**20. Верификация и аттестация ПО**

**Верификация** — это процесс проверки, правильно ли реализовано программное обеспечение относительно спецификаций и требований.  
Проще говоря — **делаем ли мы продукт правильно?**

* Проверяется соответствие разработанного кода и архитектуры документированным требованиям.
* Обычно включает:
  + Инспекции кода.
  + Ревью дизайна.
  + Статический анализ.
  + Юнит-тестирование.
* Пример: проверка, что модуль выполняет заложенный в ТЗ алгоритм

**Аттестация ПО** — это более общий термин. Это официальное подтверждение, что продукт соответствует заданным стандартам, нормативам и может применяться по назначению.  
Можно сказать, что аттестация — это форма сертификации или формализованного приёмочного тестирования.  
Она часто применяется в:

* Критичных областях (авиация, медицина, оборонка).
* Госзаказах.
* Разработке ПО с высокими требованиями безопасности.

Обычно включает:

* Итоговое тестирование (валидация).
* Проверку всей документации.
* Аудит процессов разработки.
* Выдачу официального документа (свидетельства о допуске к эксплуатации).

**1. Требование** – описание того, что система должна делать, какими свойствами обладать и каким ограничениям соответствовать.

Классификация требований:

* Функциональные – что система должна делать (например, система должна позволять пользователям регистрироваться)
* Нефункциональные – как система должна это делать (дизайн, бд)
* Бизнес-требования – высокоуровневые цели организации
* Пользовательские требования – что пользователи хотят получить от системы
* Системные требования – детальное описание функций системы

Уровни требований – иерархия требования к проекту, включает три уровня:

1. Верхний: бизнес-требования – цели и задачи бизнеса, которые проект должен решить, описывает, почему проект создан и какие бизнес-выгоды от него ожидаются (например: цель компании «увеличить продажи на 20%»)

2. Второй: пользовательские требования – потребности пользователя, как система должна помочь в решении задач (use-cases «покупатель может оплатить заказ»

3. Третий: системные требования (функциональный) – детализированные требования, описывают как система должна функционировать (функциональные и нефункциональные требования) (иные технические спецификации)

Методологии

Некоторые подходы к управлению требованиями:

-Agile — итеративный подход, подчёркивающий адаптивность, непрерывную обратную связь и сотрудничество заинтересованных сторон. Требования постоянно уточняются, что позволяет командам быстро реагировать на изменения. Подходит для проектов с меняющимися или неопределёнными требованиями.

-Waterfall — линейный, последовательный подход, проходящий через отдельные фазы: сбор требований, проектирование, внедрение, тестирование и развёртывание. Каждая фаза должна быть завершена перед переходом к следующей. Подходит для проектов с чётко определёнными требованиями с самого начала, например, в регулируемых отраслях, где обширная документация и стабильность имеют решающее значение.

-Гибридный подход — сочетает элементы Agile и Waterfall, обеспечивая гибкость для команд, которым нужна структура в некоторых областях, но адаптивность в других. Обычно подход Waterfall применяется на ранних этапах проекта, в то время как Agile используется во время разработки, тестирования и текущих циклов обратной связи.