

**The 2022
VFX
company**

LITE v0.01

Peter Timberlake

A paper about the best way to run a VFX company in 2022

Hi!

You're probably wondering why you're reading black-on-white text like this is middle school. It's because you're reading a 'paper' - a pseudo-academic one I'd call it, thus cementing my self-appointed title of 'the biggest nerd in Los Angeles'. This is a shortened paper based on a much longer paper I've been working on called *'The 2022 VFX Company'*.

This paper meant as sort of a pitch, a vision for how the VFX company of the future could operate (and perhaps should).

One other note - I've titled this paper *'The 2022 VFX Company'* with the specific intention that it will become outdated. I expect that certain principles in this document will be evergreen - certain things may be applicable forever. But if your company adopts the principles in this paper and is preaching them dogmatically in 2052, it will be immediately apparent that your company is doing things in an undesirably 'vintage' way. I am humble enough to admit that my 2022 ideas will become outdated - probably very soon. And your company should be humble enough to admit the same.

VFX companies' needs are always changing, and that bit of truth should be embedded in the title of this paper.

Let's get started!

intro

The reason I've written this paper is that I've read a lot of literature on entrepreneurship, management, business, logistics and supply chain management, design, sociology, etc, all with the goal of creating a better VFX company. What I've realized is that most of the available literature deals with product-industries, not service-industries. In other words most literature is about companies who meticulously design one product, then replicate that product exactly thousands of times. There's not much literature about 'service' industries - companies who design something different each for each and every client. I did learn a lot from this reading, but it took me a long time to realize that I need to be trying to solve different problems than these writers. I work in a service industry, not a product industry. Thus started my journey reading lots of other kinds of things besides the traditional 'MBA' wisdom.

The core problem we need to solve when growing and improving a VFX company is - we need to have structure and protocol, or our lives become a chaotic wreck of late nights and improperly named files, frustrated artists, and low quality work. On the other hand, we also need to have flexibility - we need to be able to respond to new client requests, new kinds of work, new schedules. What should be rigid, and what should be flexible?

I've worked at some commercials companies (large ones which you've heard of) that have almost no structure, and the nightmare is worse than I just described. I've also worked at some film companies which wouldn't be able to complete commercials work because they're much too rigid. I haven't worked at every company (not even close), but I've never seen anyone who nailed that blend of flexibility and rigidity. But I believe it is possible and I think I have a plan for achieving it.

What I ultimately stumbled upon is a handful of principles which come from the early programmers of Linux - the open source operating system. Namely the idea of modularity and 'bottom-up-problem solving'. Modules are small tasks which are really rigid, they have a clear in-and-out point. The modules work the same way every time. But they're small, can be mixed and matched, and arranged to fit the changing needs of a project. One principle of modularity is that workflows are not *engineered* to order (workflows are not engineered to create the same product every time), they're **configured**-to-order (configured to create a different outcome by assembling many small bits of workflow). We'll get into this more later.

Also - as a footnote - if we're being honest, I think I must be a little OCD or something because it drives absolutely fucking crazy when companies do things in stupid ways they can't defend, but stick by them unquestioningly.

So this is my crusade to build a model for a VFX company in the smartest way I could think of.

the modular staff

I think the VFX company of the future won't organize itself into roles and job titles. It will organize itself around modular 'tasks' - or 'units of work'. This isn't coming from an idealized hippie manifesto, it's a solution to a lot of very real logistics and supply chain problems which I've experienced directly at large, famous, VFX companies.

Modules would be small units of work with clear in and out points. They might be things like:

'create .mov transcode'

'Upload to sharable format'

'give feedback to outsource vendors'

'run dailies'

etc.

Each module would have an associated training video with the same title as that module. Those training videos would largely be made by the staff members who most frequently complete those tasks. All the training videos would be easily searchable on a 'wikipedia-like' database, so that if a staff member needs a quick 'on-the-spot' training, they can quickly receive training without disrupting anything.

Let's take a look at a problem I just encountered *literally yesterday* at the time of writing, how the rigid role/assembly-line style workflow failed, and how the modular system would have solved it.

The way most VFX companies make commercials is: data flows from department to department, assembly line style. The modeler passes their model to lighting, the lighter passes their render to compositing, the compositor renders and passes their render to editorial, and editorial passes the edit to data services to share with the client. This is top-down thinking.

Yesterday my team was trying to post for our client, and data services were unavailable due to a child's birthday party. Thus data was not able to flow between departments, the assembly line broke, rendering the entire multi-million-dollar VFX facility effectively worthless.

Let's take a look at how the modular system would have solved this.

The particular task which broke the top down system was that no one knew how to access Aspera. We're all college educated veterans in the VFX industry, perfectly capable of dragging and dropping a few files from a file browser onto an Aspera uploader. If we had been working within the modular system, any one of us could have learned the task (using the training video), and shuffled that module onto our task list. We could have *re-configured* the modules within the system accordingly, instead of being stuck with the broken part of the engineered system.

In the commercials business we need to be flexible for all kinds of reasons.

other solved problems - (like...seriously big problems)

training

Product-industries are better at training than service-based industries. I think I know why. When I worked at Starbucks, they had a week long training where they taught me everything I would need to know on a day-to-day basis. Then every day, I used pretty much everything I learned in the training - nothing more and nothing less. This is because every day, we made the same handful of different drinks, we never had to bake a custom cake, we never had to source specialty coffee beans at a customer's request. We provided a few mass produced products.

In the VFX industry, every project requires different skills, some of them may be 'must-know' but may only come up every once-in-awhile. Because of that, it's hard to justify a week long training period where staff learns all sorts of things they may never need. It could be a huge waste of time and money because it's hard to predict what training will be needed, because it's hard to predict what projects will need.

With the modular system, staff can learn the skills they need, only when they need them, not any sooner or any later. And they don't need to spend a ton of time learning things they may never need.

With this system, similarly to how we can configure our system for projects, each staff member can configure their training to their needs. We won't need any 'top-down' training programs designed by people who may not have any direct experience with more obscure roles. We won't need ~20+ individual programs for each unique role at the company, because each program will be configured on a per-need basis.

scalability

This training module system also has the ability to quickly onboard new people, without much one-on-one training. It ensures that new hires will be doing things in the ways we prefer as a company, and enables us to hire more people to take on bigger problems in the most frictionless way possible.

best practices

It also codifies the best, most well-researched way to do something. It ensures that freelancers won't do things in 'the way we did it at my old company'. (Of course if their way is better, we can adopt their way)

consistency

This system ensures more shared understanding between employees. It ensures handoffs are consistent every time, and between different people in different roles.

Implementation

I envision this module database - both structurally and culturally - as a Wikipedia of company practices. Wikipedia is easy to search, concise, and easy to read. It's also crowd-sourced and rigorous. But perhaps most importantly, it's fluid and competitive. Authors are continually looking for articles which might be inaccurate or incomplete, and writing or refining articles is a point of pride for Wikipedia authors. Wikipedia is constantly changing and improving, and that should be the model for our internal module database.

And finally, a bit of logic:

Buckle up - we're going to get into some logic which is core to how I'm thinking about this, and which will come up again later in the paper.

'Top down' problem solving generally asks the question: "I want to build a something, what problems will I need to solve in order to build that". Let's use the example: "I want to build a car - what problems do I need to solve in order to build a car?" Then the engineer, checklist-style, solves each problem they need to solve in order to build that car. This has its upsides. It means the engineer takes the least amount of time possible to solve the problem. It's direct and streamlined.

'Top down' is the way that most well-organized product-industries solve problems.

Now let's imagine a different task. Let's take 3 engineers, and ask them to build a car, a truck, and a van, all at the same time, and all with the guarantee that each vehicle will be unique - one of a kind. You may now recognize the metaphor - this is what we do in the VFX industry.

Engineers could use the 'top down' methodology, and start solving problems checklist-style, effectively starting from scratch for each and every car they build. They may be solving problems which are already being solved by other engineers, and they're likely solving them differently - maybe even slightly incorrectly due to being in a rush. And then after the product is finished, we need to start from scratch again.

The top down methodology is best if we need to design a product once, then reproduce the same product thousands of times. The top down methodology is so ubiquitous in our daily lives that we may not even recognize it as being a unique logical strategy, we may even just think of it as the 'way things are done.' But we're making new, unique things every time, all the time. We need to think about this problem differently.

Now let's imagine the 3 engineers built their car, truck, and van using a modular 'bottom up' approach. Instead of redesigning a steering wheel, a radio, a gas pedal, a rear-view-mirror, a seat belt - what if the engineers could use existing, pre-developed modules to avoid redesigning each and every part in their car, every time. The only thing the engineers would need to account for is the 'in-and-out' points of each piece. Each piece would need to be put together with standard screws and attachments.

The engineers would be 'configuring' to order, rather than 'engineering' to order. This is how I believe we should be structuring our VFX companies.

Will This Solve All of Our Problems?

No. In fact, this structure will - on its own - solve none of our problems. What it *will* do is store our problem solving for future use. That means that if we solve a particular problem on one project, theoretically that solution will be available to us forever, across all people in all departments. The rigid, 'top down' system is designed to work perfectly if there are no unexpected issues or client requests. After years in the commercial business, I find the naivety and hubris of that idea that 'everything will go as planned' pretty ridiculous, yet somehow still very pervasive.

More than a solution, the modular system is a framework which expects to encounter problems. It codifies and divides our work into small enough pieces that we can configure them differently when the inevitable problems do arise.

the modular pipeline

The Modular Pipeline is the sister to the modular staff approach, we'll get into their relationship more at the end of this section. Note - I've already begun very successful work on this and have received multiple requests from other companies to consult and implement it within their pipelines. Here's a link to a demo video:

https://www.youtube.com/watch?v=y9QFHWZTUJs&ab_channel=Tarantula

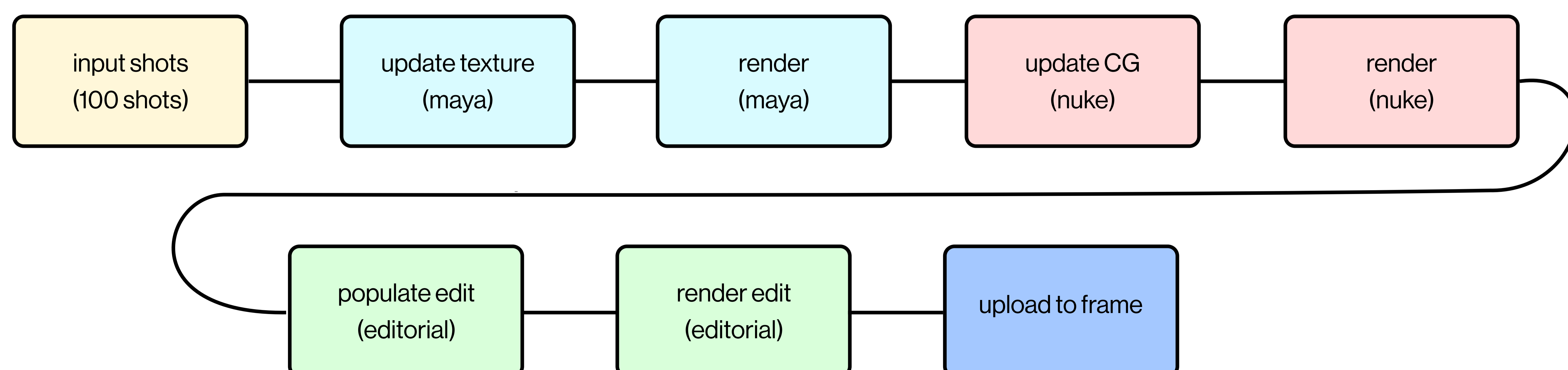
Here's how it works:

Similar to the modular staff, the modular pipeline divides pipeline tasks into modules so that they can be used by anyone in any department. These modules are represented by nodes in a node based interface which is accessible to all staff in all departments. Those nodes can be things like 'generate .mov', 'upload to frame.io', 'create denoise plate', 'update comp read node', or truly any pipeline process we can think of. On the following page is a image of how this interface looks.

I developed this modular approach to solve a particular problem - at most VFX facilities, code is rewritten and rewritten over and over for different projects or different softwares. It's only accessible to people who know how to use command line interfaces, or it's buried within long, difficult to search menus. In short, a lot of project time is wasted doing work which was already done for a different project, a different software, or a different department.

I developed 'Ripple' in order to ensure development could be saved forever across all departments, and be accessible to people without expert knowledge of softwares. But what I didn't realize is the ridiculous amount of power this opens up in terms of iteration.

For example, imagine we have a fully animated project with 100 shots. Our client asks us to update the lipstick color of a character. To achieve this, traditionally we'd need to open 100 maya scenes, update a material, hit render, open 100 nuke scenes, update the CG render, hit render. This sounds kind of trivial, but at most facilities, this would take an entire *week* of a multidisciplinary team doing tedious work. But what if we had a nodal interface to make these updates? Our node tree might look something like this - we could even render an edit and upload to frame.io if we wanted to:



We could do all of this work just by setting up the algorithm, and pushing the 'go' button.

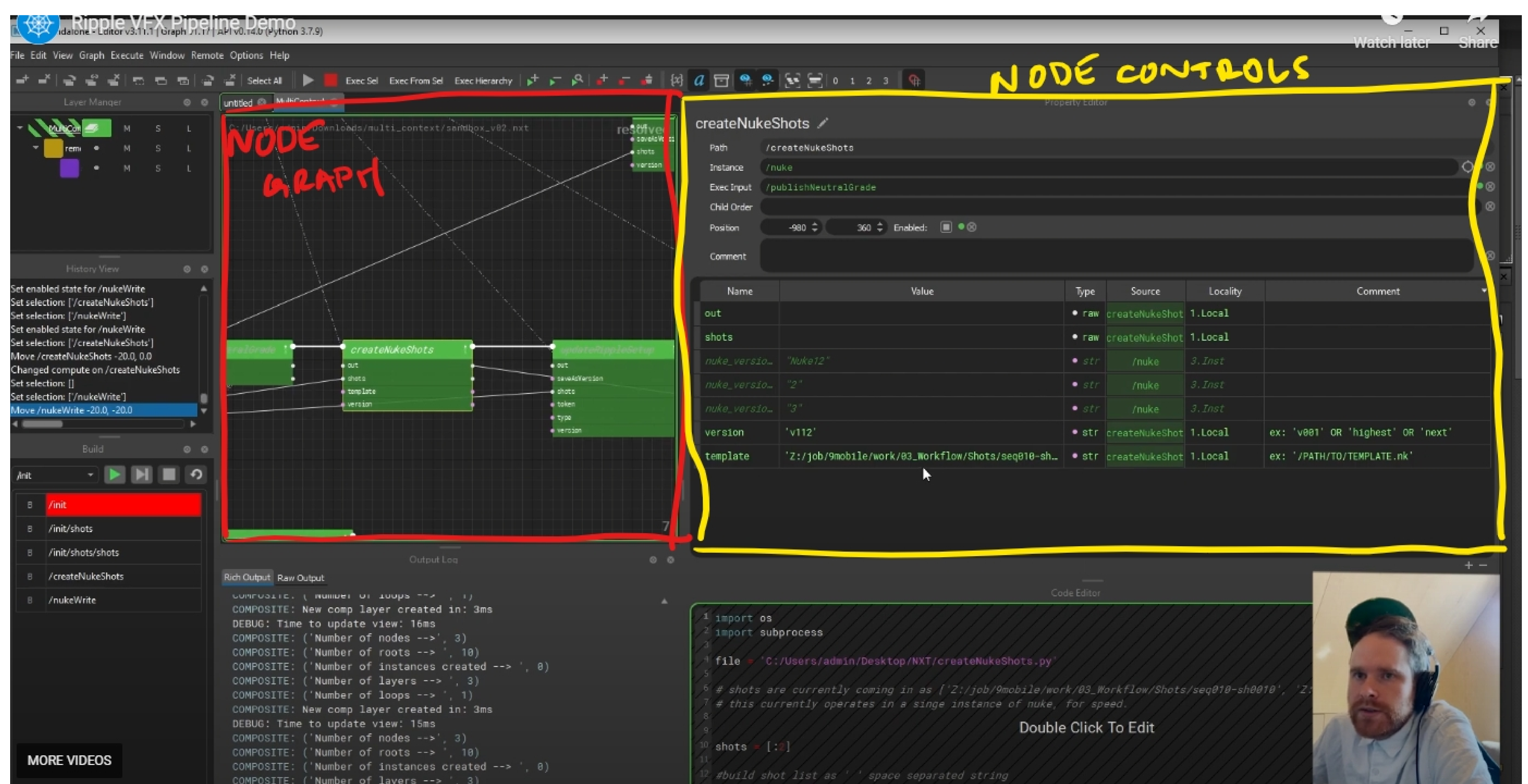
To refresh the 'top-down' vs 'bottom-up' terminology, most VFX pipelines have a 'top-down' approach. This means that a software architect studied an 'average project' at a VFX company, then designed a machine that would facilitate re-creating that 'average project'. We've already discussed the systemic shortfalls of the 'top-down' approach within service-industries, so let's not belabor the point too much - but let's look at a couple examples of problems the modular system would solve:

It's impossible to top-down design a system that is optimal for all projects or all roles. And worse, pipeline engineers often don't come from production backgrounds, so they tend to be really bad at designing software for the people who use it. This means that if an architect designs a huge system which takes years and thousands of dollars, artists may not even use it. If a portion of a pipeline breaks, it can be really time consuming, expensive, and catastrophic to fix.

This top down design approach usually results in staff coming up with their own ways to circumvent the difficult-to-use pieces of the software.

If we wanted to make a big change to our 'top down' pipeline, we need to start from scratch. In the modular system, we can replace little pieces at a time without interruption to the system. It is built with the understanding that pipelines are a never-ending project. They will always be expanding and contracting and improving, and we should develop a system which facilitates that - not one that inhibits that.

This puts the power of the pipeline into the hands of the people that use it. We can use it however makes most sense for the project. If a piece of the pipeline is missing, we can either create it ourselves (and save it for future projects), or ask a developer to create it without waiting for years for it to be implemented.



Codification: How Writing Software Forces Us To Find the Best Way

Writing software forces us to solve really complex problems - even complex *moral* problems. The most famous modern-day example of this principle is the "who should the self-driving-car hit" question. The thought experiment poses the question, "If a self driving car is forced with the option of striking an elderly businessman, a young homeless woman, or killing the driver, who should it kill?" . Philosophically, there isn't a clear answer. But self-driving cars have the potential to be much, much safer than human-piloted cars, so using self-driving cars is clearly on the horizon. And computer scientists will be forced to make a decision. Similarly, when we write code for our companies, we're forced to make a decision and find the best ways of performing processes.

When we write a process into software which was formerly completed by humans, it forces us to take some time to really think about what the best way of doing something is. It forces us into a state of mind that doesn't do what's most expedient, what's best for us, or what's easiest, but what's the truly most utilitarian methodology. And if something is hard-coded in a slightly wrong way, the *right* way becomes immediately clear to all of the users of that software. This is something we've encountered time-and-time again , both on a personal level, and a societal level. We feel blood-boiling frustration when a piece of software doesn't work how we think it should. And as a society, when a large tech company like Facebook hard-codes some devious policy, our moral 'compasses' can sense it right away. But in the context of VFX workflows - I believe that every small, well-intended codification is one step away from repeating boring work, and one step toward innovating new things.

Brother and Sister : The Relationship Between the Modular Staff and the Modular Pipeline

Modularizing our staff's work and our pipeline's work organized encourages us to ask the question, 'what if this staff module could be replaced with a software module. Many times, the answer is 'yes'.

Any time we can, we should be looking to take the weight off of staff, and place it onto machines. Computers are much faster, much cheaper, much more consistent, and much happier to do repetitive work, and our staff is much better than computers and much happier doing creative, innovative work. This is - of course - not to replace people's jobs. It's because if computers do our work for us, it enables our staff - whose time is much more valuable - to develop *new* ideas, create *new* relationships, learn *new* skills - not repeat the same boring tasks over and over. Repetitive, tedious work is demeaning to ask of our staff - that is work for computers.

Galapagos and Dunbar

In the full version of this paper, I outlined how large VFX companies can grow new talent and toolsets by breaking themselves down into smaller units internally. They could hit the 'reset' button on certain parts of their workflows and ask the small units to try and operate without those workflows in order to try and replace it with something more optimal. Effectively they could reset certain pieces of their company back to 'start-up' mode.

I cited a lot in this section - I cited the fact that small young companies often do incredible things with a small amount of resources because their starting place is using the latest, most optimized technologies, whereas the incumbent VFX companies are fully vested within old, bulky workflows. I cited Galapagos, where a certain amount of quarantine and forcing species to survive in an ever-changing environment has accounted for incredible amount of biodiversity and novelty. I cited the sociological principle of 'diffusion of responsibility' - the idea that in larger groups we're less likely to be proactive because we assume it's another person's responsibility. And Robin Dunbar and Noah Yuval Harari's research, which theorized that we - as humans - tend to thrive and feel most connected in groups between ~60- 150, and subgroups of around ~10-15, and have a tendency to get pushed out of larger groups.

In the case of a small company, I don't think this piece of the paper is especially applicable. But for big companies, I have a 'hypothesis' that this principle will be a game changer, and further extends the principle of modularity from tasks - to code - to the workforce.

Outsourcing and friction.

I believe that the future of the metropolitan VFX company will be a largely client facing and strategy oriented in function. The staff working in LA's job won't be to complete shots (most of the time), it will be to supervise, facilitate, and organize other people doing the VFX work in less expensive places. To do VFX well, it takes either an army of people - or a lot of a few people's time. Traditionally, the commercials approach has been to ask LA staff to work 80 hours a week, which is - from my point of view - no longer tenable.

I believe that the most successful VFX companies of the next decade will have that success by reducing the friction of outsourcing work. The metropolitan offices won't be where most work is done - that work will be done in places where it's less expensive to work. The metropolitan offices will be responsible for organizing that outsourcing work, guiding, facilitating, reducing miscommunications, setting up templates, reviewing, and of course attending to the direct requests of the client.

I think the only way to think about the outsourcing problem is with protocol and software. VFX is already complicated without international handoffs - adding that additional layer of cultural and data exchange needs a robust software system to support it. In fact - as much as we can - we should think about the software as the company in this regard. We should be able to onboard almost any overseas artists, and get a good work from them by designing our software in a way that supports them and adheres their work to our internal methodology.

As you probably know, I founded tarantula with this goal. Tarantula is designed to do exactly this. You likely know about tarantula, so I'll leave you with the website as a reference:

tarantula.la

I believe that the modular system will also unlock some important questions with the outsourcing process.

What is the goal of all of this?

The goal of all of this is to treat our staff and ourselves with dignity, make our staff feel supported, make sure we're solving problems only once, and give them the opportunity to create new ideas, work on new problems, and not spend too much of their life at work. But it also happens to be more profitable.

I've been saving this one last idea for the end - because I didn't want to lose you early. VFX companies have a long history of treating their employees terribly. VFX staff are not the same as on-set staff. Because of the infinite possibilities of VFX, VFX staff are - at many companies - asked to work indefinitely until a project delivers. We don't have unions. We're accustomed to 60hr weeks, and 80hr weeks are not uncommon. This is not acceptable on any level.

The California state government defines the reasonable time expectation for a worker at 40 hours/week. Do we really want to question the dignity of our workers by asking them to work double the amount of time defined by the State Government as 'reasonable'? The VFX company of the future will need to take this into account in order to retain workers - we're reaching a turning point in the industry where the best workers will not tolerate this, nor should they. I've personally seen companies already begin to implode because of this one issue.

Of course I am coming at this problem from a 'humanitarian' angle. But I'm also a project management zealot. Asking staff to work lots of overtime hours is extremely inefficient pay-wise, productivity wise, and burnout-wise.

4x staff @40hr/wk at \$50/hr = 120 work hours at \$8,000

2x staff @80hr/wk at \$50/hr (plus overtime) = 120 work hours at \$10,100

In both examples, our staff has worked 120hr/week. But in the overtime example - mentioning nothing of the questionable way we've just treated our employee, asking excessive overtime costs 26% more for the same number of hours worked, and those overworked employees start making careless mistakes, become resentful and damage morale for everyone, and start thinking about quitting.

There is a rich history in the VFX business of putting the responsibility of 'getting the project out the door' fully on the shoulders of heroic, tireless, virtuoso VFX artists. Of course we need great people to deliver projects. But the success or failure of our projects should never, ever depend on whether or not a single heroic person can survive with no sleep without collapsing. This is a failure of preserving our staff's dignity, and a failure of good project management. It sounds crazy, but this is a common way of working for some companies. To put it in the words of an Icelandic friend, if employees are doing unplanned overtime, it's the result of a project management failure.

It's a high goal, but I believe all of these problems can be solved with great protocols, great software, great workflows, and great project management. And I believe the path forward to create a great VFX company lies within the hypotheses I've just laid out.

You made it! Thanks for taking the time to read this. I hope you've seen the glimmer of a north star contained in this paper. Don't hesitate to reach out with hate mail.
peterhtimberlake@gmail.com

MUCH <3
- PETER