# Perfect Software Engineering Notes (XP, Scrum, Security, RE, Cyber Attacks)

**Software Engineering Essential Notes**

---

## 1. eXtreme Programming (XP) - 12 Principles with Definitions, Advantages & Disadvantages:

1. **The Planning Game** - Collaborative planning between developers and customers.
2. **Small Releases** - Frequent, incremental delivery.
3. **Metaphor** - Shared, simple system-wide analogy.
   - Adv: Shared understanding, better communication.
   - Disadv: May oversimplify.
4. **Simple Design** - Minimal design needed to work.
5. **Test-Driven Development** - Write tests before code.
6. **Refactoring** - Improve code structure without changing functionality.
7. **Pair Programming** - Two developers collaborate on one task.
8. **Collective Ownership** - Anyone can change any code.
9. **Continuous Integration** - Integrate and test code frequently.
10. **40-Hour Week** - Sustainable pace.
11. **On-Site Customer** - Customer present for feedback.
12. **Coding Standards** - Consistent, readable code.

**Advantages of XP:**

- Encourages frequent releases and early feedback.
- Reduces bugs via continuous testing.

**Disadvantages of XP:**

- Not suitable for large teams.

- May lack comprehensive documentation.

---

**2. Scrum Roles - Definitions, Advantages & Disadvantages:**

- **Product Owner (PO):** Defines product vision, manages backlog.

- **Scrum Master (SM):** Ensures Scrum rules are followed, removes impediments.

- **Developers:** Build and deliver the product increment.

**Advantages:**

- Promotes collaboration and transparency.

- Adaptable to changing requirements.

**Disadvantages:**

- Requires experienced team.

- Can be disrupted by unclear requirements.

---

**3. Requirement Engineering - Definition & Process:**

**Definition:** Systematic process to gather, define, and manage requirements.

**Process Includes:**

- **Feasibility Study**

- **Requirements Elicitation**

- **Requirements Specification**

- **Requirements Validation & Verification**

- **Requirements Management**

---

**4. Software Security - Key Properties & Steps:**

**Definition:** Practice of building software free from vulnerabilities.

**Security Properties:**

1. **Confidentiality** - Prevent unauthorized access.

2. **Integrity** - Prevent unauthorized modification.

3. **Authentication** - Confirm user identity.

4. **Non-repudiation** - Trace actions to users.

5. **Availability** - Ensure system uptime.

---

**5. Cyber Attack - Definition:**

A **cyber attack** is a malicious attempt to damage, disrupt, or gain unauthorized access to computer systems, networks, or data.

**Examples:**

- **Malware**

- **Password Attacks**

- **SQL Injection**

- **Zero-Day Attacks**

---

This note captures essential practices in modern software engineering including agile methodologies, team roles, requirements engineering, and security.