Процессы жизненного цикла проекта представляют собой последовательность этапов и действий, необходимых для успешного выполнения проекта. Использование этих процессов зависит от типа проекта, его характеристик, методологии и практик, применяемых в организации. Вот общий обзор наиболее часто используемых и менее часто используемых процессов в реальных проектах:

Наиболее часто используемые процессы:

Инициация проекта:

Процесс начальной фазы, в ходе которой определяются цели, объем и ограничения проекта. Он широко используется, так как устанавливает основы для последующих этапов и обеспечивает четкое понимание целей.

Планирование проекта:

Этот процесс включает разработку плана проекта, включая график, бюджет, ресурсы и качество. Планирование проекта неотъемлемая часть успешного выполнения проекта.

Исполнение проекта:

Этот этап включает в себя непосредственное выполнение работ согласно плану проекта. Он всегда присутствует в каждом проекте, поскольку здесь осуществляется основная работа.

Мониторинг и контроль:

Процесс активного наблюдения за выполнением проекта, выявления отклонений от плана и коррекции сроков и бюджета. Это важно для обеспечения качественной реализации проекта.

Завершение проекта:

Фаза, на которой проект формально завершается, результы передаются заказчику, и учатся уроки из проекта. Этот процесс важен для оценки успеха и для обратной связи.

Управление изменениями:

Управление изменениями помогает в учете изменяющихся требований и условий проекта. Это важно для адаптации к изменениям во время выполнения проекта.

Менее часто используемые процессы:

Управление рисками:

Некоторые проекты пренебрегают управлением рисками, особенно если считают, что риски невелики или предсказуемы. Однако, управление рисками может быть критически важным для предотвращения непредвиденных проблем.

Управление стейкхолдерами:

В некоторых проектах не уделяется должного внимания управлению стейкхолдерами и коммуникациями с ними, что может привести к недопониманию и конфликтам.

Управление качеством:

Некоторые проекты могут сокращать или пренебрегать процессом управления качеством из-за ограничений бюджета и времени. Однако, это может повлечь за собой проблемы в будущем.

Выбор использования тех или иных процессов зависит от специфики проекта, его бюджета, сроков и критической важности. Важно адаптировать процессы к уникальным потребностям и рискам каждого проекта, чтобы достичь успешных результатов.

| **№ п/п** | **Название стадии (этапа)** | **Содержание работ** | **Результат работ** |
| --- | --- | --- | --- |
| 1 | Планирование | Определение целей и требований к игре, создание концепции, разработка плана проекта с бюджетом и графиком | Документация с планом разработки, концепция игры |
| 2 | Проектирование | Разработка детального дизайна игры, включая характеристики персонажей, уровни, механику, арт-концепты, анимацию и звуковое сопровождение | Документация с дизайном игры, арт-активы, прототипы |
| 3 | Разработка | Создание игровых элементов, программирование игровой логики, графики, анимации, музыки и звуков | Играбельный прототип или версия игры |
| 4 | Тестирование | Тестирование игры, выявление и регистрация багов, анализ производительности и сложности игры | Отчеты о багах, исправленные ошибки |
| 5 | Оптимизация | Оптимизация игры для различных платформ и устройств | Оптимизированная версия игры |
| 6 | Релиз | Выпуск игры на выбранных платформах, маркетинговая кампания, создание пресс-релизов и рекламных материалов | Игра доступна для скачивания или покупки |
| 7 | Поддержка | Поддержка игры после выпуска, исправление багов, выпуск обновлений, добавление нового контента, взаимодействие с сообществом игроков | Регулярные обновления и поддержка |

Созданная мной модель жизненного цикла создания игры напоминает агильную методологию разработки, особенно с учетом современных подходов к разработке игр. Агильные методологии, такие как Scrum и Kanban, поддерживают итеративное развитие и гибкий подход к управлению проектом.

В данной модели:

1. **Планирование** напоминает "Спринт-планирование" в Scrum, где определяются цели и план разработки для конкретного периода времени.
2. **Проектирование** соответствует разработке дизайна и документации, которые часто выполняются в начале каждого спринта.
3. **Разработка** включает в себя создание игровых элементов и программирование, что является основной работой во время спринтов.
4. **Тестирование** подразумевает проведение тестов и выявление ошибок, что также интегрировано в процесс разработки в рамках агильных методологий.
5. **Оптимизация** подходит для постоянной оптимизации проекта в рамках агильного подхода.
6. **Релиз** отображает выпуск продукта по завершении спринта или его части.
7. **Поддержка** включает в себя поддержание и обновление игры после выпуска, что также обычно учитывается в агильных проектах.

Тем не менее, важно отметить, что жизненный цикл разработки игр может варьироваться в зависимости от конкретного проекта и команды разработчиков. Некоторые игры могут предпочесть более традиционные подходы, такие как водопадная методология, особенно если проект крупный и требует строгого планирования.

Для разработки программного модуля "Учет успеваемости студентов" рассмотрим подходящую модель жизненного цикла (ЖЦ) информационной системы (ИС) и обоснуем выбор.

**Скрам (Scrum)** - агильная модель жизненного цикла ИС.

**Обоснование выбора**:

1. **Четность номера зачетки: 1**. Учитывая, что номер зачетки четный, мы рассматриваем агильную модель Scrum.
2. **Характеристики проекта**:
   * Проект связан с оперативным учетом успеваемости студентов и предполагает долгосрочное хранение данных об успеваемости на протяжении всего срока обучения.
   * Требуется оперативность и гибкость внесения изменений в ИС, так как в процессе обучения и учета могут возникать новые требования и корректировки.
   * Используется внешний и внутренний мониторинг и обратная связь от студентов и преподавателей, что подразумевает необходимость быстрого реагирования на изменения и потребности пользователей.
3. **Обоснование выбора**:
   * Scrum подходит для проектов, в которых требуется гибкость и возможность реагировать на изменения требований в процессе разработки. В данном случае, поскольку требования могут меняться или уточняться в зависимости от образовательных процессов и потребностей студентов, Scrum позволяет вносить коррективы в ходе разработки.
   * Scrum способствует внесению обратной связи и участию пользователей (студентов, преподавателей) в процессе разработки, что полезно для сбора требований и выявления проблем оперативно.
   * Методология Scrum позволяет работать в коротких итерациях (спринтах), что способствует оперативности и проверке результатов на ранних этапах.

Основываясь на указанных характеристиках проекта, агильная модель Scrum представляется наиболее подходящей для разработки программного модуля "Учет успеваемости студентов", так как она позволяет эффективно управлять изменениями, включать пользовательскую обратную связь и достигать оперативности в реализации проекта.