471029

Automated Resource Leveling and Scheduling at LAIKA

Michael Nowakowski
LAIKA, LLC

Learning Objectives

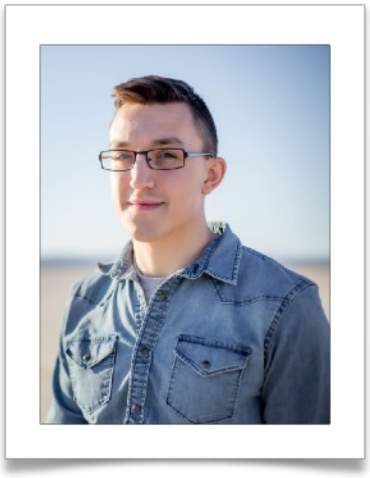
- Learn how to create a data model representing work for a project.
- Learn how to balance a schedule's resource usage using Shotgun's Generative Scheduling.
- Learn how to design effective visualizations of schedule data.

Description

To manage the ever-increasing complexity and collaboration in the creation of physical assets for LAIKA's hybrid stop-motion/CG features, an integrated browser workflow was developed that enables automated resource leveling and scheduling in a centralized system accessible to the entire studio.

Highlights include a new custom charting application that integrates with the existing Shotgun infrastructure, as well as the use of Shotgun's new Generative Scheduling, an automated resource-leveling system that effectively balances and assigns resources. Together, they turn a time-consuming and error-prone manual leveling process into a fast and automated procedure that generates an accurate, adjustable, easy-to-communicate schedule.

Speaker(s)



Michael Nowakowski is a Pipeline Technical Director at LAIKA. He focuses on studio workflows related to scheduling, fabrication, and asset organization.

On his Instagram profile, @michael__nowakowski, he chooses to describe himself as "an international multimedia art collective specializing in pictures of trees".

He now wears different (arguably more fashionable) glasses than he did in this headshot.

Introduction

LAIKA is a hybrid stop motion/ CG animation studio near Portland, OR. Fueled by the vision of our President & CEO Travis Knight, we create original feature films that push the boundaries of animated filmmaking. Our five releases so far are:

Missing Link (2019)

Kubo and the Two Strings (2016)

The BoxTrolls (2014)

ParaNorman (2012)

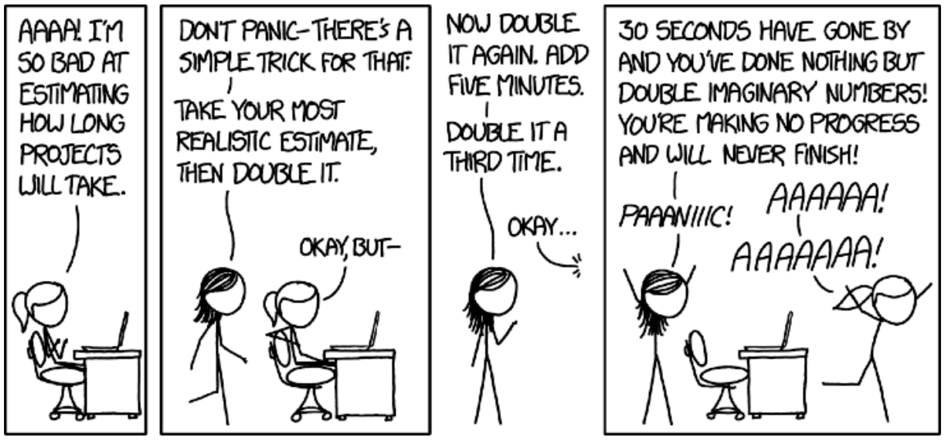
Coraline (2009)

In addition to critical and popular acclaim, every one of our films has been nominated for the Academy Award® for Outstanding Animated Feature. Our most recent film, *Missing Link*, also earned a Golden Globe for Best Animated Motion Picture.

Creating our films is a massive undertaking. Stopmotion animation requires the creation of thousands of sets, props, puppets, and costumes that must both express the creative language of the film and be able to be rigged and animated in painstaking frame-by-frame detail. Perhaps even more daunting is the challenge of coordinating work between our fabrication groups to ensure all our assets fit into the same world and are delivered to the appropriate stages in time for their moment in the spotlight.

Our Production Technology department has been an integral part of enabling LAIKA's success. For some insights into the work that this presentation builds on, please check out my colleague Tony Aiello's talk from 2018 — Shotgun for Production Management in LAIKA's Animated Features (Autodesk University 2018)

Now, on to the part where we talk about scheduling.


AAAA! I'M SO BAD AT ESTIMATING HOW LONG PROJECTS WILL TAKE.
DON'T PANIC-THERE'S A SIMPLE TRICK FOR THAT:
TAKE YOUR MOST REALISTIC ESTIMATE, THEN DOUBLE IT.
OKAY, BUT-
NOW DOUBLE IT AGAIN. ADD FIVE MINUTES.
DOUBLE IT A THIRD TIME.
OKAY...
30 SECONDS HAVE GONE BY AND YOU'VE DONE NOTHING BUT DOUBLE IMAGINARY NUMBERS! YOU'RE MAKING NO PROGRESS AND WILL NEVER FINISH!
PAAANIIIC!
AAAAAA!
AAAAAAA!

[Corollary to Hofstadter's Law: Every minute you spend thinking about Hofstadter's Law is a minute you're NOT WORKING AND WILL NEVER FINISH! PAAAAAANIIIIIC!] - xkcd.com/1658

LAIKA’s Scheduling Needs

A schedule is an essential part of enabling LAIKA’s intricate filmmaking machine.

1 - Get the right resources for the work

First, it helps us get the right resources for the production, and ensure we’re hiring an appropriate number of people with the right skills

2 - Prioritize Work

Second, it provides prioritization for our work. Knowing what is due when, and who should be working on what, lets us make a movie as efficiently as possible.

3 - Communicate Effectively

Third, the schedule is an important communication tool. As our films have grown in complexity and scale, communicating work within and between departments has become more of a challenge. An effective schedule can help departments (and shops within the departments) know how their work integrates with LAIKA as a whole.

What’s the ideal schedule?

1 - (Reasonably) Accurate, with Room to Accommodate the Unexpected

The schedule should contain all the required work for creating Assets, with a bit of wiggle room for when estimates prove to be incorrect.

2 - Encompasses All Important Information

The schedule needs to include all relevant information for decision-making. If important variables for making scheduling decisions exist only in certain people’s heads, or only in spreadsheets, the tools can’t be effective.

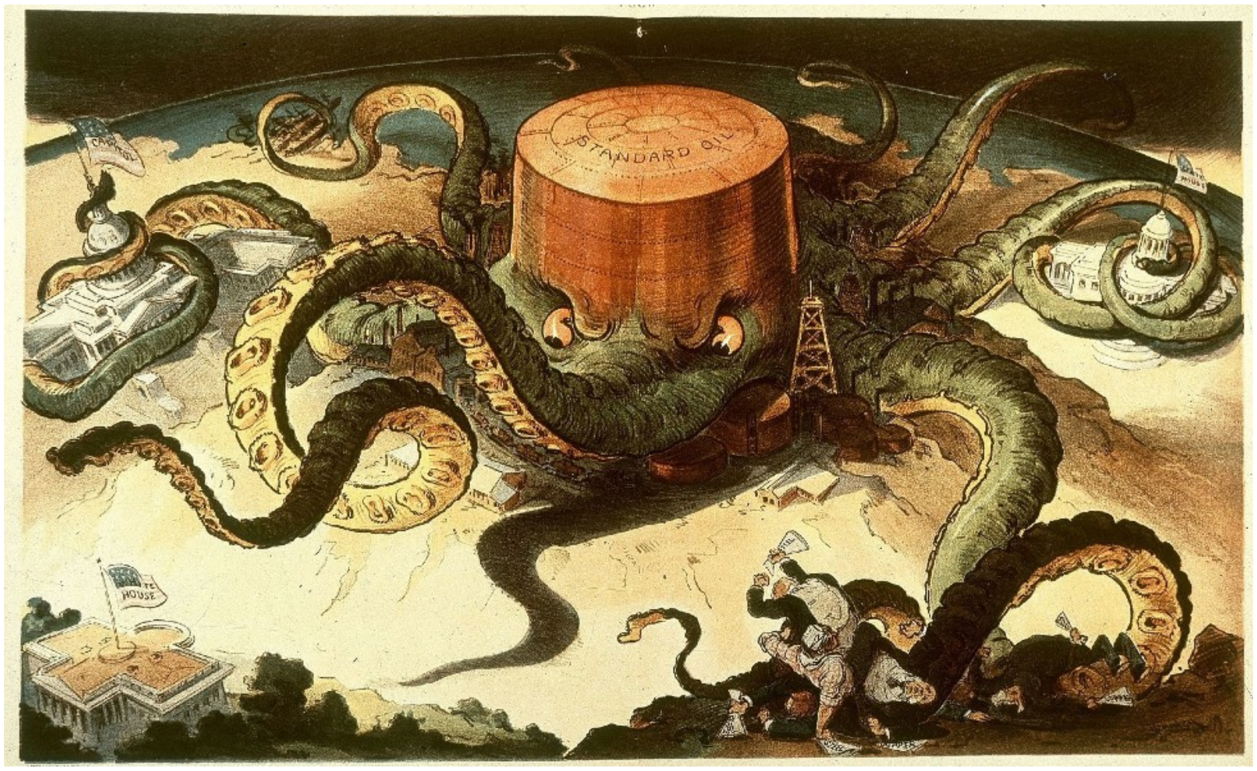
3 - Easy to Adjust

It should be easy to make adjustments to the schedule, both for recording actuals, and for making broader changes as processes change, or new creative choices alter build schedules.

4 - Easy to Communicate

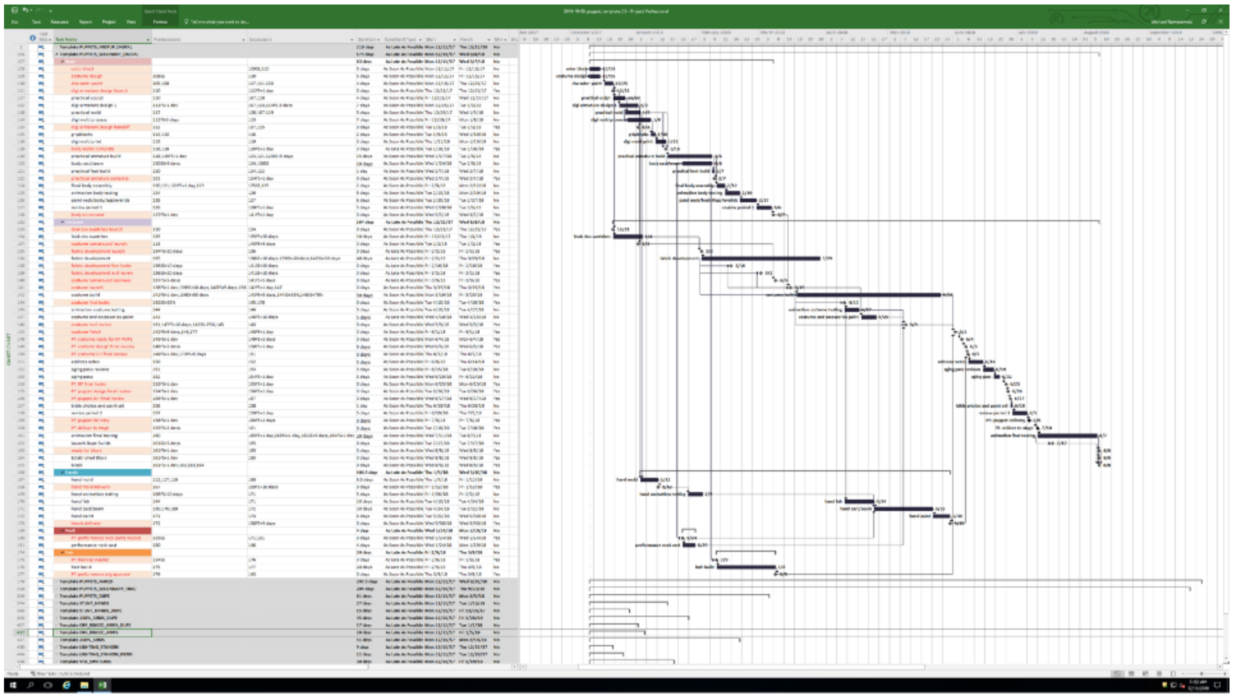
The storage format must make it easy for users to slice and dice the schedule to see just the parts they’re interested in, whether that’s a PM summarizing schedules for a company-wide overview, or an artist looking for just their Tasks.

In short, what we're aiming for is a much less sinister version of the Standard Oil Octopus — one centralized brain providing information to every department at the studio.



(but like… a friendly octopus?)

Previous Solution



Our previous scheduling solution utilized Microsoft Project to create task templates (lists of Tasks and relationships between them) that defined the work plan for different types of Assets.

For example, the first puppet we build for a character will typically include some research and development tasks, like creating costume patterns and defining fur and hair structures, so we'll create a "First Build" template for that work. After we complete the initial build, we often build duplicates of the puppet so that it can be used in multiple shots simultaneously.

For the duplicates, we don't need to include research and development tasks - we know what the puppet will look like and how to build it, so the list of tasks can be significantly shorter, and for that we make another template called "Dupe Build".

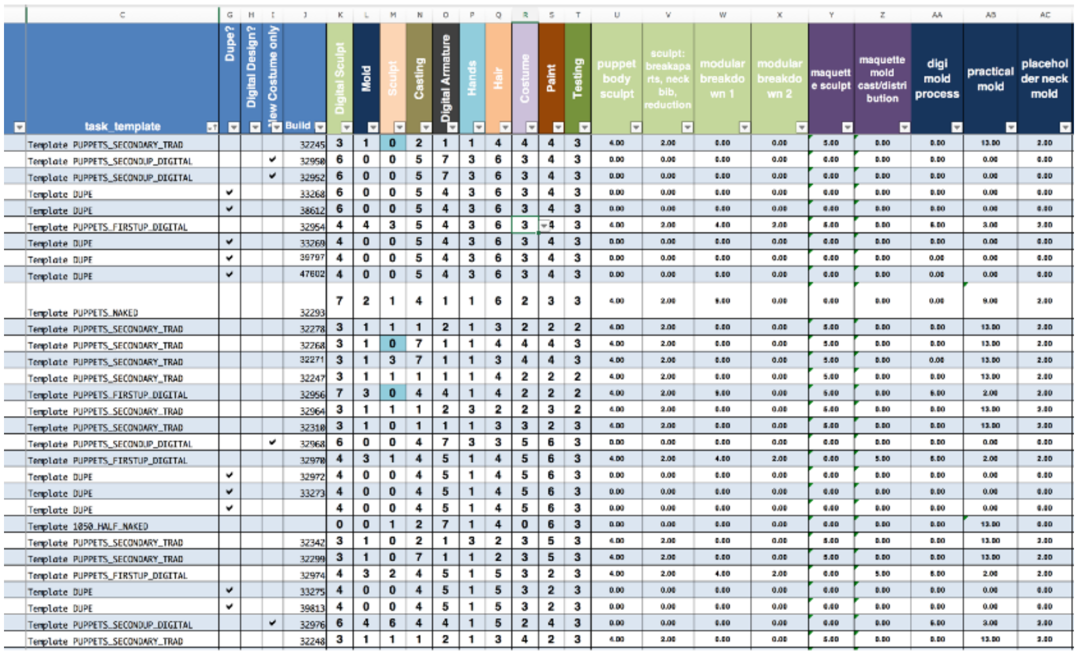
Once we've designed the different types of task templates required, we assign the task templates to builds.

For example, say we need to build 3 puppets of Sir Lionel, from Missing Link.

The first one we build will be assigned the "First Up Build" task template, and the other two are assigned the "Dupe Build" template.

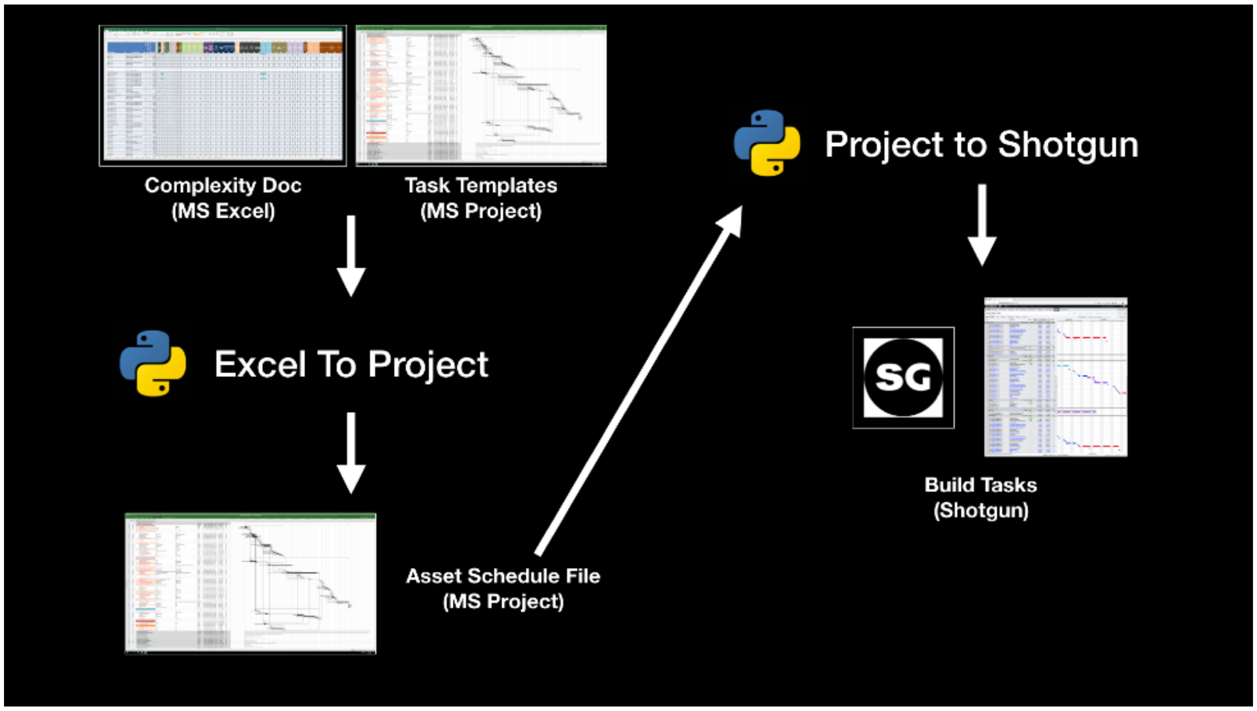
Sir Lionel (1) - First Up Build
Sir Lionel (2) - Dupe Build
Sir Lionel (3) - Dupe Build

In addition to varying task lists based on build type, we also required users to mark a "complexity", which defined the duration of each task in the task list. This extra flexibility gave users a systematized way to vary durations for Assets that should share the same task list, but might need more or less time to complete each task.



task_template	Dupe?	Digital Design?	lew Costume only	Build	Digital Sculpt	Mold	Sculpt	Casting	Digital Armature	Hands	Hair	Costume	Paint	Testing	puppet body sculpt	sculpt: breakaparts, neck bib, reduction	modular breakdown 1	modular breakdown 2	maquette sculpt	maquette mold cast/distribution	digi mold process	practical mold	placeholder neck mold
Template PUPPETS_SECONDARY_TRAD				32245	3	1	0	2	1	1	4	4	4	3	4.00	2.00	0.00	0.00	5.00	0.00	0.00	13.00	2.00
Template PUPPETS_SECONDUP_DIGITAL			✓	32958	6	0	0	5	7	3	6	3	4	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Template PUPPETS_SECONDUP_DIGITAL			✓	32952	6	0	0	5	7	3	6	3	4	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Template DUPE	✓			33268	6	0	0	5	4	3	6	3	4	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Template DUPE	✓			38612	6	0	0	5	4	3	6	3	4	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Template PUPPETS_FIRSTUP_DIGITAL				32954	4	4	3	5	4	3	6	3	4	3	4.00	2.00	4.00	2.00	6.00	0.00	6.00	3.00	3.00
Template DUPE	✓			33269	4	0	0	5	4	3	6	3	4	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Template DUPE	✓			39797	4	0	0	5	4	3	6	3	4	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Template DUPE	✓			47602	4	0	0	5	4	3	6	3	4	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Template PUPPETS_NAKED				32293	7	2	1	4	1	1	6	2	3	3	4.00	2.00	9.00	0.00	0.00	0.00	0.00	9.00	2.00
Template PUPPETS_SECONDARY_TRAD				32278	3	1	1	1	2	1	3	2	2	2	4.00	2.00	0.00	0.00	5.00	0.00	0.00	13.00	2.00
Template PUPPETS_SECONDARY_TRAD				32268	3	1	0	7	1	1	4	4	4	3	4.00	2.00	0.00	0.00	5.00	0.00	0.00	13.00	2.00
Template PUPPETS_SECONDARY_TRAD				32271	3	1	3	7	1	1	3	4	4	3	4.00	2.00	0.00	0.00	5.00	0.00	0.00	13.00	2.00
Template PUPPETS_SECONDARY_TRAD				32247	3	1	1	1	1	1	4	2	2	2	4.00	2.00	0.00	0.00	5.00	0.00	0.00	13.00	2.00
Template PUPPETS_FIRSTUP_DIGITAL				32956	7	3	0	4	4	1	4	2	2	2	4.00	2.00	9.00	0.00	5.00	0.00	6.00	2.00	2.00
Template PUPPETS_SECONDARY_TRAD				32964	3	1	1	1	2	3	2	2	3	2	4.00	2.00	0.00	0.00	5.00	0.00	0.00	13.00	2.00
Template PUPPETS_SECONDARY_TRAD				32310	3	1	0	1	1	1	3	3	2	3	4.00	2.00	0.00	0.00	5.00	0.00	0.00	13.00	2.00
Template PUPPETS_SECONDUP_DIGITAL			✓	32968	6	0	0	4	7	3	3	5	6	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Template PUPPETS_FIRSTUP_DIGITAL				32970	4	3	1	4	5	1	4	5	6	3	4.00	2.00	4.00	2.00	0.00	5.00	6.00	2.00	2.00
Template DUPE	✓			32972	4	0	0	4	5	1	4	5	6	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Template DUPE	✓			33273	4	0	0	4	5	1	4	5	6	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Template DUPE	✓				4	0	0	4	5	1	4	5	6	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Template 1050_HALF_NAKED					0	0	1	2	7	1	4	0	6	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	13.00	0.00
Template PUPPETS_SECONDARY_TRAD				32342	3	1	0	2	1	3	2	3	5	3	4.00	2.00	0.00	0.00	5.00	0.00	0.00	13.00	2.00
Template PUPPETS_SECONDARY_TRAD				32299	3	1	0	7	1	1	2	3	5	3	4.00	2.00	0.00	0.00	5.00	0.00	0.00	13.00	2.00
Template PUPPETS_FIRSTUP_DIGITAL				32974	4	3	2	4	5	1	5	3	2	3	4.00	2.00	4.00	2.00	0.00	5.00	6.00	2.00	2.00
Template DUPE	✓			33275	4	0	0	4	5	1	5	3	2	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Template DUPE	✓			39813	4	0	0	4	5	1	5	3	2	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Template PUPPETS_SECONDUP_DIGITAL			✓	32976	6	4	6	4	4	1	5	2	4	3	0.00	0.00	0.00	0.00	0.00	0.00	6.00	3.00	3.00
Template PUPPETS_SECONDARY_TRAD				32248	3	1	1	1	2	1	3	4	2	3	4.00	2.00	0.00	0.00	5.00	0.00	0.00	13.00	2.00

These
templates were then combined with custom per-asset durations from an Excel spreadsheet, put back into Project to be auto-scheduled, then finally exported via a custom app to Shotgun, a web-based project management database.



Bid Workflow - in steps:

- *Determine asset quantities and delivery dates*
- *Make MS Project Task Templates*
- *Make a "Complexity Doc" in Excel*
- *Bid Assets in Excel*
- *Combine Task Templates and Excel bids, import into MS Project*
- *Open MS Project, run the auto-scheduling*
- *Export from MS Project => Shotgun*

There were some major limitations to this process.

1. Cumbersome Roundtrip
First, the roundtrip process was cumbersome. Initial bidding often took weeks, and data in Project and Shotgun could easily get out of sync when doing updates.

2. Limited Modeling Capabilities
Shotgun's auto-scheduling features were limited, with no support for the complex dependencies needed to accurately model a detailed production schedule (significant strides have been made in the latest releases of Shotgun, so keeping all the dependency info no longer requires a custom table).

3. Visualization
Shotgun also didn't have sufficient resource visualization tools - users struggled to see what resources were overbooked, and what wouldn't deliver on time.

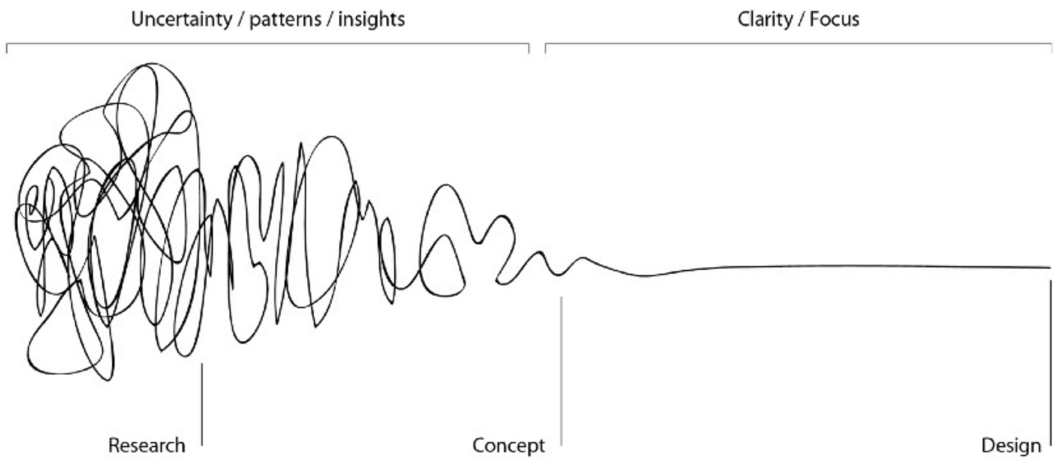
4. Manual Resource Leveling
The resource leveling process was entirely manual,
Hand-leveling such an enormous dataset lies somewhere between laborious and impossible; every push or pull in the order of tasks to benefit one group tends to come at the expense of every other group, with few easy solutions.

Perhaps even worse, the moment a hand-adjusted schedule was completed, anything from daily changes in the shooting schedule to R&D taking longer than expected to story changes could make it obsolete.

5. Actuals
All of the above issues lead to situations where no one trusted the schedule from Shotgun, so no one maintained it, which meant the data is never right and we have no useful data to use for forecasting future shows.

Without effective tools to evaluate and adjust the schedule to accommodate changes, craftspeople ended up overworked or assets delivered late, which slowed the pace of shooting and made it harder for downstream departments, like rigging, camera, animation, and visual effects, to meet their deadlines.

Design and Development



(The Process of Design Squiggle by Damien Newman, thedesignsquiggle.com)

The goal of this project was to develop a new scheduling system that would work studio-wide and enable faster iteration, as well as better resource visualization and leveling.

The first part of our process was developing data modeling tools so we could capture all the information our users needed to build an effective schedule.

Data Storage - Shotgun As Backend

We were already heavily invested in tracking, reporting, and distributing scheduling information with Autodesk's Shotgun, so we decided to follow a progressive enhancement strategy, building new features and workflows on top of our existing database and UI.

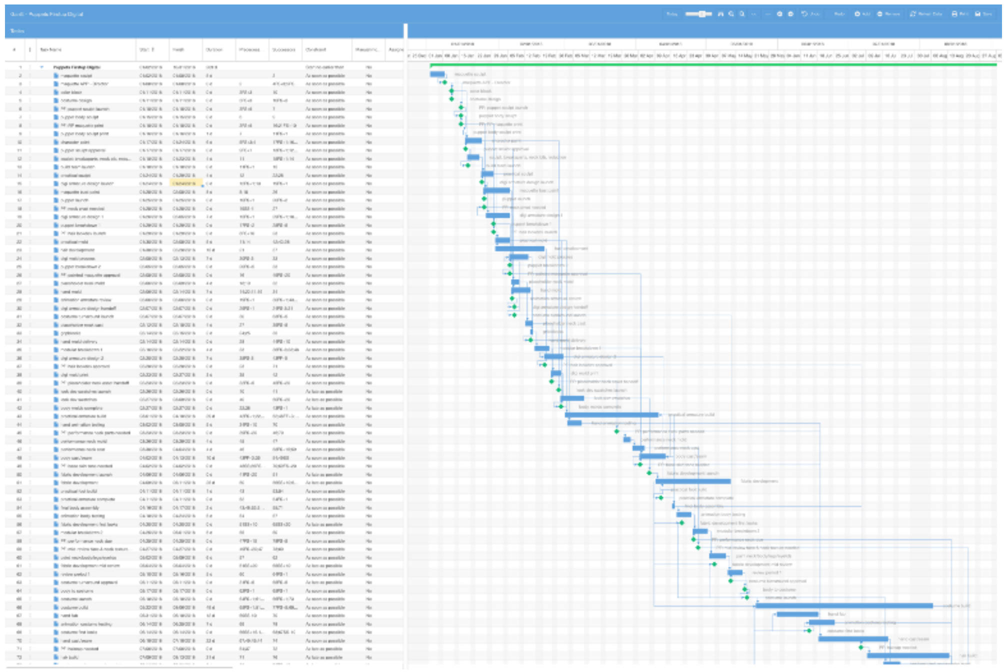
This had several advantages:

1 - Existing Tracking / Reporting Tools Still Worked

2 - Filtering / statuses / assignments / adjustments can be performed in familiar UI

3 - Minimized Scope Creep

We wanted users who were familiar with the Shotgun UI to still be able to use it when it was effective, then switch to our custom tools when they needed extra features like auto-scheduling and leveling. This also minimized scope creep - we weren't trying to reinvent Shotgun, just improve its scheduling features.



Gantt App

The first step was building an app that would enable users to capture all the complexities of a production schedule.

Shotgun’s built-in data models weren't sufficient for this, so we created a new database schema on top of Shotgun's existing scheduling infrastructure, then customized a third-party Gantt component from Bryntum to be able to edit, auto-schedule, and save task information back to the new schema.

With the new Gantt, users can select one or more entities and launch into a custom schedule editor, with just the feature set required for editing and syncing tasks directly to Shotgun.

All the standard project dependency types (FS, FF, SS, SF) are available, as are typical scheduling constraints and offsets.

To ensure correct auto-scheduling without giant data payloads, we parse the dependency trees for the selected entities at runtime, and load just the data required to correctly schedule the user's selection.

With the gantt component and custom schema in place, users can edit their templates and production schedules in the same application, with full access to auto-scheduling features like dependencies and constraints.

Even better, this data is synced directly with Shotgun, so all of our existing infrastructure for task assignments, navigation, and reporting is still usable.

Custom Data Model

* Indicates a non-Shotgun-native field / table

Tasks
A unit of work.

Key Fields:
- Start Date
- End Date
- Duration - duration of task in calendar working days
- Headcount* - utilization of a resource assigned to this task (ex. 0.5 = 50% utilization)
- Working Days* - total resource utilization (Headcount * duration)

- Milestone
- Assignees
- Assigned to Shop*
- Status
- Link (which Asset, Shot, or Task Template they're associated with)

- Predecessors*
- Constraint Type*
- Constraint Date*
- Manually Scheduled*

- Positions*

- Consilium Positions*
- Consilium Locked*

Dependencies*
Defines a relationship between two Tasks.

Key Fields:
- Source Task
- Destination Task (predecessor)
- Dependency Type (FF, FS, SS, and SF… aka "the useless one")
- Offset

Task Template
Holds a set of Tasks and dependencies to "stamp" onto another entity.

Asset
A physical or digital Asset.
For each department, we've developed a unique hierarchy of Assets to help organize the work to be done.

Art Hierarchy: Location / Set / Set Build / Set Parts
Puppets Hierarchy: Character / Puppet (Design) / Puppet Build / Puppet Subasset
Rapid: Character / Rapid Character / Variant

Fields (if used as resources like Positions):

Name
Delivery Date (“Earliest Oneline Block”, in LAIKA parlance)

Consilium Weight*
Consilium Goal*
Consilium Limit*
Consilium Scale In*
Consilium Scale Out*

Shot
A shot for the film.

Position*
An employee’s job at LAIKA - defines what work will be assigned. This data is ingested from Production Accounting so that the list of positions matches up to the list of LAIKAns available

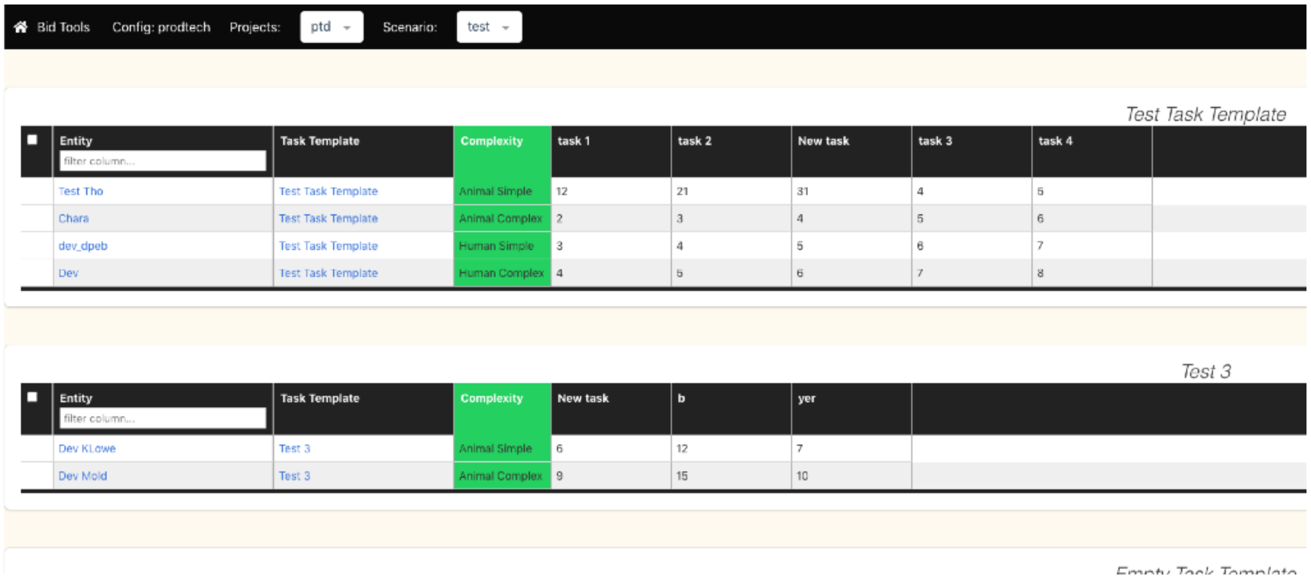
Fields:

Consilium Weight
Consilium Goal
Consilium Limit
Consilium Scale In
Consilium Scale Out

Max Workload - maximum utilization (may be less than / greater than 1)
Assignment Positions - indicates other equivalent positions that can be used when distributing work after leveling (see “Generative Scheduling to Shotgun” for more info).

Unit*
A space on our stages that can be used for filming a scene.

HumanUser
A record of a person.



Bid Tools Config: prodtech Projects: ptd Scenario: test

Test Task Template

Entity	Task Template	Complexity	task 1	task 2	New task	task 3	task 4
Test Tho	Test Task Template	Animal Simple	12	21	31	4	5
Chara	Test Task Template	Animal Complex	2	3	4	5	6
dev_dpeb	Test Task Template	Human Simple	3	4	5	6	7
Dev	Test Task Template	Human Complex	4	5	6	7	8

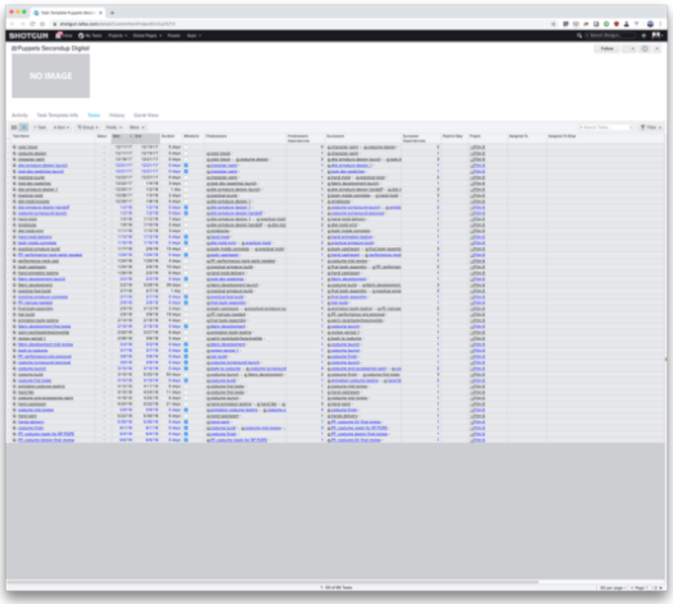
Test 3

Entity	Task Template	Complexity	New task	b	yer
Dev KLowe	Test 3	Animal Simple	6	12	7
Dev Mold	Test 3	Animal Complex	9	15	10

Empty Task Template

The Bid Process

With the schema and gantt app ready for use, it was time to move on to updating the bidding process.



We felt our bidding approach where users defined task templates and complexities for Assets was still effective, so we worked to maintain those conceptual models while streamlining the bid process.

In our new workflow, users create task templates directly in Shotgun, and can assign and edit task template and complexity assignments in a new bidding app, which updates production schedules directly.

This new bid process is entirely in-browser, with all data stored in our custom Shotgun schema.

Effective Data Modeling

How does one go about creating an effective template for an Asset?

1- Model Workflows
We encourage our Production Managers (PMs) to start by gathering information about asset creation from their shop leads (people in charge of the technical work of a particular area inside a department). The conversations with shop leads help shape the task list, relationships between the Tasks, and the number of different task templates required, which can vary greatly depending on the variety of Assets required for a show.

2- Build Templates
If the PM feels confident in their ability to estimate required work, they'll then bid out durations for specific tasks based on the complexity of each Asset. If a PM is less confident, or if a build is unusual, they often ask shop leads to weigh in on the required resources.

In general, our design goals for Task Templates are to make them as simple as possible, but no simpler. By that, we mean that the Task Template should correctly flesh out the work required to complete an Asset, accurately capturing all the major tasks and resources required, but should avoid getting too granular. Where possible, the PM should have a simple summary task like "Techvis", not a list like "Gather Reference for Techvis Launch", "Techvis Launch", "Techvis", "Techvis Review", "Techvis Revisions", "Techvis Final Approval".

Good:
["Techvis"]

Bad:
["Gather Reference for Techvis Launch", "Launch Techvis", "Techvis", "Techvis Review", "Techvis Revisions", "Techvis Final Approval"]

Once a Task Template starts down this dark path, it risks entering into Zeno's Paradox territory, with its tasks divided into ever-smaller units of work until it is impossible to conceive of completing the process. The trick is finding a balance somewhere between a single Task that says "Make the movie", and an infinite number of tasks describing all the minutiae of the filmmaking process.

3- Review and Adjust Templates As Needed
The best Task Templates are ones that capture all the work required to complete a build while being manageable enough that PMs, artists, coordinators, and shop leads can interact with the schedule without being overwhelmed. A great way to ensure that this goal is met is to do build post-mortems to evaluate which parts of the schedule were essential to maintain, and which parts could be left out without impacting the department's workflow.

It is also worth noting that "manageable" is a subjective term, and the complexity of a fully-fleshed-out data model can be staggering. For example, the Puppets department currently has 28 different Task Templates for the 378 Assets they're planning to build for our upcoming film, for a grand total of about 12,000 tasks.

The templates vary from first-up digital builds to duplicate versions of puppets to cloth swatches for VFX - and the most complex template has 97 tasks spread between 12 shops (…with a little cross-department collaboration with RP thrown in for good measure).

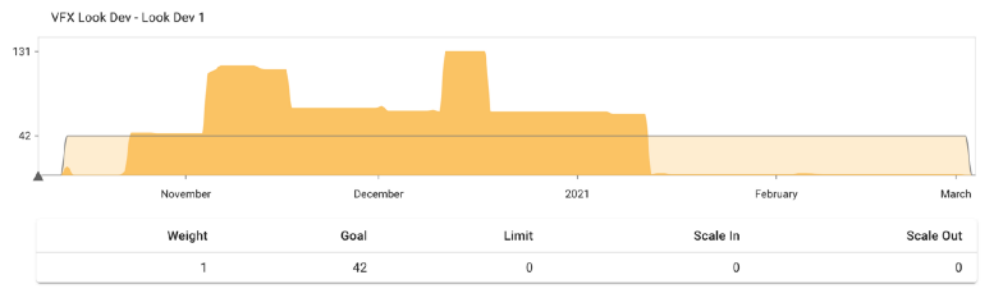
Leveling
We now had an effective way to bid and re-bid our schedules, but leveling the schedules to balance resource usage was still an error-prone, manual process.

On previous films, PMs leveled schedules by filtering down to a particular resource's Tasks, then sliding them in Shotgun so that the resource wasn't overbooked. This approach works reasonably well for a small shop (<5 people), but it doesn't scale well to departments of 20+ people, or to schedules with thousands of Tasks.

Writing a computer program to replace this process is nontrivial. Effective resource leveling is an NP-Hard problem that comes in a lot of different variations — in our case, we wanted a way to generate a schedule that met our fabrication and filming deadlines while ensuring that all of our resources were optimized -- neither over nor under, utilized.

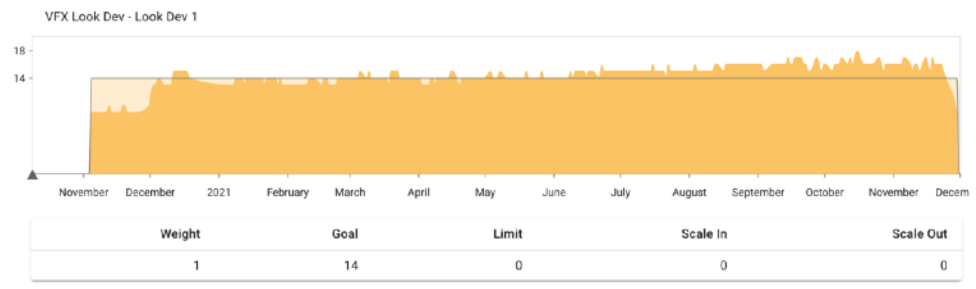
To break things down further, we need to assign start dates to Tasks so that they satisfy all their constraints and dependencies, which means putting them somewhere between their earliest possible start date, and their latest possible finish date.

We tried some in-house leveling programs on our scheduling data, as well as a modern constraint solver, but the size of our larger datasets and the processing time required for brute-force leveling made these approaches infeasible.



Weight	Goal	Limit	Scale In	Scale Out
1	42	0	0	0

VFX Look Dev - Resource Leveling — Before (Above) and After (below)

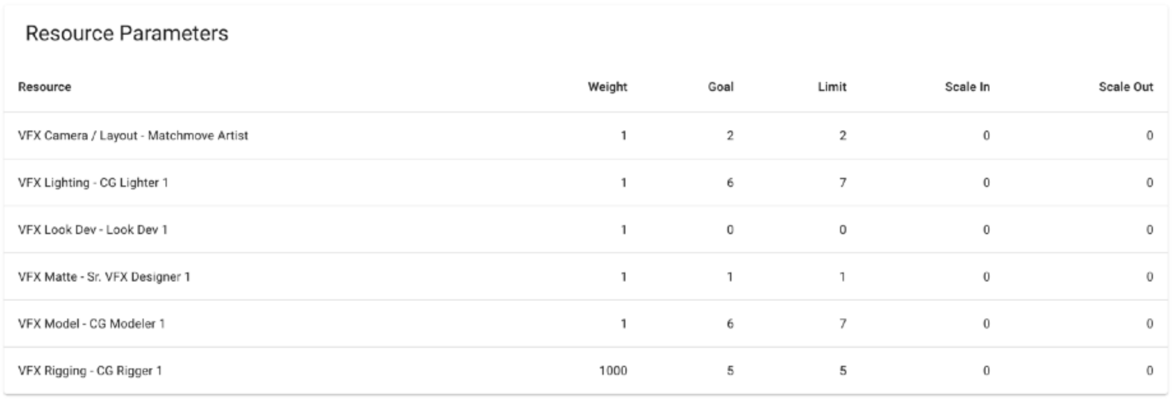


Weight	Goal	Limit	Scale In	Scale Out
1	14	0	0	0

Generative Scheduling

Fortunately, we were able to get early access to Shotgun's Generative Scheduling product (which was called "Consilium" before it was acquired by Autodesk… I tried to replace all the references to the old name, but I may have missed a few) which allows users to create multiple scenarios of a complex set of tasks, dependencies, constraints and resource requirements, and then, using a machine learning approach, automatically generate resource optimized schedules.

Generative Scheduling enables a user to create different scheduling scenarios where they specify parameters for how many tasks should be assigned to a resource.



Resource Parameters

Resource	Weight	Goal	Limit	Scale In	Scale Out
VFX Camera / Layout - Matchmove Artist	1	2	2	0	0
VFX Lighting - CG Lighter 1	1	6	7	0	0
VFX Look Dev - Look Dev 1	1	0	0	0	0
VFX Matte - Sr. VFX Designer 1	1	1	1	0	0
VFX Model - CG Modeler 1	1	6	7	0	0
VFX Rigging - CG Rigger 1	1000	5	5	0	0

There are five resource parameters to be adjusted.

"Weight" - the relative importance of meeting a resource's target curve compared to other target resource curves.
"Goal" - the ideal number of tasks for a resource at any one time,
"Limit" - the upper bound on tasks
"Scale in/out" - the number of days over which to scale the resource curve in and out.

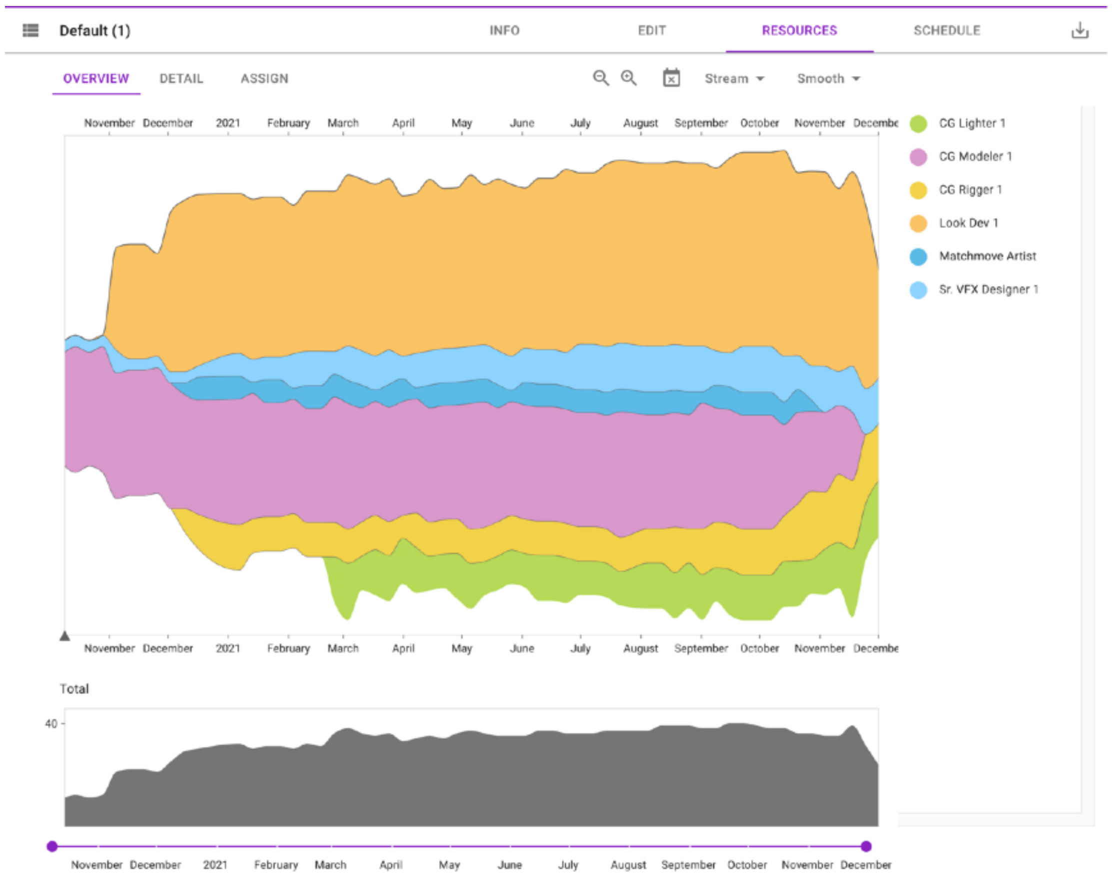
A great feature of the app is that it doesn't require any resource parameters - if left blank, the engine will try to balance the resource curves automatically, which can be helpful for the initial bidding phase of a production, where resource counts have not yet been determined.

Once the user has entered their targets and pressed "Schedule", the scheduling engine searches for a way to adjust Task start dates for the schedule (while still satisfying all constraints / dependencies) so each resource is as close as possible to its target curve.

This process happens relatively quickly - it can take anywhere from a few seconds to a few minutes to find a solution, depending on the size of the schedule.

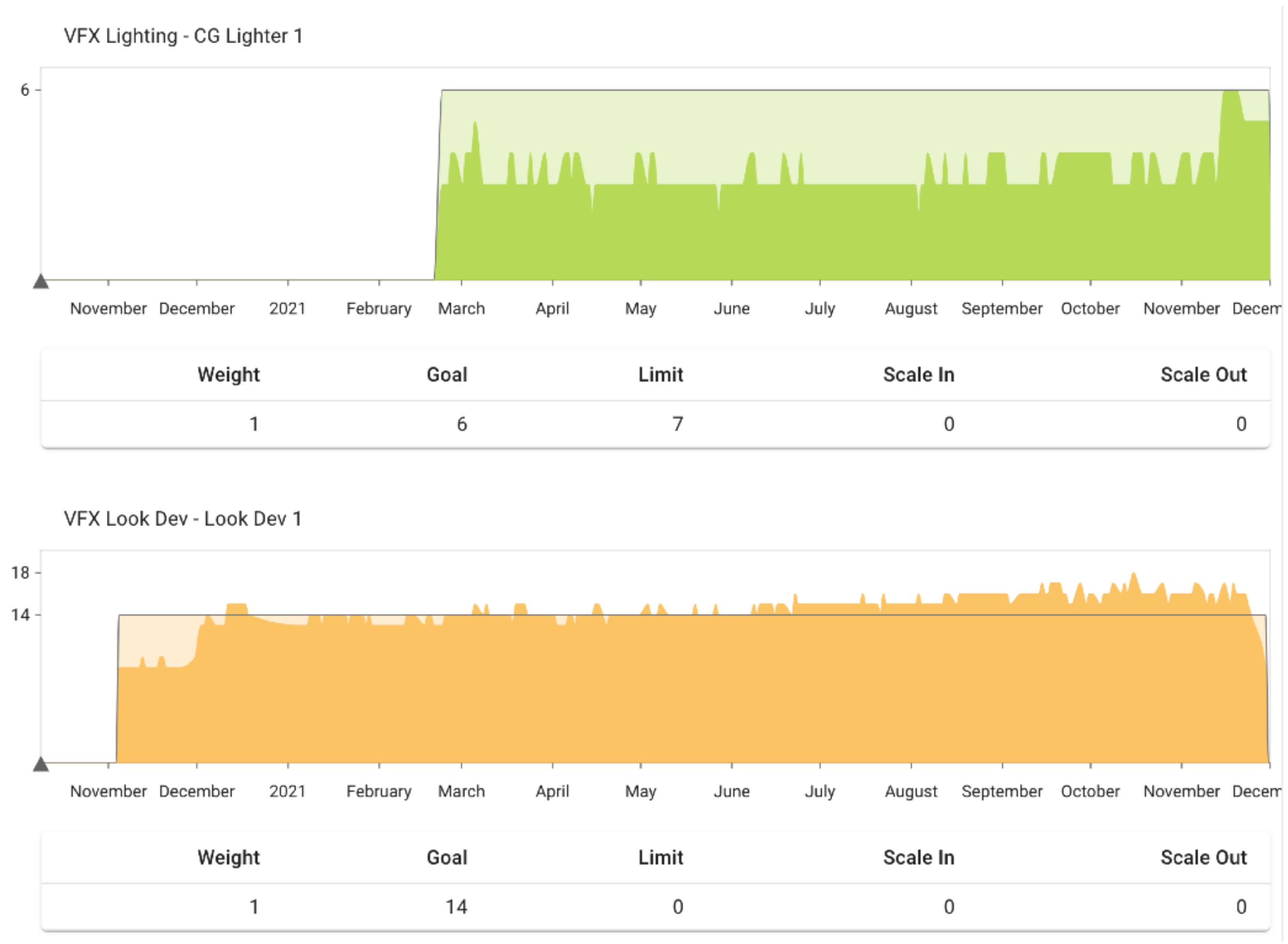
Once a solution is found, the user can use the views available in the "Resources" tab to see the results.

Resources Tab



(Resources Tab - Overview)

The Overview tab is a stream graph (not a stacked area graph) showing all the resources together. From this tab, a user can identify points of concern, then switch to the Detail and Assign views to take a closer look.



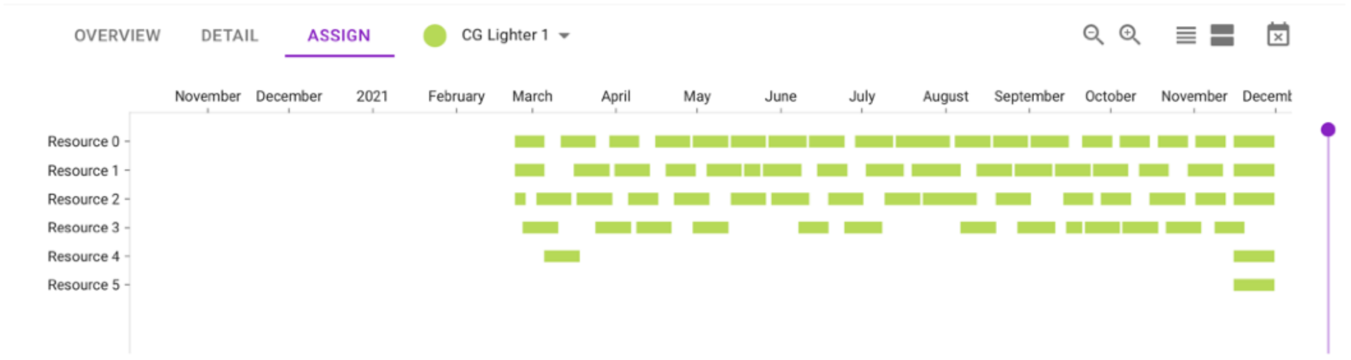
Weight	Goal	Limit	Scale In	Scale Out
1	6	7	0	0

Weight	Goal	Limit	Scale In	Scale Out
1	14	0	0	0

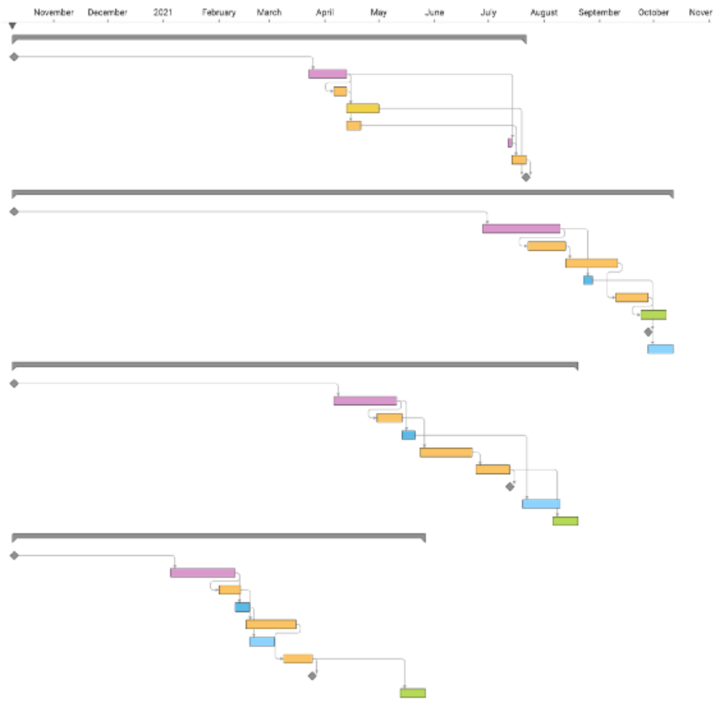
(Resources Tab - Detail View)

The detail view shows the target resource curves (depicted as a line), along with the actual utilization of the resource in the current scenario.

Loading multiple resource curves at the same time is a nice way to check the tradeoffs the app has made in its schedule -- a production manager is often tasked with making tradeoffs between the burden on different resources, and these graphing tools give them a quick way to see the results of their selections.



Another useful visualization tool is the "Assign" tab, which lays out Tasks for a resource in a Gantt view. From here, a user can see how the app has chosen to assign and overlap tasks, and see what assignments will look like when they bring the data back into Shotgun.



Lastly, Generative Scheduling offers a full gantt chart representation of its plan, which offers insight into the constraints and dependencies in a user's schedule.

Preparing Data for Generative Scheduling

It's still early days for the Generative Scheduling app, so even though our data was stored in Shotgun, we had to spend a bit of time preparing our data for its scheduling engine.

We came up with two different configurations – the first is intended for asset fabrication leveling, and the second is for leveling scene shoot dates.

Asset Fabrication Leveling

1. Tasks for each Asset
Tasks and their dependencies are loaded from Shotgun. Any Task that doesn't have dependencies is locked in place.

2. Dependencies between the Assets
After getting all the individual task templates working and entering custom per-Asset durations for each task, the PM also needs to determine the relationships between builds.

In our initial approach, this was as simple as saying that each dupe build depended on the completion of its first up parent build (and costume changes depended on completion of the first up build of a character).

Unfortunately, that wasn't nearly detailed enough, so we now link the starts of dupe builds to specific tasks in their parent builds, which provides greater flexibility for determining which tasks in a parent build need to be completed before a dupe can begin.

Example: Asset Dependencies for Foodles / Foodles Old

```
Foodles A (1)
   Foodles A (2) -- Launch Dupe Builds (Foodles A (1))
   Foodles A (3) -- Launch Dupe Builds (Foodles A (1))

   Foodles B (1) -- Launch Dupe Builds (Foodles A (1))
      Foodles B (2) -- Launch Dupe Builds (Foodles B (1))
      Foodles B (3) -- Launch Dupe Builds (Foodles B (1))

   Foodles C (1) -- Launch Dupe Builds (Foodles A (1))
   Foodles D (1) -- Costume director final review (Foodles A (1))
      Foodles D (2) -- Costume director final review (Foodles D (1))
      Foodles D (3) -- Launch Dupe Builds (Foodles D (1))
      …
Foodles Old (1)
   Foodles Old (1) -- Launch Dupe Builds (Foodles A (1))
   Foodles Old (2) -- Launch Dupe Builds
```

3. Position Assignments for each Task
These were captured in the "Consilium Positions" field as generic positions -- instead of entering each of 15 Costume Fabricators, a generic "Costume Fabricator" is used to indicate a class of position. Future versions of the app will hopefully be able distribute work among multiple

incarnations of the same position (i.e. distribute the "costume build" Tasks among Costume Fabricators 1-12). To better level across multiple positions, we've also added “Meta Positions” that summarize maximum headcount within certain groups.

4. Custom Constraints
Constraints are additional restrictions on the scheduling engine beyond the inter-task dependencies.

For Puppets, these constraints were:
a. Requiring each Asset to hit its block date

b. Locking any in-progress/complete/approved Task in-place (the engine shouldn't adjust its dates)

c. Locking certain builds in-place in their entirety.
Certain builds are already in progress, or need to deliver by certain dates, so all of their Tasks were locked in place and the engine scheduled the remainder of the Tasks around them.

Scene Leveling
This workflow is still a work in progress, but the end goal is to create a schedule that minimizes the number of Assets and Animators required to make a film.

1. Load scene tasks from Shotgun
At LAIKA, our initial shoot schedule is defined by developing a “Oneline” schedule that describes the scenes in the film and what assets are assigned to each scene. After animatics are finished, these scenes are broken down into individual Shots used to film the movie.

2. Generate Resources for each Task
For each Scene task, sg_to_consilium assigns an Animator and generalized versions of the specified Assets as resources for the Task. To make things a little more complicated, it also looks at size of the sets that are assigned to the Task and adds different “Unit” resources based on the set size.

For example, scene 0700.scn023 requires the set “Ext Front Lawn (1)”. Since we don't know how many duplicate builds of Ext Front Lawn (1) we might need, we'll assign its parent Set Design “Ext Front Lawn” as the resource. We'll also look at the size specs for Ext Front Lawn, and find that it has a designation of “XL”, with a split unit size of “M”.

Along with the sets, we'll also assign generic versions of the specified puppets, Foodles A (1) and Babbagepatch McGorkenSpork A (1), and an Animator.

Combine all those resources together, and you get something like:

0700.scn023 Resource List:
Ext Front Lawn: 1
Generic Units: 2
M Unit: 1

XL Unit: 1
Foodles A Puppet: 1
Babbagepatch McGorkenSpork A Puppet: 1
Animator: 1

… then do that for all the other Tasks, and you’ve got yourself some prepped data.

Shotgun To Generative Scheduling

The process of bringing a schedule into the Generative Scheduling app involved creating a new tool called “Shotgun to Consilium". It provides information about infeasible schedule constraints and generates a JSON export of the Shotgun schedule and constraints that can be loaded into the engine.

Handling Infeasible Constraints:
The latest version of the tool generates a table of what are considered to be "infeasible constraints" -- any block dates that are impossible for Assets to hit based on upstream dependencies, regardless of the number of people available.

For example, if an Asset is needed on March 30th, 2021 for its first scene, we'd place a "finish by" constraint on the Asset, saying it absolutely must finish by that date.

Most of the time, these constraints can be accommodated by a specific ordering of tasks, but sometimes it's completely impossible to create a schedule that meets a constraint -- for example, if we can't start building our Asset until January 1st, 2021, and the tasks required to build it will take 6 months, it's impossible to meet the "March 31st 2021" constraint date - the soonest we could deliver the Asset is July 2021.

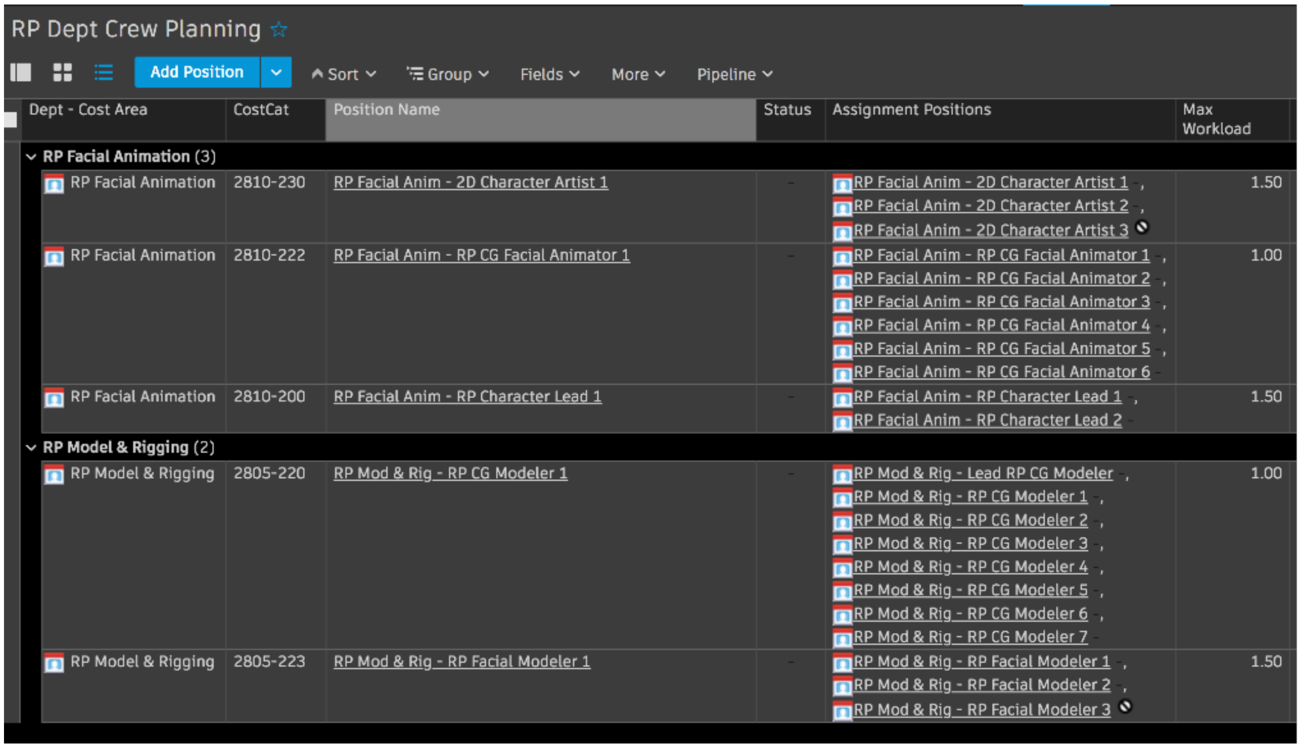
code	current_block	earliest_feasible_block	discrepancy
Babbagepatch McGorkenspork (2)	2020-05-21	2020-05-22	1
Enzo Gorlomi (1)	2021-01-26	2021-02-04	7
Antonio Margheretti (1)	2020-08-07	2020-08-18	7
Dominick Decocco (1)	2020-03-09	2020-03-26	13

When we first started working with Consilium, we discovered numerous instances of our default constraints for scheduling being infeasible - the notification before leveling helped us fix these constraints before they became an issue.

Generative Scheduling to Shotgun (“Consilium to Shotgun”)
To bring a plan back to Shotgun, we’ve developed a tool that handles copying the schedule into a planning scenario for further review.

In addition to bringing back start and end dates of Tasks, this tool is also able to create position assignments for the tasks, distributing work so that each resource has tasks assigned, but isn’t overbooked. For example, we assign only “Costumer 1” to the Tasks requiring a Costumer when we import our schedule into Shotgun Generative Scheduling, but on return to Shotgun, we want to distribute that work to Costumers 1 through 5.

To handle this, we structure our position table so that for each position used in leveling, one or more equivalent positions can be assigned (“Assignment Positions”) and used to distribute work. Additionally, the user can specify a “Max Workload” for a position to indicate whether a utilization of more or less than 100% is acceptable.

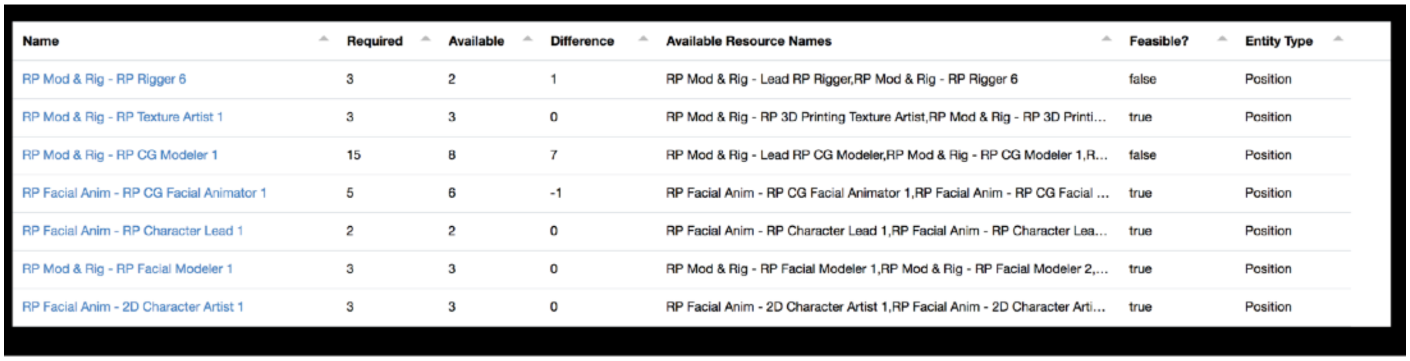


RP Dept Crew Planning

Add Position | Sort | Group | Fields | More | Pipeline

Dept - Cost Area	CostCat	Position Name	Status	Assignment Positions	Max Workload
RP Facial Animation (3)					
RP Facial Animation	2810-230	RP Facial Anim - 2D Character Artist 1	-	RP Facial Anim - 2D Character Artist 1, RP Facial Anim - 2D Character Artist 2, RP Facial Anim - 2D Character Artist 3	1.50
RP Facial Animation	2810-222	RP Facial Anim - RP CG Facial Animator 1	-	RP Facial Anim - RP CG Facial Animator 1, RP Facial Anim - RP CG Facial Animator 2, RP Facial Anim - RP CG Facial Animator 3, RP Facial Anim - RP CG Facial Animator 4, RP Facial Anim - RP CG Facial Animator 5, RP Facial Anim - RP CG Facial Animator 6	1.00
RP Facial Animation	2810-200	RP Facial Anim - RP Character Lead 1	-	RP Facial Anim - RP Character Lead 1, RP Facial Anim - RP Character Lead 2	1.50
RP Model & Rigging (2)					
RP Model & Rigging	2805-220	RP Mod & Rig - RP CG Modeler 1	-	RP Mod & Rig - Lead RP CG Modeler, RP Mod & Rig - RP CG Modeler 1, RP Mod & Rig - RP CG Modeler 2, RP Mod & Rig - RP CG Modeler 3, RP Mod & Rig - RP CG Modeler 4, RP Mod & Rig - RP CG Modeler 5, RP Mod & Rig - RP CG Modeler 6, RP Mod & Rig - RP CG Modeler 7	1.00
RP Model & Rigging	2805-223	RP Mod & Rig - RP Facial Modeler 1	-	RP Mod & Rig - RP Facial Modeler 1, RP Mod & Rig - RP Facial Modeler 2, RP Mod & Rig - RP Facial Modeler 3	1.50

(Assignment Positions / Max Workload data for the Rapid Prototyping group)



Name	Required	Available	Difference	Available Resource Names	Feasible?	Entity Type
RP Mod & Rig - RP Rigger 6	3	2	1	RP Mod & Rig - Lead RP Rigger,RP Mod & Rig - RP Rigger 6	false	Position
RP Mod & Rig - RP Texture Artist 1	3	3	0	RP Mod & Rig - RP 3D Printing Texture Artist,RP Mod & Rig - RP 3D Printi...	true	Position
RP Mod & Rig - RP CG Modeler 1	15	8	7	RP Mod & Rig - Lead RP CG Modeler,RP Mod & Rig - RP CG Modeler 1,R...	false	Position
RP Facial Anim - RP CG Facial Animator 1	5	6	-1	RP Facial Anim - RP CG Facial Animator 1,RP Facial Anim - RP CG Facial ...	true	Position
RP Facial Anim - RP Character Lead 1	2	2	0	RP Facial Anim - RP Character Lead 1,RP Facial Anim - RP Character Lea...	true	Position
RP Mod & Rig - RP Facial Modeler 1	3	3	0	RP Mod & Rig - RP Facial Modeler 1,RP Mod & Rig - RP Facial Modeler 2,...	true	Position
RP Facial Anim - 2D Character Artist 1	3	3	0	RP Facial Anim - 2D Character Artist 1,RP Facial Anim - 2D Character Arti...	true	Position

(Table showing results of assigning positions to a leveled schedule)

Generative Scheduling Summary
Generative Scheduling has made a big impact on our scheduling workflow, and is being used in production at LAIKA for managing tasks in several departments, most notably as the primary means of scheduling the Puppets and Art departments, with ongoing experiments for Rapid, VFX, and Stage workflows.

To summarize its key features:

1. Leveling for All Resources Simultaneously
The engine is able to consider all the resource groups at once while still placing higher priority on certain groups, and provides a resource detail view that makes it easy to compare available headcount to workload.

2. Effective Visualization
The Generative Scheduling Resource views provide an effective way for production managers to see tradeoffs in weights and curves for different resources, something that's hard to do when manually scheduling.

3. Fast Schedule Iteration
Third, it offers fast schedule iteration.

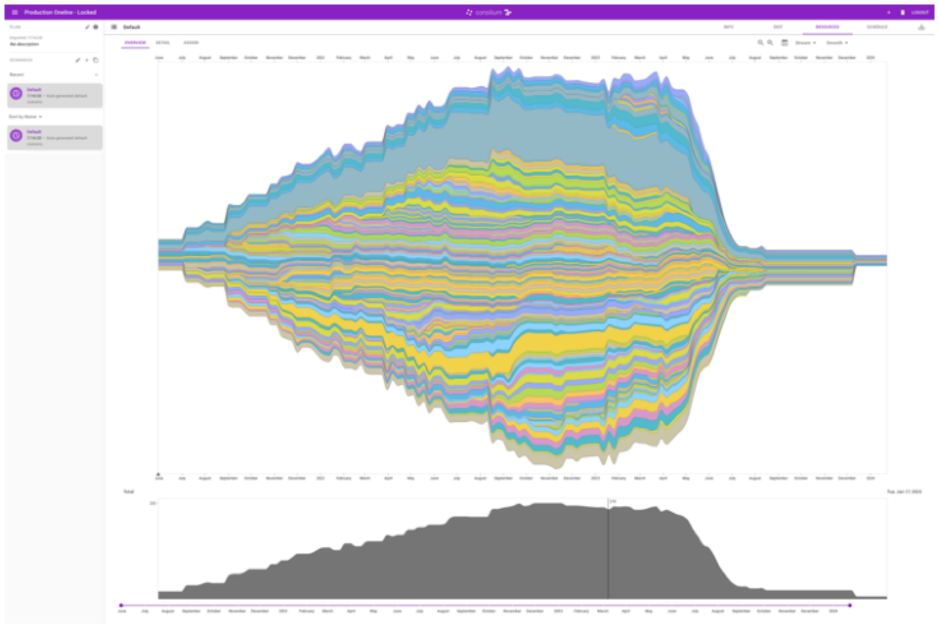
Schedule data is constantly in flux, so being able to adjust and level a schedule in a matter of minutes means that the schedule can respond quickly to changes in shoot schedules or staffing changes. It also enables the user to evaluate how different scenarios might affect their resources without spending days rearranging their schedules.

4. Assignment
Finally, the engine is able to assign tasks to specific resources, which enables users to assign tasks to a "class" of resource (like a generic costumer) and have the app assign and evenly distribute work between Costumers 1 through 5.

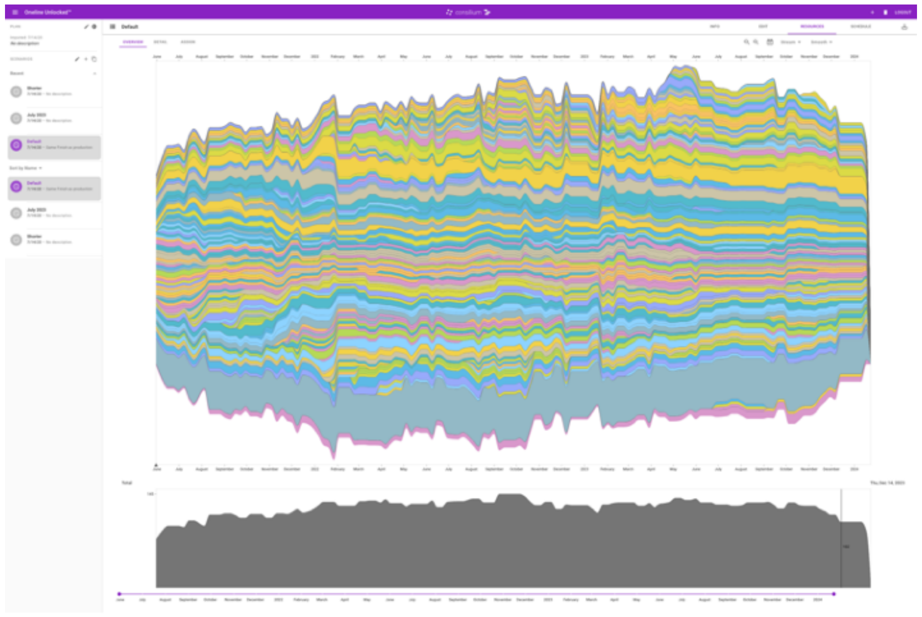
Generative Scheduling / Human Scheduling Comparison

Let’s take a look at an example of our success with our new scheduling features.

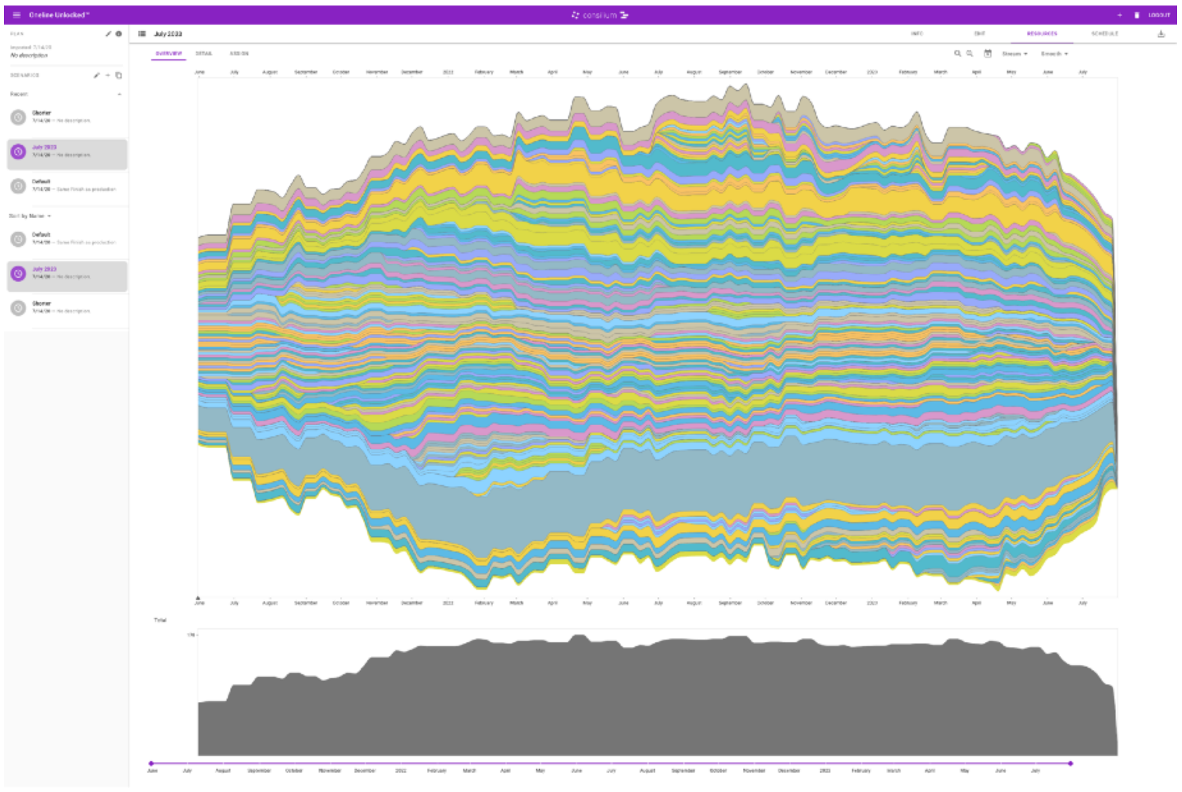
First, here's an example of what LAIKA calls the "Oneline" schedule - each task represents a scene in a film, and the goal is to minimize the number of Assets required



Above is the human-leveled schedule, which requires 334 Assets to shoot the film over 3 and a half years. Below is a Generative Scheduling-leveled schedule with the same start/end dates, which is able to reduce the number of assets required to just 251.

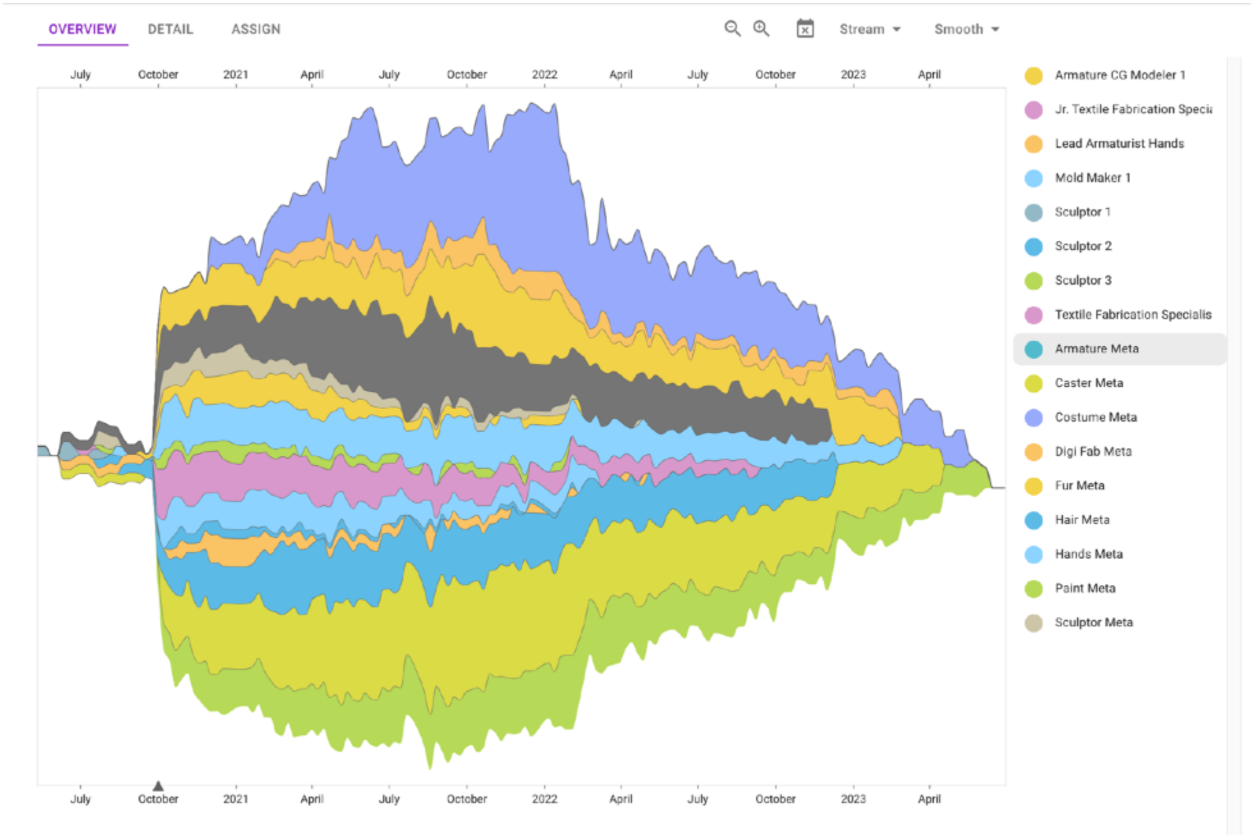


The iteration speed of the scheduling engine is also impressive. The human leveled schedule took weeks of time to build, whereas running it in the app takes less than a minute.

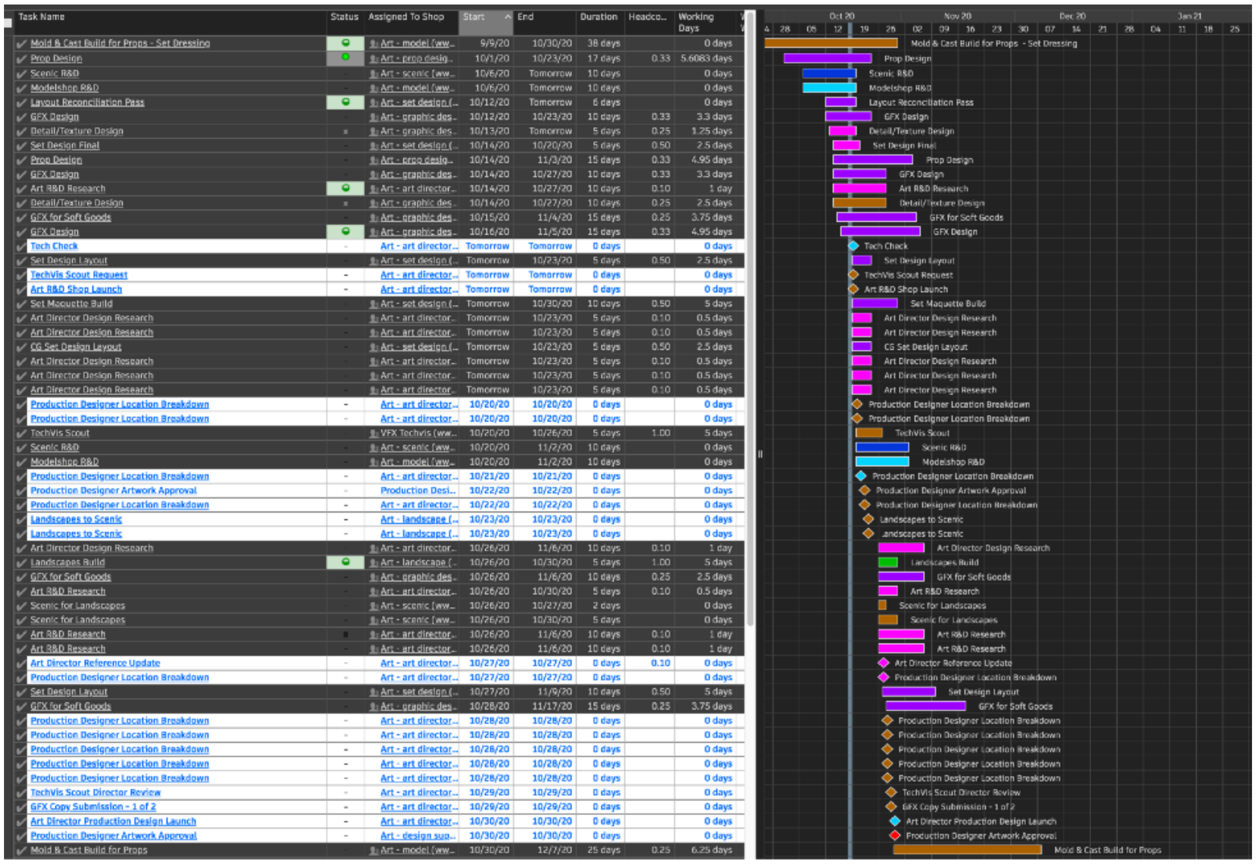


This means we have time to run other scenarios - like a "finish the movie a year sooner"" scenario that uses 308 Assets, or a "finish the movie six months sooner" scenario, which uses just 270 Assets – a significant labor reduction over the initial human-leveled scenario.

Second, here's an example of the puppet schedule for our upcoming film. It has 12,000 tasks in it, and represents work for a department of 60 people. With these new tools, it's possible to quickly run scenarios where we put emphasis on the needs of a particular department -- for example, leveling with a focus on the costume department, or changing gears and emphasizing the mold making group.



Day to Day Task Management in Shotgun



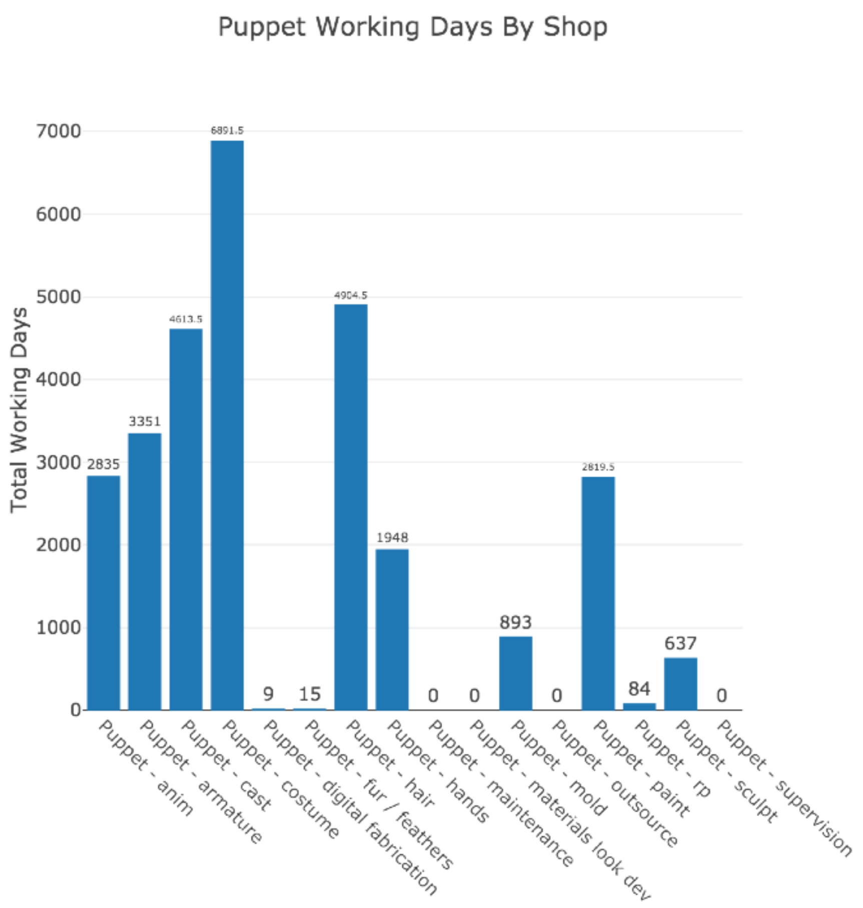
Task Name	Status	Assigned To Shop	Start	End	Duration	Headco...	Working Days
Mold & Cast Build for Props - Set Dressing	●	Art - model (ww...	9/9/20	10/30/20	38 days		0 days
Prop Design	●	Art - prop desig...	10/1/20	10/23/20	17 days	0.33	5.6083 days
Scenic R&D	-	Art - scenic (ww...	10/6/20	Tomorrow	10 days		0 days
Modelshop R&D	-	Art - model (ww...	10/6/20	Tomorrow	10 days		0 days
Layout Reconciliation Pass	●	Art - set design (...	10/12/20	Tomorrow	6 days		0 days
GFX Design	-	Art - graphic des...	10/12/20	10/23/20	10 days	0.33	3.3 days
Detail/Texture Design	x	Art - graphic des...	10/13/20	Tomorrow	5 days	0.25	1.25 days
Set Design Final	-	Art - set design (...	10/14/20	10/20/20	5 days	0.50	2.5 days
Prop Design	-	Art - prop desig...	10/14/20	11/3/20	15 days	0.33	4.95 days
GFX Design	-	Art - graphic des...	10/14/20	10/27/20	10 days	0.33	3.3 days
Art R&D Research	●	Art - art director...	10/14/20	10/27/20	10 days	0.10	1 day
Detail/Texture Design	x	Art - graphic des...	10/14/20	10/27/20	10 days	0.25	2.5 days
GFX for Soft Goods	-	Art - graphic des...	10/15/20	11/4/20	15 days	0.25	3.75 days
GFX Design	●	Art - graphic des...	10/16/20	11/5/20	15 days	0.33	4.95 days
Tech Check	-	Art - art director...	Tomorrow	Tomorrow	0 days		0 days
Set Design Layout	-	Art - set design (...	Tomorrow	10/23/20	5 days	0.50	2.5 days
TechVis Scout Request	-	Art - art director...	Tomorrow	Tomorrow	0 days		0 days
Art R&D Shop Launch	-	Art - art director...	Tomorrow	Tomorrow	0 days		0 days
Set Maquette Build	-	Art - set design (...	Tomorrow	10/30/20	10 days	0.50	5 days
Art Director Design Research	-	Art - art director...	Tomorrow	10/23/20	5 days	0.10	0.5 days
Art Director Design Research	-	Art - art director...	Tomorrow	10/23/20	5 days	0.10	0.5 days
CG Set Design Layout	-	Art - set design (...	Tomorrow	10/23/20	5 days	0.50	2.5 days
Art Director Design Research	-	Art - art director...	Tomorrow	10/23/20	5 days	0.10	0.5 days
Art Director Design Research	-	Art - art director...	Tomorrow	10/23/20	5 days	0.10	0.5 days
Art Director Design Research	-	Art - art director...	Tomorrow	10/23/20	5 days	0.10	0.5 days
Production Designer Location Breakdown	-	Art - art director...	10/20/20	10/20/20	0 days		0 days
Production Designer Location Breakdown	-	Art - art director...	10/20/20	10/20/20	0 days		0 days
TechVis Scout	-	VFX Techvis (ww...	10/20/20	10/26/20	5 days	1.00	5 days
Scenic R&D	-	Art - scenic (ww...	10/20/20	11/2/20	10 days		0 days
Modelshop R&D	-	Art - model (ww...	10/20/20	11/2/20	10 days		0 days
Production Designer Location Breakdown	-	Art - art director...	10/21/20	10/21/20	0 days		0 days
Production Designer Artwork Approval	-	Production Desi...	10/22/20	10/22/20	0 days		0 days
Production Designer Location Breakdown	-	Art - art director...	10/22/20	10/22/20	0 days		0 days
Landscapes to Scenic	-	Art - landscape (...	10/23/20	10/23/20	0 days		0 days
Landscapes to Scenic	-	Art - landscape (...	10/23/20	10/23/20	0 days		0 days
Art Director Design Research	-	Art - art director...	10/26/20	11/6/20	10 days	0.10	1 day
Landscapes Build	●	Art - landscape (...	10/26/20	10/30/20	5 days	1.00	5 days
GFX for Soft Goods	-	Art - graphic des...	10/26/20	11/6/20	10 days	0.25	2.5 days
Art R&D Research	-	Art - art director...	10/26/20	10/30/20	5 days	0.10	0.5 days
Scenic for Landscapes	-	Art - scenic (ww...	10/26/20	10/27/20	2 days		0 days
Scenic for Landscapes	-	Art - scenic (ww...	10/26/20	10/30/20	5 days		0 days
Art R&D Research	■	Art - art director...	10/26/20	11/6/20	10 days	0.10	1 day
Art R&D Research	-	Art - art director...	10/26/20	11/6/20	10 days	0.10	1 day
Art Director Reference Update	-	Art - art director...	10/27/20	10/27/20	0 days	0.10	0 days
Production Designer Location Breakdown	-	Art - art director...	10/27/20	10/27/20	0 days		0 days
Set Design Layout	-	Art - set design (...	10/27/20	11/9/20	10 days	0.50	5 days
GFX for Soft Goods	-	Art - graphic des...	10/28/20	11/17/20	15 days	0.25	3.75 days
Production Designer Location Breakdown	-	Art - art director...	10/28/20	10/28/20	0 days		0 days
Production Designer Location Breakdown	-	Art - art director...	10/28/20	10/28/20	0 days		0 days
Production Designer Location Breakdown	-	Art - art director...	10/28/20	10/28/20	0 days		0 days
Production Designer Location Breakdown	-	Art - art director...	10/28/20	10/28/20	0 days		0 days
Production Designer Location Breakdown	-	Art - art director...	10/28/20	10/28/20	0 days		0 days
TechVis Scout Director Review	-	Art - art director...	10/29/20	10/29/20	0 days		0 days
GFX Copy Submission - 1 of 2	-	Art - art director...	10/29/20	10/29/20	0 days		0 days
Art Director Production Design Launch	-	Art - art director...	10/30/20	10/30/20	0 days		0 days
Production Designer Artwork Approval	-	Art - design sup...	10/30/20	10/30/20	0 days		0 days
Mold & Cast Build for Props	-	Art - model (ww...	10/30/20	12/7/20	25 days	0.25	6.25 days

Once a leveled schedule is back in Shotgun, day to day management of the schedule is handled by updating assignees, start / end dates, and statuses directly in the Shotgun Gantt view. This is effective for short-term scheduling, as it gives users complete control over the layout of their schedule. For situations where a user needs to push out all future tasks on a build, our custom Gantt tool can be used to auto-schedule the remaining tasks.

When bigger changes are needed (delivery date changes, or build additions/removals), our approach is to lock the next month of work in-place, then run Consilium to re-level the schedule around the locked tasks.

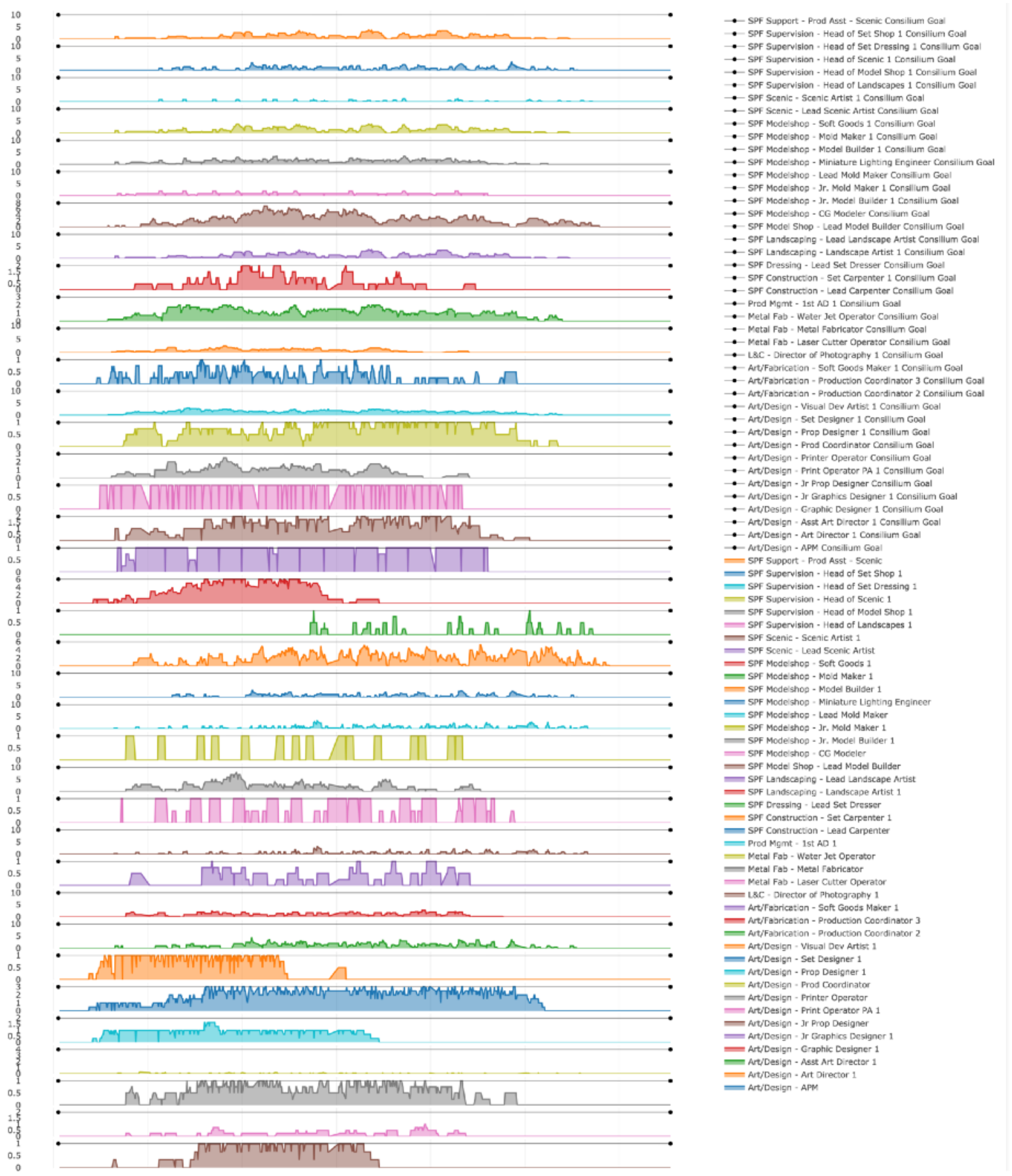
Custom Shotgun Workflows

Shotgun's Gantt chart and Task table tools provided a great way to filter and interact with a leveled schedule, but summarizing and visualizing task data remained an issue. Users were interested in evaluating the state of their schedule and the impact of smaller manual moves without round tripping to the Generative Scheduling tool, and were interested in a variety of summaries and reports on their schedule data. To meet these needs, we developed some custom graphing and reporting tools using Plotly.js.

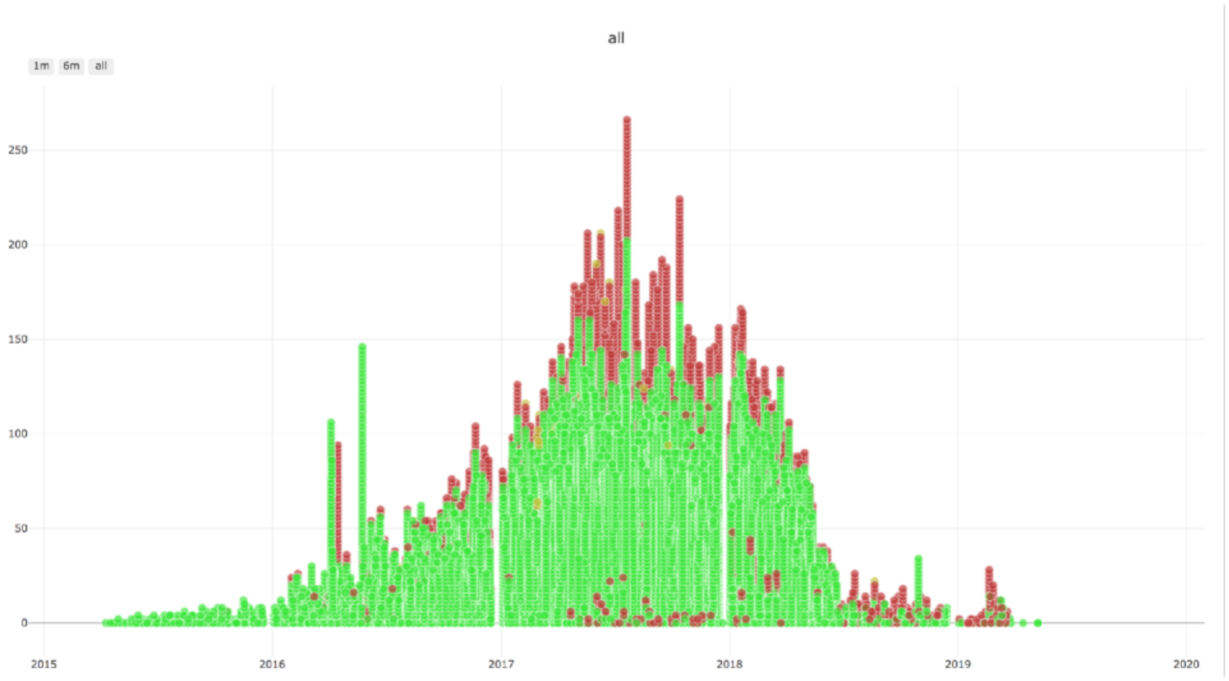


To better summarize scheduling information, we created a few new visualization tools. The first was a live updating bar chart that calculated working days by shop for the show.

The second was similar to the Generative Scheduling “Detail” view, and showed the workload for each position by day, along with a line depicting the goal for that position. The chart could also be interactively filtered to show just the workload for a particular position, or the impact of a particular set of Assets on a schedule.

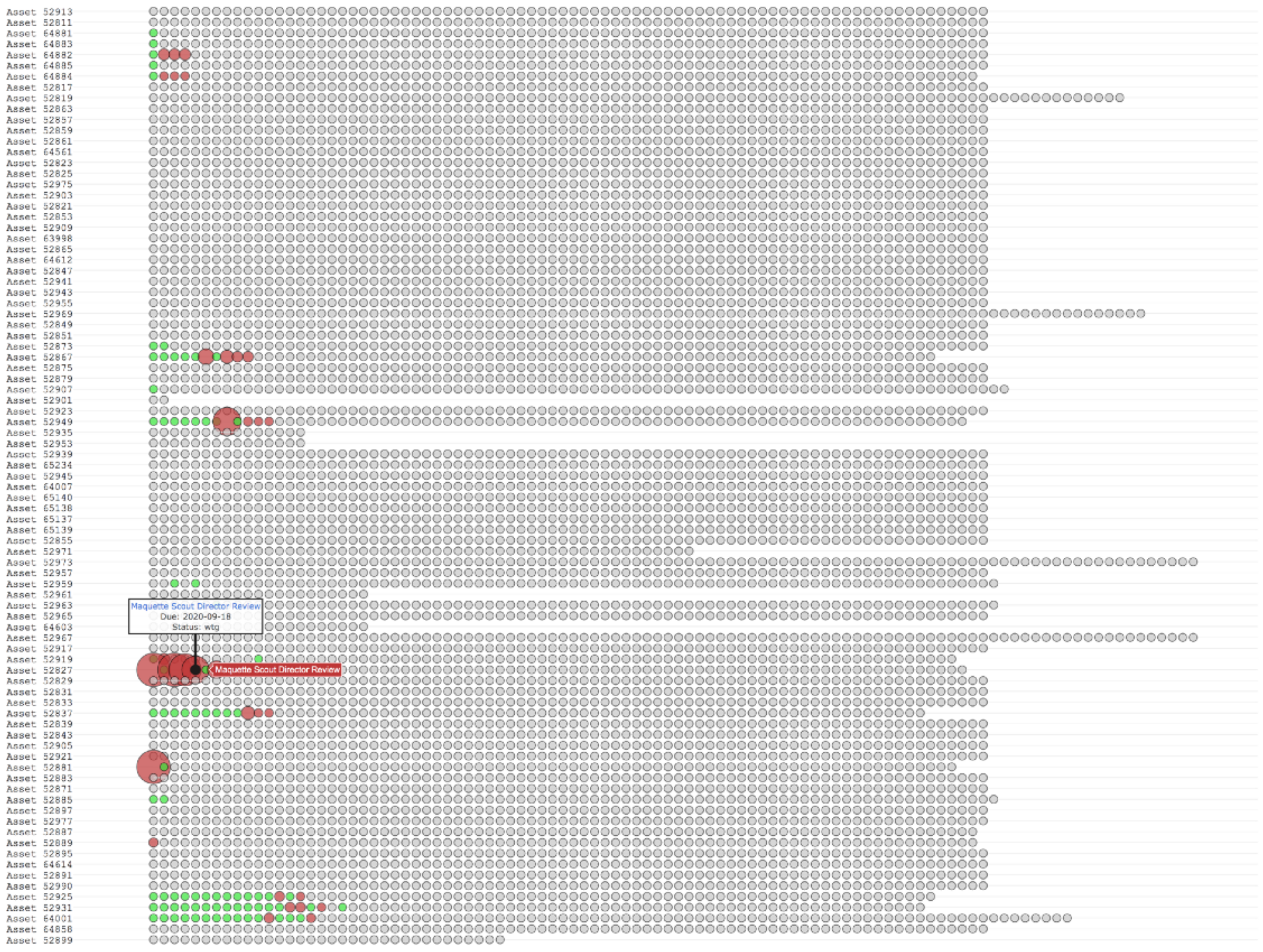


The third and fourth are experimental tools for visualizing the overall state of a schedule.



The project task summary graph shows a timeline with tasks, either for an entire show or for a particular department / shop. The color of the task circle indicates the task’s status - green for complete, grey for waiting to start, red for overdue.

This is also an example of our struggle to capture accurate actuals from our schedule — the above graph is schedule data from *Missing Link*, which apparently has unfinished work despite coming out a year ago.



The "timeline graph" similarly shows a list of tasks laid out over time, but in this case, grouped by the Asset they help create. As in the project graph, this graph uses color to communicate the task status - green for complete, grey for waiting to start, red for overdue. A unique part of this schedule is that a Task's circle gets bigger based on the number of days that a Task is overdue.

The timeline graph can also be configured to show a specific list of task names along the x-axis if a user is only interested in progress toward specific build milestones.

Designing Effective Visualizations

One of the challenges of this project was coming up with effective visualizations for displaying scheduling information.

If applied incorrectly, graphing and reporting tools often end up being distractions. A shiny viz library can become the proverbial hammer that makes all upcoming projects look like nails.

We were looking for new ways to summarize information beyond the traditional gantt chart view, and display it in ways that helped users get better insights into their data.

Here are some our key learnings, laid out as a step-by-step recipe for better visualizations.

1 - Understand the user's questions.

The best place to start is understanding what a user is trying to learn from their data. Asking questions about their workflow is helpful, but we found that the best way to learn was a desk side interview where a technologist had a chance to watch a user work through day-to-day tasks in the existing system. Don't start designing until you know what questions you're trying to answer. Down that dark path lie "balloon graphics" - nice to look at, but empty inside.

2 - Learn about the Dataset

Develop as much domain knowledge as possible for the dataset in question. This will help you write more efficient queries, and get a better sense of how your particular dataset compares to datasets used in other visualizations.

3 - Look at a LOT of Visualizations

Speaking of other visualizations… spend some time looking at charts used to solve similar problems (Edward Tufte's books are a great resource for thoughtful critical analysis). Effective visualization is a highly creative endeavor, and inspiration can come from a wide variety of sources.

4 - Experiment. Iterate. Experiment Again.

Start simple - sketch the concept out on paper to see if it makes sense, then try coding a basic example. Whenever possible, use real data in your experiments; toy datasets often lack the complexity of the real thing, and that can lead to poor design choices (at LAIKA, even our biggest datasets are "small" to Big Data practitioners, so this may or may not be feasible).

Scheduling Process - In Summary

Now that you've learned about our tooling, let's run through our scheduling process, step by step.

1. Bidding
The process begins with a bidding phase, where the tasks for a schedule are created.

During this phase, task templates are created using our custom gantt tool, and complexities are assigned using the bid app to create a schedule in Shotgun.

2. "Shotgun => Generative Scheduling"

Second, the user runs our "Shotgun To Generative Scheduling" tool, selects a configuration and plan start date, presses "submit", and gets back:

- A JSON representation of the schedule formatted for Generative Scheduling.
- A webpage that displays:
    a. a table specifying all the infeasible constraints
    b. a tree representation of all the inter-activity dependencies

3. Generative Scheduling
Third, the user uploads their plan to Generative Scheduling and levels their schedule.

4. "Generative Scheduling => Shotgun"
Fourth, the user downloads their schedule from Consilium and uploads it back to Shotgun.

5. Day to Day Task Management in Shotgun

The user manages their schedule in Shotgun, and performs a Shotgun <=> Generative Scheduling roundtrip when making major changes.

Limitations

Our new scheduling workflow has been quite successful so far, but it does have some limitations.

1. Data Modeling
First, Using the Gantt and Generative Scheduling successfully requires developing a rich dataset that encompasses all the information a user manipulating a schedule by hand would use.

This can be time-consuming, and finding the right representation can be difficult.

If data is modeled incorrectly or incompletely, the engine may make unintuitive leveling choices - like deciding to create months of space between two tasks that would ideally be done days apart to avoid context switching.

Additionally, certain scenarios (like tasks having varying durations based on assignee) are still difficult to represent in our scheduling data model, or aren't supported by the scheduling engine.

2. Maintenance
Second, Keeping a schedule and its dependencies up to date is a major time investment - it requires frequent adjustments to task statuses, start/end dates, and assignments.

We need accurate actuals in order to effectively level the remainder of the schedule -- in one department on a recent project, the schedule was very out of sync with the actuals, and it made it difficult to assess how much work remained to be leveled.

3. Learning Curve
Third, thinking in terms of constraints and dependencies can be a challenge to those accustomed to manual resource leveling, and some production managers have been reluctant to embrace the new leveling options provided by Generative Scheduling, preferring instead to stick to their traditional processes.

4. Time-based Scheduling
Shotgun’s built-in Tasks (and the current Generative Scheduling engine) are limited to scheduling in single-day increments, with no support for scheduling by hours /minutes. To bring these scheduling capabilities to the stage schedule and 3D printing / machining pipelines will require scheduling down to the hour.

Future Work
There are definitely some areas we'd like to improve on in the future.

1. Studio-wide scheduling
First, we have not yet attempted to combine all of LAIKA's scheduling information into a single schedule optimized in the Generative Scheduling app.

Such a schedule would be enormously complex, but would aid in visualizing and supporting the frequent interdepartmental collaboration at LAIKA.

2. Scheduling Engine Improvements
Second, there are some scheduling engine improvements we'd like to see.
We intend to continue working with Phil (and his new Autodesk team) to push Generative Scheduling further and are especially interested in how it could more closely match a human-intuitive schedule.

For example, an affinity function for tasks linked to the same build could help the scheduling engine keep assignees focused on a single build at a time, and reduce the overhead of context switches.

We'd also like to be able to provide users with fine-grained control over resource curves, so that different departments could scale in and out at different points in a project's timeline.

3. New Visualization Approaches
We'd like to continue to explore new strategies for visualizing schedule information, like network diagrams and kanban boards, to see if they can help us capture better actuals and streamline communication.

Conclusion

The introduction of our new scheduling workflow has made a big impact at LAIKA – changes are now significantly easier to accommodate, and there is increased confidence in the scheduling process.

Filmmaking at LAIKA happens over a very long timeframe, with each film taking several years to complete, so unfortunately we don't have a full sense of time or cost savings yet, but results so far have been quite promising, with users able to turn around new bids and resource lists faster than ever before.

Acknowledgements

Thanks for reading this document! Feel free to reach out if you have questions.

Lastly, none of this would have been possible without the efforts of a whole team of people. I'd like to thank the early adopters of this workflow for embracing the new workflow and providing feedback on every step of the process.

Early Adopters:
Carlyn Siegler
Marina Capizzi
Finley Mulligan
Derrick Huang

And also thank the folks involved in developing these workflows —

Jeff Stringer, who provided the resources and guidance for the new workflow.
Phil Peterson, for his ongoing development of the Generative Scheduling app.
Tony Aiello and Owen Nelson, for their foundational work on scheduling at LAIKA.
The LAIKA Shotgun Team (Ben Brandt, Emilee Chen, Paul Kubala, Daniel Pebly, and Ellen Duong)

Reference

Scheduling - xkcd.com/1658

The Process of Design Squiggle by Damien Newman, thedesignsquiggle.com

Wikimedia - Standard Oil Octopus
https://upload.wikimedia.org/wikipedia/commons/a/a0/Standard_oil_octopus_loc_color.jpg

Shotgun for Production Management in LAIKA's Animated Features (Autodesk University 2018) - Tony Aiello - https://www.autodesk.com/autodesk-university/class/Shotgun-Production-Management-LAIKAs-Animated-Features-2018

Stacked Area Graphs Are Not Your Friend - Myles Harrison
https://everydayanalytics.ca/2014/08/stacked-area-graphs-are-not-your-friend.html

Related Material:
Material I found insightful while developing this workflow.

The Design of Everyday Things - Don Norman
https://en.wikipedia.org/wiki/The_Design_of_Everyday_Things

Beautiful Evidence - Edward Tufte
https://www.edwardtufte.com/tufte/books_be

Project Management Graphics (or Gantt Charts) - Edward Tufte Message Board
https://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=000076

The Mythical Man-Month - Fred Brooks
https://en.wikipedia.org/wiki/The_Mythical_Man-Month

Choose Boring Technology - Dan McKinley
https://mcfunley.com/choose-boring-technology

Technology:
Links to some of the tech used in this project.

Flask
https://flask.palletsprojects.com/en/1.1.x/

Plotly.js
https://plotly.com/javascript/

Shotgun API
https://developer.shotgunsoftware.com/python-api/

Bryntum Gantt
https://www.bryntum.com/products/gantt/