Phase 8: Data Management & Deployment

👉 **Goal:** Ensure clean data, safe migration, and smooth deployment of the Smart Service Request System to production.

---

1. Sample Data Import

- Use **Data Import Wizard** to load sample Accounts, Contacts, and Service Requests.
- Validate that relationships (Account → Contact → Service Request) are correctly maintained after import.

---

2. Data Quality Rules

- Create **Validation Rules** to prevent incomplete records (e.g., Resolution Notes required before closure).
- Set up **Duplicate Rules** to stop duplicate Contacts or Service Requests.
- Apply **Picklist Value Restrictions** to maintain consistency.

---

3. Backup Strategy

- Schedule regular **Data Export** in Salesforce to back up Accounts, Contacts, and Service Requests.
- Store backup files securely for recovery in case of accidental data loss.

---

4. Deployment Tools

- Use **Change Sets** to move customizations (objects, fields, layouts, automation, Apex) from **Sandbox to Production**.
- Ensure dependencies (like fields, validation rules) are included in the deployment.
- For larger projects, consider using **Salesforce CLI or ANT Migration Tool**.

---

5. Sandbox Usage

- Perform all development and testing in a **Sandbox environment**.
- Only push final changes to Production after successful testing.
- Use **Partial Copy Sandbox** if real test data is required.

---

6. Post-Deployment Testing

- After deploying to Production, test full scenarios:
  - Create → Assign → Resolve a Service Request.
  - Validate reports and dashboards.
  - Check integrations (Email-to-Case, API, Experience Cloud).

- Fix any issues before go-live.

---

7. End-User Training & Documentation

- Provide staff and managers with training sessions or user guides.
- Share documentation on how to create, assign, and close requests.
- Educate admins on backup, monitoring, and troubleshooting.

✅ With Phase 8, the Smart Service Request System ensures **data integrity, safe deployment, and smooth transition to production**, reducing risks of errors or downtime.