

# ...SABOTAGE

CAN OUTWEIGH

# PRODUCTION



PRODUCTION OF U.S.A.C.  
INTELLIGENCE SECTION

**IF YOU QUESTION IT-REPORT IT!**

COME  CALL 242  or WRITE  - INTELLIGENCE OFFICER.

# Sabotage at Modern Workplace

Tomas Kejzlar

This book is for sale at <http://leanpub.com/modern-sabotage>

This version was published on 2016-02-16



Leanpub

This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2015 - 2016 Tomas Kejzlar

# Contents

<b>Versions of this book</b> . . . . .	<b>1</b>
<b>Preface</b> . . . . .	<b>2</b>
<b>Why sabotage?</b> . . . . .	<b>3</b>
Motivation for sabotage . . . . .	3
Motivation for stopping sabotage from happening . . . . .	5
<b>Types of sabotage</b> . . . . .	<b>6</b>
Pseudo-productive sabotage techniques . . . . .	6
Obstructive sabotage techniques . . . . .	6
Destructive sabotage techniques . . . . .	6
<b>Common sabotage techniques</b> . . . . .	<b>7</b>
Hold many long meetings . . . . .	8
Never decide anything by yourself . . . . .	11
Never let any decision to be made . . . . .	14
Always advocate “caution” . . . . .	16
Misunderstand orders . . . . .	19
Insist on perfection . . . . .	20
Use documentation to your “advantage” . . . . .	22
Postpone any tough conversations . . . . .	24
Punish anyone trying to “steal” control from you . . . . .	26
Write email, even if the recipient is sitting next to you . . . . .	28
Impress everybody by making many statements . . . . .	29

## CONTENTS

Cover up your mistakes (blame someone else, if you have to) . . . . .	31
Marginalize achievements of others . . . . .	32
Use your phone when on a meeting or when talking to others . . . . .	33
Build on functional expertise, discourage generalization	34
Never ask “why?” . . . . .	36
<b>Sabotaging organizational changes and adoption of new processes and tools . . . . .</b>	<b>38</b>
A methodology is just a word . . . . .	38
Create many new roles with complex descriptions . . . . .	39
Insist on creating formal processes . . . . .	39
Disrupt communication channels . . . . .	40
Add “social” aspect to all changes . . . . .	40
Use feedback in a clever way . . . . .	42
Create precise change plans . . . . .	42

# Versions of this book

This book is supposed to be a living document. We will do our best to add any new sabotage techniques that we discover (or are pointed to by you, our readers) and any new recognition & removal techniques we come up with or see being used and effective.

In order for all of us to keep track of all the changes we make, below is a version (release) history including major changes we've done in each version.

- 0.1 (05/01/2016), TK: initial version

# Preface

It may seem strange for us to write a book about sabotage. Surely, our job as agile coaches and change agents should be to help organizations succeed? That is indeed true, but sometimes, helping means showing bad examples and learning from them.

And that - in our minds - is the current status of organizational sabotage. It is everywhere around us. And most of the time, we are so accustomed to it that we don't see it. We see sabotage as "the way we do business around here", something normal and even valuable. Yet, sabotage is destroying our organizations. It is destroying creativity and freedom at our workplaces. It makes enemies from former friends and colleagues.

# Why sabotage?

Since proto-humans were competing for cave space, we've been sabotaging each other. Wherever there is a power relationship, with the subservient abused by the ones with influence, resources, and connections, at least a few of the underclass will be ready to take action. Without the means to fight back openly, other ways must be found.

Strategic Services, forerunners to the CIA, wrote the key principles for successful sabotage during WWII, distributing *The Simple Sabotage Field Manual* to Allied sympathizers resisting German occupation. Declassified in 2008, its remarkable how many of the Sabotage Field Manuals recommendations are applicable to the modern workplace.

## Motivation for sabotage

Your motivation as a saboteur can come from any source. Does your boss treat you like dirt? Is your work disrespected, while you toil long hours? Are you underpaid, under stress, and underclass? Expressing your feelings with a gun at the workplace is too extreme, especially given the inevitable consequences. But there are many opportunities to both ruin your boss's day and have a bit of fun while not breaking any laws at all.

And if you do it right, for all the harassment and demoralization you delivery, you could even get promoted.

Your priority is to not be caught. Always use techniques that appear to be innocent. Do nothing illegal. Carry no special tools or equipment, and certainly you should never hack into secure systems or attempt to cause any harm to people. Any of these amateur,

obvious, or illegal methods will be discovered sooner or later, and you'll be found and punished. Even in circumstances where your destruction could be blamed on large numbers of people, there's no reason to risk being caught or seeing the whole group punished.

Besides, its more effective and infuriating for your targets when everything you do is apparently innocent.

The Simple Sabotage Field Manual was aimed at workers who might set fires, short circuits, derail trains, and otherwise harass the German army during war time. But the authors (who remain secret to this day) included a timeless chapter on General Interference with Organizations and Production

This is the starting place for practical and lawful sabotage. The recommendations here are legal, so even if you are detected the very worst that can happen is you get fired. Deliciously these techniques are so common in our offices, logistics centers, and factories that you likely won't be noticed as a troublemaker...and you will be surprised how often you will be praise, power, and promotions for sabotaging productivity.

No one has spent much time in the modern workplace without noticing that the most incompetent are frequently promoted. One of the reasons for this is that they're actually quite good at sabotaging the work of others, thereby making themselves look good. At organizations with stack ranking, where a certain percentage are identified as stars and promoted while simultaneously an equal number are named losers and fired, sabotage is rife. Teams select weak members specifically so they can be blamed and let go, while the most talented seek out situations where only they can be productive, at the expense of everyone else, so they'll survive. If someone else begins looking too good, ways are found to ensure their failure, and engaging in political engineering is rewarded over any real engineering.

In such an environment, sabotage isn't an option. You must use it to keep your job and get ahead. With the advice in this handbook,

you'll have the tools needed to succeed.

This short handbook is organized into sections. We start with a review of the classic techniques which were offered to the partisans of the Second World War, followed by a section of suggestions for the office workers of today. Then we continue with a section specifically meant for IT workers and modern managers, including those who work in the trendy agile environment. We conclude with thoughts on long term sabotage, and how we can negatively affect the future.

As you are reading these sections, you may ask yourself what you can do to increase sabotage, or you just might notice that more than a few of these recommendations are already established as common practice at your place of work.

## **Motivation for stopping sabotage from happening**

TBD.

- sabotage does not increase your happiness at work
- sabotage won't help you achieve something great
- sabotage is ultimately the last weapon of the weak against the powerful - eliminating it means shifting long-lived paradigms about work and management

# Types of sabotage

Our main intention with this book is to make you aware of all possible sabotage techniques and help you fight it. Therefore, we have classified sabotage techniques outlined in latter chapters of this handbook into 3 different groups based on their impact. The three levels are:

- pseudo-productive
- obstructive
- destructive

Now, let us have a deeper look at each level and the danger it typically possesses. Later, we assign each sabotage technique we discuss one of these levels. When fighting sabotage, you will naturally want to eliminate those techniques, that cause the biggest problems - the destructive ones.

## **Pseudo-productive sabotage techniques**

The techniques described here can actually be a normal part of productive work, if done in a proper way and in the right circumstances.

## **Obstructive sabotage techniques**

## **Destructive sabotage techniques**

# Common sabotage techniques

All right, let's just jump straight to the point. And to do so, we will introduce some of the most common sabotaging techniques. Note that all of the techniques are completely legal and they usually require no expertise or advanced skills from the person doing them.

For every technique we mention, we also include some more practical and interesting information apart from the sabotage technique itself (which you will without a doubt find very interesting). These are categorized in two sections: *recognition* and *removal*. Pretty self-explanatory, here we try to collect the most obvious tell-tell signs of the sabotage technique in question so that you can recognize it easily. And in the removal section, we provide some hints how to remove the sabotage from your organization (or, at least, how to try to remove the sabotage from your organization).

## Hold many long meetings

Are you lonely?

Tired of working on your own?  
Do you hate making decisions?

**HOLD A MEETING!**

You can –

- See people
- Show charts
- Feel important
- Point with a stick
- Eat donuts
- Impress your colleagues

All on company time!



**MEETINGS**

THE PRACTICAL ALTERNATIVE TO WORK

Hold many long meetings

*Category: obstructive*

The longer, the better. And of course, make sure nothing gets decided on these meetings. To do so, never send agenda of these meetings in advance, so that you spend at least the first 10 minutes discussing what the meeting should be about.

While at it, make also sure that you invite as many people as you can to these meetings. After all, all of your co-workers should have their say in the matters that are being discussed - so that you are creating a harmonious workplace and applying democratic principles.

During the meetings, try to go off-topic as frequently as possible. Illustrate all of your points by your past experiences, preferably in form of long and complex stories.

Try to finish the meetings with more open issues than they started with. And suggest planning follow-up meetings to “discuss the few remaining issues”. This way, you and your co-workers will have more and more meetings to attend to and they will have less and less time to actually do something that produces value (you certainly do not want to produce any value! you want to have democratic workplace where everything is decided by communities).

### **Tomas: The meeting that never ended**

*I do not admit it easily, but I have successfully used this technique in the past (of course my sabotage has been morally justifiable). I have persuaded my colleagues to discuss possible reduction of our application documentation. What happened was miraculous - one of my colleagues, the chief analyst responsible for the documentation, immediately set up a meeting in order to create a “working group” to discuss possibility of documentation reduction. 6 people have been invited. At the first, hour long, meeting, nothing got decided and the conversation was mostly even not to the point. The only conclusion was to hold another meeting. The second meeting also did not bring any conclusion and so a third one - with more people to “aid the decision making process” invited. And guess what? Nothing got decided on the third meeting and another has been scheduled. And, slowly, the agenda of the meetings shifted from “we want do slim down documentation” to “we want to put even more effort in the documentation to make it the best”.*

## **Recognition**

- Look how many meetings are in peoples’ calendars. Filled calendars for week ahead or more are a tell-tell sign of this sabotage.

- Look at description of meetings and number of attendees. The longer meetings and larger audiences, the more probable this sabotage technique.
- Look for repeated meetings and / or for spin-off meetings, where the original issues is on the agenda for more subsequent meetings in a row.
- Watch how people react to issues - is their first instinct when asked to solve a problem to hold a meeting?

## Removal

- Get rid of some of the meetings. Start by cancelling subsequent meetings where no decision has been made.
- Insist on short meetings. You can introduce a hard rule that a meeting cannot last longer than 30 minutes and a maximum of say 5 people can attend.
- When participating on a meeting, always try to reach a conclusion or to decide as soon as possible. If it is clear that decision cannot be reached amongst the participants, cancel the meeting right at the start.
- If you still run serial status meetings, think about canceling them and gather status via e-mail (if you have to)<sup>1</sup>
- Ask other meeting participants what value the meeting had for them using ESVP<sup>2</sup> or a similar method.

---

<sup>1</sup><https://www.techwell.com/techwell-insights/2014/09/beware-serial-status-meetings>

<sup>2</sup><http://www.funretrospectives.com/esvp-explorer-shopper-vacationer-prisoner/>

## Never decide anything by yourself



Use committees to decide everything

*Category: obstructive*

Surely, you do not want to make any decision. Decisions mean something might actually get done and we want to stop that at all costs. If you are lucky enough, your organization already has a plenty of committees that you can ask for approval. And if not - suggest their creation and back the idea by exaggerating the impact of what you are deciding - for example point at the impact of this decision on company's strategy.

Alternatively, you can have your superiors decide. And while you are at it, don't ask for permission only once. If your superior decides or approves something, ask again, possibly raising some potential risks you have not mentioned before. The advanced alternative of this is to ask your direct superior and after he decides, ask his superior the same thing, possibly hinting on poor decision-making of your superior.

### **Tomas: A hierarchy of committees**

*In a large corporation I used to work for, we had some quite rigid processes, which resulted in the average delivery time of projects lasting about 24 months. The corporation, being pressed by competition, decided to change the processes and shorten the so-called time-to-market by 50%. So, the number of project phases has been reduced from 3 to 2, the first feasibility phase timeboxed to 3 months. However, each project wishing to move from the first phase to the second one had to pass the following committees: a pre-validation committee, an IT architectural review, an IT validation committee, a business validation committee and a steering of the project plus (in certain cases) a so-called "project-management-committee". Out of the three months timebox, only the approval process took (best-case scenario) 5 - 6 weeks and of course all the materials needed to be prepared before submitting the project to the first pre-validation. Those project managers who wanted to really deliver projects found some clever - but methodically not allowed - solutions, while the ones just surviving did everything by-the-book resulting in delays, confusion and frustration of people actually doing the project work.*

### **Recognition**

- Look at how decisions are usually made. Do you need numerous approvals for every decision / proposal?
- Study your current processes. Are there any formal check-points where you involve committees to make a decision (or validate it) while these committees have no added value?
- Calculate how much time on project is spent delivering something of value as compared to validating, presenting and submitting information to committees.

- Analyze the overall decision-making style of the organization. Hierarchical organization with strong top-down decision making processes usually suffer much more from committee overload.
- Look how your colleagues approach decision making - are they afraid to make decisions and usually want to have a higher level of approval?
- Look at management practices. Are managers willing to be held accountable for their decisions? Or do they rather decide in groups so that they can be collectively un-accountable?

## Removal

- What you can, decide on your own. Try to push decision making (authority) to the lowest possible level where the information necessary for the decision resides.
- When faced with a committee, try at least to get the decision per-rollam. If that is not possible, try at least to submit your proposals in such a way, that you propose a decision / solution and the committee only accepts that, minimizing unnecessary discussion among people who lack the information necessary to make correct decision.
- If you have the authority, eliminate some of the committees and have people accountable for decisions made instead of committees.
- Try to create *delegation boards*<sup>3</sup>, clearly stating how various kinds of decisions are made. When you have them, try to delegate as much as possible to lower levels. This builds accountability and trust.

---

<sup>3</sup><https://management30.com/product/workouts/delegation-employee-engagement/>

## Never let any decision to be made

*Category: obstructive*



A decision is a decision only if you want it. Otherwise it is just a temporary setback. - Humphrey Appleby.

And if it has been made, try to open the topic again at the next meeting. Again, point at associated risks (however theoretical they might be). Better still, insist on written minutes from each meeting and when their draft is distributed, reject them because of bad wording.

This is especially well done in today's world of technology. When minutes arrive to your inbox, find (you may even at random) a statement you believe is not written exactly as has been presented at the meeting and raise an objection. And of course, send your objection to all of the attendees and carbon copy to your superiors, their superiors and other people you feel may be interested. By doing it this way, you have disrupted the work of many more people who have no information and who will probably misunderstand and join the discussion with completely irrelevant solutions. And, after the flame war has begun, be the one who is the only sane one - suggest the best place to clear all the miscommunication would be

another meeting.

**Tomas: e-mail wars over minutes**

*Quite some time ago now, I have attended a meeting regarding setting up some processes in JIRA software. The meeting itself went (from the sabotage-driven mind perspective) horribly well. Only the people who actually knew something about the topics have been invited, the discussion was nicely moderated and after some 30 minutes, an agreement has been reached - introduction of new state "analysis" in the JIRA workflow. But, when the minutes came later that day, one of the participants quickly responded that he wanted the status to be named "business analysis" (something completely irrelevant to the issue being solved by the new state). In no time, an e-mail war began and slowly, more and more people were in the communication, each of them trying to help, but in reality creating more confusion over a trivial matter. At the end, another (huge) meeting has been scheduled and the original decision has been revoked and discussed over and over again, in the context of the entire company (and the entire universe as well). This had an appalling effect on satisfaction of everybody in the original group and resulted in these people never proposing any good changes again.*

## Recognition

- Look at the usual decision-making process at your organization. Are decisions made quickly, or are many people involved to verify any proposed decision?
- When a decision is made or an agreement is reached, is it usually then disputed by someone?
- Are minutes from meetings frequently commented upon and changed many times before they are finalized?

- Are people generally allowed to make any decisions, or are the processes so complicated they actually prevent anyone from making any decision?

## **Removal**

- Try to simplify decision-making processes as much as possible. Give power to the people who have the information necessary to make a good decision and trust them.
- Limit the number of people within the decision-making process. Eliminate the ones that are in the process just because of their authority.
- Look for long e-mail discussions over precise wording of minutes and for people engaging in these discussions - it is probable they are (unknowingly) sabotaging your company.

## **Always advocate “caution”**

*Category: destructive*



Always advise caution and responsibility

Like with the previous techniques, you do not want any decision to be made. What is decided may actually be worked on and we cannot allow that. So, when - eventually - something gets decided, try to change the decision by advocating “caution”, pointing at possible “risks” and “protecting the company from harm”. Insist on further discussion - preferably by planning new meetings to tackle the subject or by referring the matter to a committee.

This technique is especially useful when dealing with innovations or other new stuff. You can point at various risks connected to innovating - wasted money, not fulfilling clients needs, taking “senior resources”<sup>4</sup> from their current assignments. When in contact with an innovative idea, suggest a in-depth analysis, market survey or just plainly dismiss the idea as not feasible (you can use the *we have tried this in the past and it did not work* phrase).

## Recognition

- Look at how the company approaches innovative ideas. Do they usually require some formal approval and validation

---

<sup>4</sup>For more about resources sabotage, see the [Replacing people with resources](#).

process?

- Try and measure queue lines<sup>5</sup> you have. The longer the queue lines, the slower the actual delivery is and the fewer satisfied clients of your products.
- Listen for classical signs of this sabotage technique like *this needs more thinking, good idea, but our clients wouldn't use it, we have tried something similar and it didn't work out, we have more important things that to do this nonsense.*

## Removal

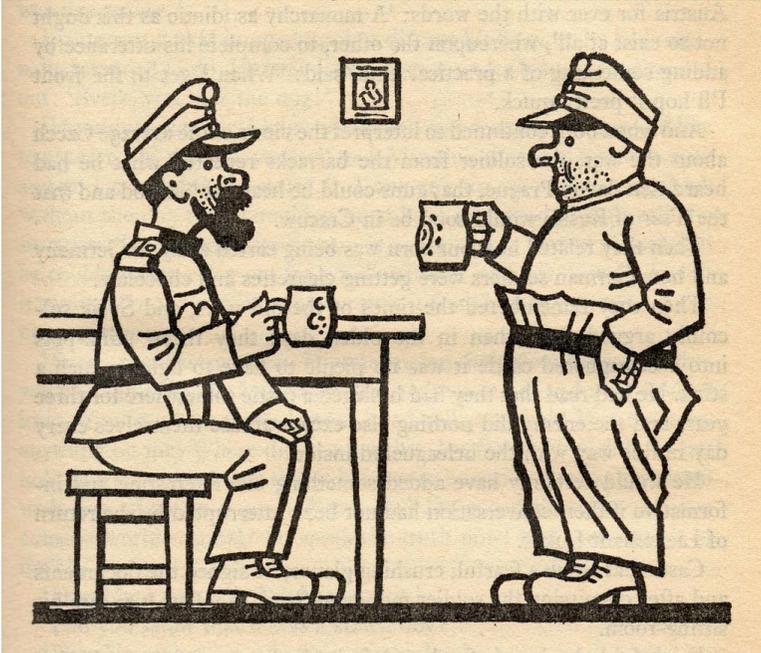
- Give people the time and space to innovate - separate from their regular work.
- Try to create regular *Shiplt*<sup>6</sup> days to promote innovation and generate new ideas for your business (this also helps all of the employees to engage more).
- Look at people who are constantly blocking innovation and advising caution and try to move them out of the way (these people can be good QA staff for instance).

---

<sup>5</sup><https://management30.com/product/workouts/problem-management-improvement-management/>

<sup>6</sup><http://www.danpink.com/2011/07/how-to-deliver-innovation-overnight/>

## Misunderstand orders



Svejk, a fictional character who followed all orders exactly

Pretty obvious. If any order can be misunderstood, misunderstand it. Do not ask anyone for clarification, just do it and perhaps ask for clarification afterwards (so that you have spent time doing something absolutely worthless). When someone tries to blame you, point at the processes and try to setup new meetings to discuss dysfunctionality of these processes - hopefully you will be able to replace them with much more complex (and error-resistant ones).

The opposite of this - equally effective - is to follow orders exactly. Again, do not ask for clarification, just do it. And if questioned, you can do the same as with misunderstanding.

Both of these techniques can also lead to the necessity to make many

more written orders to prevent miscommunication. Insist on their perfection and formalities, like sign-offs and so on. Again, the rule here is to avoid doing any actual work.

## Recognition

- Look at people around you. Do they use their brains or just follow orders you give them without thinking?

## Removal

- When giving orders, be as clear as possible, not allowing any room for misunderstanding. Better still, try to give as few orders as necessary and let the people move up the leadership ladder<sup>7</sup>.

## Insist on perfection

Surely, you do not want to present imperfect results to your clients. Or to your peers. Therefore, insist on perfection in all aspects of your work. Always fix all bugs you find in a software, even if that means missing the release date or not working on new features. Check and double-check everything, especially the formal side of things and adherence to existing processes.

Never accept any shortcuts and insist that processes must be obeyed. After all, they are there for a good reason.

### **Tomáš: how we fixed a bug and made clients unhappy**

*I remember that once, we were releasing our huge application. Before we made it available to clients, people from the company*

---

<sup>7</sup>[https://www.youtube.com/watch?v=QYf\\_8Yrr\\_pw](https://www.youtube.com/watch?v=QYf_8Yrr_pw)

*made some final “acceptance” tests. They found several bugs, including one where there was essentially a typo in text. However, the business people insisted on fixing the bug before the software went live. All of this was happening during the weekend, and in the small hours of Sunday, the developers just made a quick fix, ran some basic tests, made a new build of the application and deployed it. Fast forward to Monday morning? call centre is flooded with complaints from clients who are unable to make foreign payments. A quick analysis and fix is deployed. A subsequent analysis reveals, that by fixing one line of text, a new and very serious bug has been introduced and in the haste no-one discovered it.*

## Recognition

- Look at how bugs or defects are treated - does every product have to be completely bug-free before you present it to clients?
- Are formalities more important than the actual product? For example, do you delay releasing a perfectly good product because of incorrectly filled documents?
- Do you have a separate QA team, or even a QA department, whose primary function is to check how others adhere to prescribed processes and creation of new processes that take care of all found exceptions?

## Removal

### Use documentation to your “advantage”



#### Advanced method - over-document everything

Quite simple. When developing new software, don't write any documentation. After all, the code by itself is self-explanatory and the documentation will be only additional workload.

The opposite, and more advanced method of doing this is **overdoing the documentation**. Document everything in the maximum possible detail and create an elaborate structure of the documentation. However, make sure that the documentation is really inconsistent, that you are using different acronyms across it and possibly store it at different places. Also, make the documentation really hard to change and scatter the information between multiple files / pages / documents without providing any link. That way, the

documentation will become useless without any direct intervention or intent.

**Tomáš: how documenting everything made (almost) everyone unhappy**

*When I was working at a big financial corporation, we managed to sabotage ourselves in exactly this way. Our system, the internet banking platform, was considered a business-critical one, with dozens of developers taking care of both the maintenance and new feature development. This led to generating huge amounts of documentation. Very soon, this documentation became so extensive that it was barely maintainable and many analysts were more documentarists. And more problems started appearing - some people were so disappointed they stopped updating the documentation, which made it less and less useful (but others were still working with it, causing even more troubles). Business people also were not happy, as the extensiveness of the documentation made it very difficult to find anything. Then we had a great idea - to create “upper layers” for the documentation and delete the low-level pile of mush. First part, creating the upper layers, has been quite successful. But deleting something was not - somebody was always claiming he needs that part for his work. So in the end, the documentation grew even bigger.*

## Recognition

- How detailed is your documentation? Do you have any? Or does it cover everything to the greatest possible detail?
- How much time do people usually spend documenting in comparison to actually thinking and doing work?
- Do you have many formal procedures regarding documentation - prescribing how to document, to what detail and so

on?

- Is there a special process or a tool to ensure that the documentation is formally correct? Do people rectify found errors prior to doing development work?

## Removal

- Do not forget the real meaning of Agile Manifesto when it comes to documentation - *working software over comprehensive documentation*<sup>8</sup>, which means you should document on an adequate level and never give priority to documenting before making the software you are documenting actually work.
- If you have an extensive documentation, try to remove some of the low-level parts of it and watch if they will be missed by someone.
- If you do not have any documentation, write some. Focus on high-level things that cannot be easily discovered other ways (i.e. by looking at the code of the software you are building or at the product you are selling).

## Postpone any tough conversations

Perhaps you have a problem with your colleague. Or you have made a commitment you won't be able to keep. Don't tell anyone in advance. Do not try to solve things. Solving problems with your colleagues by talking to them is silly anyway, right? And you may be viewed as the troublemaker. And when you feel you won't be able to keep a commitment you've made, you may be seen as a low-performing worker. Better not to let anyone know - a bigger problem might occur that will hide your mistake, or someone else

---

<sup>8</sup><http://agilemanifesto.org/>

might screw up and then you can just blame them and keep your good reputation.

By postponing any difficult conversation, or by refusing to admit you will not be able to keep commitments you've made you are making the problem only worse. Unsolved problems with your colleagues will start manifesting themselves in your daily interactions, whether you want it or not. And when you don't tell anyone you won't be able to deliver something you've previously committed to, you are giving them no chance to do something about it and prevent negative outcomes.

## **Recognition**

- How often are problems reported only at the end of a project, iteration, sprint, or generally when the deadline approaches?
- Are there any tensions between people working together? How often are these escalated to the management level without the people trying to settle them first?

## **Removal**

- Never punish anyone bringing bad news. If someone is unable to keep a commitment and tells you that, he should never be punished in the first place. He should be thanked for raising the risk without delay. If the situation repeats, perhaps this person should be offered some help (again, no punishment!) to better organize his time, to raise his awareness of the difficulty of the tasks he is handling and so on.
- Encourage direct talk between people and lead by example. Many people behave the same way others do, which means if you set a trend (or a behavior they will see and like), they will try to adopt it.

- Look for people who gossip and talk about others behind their backs. Firstly, explain to them that such a behavior is unwanted and harmful in the environment you are working in and ask them to stop doing it and solve problems with their colleagues directly. If they continue doing it, you may be forced to ask them to either stop doing it or leave the company (this is a last resort measure and you should never use it unless you've previously tried multiple different other options).

## **Punish anyone trying to “steal” control from you**

So, you are the boss. And someone approaches you with an idea and wants to run an experiment. Perhaps he wants to decide something. Never allow that. This person is certainly trying to steal control from you and supersede you. The only way you can ensure high productivity is by keeping things firmly under your control. And if anyone comes and wants to decide something, never mind how marginal it might be, don't allow that.

This has two very nice effects. The first one is that you have become a central decision-making point for everything. And thus the decision making process inevitably slows down - there are simply too many decisions to be made. If someone above you questions this, use the *“I know it may slow things down a little, but I need to be aware of what is going in my teams, otherwise the employees wouldn't do what we want them to do and they'd just slack off”*. That will do the trick and you will also reaffirm the command&control mindset within the organization and slow the decision making processes even further, as this may “inspire” other managers.

The second side effect is that you will undoubtedly start making some really bad decisions. This of course wouldn't be your fault at

all - simply, some of the crucial information necessary to do those decisions haven't been available to you. This is also a good excuse in case your superiors start questioning you - just blame the people on the team for not providing you with all the adequate information. Moreover, you may use this to actually appoint "chiefs" or "leads" within your teams, so that there is a clear responsibility to push the relevant information up the management chain.

In the end, this leads to making wrong decisions, disempowering the people doing the work and tightening the command&control mindset (which we know does not work in any creative environment).

## Recognition

- Does it happen that your superiors won't let you make any decisions?
- Do your superiors try to check and validate all the decisions you've made, regardless of whether they actually have the knowledge or information?
- Do you want your team members to ask for permission before they do anything?
- Are you afraid of people not doing what they are supposed to do without you micromanaging them?

## Removal

- If you are a team member trapped in this situation, there is not much you can do. You can start using other sabotage techniques outlined in this book to point at the absurdity of what is going on. If you are brave enough, you may suggest your managers to create a delegation board<sup>9</sup> or a ladder of leadership<sup>10</sup>, clearly stating who makes what kind of deci-

---

<sup>9</sup><https://management30.com/product/workouts/delegation-employee-engagement/>

<sup>10</sup><http://davidmarquet.com/LadderOfLeadership/>

sions; in the process of making this board you will have the opportunity to explain to him that some decisions should be made at the team level, where all the necessary information for making them resides. Or you can start looking for a new job.

- If you are a manager, try becoming a leader instead. And admit to yourself, that for some decisions, there are more appropriate people with greater knowledge to make them. Do not be afraid to delegate the power to your team members, but also never do it blindly, without first explaining what you want to achieve and making sure the team members have the skills to actually achieve it.

## **Write email, even if the recipient is sitting next to you**

Everybody likes using e-mail, right? So use it, even if the person you are speaking to is sitting in your close proximity. Speaking to someone may distract him from what he is doing, so it is better to communicate via e-mail, because one reads his e-mail only when he wants to.

Using e-mail extensively will have two great (for any saboteur) effects. Firstly, it dramatically increases the number of e-mails everyone in the organization receives, making reading and responding to e-mails very time-consuming and forcing people to dig through piles of unread emails instead of doing any real work.

Secondly, e-mail communication is not nearly as effective as face-to-face, so information will get lost, misinterpretations will happen and all this will result in delays and frustration.

## Recognition

- Do you receive e-mails regarding trivial matters from people sitting close to you?
- Are you sending lots of e-mails to people sitting close to you even when they are at their workplace?

## Removal

- Start by setting a personal example - when you get an email from someone sitting close to you, resist the urge to hit the *reply* button and use your two feet, come to the person and discuss the matter in person.
- Suggest to others that instead of writing emails, they could come over and discuss matters with you personally. Try to persuade them that it will be faster and more efficient - no loss of information and no need to type long emails.

## Impress everybody by making many statements

When you are at a meeting, try to make as many statements as possible. Don't listen to others, don't try to understand the situation they are in. Just give many statements, preferably quotations from books or (better still) from your own experience, articles, blogposts and so on.

Using this technique, it is almost certain that you will make all the meetings much longer than necessary, because people will generally be discussing the statements you make. Hopefully, people will also start trying to practically implement the ideas in the statements you make; however without understanding the situation at hand or the background, many of these ideas will be theoretical and simply wrong.

**Tomás: I am the expert, you all listen to me**

*I remember that some time ago, I have been working for a company that went through reorganization. Apart from making things worse (which is the result of almost all reorganizations), this one resulted in hiring some new people. One of them was a former manager from another big company, who was supposed to help the agile adoption. He wasn't a bad guy and had a quite good understanding of what I'd call technical agility (not knowing much about the principles, values, beliefs and what is the goal of agile but having a good knowledge about the events that need to take place, artifacts that should be present and roles that need to exist). However, at every meeting he was bosting himself that he is the agile expert and told the same story how he was the chief agile guy at another company over and over again. This shortened all the meetings (or their useful time), disgusted some people but affirmed the management in their persuasion that they have chosen the right guy.*

**Recognition**

- In meetings, are there people who are always trying to hijack them, not allowing anyone else to speak and sharing only their views, however irrelevant these may be?
- When you are deciding something, are you given the chance to explain the situation at hand and the background behind it and then come to the solution, or is the solution picked by the ones knowledgeable and aware of all the “best practices”?

**Removal**

- Never allow monologues to dominate meetings. Facilitate the meetings so they are *discussions*, not a *series of monologues*.

- Always look whether you have all the information needed when you are discussing certain theme. Never allow best practice to dominate the discussion.

## **Cover up your mistakes (blame someone else, if you have to)**

If you make a mistake, don't tell anyone. This is kind of similar to not letting anyone know you won't be able to keep a commitment you have previously made. When you realize you've made a mistake, your effort must be to find somebody else to blame for that, but keep quiet about it all the time.

By this, you are dramatically reducing any possibility of the mistake being fixed, because (if you are lucky) by the time the mistake is discovered, it is too late to remove it. And by blaming someone else, no harm can come to you. Alternatively, you can point out that the mistake (being a someone else's fault) is a result of an oversight in a process and suggest the process is corrected (= made much more complex to deal with unlikely situations).

### **Tomáš: attack is the best form of defense**

*One of my former colleagues was responsible for deploying mobile applications to the production environment. This was done via a web interface. One day, a new version was ready, but should have not been published yet. However, he was so confident that he did not notice the web interface has changed. And instead of just saving the version for later, he published it. This had a major impact on the users - the application stopped working after they updated it. Soon, panic ensued. The problem was, that the new version number was not in the systems and the users were denied access. And sadly, no-one had the rights to change the version numbers. After a solution has been*

*found (publishing another version with the right number), this colleague of mine did not admit he made mistake. No. First, he blamed Google - they changed the interface, bastards! Then, he blamed others who did not have access to the system controlling the versions - surely, it was their fault, because if they've had the access, the problem would never happen.*

## Recognition

- Do mistakes appear only when it is too late to do anything about them?
- Do people blame each other for mistakes?
- Do mistakes usually result in creating new complex processes, or even departments (such as QA and similar)?

## Removal

- Never punish anyone for admitting a mistake. Instead work together with them and help them fixing it before it causes any harm.
- Admit mistakes you've made yourself, contribution to the open culture you want to be a part of.
- When somebody is constantly blaming others for his mistakes, explain that this is an unwanted behavior and you want everybody to admit mistakes and learn from them. If the blame-game does not stop, perhaps this person is not the right one for the environment you are trying to build.

## Marginalize achievements of others

TBD (@Tomas).

## Use your phone when on a meeting or when talking to others

If you get a phone call during a meeting or when talking to someone, just go and answer it. You may pretend the call is really important (but it has to be believable as well, you cannot get away with ridiculous claims such as the president calling you, unless of course you are in direct connection with the president) to have an excuse (and by doing so you may also *spread some rumors*, which is another great sabotage technique).

Interrupting a meeting or a conversation with a phone call gives you the opportunity to later ask others to repeat what has already been said, prolonging the conversation. Practically, you will be able to achieve two things - delay everything and poisoning the environment (as no-one really likes to be asked to repeat something or to be constantly distracted by phone calls).

You may notice, that this sabotage technique has been so successful in the past that now many companies have at least some informal guidelines stating that people should not take phone calls during meetings. However, the funny fact is that they suggest you go out of the meeting, take the call and then return, which in terms of this sabotage technique is not a problem.

### Recognition

- Look at how frequently are people leaving meetings in order to “take important calls”.

### Removal

- Ask people not to use their phones during meetings. You may even ask them to put their phones in a box (or in a pile)

when they come to the meeting or to leave if they think the meeting is not important for them (you may also learn about the quality of your meetings this way).

## **Build on functional expertise, discourage generalization**

Insist on having experts for everything. And on that only the experts can do their stuff and no-one else. And if you are not an expert in some area, try to hand over the work to someone else, who is an expert, because clearly he will be able to finish the work in much less time.

Having experts for everything and discouraging generalization has two effects. Firstly, you will probably slow everything down because some of the experts will get overloaded with work. Also, during the numerous handoffs some information surely will not be passed along and delay and confusion will happen. Secondly, disallowing generalization will mean that people will less and less communicate with each other, they will tend to “mind their own business” and when problems will arise, they will try to blame someone else instead of trying to solve them as a team.

### **Tomáš: bring me the one expert**

*I remember that once I was suggesting some organizational changes - transformation from silo-based functional domains to cross-functional team - in one company. While I was presenting these thoughts to the CIO, he asked me to “show him the one person responsible for JAVA code” and “the one responsible for the architecture”. I tried to explain there is not such “one person” and that the knowledge and decision power go together and they are distributed amongst the team. He did not believe me and insisted on having that one “super-*

*expert". This wish, which (because he was the CIO, remember) eventually became reality led to disengagement of people and wrong decisions made by the "super-experts" due to them lacking some key information. However, at least there was someone to blame - from the CIO's perspective.*

## Recognition

- Is anyone at the company asking for experts?
- Does the work get handed over several times, so that the experts do their part?
- Is the decision making process centralized in the hands of few chosen experts?
- Are people allowed to learn new things outside of their primary domain of expertise, or do they stick in their one domain forever?
- Do you have cross-functional teams aiming for a business goal, or do you have silos where the overall goal is not known and every silo only does its part of the work?

## Removal

- When you have the opportunity to learn something new, do it. If you do not have such opportunity, try asking for it.
- Try to form teams around the work that needs to be done with everybody contributing instead of passing the work from one group of experts to another.
- Try to push decision making power down the hierarchy. There are several tools that may help you both explain and

ease this process such as Delegation Boards<sup>11</sup> and Ladder of Leadership<sup>12</sup>.

## Never ask “why?”

Never question any orders you get. After all, you are in the company to *do*, not to *think*. Thinking is best left to the managers and generally people up the corporate ladder. So, fulfill any orders you get without questions and to the letter.

Blindly following orders places all the decision power on your superiors. Even in the (highly unlikely) case they are really geniuses, bad decisions will inevitably happen, which will slow the company and result in some form of dissatisfaction. Remember, that even if your superiors are really great people, one great person telling 100 others what to do is much less than 101 great people sharing and participating in what to do and then taking action.

### **Tomáš: I know better than all of you**

*Once, I have been at a large company, and a new deputy CIO came in. He was indeed a very smart guy and had a decent background in both engineering and management. However, shortly after he came, he began directing all the engineers and telling them what to do - what frameworks should they use, how should they write the code and similar stuff. It went to the extreme that this guy, co-leading a department of a couple of hundred people, began reviewing code commits. I didn't have the chance to observe this behavior for long term, but very soon, the engineers stopped thinking about the problems and just did what has been ordered. This resulted in some poor designs and other development problems, as even this guy was really good, he could not match the knowledge and experience*

---

<sup>11</sup><https://management30.com/product/workouts/delegation-employee-engagement/>

<sup>12</sup><http://davidmarquet.com/LadderOfLeadership/>

*of dozens of developers.*

## **Recognition**

- How often do people question what has been suggested by management (or by others)?
- What is the decision making process - are people allowed to decide how they are going to accomplish their jobs, or do they get it prescribed?
- How often, when retrospectively a problem or failure, some of the engineers either directly or indirectly admit they knew about the problem, but just didn't want to bring their concerns upfront?

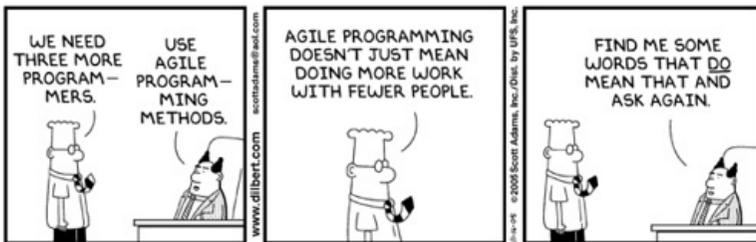
## **Removal**

- Encourage people to question things. Never assume that you have the ultimate solution to a problem.
- Allow everyone to decide how are they going to do their job. Probably, those people know what needs to be done and how.

# Sabotaging organizational changes and adoption of new processes and tools

Many organizations in today's world are constantly running change initiatives. And there are specific ways in which you can sabotage these initiatives and ensure that nothing changes. What is even better, you will be able to attribute all the dysfunctions to the new methodology.

## A methodology is just a word



### Any methodology is just a word

Any new methodology you are trying to introduce is just a word - like “agile”, “lean” or whatever is the silver bullet you have chosen.

So, when adopting it, do not change any processes, just change the names. For example, if you are adopting agile (common nowadays), just rename those long functional specifications as user stories. Or rename marketing people as product owners.

While you are doing this, insist on keeping the same budgets and schedules for your projects - because during any change, productivity usually drops, right?

And, when the new methodology does not bring any benefits, blame the methodology itself. After all, you had all the roles, artifacts, you even kept your original budgets and times, so it must have been a failure of the methodology itself.

## **Create many new roles with complex descriptions**

Any new methodology ought to have many roles, preferably absolutely different from the ones you currently have. And if it doesn't (which usually means the methodology is focused on productivity, the exact opposite of what you are trying to achieve), just create them. In agile / SCRUM, you can for example have Product Owners, Chief Product Owners, Program Product Owners, Business Domain Product Owners and so on.

When you have these roles, create really long and complex descriptions of responsibilities for each of them. While at it, make sure these responsibilities - and descriptions of course - overlap, so no one will really know if a certain responsibility belongs to one role or the other, including the people who fill the roles.

## **Insist on creating formal processes**

In order to have everything under control, and to make the most out of your new methodology, formalize it to the maximum extent possible.

Create new formal processes that cover every alternative of what you do and insist on adhering to them exactly, again in the order to

maximize the benefits of this new wonderful methodology.

## **Disrupt communication channels**



**Disrupt the communication by changing everything**

Don't rely on existing communication channels when communicating the change. Create entirely new communication channels. And make these theoretically marvelous but practically impossible to use by anyone, such as communicating on a wiki where nearly no-one has access or using some archaic form of communication that most of the people cannot use.

By this you not only dramatically reduce the flow of information about your change, you will also destroy existing communication network. If you are lucky, you will create many more saboteurs - some because they did not get all the information about planned change, others because you've destroyed their communication networks.

## **Add "social" aspect to all changes**

Do not focus only on the technical side of any change. Add some modern social aspects to it. These you want to promote as being an integral part of your change ensuring its success.

What exactly you do is up to your imagination. Perhaps you can introduce 15 minute transcendental meditation before every meeting in order to calm everybody's emotions and better focus on the agenda. Or you can plan some yoga lessons during work time for the people to get rid of stress.

The possibilities are endless here. Anyway, no matter what you introduce, make sure it gets enthusiastic support. And punish everybody who suggests that instead of these social things "maybe we should just get some work done".

## Use feedback in a clever way

### Create precise change plans



Plans are worthless - but planning is essential, Dwight D. Eisenhower. Just do the opposite

When planning any change, always create very detailed plans for communication and for the change itself. Insist on creating and perfecting the plan before you actually do anything. And when the plan is finished, review it - with any luck, significant portions of it will be out of date and you will have to re-plan.