GDPR and Privacy by Design: What Should Software Engineers Know About GDPR?

Devolutions

THIS ARTICLE WRITTEN BY JULIAN HOOKS, BUSINESS WRITER, IS PART OF OUR GUEST BLOG SERIES.

PLEASE CONTACT US IF YOU WOULD LIKE TO BE FEATURED ON OUR BLOG.

As a software engineer, you have likely heard mutterings about the General Data Protection Regulation (GDPR) brought in by the European Union.

Sound confusing ? Let's run through a few of the facts about GDPR and why it matters for a software developer.

1. Right to access

First off, as explained in betipy's infographic, **those who take part in working with GDPR will need to appreciate that right to access data** is now a proven customer right. You should therefore look to make sure that any software you write or manage makes it easy for data to be accessed or provided to a customer, should they ask.

Until GDPR, this was never a major industry requirement. That, though, is about to change drastically.

2. Right to rectification

Users will also have the right to rectify all information. Make sure that all back-end software development makes it easy to access this information and to rectify the problems which exist.

The right to rectifying your issues with data is now part of GDPR, and you can expect errors to be made. As a software engineer, **your best bet is to ensure that all software developed is easy to access, so data can be quickly edited and changed**.

You should look to make sure that real-time access is offered, as users will prefer to make immediate changes instead of waiting numerous days or weeks for in-house staff to make the change.

3. Right to erasure

The other major right provided by GDPR is that users can now ask you to erase all data about them in a timely manner. The easier you make it for your clients to restore and remove data, the easier it will be for them to comply with GDPR. Keep that in mind, and you should be much more likely to help people comply accordingly.

All data must be removed if it is no longer needed for the original purpose, it was put through unlawfully and/or it has no real legitimate reason for staying on your systems. **By making it easier for your team to remove data, you'll make it much easier to solve issues with data usage**.

4. Designing for privacy

In the future, you should look to ensure that you design all of your new software builds with privacy very much in mind. If you do this, you are much more likely to avoid any user issues, while helping your clients get the most out of any software you develop.

We recommend making some smart choices based on the privacy requirements of your client. Revisit things like customer logging and user data entry, and ensure that all parts of the software match up with GDPR. **Ask yourself the following questions as you go:**

- Does this system contain private or sensitive information? If so, does it need to ?
- Is there anything within the system that might not be private, but could be otherwise compromising to someone ? If so, why ?
- What would be the risk to your client(s) if the database were accessed ?
- What size of database have you created ?

By following the above, you should be able to create clear and full audit trails. Make sure that you create a system which is extremely well protected to minimize the damage done in a data breach.

Better forensics allows you to ensure that your software meets GDPR regulations, and it helps to minimize data loss or privacy invasion in the event of a data breach. Your reputation as a software engineer today relies not just upon good software development, but best practices for privacy control.

Keep in mind that any software you create today must have privacy at the forefront, as it will play a leading role in how your business operates for years to come.