

# SOURCE AND VERSION CONTROL IN DEVOPS



DevOps teams are designed to embrace the tenants of Agile while utilizing advanced tools for increasing productivity and deployment rates. Version control, also known as source control, is one of the key tools used by successful DevOps teams for reducing development time and increasing the rate of successful deployments. Version control helps software engineering teams to collaborate at the speeds required by today's constantly shifting IT environment.

Strong version control systems that are leveraged to their optimum output make all the difference when it comes to DevOps success. For clarity's sake, let's briefly outline the basics of version control.

*(This article is part of our [DevOps Guide](#). Use the right-hand menu to navigate.)*

## What is Version Control?

Version control is called by various monikers such as source control, version history, and VCS (Version Control Systems). Regardless of what you call it, version control is any system or tool used to store and track changes to projects over time. If you have ever used file names like *DevOpsOnesheetFinalVersionMK2.pdf*, you have technically made use of a version control system (though an awful one).

GitHub is a well-known example of a version control tool and has become somewhat of an industry standard especially when concerning open source projects. Version control tools like GitHub keep track of every change that's made to a project over time and makes a note of when the changes were made, what the changes were, and who made them. Most version control tools will also allow the person submitting their changes to provide annotations indicating why the changes were made - providing invaluable context for whoever views these new changes.

Version control systems also support the ability to make "branches" of the main project "trunk" to allow for changes to be tested without altering the original project. Once these branches have been successfully tested and verified, they can then be "merged" with the main project. Using version control like this allows for multiple coders to simultaneously work on the same project without creating dependency issues or interrupting each other's workflow.

Sounds pretty handy, right? The benefits of utilizing strong version control tools are such that most would consider version control mandatory for engineering teams of all sizes.

## **Why is Version Control Necessary?**

Version control plays a core role in the success of DevOps teams. According to the 2014 annual State of DevOps study, version control has consistently been one of the best indicators of overall software engineering performance. There are numerous reasons for this direct correlation between finely tuned version control practices and successful rapid deployment in IT departments.

### **Faster and Simplified Bug Squashing**

As anyone who has ever done any amount of coding knows, bug hunting and squashing take up what feels like 95% of the time spent coding. The massive amount of time and frustration that bugs can cause is notorious among software engineers. Thanks to the constant and detailed tracking of changes made to projects, version control provide immediate insight into where things might have gone wrong.

Sometimes bugs can take longer to find that the effort is worth. Thankfully, version control can be invaluable in these cases as well. Not only does version control give you a good idea of when and where the bug was introduced, but it can also be used to instantly revert to previous functional versions of the project. This means when all else fails, reversion can be used to ensure there is a viable launching point for the new features to be developed again without having to scrap the whole project and without losing too much work.

### **Simultaneous Development**

Thanks to the nature of programming, working alongside other coders on the same project can be a real nightmare at times. Dependencies (when bits of the program rely on other bits of the program to function properly) can create massive headaches when changes are made unbeknownst to each team member. Even though linking up with the new changes can often be a fairly simple process, not having knowledge of those changes often results in new bugs being created despite the fact that there are no actual errors in the code itself.

With version control tools, branches are created to ensure the changes one engineer makes isn't being impacted by another programmer's work at that time. Additionally, version control systems allow for the ability to provide notes alongside each change made. When it comes time to merge the work together, integration is simplified thanks to the well-documented changes each engineer made.

### **Increased Reliability of Final Product**

Version control has an immediate impact on the quality and reliability of each deployment. Version

control systems make working with others a much simpler and rewarding process due to the increased visibility of changes as well as the context provided for each modification. Communication is the most important aspect of working as a team and version control enhances communication without requiring teammates to spend all day writing memos to each other.

The visibility of changes and the context provided alongside these changes make it much easier for everyone on the project to get a handle on what happened and why so they can all come together to stay on target and continue to produce results. This kind of empowered communication results in better products being delivered in less time all while increasing team morale.

The symbiotic relationship between version control and top-notch software engineering performance is readily evident. Changes can be made at a much faster pace and with increasing levels of reliability when version control systems are properly leveraged by the team. The purpose of DevOps is to speed up the entire production process while also increasing the quality of work performed. Version control provides immense aid in bringing these goals to reality by empowering team communication and product development success.

## DevOps: Solutions for You

If DevOps sounds like a good fit for your organization's needs but you want to make sure you get all the details right when it comes to implementation and version control, BMC is the IT solution partner you need. Read more about how automation and DevOps systems can help increase the rate at which you deploy products with BMC's free eBook: [Automate Cloud and DevOps Initiatives](#).

BMC expert consultants are available to work with you to bring their knowledge and expertise to your organization. [BMC Consulting Services](#) are available for answering any questions you might have regarding the DevOps system and its implementation. We also provide custom tailored [Deployment Services](#) for your organization to tackle the unique challenges you face. When partnering with BMC, you get:

- **Faster service delivery:** Agile releases that keep up with rapid demand
- **Visibility across data:** Ensure compliance and data accuracy
- **Cost-effective service:** Increased productivity and performance
- **Experienced DevOps professionals:** Equip you with the tools you need for success
- **Conversion or upgrade:** Seamless modernization or total replacement
- ***All tailored for the specific needs of your organization.***

Download or view the [Solution Implementation Overview](#) online to learn more about how [BMC Consulting Services](#) can help you. Learn more about how [DevOps and Application Deployment](#) best practices can enable your teams to create better software faster than ever before. Then contact the experts at BMC to learn more about how to incorporate version control systems into your DevOps practices for enhanced building, testing, and deployment success.