



WHITEPAPER

TAKE THE PRODUCTIVITY OF YOUR IT TEAM TO THE NEXT LEVEL: WITH **CLOUD DEVELOPMENT ENVIRONMENTS**

Key Facts

- 61% of developers surveyed report that they have to fix problems in their local development environment at least once a month. 31% even have to fix problems at least once a week. This leads to productivity losses.
- Standardisation guidelines help, but individual problems with local development environments still occur. The required performance is not always available for computationally intensive tasks. Source code is stored insecurely on the computer. Branch switching costs time.
- Cloud Development Environments (CDEs) are a solution for this problem. CDEs are standardised remote environments in which developers have all the tools they need for their tasks at their fingertips.

Introduction

Finding software developers is hard. Keeping them isn't easy.

These two factors lead to two considerations:

- The productivity of their own developers must be optimised.
- The developer experience must be improved. Developers must have access to tools and environments that they like to use.

A major obstacle to achieve these goals are local development environments. Hours are spent on setting them up and maintaining them. In particular, operating the software to be developed locally and the local build are a major challenge.

Cloudomation drastically reduces this effort: With standardised development environments, so-called Cloud Development Environments (CDEs). CDEs are immediately available to all developers and can be individually configured.

This whitepaper describes, which problems Cloud Development Environments solve, which advantages CDEs bring, how you can use CDEs to increase the productivity and how to improve the quality of software development in your company.

Local Development Environments

Normally, software developers work in their local development environment. Since every local development environment is set up differently and is not standardised, individual problems arise that are time-consuming and difficult to solve.

In addition to the IDE and source control, which are usually set up quickly, many dependencies have to be set up and kept up to date, getting local build tools up and running and usually the most challenging: to run the software to be developed locally. Self-developed tools and scripts are often used, which are error-prone and difficult to use.

Elijah Bate of theoptimisedweb.com describes it like this:

„Setting-up new dev environments or trying to refresh your local machine for different projects when configurations are different honestly sucks. The amount of time other developers and I have spent trying to debug nodejs version issues, installing and uninstalling different SDKs, and finding and setting up environment variables and secrets, is just insane.“

The effort for maintaining the local development environment is vast. A survey among Gitlab users confirms this: 61 % of the developers surveyed report, that they spend time troubleshooting their local development environment at least once a month. 31 % even at least once a week. The constant troubleshooting and dealing with the development environment, e. g. when a bug in an old version has to be fixed, leads to significant productivity losses and frustration.

Where are the tools that allow software developers to work productively at all times?

Cloud Development Environments

CDEs are remote environments in which developers have all the tools they need for their work. An CDE is therefore a tool that enables remote development and makes it easier. CDEs claim to make local development environments obsolete.

„Remote“ means that developers don't have the software to be developed available locally on their laptop or computer and start the build process there, but connect to another environment that takes over. This can be in the employer's data centre, or in the employer's private cloud, or a SaaS offering. Different CDEs offer different options here.

Read more:

[Cloud Development Environments: The ultimate Guide](#)

Benefits

This section provides an overview of how CDEs can help companies to take the productivity and developer experience to the next level.

Standardisation

One problem with local development environments is that each one is different.

Even with strict device and software policies, the possible disruptive factors can be limited, but never completely eliminated. Nevertheless, individual problems arise.

This is particularly frustrating when you consider how time-consuming it is to develop standardisation guidelines and continuously enforce them. It often creates a lot of resistance and frustration among developers, because they have to work with software they don't like.

How would it be if developers had completely standardised development environments available at the push of a button, which were created anew each time and corresponded 100 % exactly to the policies? And next to that, developers can be given a free hand as what they use locally for the operating system, browser and other tools?

This is possible with CDEs. The environments can also be configured in such a way that they come as close as possible to the production environment of the software to be developed (e. g. running a Kubernetes cluster on a CDE also becomes feasible, which can lead to a lot of headaches if you do that locally). This reduces the „works on my machine“ problem. And leads to a significantly higher developer experience at the same time.

Time saving

Joel Spolsky writes: "Top notch development teams don't torture their programmers. Even minor frustrations caused by using underpowered tools add up, making programmers grumpy and unhappy. And a grumpy programmer is an unproductive programmer."

Business software development means that computationally intensive work has to be done. This often leads to frustration among developers because the necessary equipment isn't available. For example, when the RAM almost bursts into flames due to a tool and waiting for tests etc. to finish. And management? Management is also frustrated about the cost of providing equipment.

Because CDEs take over performance-intensive processes and take place on a remote server, the choice of working device can be more economical and reduce costs for companies. With CDEs, however, developers have all – and even more – resources available that they need.

Gone are the days when developers wait forever for the local build to finish.

Security

In times of remote work, supervisors sometimes don't know exactly where their developers are.

Collaboration with freelancers around the world is becoming more and more normal. And with that, the issue of source code and data security is becoming more urgent.

What happens if the laptop is left behind in a taxi in Delhi? Or what happens if the colleagues in the USA don't have their own lockable office in the co-working space, but simply leave their work equipment open overnight?

Once the end device is lost, access to remote resources can be blocked – but what is on the laptop is lost forever.

A headache category of its own: access to personal data necessary for testing the software. What is the best way to handle this, legally and also practically, when the team is scattered all over the world?

Measures such as encrypting hard disks, policies for using trusted networks, disabling USB ports, working via VPNs – many possible solutions exist. All of them are problematic because each one involves significant effort, is difficult to enforce and yet only partially solves the problem. What to do?

You can guess the answer: CDEs can be fully air-gapped so that source code never ends up locally on the developers' computers. An Air-Gap is used where the highest security requirements are needed. It's an environment that is isolated from other devices and from the Internet. Information can be exchanged within the network, but it never leaves it. Test data can be used in CDEs, but shielded so that developers don't have direct access.

For developers, working in the CDE is comfortable, so there are no incentives for misuse, which often takes place in the service of productivity and with the best intentions. And since source code and data are never stored locally, the risk of loss or theft of the device is zero.

Productivity

All developers know it: Changing branches in the local development environment or even going back to an old version of the software to fix or check a bug is tedious.

Everything has to fit together: all dependencies have to be in the right version, all components of the software have to work together - and who knows exactly which versions were compatible?

After the bug fix, the whole thing goes in reverse: pull your own latest feature branch again, update dependencies, and hope that the customer who is still using the old version won't find a bug again any time soon. All of which have already been fixed in the new version anyway.

The solution? Sure, forcing all customers to always use the latest version. Unfortunately, this is often not possible in reality. And even without customer bug fixes in old versions, switching branches is often necessary, it's a part of normal development work. So the solution can't be to make branch switching obsolete. A nicer solution: CDEs.

CDEs are available ready-to-use in all required versions. If required, CDEs can be created not only for major versions, but also for specific commit hashes of individual branches. Each developer can run several of them in parallel. This suddenly makes working in different branches child's play.

This not only increases productivity and reduces frustration, but also leads to significantly better customer support. Because if it's easy, you're much more likely to help.

Key Points

- Setting up the local development environment takes time. The maintenance requires continuous effort. 61% of surveyed developers report that they have to fix problems at least once a month. 31% even at least once a week. This leads to productivity losses.
- Standardisation guidelines help, but individual problems still arise with local development environments. For developers, standardisation can be frustrating, because tools they love can no longer be used.
- Computationally intensive tasks require equipment that provides the necessary performance. However, the appropriate equipment isn't always available (often also for cost reasons). This leads to frustration among developers.
- The source code is on the developer's computer. If the device is lost or stolen, the existence of the company is at stake.
- Branch switching takes time. Dependencies must be in the correct version, all components must fit together. The whole process takes time.
- The solution are Cloud Development Environments (CDEs). CDEs are remote environments in which developers have all the tools they need for their work at their fingertips. CDEs aim to make local development environments obsolete, standardise them company-wide, come close to the production environment, are available at the push of a button and give developers a free hand in choosing their local tools. CDEs take over performance-intensive processes - developers can theoretically work on any device. Costs for expensive tools are reduced. The development environments can be set up air-gapped so that the source code never ends up locally on the computer. With CDEs, all versions of the software are immediately available. All you have to do is call up the right CDE.

**PUT THE EFFICIENCY OF YOUR
CURRENT DEVELOPMENT ENVIRONMENT
TO THE TEST AND START WITH CLOUD
DEVELOPMENT ENVIRONMENTS FROM
CLOUDOMATION**

[Read more](#)

[Contact us](#)



A product of Starflows OG
Darnautgasse 6/6, 1120 Vienna, Austria

+43 699 10 69 5915
www.cludomation.com
info@cludomation.com



Supported by the Vienna Business Agency.
A fund of the City of Vienna.