TDD, или Test-Driven Development (Разработка, управляемая тестами), - это методология разработки программного обеспечения, которая подразумевает создание тестов для функциональности ПО до того, как эта функциональность будет фактически реализована.

*Этапы:*

1. Создание тестов

2. Запуск тестов

3. Реализация функциональности

4. Запуск тестов снова

5. Рефакторинг

6. Повторение цикла

BDD, или Behavior Driven Development (Разработка, ориентированная на поведение), - это методология разработки программного обеспечения, которая сосредотачивается на описании поведения программы с точки зрения её пользователей и интересующих сторон:

1. Описание поведения

2. Описания сценариев

3. Автоматизация сценариев

4. Итерации и разработка

5. Совместная работа команды

6. Создание документации

7. Постоянная проверка соответствия

*Этапы разработки ПП:*

1. Проектирование;

2. Реализация базовых функций

3. Разработка pre-alpha-версии

4. Разработка alpha-версии

5. Разработка pre-beta-версии

6. Разработка beta-версии

7. Замораживание пользовательского интерфейса

8. Подготовка к финальному тестированию

9. Последняя проверка целостности

10. Выпуск

11. Организация сопровождения.

*При проектировании:*

1. анализ предметной области

2. создание спецификаций (ТЗ)

3. предложения по программной реализации

4. формулирование требований по внутреннему интерфейсу

5. формулирование требований по внешнему дизайну

*Тестирование на этапе проектирования:*

1. изучение будущего ПП и предметной области

2. анализ спецификаций

3. анализ пользовательских данных

4. анализ предполагаемого пользовательского интерфейса

5. разработка эскизного плана тестирования

6. тестирование купленных программных и аппаратных инструментальных средств

7. обоснование потребности в дополнительных программных и аппаратных средствах

*Реализация базовых функций. Разработка ПП на данном этапе включает:*

1. Постановку локальных задач
2. создание основных программных модулей
3. тестирование модулей

*Тестирование на этапе реализаций базовых функций:*

1. неформальное тестирование
2. элементы формального тестирования
3. подготовка БД ошибок

*Pre-alpha. Цели и задачи разработки пре-альфа-версии:*

1. Исследование концепции

2. Прототипирование

3. Техническое исследование

4. Получение обратной связи

*Тестирование на этапе Pre-Alpha:*

1. анализ окончательного варианта спецификации
2. разработка первого рабочего варианта плана тестирования
3. разработка основного ядра ручных тестов ПП
4. выполнение базового тестирования ПП
5. постановка задач в области автоматизации тестирования
6. обоснование полной или частичной автоматизации тестирования

*Разработка Alpha версии:*

1. дальнейшее программирование и отладку модулей

2. тестирование ПП

3. разработку документации, например, руководства оператора (пользователя), руководства системного программиста и т.д.

4. исправление ошибок

5. создание недостающих драйверов устройств

6. начало разработки контрольного примера для приемо-сдаточных испытаний

*Разработка Beta версии:*

1. завершение отладки функций

2. исправление подавляющего большинства найденных ошибок

3. создание или доработку установочных утилит

4. разработку дополнительных контрольных примеров

5. редактирование пользовательского интерфейса

6. подготовку дисков и файлов для бета-тестировщиков

*Тестирование на этапе Beta:*

1. утверждение окончательного тестового плана
2. продолжение работ по автоматизации тестирования
3. выработка рекомендаций по редактированию программной
4. тестирование инсталляции, деинсталляции и автообновления программы
5. полный цикл тестирования ПП
6. быстрое повторное тестирование исправленных фрагментов программы
7. тестирование на virgin-машине
8. тестирование внутри и вне компании
9. оценка результатов выполнения календарного плана

*Замораживание пользовательского интерфейса:*

1. вносятся только изменения, не отражающиеся на пользовательском интерфейсе

2. исправляются ошибки

3. вносятся изменения, связанные с повышением производительности программы

4. реализуется окончательный вариант контрольных данных

5. реализуется окончательный вариант утилит инсталляции, деинсталляции, автообновления.

*Тестирование на этапе «Замораживание пользовательского интерфейса»*

1. многократное выполнение регрессионных тестов
2. тщательное выполнение тестового плана
3. поиск наиболее заметных ошибок
4. окончательное завершение работ по автоматизации тестирования
5. разработка документов

*Подготовка к финальному тестированию:*

1. полный цикл тестирования по тестовому плану
2. создание списка отложенных проблем
3. оценка надежности программы

*Проверка целостности:*

1. исправление ошибок

2. подготовка демонстрационных материалов

3. архивация исходных кодов

4. завершение работ над материалами для приемо-сдаточных испытаний

*Работы по тестированию на этапе «Проверка целостности»*

1. тестирование в реальном режиме
2. анализ плана тестирования и ошибок
3. подготовка носителей для размножения
4. проверка на вирусы
5. полная архивация всех материалов

Этап «Выпуск». Итоговая документация:

1. тестовый план

2. тесты (с описанием всех тестовых случаев)

3. отчеты на основании проверочных листов

4. отчёты на основании БД ошибок

5. итоговый отчет по результатам тестирования

Сопровождение. Корректирующее сопровождение:

1. в короткие сроки научиться воспроизводить обнаруженные дефекты по описанию пользователя, а также помочь программистам максимально локализовать причину их возникновения
2. выполнить все тесты ПП после исправления дефекта

Сопровождение. Улучшающее сопровождение:

1. разработка дополнительных тестов, предназначенных для проверки новых функций
2. проведение полного цикла тестирования ПП
3. корректировка тестовой документации