

**Empowering
people
to be awesome.**



The Hackerspace BLUEPRINT

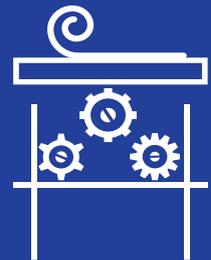


Table of Contents

- We are a Hackerspace 5
 - We learn from our failures 5
- 1. Overview 7
- 2. Do-ocracy 8
 - In short 8
 - Why not Democracy or Consensus? 8
 - How? 9
 - Noncoercive authority 9
 - Limitations 10
 - A do-ocracy is not a 10
 - Further reading 10
- 3. Meetings 11
 - What should I discuss on a meeting? 11
 - How do I schedule a meeting? 11
 - How do we make a decision during a meeting? 12
 - Why not regular consensus? 12
 - General Assembly and Statutes 13
 - The formal decision making model 13
 - Week before meeting 14
 - Meeting 1 14
 - Meeting 2 15
 - Meeting 3 15
 - Point system 15
- 4. The Board 18
 - Why is there a board? 18
 - What power does the board have? 19
 - When should I talk to the board? 19
 - Who should be in the board? 19
 - How does the board get elected and expelled? 20
- 5. Membership 21
 - How do I become a member? 21
 - What are the responsibilities of a member? 22
 - Non-members 22
- 6. Guidelines 23
 - Loopholes 23

Projects	23
Personal projects	24
Space projects	24
Property and tools	24
Personal property	24
Space property	24
Space maintenance	25
Cleaning	25
Exit space	25
Throwing things away	25
Social behavior	26
Noise	26
Network/security	26
7. The Legacy	27
Acknowledgements	28
Licenses	28

We are a Hackerspace

A hackerspace is a physical place, run by people interested in various aspects of constructive & creative hacking. From finding ways to make your beer cold in a matter of seconds to building a do-it-yourself sms-voting-system with an old android phone.

We are a breeding ground for awesome ideas. We provide a nest where those ideas can become reality. We operate by collaboration and openness. We allow people to fail, and to try again.

We learn from our failures

We created our very own hackerspace in Ghent. We had two rules: be excellent to each other and decide everything by consensus. We thought normal human interaction and common sense would solve all problems. Sadly, this was not true: internal conflict almost killed our hackerspace. Instead of giving up, we decided to “Hack the Hackerspace”; we organized a number of workshops to figure out what went wrong and how to build a better hackerspace. We found that the problems could all be traced to the following root causes:

- We cannot rely on common sense because **people have different realities**.
- **People have different, conflicting goals**. Because of that, consensus will never be reached on certain things. Problems will arise and they will not be solved.
- **Having no solution is worse than a bad solution**. Because when problems aren’t solved, they pile up until the community falls apart.

We knew that, in order to fix this, we needed a system that gets the best out of everyone and enables us to be awesome! During many late-night discussions, we fleshed out this system and wrote it down with the working title of “hack the hackerspace”. This eventually became “the hackerspace blueprint”, a document that describes how to run a hackerspace in a way that brings out the best in people. You can download **the latest PDF version of this document**¹. An epub version² is also available for e-readers.

We have been refining this system for many years, tweaking it when we encounter new issues and explaining important parts in more details. This document specifically describes how Hackerspace Gent runs, but it is generic enough that it can be easily adapted to other hackerspaces and other self-governing organizations. Feel free to use and remix this for your own benefit, learn from our mistakes and let us know what you think of it!

The goal of this system is to empower people to get the best out of themselves. It stimulates collaboration and enables people to think and solve problems creatively. We know that this system will be

¹<https://github.com/0x20/hackerspace-blueprint/releases/latest/download/hackerspace-blueprint.pdf>

²<https://github.com/0x20/hackerspace-blueprint/releases/latest/download/hackerspace-blueprint.epub>

flawed from the start. We know that control of people is evil, but a flawed system is better than no system. Moreover, we continuously update this system to make it better. That is why the hackerspace blueprint is on GitHub³. We want to make it easy for the community to contribute to it, and for other organizations to use it as a basis for their own system.

³<https://github.com/0x20/hackerspace-blueprint>

1. Overview

The main system running the Hackerspace is **the do-ocracy**, explained further in Section 2. In short: If you want something done in the hackerspace, either do it yourself or convince someone else to do it for you. The goal of the do-ocracy is to lower the barrier to contributing as much as possible.

However, when you want to do something that is irreversible or affects the core infrastructure of the space, you can discuss it with members on a **meeting** as explained in Section 3. These meetings can also be used to make sure that people will support what you want to do.

There are two issues that are hard to fix using do-ocracy and meetings: interpersonal conflict and core infrastructure maintenance. Section 4 explains how **the board** solves these issues as the “wardens” and “counselors” of the space. Unlike many other organizations, the board has no power over what the space should do.

Section 5 explains how you become a **member** of the space.

Every organization has a number of unspoken guidelines of how you should behave. We have written down some of ours in Section 6 in order to make it easier for people to get a sense of what to do in certain situations. These **guidelines** are not meant to be strict rules, but they give an overview of what is good and bad behavior in the space.

Because every good idea that was once written down has been misinterpreted, we include the information that helped us create this system and the guidelines. We call it **the legacy**. It is a collection of discussions, articles, research and books that we used to create the hackerspace blueprint. It gives more context about why the system is the way it is. The legacy should be used as a “cipher” to interpret the system and the guidelines better and to explain the rationale behind them.

2. Do-ocracy

In short

- If you want something done: **Just do it!**
- Have you done something? **Great!, now tell others about it.** Tell them what you did, and why you did it. A shared chat room or mailing list is a great place to do so. Telling other people about your actions lets them know who to thank and will give you more support.
- If somebody complains: Either **revert it**, or work out a solution with the person who is complaining.

Do-ocracy is an organizational structure in which individuals choose to pick up roles and execute tasks by themselves, rather than getting them appointed by others. Responsibilities and authority are attached to people who do the work, rather than to the elected/selected officials. **Doing a task is in itself justification for you being the person who does that job.**

Why not Democracy or Consensus?

Democracy and Consensus suffer from the same issue: everyone has an opinion about something, and most people like giving that opinion. In those systems, a lot of time and energy is put into debating what the best solution or the best compromise is. This results in a number of issues.

- It takes up a lot of time and energy that could be better spent actually doing stuff.
- It's very easy for problems to not get solved because the group doesn't agree on what the best solution is. This is a big issue because **a bad solution is often better than no solution.**
- It puts too much focus on the idea instead of its execution, even though research shows that the execution of a decision is often more important than the decision itself.
- Group decision making often leads to diluted outcomes, that contain elements of everyone's opinion but that nobody fully supports. As a result, people will be less enthusiastic about putting time and energy into actually doing the thing. This will eventually lead to worse outcomes because the impact of a well-executed good idea is a lot better than that of a badly-executed perfect idea.
- It rewards armchair critics and armchair activists. If the only thing you want to contribute is your opinion, you should not force other people to take that opinion into account.
- It encourages long feedback loops: because it takes a long time to make a decision, decisions need to be as perfect as possible. This is because it will take a long time to fix any issues. The fear of a bad decision will cause longer, more elaborate debate which increases the time it takes to make a decision. Modern software development practices such as Agile, Scrum and Kanban all stress the importance of having short feedback loops.

How?

A do-ocracy naturally emerges when the environment is right. There are a number of important factors.

- **Allow people to fail.** People need to feel safe knowing that they are allowed to try, and to fail. Thus, when people fail, we need to be kind and help them get better instead of berating them. The hackerspace gives everyone room to grow, and failure is part of that. For more information, read up on the idea of “blameless post-mortems” in the IT operations and DevOps communities.
- **Ask for help and help others.**
- **Trust each other.**
- **Focus on what you have in common instead of what you disagree on.**
- **Recognize and reward the people doing stuff.**

Noncoercive authority

“Coercive power is the ability to influence someone’s decision making by taking something away as punishment or threatening punishment if the person does not follow instructions.”

Coercive power plays a big role in modern society. People do what their boss tells them because they’re afraid of getting fired. People don’t steal because they are afraid of punishment. It is so ingrained into our society that it’s sometimes hard to realise that there are other ways to convince people to do something. However, you’ll get a lot better results when people are convinced of what they do.

It is a misconception that, in a do-ocracy, nobody is in charge or nobody has authority. The people doing stuff have authority over that project, although the power they have is non-coercive and they lose that power when they stop doing stuff.

A do-ocratic example: 30 people are going to Burning Man and camping together. Mary asks, “What if we organize a food pool so we can all cook and eat together?” Others answer, “Sure, I’d be a part of that,” or “I can make cake on Friday night.” Soon, Mary is calling campmates to borrow pots, pans and utensils, collating different people’s dietary restrictions, collecting money for food, and organizing trips to the store to buy supplies. At camp, she posts work signup sheets for cooking and cleanup, answers questions, and fills in when others can’t (or don’t) do their shifts.

A new campmate may grumble, “Jeez, why does Mary get to decide what everyone eats and when they work? Who put her in charge?” The answer is “the do-ocracy put her in charge”. The very act of organizing the food pool puts her in charge of the food pool. She can’t force you to eat the food or work a certain shift, but you have to respect her authority. This means that if you want to use the pots and pans for something different, or if you want to use the food money to order different food, you have to

ask her first. However, if she disappears in the middle of the camp, leaving the food pool in disarray, she loses that authority and anyone else can step in.

Limitations

Some things are too sensitive to be handled by do-ocracy alone, or are irreversible, like throwing things away. Refer to the Sections on the board, meetings and the guidelines for more information of when strict do-ocracy doesn't apply.

In general, if an action is irreversible, do-ocracy does not apply and you should discuss it with the larger group.

A do-ocracy is not a ...

- **Democracy** - In a democracy, everyone has a say in what gets done. In a do-ocracy, everyone does jobs that they think need to be done, **without everyone's input**.
- **Meritocracy** - In a meritocracy, the most qualified people for a job are selected for that job. In a do-ocracy, whoever does the job gets it, no matter how well they're qualified.

Further reading

- The CommunityWiki has a thorough explanation of do-ocracy⁴.
- Embassy SF talked their experience with do-ocracy in a shared house⁵.

⁴<http://www.communitywiki.org/en/DoOcracy>

⁵<https://medium.com/embassy-network/an-evolving-doocracy-3a6123f9b170>

3. Meetings

What should I discuss on a meeting?

Since the hackerspace is mainly a do-ocracy, there is actually very little that **needs** to be discussed on a meeting. Do you want something done? Well then, don't talk about it, just do it. Moreover, meetings only have limited power since, even if everyone in the meeting agrees that "A" is something great to do, it will only be done if someone actually does it afterwards. **Therefore, meetings should be primarily focused on convincing people to do something or asking feedback on what you want to do instead of on deciding what should be done.**

Things you should discuss on a meeting:

- Before using money of the hackerspace, discuss it on a meeting and make sure the board knows about it.
- When organizing big events in the space.
- If you want to do something that affects a lot of people in the space, and is hard to reverse when people complain.

Things you can discuss on a meeting:

- You want everyone to be on the same page about a topic or issue.
- If you want to make sure people will fully support what you want to do, you can hold a meeting to explain what you will do and ask for feedback.
- If you want to do something that requires help from many people, you can use a meeting to convince people to help you.
- If you have issues with the actions of a certain member and you failed to solve it with that member personally or you want the group's opinion on it.

How do I schedule a meeting?

Any member can schedule a meeting.

1. Create a pad for the meeting topics and the meeting notes. Use the pad of the previous meeting to see what it should look like.
2. Pick a date (preferably not during the social evening).
3. Announce the meeting in the Main channel and in the Changelog **at least a week in advance**. The announcement should include:
 - The url to the pad for that meeting

- The date of the meeting in ISO 8601 format (2021-11-14) and the time of the meeting in 24-hour format local time (20:00).

Important and possibly controversial topics such as membership applications need to be on the pad at least a week in advance. **If you add such a topic after the meeting is announced, you need to add a new entry to the Changelog.**

How do we make a decision during a meeting?

During meetings, the normal decision making model is “provided nobody says no”: you discuss a topic and propose an action during a meeting. You ask if anyone objects to that action. If nobody objects, you have the permission to do that thing. When a decision has been made, you need to write down the exact decision in the meeting notes.

Note that this intentionally favors the “yes” vote: there is a slight barrier to speak up and say “no”. The thinking behind this is that we want to make the barrier to “doing” as low as possible. We only want people to voice their opinions when they think it’s really important or when they are explicitly asked.

Why not regular consensus?

Consensus-based decision making⁶ aims to involve as many stakeholders as possible in a decision. This is the exact opposite from our system, where we want to involve as few stakeholders as possible in a decision, in order to lower the barrier to “doing” as much as possible. Do-ocracy gives as much power as possible to the person doing it, instead of to the people who have an opinion on it. When you want to do something, you have to make sure that nobody will hate it, instead of making sure that everyone is pleased.

Having as many people as possible involved in a discussion encourages Bikeshedding⁷: long useless discussion about trivial details that don’t really matter in the bigger picture. This idea stems from Parkinson’s Law of triviality⁸, which shows that you can easily get approval for building a multi-million dollar atomic power plant, but if you want to build a bike shed, you will be tangled up in endless discussions about the color of the shed. This is because a power plant is so complicated that people cannot grasp it, while anyone can build a bike shed over the weekend so everyone can comment on it. People enjoy commenting on it because they want to be part of the discussion and they want to add a touch and show personal contribution. Although for the people voicing their opinion this might be enjoyable, it easily kills the passion of the person who wants to get things done, and it slows everything down to a crawl.

⁶https://en.wikipedia.org/wiki/Consensus_decision-making

⁷<http://bikeshed.com/>

⁸https://en.wikipedia.org/wiki/Law_of_triviality

General Assembly and Statutes

A general assembly is different from a normal meeting in that it is governed by the rules in the official statutes of the legal entity “Hackerspace Gent (0x20)”. As a Belgian non-profit organization, Hackerspace Gent is required to host a General Assembly each year. During a general assembly, the board is elected.

The statutes dictate the following.

- The General Assembly needs to be announced at least 10 days beforehand.
- 50% of members need to be present during a General Assembly.
- 2/3 of members need to be present in order to change the statutes and/or the board.
- Decisions during the General assembly need a 2/3 majority of the present members.
- The board needs to have at least 3 members.

You can find the latest version of the statutes in the Government gazette (staatsblad) (Dutch-language).

1. Go to <http://www.ejustice.just.fgov.be/tsv/tsvn.htm>
2. Fill in 0x20 in the “Benaming” field and press the “Opzoeking” button. Now you should see a number next to the “opzoeking” button, this is how many search results there are. *0x20 is the short name for “Hackerspace Gent”, previously “Whitespace”.*
3. Press the “lijst” button to see all the search results.

More important information from the statutes.

- The hackerspace needs to have at least 4 members.
- New members need to be approved by the board.

The formal decision making model

Most decisions don’t require a rigid structure but we need a rigid structure to fall back on when there is extreme conflict that divides the space or when people don’t agree on how a decision is made. In such cases, a consensus-based system is used in order to re-unite the space.

In short; the topic needs to be put on the agenda three days before the first meeting. During the first meeting, a decision needs 100% consensus. If no decision is made, a second meeting is scheduled where a decision on a topic only requires 80% consensus, so a decision is made when 20% or less members object. If no decision is made, a third meeting is scheduled where a decision is made using the “point system”, an over-complicated system where a decision will always be made.

PLAN/TIME	ACTION	DECISION
Three or more days before the meeting	Put the topic on the agenda.	
Meeting 1	Discuss in group, listen, learn and build compromise	100% consensus
Meeting 2	Discuss in group, listen, learn and build compromise	80% consensus
Meeting 3	Discuss in group, listen, learn and build compromise	Point system

Week before meeting

The topic is put on the agenda of the meeting and is announced on the Mattermost (chat) server. This needs to be at least 3 days in advance.

Meeting 1

The topic is discussed in the group and requires a 100% consensus to reach a group decision. The motivation for striving for consensus is because it comes with characteristics that benefit the hackerspace:

- encourages discussion
- forces listening to opposing ideas that can give new insights
- can bring smarter compromises
- <https://en.wikipedia.org/wiki/Consensus>

The required 100% consensus also means that a very small minority can block a decision. That is a desired feature but it comes with a responsibility. When a small minority or even an individual feels very strongly that a proposed decision is not correct they have the option to block a decision. This does not stop a decision completely because only the first meeting requires a full consensus. This means that the opposers need to use the time between the first and the second meeting to convince their fellow members of their viewpoint.

Meeting 2

The topic is discussed again but now a rough consensus of 80% is required to reach a decision (https://en.wikipedia.org/wiki/Rough_consensus). If the small minority of last week was not able to convince enough fellow members the decision will be passed with rough consensus of 80%. When their viewpoint makes enough sense to fellow members, critical mass must be found to reach a new compromise. All members joining the discussion must strive to reach the rough consensus, to build the compromise.

Meeting 3

When all has failed, or the problem is too controversial, but a decision is still required, the point system below will be used to reach a decision.

Point system

The point system is a **last-resort** option. This should not be the general process of resolving conflicts. If the space is starting to use this too much, that means that there is a structural problem in the group dynamic.

The point system has a few basic rules:

- Each voter has a certain number of points that they can distribute over the different proposals.
- The proposal with the most points wins.
- In case of tie; re-vote.
- **Number of points per voter = (#_of_proposals * 2) + 1**
- Results should be given to the group in binary format: which proposal won and which lost. This is to strengthen the support of the decision.

“No decision” is worse than a “bad decision”. Conflict has to be solved eventually. That is why there is this last-resort option. However, we want to discourage people from blocking consensus. The point system has the following advantages:

- The outcome is not always clear because balanced ideas can still win, even if the minority would vote for them.
- The minority will gain from convincing the majority that their idea is not completely ridiculous.
- People have the ability to vote for, and thereby support, multiple ideas.

In the point system, every voter gets some points that they can distribute between the different options.

Examples

Vote without points	Points to proposal A	Points to proposal B
A	3	2
A	4	1
A	3	2
A	3	2
A	4	1
A	3	2
B	2	3
B	0	5
B	1	4
B	1	4
TOTAL	24	26

As you can see in this example, a less extreme proposal that, on first sight, has the minority of the votes, can still win. This gives the minority the incentive to come up with moderate ideas that everyone can agree with.

Vote without points	Points to A	Points to B
A	4	1
A	5	0
A	5	0
A	4	1
A	5	0
A	5	0
A	4	1
B	0	5
B	0	5
B	0	5

Vote without points	Points to A	Points to B
TOTAL	32	18

However, extreme ideas will not be able to “win”. With extreme ideas, the outcome of this model will be the same as with a +50% majority.

4. The Board

The board exists to make sure the hacker environment survives. The board members are not the leaders of the space, they are the firemen of the space. They make sure the physical space stays available and the members keep loving each other. Apart from that, the board members should be *indistinguishable* from normal members. The board members don't have any more say over the direction of the space or the projects in the space than any other member.

Specifically, the board has two roles, and for everything that doesn't fall into these roles, the board members are regarded as regular members.

1. *Warden of the physical core infrastructure of the space.* This stems from the infrastructure pattern⁹. Provide a room with power, internet, a bar and a kitchen¹⁰ and the hackers will come. An important aspect of this is keeping a good relationship with the surroundings as said in the landlord and neighborhood pattern¹¹.
2. *Counselor for the people in the space.* When conflict happens that can't be resolved in the group, the board is responsible for resolving the conflict. A great way to do this is the private talk pattern¹²: go talk to the involved parties in private, listen to them and let them know how the group feels.

It's important for the board to communicate openly about what they do.

Both jobs are critically important to the space. Many hackerspaces disbanded because they were kicked out by their landlord and many hackerspaces fell apart because of internal conflict. It is important to get the right people in the board. **This is why the board has very little power and only in exceptional circumstances: so it doesn't attract people who want to be in power.**

Why is there a board?

There are two reasons why Hackerspace Gent has a board:

- A Belgian non-profit organization (VZW) requires a board which is legally liable in case something goes wrong. In the past, we had a board on paper, but the space was completely run by consensus. This caused a bunch of issues since the board was legally liable, but it didn't actually have any power to prevent bad things from happening.
- People don't like conflict and confrontation. If nobody speaks up and nobody actively tries to resolve conflicts, people just ignore it until it explodes, taking down half the space with it.

⁹https://wiki.hackerspaces.org/The_Infrastructure_Pattern

¹⁰https://wiki.hackerspaces.org/The_Kitchen_Pattern

¹¹https://wiki.hackerspaces.org/The_Landlord_and_Neighbourhood_Pattern

¹²https://wiki.hackerspaces.org/The_Private_Talk_Pattern

Hackerspace Gent almost disbanded after such an explosion and we vowed to never have it again. Thus, the board is responsible for speaking up and fixing conflicts, even if that is really uncomfortable.

What power does the board have?

The board can temporarily ban people from the space if they think it's necessary for resolving conflict and protecting the space. If they do that, they need to let the members know who is banned and why. In the next meeting, the problem has to be discussed so the members can decide on a permanent solution.

When should I talk to the board?

When there's a fire, you call the fire department to come help you. When you want to light a huge 50 meter bonfire, you check with the fire department to see if they think it's safe.

The same applies to the board, talk to them when there is a big issue and check with them when you do something that's part of their role (warden and counselor).

Here are some examples of when you should check with the board before you do it.

- If you're spending space money.
- If you want to make changes to membership fees, the bar, the space shop, ...
- If you want to make changes to the electricity, and internet.
- If you want to make changes to the space building that affect fire safety, electrical safety, structural integrity or changes that would make the building collapse.

This isn't necessarily to get their approval, more to give them a chance to stop you if they think it's a bad idea. *Note: If something has been decided on a meeting with the board present, you can assume the board doesn't have any objections to it, and you can just do it.*

Moreover, if people are abusing the space, people in the space, or you, then it's best to inform the board. You're free to ask help from anyone you feel comfortable with, but it's best to also inform a board member so that they can intervene when someone harasses multiple people in private.

Who should be in the board?

The board does not have any say about what other members are to do, and you want people in the board that like/keep it that way.

- The “warden” role requires people who are responsible and dependable. The kind of people who say “maybe that’s not such a good idea, we might get thrown out if we do that”.
- The “counselor” role requires people who are open communicators, good listeners, and good at defusing a situation.

Both roles require people who are trusted by the members, are open to feedback, and who communicate openly about what they’re doing. Since a position with power is controversial (rightly so) in the hacker community, it’s incredibly important that the members trust the people in the board. The board will make difficult decisions and the members need to trust that these decisions are the right ones for the space, not just the right ones for the people in the board.

How does the board get elected and expelled?

During a general assembly, the members vote on electing and expelling the board according to the statutes. For more information, refer to the “Meetings” Section.

5. Membership

How do I become a member?

1. Fill in the membership application form¹³ and get two people to sign as sponsors. They will be responsible for making sure you get settled nicely, that you understand how the hackerspace works, and know how you can participate.
2. Announce your membership application on the Changelog. Include in the message who your sponsors are.
3. Attend a meeting and present your application. On this meeting you announce you want to become a member and who your sponsors are. If you want, you can talk a bit about why you're interested in the space. *Note: we won't discuss your membership further on this meeting and will not make any decision about it.*
4. On a next meeting, your membership will be decided upon. It's useful to have you present at that meeting, but this is not a requirement. As a bonus for coming to the meeting, membership is often the first thing to be decided on a meeting. When you get voted in, you can join the rest of the meeting as a member! Membership decisions at a meeting follow the same process as other meeting decisions. See Chapter 3 for more info.
5. Create an automatic bank transfer to transfer the membership fee each month. You will get your key after you made your first payment.

Why this process?

- Step #1 ensures that two people know and trust the applicant enough to vouch for them. It also ensures that the “culture” of the space gets transferred to new members.
- Step #2 ensures everyone knows you want to become a member.
- Step #3 ensures that the applicant understands how meetings are run and that there is some time between the applicant first arriving in the space and the applicant being voted in.
- Step #4 ensures that there is enough support for the applicant. The meeting decision model is used for the same reasons as to why it's used for meetings: a bad solution is better than no solution.
- Step #5 gives the new member an incentive to pay asap.

Note: If you want more context, see the `HTH_2018-11-17_membership.md` document in “the legacy” for more discussion on the membership procedure.

¹³https://wiki.hackerspace.gent/Membership_form

What are the responsibilities of a member?

The members create and maintain the hacker environment.

When a conflict/problem can not be resolved between individuals, via do-ocracy or when it impacts the group, a group decision is required. Any member can request that a decision is made on a meeting instead of by do-ocracy or individual members.

The members should do the following things as a group.

- Create and patch the hackerspace blueprint.
- Solve problems in the hackerspace when do-ocracy cannot fix them.
- Electing the board during a General Assembly.

The individuals have to be excellent.

- Organize workshops, events, lectures.
- Follow do-ocracy.
- Actively try to fix problems.
- Maintain personal safety and that of others.
- Follow and enforce the Guidelines.

Non-members

Non-members are also an important part of the space. They can contribute to the hacker environment and they can be potential members. However, non-members have less privileges than members.

- Non-members are only allowed in the space when they are in company of a member. That member is responsible for the actions of the non-member.
- Non-members have to follow the guidelines. A non-member is not allowed to challenge a decision made by the group. If the non-member disagree with a decision made by the group, then they should become a member and bring the topic forward on a meeting.

6. Guidelines

The hackerspace is a shared resource created *by* the community, *for* the community. It only exists because people think it's valuable enough to nourish it. Without the community, the hackerspace would simply not exist, so it is very important that we as a community collaborate and keep it fun for everyone.

We want you to be a part of it, but you need to do three things.

- Use common sense,
- be excellent to others,
- and don't be an asshole.

People have different realities, values and morals, resulting in different ideas for how to do these three things. To get around these differing realities use empathy, not cunning. Continuously convincing others to see things your way will get you what you want in the short run but can breed resentment in the long run. Going out of your way to understand and to accommodate the other person's point of view strengthens the community itself. The guidelines below describe what the hackerspace thinks it means to use common sense, be excellent to others, and not be an asshole.

First and foremost, the golden rule¹⁴: treat others the way you want to be treated.

Loopholes

The goal of the do-ocracy and these guidelines is to create an awesome community. The do-ocracy gives you a lot of freedom, but you need to use it wisely. People who try to exploit loopholes in this system to do things that damage the community cannot handle the freedom of a do-ocracy and will be pushed out very quickly. Do not read these guidelines like a lawbook, but read it like a cookbook. It doesn't matter if you use a bit more sugar than the recipe says, as long as your goal is to make the cake better.

Projects

There is a clear distinction between "personal" and "space" projects. Keep this in mind so you know what to expect and what people can expect of you.

¹⁴http://en.wikipedia.org/wiki/Golden_Rule

Personal projects

- You have full control over how to do the project.
- The project stuff/material is considered your personal property.
- You decide what happens to the end-result of the project.

Space projects

- The do-ocracy decides how to do the project.
- The project stuff/material is considered space property.
- The do-ocracy decides what happens to the end result of the project.

Property and tools

Personal property

- Only members are allowed to have personal property in the space. You get one box (the membership box) where you can leave your stuff. If you need more space for your projects, bring it up in a group meeting.
- If you break personal property of another member, you have to fully reimburse the member's losses.
- All personal property that is not in a members box has to be labeled (including tools and machines).

Space property

Using space property

- When you are using tools/infrastructure from the space, you are effectively borrowing that item from the community.
- Return borrowed items as soon as possible in the same or better condition than when borrowed.
- So when using something, clean it afterwards and put it back in its place.
- If you are not trained to use tool X, don't use tool X but ask an expert to teach you first.
- If you use one of the public workstations, please shut it off when you are done.
- If you use the printer, please deposit some cash to pay for consumables.

Damaging or losing space property If you damage or lose space property, you have to notice the community immediately in the Changelog or Main channel on mattermost. Explain what happened and if/how you will fix it.

Taking space property out of the space Only members are allowed to take space property out of the space. If you take space property out of the space, you have to notify the community immediately, for example in the Changelog or Main channel on Mattermost. If someone disagrees with you taking it out of the space, make that person happy or put it back.

Space maintenance

Cleaning

- Keep the dishes clean: when using the dishes, clean your dishes and any dishes that are standing there. When you see other people using the dishes, and they forget cleaning them, give them a gentle reminder.
- Keep the desks clean, feel free to use the desk space for your stuff, you can leave your stuff on the desk when you just 'pop out for some food', but leave a note stating when you'll be back. *Do Not* leave it there until the next morning.
- Remove empty packaging, from food or beverages.
- Every once in a while there will be a cleaning day in the space, as a good upstanding member of the community you should attend one of these at least once quarterly. Many hands make light work.

Exit space

- If you are the last person to leave the space, it's your responsibility to clean up. If you see people leaving, please alert them if they have left their trash in the space.
- Switch off all power consuming things
- Close the roof
- Read and follow the checklist at the door.

Throwing things away

- Some things that seem a useless waste of space to you might be very valuable to other people. When you throw things away, you have to let the community know and give them enough time to object.

Social behavior

- When in doubt if you're doing the right thing, you probably aren't.
- Share your love and passion, but respect people's boundaries.
- Don't hack people without their explicit consent.

If you have any issues with other people's behavior, please notify the board. The board will help if you want, but even if you don't; still notify them so they know of the problem.

Noise

People are trying to concentrate in here so:

- Mind your voice, volume. If you are talking to someone on the other side of the space, everyone in between can hear you; move closer.
- We know you like X music, but use headphones or keep the volume low.
- If you need to make a lot of noise, ask if you are not intruding/disturbing.
- Some moments are 'louder' than others, so it's not always easy to follow. Sometimes library/office-rules apply, sometimes workshop-rules and sometimes bar-rules. When in doubt, check with the other members.

Network/security

- Just leave other people's stuff alone, don't post "funny" social network status updates on unattended logged in computers.
- Don't sniff the network, no ssl-strip / rogue dhcp / random script-kiddo stuff. It's been done before. It's lame.
- Don't congest the network with (legal) torrenting, just behave nicely, so we don't have to write an AUP.

7. The Legacy

Because every good idea that was once written down has been misinterpreted, we included information that led us to the system and the guidelines in this repository as The Legacy. This should be used as a “cipher” to interpret the system and the guidelines correctly and to explain a bit of the rationale behind them.

Some of these documents, discussions and comments on them are archived in the Legacy folder of the hackerspace blueprint repository¹⁵.

- **The fall of the hacker groups**¹⁶ makes the claim that it is becoming rare for creativity to arise from groups or teams. Even though the technological advances should make it easier to create and maintain hacker groups, they are becoming increasingly rare. The author poses the theory that this is because the same technological revolution bombards us with a constant flow of information and fear, and because we dread the thought of being alike, of sharing multiple views and opinions. As such, we are turning progressively judgmental of who we should be partnering with, on the basis that “they do not understand”.
- **The tyranny of structurelessness**¹⁷ is an essay by American feminist Jo Freeman inspired by her experiences in a 1960s women’s liberation group that concerns power relations within radical feminist collectives. The essay looks back on the experiments of the feminist movement in creating organizations that do not have any structure or leadership. Jo Freeman states that leadership and structure did actually exist in these organizations but its existence was denied. This made it hard to hold the leadership accountable for their actions and made it hard for newcomers to figure out how the organization worked. As a solution, Freeman suggests formalizing the existing hierarchies in the group and subjecting them to democratic control.
- **Dealing with disrespect**¹⁸ by the legendary Ubuntu Linux community manager “Jono Bacon” explaining that “Most people are good people”. “This book is about helping us to focus on good people creating good things, to preserve that spirit of sharing, and to protect against those whose primary contribution is obstruction and disrespect.”

¹⁵<https://github.com/0x20/hackerspace-blueprint>

¹⁶<http://phrack.org/issues/69/6.html>

¹⁷<http://www.jofreeman.com/joreen/tyranny.htm>

¹⁸<http://dealingwithdisrespect.com/jonobacon-dealingwithdisrespect-1ed.pdf>

Acknowledgements

Licenses

Unless otherwise stated, this work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, see the LICENSE file in the repository, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Some of the cover art is Designed by macrovector / Freepik see LICENSE . freepik file in the repository for the full license.

**A guide to
hackerspaces
by
Hackerspace.Gent**

