# Are Developers Really In Charge Of Security? What We've Learnt So Far

Naturally every company has its own individual path to follow - however it is essential all organisations understand that the roles of their security and dev teams are changing.

Most companies now fully embrace Agile methodologies, Lean Development, and DevOps practices. The software development cycle is almost universally shorter, with most development teams expected to work to a fast-paced, continuous delivery schedule.

From a developer's point of view, continuous deployments are - quite literally - continuous. According to security teams, however, devs are reluctant to take responsibility for any bugs that arise - either because they're too busy - or don't take security seriously enough. Is this a fair assumption? GitLab asked developers for their opinions.

## GitLab's 4 main developer takeaways

The recent GitLab analysis reveals four main takeaways for developer teams:

- Continuous Development (CD) really is - well real!
- Most developers feel they're solely responsible for security - and find this unacceptable
- Development teams are demanding ( and often failing to acquire) the collaboration and design tools they need in order to remove the barriers to DevSecOps
- Delays to correct DevSec processes continue to exist. Testing - along with code reviews and code development ( for the second year running) - are continuing to prevent processes from running smoothly

# Where are we in terms of fixing problems for developers?

Developers are the ones who benefit most from DevOps, due to increased code quality, improved time to market, less manual testing and better collaboration.

According to the GitLab survey, 60% of developers said their organisations deploy multiple times. Planning, testing, code reviews and code developments are big issues and are the main causes behind delays or even complete process stoppages.

The majority of developers say code reviews are very important and valuable for improving security and code quality. They also complain that code reviews occur daily -  and that there are no shared best practices. In GitLab's report they expressed the following views:

## "Every developer has to explain what he/she did and how they achieved this."

## "Code reviews take time due to the lack of reviewers and automated tools in use."

So code reviews are difficult. Security, however, appears to be just as problematic, with devs feeling overburdened with responsibility. The following statement illustrates a common frustration:

## "I actively advocate for security best practices but it falls on deaf ears."

GitLab’s survey verifies the findings of our own report (Clayton 2020) which revealed that, incredibly, application security still appears to be a ‘work in progress’ in many organisations.

Security is cross-functional in teams, and requires close collaboration with developers - nonetheless it seems as though security guys and devs can’t be friends!

And when we say “everyone is responsible for security” it usually turns out ‘no-one is responsible’, so clarity is needed.

```
efaultProps = {
 'default',
deAvatar: false,


serDetailsCardOnHover = showOnHover(UserDetailsCard);

serLink = ({
,
,
ndaryLink,
dren,
udeAvatar,
ne,
(
n className={styles.container}>

includeAvatar && (
  <UserDetailsCardOnHover
    user={user}
    delay={CARD_HOVER_DELAY}
    wrapperClassName={styles.avatarContainer}
  >
    <Avatar user={user} />
  </UserDetailsCardOnHover>
)


div
 className={classNames(
   styles.linkContainer,
   inline && styles.inlineContainer
 )}
>
 <UserDetailsCardOnHover user={user} delay={CARD_HOVER_DELAY}>
   <Link
     to={{ pathname: buildUserUrl(user) }}
     className={classNames(styles.name, {
       [styles.alt]: type === 'alt',
       [styles.centerName]: !secondaryLink,
       [styles.inlineLink]: inline,
     })}
   >
     {children || user.name}
   </Link>

   {!secondaryLink
     ? null
     : <a
       href={secondaryLink.href}
       className={classNames(styles.name, {
         [styles.alt]: type === 'alt',
         [styles.secondaryLink]: secondaryLink,
```

# Clayton’s report revealed a lack in standardisation

In our 2020 report we found that, although security is brought into the development process earlier, only 24% of companies said they have static application security testing. We also revealed that:

- 60% of developers do not run SAST scans
- Security teams do not have standard and best practices in place
- Security teams complain that devs find only 25% of bugs and vulnerabilities
- Three-quarters of bugs are nasty surprises that persist into a later stage in the process, at which point it’s harder to find and fix vulnerabilities

Devs are busy doing deployments, so, unsurprisingly, they’re often unable to prioritise bug fixing. Clearly, the situation is still far from ideal. But having a good understanding of what needs to be fixed is a good starting point.

Security must involve both good practices and ownership with DevSecOps the way to close the existing loops in the Developer ecosystem. DevSecOps helps organisations to achieve better code quality, with less manual testing, and above all, create faster deployments.