

## Сравнение процесса тестирования в водопадной модели и SCRUM

**Водопадная модель** предполагает однократное выполнение каждой из фаз проекта, которые, в свою очередь, строго следуют друг за другом. По структуре она похожа на жизненный цикл ПО (разработка требований, проектирование, реализация, тестирование, эксплуатация и поддержка).

Тестирование в явном виде появляется лишь с середины развития проекта, когда завершены предшествующие этапы, и достигает своего максимума в самом конце. И в данном случае речь идет о всем цикле тестирования от разработки стратегии и плана тестирования до отладки ПО и завершения тестирования, то есть модель не предусматривает хотя бы разработку тестовой документации, пока осуществляется процесс разработки. Тут можно выделить два следствия: 1) Это является недостатком водопадной модели с точки зрения тестирования, потому как всем известно, что чем позже выявлен дефект, тем дороже он будет стоить; 2) Фактически тестировщики “простаивают” в течение предыдущих этапов, занятость является неравномерной в рамках жизненного цикла одного проекта.

Преимуществом для тестировщиков можно считать то, что данная модель предполагает наличие стабильных, четко сформулированных требований перед началом разработки, и у каждой стадии есть четкий проверяемый результат. В отличие от ситуаций с отсутствием требований или их низким качеством, не надо тратить время (свое и других участников команды) на воссоздание необходимой документации - требований, критериев готовности и других.

**Scrum** представляет собой фреймворк в рамках гибкой методологии Agile, который определяет роли (Product Owner, Scrum Master, Development Team), события (Sprint Planning, Daily Scrum, Sprint Review, Sprint Retrospective) и артефакты (Product Backlog, Sprint Backlog, Increment). Scrum организуется в виде спринта, представляющего собой короткую рабочую сессию (до 4 недель, чаще всего 2 недели), по итогам которой необходимо достигнуть заранее заданных результатов.

В Scrum тестирование интегрируется во все спринты. Тестировщики входят в состав Development Team (команды разработки), при этом команда воспринимается как единое целое. Тестировщики выполняют запланированные задачи и коммуницируют с другими участниками команды для достижения целей спринта. В частности, совместно с командой определяется стратегия, виды и инструменты тестирования, также тестировщики участвуют в событиях спринта. В связи с этим подходом для тестировщиков на первый план выходят софт-скиллы. Например, в отличие от каскадной модели, в быстро меняющихся условиях документацией могут пренебрегать, и тестировщику для качественного выполнения работы необходимо взаимодействовать с командой, изолированно он маловероятно сможет качественно выполнять свою работу.

**Отличия тестирования в указанных методологиях:**

<b>SCRUM</b>	<b>Waterfall</b>
Тестирование осуществляется в процессе всего спринта	Тестирование является отдельным этапом после завершения предыдущих
Документация может быть не достаточна, нужен дополнительный сбор требований	Исчерпывающее документирование проекта благоприятно для тестировщиков
В разработке требований участвуют тестировщики	Тестировщики могут быть задействованы в разработке требований, а могут и не быть
Команда тестировщиков и команда разработчиков работают совместно	Команда тестировщиков и команда разработчиков работают отдельно
Процессы разработки и тестирования скоординированы	Процессы разработки и тестирования разделены
Тестирование параллельно разработке	Тестирование после разработки
Тестируется каждый модуль после его разработки (плюс регресс)	ПО тестируется сразу целиком
При необходимости тестировщики взаимодействуют с заказчиком	Тестировщики не взаимодействуют с заказчиком