrote his infamous masterpiece “The Prince”. This bible of realpolitics holds such power to shock people, that Machiavelli has been accused of being in collusion with the Devil himself.

In software organizations, politics is often playing a major role, and people loosing the battle of power can end up being very frustrated.

Taking up the tradition of Machiavelli, this article gives some cynical advice to the people struggling for power in larger software organisations.

Who’s got the power?

Machiavelli’s unique book is a careful psychological study on how men of present and past won or lost their power. He reveals the dirty tricks of politics, and explains uncompromisingly what a *wise prince* should do in order to stay in power.

Based on experiences from various software organizations I will try to draw some parallels. The examples I use are based on real events as they happened.

First, consider who’s got the power in a software organization: The top management? The project leaders? The quality people or perhaps the developers?

The quick answer is *the top management*, (especially if you’re one yourself...) but in larger organizations, top management doesn’t have the detailed information needed for deploying their power very effectively. They often rely on the *project leaders*, who run the daily business.

The project leaders organize the work, but they rely on the developers, who control the progress.

Top Management challenges

Suppose you are a top manager in a software company. You want the company to make more money, of course, so I guess you follow the sales closely. But if you want to dominate the rest of the organization, you face a huge challenge.

Take the advice of Machiavelli:

When states ... have been accustomed to living freely under their own laws, there are three ways to hold them securely: first, by devastating them; next, by going and living there in person; thirdly, by letting them keep their own laws.

First, you can of course *fire* disobedient subordinates, but listen:

If ... it proves necessary to execute someone, this is to be done only when there is proper justification and manifest reason for it.

Machiavelli continues:

...cruelty is used well, when it is employed once for all ... Cruelty badly used is that which ... as time goes on ... grows in intensity.

So the point is that if you want to fire someone, the case must be clear, and it seldom is in a political game. Moreover, firing people is something you cannot keep doing or you will be out of business. (Maybe you'll be out of business anyway...) Thus, devastating people is not a lasting strategy.

Now, if you are an ambitious top manager, you cannot accept the third option of letting the organization keep its own laws (though that's exactly what often happens).

This leaves you with the second option: To go and live there, that is, to be present in your organization and to be committed. That's why *management commitment* is always stressed when making changes.

Unfortunately, top managers often make the same classical mistakes like e.g. Louis XII, King of France 1498, who inherited claims in Italy:

Louis had ... made these five mistakes: he had destroyed the weaker powers; increased the power of someone already powerful in Italy; brought into that country a very powerful foreigner; stayed away from Italy himself; failed to establish settlements there.

They remove the power of the quality people and give it to the group of project leaders for which the most powerful is made boss. Then they fail to be present in the organization, to listen in, and to be truly committed.

The wise project leader

Next, put yourself in the position of a busy project leader operating in a larger organization.

Your goals are simple: You want to stay in power and you want to conquer bigger and more prestigious projects.

You must have your goals clear in mind in *anything* you do. Never go to a meeting without knowing exactly what you want and without a hidden agenda on how to get it.

Having identified your goals, we can discuss your means. Machiavelli says:

There is no doubt that a prince's greatness depends on his triumphing over difficulties and opposition.

I recommend that you use any chance of creating an appearance of *chaos* and lurking catastrophe in the project. Take Alexander VI, Pope 1492, as an example:

What he had to do ... was to create disorder, throwing their states into a turmoil, so he could win secure control of part of them.

I have seen lots of people being fooled by this trick and it really is a wonderful double safety measure: If something goes wrong, you just pick from your *list of excuses*. If you, on the other hand, succeed, you are a hero. And top management *never* removes a hero.

Public and private plans

As a project leader you have an agreement with the company to deliver some product. The *project plan* can be seen as a *contract* on this deal. Naturally, you want this contract to be in favor of you, but if you sense that this is not possible, you want it to be *vague*. A tough project plan ties you up and limits your power to control the project. When you later want to change decisions, people will say “but you wrote that...” Eventually, your credibility will suffer. As Machiavelli wrote:

...a prudent ruler cannot, and must not, honour his word when it places him at a disadvantage and when the reasons for which he made his promise no longer exist.

Don't admit this in your attempt to avoid a precise plan. Instead, put it this way:

“This project is like a *new world* to us. Only, the night before release, we can make *realistic* plans. We are, however, driven by a *common vision* and besides: It's like chess: The most important move is your next. Let's focus on that.”

I conclude, that regarding planning, a wise project leader should beware the *output*, i.e. the actual plan. No good leader, however, would skip the planning *process*. Therefore, the existence of both a public and a private project plan is quite natural. They serve two completely different purposes.

In general, never be too eager to follow any corporate procedures (e.g. a quality system) the company may have established in order to control your activities.

...if a prince wants to maintain his rule he must learn how not to be virtuous.

Just say that all this bureaucracy isn't bringing much *value addition* to your project.

War

When it comes to *war* in a software organization, it's likely to be a fight of *human resources*. If you don't know this game you first loose respect, then you'll eventually loose power:

... a prince who does not understand warfare, as well as the other misfortunes he invites, cannot be respected by his soldier or place any trust in them.

There are many techniques of this warfare. The bullets are arguments and actions. Just to mention a few:

“Projects starting up are always difficult and unstable. So they need the best and most competent people which are to be taken from all other projects that are running and therefore stable.”

Or how about this:

“When stealing a particularly resource, go to his office with some prominent allies and stress the importance of his qualifications and the gravity of the situation. Use the *salami tactic* and don't involve his boss.”

Attack the other project leaders indirectly on their *competence* or on their company *idealism*. Whatever you do, take the offensive:

...if you do not declare (war) yourself you will always be at the mercy of the conqueror.

But beware: If your top management is clever it will interfere in the resource war, and here *you* are a resource, too!

A clever top management can react to power building up by changing project leaders frequently or even to appoint two leaders for the same project for some period.

When top management is interfering, you should watch your allies and the people you have suppressed in your quest for power:

...as soon as a powerful foreigner invades a country all the weaker powers give him their support, moved by envy of the power which has so far dominated them.

Discourage “improvements”

Being a project leader, your decisions are affecting other peoples' work in the company. They want to have influence so they come to you with all kinds of advises to change:

A prince must ... always seek advice. But he must do so when he wants to, not when others want him to; indeed, he must discourage everyone from rendering advice about anything unless it is asked for.

You should not listen to their silly ideas: In the best case they don't understand the problems you are facing in your position. In the worst case, they are manipulating you to gain power themselves.

As the wise leader you are, you should avoid radical changes. In particularly, getting involved in SPI's (Software Process Improvements) is dangerous. Listen:

...there is nothing more difficult to handle, more doubtful of success, and more dangerous to carry through than initiating changes in a state's constitution. The innovator makes enemies of all those who prospered under the old order and only lukewarm support is forthcoming from those who would prosper under the new.

It is not in any project leader's interest to be at any higher level of "maturity" than CMM level two. The *repeatable level* means you're keeping your power.

Lucky for you, SPI's are easy to fight since few has yet convincingly proven the economical benefit.

Consultants stink

Top management sometimes hires people from outside in an attempt to change the situation. They are a *pain*, of course, but you should not take them seriously:

Wise princes ... have always shunned auxiliaries and made use of their own forces. They have preferred to loose battles with their own forces than win them with others.

There are two kinds of consultants: First, the academic type, that haven't got any real experience. They are as easy as they are naive. They tend to talk in some abstract terms they call *processes*, instead of focusing on people and concrete tasks.

Secondly, there is the practical type of consultants that left development because they were too weak to change the organization from inside. Machiavelli would have called them weak princes in refugee planning conspiracies. Such men often fear their own failure.

Fortunately, the more fanatic and religious these smart-guys are compared to the organization, the easier it is for you to play the role of "mediator". This puts you in a position to hide your own ideas in the compromises thus getting it your way.

When dealing with consultants, let them know, occasionally, who's in charge. Say: "I am your *customer*. You have to *sell* your ideas much better."

The goodwill of the people

Though consultants themselves are too weak to threaten you, they may spread nasty ideas. As for the real dangers to a project leader, Machiavelli points to the following:

There are two things a prince must fear: internal subversion from his subjects; and external aggression by foreign powers.

In my translation, a wise project leader should therefore fear the top management and the anger of the developers.

Machiavelli stresses again and again the importance of the goodwill of the people. And you can easily get the anger of the developers, since many of them keep insisting on doing a "great job":

The people are more honest in their intentions that the nobles are, because the latter want to oppress the people, whereas they want only not to be oppressed.

Remember that *men must be either pampered or crushed*, so you must praise your developers constantly and give them presents at kick-outs. You must control what information they get and observe their lines of communication (make a *gossip matrix*). Above all, keep them busy (*overload* them) with their beloved technology. If you have to make unpopular decisions blame the top management. Avoid writings on negative issues.

As for the people you cannot buy, neutralize them, frustrate them, and then get rid of them (if they haven't left already). In this process, avoid public confrontations. Use *non-decisions* instead, that is, hesitation to make clear decisions on issues important to them (i.e. deliberate *laissez-faire* tactic).

I believe the worst thing that can happen is if top management persuades the developers to change or even worse vice versa. Management commitment and bottom-up improvements² are a deadly cocktail for project leaders. Fortunately, top management seldom pays attention to the people below the clouds of power in the organization.

The morale of this story is perhaps, that politics in organisations should be exposed, for example through a trisection of the power, and that top down and bottom up approaches should be combined when changing the organisation.

References:

1. Machiavelli, Niccolo, "The Prince". Translated by George Bull. Penguin Books.
2. Jakobsen, A.B. (1998), "Bottom-Up Process Improvement Tricks", *IEEE Software*, Jan./Feb. 1998, pp. 64-68.

Allan Jakobsen has an educational background in mathematics and physics from Copenhagen University and Dartmouth College. He has been working in the software business with maintenance, design, coding, test, quality assurance, and as an SEPG member. He is now working for the Danish company DELTA as a consultant in best software practices and process improvements.