

Одно из самых главных в работе архитектора - убедиться, что все НФТ и арх.значимые требования выполняются системой.

Но как это проверить?

Для этого придумали **Quality Scenarios (Сценарии проверки качества)**.

Сценарии качества, или Quality Scenarios, — это эффективный инструмент для анализа нефункциональных требований (НФТ) к системе. Они представляют собой конкретные ситуации использования системы, описанные с точки зрения интересующих нас аспектов качества, таких как производительность, надежность, доступность, масштабируемость, безопасность и других.

Основное назначение сценариев качества — преобразовать абстрактные и общие требования в конкретные и измеримые. Это позволяет архитекторам и разработчикам лучше понять, что именно требуется от системы, и позволяет тестировщикам определить, соответствует ли система этим требованиям.

Сценарии качества очень важны на этапе проектирования и реализации архитектуры системы. Они помогают архитекторам принимать обоснованные решения, учитывая нефункциональные требования, и выбирать подходящие технологии и архитектурные паттерны.

Проверка соответствия архитектуры системы ранее обозначенным НФТ важна по нескольким причинам:

- Подтверждение достижения целей: Проверка позволяет удостовериться, что реализованная система соответствует всем заявленным требованиям и способна достичь поставленных целей.
- Выявление проблем на раннем этапе: Если проверка проводится на ранних этапах разработки, возможно обнаружить и устранить проблемы до того, как они станут критическими.
- Соответствие ожиданиям пользователя: Система должна соответствовать ожиданиям пользователя по качеству и производительности. Если архитектура не отвечает НФТ, пользовательский опыт может быть негативным.
- Обеспечение долгосрочной устойчивости: Проверка НФТ помогает обеспечить, что система будет устойчивой и надежной в долгосрочной перспективе, с учетом возможного увеличения нагрузки и изменения требований.

Поэтому, разрабатывая архитектуру системы, важно не только ориентироваться на функциональные требования, но и учитывать и проверять нефункциональные требования, включая качественные сценарии.