THE HIDDEN PRODUCTIVITY COSTS DISRUPTING YOUR RELEASE TIMELINES

Unity experts share insights on systems and processes to help you streamline your development pipeline.

How many times in the past month have you had to drop what you're doing and spring into action to solve an urgent problem that needs a fix *right now*? One of the most common game development pitfalls is letting small, seemingly inconsequential inefficiencies snowball into bigger issues.

When your flow state is interrupted multiple times a day – whether it's waiting for a project file to download or solving yet another merge conflict – that's when things start falling through the cracks. Bugs begin to infiltrate your code. Deployment cycles lengthen. All-nighters become the norm. Friction builds with every slip-up, and soon it's felt across the team. Frustration grows, morale begins to falter, and it's clear that your studio's way of doing things *just doesn't work*.

Implementing DevOps solutions can help you break this chaotic cycle. With the right processes in place, you and everyone on your team can spend less time task-switching and get back the focus you need to produce your best work. Well-defined operating principles and the right set of tools improve collaboration and speed up deployment cycles. Launch polished, stable games on time and at quality, again and again – all while minimizing crunch.

Unity has worked with countless games developers and studios over the years, and many of the experts on our team are gamedevs themselves. In our experience, version control systems (VCS) are the foundation for effective DevOps, but they can also be one of the biggest pain points for studios. That's not a coincidence.

In this e-book, we've collected insights from our product and engineering teams on three of the most common version control pitfalls holding studios back from doing what they do best: creating. Read on for a look at the hidden productivity pitfalls disrupting your release timelines – and how VCS solutions like Plastic SCM can help you get back on track.

→ PROBLEM #1

PARALLEL DEVELOPMENT

PABLO SANTOS

SENIOR MANAGER, SOFTWARE ENGINEERING

As a cofounder of Codice Software, the company behind Plastic SCM, and former dean of the Colegio Profesional de Ingenieros en Informática de Castilla y León, Pablo has more than 20 years of experience in software development. Here, this Unity version control and DevOps expert shares his thoughts on parallel development and how version control can help accelerate collaboration.

→ THE SITUATION:

ONE PROJECT, MULTIPLE WORKFLOWS

Unless you're a team of one, you're going to have several team members working on the same project. Having extra hands can obviously help you scale at speed, but it can also slow things down.

Parallel development is when various team members are working in multiple branches, with the ability to merge them later. Coders typically work this way, while artists tend to work on a single branch with locks since they deal with individual files and assets.

→ THE COST:

TIME YOU'LL NEVER GET BACK

The impact on productivity here is obvious and immense: *Tons of time lost*.

The cost in productivity from having to redo work, solve merge conflicts, track down lost files, or wait to download changes you'd already committed before results in slower release cycles, internal conflicts between team members, and the inability to quickly fix bugs that users find in the game.

→ THE PROBLEM:

LOST WORK AND INTERNAL STRIFE

While there's an advantage to even a solo gamedev using a VCS, it's downright essential when working with multiple team members.

Not using any version control can quickly result in a situation where developers overwrite one another's code, resulting in lost work and a lot of frustration. However, even the use of common VCS tools today creates issues for game development teams.

For example, if the entire team is on Git, one of the most common scenarios is that some of the more non-technical team members, like artists, won't have a full understanding of the tool. It becomes very easy for them to lose changes, especially without the ability to lock files. And because the system isn't transparent and easy to follow, there's often an overreliance on developers to solve problems, which results in even more lost work.

Because of Git's inability to lock files, many gamedev teams use Perforce across the team to stick to one single system and source of truth. However, developers may be dissatisfied with the speed and performance of Perforce's "task streams" approach to branching and merging. This can add another layer of complexity to an already complex process, especially when setting up continuous integration. Most creators we talk with say they wouldn't even try task-based development with Perforce. It's just not an ideal workflow.

© 2021 Unity Technologies

THIS EXPERT RECOMMENDS

It’s important to establish a set of versioning best practices, create documentation, and update that documentation regularly.

Having clear rules for things like file organization and making sure they’re widely socialized keeps your team working in lockstep.

I also recommend choosing a VCS that works for *everyone* on your team. For example, artists should be able to contribute to the project independently, without relying on programmers. By implementing a version control system that’s easy for everyone to use, making changes to project files becomes less of an intimidating process.

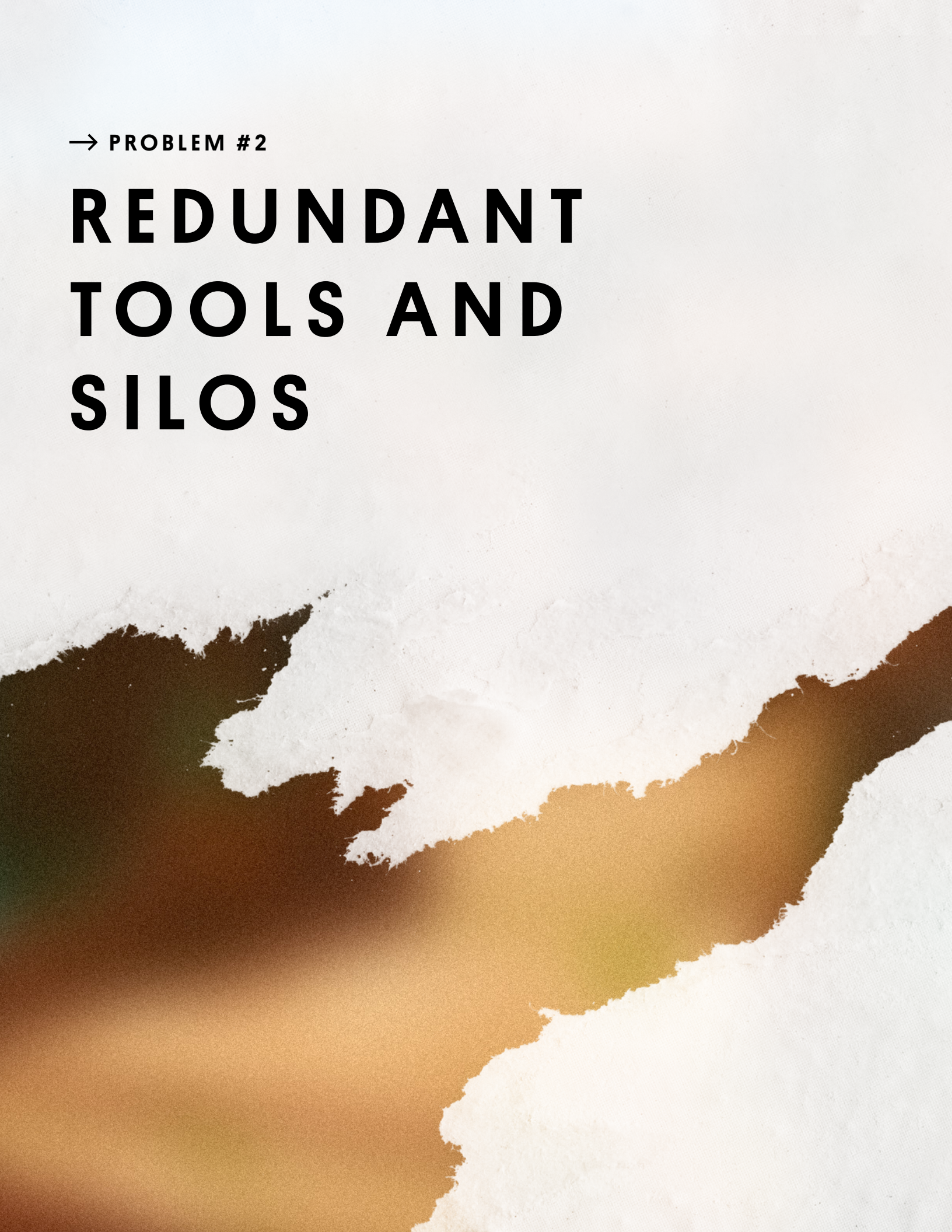
LEARN MORE

Montreal-based studio KO_OP dealt with asymmetry and miscommunications between artists and programmers while making their game *Goodbye Volcano High*. Learn how they solved progress-blocking issues by implementing the right VCS.

Goodbye Volcano High by KO_OP

→ PROBLEM #2

REDUNDANT TOOLS AND SILOS



STACEY HAFFNER

SENIOR MANAGER FOR DEVOPS

Stacey is an indie generalist developer who has operated a small studio since 2015. She formerly taught Unity online on Microsoft's Channel 9 (.GAME) and now runs gamedev-resources.com. Here, Stacey discusses version control tools and explains why streamlining your solution set is the best path forward.

→ **THE SITUATION:**

DIFFERENT ROLES, DIFFERENT TOOLS

Unless you're a team of one, you're going to have several team members working on the same project. Having extra hands can obviously help you scale at speed, but it can also slow things down.

Parallel development is when various team members are working in multiple branches, with the ability to merge them later. Coders typically work this way, while artists tend to work on a single branch with locks since they deal with individual files and assets.

→ **THE COST:**

TWICE THE WORK WITHOUT THE PAYOFF

In game development, every second is needed to rapidly iterate and test new ideas. While multiple version control tools are often adopted to solve time-wasting issues like syncing or merge conflicts, they can often compound these issues or create other drains. For example, they can increase the time spent creating new processes or workflows without bilateral integrations, or the financial cost of maintaining two systems.

→ **THE PROBLEM:**

MORE WORKFLOWS, MORE ISSUES, MORE MAINTENANCE

To explain further, a team will usually start off as either Perforce- or Git-based (e.g., GitLab/GitHub), and deal with the pains that each unique workflow creates. In Perforce, they lose the ability to work distributed and with ephemeral branches. With Git, they run into issues around file storage due to distributed work. Teams can use Git Large File Storage (LFS) to ensure only one version of the binary is stored on the machine, but this comes at the cost of speed and a confusing workflow that pretends to be centralized but isn't.

There are two common problems that studios will encounter when using multiple solutions. The first is maintaining two types of version control technologies for the project – which usually includes cumbersome workflows or processes for integrating the two. The second is centered around knowledge and training. With two very different systems, you need experts in both to help unblock a creator or entire team when something goes wrong.

© 2021 Unity Technologies

THIS EXPERT RECOMMENDS

Ideally, you would want all assets managed and versioned in the same version control system, with workflows tailored for the different needs of your team.

It's important to examine your processes and tooling, then decide whether they're really working for your studio.

This would allow your studio to avoid a lot of these common headaches around having to create disparate pipelines to transfer assets from team to team.

We recommend a single VCS that can accommodate multiple workflows or has robust integrations between systems, to allow for tracking who is working on what and the latest versions of each asset.

LEARN MORE

Learn more about what to consider when evaluating version control systems.

Crying Suns by Alt Shift

→ PROBLEM #3

TOO BIG TO COMMIT

MARIE-CHRISTINE BABIN

LEAD PRODUCT MANAGER FOR VERSION CONTROL

Marie-Christine worked in game and interactive attraction development for eight years before joining Unity in 2019. As lead product manager for version control, she's helping to craft the vision for how version control can help create a collaborative environment that's inclusive of all team members, regardless of their role. Here, she shares her thoughts on a silent killer of productivity: time spent waiting to download project files.

→ **THE SITUATION:**

PROJECT FILES NEED FREQUENT UPDATES

Working with large files and binaries is one of the unique elements of building a game – but it can cause issues with version control. Let's say you're a developer working with Git, and you're trying to check in your code. To do that, you first need to update your workspace with the latest changes from other team members (this is the equivalent of pulling and pushing).

→ **THE COST:**

EVEN MORE TIME WASTED

Time spent waiting for updates to download is time wasted. That time adds up. A faster system allows your teams to focus on completing their tasks without the tools bringing projects to a standstill.

→ **THE PROBLEM:**

WORKFLOW DISRUPTIONS

Some version control systems are extremely slow when it comes to working with large binaries. Imagine how frustrating it is when you have to wait five minutes between the moment you click update and the moment you check in and can finally move on with your life.

In gamedev, you need to move fast. If it takes too long to implement changes in the game or respond to bugs due to inefficiencies at the very start of the pipeline, these problems will only compound later on.

Many studios use solutions like Perforce or Git LFS to solve issues with large binary assets, but each comes with the same compromises outlined to the left.

THIS EXPERT RECOMMENDS

Choosing a file-based VCS can speed up work a lot.

With a system like this, contributors can download only the files they need to make changes to, instead of the entire project. It's much faster, and more secure, but you have to ensure that everyone is organizing their files in the same way for it to be truly effective.

Think ahead when choosing your version control solution, as it can save you a lot of headaches later on. Games are getting bigger and more complicated – the sooner you implement the right version control system, the better set up you'll be for lasting success.

LEARN MORE

Learn about Sycoforge's approach to iterative development with their game Return to Nangrim and how Unity Plastic SCM helped them adjust to their project's growing scope.

Return to Nangrim by Sycoforge

PLASTIC SCM: VERSION CONTROL FOR GAMEDEV

Version control is foundational for DevOps, but it's also a central source of conflict for game studios, and there aren't many solutions that truly work for every contributor. Plastic SCM is uniquely positioned to support game developers precisely because it solves these time-wasting issues. Here's how it stacks up against the competition.

© 2021 Unity Technologies

→ PARALLEL DEVELOPMENT

Effectively facilitating parallel development is a must, and each member of the team might have different preferences. Git-based tools such as GitHub, GitLab, or BitBucket are some of the most popular version control systems around, especially with developers. Due to Git's popularity, users will have access to a robust amount of user enhancements, integrations and interoperability. However, as team and project size scales, many larger and AAA game studios use a centralized workflow solution like Perforce to better accommodate artists and non-technical users while handling large binary assets.

Due to its robust branching and merging capabilities for developers and the ability to work centralized and lock files for artists, a system like Unity's Plastic SCM can help. If you're working in the Unity Editor, Plastic SCM's seamless integration with the Unity UI encourages collaboration.

→ REDUNDANT TOOLS AND SILOS

While artists and programmers may generally have different preferences for how they use version control, it's important to know the pros and cons of over-correcting in either direction. Especially when working remotely, having a single version control system and "source of truth" can help to eliminate many of the frustrations and time-wasting consequences of maintaining multiple different pipelines.

Even with their various strengths and weaknesses, a lot of systems such as GitHub, Perforce, or even Dropbox or Google Drive can be set up to suit your workflow. Look for integrations or extensions to help cut down on some of the collaboration issues, or opt for a single system that's designed for all parties.

Plastic SCM supports working with a single repository in different modes. This feature gives you the power of multiple workflows without the hassle of maintaining and connecting two different systems.

In Plastic SCM, an artist can use the Gluon UI to work with the repository in centralized mode, thereby simulating the same style of working with a solution like Perforce. A developer can use the default UI to work with the repository as in a distributed mode, thereby simulating the same workflow as a Git-based solution like GitHub or GitLab.

© 2021 Unity Technologies

→ TOO BIG TO COMMIT

Game development means working with large projects. As your studio scales, the source control management system should be able to scale as well.

While Git's distributed model was not designed for a game development use case, there are several extensions designed to allow Git to handle binary files with greater ease, Git-LFS being the most widely used. Still, Git was not designed with managing art assets in mind, which is why other solutions like Perforce are often implemented at larger studios.

Plastic SCM is the most performant solution on the market when it comes to handling large files, which is ideal for game development. It significantly reduces the downtime waiting for files to load – Plastic SCM can be up to 5–8 times faster than other solutions. You're still dealing with the same content, but now you aren't waiting on a five-minute download multiple times a day. Instead, you're getting what you need in seconds.

CREATE WITHOUT COMPROMISE

Plastic SCM is specifically designed for game development, supporting artists and developers in studios of all sizes. It empowers artists with file-based workflows and an intuitive UI, as well as including robust branching and merging for programmers to ensure that all users benefit from improved collaboration and iteration agility.

Plastic centralizes work without cloned repositories, and it supports huge files and repositories with fast, WAN-optimized data transfers. Its Gluon workflow tool lets the least technically inclined team members easily apply any version control protocols.

Speed up your workflows and build out your DevOps toolchain with Unity Plastic SCM. Create without compromise, and work efficiently to get to market on time, at the quality you expect. Get started free today.


Unity®