Technical Due Diligence Checklist

1. Engineering Culture & Human Capital Risk

Objective: Quantify the "Bus Factor" and validate the maturity of the engineering process.

Status	Audit Task	Investigation Method / Question	The Red Flag (Warning Sign)	Criticality
[ ]	**Verify "Bus Factor"**	Run a git analysis tool (e.g., git-fame) or review commit logs for the last 6 months.	**>50% of active code** was written by a single person (or someone who has left).	🔴 **High**
[ ]	**Test Knowledge Distribution**	Ask: *"If [Lead Architect] is unreachable for 2 weeks, can the team deploy a hotfix?"*	Answer is a hesitant "No" or "We'd have to wait."	🔴 **High**
[ ]	**Assess "Deployment Fear"**	Ask: *"Do you deploy on Fridays?"*	Answer: *"Never, it's too risky."* Indicates a brittle CI/CD pipeline.[1]	🟠 **Med**
[ ]	**Check Deployment Frequency**	Review CI/CD logs. How often does code go to production?	Code is released **less than once every 2 weeks** (Low DORA metric score).	🟠 **Med**
[ ]	**Validate Agile Process**	Request notes from the last 3 Sprint Retrospectives.	Notes are missing, or they list the same problems 3 times in a row (Performative Agile).	🟡 **Low**
[ ]	**Audit Access Control**	Pick 3 former employees and	Access remained active **after** their	🔴 **High**

		check logs for GitHub/AWS/Slack access revocation.	termination date.	

2. Software Architecture & Scalability

Objective: Identify "Distributed Monoliths" and hard scalability ceilings.

Status	Audit Task	Investigation Method / Question	The Red Flag (Warning Sign)	Criticality
[ ]	**Check Database Isolation**	Review architecture diagrams for microservices.	**Shared Database Pattern:** Multiple microservices read/write to the same DB tables (High coupling).	🔴 **High**
[ ]	**Audit Database Queries**	Enable slow query logs and check for "N+1" query patterns on list pages.	A single page load triggers **hundreds of SQL queries**.	🟠 **Med**
[ ]	**Review Schema Hygiene**	Check table definitions for core entities (Users, Orders).	**JSONB Abuse:** Core relational data is dumped into JSONB columns to avoid schema design.	🟠 **Med**
[ ]	**Verify Horizontal Scaling**	Check where user sessions and file uploads are stored.	stored on **local server disk** instead of distributed cache (Redis) or Object Storage (S3).	🔴 **High**
[ ]	**Identify EOL Components**	Inventory version numbers of databases and OS.	Running on **End-of-Life versions** (e.g., Python 2.7, PostgreSQL 9.6) that receive no	🔴 **High**

			security patches.	
[ ]	**Analyze Circular Deps**	Review service-to-service call graphs.	Service A calls Service B, which calls Service A (Circular Dependency leading to deadlocks).	🟠 **Med**

3. Source Code Quality

Objective: Move beyond "gut feeling" to data-driven quality metrics.

Status	Audit Task	Investigation Method / Question	The Red Flag (Warning Sign)	Criticality
[ ]	**Measure Complexity**	Run a static analysis tool (SonarQube/Lizard). Check "Cyclomatic Complexity".	Files with complexity score **> 50**. These are "God Classes" and are effectively untestable.	🔴 **High**
[ ]	**Scan for Duplication**	Run a "Copy-Paste Detector" (CPD) scan.	Code duplication rate is **> 15-20%**. Increases maintenance cost linearly.	🟠 **Med**
[ ]	**Audit Test Pyramid**	Review the ratio of Unit Tests to UI/End-to-End Tests.	**"Ice Cream Cone" Pattern:** 90% slow Selenium tests, 10% fast Unit tests.	🟠 **Med**
[ ]	**Check Dependency Age**	Audit package.json or requirements.txt.	Core frameworks (React, Rails, Spring) are **> 2 major versions behind**. (Upgrade = Rewrite).	🔴 **High**

[ ]	**Review Comment Density**	Sample 5 core files. Look for "Warning" comments.	Comments like *"// DO NOT TOUCH THIS,"* *"// Hack,"* or *"// TODO: Fix later"* from 3 years ago.	🟡 **Low**

4. Infrastructure & FinOps

Objective: Ensure the business model is structurally profitable (Unit Economics).

Status	Audit Task	Investigation Method / Question	The Red Flag (Warning Sign)	Criticality
[ ]	**Analyze Cloud COGS**	Plot "Hosting Bill" vs. "Revenue" for the last 12 months.	**Linear Scaling:** Cloud costs grow 1:1 with revenue (Zero economy of scale).	🔴 **High**
[ ]	**Check Tenant Costing**	Ask: *"What is the exact hosting cost for Customer X?"*	Team **cannot answer**. They lack tagging/visibility into unit economics.	🟠 **Med**
[ ]	**Verify Disaster Recovery**	Ask for the *report* from the last DR simulation.	**No recent test** (last 12 months). RTO/RPO policies are likely theoretical.	🔴 **High**
[ ]	**Audit Infrastructure as Code**	Review Terraform/CloudFormation repos.	**"ClickOps":** Infrastructure is managed manually in the console (Unrepeatable recovery).	🟠 **Med**
[ ]	**Gross Margin Check**	Calculate SaaS Gross Margin (Rev - COGS / Rev).	Gross Margin is **< 70%** (indicates inefficient architecture or poor pricing).	🔴 **High**

5. Security & Governance

Objective: Identify liabilities that could result in fines or deal-breaking breaches.

Status	Audit Task	Investigation Method / Question	The Red Flag (Warning Sign)	Criticality
[ ]	**Scan for Secrets**	Search codebase for regex: AWS_SECRET_KEY, BEGIN RSA PRIVATE KEY.	**Hardcoded credentials** found in the source code repository.	🔴 **High**
[ ]	**Review Pentests**	Compare the last 2 annual Penetration Test reports.	The same **High/Critical vulnerabilities** appear in both reports (Broken remediation process).	🔴 **High**
[ ]	**Check Vendor Risk**	Ask which security questionnaire they use for vendors.	They do **not use** standard forms like CAIQ (Cloud) or SIG Lite.	🟡 **Low**
[ ]	**Audit MFA**	Check Identity Provider (Okta/Google Workspace) settings.	MFA is **not enforced** for all users, or SMS (insecure) is the only option.	🔴 **High**
[ ]	**Verify Privacy (GDPR)**	Ask to see the "Record of Processing Activities" (RoPA).	No documentation exists on where customer PII is stored.	🟠 **Med**

6. Intellectual Property & Legal

Objective: Confirm you are actually buying the code you think you are buying.

Status	Audit Task	Investigation	The Red Flag	Criticality

		Method / Question	(Warning Sign)	
[ ]	**PIIAA Audit**	Audit HR files for "Proprietary Information and Inventions Assignment Agreements".	**Missing agreements** for key early founders or contractors.	🔴 **High**
[ ]	**License Scan**	Run a scan for "Copyleft" licenses (AGPL, GPL).	**AGPL libraries** linked to the proprietary core engine (Risk of forced open-sourcing).	🔴 **High**
[ ]	**Export Control**	Check if software uses non-standard encryption.	**No classification** with Bureau of Industry and Security (BIS) despite global sales.	🟠 **Med**

7. AI & Data Assets (Emerging Risk)

Objective: Assess risks specific to Generative AI and Machine Learning.

Status	Audit Task	Investigation Method / Question	The Red Flag (Warning Sign)	Criticality
[ ]	**Training Data Rights**	Ask for the "Data Inventory" and legal license for every training dataset.	**"Scraped" data** used without clear commercial license or consent.	🔴 **High**
[ ]	**AI Wrapper Risk**	Review the "AI Architecture".	Product is a **thin wrapper** around OpenAI API with no proprietary model or defensibility.	🟠 **Med**

[ ]	**Output Ownership**	Review customer contracts regarding AI-generated content.	Contracts are **silent on who owns** the output (Customer vs. Vendor), creating IP ambiguity.	🟠 **Med**