					121.1151.01.01		
Изм.	Лист	№ докум.	Підпис	Дата			
студент		Король Д. В.			Анализ предметной области	Лит.	Лист 1 / Листов 11
студент		Брызгалов М. В.					
студент		Фадеев П. В.					НУК
преподаватель		Фарионова Т. А.					

Лабораторная работа №1

Тема: Анализ предметной области

Цель: Приобретение навыков составления протокола встречи с заказчиком, одностраничного описания проекта, словаря предметной области. Приобретение навыка выбора методологии.

Разработчики: Король Д.В. , Фадеев П.В. , Брызгалов М.В.

Протокол встречи с заказчиком

Приложение “Финансовый менеджер”

Платформа: мобильные устройства с операционной системой Android.

1. Назначение приложения: учет расходов, доходов и долгов пользователя, планирование расходов.

2. Идея приложения: приложение предназначено для упрощения учета денежных операций пользователя и их оптимизации с целью экономии времени и средств.

3. Приложение рассчитано на широкий круг пользователей.

4. Функционал программы:

4.1 Хранение информацию о текущем состоянии счетов пользователя, включая наличные.

4.2 Ввод данных о транзакциях (доход и расход)

4.3 Просмотр истории транзакций по конкретному счету и по всем счетам.

4.4 Просмотр статистики в виде графиков, столбчатых и секторных диаграмм.

4.5 Просмотр транзакций в разных валютах.

4.6 Планирование расходов и доходов.

4.7 Учет долгов

4.8 Изменение тем оформления интерфейса.

4.9 Настройки приложения.

4.10 Управление счетами

5. Интерфейс приложения:

5.1 Главный экран

5.1.1 Состояние счёта

5.1.2 Кнопки для добавления транзакций

5.2 Меню

5.2.1 Кнопка выбора темы оформления интерфейса

5.2.2 Кнопка выхода из приложения

5.2.3 Кнопка для настроек

5.2.4 Переключения между окнами приложения

5.2.5 Имя пользователя

5.3 Меню добавления транзакций

5.3.1 Счёт

5.3.2 Выбор суммы и валюты транзакции

5.3.3 Выбор и просмотр категории расходов

5.4 Статистика

5.4.1 Диаграммы расходов

5.4.2 Статистика расходов по категориям

5.5 Строка заголовка

5.5.1 Название текущего пункта

5.5.2 Кнопка меню

5.6 Режим планирования

5.7 Настройки

6. Для корректной работы приложения необходимо мобильное устройство с платформой Android 5.0+.

7. ПО будет распространяться в магазине мобильных приложений Google Play Market.

8. Модель распространения ПО: условно-бесплатная.

9. Приложение должно быть полностью разработано в течение 6 месяцев.

Одностраничное описание проекта

ПО «Финансовый менеджер»

Цель: Разработка программного обеспечения для помощи учёта доходов и расходов пользователя, с целью максимизации первых и минимизация последних.

ПО предназначено для решения проблемы учета расходов, доходов и долгов пользователя, а также их анализа и оптимизации:

1. Учет доходов и расходов человека;

2. Хранение информации о текущем финансовом состоянии пользователя и вывод статистики его изменения;

Функции, доступные пользователю при работе с приложением:

1. Анализ транзакций пользователя

2. Просмотр истории транзакций по конкретному счету и по всем счетам.

3. Просмотр статистики в виде графиков, столбчатых и секторных диаграмм.

4. Планирование расходов и доходов.

5. Учет долгов

6. Изменение тем оформления

7. Настройка приложения

ПО должно быть полностью разработано в течении 6 месяцев.

ПО будет распространяться в магазине мобильных приложений Google Play Market.

Модель распространения ПО: условно-бесплатная.

Словарь предметной области

Транзакция	операция по перемещению денежных средств со счёта или на счёт пользователя. Результат транзакции - изменение состояния карточного либо наличного счёта.
Расход	Транзакция, в результате которого сумма денег на счёте уменьшается.
Доход	Транзакция, в результате которого сумма денег на счёте увеличивается.
Долг	Сумма денег, которую пользователь должен вернуть кредитору либо заемщик должен вернуть пользователю.
Процентная ставка	Коэффициент сложного либо простого процента. Сумма долга может перерасчитываться один раз в указанный период либо непрерывно (линейно для простого процента и экспоненциально для сложного процента).
Запланированная транзакция	Транзакция, размещенная в разделе планирования расходов и доходов. Не влияет на состояние основных счетов.
Журнал транзакций	Экран приложения, на котором отображается история состояние счета с указанием всех транзакций за выбранный промежуток времени (неделя, месяц, год).

Методология

Рассмотрим две методологий и выберем среди них одну.

Каскадная модель (англ. waterfall model, иногда переводят как модель «Водопад») — модель процесса разработки программного обеспечения, в которой процесс разработки выглядит как поток, последовательно проходящий фазы анализа требований, проектирования, реализации, тестирования, интеграции и поддержки.

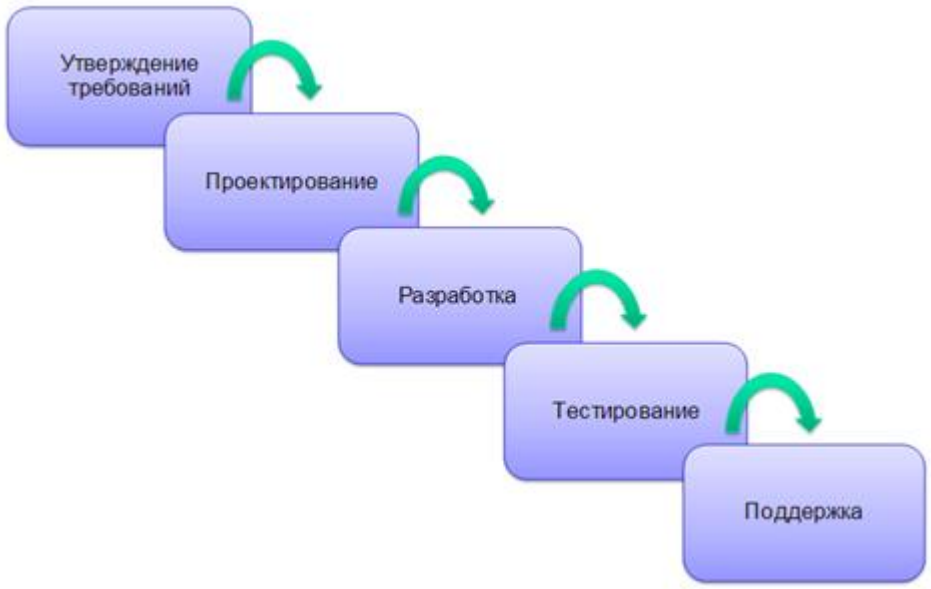

рис. 1. Каскадная модель

Её главные плюсы:

+ Полное документирование каждого этапа;

+ Четкое планирование сроков и затрат;

+ Прозрачность процессов для заказчика;

Каскадный подход хорошо зарекомендовал себя при построении ИС, для которых в самом начале разработки можно достаточно точно и полно сформулировать все требования, с тем чтобы предоставить разработчикам свободу реализовать их как можно лучше с технической точки зрения. В эту категорию попадают сложные расчетные системы, системы реального времени и другие подобные задачи. Однако, в процессе использования этого подхода обнаружился ряд его недостатков, вызванных прежде всего тем, что реальный процесс создания ПО никогда полностью не укладывался в такую жесткую схему. В процессе создания ПО постоянно возникала потребность в

возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений.

Гибкая методология разработки (англ. agile software development, agile-разработка) — обобщающий термин для целого ряда подходов и практик, основанных на ценностях Манифеста гибкой разработки программного обеспечения и 12 принципах, лежащих в его основе

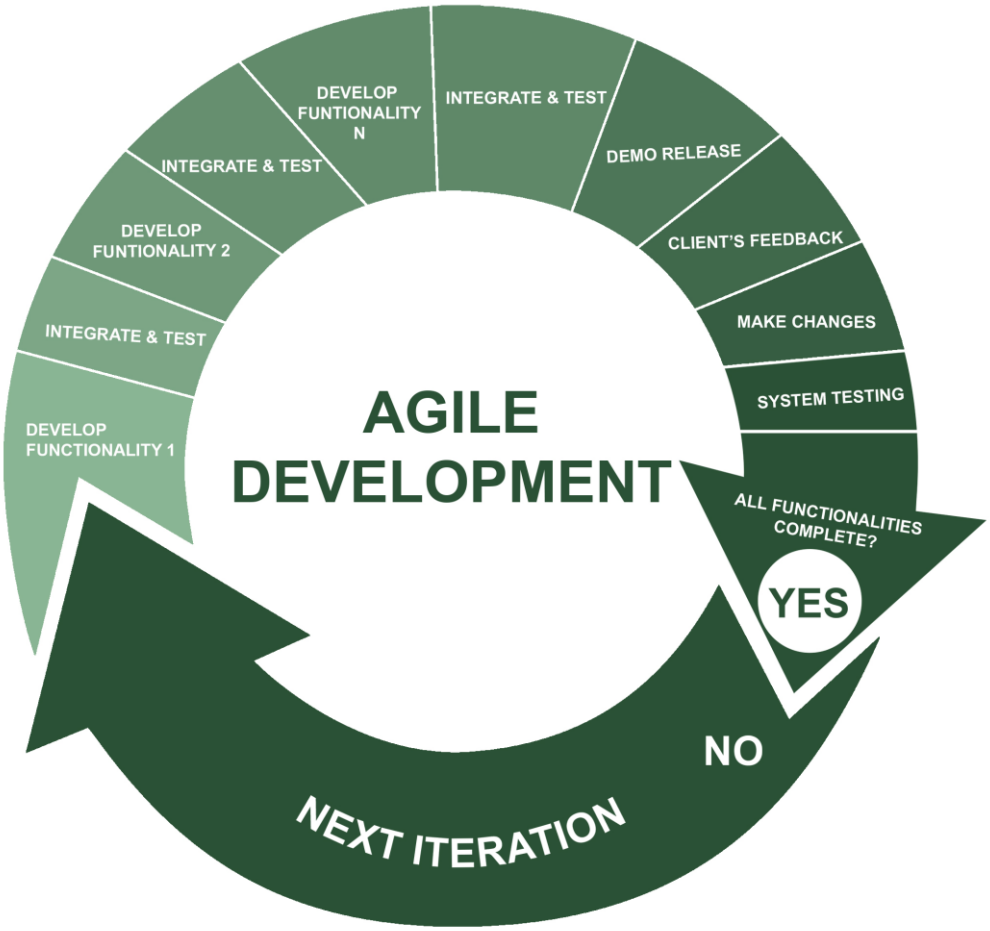

рис. 2. Модель гибкой методологии

12 принципов, которые составляют Agile Methodology, можно поделить на 4 главные идеи:

- Приоритет людей и общения над инструментами и процессами;
- Приоритет работающего продукта над полной документацией;
- Приоритет сотрудничества с заказчиков над утверждением контракта;

- Приоритет готовности меняться над следованием первоначально созданному плану.

Большинство гибких методологий нацелены на минимизацию рисков путём сведения разработки к серии коротких циклов, называемых итерациями, которые обычно длятся две-три недели.

Каждая итерация сама по себе выглядит как программный проект в миниатюре и включает все задачи, необходимые для выдачи мини-прироста по функциональности: планирование, анализ требований, проектирование, программирование, тестирование и документирование.

Хотя отдельная итерация, как правило, недостаточна для выпуска новой версии продукта, подразумевается, что гибкий программный проект готов к выпуску в конце каждой итерации.

По окончании каждой итерации команда выполняет переоценку приоритетов разработки.Agile-методы делают упор на непосредственном общении лицом к лицу. Большинство agile-команд расположены в одном офисе, иногда называемом англ. bullpen.

Как минимум, она включает и «заказчиков» (англ. product owner — заказчик или его полномочный представитель, определяющий требования к продукту; эту роль может выполнять менеджер проекта, бизнес-аналитик или клиент). Офис может также включать тестировщиков, дизайнеров интерфейса, технических писателей и менеджеров.Основной метрикой agile-методов является рабочий продукт.

Отдавая предпочтение непосредственному общению, agile-методы уменьшают объём письменной документации по сравнению с другими методами. Это привело к критике этих методов как недисциплинированных.

SCRUM (англ. «схватка» — термин из регби, обозначает стартовое состояние команд перед вбросом мяча) — минимально необходимый набор мероприятий, артефактов, ролей, на которых строится процесс SCRUM-разработки, позволяющий за фиксированные небольшие промежутки времени, называемые спринтами (sprints), предоставлять конечному пользователю работающий продукт с новыми бизнес-возможностями, для которых определён наибольший приоритет. Методология базируется на идеях, озвученных в статье Таекучи и Нонака «The New New Product development Game», и базируется на командной работе, по аналогии с тем, как в регби команда действует сообща, ради достижения общей цели. Возможности к реализации в очередном спринте определяются командой в начале спринта на совещании по планированию спринта Sprint Planning Meeting.

Для оценки предстоящего объёма работ на спринте чаще всего используются относительные оценки, и практика покера планирования(Planning Poker).В конце спринта Scrum-команда встречается на обзорном совещании результатов спринта (Sprint Review — старой название Demonstration) с заказчиком, и представляет ему инкремент бизнес-продукта (версия продукта с законченным набором функциональности, который уже можно отдавать заказчику и пользователю для использования), который она успела сделать за спринт.

Цель Sprint Review — получение обратной связи от заказчика, чтобы понять, на чем нужно делать акцент в дальнейшем, и какой должен быть следующий инкремент бизнес-продукта.Строго фиксированная небольшая длительность спринта (от 1 до 4 недель) снижает риски, и дает возможность быстро получить обратную связь от заказчика, чтобы скорректировать видение продукта.

В результате обсуждения рабочей группой, в качестве методологии для разработки проекта выбрана методология SCRUM. Возможность гибко управлять процессом разработки в условиях небольшого состава команды разработчиков и изменяющихся требований заказчика послужила главным доводом в пользу данной методологии. Кроме того, важным для нашего проекта преимуществом данной методологии является снижение рисков при разработке: процесс разработки приложения является во многом экспериментальным, и SCRUM позволит при необходимости переосмыслить ранее принятые решения.

Стратегия конструирования	Минимально необходимый набор мероприятий, артефактов, ролей, на которых строится процесс SCRUM-разработки, позволяющий за фиксированные небольшие промежутки времени, называемые спринтами (sprints), предоставлять конечному пользователю работающий продукт с новыми бизнес-возможностями, для которых определён наибольший приоритет.
Адаптивность процесса	Нефиксированные требования, отсутствие монолитности и инкрементный в выполнении задачи позволяют нам иметь возможность изменить продукт в соответствии с новыми условиями или вовсе улучшить его.
Этапы и связи между ними	Шаг 1. Создание бэклога продукта Шаг 2. Планирование спринта и создание Бэклога спринта Шаг 3. Работа над спринтом. Шаг 4. Тестирование и демонстрация продукта Шаг 5. Ретроспектива Вернутся на шаг 1
Формулировка требований	После выполнения одного из циклов разработки, обсуждают получившийся результат и делают выводы: нужно ли ещё что-то изменить, добавить или вовсе удалить. Таким образом новые требования могут возникать по мере разработки приложения и команда может оперативно реагировать на их появление.

Вывод: Был проведен сбор рабочей группы и предварительный анализ предметной области методом "мозгового штурма", выполнены работы по обоснованию разработки программного обеспечения, проведено интервью с заказчиком, формализованы функциональные и нефункциональные требования к программному обеспечению.