Perfect 👍 Let’s make precise WSTG notes for

**📝 Testing for the Circumvention of Workflows (WSTG-BUSL-07)**

**🎯 Purpose**

To determine if an attacker can **skip, replay, or reorder steps in a multi-step business process** to gain unauthorized benefits.  
Many web apps rely on **client-side navigation** and assume users follow the intended flow — attackers abuse this by **jumping directly to sensitive steps**.

**⚡ Common Attack Scenarios**

1. **Checkout Process Bypass**
   * Skipping payment step and going straight to order confirmation.
2. **Registration Process**
   * Skipping email/phone verification → account creation without validation.
3. **Authorization Escalation**
   * Accessing /admin/approve without going through login.
4. **Skipping Validation Steps**
   * Submitting loan application directly at "approved" step.
5. **State Parameter Manipulation**
   * Changing step=3 in URL or POST to move ahead in workflow.

**🔍 How to Test**

1. **Map the Workflow**
   * Document required steps (e.g., Add to Cart → Checkout → Payment → Confirmation).
2. **Force Navigation**
   * Try accessing later steps directly via URL/bookmarks (e.g., /checkout/confirm).
3. **Replay/Modify Requests**
   * Intercept with **Burp/ZAP** and change hidden fields or state parameters.
4. **Skip or Reorder Steps**
   * Send requests out of order, omit required fields, or bypass validation pages.
5. **Check for Server-Side Enforcement**
   * Is the backend verifying that prior steps were completed? Or does it trust client-side flow?

**🛡️ Mitigation**

* Enforce **server-side state management** (track current workflow stage).
* Use **unpredictable, one-time tokens** per workflow step.
* Validate **prerequisites** before advancing to the next step.
* Ensure hidden parameters and client-side logic cannot override state.
* Apply **logging & anomaly detection** for skipped steps.

✅ **Key Takeaway:**  
If workflows are not enforced **server-side**, attackers can skip validation, payment, or authorization steps — leading to fraud, privilege escalation, or bypass of controls.

👉 Do you want me to also create a **tester’s playbook (step-by-step workflow abuse checklist)** so you can directly apply it when assessing real apps?