---

1. Current Environment Summary

- **Platform**: SharePoint Server 2016
- **Usage**: Static or semi-static faculty websites — *no search, profiles, or document collaboration*
- **Design Tool**: SharePoint Designer 2013
- **Upgrade Plan**: Move to **SharePoint Server Subscription Edition (SE)**
- **Concern**: SharePoint Designer's retirement (end of support in 2026) and future editability of site pages

---

2. Technical Context: What Happens on Upgrade

When upgrading from SharePoint 2016 to **Subscription Edition (SE)**:

- Classic pages, master pages, and custom layouts **continue to work**; SE is fully backward-compatible with SharePoint 2016 site collections (once upgraded via database attach).
- SharePoint Designer 2013 can **still connect** to SE — Microsoft did not block connectivity.
- However, **no new design tool** will replace SharePoint Designer officially, and SE **does not include a modern equivalent** for direct HTML/ASPX editing.

In short: after upgrading to SE, your existing web pages and site structures will remain functional, and SharePoint Designer 2013 can still be used — but only until **its support and likely interoperability break around 2026–2027**.

---

3. Risks After 2026

- **No updates or compatibility fixes** for SharePoint Designer (SPD).
- **Windows or authentication model updates** may eventually prevent SPD from connecting to SE (e.g., TLS 1.3 enforcement or deprecation of legacy auth).
- New **browser and OS security models** could block certain design surface features.
- **No supported Microsoft alternative** will exist for editing ASPX pages directly inside SharePoint.

---

4. Recommended Strategy

To preserve editability and manage risk, consider one of the following paths depending on your long-term plans:

Option A – Short-term (2024–2026): Continue SPD until deprecation

- Keep using SharePoint Designer 2013 for maintenance.
- Document all customization (master pages, layouts, CSS, scripts).
- Gradually refactor hard-coded content (custom HTML/JS/CSS) into reusable templates or **Content Editor / Script Editor web parts**.
- Ensure compatibility by testing in SE staging before final upgrade.

Pros: Minimal disruption.
Cons: Temporary solution; tool obsolescence remains.

Option B – Mid-term (2026+): Transition to Modern or External Editing Workflow

1. **Convert classic pages to modern pages** where feasible:
   - Modern pages can be edited **via browser only**, no external tool needed.
   - They use responsive layouts and are compatible with future SharePoint versions.
   - For campus content, you can create **modern Communication Sites** (one per faculty).
   - Custom CSS/JS can be migrated using the **SPFx (SharePoint Framework)** if necessary.
2. **Externalize page management**:
   - Manage HTML content outside SharePoint (e.g., in a static site generator like Hugo, Jekyll, or an internal CMS).
   - Deploy pages to SharePoint as static files or via automation (PowerShell + REST API).
   - This approach separates **content design** from **SharePoint infrastructure**.
3. **Use PowerShell or PnP (Patterns and Practices)** for updates:
   - PnP PowerShell or PnP Framework can manage and update page content programmatically, bypassing SPD altogether.

Pros: Future-proof and standards-based.
Cons: Requires migration and staff adaptation.

Option C – Long-term: Static Web Hosting Alternative

If SharePoint is used purely as a **publishing platform for faculty pages** (no dynamic features), consider:

- Migrating those sites to **Azure Static Web Apps**, **IIS static hosting**, or **WordPress multisite**, depending on your governance model.
- This eliminates SharePoint's overhead entirely for what is essentially static HTML content.

Pros: Simplified management, cheaper long-term, no SPD dependency.
Cons: Requires migration and loss of SharePoint-specific features (permissions, navigation inheritance, etc.).

---

5. Practical Recommendation

For now:

- Proceed with upgrade to **SharePoint Subscription Edition**.
- Keep **SharePoint Designer 2013** for maintenance until 2026.
- Start a **transition plan in 2025** toward modern pages or an external editing process.
- Maintain a **test SE environment** to verify SPD compatibility after each SE cumulative update.
- Begin documenting and cleaning up all **custom master pages and scripts** to simplify eventual migration.

---

6. Summary Table

Timeframe	Editing Tool	Approach	Longevity	Notes
2024–2026	SharePoint Designer 2013	Continue maintenance	Short-term	Fully compatible with SE
2026–2028	Modern pages / PnP automation	Migrate to browser-based editing	Mid-term	Supported by Microsoft
2028+	Static or hybrid CMS	Replace SharePoint for pure web hosting	Long-term	Simplifies architecture

---