In 2011, Blackboard learn, a widely used learning management system, faced growing challenges due to its aging, monolithic architecture. The system had become difficult to maintain, resistant to change, and slow to evolve, creating bottlenecks in development and delivery. To address this, the development team adopted the Strangler pattern, a gradual approach to system modernization.

The primary issues stemmed from the monolithic nature of blackboard learns codebase. Any change to the system had the potential to introduce bugs in unrelated areas due to the lack of seperation between components. The accumulation of technical debt over time made even small changes expensive and time consuming. Additionally, the software's long release cycles limited the team's ability to respond quickly to feedback or deploy new features in a timely manner, further frustrating users and developers alike.

To address these challenges, Blackboards engineering team began refactoring the system using the Strangler Pattern. This approach involves gradually building a new system around the edges of the old one, eventually replacing the legacy's components piece by piece. The team introduced new functionality through modular services that operated independently from the legacy codebase.

One of the key lessons from Blackboards experience is that incremental refactoring is both feasible and effective for modernizing legacy systems. The Strangler Pattern allowed the team to improve the architecture over time without resorting to a risky, large-scale rewrite. Modularity emerged as a critical benefit, as the decoupled services improved scalability, maintainability, and development speed. Maintaining business continuity during the transition was also essential. By making changes behind the scenes, the team ensured that users continued to receive a stable experience.