

Unit 5: PROJECT ORGANISATION

VFX Production

Large-scale VFX projects may require collaboration between hundreds of different people over several different sites globally, which is the reason that production teams exist. Production hold the responsibility of managing artists, liaising with clients and creating & maintaining realistic work schedules, among other things.

One system that is popular among many VFX studios is *Shotgun*. Shotgun is primarily a project management system designed specifically for animation and VFX production, which includes an integrated asset management and versioning system. It's used by artists and production alike, with one of its major selling points being its all-in-one approach; Production can use it to easily set team milestones and distribute tasks, while artists can use it to keep track of their work and respond to all their latest notes.

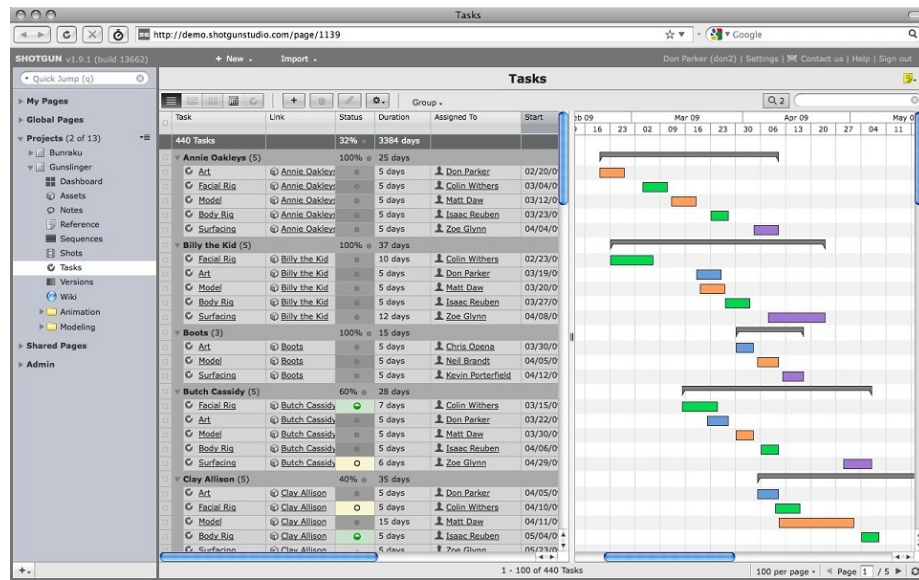


Figure 1: Shotgun tasks view

Version Control

An effective way to track the progress of something being developed is by using version control. This applies not only on the artist side for digital assets but to software packages and scripts also. In its most simplest form, this means saving your work into at sensible intervals and keeping a history of your previous work. This is useful for creative development as it allows an artist or a supervisor to

look at the development of an asset to ensure that it is heading in the right direction. Shotgun has a version control system built in, and is packaged with the review software RV, which allows you to compare and annotate different iterations of an artist's work.

On the technical side this method allows you to look at the history of a tool to help spot or prevent issues that in the development of it. Package management systems are often used for deploying tools and scripts to studios, which almost always will include its own versioning system. *As explained in Unit 8, Git is also very commonly used as well, as a version control system for of source code. The differentiation being that package management systems will version an entire package on build, while Git manages all code changes on any number files within the package, between and including package version iterations.*

Plugin/software Production

While the idea of an all-in-one pipeline solution sounds great, there is always room for development to meet the ever-changing demands of clients and to best suit the studio. For instance, often different clients will use different naming conventions they wish you to follow. Some studios have technical developers crewed to shows, who create and maintain custom tools to aid production. *Explained more in unit 1.*