

## How to Attract More Investments With the SSDLC Approach

Engaging in secure software development not only safeguards your software but also cultivates a strong reputation for your company. Investors will be more inclined to consider you as a potential partner when they are confident that your product ensures data security and aligns with industry standards.

### Introduction to Secure Software Development Lifecycle

SDLC stands for software development life cycle. It is a set of successive steps used to plan, build, and deploy software programs. You can think of it as a detailed plan that describes tools and means for the project implementation. The downside of SDLC is that it does not embed the security level in its workflow. Such a situation may lead to potential bugs and

### We value your privacy

We use cookies to enhance your browsing experience, serve personalized ads or content, and analyze our traffic. By clicking "Accept All", you consent to our use of cookies.

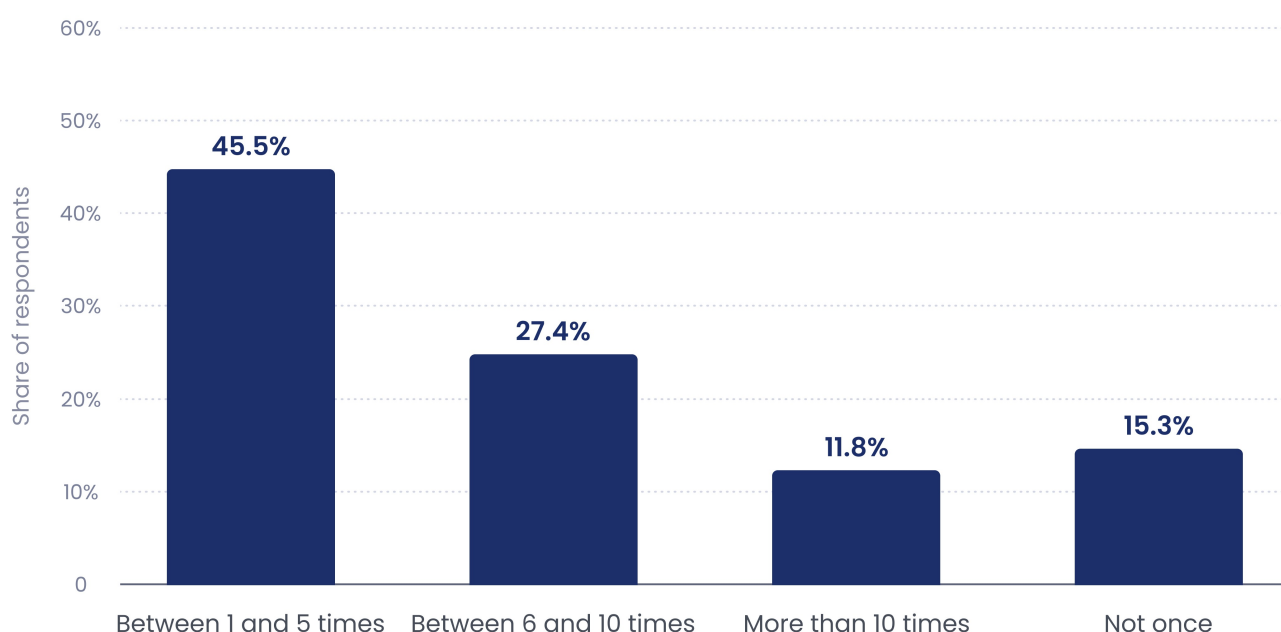
Accept All

Reject All

Customize

## Increased and more sophisticated hacker attacks

Tech innovations are constantly evolving and penetrating more and more spheres of everyday life. Unfortunately, as technology grows, cybercriminals master their skills as well. Hacker attacks are becoming more advanced, and the number of victims is constantly increasing. According to the latest data, almost half of the surveyed companies worldwide confirm their global network has been compromised by hacker attacks from one to five times in the past 12 months. 15% have experienced attacks more than ten times during the year.

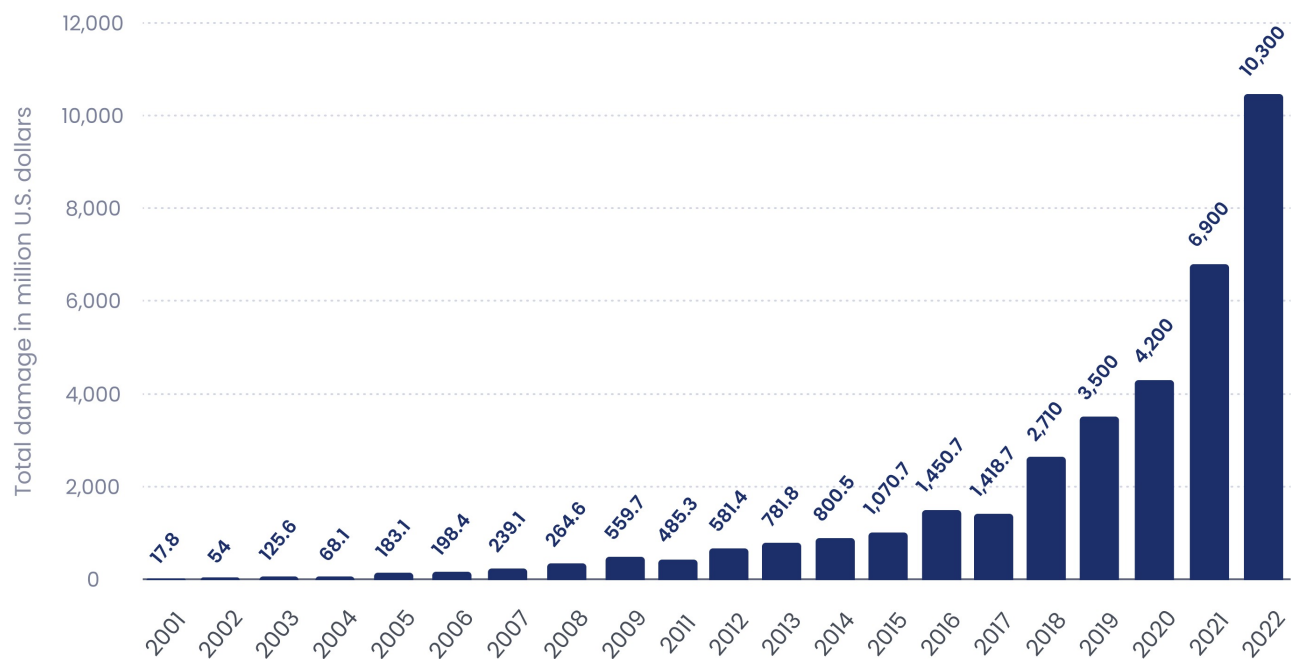


[Anyone](#) from a financial corporation to a travel agency can be targeted by cyberattacks today, and both local and cloud services are exposed to malicious intrusion. This situation is alarming for both users and owners. While the former worry about their data, the latter are concerned about reputation. Their fears are well-founded: a consumer survey from [Ping Identity](#) shows that customers quickly lose trust in compromised brands. 78% of respondents said they would most likely stop using an attacked service, and 36% of users admitted they would leave such a service forever.

## High probability of suffering losses

Probable financial failure is one of the main reasons why companies are concerned about security. Money losses from cyberattacks are now snowballing. In 2020, the total amount of damage reached \$4 billion, which is \$700 million more than the previous year.

## Annual amount of monetary damage caused by reported cybercrime in the United States



Financial ruin is a real threat to both large and small companies. It influences huge giants like [Sony Pictures](#), [WhatsApp](#), [eBay](#), and small startups that have just started their software development path. Besides money losses, security holes lead to many other negative effects that hinder the success of a business. The most significant harms manifest themselves in:

- violated intellectual property rights
- destruction of business processes
- destroyed reputation
- frustration of clients
- legal proceedings

### Issues at the legislative level

This is especially relevant for fintech and healthcare apps. Even though they go through many checks and compliance procedures, there is always a chance of malicious break-ins and data theft. If hackers access money accounts, thousands of clients will suffer losses, and the company will be obliged to compensate for the damage caused. With healthcare apps, things are even worse because they handle sensitive data about the state of health, and it is hard to predict how intruders may want to use it. In any case, if such data is leaked, users can go to court, and the amount of moral damage will most likely be assessed as high. A company facing this problem could be dragged into life-long litigation and sentenced to hefty

payments and fines.

Companies operating in other fields must also remain vigilant and treat personal data as paramount. A shining example of malicious imprudence may be the American retail store Target, which was hacked in 2014. As a result, cybercriminals accessed more than 70 million credit card records. The business owners were involved in lengthy lawsuits and consequently lost about \$10 million.

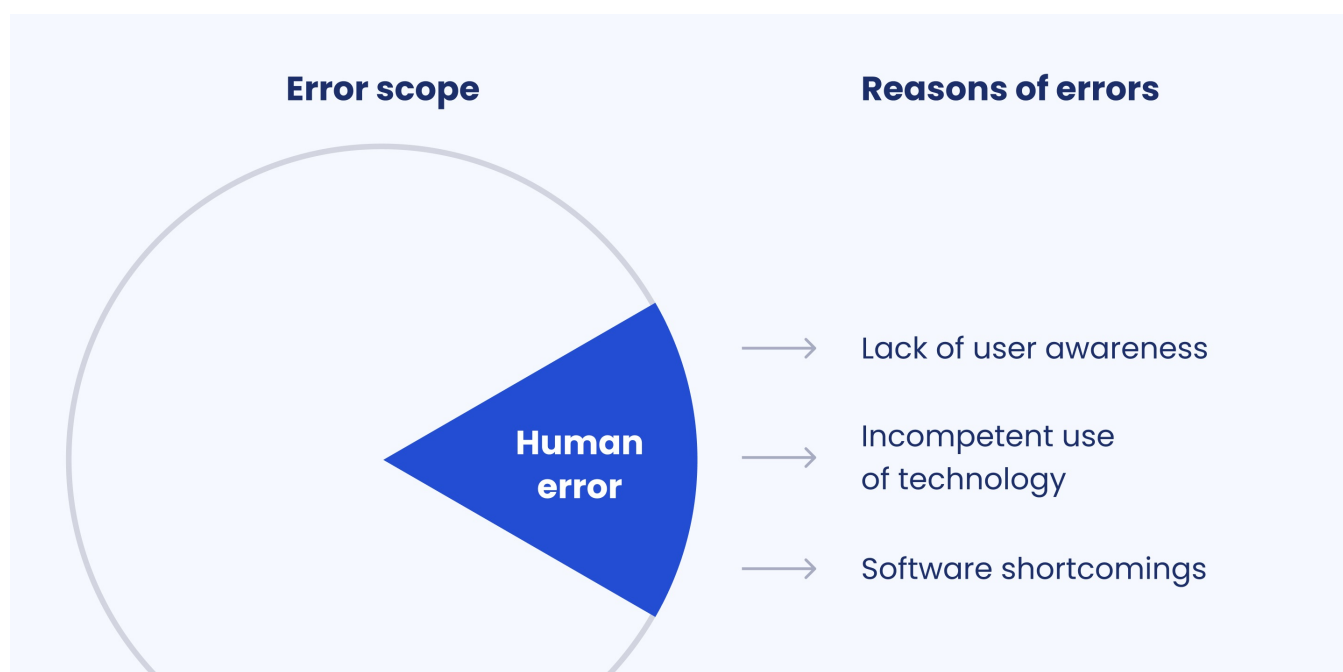
### Business dependence on corporate systems


In today's world, no one does business on paper. All data and processes are handled by corporate systems: ERP, CRM, accounting tools, etc. Unauthorized access to this kind of software and data leakage often means a partial or complete stop of the company's workflow. As a result, the owners bear enormous money losses, face damaged customer relationships, and invest in business recovery.

A wise choice of a corporate system may help avoid problems. If you want to use off-the-shelf software, ensure the provider pays enough attention to security. For that, study their documentation, consult technical experts, and read reviews. If you are going to order a custom solution, agree with the contractors on all the program implementation steps and make sure it proceeds according to the secure software development standards.

### The presence of the human factor

According to [IBM](#), human errors cause nearly a quarter of data breaches. The main reasons for this are lack of user awareness, incompetent use of technology, and, of course, software shortcomings.





User education and motivation can correct the first two points, while secure SDLC techniques will take care of the third one. Reliable software weakens harmful human impact by providing robust protection mechanisms and special defensive tools. Maximum automation, different access roles, and a properly designed architecture help minimize the possibility of data leaks. Besides, organizations must develop and maintain a culture that values positive safety behavior. They need to convey to users that security begins and ends with every person associated with their infrastructure, business, and services.

What effect does a data breach have on companies?

Any information leak causes a negative economic effect on the company. For joint-stock firms, it can also mean a decline in a share price.

---

## Data breach influence on strong brands and startups

So, let's imagine that a big name like Facebook or Instagram faces a data breach. Should we expect a negative reaction from the public, which will reduce the share price? Definitely, yes. Users will be unhappy with the fact that a world-known company was unable to withstand a hacker attack. Some of them will decide to stop using the service or even appeal to a court.

But there is also a reverse side of the coin, which comes from brand loyalty. Let's go back to the Facebook example. It has faced social media data breaches several times. Last time [533 million](#) entries with phone numbers and addresses were hacked. However, this situation had little effect on their share price due to:

- high user loyalty
- stable revenue
- embedded risk in the share price

It follows from the above that promoted companies can benefit from trust credit. Users who are addicted to a company's services may forgive some accidents and continue using the platform.

It is worth noting that very few organizations achieve such a level of trust. In most cases, firms should prepare for a vast customer outflow, leading to decreased revenue and falling stock prices.

Also, startups are in a more precarious position compared to mature companies. They don't have a large customer base, strong reputation, and stable income. If a data breach occurs at the beginning of their path, it will likely discourage users and may put back further business development. In such circumstances, the share prices are likely to fall, increasing the chances of company bankruptcy.

---

## Why SSDLC?

SSDLC is a process for building secure software systems. It is based on typical stages of software development but describes how to integrate security into the well-oiled procedure. Today, only a few companies take preventive measures at the planning phase. More often, security gaps are revealed later - during testing or after release. In such a situation, stakeholders must look for a way out in the shortest terms. Being in a rush, they may accept a second-rate solution, which results in an increased project cost and a damaged reputation.

The secure SDLC sets the most effective workflow, checking all possible vulnerabilities from the beginning. It scans for security holes during designing, configuring, coding, and testing. Adding security checks to all stages of the product lifecycle delivers robust software with optimal effectiveness. Here are some benefits you are going to receive after adopting secure SDLC:

- reliable software
- early detection of flaws
- faster data recovery
- less or no damage from cyber-attacks
- fewer business risks as a whole

Secure SDLC consists of five stages:

1. Studying requirements
2. Designing architecture
3. Coding
4. Testing
5. Deploying

Each stage has an obligatory security element, which we will cover in the next section.

---

## **How does a secure SDLC work?**

As mentioned above, secure SDLC is implemented through the usual steps of a software execution plan. The distinctive feature is the obligatory security measures added to each stage of the development life cycle. Let's see what these measures are like and how they shape the typical SDLC phases.

### **1. Studying requirements**

At this stage, the development team representatives examine the project details. Here, it is essential to study not only the software features but also the customer's business as a whole. When the project's environment is learned, the development team moves on to studying narrower system requirements. Next, it lists demands for a secure software lifecycle and produces terms of reference.

### **2. Designing architecture**

Program architecture is the process of transforming software characteristics such as flexibility, scalability, reusability, and security into a structured solution that meets both technical and business needs. At this step of secure SDLC, the team gathers all the requirements collected earlier and puts them in a single working system. To do it most efficiently, the presence of tech specialists is vital. The BAs and PMs should work closely to design a robust app that is equally efficient from the tech, business, and user perspectives.

Besides adhering to the general principles of architecture design, the team draws up threat models based on the security demands in the previous step. Threat modeling helps find points where security measures are essential for proper app work.

It is usually carried out in five stages:

By knowing where the danger comes from, the team can develop a robust defense system and excel in securing the software. That is, define the security tools and methods that will be applied during software development.

### **3. Coding**

Here, the developers enter into direct code writing. They do it with security in mind,

adhering to outlined requirements and paying special attention to weak points in the system. Debugging and error fixing proceed under pre-planned schemes; however, they must be flexible enough to propose non-standard solutions in urgent situations. All steps are documented and backed up, which helps easily roll back to the prior version in case of a critical error.

Developers and testers play the leading role in the coding stage of secure SDLC. Together, they produce a safe code derived through multiple checks and extensive testing. By checking small portions of code and fixing bugs immediately, they minimize errors in the final product release.

Below are some techniques that are often used at the coding stage of secure SDLC:

Developers should also be familiar with the best practices of hacking software worldwide. Hacking software usually searches for hidden flaws, ultimately identifying vulnerabilities and exploiting them for malicious attacks. Each year, the list of such problems is updated, and everyone can access it through the Internet. It is called CWE or Common Weakness Enumeration. In 2022, the first two places in the "chart" were occupied by Out-of-bounds writing and Cross-site scripting, respectively. You can see the complete list of modern software problems below.

#### **4. Testing**

Development and testing go hand in hand in secure SDLC. That is why we have already touched on this topic in the section above. Now, we'll consider it in more detail.

The security testing aims to identify flaws in the system's defense mechanisms that protect data and maintain functionality. The QA engineers perform code inspection, security metrics check, and executable file scan. The most common security testing techniques cover the following:

#### **5. Deployment**

If all security measures were properly taken in the previous steps of secure SDLC, then the number of bugs and issues should be minimal during deployment. However, you will still meet things that need to be polished or corrected. That is why it is crucial to stay sharp and continue monitoring system operations and high-chance vulnerabilities.

At the deployment phase in SDLC, the team members conduct a thorough security audit and assessment. They scan the system for major and minor bugs and provide immediate fixes if



necessary. They also compose documentation for the developed system and train users to work in it.

To avoid negative results after release and make sure the program is reliable, the team may apply beta testing. It involves rolling out the product to a limited number of users. In return, they agree to report on all glitches they face. This approach gives a good chance to look at the product from a different perspective and discover flaws that may have been overlooked. After completing all checks, the system is released to a wide audience, however the team continues to monitor and support it. If a security gap is found, the security development lifecycle makes a fresh start.

---

## Guide to establishing SSDLC

More and more engineers choose secure SDLC because it prevents potential security problems and minimizes the need to do major reworks in a ready app. The development team faces fewer bugs in the output, and the stakeholders get a stable product in a shorter time frame. Realizing all the benefits, the owners study ways to apply such a win-win approach in their development process. Below, we share how to enter this path and launch SSDLC in your company.

### **Step1. Hire cybersecurity specialists**

Businesses are observed to have a growing concern about SDLC cybersecurity. The ISC2 report states almost half of the surveyed companies want to increase their security staff.

This trend is not surprising, as cyberattacks are evolving and improving day by day.

To build secure software, you need to use the latest technologies and advanced techniques. However, the key role in establishing SDLC cybersecurity belongs to qualified staff, which raises the question: how to hire appropriate specialists? Well, first you need to know who you are looking for.

#### 1.1. Define the role of cybersecurity specialist

It is necessary to understand who an SDLC cybersecurity specialist is. On the one hand, it is clear they work on software protection systems; on the other, it is not entirely obvious what tasks they perform. So, before starting a headhunt, define your business needs, relevant types of work, and suitable staff roles. Just like other IT experts, cybersecurity software engineers have different specializations and varying skills and responsibilities. Let's consider them in more detail.

## **Application Security Engineer**

Such specialists build and align secure software development processes. They have a solid technical base, vast experience in coding, and know the difference between a secure development cycle and a regular one. The primary duties of an application security engineer are to:

- Build threat models. This means detecting app vulnerability spots, predicting ways of hacker attacks, and building reliable defensive mechanisms.
- Suggest secure code best practices. The security engineer educates developers to adhere to secure code development standards and approaches. They also review the written code.
- Perform code review. It is necessary to make sure the source code does not contain security gaps and issues.
- Control vulnerability testing. An application security specialist may take part in security testing or review the test results.

## **System Security Engineer**

They are in charge of building a secure software network and IT infrastructure. Such specialists configure the software and hardware components to ensure the network is highly resistant to hacker attacks. The main responsibilities of network engineers are to:

- Set up firewalls, data encryption programs, and routers
- Adjust virtual private networks and install email security programs
- Ensure the protection of servers on physical and program levels
- Track security issues and make reports for further analysis

## **Penetration Test Engineer**

The penetration testers perform special tests that simulate various malicious intrusions and evaluate the system's response to such attacks. In other words, [pentesters](#) act as hackers but attack the program with good intentions - to develop robust security measures and a strong protection plan. Their scope of duties is as follows:

- Perform penetration testing, vulnerability scanning, and secure code reviews.
- Build methodology, procedures, and documentation for security tests.
- Take part in malware engineering and reverse engineering.

- Assess the security of hardware, servers, and other network devices.

## **DevSecOps Engineer**

A Security DevOps engineer incorporates security into every step of the application development. They are in charge of data protection and take care of project safety from start to finish. Like a regular DevOps, the SDLC DevSecOps is involved in deploying, scaling, monitoring, and backing up apps and their parts. However, the latter uses many extra tools and methods like SDLC risk management, threat modeling, and cybersecurity to detect and analyze the threats. Key DevSecOps Engineer duties are to:

- Recognize the security threats and configure the network infrastructure
- Set up customized tools for security reasons in DevOps
- Implement solutions for system scalability, automation, and security
- Use configuration management tools to set up a smooth app operation

## **Security Architect**

A security architect uses specific techniques and tactics to organize a secure software system. They embed protective measures into all layers of software development, from app logic and backend components to intersystem connections and interface parts. The duties of a security architect are to:

- Develop secure software design methodology, functioning algorithm, and information processing methods
- Choose and calculate information protection measures
- Design data security systems
- Layout database and choose secure ways of app deployment

## **Security Consultant**

Security consultants are not directly involved in software development and testing but advise on proven ways to do it safely. They should be well-versed in the types and methods of malicious infiltration, the latest security systems, and the legal framework that governs software security. Consultants often repeat the duties of security architects. As a rule, they are hired to consult on specific issues without ongoing project support. Their duties are to:

- Identify and point out flaws in the network infrastructure

- Monitor key parameters of system performance
- Suggest ways to prevent data leakage
- Work on comprehensive app protection

## 1.2.Choose a cybersecurity specialist for your project

Depending on your project size and nature, you may need all of these people or just some of them. For a small app, it may be enough to hire a security consultant only. After examining your product, they will guide your developers through the 'dos and don'ts' of a secure development lifecycle. If you have extensive software, especially related to finance and healthcare, you will likely need a fully packed team of security specialists. They will accompany your project from planning to release and report on every nuance of the secure software development process.

### **Step 2. Use SAMM framework**

When you are imbued with the need for secure software engineering and have classy specialists, it is time to take action and begin the SSDLC setup. But where to start? What security measures to take and what procedures to add to your workflow?

It is near-impossible to answer these questions without a well-considered plan. If you want the secure software development framework to bring results, it is necessary to implement it holistically. You would need to introduce the security element to each development stage, department, and employee.

Fortunately, a scheme that makes the above task easier exists. It is called SAMM or Software Assurance Maturity Model framework. It is essentially a set of rules that helps assess the current security level and, based on this, develop procedures that will ensure an entire cycle of secure development. The SAMM framework is suitable for companies of all types and sizes. It breaks software development into four business processes, each including three types of secure software development practices.

## 2.1. Set goals

First, it is necessary to understand why a secure SDLC is needed and what aims you want to achieve with its help. For that, list critical purposes for each SAMM domain that will promote your overall security system.

## 2.2. Produce an activities checklist

Next, draw up activities to help you advance your goals. Develop a detailed plan that will fully describe your path to building a secure system.

- ✓ Assess current business risks
- ✓ Assign a risk level to the data and the applications
- ✓ Define security actions for each risk level
- ✓ Develop internal security SDLC standards
- ✓ Start project audit practice
- ✓ Organize role-specific workshops
- ✓ Hire experienced security coaches
- ✓ Create a formal application security support portal
- ✓ Implement regular security tests and certification
- ✓ Create detailed threat models
- ✓ Develop attacker profiles
- ✓ Record abuse cases from each project
- ✓ Establish a threat rating scale
- ✓ List the security steps for each threat model
- ✓ Draw up a control matrix to assess resources and perspectives
- ✓ Compose a list of software frameworks
- ✓ Create reference examples of architecture and design
- ✓ Develop data-flow diagrams for sensitive resources
- ✓ Run code reviews
- ✓ Build release gates for code review
- ✓ Implement test cases based on the security requirements

- ✔ Use penetration testing and auto testing tools
- ✔ Set up acceptance criteria for all tested indicators
- ✔ Appoint a competent person for security issues
- ✔ Develop a clear issue response process
- ✔ Collect per-issue metrics
- ✔ Use the latest version of infrastructure components
- ✔ Document steps to follow when facing application alerts
- ✔ Adjust change-management procedures for each release
- ✔ Use code signing

### 2.3. Create assessment worksheet

The next step is to assess how well prepared your organization is in various aspects of security. For this, develop a questionnaire and pass it to authorized persons. Ask them to provide answers using grades from 0 to 3. Zero would indicate the lowest security level, and three would be the highest one. See the detailed description of each level below.

In a real-case scenario, it is not always possible to identify a specific rating. This is because some security measures may exceed the actions prescribed for a lower level but be insufficient for the higher one. In this case, the company can add the "+" sign to the current security level and change it to a higher grade later. This rule is also valid for the third level of maturity. That is, if you take more security measures than the SAMM framework prescribes, you can set the 3+ mark. Below, you can see an example of a security architecture questionnaire.

### Step 3. Develop your own security program

Secure SDLC is a part of the company's safety philosophy. It smoothly integrates into well-established processes that provide reliable protection at all enterprise levels. The security program within an organization is based on industry-accepted standards and best security practices in software development. For all that, it is a unique solution that requires a solid plan and individual approach. A security program is designed after considering key factors that affect organization activity. They are:

- business location
- company size
- activity field
- external relations
- corporate culture
- objectives pursued
- current security state

Although there is no unified recipe to protect from security threats, you can use proven measures to build your own security program as a part of secure SDLC:

---

## Benefits of Secure Software Development Lifecycle

Companies that value their resources, and want to avoid unnecessary costs, benefit from adhering to secure development lifecycle. Even though the security element increases the development time and budget, it brings significant savings in the long run. It is primarily due to the fact companies do not need to spend money on fixing bugs and security gaps.

According to [Aberdeen](#)'s research, prevention of one security breach almost fully covers the annual costs of setting secure development.

[HP](#) claims it is 30 to 100 times more expensive to fix bugs after release than on the requirements analysis phase.

[NIST](#) report says post-release bug fixes are 30 times more expensive than on-the-fly bug fixes.

Besides, a secure software development life cycle has a positive effect on other aspects of the product lifecycle and is manifested in the following:

### Increased product value

According to Toyota's philosophy, there are value-added and non-value-added processes in the development stream. Customers pay only for value-added processes, while non-value-added processes only delay the release of the finished product.

In the picture above, you can see software rework and bug fixes occupy a prominent place among inefficient processes. With a secure SDLC, you can reduce or entirely remove these activities, adding value to your final product.

### **Rapid software recovery**

According to IBM experts, it takes about 280 days to identify and eliminate violations. Companies that rectify problems in less than 200 days spend on average \$1 million less. Others are losing vast sums simply because they cannot find a security hole.

Secure software development life cycle does not provide a 100% guarantee of data security but minimizes the likelihood of failure by introducing protective mechanisms at all stages of development. It also helps to quickly identify the breach source, which means saving money and the company's reputation.

### **Grown brand confidence**

Needless to say that reliable software gains customers' trust faster. It works with high performance, no downtime, and a minor number of glitches. Stable system operation enhances your reputation and orients clients towards long-term cooperation.

Given this, secure SDLC becomes especially relevant. With its help, you can minimize errors in the output and develop a better and more robust product that will quickly gain users' trust. In addition, a secure software development life cycle is beneficial from an economic point of view. By investing in quality, you minimize the cost of fixing bugs, avoid unhappy customers, and give your software more chances to reach a wider audience and bring bigger profits.

---

## **SSDLC with Erbis**

**Looking to establish a secure SDLC?**

**Connect!**





---

### **Read Also**

---

---

**How to Attract More Investments With the SSDLC Approach**

---

---

**How to Execute Technology Migration Without Business Harm**

### **Articles**

---

---

**Erbis Wins Prestigious Excellence in Custom Software Solutions 2024 Award**

---

---

**15 Best IT Outsourcing Companies to Save Costs and Boost**

## Efficiency



Show All Articles

## Latest News

Erbis Wins Prestigious  
Excellence in Custom  
Software Solutions  
2024 Award

Erbis CEO Anton  
Zimarov Named 2024  
CEO of the Year by  
European CEO Awards



Show All News

Share your ideas, get our solutions

Full name \*

Email \*



How can we help you? \*

By submitting this form, you allow Erbis processing your personal information as set out in [Privacy Policy](#) & [Cookie Policy](#) \*

Send Message

Or Book A Call



Expert development

Expert delivery



hello@erbis.com



+13416996437

#### UNITED STATES

50 California St,  
San Francisco

#### POLAND

Wielicka Street 28,  
Kraków

#### UKRAINE

10B Mechnikova St,  
Dnipro



