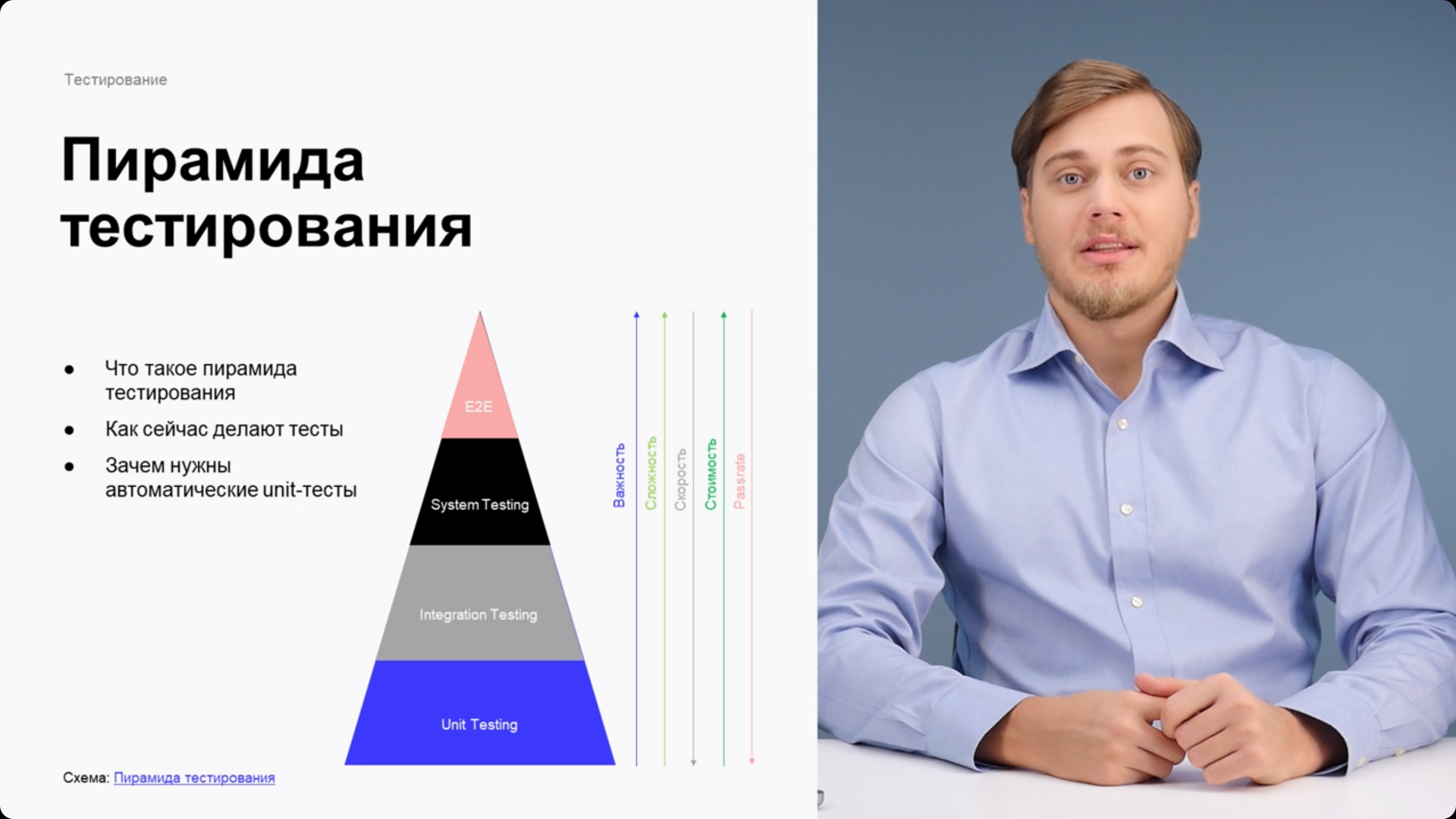
**10.1 Пирамида тестирования**



Пирамида тестирования – это модель, которая представляет собой иерархическую структуру тестирования, где основой являются Unit Testing и автоматизированные тест интеграции, а верхним уровнем являются ручные тесты и тесты пользователей.

Unit тесты - это тесты, которые используют для проверки отдельных компонентов или модулей программного обеспечения, т.е. unitы. Юнитами могут быть как функции так и классы или даже модули, тестирование которых можно проводить отдельно.

Автоматические тесты интеграции – это тесты, которые проверяют интеграцию различных компонентов или модулей программы между собой. Они проверяют, что тесты работают вместе как ожидается и корректно взаимодействуют между собой.

Идея пирамиды тестирования состоит в том, чтобы использовать больше автоматических тестов на нижнем уровне, чтобы повысить скорость тестирования и меньше ручных тестов на верхнем уровне, чтобы сохранить ресурсы.

Как сейчас пишут тесты? Юнит тесты и интеграционные тесты могут быть смешаны на разных этапах разработки. Общая стратегия состоит в том, чтобы сначала написать юнит тесты для отдельных компонентов системы, будь то функция или класс для того, чтобы проверить как эти компоненты работают в изоляции. А затем написать интеграционные тесты для того, чтобы проверить как эти компоненты работают вместе.

Интеграционные тесты могут быть разделены на несколько категорий в зависимости от того какие компоненты системы они проверяют. Например, и.тесты могут проверять взаимодействие компонентов внутри системы, а другие взаимодействие между различными системами.

Стоит отметить, что смешивание юнит тестов и и.тестов может зависеть от конкретных ситуаций, а также будет отличаться в различных компаниях и командах разработки.

Важно понимать, что нет строгих правил по смешиванию юнит тестов и и.тестов, поэтому нужно отталкиваться от потребностей проекта и конкретных ситуаций.

Юнит тесты являются важным инструментом для разработчика, т.к. позволяют проверять корректность кода на индивидуальном уровне изолированно от других частей системы. Это позволяет разработчику уверенно изменять код зная, что функциональность не будет нарушена. Ведь если функции или другие элементы системы изменят своё поведение, то юнит тесты не пройдут, а упадут, что позволит разработчику заметить ошибку и исправить её до того, как код пройдёт дальше.

Часто автотесты встроены в процесс разработки и/или интеграции, поэтому они могут быть запущены автоматически при публикации новой версии кода или даже при запуске кода на локальной машине, что позволяет найти исправить ошибку на ранних этапах разработки.

Юнит тестирование позволяет сохранять стабильность кода и поддерживать более высокое качество кодовой базы.