

*Кургуз Михаил Андреевич*

студент

Научный руководитель

*Копелиович Дмитрий Игоревич*

канд. техн. наук, доцент

ФГБОУ ВО «Брянский государственный технический университет»

г. Брянск, Брянская область

## **ИНСТРУМЕНТЫ УПРАВЛЕНИЯ ПРОЕКТАМИ ДЛЯ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЙ**

*Аннотация:* в статье ставится актуальный вопрос о том, как эффективно управлять командой разработки для достижения целей продукта, анализируются основные подходы к управлению проектами. На основе лучших практик, рассмотренных методологий формируется свой поэтапный подход с описанием, который эффективен в рамках разработки конкретного продукта.

*Ключевые слова:* эффективное управление командами разработки, каскадная модель, инкрементная модель, итерационная модель, журнал продукта, журнал итерации, ретроспектива.

Эффективное управления командой разработки является очень важным аспектом для любой IT-компании, которая занимается разработкой программного обеспечения. С учетом быстрого темпа роста технологических изменений, увеличения конкуренции и требований рынка, правильное управление командой становится ключевым фактором успеха проекта.

Актуальность темы эффективного управления проектами обусловлено следующими факторами.

1. Рост сложности проектов. Современные проекты становятся все более сложными и масштабными, что требует гибкого сотрудничества между участниками команды, правильного распределения задач и контроля за выполнением работ.

2. Глобализация. Команды разработчиков могут находиться в разных частях страны, что создает дополнительные трудности в коммуникации с участниками команды.

3. Требования рынка. Быстрое время выхода на рынок, высокое качество продукта и удовлетворения потребностей клиентов становятся основными приоритетами. Эффективное управление командой помогает достичь данных целей, оптимизируя процессы разработки.

Таким образом, актуальность проблемы эффективного управления командами разработки неоспорима. Руководители проектов должны постоянно совершенствовать свои навыки управления и использовать правильные подходы к управлению, следить за трендами в области разработки ПО и стремиться к созданию профессиональных и сплоченных команд разработчиков.

Начнем с того, что существует два основных подхода при разработке ПО: каскадная модель (водопад) и инкрементная модель [1]. Главное различие между двумя этими подходами заключается в том, что в каскадной модели все этапы разработки идут последовательно и каждый последующий этап не начинается до полного завершения предыдущего. Кроме того, в модели «водопад» завершенность этапа обуславливается пакетом документации для данного этапа. В инкрементной модели разработка ведется по спринтам – временным отрезкам и по завершении каждого спринта мы получаем рабочую версию продукта, которая улучшается функционально с каждым спринтом. На рис. 1 представлены каскадная и инкрементные модели разработки программного обеспечения.

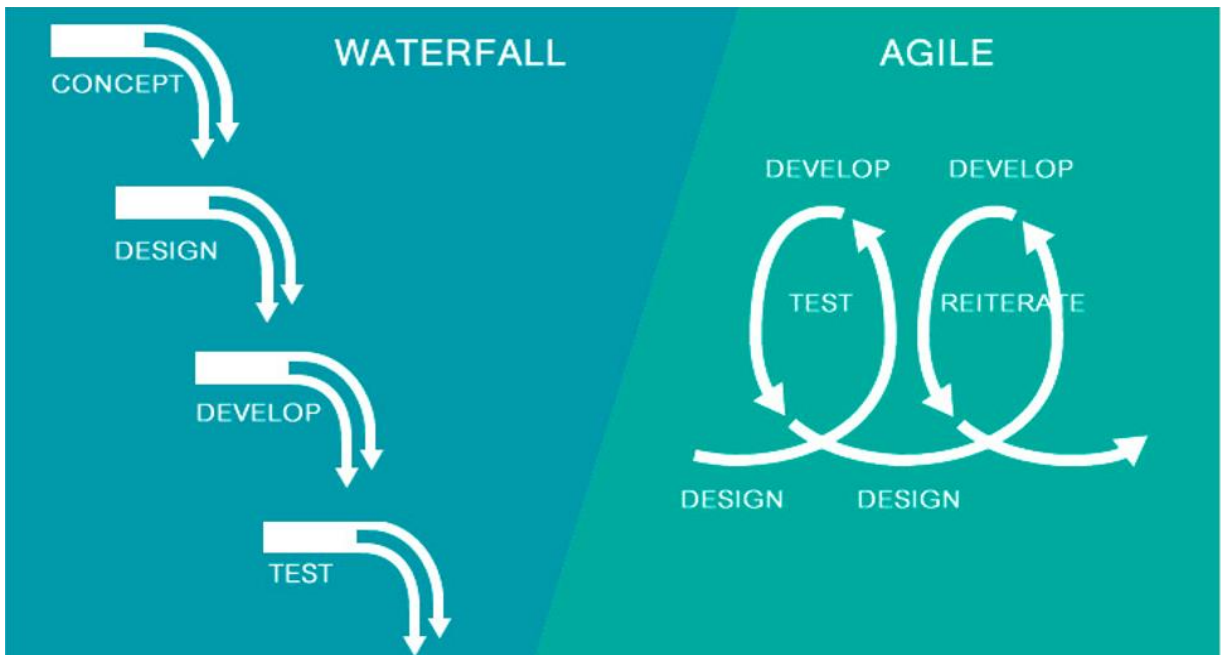


Рис. 1. Отличие каскадной и инкрементной модели

Необходимо также различать итерационный подход к созданию продукта от инкрементного (рис. 2). При использовании итерационного подхода мы собираем продукт по частям, но заказчику или конечному пользователю предоставляем готовый продукт только на последней фазе. При инкрементном подходе на каждом этапе поставляется минимально жизнеспособный продукт, который может оценить заказчик или пользователь, что является важным для получения обратной связи и выстраивания нового спринта разработки.

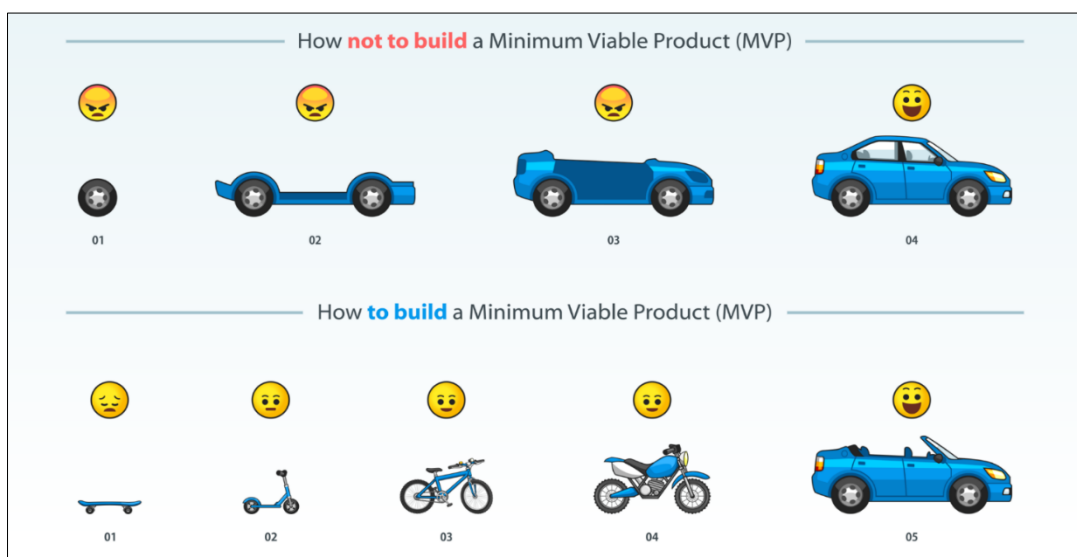


Рис. 2. Отличие инкрементного подхода от итерационного

Еще одним важным преимуществом использования инкрементного подхода к управлению проектом является низкая цена исправления ошибок. На рис. 3 отражена зависимость стоимости исправления ошибок при обнаружении дефектов на этапе жизненного цикла продукта от времени. График показывает, чем раньше мы получили обратную связь от заказчика, тем ниже цена исправления ошибок. В связи с этим, данная методология реагирует таким образом, что мы делаем небольшой фрагмент по всем фазам разработки сразу и выдаем конечному пользователю, тем самым получая инкремент.

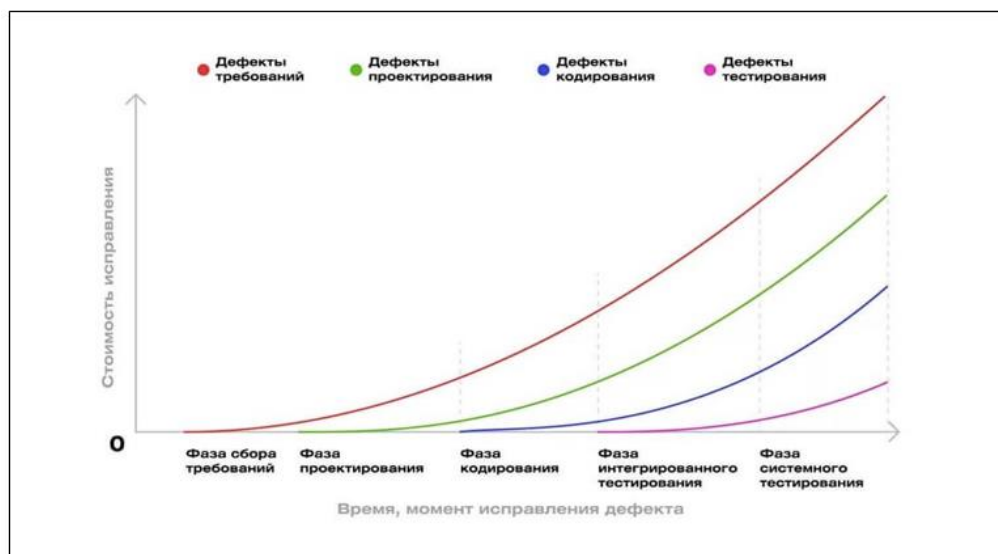


Рис. 3. Цена исправления дефекта в зависимости от фазы проекта

Учитывая преимущества инкрементного подхода, будем использовать его основные принципы на практике. На рис. 4 представлена модель разработки веб-приложения с учетом использования инкрементного подхода.

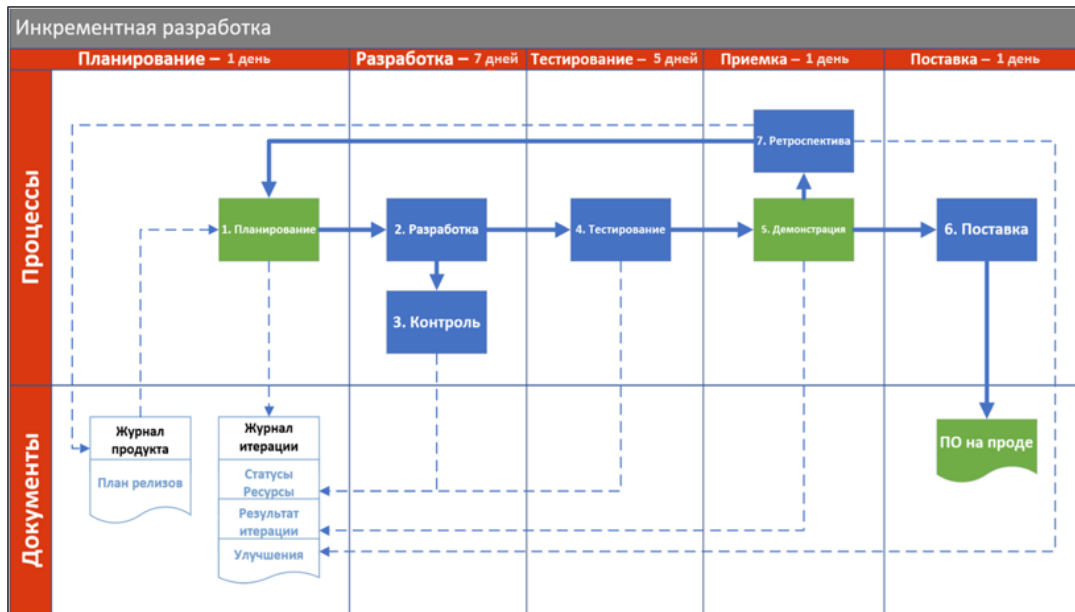


Рис. 4. Модель инкрементной разработки

Модель разделена слева на процессы и документы, которые формируются в результате процессов, сверху на временной отрезок. На данной модели синим выделены процессы, в которых участвует команда, зеленым выделены процессы, в которых участвует заказчик. Точкой входа является сформированный журнал продукта, который включает в себя план релизов (итераций). На основании журнала продукта проводится планирование и формируется начальная версия журнала итерации – те задачи, которые планируется выполнить в рамках итерации. По мере выполнения процессов журнал итерации пополняется данными о статусе задач, исполнителях, времени на выполнение.

Разберем более подробно структуру артефактов (документов), их особенности и назначение.

Как было сказано ранее, точкой входа в процесс разработки является журнал продукта. На рис. 5 показан пример ведения журнала продукта, вести его можно по-разному и в разных системах трекинга задач. Особенностью журнала является список всех функций для реализации, которые известны в текущий момент времени. Как видно на рисунке все функции сгруппированы в логические блоки, кроме того, чем выше находится блок, тем он более приоритетен в разработке. На основе общего списка задач формируется план релизов (итераций), как правило протяженностью две недели. Каждый релиз имеет четко сформулированные цели, которые

он преследует, а также набор блоков функций, которые планируется реализовать для достижения цели в данном временном отрезке. Таким образом, весь журнал продукта предварительно делится на N итераций с датами поставки на продуктовый сервер и целями, которые планируются достигнуть.

Релиз 2023.1.1. от 07.04.23	Релиз 2023.2 от 13.04.23	Журнал продукта
Цель 1. Исправление замечаний после выпуска релиза 2023.1	Цель 1. Улучшить интерфейс для импорта отчётов и формирование проектов ОПР Цель 2. Завершение работ по интерфейсу ПРМ из нескольких отчётов Цель 3. Завершение работ по улучшению внешнего вида существующей аналитики Цель 4. Разработка инфраструктуры для алгоритмов оптимизации СОУТ Цель 5. Разработка инструментов ручной оптимизации классов условий труда	
<b>ПНЗ. Сводный отчёт. Улучшение интерфейса</b>	<b>ПРМ. Общее для ПРМ 1 и ПРМ *</b>	<b>ПНЗ. Настраиваемая проверка идентификации факторов</b>
Убрать кнопку возврата к рабочим местам	Починить общую галку, чтобы не было мерцания	Разработать возможность редактирования настройки
Название переместить в М4	Дополнительные столбцы выровнять по середине	Разработать возможность перепроверки ПРМ с
"Тип" переименовать как "Тип ПНЗ"	Сделать горизонтальный скролл больше и более заметным	Проверить корректность алгоритмов ПНЗ для загруженных отчётов Полюс-Золото
Сделать на ошибке и предупреждении бейджи	В ПРМ 1 заменить "Отчёт" на "Рабочие места"	<b>ПРМ. Возможность подбора профессий</b>
Колонка 2 "Алгоритмы выявления признака нарушения законодательства"	Починить "Вид" --> "Показывать названия профессии"	Добавить автоподбор профессий для выбранных РМ
<b>ПРМ. Переформирование меню МЗ</b>	<b>Материалы. Улучшение внешнего вида</b>	Добавить выбор профессии через диалог для выбранных РМ
"Настройки" перенести вправо (не увидела на ПРМ в МЗ вообще кнопки с настройками)	В названии ПРМ убрать время и сделать дату формирования русского формата (дд.мм.гггг)	<b>Общее. Глобальный поиск</b>
Убрать пункты в "Настройках" (закомментировать)	<b>ПРМ *</b>	Сделать поддержку поисковых провайдеров для Материалов СОУТ
Заменить "Отчётные формы" на "Отчёт СОУТ" и переставить после "Рабочие места" (вторая кнопка)	Выводить раскрывашку только там, где нужно раскрывать	Сделать поддержку поисковых провайдеров для РМ
<b>ПРМ*</b>	В таблах с названиями отчётов в кружочке сделать порядковый номер в рамках ПРМ	<b>ПРМ. Умный фильтр</b>
В таблах с названиями отчётов в кружочке сделать порядковый номер в рамках ПРМ (В ТАБАХ В КРУЖОЧКЕ НЕ ПОРЯДКОВЫЙ НОМЕР)	<b>Материалы. Импорт отчётов</b>	Мощный фильтр рабочих мест

Рис. 5. Журнал продукта и план релизов

По окончании итераций может возникнуть ситуация, когда не все функции были успешно реализованы, либо заказчик внес корректировки в приоритет или работу тех или иных функций. В таком случае журнал продукта пополняется новыми задачами, более приоритетные блоки задач идут в ближайшие релизы, невыполненные задачи смещаются на один релиз. Важно отметить, что журнал продукта – это общедоступный документ, доступ к которому имеет как команда, так и заказчик, что позволяет своевременно принимать решения и вносить корректировки в процесс безболезненно.

№		Релиз 2023.4 от 04.05.23	Дополнительные работы
1	Объединение ПРМ. Структура файлов	Разработать структура WpList.xml	Устранение локальных проблем, связанных с получением версий проекта
2		Разработать структуру Modifications.xml	
3		Разработать структура Map.xml	
4		Разработать структура Filter.xml	Консультация коллег
5		Разработать структуру ProjectUT (Multiwplist.xml)	
6		Доделывание структур данных при необходимости	
7	Объединение ПРМ. Создание репозитория	Разработать MapRepository	Работы, связанные с обновлением сайта (сбор пакетов, выкатывание, настройка)
8		Разработать FilterRepository	
9		Разработать ModificationRepository	
10		Разработать структура wpListRepository	
11		Разработать структура ProjectUTrepository	
12	Объединение ПРМ. Алгоритмы	Разработать Алгоритм преобразования wpList в Map	Планирование и обсуждение деталей задачи (участие в звонках)
13		Разработать Алгоритм преобразования Map в wpList	
14		Разработать алгоритм преобразования Excel в wpList	
15		Разработать Алгоритм объединения rz в map	Устранение ошибок ранее рабочих функций
16		Разработать Алгоритм внутреннего объединения в Map	
17		Разработать алгоритм преобразования деклараций в wpList	
18		Разработать Алгоритм внутренних проверок Pnz	ПростоЙ в работе
19	Разработать алгоритм интегрирования протокола в wpList		

Рис. 6. Журнал итерации на стадии планирования

После формирования плана релизов наступает процесс планирования итерации, основу которого составляют блоки задач, принятые для разработки в текущем спринте, здесь же фиксируется точная дата релиза. Помимо основных блоков задач классифицируются дополнительные работы, которые могут возникнуть в процессе работы и затрачивать время исполнителей в рамках релиза. Классификация дополнительных работ может отличаться от примера, в зависимости от особенностей процесса разработки. На рис. 6 представлен пример журнала итерации на стадии планирования.

Следующим этапом является процесс разработки и контроля. Задачи распределяются по исполнителям, происходит оценка плановой трудоемкости задач. Данную процедуру можно осуществить с помощью метода «Planning Poker» [2]. Разработчики в течение рабочего дня изменяют статус своих задач, при этом у каждого исполнителя в момент времени находится всегда только одна задача в работе. Статусы задач могут классифицировать по-разному в зависимости от проекта. В конце рабочего дня происходит распределение рабочих часов между основными и дополнительными задачами (рис. 7).

Распределение рабочих часов исполнителей по задачам необходимо для того, чтобы проследить какую часть рабочего времени исполнитель тратит на



реальные задачи журнала итерации, а также для последующего построения диаграммы сгорания задач «Burn Down» [3].

№		Релиз 2023.4 от 04.05.23	Исполнитель	Статус	Статус к релизу	Плановая трудоемкость	25.04.2023	26.04.2023	27.04.2023
1	Объединение ПРМ. Структура файлов	Разработать структура WpList.xml	Вараксо	Готово	Готово	2	0,5	0,5	0,5
2		Разработать структуру Modifications.xml	Вараксо	Готово	Готово	2	0,5	0,5	0,5
3		Разработать структура Map.xml	Вараксо	Готово	Готово	2	0,5	0,5	0,5
4		Разработать структура Filter.xml	Вараксо	Готово	Готово	2	0,5	0,5	0,5
5		Разработать структуру ProjectUT (MultiWpList.xml)	Вараксо	Не в работе	Перенос	2	0,5	0,5	0,5
6		Доделывание структур данных при необходимости	Вараксо	Готово	Готово	2			
7	Объединение ПРМ. Создание репозитория	Разработать MapRepository	Вараксо	В работе	Перенос	2		1,5	0,5
8		Разработать FilterRepository	Вараксо	Тестирование	Готово	2		1	0,5
9		Разработать ModificationRepository	Вараксо	Тестирование	Готово	2		1	0,5
10		Разработать структура wpListRepository	Вараксо	Готово	Готово	2			0,5
11		Разработать структура ProjectUTRepository	Жемоедов	Готово	Готово	2			1

Рис. 7. Журнал итерации на стадии разработки и контроля

Когда текущая итерация подходит к завершению наступает процедура релиза. Основные действия команды в день релиза:

- в день релиза разработчики не берут в работу новые задачи;
- производится оценка задач со статусом «в работе» с точки зрения завершения ее в рамках данной итерации;
- задачи, которые не были взяты в разработку автоматически переносятся в следующий спринт с наивысшим приоритетом;
- сборка системы и обновление на продуктивном контуре;
- тестирование и устранение критических багов при наличии;
- демонстрация результатов итерации заказчику.

На этапе демонстрации руководитель проекта показывает заказчику разработанные функции в рамках итерации, заказчик дает обратную связь о том, что сделано хорошо, что необходимо исправить или доработать. По окончании встречи с заказчиком формируется протокол встречи, подводятся итоги и производится актуализация плана релизов и журнала продукта.

После демонстрации проводится ретроспектива внутри команды. Ретроспектива – активность команды, целью которой является подведение итогов прошедшей итерации. Ретроспектива позволяет найти проблемные места в рабочем процессе, определить над чем стоит работать более внимательно. С течением времени,



ретроспективы позволяют понять устраняем ли мы проблемы, возникшие ранее, или внедренные идеи по исправлению проблем только усугубляют ситуацию.

При использовании описанного подхода к управлению командой разработки можно достичь значимых результатов. Заключение поможет подытожить ключевые преимущества и результаты работы использования такого подхода:

- всегда опираемся на общедоступный и актуальный план работ;
- понимание текущего состояния разработки проекта;
- понимание реального времени на разработку основных задач;
- своевременная обратная связь от заказчика и корректировка приоритетов.

### ***Список литературы***

1. Сазерленд Д. Scrum. Революционный метод управления проектами / Д. Сазерленд. – СПб.: МИФ, 2022. – 272 с.

2. Как оценивать задачи по методу Planning Poker [Электронный ресурс]. – Режим доступа: [https://skillbox.ru/media/management/planning\\_poker](https://skillbox.ru/media/management/planning_poker) (дата обращения: 16.04.2024).

3. Диаграммы сгорания задач: что это такое и как использовать [Электронный ресурс]. – Режим доступа: <https://kaiten.ru/blog> (дата обращения: 16.04.2024).