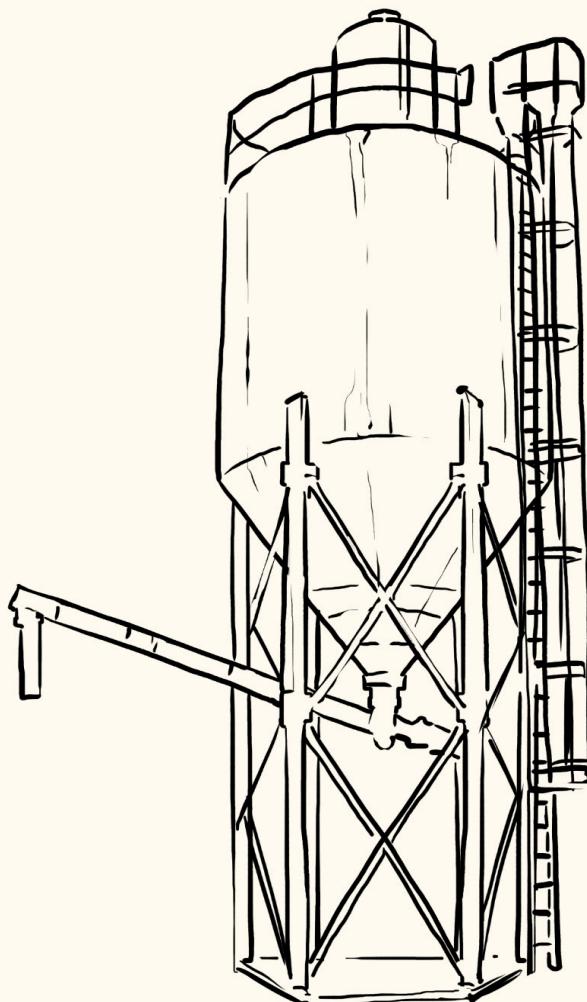


MAKING TOOLS FOR CREATIVE PEOPLE



CONTENTS

Summary / Résumé	2
0. Introduction	4
I. Which tools	9
II. Why we like making tools in the workplace	14
III. Tools for productivity only - the issues	18
IV. Hints for making better tools for creative people	24
Conclusion	36
Bibliography	38
Acknowledgments	39

SUMMARY

"We shape our tools, and thereafter our tools shape us."

This overused quote is close to the issue we tackle in this essay:
How to make better tools, in order to improve our
(professional) lives.

We start by pointing out that productivity and control are the main reasons we use and make tools in the workplace, even in creative processes, and try to list the pros and cons of this approach.

We then conclude that a change of paradigm is needed to design and make better tools, and finally we explore and open up potential and practical leads to shift our points of view.

RÉSUMÉ

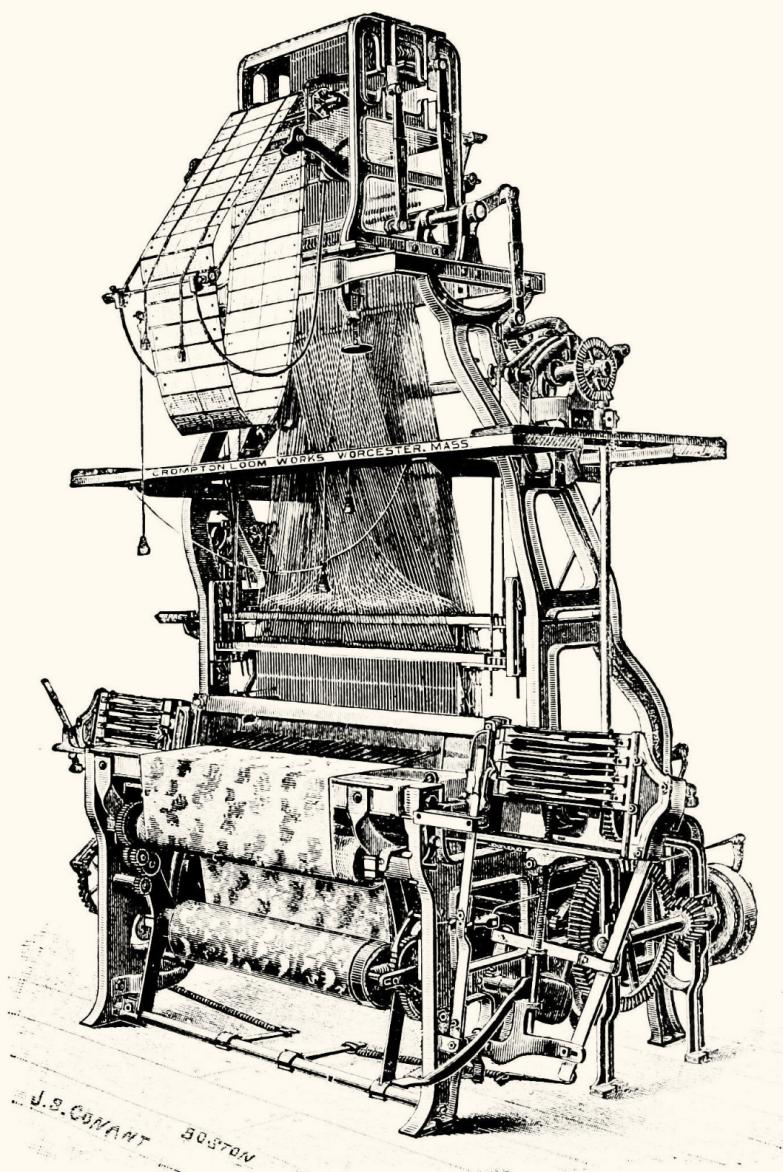
"Nous façonnons nos outils, et par la suite ils nous façonnent." Cette citation vue et revue cerne assez bien le problème abordé dans ce mémoire:

Comment faire de meilleurs outils, afin d'améliorer nos vies (professionnelles).

Je commence par montrer que la productivité et le besoin de contrôle sont les raisons principales qui nous poussent à utiliser et faire des outils en entreprise, même dans les procédés créatifs, puis je tente de dresser une liste des avantages et inconvénients de cette approche.

J'en conclu qu'un changement de paradigme est nécessaire afin de pouvoir penser et faire de meilleurs outils, puis j'explore des pistes potentielles et concrètes pour déplacer nos points de vue.

INTRODUCTION



Drawing of a power-loom for making carpets,
a device similar to the Jacquard loom

0. INTRODUCTION

WHERE MY INTEREST IN TOOLS CAME FROM

I love graphic design. That's why I stopped doing it as a professional and now do programming for a living instead. I went from print graphic designer and illustrator to interactive developer.

I hated doing the same tedious - and often nonsensical - tasks over and over again. In the print graphic design industry (advertisement and book publishing) I was faced with the lack of interest and understanding of my peers and bosses in the subjects of automation, tools and process enhancement. The processes were so established, I felt nobody wanted to change or even question them. I felt stuck.

That's why I started programming in the first place: making tools for myself.

Furthermore, what's the point of designing patterns or graphic systems if you can't describe them with the appropriate language, and always have to mimic/fake them with inappropriate tools/clunky software ? That was my philosophy at the time.

Now I do realize that I had a different understanding of the job than most people.

THE AUTOMATION STRUGGLE

Automation is the act of substituting partially or completely a human with a device in the accomplishment of a task. If the substitution is partial and a person "pilots" the execution of this task, the device can be referred as a tool.

Tools I am writing about in this essay involve a decent amount of automation. Here's an anecdote illustrating the automation struggle which started with the industrial revolution:

INTRODUCTION

"Designed in 1805, the Jacquard loom was capable of weaving incredibly complex and detailed patterns in a fraction of the time that a manual master weaver would take to create the same product. This revolutionized the manufacture of patterned textiles, allowing them to be produced at a fraction of the cost, and therefore making them available to a new market of middle-class consumers, rather than being reserved for the wealthiest in society.

However, the loom was not without its opponents when it was first introduced to industry. Master weavers in the early 1800s took many years to learn their trade, and many were angry at being replaced by a machine that could do the job more efficiently. As an act of protest, weavers began removing their shoes and throwing them into the looms, breaking the threads and rendering the looms temporarily useless. The shoes worn by workers in France at this time were called sabot, and this is where the word 'sabotage' comes from."

Quote from Weaving Numbers: The Jacquard Loom And Early Computing by Francesca Elliott (17 oct. 2017) from the Manchester Science & Industry Museum website

When I was doing print graphic design, trying to suggest automation and new processes, I heard this concern from my peers several times: "Isn't it a bad thing to automate for an artisan-like job such as graphic designer?" My answer to them today would be that:

Today, doing everything by hand in graphic design is mostly relevant for very small projects or in the haute-couture/luxe industry. Actually, designing a brand identity has always been designing a system:

The industrial revolution gave birth to movements like Arts & Crafts, Art Nouveau and Art Deco, where artists and designer were hardly distinguishable, ranging from painting to architecture, furniture

INTRODUCTION

design and typography. These were the beginning of new, truly transdisciplinary and multi-media approaches of arts and design.

So the newly born brand identities had to exist in a multitude of contexts. Thus the designers had to think about a **graphic system** so the brand could be recognizable.

More recently but before the internet era (in the early 1980s), people like Étienne Robial would think in systems and grids (legacy from the Bauhaus and Swiss design) to promote a generative brand identity. While using UHU glue sticks and color pencils to design it.

Today graphic designers still have to create for a variety of objects and situations, but most of all for screens with all types of sizes and interactivity which are handled by highly-visually-educated people.

And more broadly, these very tools and automation actually gave birth to new occupations and *crafts* like design. Today more than ever, creativity in the workplace cannot be dissociated from the technology related to it.

BEING CRITICAL ABOUT ONE'S JOB

One of the things I enjoy the most in my job as a programmer-designer is helping people and feeling smart while doing so.

"We focus so much on what we can do, that we forgot to ask ourselves what we should and shouldn't do. But in the end, the monsters that we unleash into the world will be named after us."

Quote from Mike Monteiro's talk [How Designers Destroyed the World](#)

Monteiro points out the responsibility of the designer and calls it "not a burden, but the privilege and the potential [...] to push humankind forward." That sure is some large-scoped mindset we're getting into right there.

Without being so sure about the impact on the whole world, we can be sure that the tools we build impact hugely the daily routines, and

INTRODUCTION

thus the lives, of all the people using it, and interacting with it, willingly or not, directly or not.

That is why I think it is necessary to keep a mindset that is as critical and open as possible when designing or working on a tool, even open to the eventuality that it is not needed (or even harmful).

THE PROBLEM I SEE

In enterprises where processes are challenged, automation and tooling are mostly perceived as a good thing. Nonetheless we tend to see these through the productivity scope only, even in the businesses involving creative processes. This essay tries to expand and analyze why it is so, and also to answer the question:

**How can we shift the way we think about tools
for creative jobs and create better ones?**

I. WHICH TOOLS FOR CREATIVE JOBS

I will focus on tools handled by people with a "creative job", that is to say people who use their knowledge, experience, sensibility, know-how, insight, judgement and tools to produce original content in their daily routine. I will assume most of them are working in a larger production organisation / workflow / pipeline.

My personal experiences with such creative jobs include mostly visual content creation:

- working as and with print and digital graphic designers (producing books, brand identities, advertisement visuals, websites interfaces, etc.),
- through collaborative work with journalists (producing photographs, videos, data visualizations, maps, explanatory infographics, etc.).

Hopefully most of the content in this essay relates to a larger definition of "creative jobs", but I intend not to write about contexts and situations too far from my personal experience.

DIGITAL TOOLS

Tools have always been part of the history of mankind. It is a word used in a variety of ways. An artisan can think about his utensils, a construction worker can think about his instruments, and a UX designer can think about ideation tools. People like Victor Papanek or Mike Monteiro believe design and by extension the designer should be a tool for the people. Ivan Illich, in his book Tools For Conviviality, considers tools in a very broad sense: he thinks that institutions and infrastructures are also tools.

WHICH TOOLS

I will keep in mind digital tools, used in a production process for content creation or diffusion. Hopefully some parts in this essay are applicable to other type of tools too.

Among these tools, I can think of different types, which are not always mutually exclusive:

SERVICES

These are the tools that release their users from having to do the task and take responsibility for it. The user asks for something to be done, in the most simple - or simplistic - way, frequently in exchange for some sort of value (money, personal info or attention) to the service supplier. The persons using it are referred to as "clients". The service only exists if the business behind it keeps on supplying it.

As we will develop on later, one issue with the philosophy behind these is that it could dispossess their users, in very insidious ways.

Examples: Notion, Squarespace, Webflow...

FRAMEWORKS

A framework can be defined as a set of tools, components and rules, which share the same goal(s) but are interdependent.

This can be a reassuring environment to work in for someone who is comfortable with it, because it gives a lot of power while still being opinionated on the "right" way to do things. Nonetheless some frameworks have a non-negligible amount of complexity, and are neither fast nor easy to handle properly.

Examples: Unity, Framer, Svelte, Bootstrap, design systems...

PLUGINS AND LIBRARIES

Plugins and libraries are additional functionalities you install on a pre-existing tool. They come in all shapes and sizes, but always rely on what they are installed on (one-way dependency).

WHICH TOOLS

Examples: Ease & Wizz, Grid Modeler, Lottie, Photoshop brush packs...

TOOLKITS

A toolkit is a collection of tools which do not rely on each other. They are gathered in the same place (the "kit"), and are intended for a specific type of user. Sometimes these tools can also be interoperable (use the same file type or standard).

Examples: LibreOffice, Adobe Suite, Google Workspace, the AFP's Toolkit (for journalists)...

OPEN OR CLOSED-SOURCE

Some tools' implementation are not accessible to their users, for security, profitability or other reasons, and thus are closed-source. A black box with an interface. They are often made by businesses who think that they can satisfy specific client needs and market their tool to target them.

Closed-source examples: Adobe Photoshop, Cinema 4D, most services...

An open-source tool, with proper documentation, gives the user the power to become a developer, and fix their own tool. It's easier for an open-source tool to develop a community, which in turn can enhance and extend the tool, promote it, help other users, etc. But open-source also comes with its issues.

Open-source examples: Observable, Blender, Krita, Inkscape, Godot...

WITH INTERFACES

All these tools include the concept of interface:

- visual interfaces, like GUIs (Graphic User Interfaces), which can be windows, buttons, knobs, text areas, and all kind of shapes

WHICH TOOLS

and animations that present information or feedback through passive or interactive behaviors.

- functional interfaces, like APIs (Application Programming Interfaces), which are the abstract "surface areas" or activable points exposed from a program to the developer, which in turn they can use in their own program.

NOT USING AI EXTENSIVELY

AI-generated content is coming for us, (and in some places it's already there) that is a sure thing. When we see the avocado armchair designs or all the style transfer algorithms, it sure is exciting and worrisome at the same time. But we are not really discussing AI or Machine Learning in this essay because:

- I have very little experience and knowledge in these fields,
- it is pretty hard to reverse-engineer this type of content/tools (partly for the above reason),
- and also because **I am interested in the design process, in the broader/human sense of the term**, which I believe is prior to AI-generated content. The thinking could be enlarged by including these as "tools" (after all Adobe is using trained models in a lot of their recent tools), but I chose to reduce my scope with the definition of the word "tool" depicted above in the introduction.

MADE OR MAINTAINED IN THE WORKPLACE

I would like to emphasize on tools specifically made in the workplace. There are already plenty of tools available for plenty of tasks and workflows. Most of the time businesses bother to make their own tools only when they have specific needs, and/or need more adjustability on their workflow, because it can very time and resource-consuming. Furthermore, to "just" modify an existing tool, one needs to understand it somewhat deeply. And of course getting

WHICH TOOLS

to know the existing tools can also be very-time consuming, hence the custom-made simpler ones.

Examples:

Tools for helping creative jobs with very specific workflows, most of the time connecting several coworkers with different skills, like:

- journalists working with graphic designers and developers to make data-visualizations,
- designers also working as integrators with developers and technical artists to make video games,
- graphic and UX designers working with developers to make applications.

II. WHY WE LIKE MAKING TOOLS IN THE WORKPLACE

As I wrote before, in businesses challenging their processes, automation and tooling is mostly perceived as a good thing. Here are some of the reasons why.

PRODUCTIVITY

Nowadays productivity is the first thing that comes to mind when talking about work. Even creative work.

SAVING TIME

Time is one of the most valuable things in the workplace, especially when it comes to a production pipeline. Compressing time into tools by automating repetitive tasks is a no-brainer for most business owners for this very reason.

EASING THE PROCESS

In a project, complexity can be distracting when you work on the bigger picture. We tend to make tools to hide complexity or to avoid it. Simplification can help focusing on problems that are not technical. It can also be necessary to switch between levels of abstraction in a project, and such tools can help to do that.

HANDLING LARGE AMOUNTS OF DATA/CONTENT

In today's world it is very important to be able to deal with a lot of data or content. In a lot of situations it's even necessary in order to be competitive or relevant. Tools and workflows are a great way to manage big pools of anything. Big-data is not really a space where one can be "creative", but it surely impacts our mindsets and decision-making processes. Also tools using machine-learning and trained AI do rely on these huge amounts of data.

WHY WE LIKE MAKING TOOLS IN THE WORKPLACE

BEING PROLIFIC

Creating more results than before is generally satisfying, especially in a creative process. Creating or generating a lot of content can be very useful for both iteration and demonstration.

For the creator, iteration is much easier if they just pick something from a (semi-)automatically generated list of results, rather than having to discard some of their carefully crafted works.

Similarly, demonstrating that a set of rules (as related to a brand identity for example) is working perfectly with a huge amount of examples coming off of said tool is a good way to sell a process and an idea. Also it's much easier to show-off or advertise oneself with huge amounts of content.

CONTROL

Some amount of control is necessary in the workplace. Tools can be a good way to assert control and check if everything is working as expected.

PREDICTABILITY

Most of the time creative jobs still have to answer some specific expectations. And it's tempting to see them as people who must be reframed, refocused or supervised. Tools can help with that, by removing some choices (and possibilities) from the workflow.

PROJECT MANAGEMENT

Creative work also has its share of project management. Whether dedicated persons do it, or the creative workers do it themselves, project management is well done when it's close and intimate with the process.

WHY WE LIKE MAKING TOOLS IN THE WORKPLACE

Project management tools could be used to split, describe or assign tasks, establish and base the workflow, coordinate workers, and keep logs. Some also use them for communication.

DISCARDING HUMAN ERRORS

When the content is prone to errors and its validity could be at stake, it is reassuring to delegate most of the data-manipulation to an automated process. Indeed if the algorithm is well tested, there should be no reason whatsoever one we should worry.

Actually there is a big **but** here. More on that in part III.

OVERALL: PRODUCTIVITY AND CONTROL

These are not the only reasons we like tools in the workplace, of course. But I think these are the main and most obvious ones to most workers and decision makers.

In the consumerist society we're still living in, production continues to be the main criterion in a lot of companies and businesses. We don't need control because we are control-freaks (not everyone is), but because we think it's a requirement for being as productive as possible.

WHY WE LIKE MAKING TOOLS IN THE WORKPLACE



Photograph of a big BUT sign

III. TOOLS FOR PRODUCTIVITY ONLY: THE ISSUES

Today, in a world where energy should be a big concern, it is cheaper to print a bunch of medium-sized circuit boards with different types of materials (extracted from the ground or synthesized), coatings, other treatments and small electronic components and ship them across the world than to print a bunch of books made of paper and ink. This makes little to no sense to me.

Productivity (and the growth associated with it) does not have a meaning on its own. Not anymore.

"The twenty-first century has an analogue: it's easier for most people to imagine the end of the planet than to imagine the end of capitalism. We need an intellectual state shift to accompany our new epoch."

Quote from A History of the World in Seven Cheap Things
by Raj Patel and Jason W. Moore

Capitalism is all about making profit. It tends to constantly exploit any resources (and any lives) wherever it's easiest. Making something easy to exploit is by turning it into a production line.

That means capitalism is extracting as much "value" as possible from anything it can include into a production line, oftentimes by pressing it like a fruit one would want to extract juice from, and then tossing it away when it's empty. It relies on consumerism.

It thrives on free and cheap labor. Free care labor is a big issue Patel and Moore reflected upon in their book, but here we are referring to cheap labor in very automated industries.

TOOLS FOR PRODUCTIVITY ONLY: THE ISSUES



Photographs of silos by Bernd & Hilla Becher

CONTROL BY ENCLOSURE: SILOING

THE FUNCTIONAL PARADIGM

It's in the air: we think about processes like boxes with inputs and outputs. It helps to think like this and it's reassuring when we care about controlling the process. In Functional Programming, we avoid direct interactions between entities (what Object Oriented Programming would do), because these are the main source of bugs. This is what functional programmers aim for: predictability.

TOOLS FOR PRODUCTIVITY ONLY: THE ISSUES

Write the purest functions possible, avoid side-effects and externalities, declare and label everything with explicit function names and push the necessary side-effects to the outer edges of the program.

The issue is that programming computers is not the same thing as managing people.

SOCIAL DIVIDE OR "DIS-SOCIATION"

Humans are social beings. They need to feel included in a group to be happy and bloom. Anything related to the concept of "individuation" actually. It's also a condition to be able to trust and fully engage.

Bernard Stiegler wrote about "dis-sociation" as the phenomenon where groups and individuals cannot assert themselves, create, or feel any satisfaction when they are unable to participate socially. To create silos is to hinder social activities and to hinder individuals and groups in the creation of their identities.

Also having a "corporate culture" does not mitigate any of the distancing between teams (it is not a social link).

THE ILLUSION OF CONTROL

It is sometimes OK to separate or put things aside to avoid unresolved (or unresolvable) issues, depending on what it is and how it is done. But it should not be done to create the **illusion** of control.

ERROR-PROOF

Computers are less likely to do mistakes than humans. They do precisely what we tell them to. The issue with this is that when there *are* errors (like edge-cases, bugs, erroneous data, "misuse" of the tool, unexpected side-effects...) they could be hidden and/or much more harmful.

TOOLS FOR PRODUCTIVITY ONLY: THE ISSUES

Using a computer to do a repetitive task can save you from some amount of human error, but you introduce more complexity and/or abstraction by doing so, and mistakes are plenty around these, they are just different.

Some externalities (or side-effects) are unpredictable. They can be beneficial or destructive. If that's the case your tool and workflow must be resilient and flexible enough to adapt. It's easier for humans to adapt than for complex pre-existing algorithms and infrastructures.

Also considering edge-cases as errors can be a slippery slope.

HUMAN-IN-THE-LOOP

Concepts like the **human-in-the-loop** can be insidious because they can give the impression that it puts the human as a responsible safeguard inside the computer but most of the time he's merely embodied as another computational step in the whole process ("in the loop"), a cog in the machine. Also it is not a guarantee to relegate a person to "monitor" a device accomplishing a task, even - especially - if that person used to do the task by hand.

SERVICES AS TOOLS

Opaque tools like services dispossess their users of their control over the process and create dependency/addiction issues: a service makes its users forget about the underlying processes, and thus makes them rely on the tool functioning properly to do the task at all.

Today they are unavoidable in some professional fields.

STANDARDIZED AND DETERIORATED CRAFT/WORK

Automating and adding tools to a creative process is not always enhancing it, it can also standardize it, in a bad way.

TOOLS FOR PRODUCTIVITY ONLY: THE ISSUES

LOSING VALUE

We have to be aware of the diminished quality, both perceived and real, caused by standardization.

When edge-cases are seen as errors which must be avoided or erased, this can be very harmful to the creativity and originality of the process and the result.

It can also be harmful to the job itself, when it dispossess its user from creative tasks. For example whenever I see a graphic "designer" doing only executant work, I die a little inside.

ALIENATING WORK (CLICK-WORK/MICRO-WORK AND THE LIKES)

Repetitive tasks are part of what tools and automation aim to diminish. But once complexity is hidden through standardization, it can also result in boringly simple and repetitive tasks.

Micro-workers must comply to the algorithm and interfaces they are faced with whereas QA departments find small quirks and give feedback to the designers and developers to fix it. Creative workers should not be considered low-skilled workers (nobody should).

Creative workers should be considered highly-skilled workers, and given tasks reflecting that.

FORGETTING KNOW-HOWS

If their work is simplified away and they are only given tasks below their potential or present skill-set, there are several risks: they might get bored, and they *will* lose some of their skills and know-hows.

It is in this way that "magic" tools (as in "closed-source") and services can be insidious. Once you have lost skills because of a lack of practice, it is hard to get back on tracks, and you can become dependant on specific tools.

LOSING MEANING

Standardizing processes and expectations related to creative tasks can subjectively remove meaning from them. A creative person who does not find meaning in their job anymore reminds dangerously of "bullshit jobs" as described by David Graeber.

"Hell is a collection of individuals who are spending the bulk of their time working on a task they don't like and are not especially good at. Say they were hired because they were excellent cabinet-makers, and then discover they are expected to spend a great deal of their time frying fish. [...]"

I would not presume to tell someone who is convinced they are making a meaningful contribution to the world that, really, they are not. But what about those people who are themselves convinced their jobs are meaningless?"

Quote from [On the Phenomenon of Bullshit Jobs: A Work Rant](#)
by David Graeber

CRIPPLED CREATIVITY

All these points could lead to some sort of despair or disenchantment in the workplace.

If a business relies on being highly productive to have any value, it is understandable that control is a very precious thing to have. But making the process a production line tends to standardize everything, and it can create a lack of originality, which is pretty bad for a creative process.

The exciting and original aspect of jobs like data journalist or graphic designer is what make them valuable. A bored data journalist might be boring, and won't pick anyone's interest. Like most jobs, these creative jobs are something done best when the people are engaged and willing to do it.

IV. HINTS FOR MAKING BETTER TOOLS FOR CREATIVE PEOPLE

There are several ways to shift our productivist tooling mindset. Some are closely related to the tools themselves, and some are more general ideas and intents that can be embodied through tools, among other things.

LOOKING FOR WORDS TO DESCRIBE NEW PARADIGMS

The language and words we use shape the way we think. Jargons from other professions can sometimes be very useful. Stealing words from other domains is not always a harmless act, but it does help to create new ideas and paradigms for designing tools. Here are a few words I came across.

TO DESCRIBE THE CREATIVE ACT:

These two words are examples of precise or niche concepts which can convey a special quality to the creative action, and enrich it:

- **Craft**

This one is starting to be more and more used by designers, even if some of them find it to be a stretch. The idea behind this word is to emphasize on the fact that designers are working carefully and meticulously, like craftsmen/artisans would do.

- **Praxis**

This word is used in various ways amongst different fields (political, educational, spiritual and medical for example). I encountered it many times in visual arts. It describes practice with intent, which is also self-reflecting. An artist builds up their own praxis as part of their artistic identity. It also suggests that practice of one's technical skills is associated with both sensibility and intellectual growth.

TO DESCRIBE TOOLS:

These two words from the philosophy field help to think about tools as abstract concepts.

- **Hypomnema (or Hypomnemata)**

"Understanding hypomnematas is understanding that memory (individual and social) is not only inside brains but also between them, inside artifacts."

Quote from Ars Industrialis' website (Translated from French)

An hypomnema (or hypomnemata) looks like a pretentious word but is quite useful. It refers to a technical (and man-made) medium for memory. For example writing, photography, or audio recording are all actions involving the creation of hypomnemas. They enable the externalization of memory, and therefore the storage or transmission of it, if one is able to read or decode it.

This concept is useful to better comprehend how the mental workload is "unloaded" and stored inside the tools or media we use.

An example of an hypomnema that can be a great source of social divide: ideograms. When someone reads an ideogram they don't know, they can't pronounce it. Thus separating strongly the highly educated from the others.

Such divides caused by a specific or obscure encoding, language or jargon, exist between and even within all professional fields. It can be purely accidental or residual (confusion or bad communication between coworkers is rarely intentional).

Also tools can be a form of hypomnema: by unloading an idea, a mental workload or an intent into a tool, it can either be transformed into something we want, be stored for later, shared and help others, and/or it can also dispossess us from it, or the processes revolving around it.

This means tools are double-edged swords, also called "pharmakons" by Ars Industrialis.

• **Pharmakon**

A word from ancient Greek meaning "remedy", "poison" and "scapegoat" at the same time. Any technological object can be apprehended as a pharmakon: it is both a remedy and a poison.

Another way to phrase it is "something that can care for us and be beneficial, and also something we must care for, so it is not harmful".

In his book *The Re-Enchantment of the World: The Value of Spirit Against Industrial Populism*, Bernard Stiegler explains how technologies, and by extension tools, are a common ground for most groups of the society where there are different struggles. For example he describes how letting the "society of control" take over tools may cause profound social divides, which could lead to the collapse of human values, desperate behaviors and dramatic events.

AIMING FOR FLEXIBLE WORKFLOWS

AVOID WATERFALL AND GO FOR ITERATION

Waterfall workflows are getting old. They lack the flexibility and adaptability that our fast-paced world requires. Plus they impose a very strict vertical hierarchy and unidirectionality in the workflow.

If your tool has to be complex because you are working on complex stuff, you should aim for a tool that can be in the center of your teams. Everyone should be able to contribute at the same time, as much as possible. With this it should be easier to enable iterability.

ENABLE SERENDIPITY

Productivity does have a creative value for experimentation. If you manage to enable iterability on your projects, you could also enable serendipity: finding good ideas or results unexpectedly, and re-injecting them in the project. Tools and workflows that facilitate experimentation are important for the creative process, and being prepared for happy accidents is very valuable.

ENABLE TRANSDISCIPLINARY DISCUSSIONS

A tool that is used by people with different skill-sets and roles is both a link and an interface between teams. It should ease communication between them. If people with different backgrounds and skills manage to use the same tool, in a relevant and complementary way, they can discuss and understand each other's jobs better. The idea is to include more people in the creative processes.

CARING FOR YOUR USERS

An opinion I stand by: it is the tool creator's job - and any decision maker - to actually care about their users, in order to have a positive impact on their daily lives and thus on society.

INCLUDE YOUR USERS IN THE DESIGN PROCESS

When creating or modifying tools, you may want to automate specific tasks. User-centered design intend to focus on the user. But you should not only consider them as the center of attention of the design process. They should also be included in it.

For example you can ask questions like: **Who** defines a task as repetitive, tedious or nonsensical ?

Several persons must agree on this, and the users must have a voice.

HINTS FOR MAKING BETTER TOOLS FOR CREATIVE PEOPLE

An seemingly unrelated example: I visited friends who have a pretty big kitchen, and I noticed they don't own a dish-washer. When I asked why they didn't get one (I was craving for it at home), one of them answered that it was a special moment for him. He could empty his mind and relax while doing the dishes, and did not think it was tedious at all. So he felt he would lose this precious moment they've had a dish-washer.

On the other hand, I still want one.

REASSERT THE VALUE OF CREATIVE WORKERS

Creative people have a huge value. They are skillful. You don't want your creative workers to be perceived or become low-skilled workers. You must care for them, and tools should be a very important thing to consider for that.

LET THEM EMBRACE COMPLEXITY

As we've seen through the deterioration of one's craft through standardization, simpler isn't always better. It does not mean you should overwhelm them with information, but they should be able to see and access complexity whenever necessary. If you value your creative workers, their tools should not take away the inherent complexity related to their work.

An other way to put it is: "Easy is boring" (Cas Holman).

"Through her company Heroes Will Rise, she designs and manufactures tools for the imagination. These materials are manipulable parts and pieces which inspire constructive play, imaginative forms, and cooperative interactions between people. [...] "What is it?" and "What does it do?" are answered in intuitive details, play prompts, and semiotic clues, making the instructions implicit in the tool/toy."

Quote from the About section of Cas Holmans' website



Picture of Rigamajig, a kit made for children to play and build together whatever they can think of

MAKING YOUR TOOL MORE ACCESSIBLE

SHORT FEEDBACK LOOPS

People use different mental models and have unique sensibilities. A visual thinker might need visual clues or interaction to completely understand a new idea.

If your tool gives instant (visual) feedback to what is being modified, these users will feel comfortable and have a better grasp of what they're actually doing. It's also a fast forward for learning.

HINTS FOR MAKING BETTER TOOLS FOR CREATIVE PEOPLE

In the field of programming, there are many projects that use visual feedback brilliantly, like Processing (very close to my heart), Scratch (aimed at children), Observable (for people working with data and visuals), and Svelte (addressing junior and senior programmers alike), for example.

(More on Observable at the end of this part.)

All these projects use a short feedback loop: you add a new line of code, and you instantaneously see the impact on your program. If you break something without knowing it, the feedback loop is so short that you instinctively know what modification it's related to.

The screenshot shows a web browser displaying the Svelte tutorial page. The URL is [https://svelte.dev/tutorial/transitions](#). The page title is "SVELTE". The main content area is titled "Transitions / Custom CSS transitions". It contains a snippet of text explaining that the `svelte/transition` module has built-in transitions, but it's easy to create custom ones. Below this is the source code for the `fade` transition:

```
function fade(node, { delay = 0, duration = 400 }) {
  const o = +getComputedStyle(node).opacity;

  return {
    delay,
    duration,
    css: t => `opacity: ${t * o}`
  };
}
```

The right side of the screen shows the Svelte REPL interface. A file named `App.svelte` is open, containing the following code:

```
<script>
import { fade } from 'svelte/transition';
import { elasticOut } from 'svelte/easing';

let visible = true;

function spin(node, { duration }) {
  return {
    duration,
    css: t => {
      const eased = elasticOut(t);

      return `
        transform: scale(${eased}) rotate(${eased * 1000}deg);
        color: hsl(
          ${--(t * 360)},
          ${Math.min(100, 1000 - 1000 * t)}%,

```

The REPL interface includes tabs for "Result", "JS output", and "CSS output". The "Result" tab is selected, showing the value `visible`. Below the code editor is a large orange text overlay that reads "transitions!".

Screenshot from the Svelte tutorial, which is easily accessible from the homepage of this javascript framework.

HINTS FOR MAKING BETTER TOOLS FOR CREATIVE PEOPLE

The Svelte interactive tutorials nail it pretty hard. They serve both as didactic explanations and interactive examples, using the tool directly in the browser, without having to install or configure anything: pure joy (instant gratification) for any user, and very beginner friendly.

Disclaimer: Having a very approachable and beginner-friendly side to your tool does not mean you should infantilize your users. I've seen people call graphic designers "creative" as a condescending and demeaning term, as a person not in touch with reality, unable to understand the complexity of the business issues, and thus their own job. Don't be that person.

I'm just saying that people in the educational and beginner-friendly fields are doing amazing work on the design of their tools and content, and we should be inspired by them.

HIGH QUALITY DOCUMENTATION AND TUTORIALS

Multiply the means to explain your tool. A documentation may be redundant with a highly intuitive interface or a well written codebase, but some users might need it (furthermore if there are hidden features for power users or advanced functionality).

If your tool is big and complex, don't be scared of redundancy and try to create a documentation, a list of written, interactive or video tutorials, and demos or examples. This will also help you with the next point.

BUILD A COMMUNITY AROUND YOUR TOOL

It's harder to build a community if you're working in closed-source, but the following still applies. Having a community around your tool is beneficial for its diffusion, and can enable mutual help between users. It is by no mean necessary, but can be a good auxiliary way to support social activities (connected to your business) too, which as seen above are important for creativity.

HINTS FOR MAKING BETTER TOOLS FOR CREATIVE PEOPLE

Also a community can help you enhance and maintain your tool, by having lots of feedbacks and/or contributions.

MAKE YOUR TOOL EASY TO FIX

This applies mostly to open-source: if you have a high quality documentation and a community, you should be able to make your tool (relatively) easy to fix for the users themselves. What differentiates a skilled user who understands their tool from an excellent one is the ability to fix the tool themselves.

Plus, everything breaks at some point. When you won't be able to maintain your tool properly, for any reason, your project can keep on living or be used partially by your most involved users.

THINKING MID TO LONG-TERM

In programming there is this concept of Technical Debt. It is just like real debt. It basically says that duct-taping is ok if it's very quickly replaced by a proper, long-term solution. If you iterate over duct-tape, you rapidly end-up with snowballing complexity and a big pile of nonsensical stuff.

Duct-tape design and programming is good for prototyping, but not so much for long-term tooling.

The main goal of an additional tool should be to improve the quality of life of its creative user, not just give them a new toy or gimmick that will bore them after a while.

A SMALL CASE STUDY: Observable

This may not be totally objective as Observable is not a tool made by an enterprise for its own business. But this is a very good example of what one can hope to achieve.

WHAT IT DOES

While in the browser, users can create and edit blocks of javascript, which are packaged to make them easily interact with each other (through reactivity). Each block displays the "result" of its content, visually if relevant (canvas), or with a log (text), which gives the user instant feedback on the different parts of their program. There are also different types of block for documenting the project or importing external content/code.

FOR DIFFERENT TYPES OF USERS

This tool, made by Mike Bostock (who is also responsible for the existence of the data-visualization D3.js library), succeeds in addressing a very diverse audience of professionals and hobbyists. Data journalists, graphic designers, programmers or just curious people can handle the tool in various ways:

- The most novice users can fiddle with existing projects made by others, available through the open and community-centered nature of the tool.
- Advanced users can easily document and comment their code at the same time as they implement it, and showcase their ideas or series of work.

Some surprising use-cases can exist too. For example some data journalists use this web-aimed tool to generate vector graphics (SVG) and import them in their favourite software, finishing with a light polish on texts and specific details. Then they are able to print it. From web to print!

FEEDBACK FROM A DATA JOURNALIST

The following feedbacks and observations, translated from French, come from data journalist Simon Malfatto, who works with both events breaking the news ("hot" content) and long-term data-bases ("cold" content).

Journalism is all about the news and information. These come first. Creation is only second to that, and constrained to that frame. But within it, there is indeed creative work and we can express ourselves.

Not being a programmer (beginner only, learning by doing), Observable gives me access to more creative freedom than other tools, because it is very adjustable. It also saves me tons of time.

It helps and urges me to learn, because other interesting projects are shared and can be edited, thanks to the community and open-source nature of the tool. It also sets me back on tracks for learning programming and libraries like D3.js.

Other, "simpler" tools (like Illustrator, Flourish, Row, etc.), which include turnkey solutions for creating infographics, are sometimes over-simplifying. For journalism, it can be a good thing, because you get a quick and easy result [which is very important for "hot" news]. But as soon as you want to do something more creative and stand-out, it's not the best option. Even in tools like Flourish where the customization is getting better and better, I still feel constrained.

COMPOSABLE OVER CONFIGURABLE

Observable is a tool that shows how the act of composition is tremendously enabling creativity. As stated brilliantly by Mike Bostock in the extensive article he wrote while making this tool:

"A medium to support discovery must be capable of expressing novel thought. Just as we don't use phrasal templates for composing the written word, we can't be limited to chart templates for visualization or a drop-down of formulas for statistical analysis. We need more than configuration. We need the composition of primitives into creations of our own design."

Quote from medium article [A Better Way to Code](#)
by Mike Bostock

...AND OTHER RELEVANT THINGS

These hints and examples are just the tip of an iceberg (the one I see, actually). If you're still reading I hope you will find interest in exploring in this direction which is basically away from the cynical means we are offered by residual paradigms.

CONCLUSION

CONCLUSION

Today, technology and tools are everywhere in our lives. In the workplace in particular, it is critical to understand deeply the tools we use, and be able to reverse-engineer or fix most of them. It should not be relegated to an obscure specialized team. In order to make better tools, we must elevate users and empower them.

A better understanding of tools and the dynamics around them can help decision makers and designers to find value in different things than pure productivity.

Different criteria must be pushed forward when designing tools. Quality of life, social well-being and long-term issues must be some of these.

To me, creativity is a very valuable thing, and it should not be a slave to productivist tools and pipelines, wherever possible. Productivity and growth are not strictly equal to value anymore. The world is changing and we need our paradigms, habits and tools to change too.

We must move productivity away from the center of our preoccupations, and put relevance instead. Those of us who can, must find and express new purposes, even - and especially - in the workplace.

BIBLIOGRAPHY

BOSTOCK Mike

A Better Way to Code - Medium post - 2018

CHAI VASARHELYI Elizabeth, Netflix

Abstract: The Art of Design - S.2 Ep.4:

Cas Holman: Design for Play - Documentary - 2019

ELLIOT Francesca

Weaving Numbers: The Jacquard Loom And Early Computing

- Article from the Manchester Science & Industry Museum website
scienceandindustrymuseum.org.uk - 2017

GRAEBER David

On the Phenomenon of Bullshit Jobs: A Work Rant - Article published
in STRIKE! Magazine - 2013

ILLICH Ivan

Tools For Conviviality - Book - 1973

LENHART Amanda and OWENS Kellie

Good Intentions, Bad Inventions: The Four Myths of Healthy Tech -
report from datasociety.net - 2020

MADSEN Rune

Programming design systems

- programmingdesignsystems.com - Book - still being written

MONTEIRO Mike

How Designers Destroyed the World - Talk - 2015

PATEL Raj and MOORE Jason W.

A History of the World in Seven Cheap Things - Book - 2017

REAS Casey

Form + Code - Book - 2012

STIEGLER Bernard & Ars Industrialis

Réenchanter le monde - la valeur esprit contre le populisme industriel

- Book - 2005

ACKNOWLEDGMENTS

Thank you all:

my tutor Thomas Parisot for helping me on choosing my subject,
Louis-Jean Teitelbaum for being an awesome educational coordinator,
Florine Fouquart (aka Flopine) for helping me on how to approach writing this essay,
Jacky Fong and Frédéric Bourgeais for putting up with me on a daily basis even when I ask so many questions,
Jules Bonnard and Simon Malfatto for their interviews and great insights on doing journalism with tools,
Ian Branston for the support and proofreading,
Louis Lagache and Mathieu Grondin for their listening and insights,
and my wife, who always put things into perspective for me.

Thank you for reading!

Please feel free to contact me, this whole thing is mainly a pretext to talk with people.

Mail: remyj.duval@gmail.com

Some social platforms: [@remydekor](https://www.twitter.com/@remydekor)

