## Команда: TEAM508



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Версия | Изменения | Дата | Автор | Подпись |
| 1.0 | Модель жизненного цикла | 14.10.2016 | Двуреченский Андрей, Корнеев Владислав |  |

**Сравнение методологии SCRUM с KanBan**

|  |  |
| --- | --- |
| Scrum | KanBan |
| заранее определены ячейки для записи задач для каждой итерации | ячейки опциональны |
| команда встречается для выделения задач на каждую итерацию | встречи команды не обязательны |
| задачи должны быть перенесены, каждая задача должна быть рассчитана на выполнение в течение 1 спринта | задачи могут быть разными по продолжительности. Ограничений нет |
| четко определенный шаблон для переноса задач с тремя столбцами (сделать, выполняется, сделано) | нет заданного визуального шаблона |
| нет ограничения по количеству работ на 1 спринт | есть ограничение по количеству работ в один временной промежуток |
| в текущую итерацию нельзя добавить новую задачу | в любой момент можно добавить новую задачу |
| все задачи по проекту выполняются одной специальной командой согласно заранее определенным приоритетам | с канбан-доской могут работать разные команды или люди, приоритезация по желанию |
| обязательны три роли: Владелец проекта, Скрам-мастер, Команда проекта | заранее определенных обязательных ролей нет |

Также были рассмотрены еще некоторые методологии. Сводная таблица приведена ниже.

**Сводная таблица методологий проектирования ПО**

|  |  |  |
| --- | --- | --- |
| Название модели | Статус | Причина |
| **Scrum** | Отклонен | Требует регулярных встреч и обсуждений, нельзя вносить изменения во время итерации, не желательно, чтобы задачи были разные по продолжительности. |
| **Xtreme Programming** | Отклонен | Постоянный рефакторинг кода. Основная документация- это тщательно прокомментированный код. Эффективное использование возможно только в команде из опытных разработчиков, что противоречит команде- студентов. |
| **KanBan** | **Одобрен** | Не требует регулярных встреч, можно управлять скоростью разработки на каждом этапе, так как  количество параллельных задач ограничивается заранее на каждом этапе, существует визуализация процесса, позволяет вносить изменения во время каждой итерации, также задачи могут быть разными по продолжительности. |
| **Rapid Application Development** | Отклонен | В RAD-модели компоненты или функции разрабатываются несколькими высококвалифицированными командами параллельно, будто несколько мини-проектов. Временные рамки одного цикла жестко ограничены. Созданные модули затем интегрируются в один рабочий прототип. На этапе интеграции могут возникать проблемы для команды – студентов. |
| **Rational Unified Process** | Отклонен | Обладает недостаточным уровнем формализма, что может приводить к рассогласованию в команде неопытных разработчиков, затрате дополнительных ресурсов на переработку кода. RUP наиболее часто используют компании с большим количеством разработчиков (свыше 40-50 человек). |
| **Microsoft Solution Framework** | Отклонен | Методология не подходит, так как очень громоздкая и на поддержку процессов уходит слишком много ресурсов. В основном применяется в корпорации Microsoft. |