Change Control Issues (Challenges in Managing Changes)

1. **Synchronization (When?)**
   - Ensuring that changes happen **at the right time** without disrupting ongoing work.
   - Example: Updating software only during maintenance hours.
2. **Identification (Who?)**
   - Knowing **who requested or made the change** for accountability.
   - Example: Tracking the developer responsible for code modifications.
3. **Naming (What?)**
   - Properly naming files or versions to **avoid confusion**.
   - Example: Using version numbers like **v1.0, v1.1, v2.0**.
4. **Authentication (Done correctly?)**
   - Ensuring only **authorized users** can make changes.
   - Example: Requiring login credentials before modifying system settings.
5. **Authorization (Who approved it?)**
   - Making sure changes are **officially approved** before implementation.
   - Example: A project manager must approve critical updates.
6. **Routing (Who's informed?)**
   - Making sure the **right people** know about the change.
   - Example: Informing all team members about an updated feature.
7. **Cancellation (Who can stop it?)**
   - Defining **who has the authority** to cancel an unnecessary or risky change.
   - Example: A security team stopping a software release due to a security flaw.
8. **Delegation (Responsibility issue)**
   - Assigning **who is responsible** for implementing the change.
   - Example: Assigning a senior developer to handle a critical bug fix.
9. **Valuation (Priority issue)**
   - Determining **how important** a change is compared to others.
   - Example: Fixing a security vulnerability before adding new features.