SSDLC Secure software development life cycle

Processes & activities to develop a software with security in mind.

Adding security

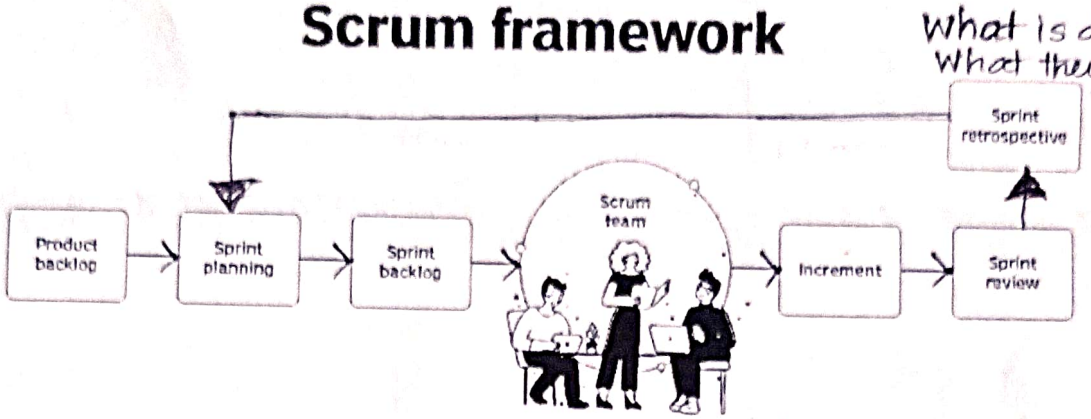
- Conduct threat modeling —— identify attacks, determine how software can protect against attacks during design phase
- Use secure design pattern —— Eg. "Fail-secure" pattern ensures s/w fails safely in the event of attack
- Implement access control, input validation to protect sensitive data.
- Follow secure coding
- Conduct security assessment, & fix vulnerabilities review

Security in ~~phase~~ phases in SSDLC

1. Requirement ——→ Identify security features in functional requirements
2. Design ——→ Page should retriever user's information database Verify if user has a valid token for session
3. Coding ——→ Make sure that code is well-written from security perspective — Use libraries that implement security — sanitizing data from database to user & vice versa.
4. Testing —— Use automated security testing tools
5. Maintainance — Patchup security issues discovered by users.

AGILE Model

Scrum framework



scrum master
product owner
product developer

Sprint — A short period of time in which developers focus on few important features

Product backlog
List of features to be developed

Product vision
Description of products
What + Why + How

→ product owner

8 Sprint planning ⟶ planning for short time not more than 8 Hours

Increment — Part of product developed in a sprint.

Extreme Programming (XP)

- A type of AGILE model
- Focuses on high quality delivery through frequent & continuous feedback
- Phases