Adrian Marquez

Module 6.2

**Case Study Summary: Strangler Pattern at Blackboard Learn**

In 2011, Blackboard Learn, a widely used learning management system, was facing severe issues with their legacy application. The system had grown so large and tightly coupled that deploying even minor updates was risky and slow, resulting in long lead times, downtime, and developer frustration.

To address this, the team adopted the **Strangler Pattern**, a technique where new functionality is built as separate services around the existing application, slowly “strangling” the monolith by replacing old components piece by piece. This approach allowed Blackboard to modernize incrementally rather than doing a risky, full-scale rewrite.

They started by identifying high-traffic, low-complexity parts of the system and rewriting them as independent services using modern DevOps practices. Over time, they rerouted traffic to the new services, which reduced deployment times and improved system reliability.

**Lessons Learned:**

* Incremental refactoring is safer than attempting to rebuild everything at once.
* Small, well-defined services are easier to deploy, test, and maintain.
* The Strangler Pattern enables modernization without halting development or causing massive system risk.
* Adopting DevOps and CI/CD practices alongside architecture changes amplifies success.