Introduction

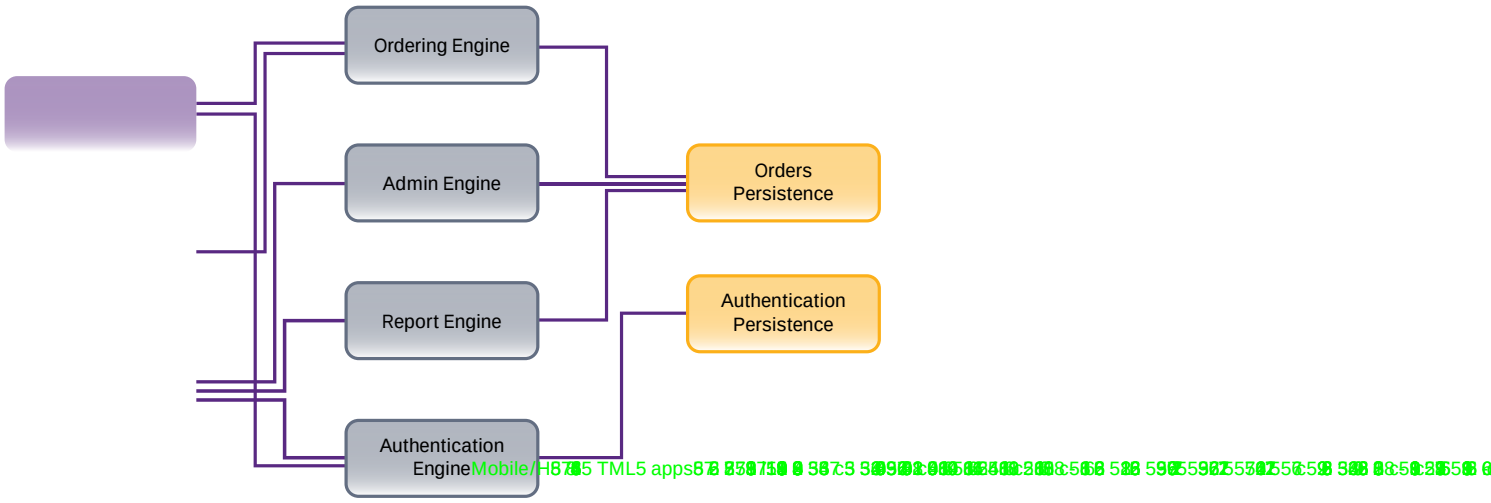
Technology merger and acquisition (M&A) due diligence demands a thorough and meticulous review of the target company's software architecture and code. This is especially critical in transactions where software is core to the valuation. Stakeholders require a shrewd evaluation not just of the code itself, but of the structure and architecture it's built on: a Design Quality Audit. This is a particularly daunting task given that the architecture rarely looks the same as the original design after months and years of development effort. But failure to perform adequate analysis can leave an organization unaware of the potential technical debt it's inheriting, and that can have significant ROI implications, because remediation efforts add time and money and directly impact future innovation.

The complexities of M&A due diligence

Manual code review provides key information about the software, but it doesn't address how code is structured and the implications on futurimctio.3 (cn/GS1 gs/TT2 1 Tf9 8li-nptd caf |c ofTdhontlim4 -)]T0 if buttoTd[debhb vthobi)9.1 c-4.89f4 v9 (ganizatioform a

Key findings that support the need for code review

Synopsys auditors have examined the software systems and development practices of hundreds of companies and have observed certain trends emerging over the years. An important finding is that high-level design elements rarely tell the whole story. How this high-level architecture is translated by the developers into working pieces of code, and how the codebase evolves over time, can make or break the stability and performance of a system.



A key factor is that software systems are constantly being updated for various reasons including:

- Changing or new requirements
- Growth in demand
- Security threats
- Upgrades to third party components
- Vendor changes and other challenges

Over time, as the implementation details stray from the original intended architecture, the pictures shared by the target of the current system become increasingly inaccurate. Developers are often on tight deadlines to ship changes, and they are forced to skip best practices like code reviews, automated testing, or secure programming activities. Lack of testing and consistency leads to codebases that are hastily patched together. The codebase becomes messy and brittle, and testing messy code that is highly interdependent can be challenging. Even more challenging is maintaining a codebase that breaks with even minimal change.

How Black Duck Design Quality Audits work

A Black Duck Design Quality Audit can help measure the modularity of a system and identify problematic components. It provides insight at the source code level and assesses maintainability of the system as well as the underlying components. It's important to note that this differs from a code quality audit, which is focused on the bugginess of the code rather than how the code itself is structured. A code quality audit is also important, but it doesn't address how the structure of the code might impede developer productivity. Using Silverthread's CodeMRI tool, Black Duck scans the software and analyzes relationships between the code components, identifying highly interdependent pieces of code that can be difficult to maintain and enhance. These clusters of problematic files (referred to as critical cores) are often the root cause of persistent maintenance problems in a codebase. The more cyclical dependencies a codebase contains, the higher the chances are of defects being introduced as developers work on it.

Choosing Black Duck to ease M&A stress

Clients faced with the difficult task of evaluating a target company's software systems should adopt a multipronged approach to analysis. Manual reviews should be complemented with an external audit of the architecture and processes, as well as a deep dive into the code. A Black Duck Design Quality Audit offers expert experience in these complex systems. It illuminates hidden trouble spots in the codebase and determines if critical components are well-designed or if they contain unhealthy areas that should be reviewed. Addressing these problem zones will lead to reduced defect rates and minimize roadblocks ttecwith ahsat

The Synopsys difference