Five Safety terminologies are:

1.Hazards

Hazards can refer to potential sources of harm or danger that can arise from the use of software or related systems. Some examples of hazards in software engineering include:

a) Data loss: This can occur if a software system experiences an unexpected failure or is hacked, leading to the loss of important data.
b) Cybersecurity breaches: Cyber-attacks, such as viruses or malware, can pose a hazard to software systems by causing data loss, theft, or damage to the system.

2. Accident (or mishap)

An accident (or mishap) is an unexpected or unintended event that causes harm or damage. An example of an accident might be a bug in a piece of software that causes it to crash or behave unexpectedly, leading to data loss or system failure.

To prevent accidents in software engineering, it is important to follow best practices in software development, such as writing clear and maintainable code, performing thorough testing, and implementing robust error handling. It is also important to have a robust incident response plan in place to quickly and effectively address any accidents that do occur.

3. Damage

Damage is a term used to describe the negative impact that can occur as a result of a software failure or malfunction. For example, if a software system used to control an industrial process fails, it could result in damage to equipment, loss of production, or even injury to personnel.

For example, an aviation software system might include multiple layers of redundant systems and rigorous testing procedures to ensure that the software functions correctly and safely at all times. If a failure does occur, the system should be designed to fail safely and minimize the potential for damage.

4. Risk

Risk refers to the potential for negative consequences to occur as a result of a project or decision. Risks can be related to various aspects of a software project, such as technical issues, schedule delays, budget overruns, and legal or regulatory issues.

For example, consider a software development project that involves building a new application for a client. One risk that the team might identify is the possibility that the client may not like the final product and decide not to use it. This could lead to lost revenue and a negative impact on the company's reputation. To mitigate this risk, the team might choose to involve the client in the development process and conduct regular check-ins to ensure that the final product meets their needs and expectations.

5. Hazard probability

Hazard probability refers to the likelihood that a hazard or error in the software will cause harm or damage. This can be assessed through various methods, such as risk analysis, which involves identifying and evaluating potential hazards in the software and determining their likelihood of occurrence.

For example, consider a software system that controls the operation of a nuclear power plant. If the software contains a hazard that could cause the reactor to malfunction, the probability of that hazard occurring would need to be carefully evaluated to determine the risk to the plant and the surrounding area. This could involve analysing the likelihood of the hazard being triggered by normal operation of the plant, as well as the potential consequences if it were to occur.

Five Security terminologies are:

1. Asset

An asset is any resource that has value to an organization and is used to create, maintain, or enhance software products or services. Some examples of assets include:

a) Source code: The source code is the human-readable instructions that make up a software program. It is a valuable asset because it represents the intellectual property of the organization and is used to create and maintain software products.
b) Documentation: Documentation is any written or visual material that explains how a software product or service works. It is a valuable asset because it helps developers, users, and other stakeholders understand and use the software.

2. Attack

An attack refers to a specific type of cyber threat in which an attacker attempts to gain unauthorized access to a system or steal sensitive data. There are many different types of attacks, including:

a) Denial of service (DoS) attack: A DoS attack is an attempt to make a system or network unavailable to its intended users by overwhelming it with traffic or requests. For example, an attacker could send a large number of fake requests to a server, causing it to crash and become unable to respond to legitimate requests.
b) SQL injection attack: An SQL injection attack is a type of cyber-attack in which an attacker injects malicious code into a database through an application that is not properly validated.

3. Control

Control refers to measures that are put in place to ensure that a system functions as intended and meets the requirements of its users. This includes measures to ensure that the system is stable, reliable, and secure.

One example of a control is input validation. Input validation is the process of checking the data that is entered into a system to ensure that it meets certain criteria. For example, a system that accepts credit card payments may have input validation to ensure that the card number and expiration date are in the correct format and that the card has not been reported as lost or stolen.

4. Exposure

Exposure refers to the potential for sensitive information or assets to be accessed or compromised. For example, if a software system stores sensitive personal information, such as social security numbers or financial data, and that information is not properly protected, it is exposed to the risk of being accessed by unauthorized individuals.

One common way that software systems can be exposed is through vulnerabilities, which are weaknesses or flaws in the system that can be exploited by attackers. For example, if a software application has a flaw that allows an attacker to gain unauthorized access to the system, the sensitive data stored within the system is exposed to the attacker.

5. Threat

A threat in software engineering refers to a potential vulnerability or weakness in a system that could be exploited by an attacker. Threats can come in many forms, such as malware, phishing attacks, or unsecured data storage practices.

For example, consider a software system that stores sensitive customer data, such as social security numbers and credit card information. If the system does not have proper security measures in place, it could be vulnerable to a threat such as a data breach, where an attacker gains unauthorized access to the data and steals it.