GRC Engineering Checklist

Adopting GRC engineering isn't something you can accomplish overnight. Many organizations start by taking careful, foundational first steps, then build on those changes over time. Here is a checklist to help you and your team stay on track as you modernize your GRC:

Ensure the Right Mix of Technical Skills

- [ ] Automation and workflow design
- [ ] Data engineering and data modeling
- [ ] Systems thinking and design thinking
- [ ] Scripting and query skills

Assign Roles and Responsibilities

- [ ] GRC product manager
- [ ] GRC engineers/technical implementation experts
- [ ] GRC translators (liaison between compliance and engineering)
- [ ] Data and control architecture owners
- [ ] Workflow and process designers
- [ ] Stakeholder experience designers
- [ ] Risk analysts

Identify Tools and Systems to Integrate

- [ ] Cloud infrastructure and configuration (e.g., AWS, Azure, GCP)
- [ ] Identity access management (IAM) (e.g., Okta, Google Workspace)
- [ ] Ticketing and workflow (e.g., Jira, ServiceNow, Asana)
- [ ] Logging, monitoring, and SIEM (e.g., Datadog, Splunk, Microsoft Sentinel)
- [ ] Endpoint and device management (e.g., Jamf, Microsoft Intune, IBM MaaS360)
- [ ] Vulnerability management and security (e.g., Tenable, Qualys)
- [ ] Code repositories (e.g., GitHub, GitLab, Bitbucket)
- [ ] GRC automation platform (e.g., Drata)
- [ ] Asset inventory and Configuration Management Database (CMDB) (e.g., ServiceNow, BMC Helix CMDB)

Align Cross-Functional Teams

- [ ] Security
- [ ] Engineering
- [ ] IT
- [ ] HR
- [ ] Communications
- [ ] Legal and compliance
- [ ] Data and analytics