

Применение Agile Scrum для реализации автоматизированного рабочего места врача терапевта в городской больнице (часть 2)

Болохнов Александр Андреевич

Аннотация: в статье приводится декомпозиция ключевого бизнес-процесса врача-терапевта с помощью метода DFD до 3-го уровня детализации. Описание процессов создано в моделях As-Is и To-Be. Для большей наглядности и удобства внедрения улучшений была подготовлена карта процессов. Программная разработка реализована в среде MS Access 2010.

2.2. Моделирование процессов в модели TO-BE на основе VACD/DFD

Модель To-Be позволяет рассмотреть ключевые бизнес-процессы медицинского учреждения после внедрения разрабатываемого программного решения. Основное нововведение - использование базы данных, позволяющей заменить бумажные носители и уменьшить время поиска необходимой информации о пациенте. Описание будет проводиться как и в модели As-Is с применением нотации VACD (для верхнего уровня) и DFD (для нижних уровней декомпозиции). Стоит отметить, что первый уровень To-Be описания в нотации VACD, ничем не отличается от модели As-Is, данной на рис. 2.1.

На втором уровне описания объект медицинской карты пациента, содержащий информацию об анализах пациента, заменен базой данных. На рисунке 2.6 приведены результаты моделирования процесса лечения пациента с учетом этой замены.



Рис. 2.6. Процесс «Лечить пациента» на 2-м уровне модели To-Be в нотации DFD

Ключевые изменения в модели To-Be по сравнению с As-Is наглядно отображены на третьем уровне описания. На рисунке 2.7 видно, что бумажные носители в процессе назначения анализов заменены базой данных. Врачу не нужно больше искать информацию в медицинской карте и тратить на это значительное время. Похожие изменения произошли в двух других ключевых бизнес-процессах, так на рисунках 2.8-2.9 представлена декомпозиция процессов назначения обследования и лекарств.



Рис. 2.7. Процесс «Назначить анализы» на 3-м уровне модели To-Be в нотации DFD



Рис. 2.8. Процесс «Назначить обследование» на 3-м уровне модели To-Be в нотации DFD



Рис. 2.9. Процесс «Назначить лекарства» на 3-м уровне модели To-Be в нотации DFD

Обобщим результаты моделирования для модели To-Be графической картой бизнес-процессов. Рис. 2.10 содержит карту процессов, демонстрирующую зависимость уровней описания и операций, что структурирует и делает более наглядной деятельность врача-терапевта.

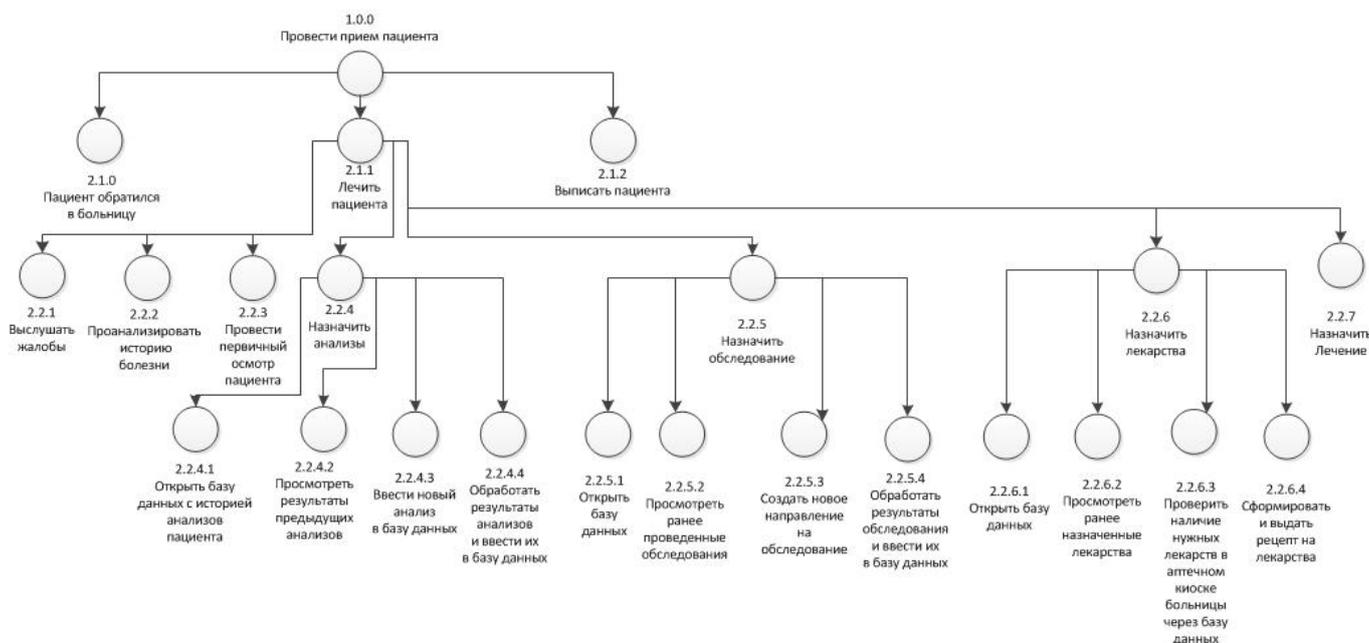


Рис. 2.10. Карта процессов для модели To-Be

2.3. Формирование структуры таблиц баз данных

Создадим необходимую для дальнейшей работы структуру базы данных, взаимосвязанную с ключевыми бизнес-процессами для автоматизации. Как правило, все данные приводят к третьей нормальной форме (далее - НФ). Согласно третьей НФ все неключевые поля, содержимое которых можно отнести к нескольким записям в таблице баз данных, должны быть вынесены в отдельную таблицу. Поэтому будем использовать базу данных, содержащую шесть таблиц: пациенты, анализы, накладные на выдачу, направления на обследования, список обследований, лекарства. На рисунке 2.11 представлена архитектура данных и связи между ними.



Рис. 2.11. Архитектура данных

Таблица «Пациенты» имеет ключевое поле «Код пациента» и связана с таблицей «Анализы» через поле «Код анализа», отношения между таблицами относятся к категории 1:М. По аналогичной схеме устроена взаимосвязь между таблицей «Пациенты» и «Накладные на выдачу» через ключевое поле «Код заявки», «Накладные на выдачу» и «Лекарства» через «Код препарата», «Пациенты» и «Направления на обследование» через ключевое поле «Код направления», «Направления на обследование» и «Список обследований» через «Код обследования». Информацию из таблиц можно разбить на классы. В таблице 2.1 дана информация о классах и типах данных, используемых в базе данных.

Таблица 2.1. Классы и типы данных

Название класса	Поле данных	Тип данных	Размер поля
Пациенты	Код пациента (Ключевое поле)	Счётчик	Длинное целое
	Фамилия	Текстовый	255
	Имя	Текстовый	255
	Отчество	Текстовый	255
	Дата рождения	Дата/время	255
	Пол	Текстовый	255
	Номер мобильного	Текстовый	255
	Номер ОМС	Текстовый	255
	Постоянное место жительства	Текстовый	255
Анализы	Код анализа (Ключевое поле)	Счётчик	Длинное целое
	Код пациента	Числовой	Длинное целое
	Дата анализа	Текстовый	255
	Название анализа	Текстовый	255
	Дата поступления результатов	Дата/время	255
	Результат анализа	Текстовый	255
Накладные на выдачу	Код заявки (Ключевое поле)	Счётчик	Длинное целое
	Код препарата	Числовой	Длинное целое
	Код пациента	Числовой	Длинное целое
	Нужное количество	Числовой	Длинное целое
	Дата оформления	Дата/время	255

Название класса	Поле данных	Тип данных	Размер поля
Лекарства	Код препарата (Ключевое поле)	Числовой	Длинное целое
	Наименование препарата	Текстовый	255
	Количество на складе	Числовой	255
Направления на обследования	Код направления (Ключевое поле)	Счётчик	Длинное целое
	Код пациента	Числовой	Длинное целое
	Код обследования	Числовой	Длинное целое
	Дата оформления	Дата/время	255
Список обследований	Код обследования (Ключевое поле)	Числовой	Длинное целое
	Название обследования	Текстовой	255
	Номер кабинета	Числовой	Длинное целое

2.4. Моделирование программных интерфейсов

Для большей наглядности на рисунке 2.12 представлена схема приложения. Рисунок отображает расположение основных элементов управления в планируемой программе. Для защиты данных от несанкционированного доступа в программе запрашивается пароль. Приложение содержит в себе двенадцать основных экранных форм.

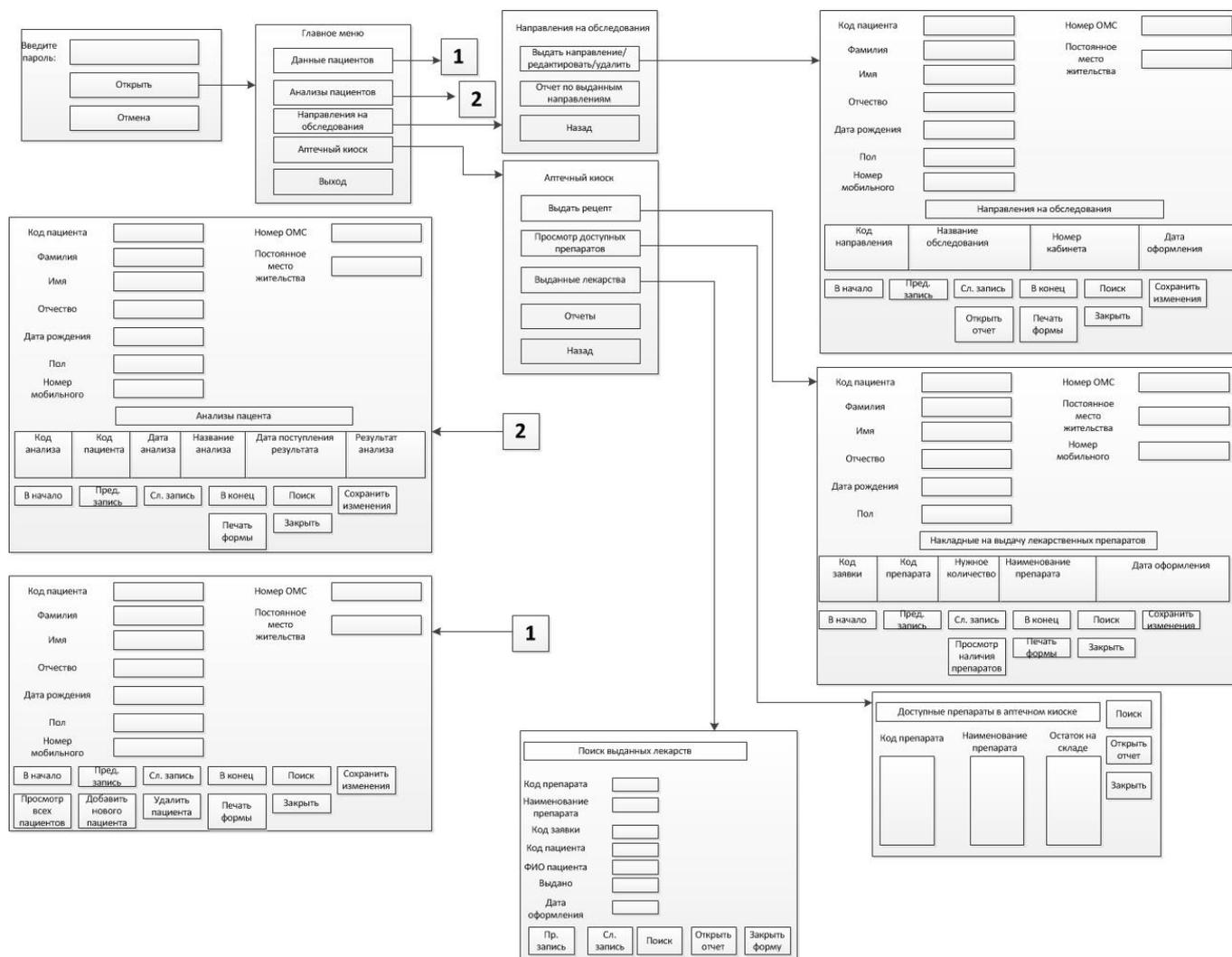


Рис. 2.12. Схема разрабатываемого приложения

Планируется создание главного меню для доступа к рабочим формам врача-терапевта. На рисунке 2.13 изображен ожидаемый вид главного меню. Разрабатываемая программа должна быть информативной и понятной пользователю, при этом она должна соответствовать всем функциональным требованиям, идентифицированным ранее в бэклоге продукта. На рисунке 2.14 дана рабочая форма программы для просмотра сведений о пациентах и их анализах. Схема формы для выдачи, редактирования и удаления направлений на обследования продемонстрирована на рис. 2.15. Кроме того на рисунке 2.16 изображен планируемый вид формы для обработки рецептов на лекарственные препараты.

Главное меню

Рис. 2.13. Планируемый вид формы главного меню

Код пациента	<input type="text"/>	Номер ОМС	<input type="text"/>		
Фамилия	<input type="text"/>	Постоянное место жительства	<input type="text"/>		
Имя	<input type="text"/>				
Отчество	<input type="text"/>				
Дата рождения	<input type="text"/>				
Пол	<input type="text"/>				
Номер мобильного	<input type="text"/>				
<input type="button" value="Анализы пацента"/>					
Код анализа	Код пациента	Дата анализа	Название анализа	Дата поступления результата	Результат анализа
<input type="button" value="В начало"/>	<input type="button" value="Пред. запись"/>	<input type="button" value="Сл. запись"/>	<input type="button" value="В конец"/>	<input type="button" value="Поиск"/>	<input type="button" value="Сохранить изменения"/>
	<input type="button" value="Добавить нового пациента"/>	<input type="button" value="Удалить пациента"/>	<input type="button" value="Печать формы"/>	<input type="button" value="Закреть"/>	

Рис. 2.14. Планируемый вид формы просмотра данных пациента и его анализов

Код пациента	<input type="text"/>	Номер ОМС	<input type="text"/>		
Фамилия	<input type="text"/>	Постоянное место жительства	<input type="text"/>		
Имя	<input type="text"/>				
Отчество	<input type="text"/>				
Дата рождения	<input type="text"/>				
Пол	<input type="text"/>				
Номер мобильного	<input type="text"/>				
Направления на обследования					
Код направления	Название обследования	Номер кабинета	Дата оформления		
<input type="button" value="В начало"/>	<input type="button" value="Пред. запись"/>	<input type="button" value="Сл. запись"/>	<input type="button" value="В конец"/>	<input type="button" value="Поиск"/>	<input type="button" value="Сохранить изменения"/>
<input type="button" value="Открыть отчет"/>		<input type="button" value="Печать формы"/>	<input type="button" value="Закреть"/>		

Рис. 2.15. Планируемый вид формы для выдачи, редактирования и удаления направлений на обследования

Код пациента	<input type="text"/>	Номер ОМС	<input type="text"/>		
Фамилия	<input type="text"/>	Постоянное место жительства	<input type="text"/>		
Имя	<input type="text"/>				
Отчество	<input type="text"/>	Номер мобильного	<input type="text"/>		
Дата рождения	<input type="text"/>				
Пол	<input type="text"/>				
Накладные на выдачу лекарственных препаратов					
Код заявки	Код препарата	Нужное количество	Наименование препарата	Дата оформления	
<input type="button" value="В начало"/>	<input type="button" value="Пред. запись"/>	<input type="button" value="Сл. запись"/>	<input type="button" value="В конец"/>	<input type="button" value="Поиск"/>	<input type="button" value="Сохранить изменения"/>
<input type="button" value="Просмотр наличия препаратов"/>		<input type="button" value="Печать формы"/>	<input type="button" value="Закреть"/>		

Рис. 2.16. Планируемый вид формы для выдачи, редактирования и удаления рецептов на лекарственные препараты

Таким образом, в рамках первого спринта были спроектированы ключевые бизнес-процессы, структура данных для них, а также ожидаемый вид программы. Что послужило основанием для разработки последующего приложения в среде MS Access 2010 [10], части которого распределены по трем последующим итерациям разработки.

3. Разработка и тестирование приложения на 2-4 спринтах

3.1. Реализация программы на 2-м спринте

В рамках второго спринта создается таблица с персональными данными пациентов: фамилия, имя, отчество и др. Фрагмент таблицы представлен на рисунке 3.1.

к	фамилия	имя	отчество	дата рож	пол	номер мобил	номер ОМС	Постоянное место жительства
1	Петров	Александр	Григорьевич	12.12.1965	М	12243243243244	1232432453543423	г. Москва, пр-т Вернадского, 78
2	Сидоров	Олег	Петрович	13.08.1992	М	12312312312321	123123123123213123	г. Москва, ул. Стромынка, 20
4	Пронькин	Андрей	Юрьевич	12.11.1992	М	12321432432423	12312445546	г. Москва, Сокольнические пер., 34
5	Иванова	Екатерина	Фёдоровна	03.04.1978	Ж	77856345453457	655555547789435654	г. Москва, Яузская ул., 7
6	Полюкова	Ольга	Владимировн	07.09.1988	Ж	87968796789	789789789879	г. Москва, Сокольнический пер., 11

Рис. 3.1. Фрагмент таблицы «Пациенты»

На основе созданной таблицы создается экранная форма, обеспечивающая просмотр, добавление и удаление информации о пациенте, а также поиск по ключевым параметрам. Фрагмент формы приведен на рисунке 3.2.

Кроме того, в контексте этого спринта создаются таблицы баз данных «Анализы», «Направления на обследования», «Накладные на выдачу», «Лекарства» и «Список обследований». Используя их, реализуются пользовательские экраны для работы врача-терапевта. Рисунок 3.3 показывает фрагмент формы «Анализы пациентов». Форма ввода «Направления на обследования» приведена на рис. 3.4.

Данные пациентов

код пациента:

фамилия:

имя:

отчество:

дата рождения:

пол:

номер мобильного:

номер ОМС:

Постоянное место жительства:

В начало | Пред. запись | Сл. запись | В конец | Поиск по пациентам | Сохранить изменения | Добавить нового пациента | Удалить пациента

Просмотр всех пациентов | Печать формы | Закрыть

Рис. 3.2. Фрагмент формы «Данные пациентов»

Анализы пациентов

Код пациента: Номер ОМС:

Фамилия: Постоянное место жительства:

Имя:

Отчество:

Дата рождения:

Пол:

Номер мобильного:

Список анализов пациента

Код анализа	код пациента	дата анализа	название анализа
1	1	04.04.2019	анализ крови общий
3	1	01.04.2019	биохимия
4	1	08.04.2019	Общеклинический анализ мочи
5	1	02.04.2019	Биохимический анализ крови
6	1	07.04.2019	Кровь на гормоны
*	(№)		

Запись: 1 из 5 Нет фильтра Поиск

В начало | Пред. запись | Сл. запись | В конец | Поиск по пациентам | Сохранить изменения | Печать формы | Закрыть

Рис. 3.3. Фрагмент рабочей формы «Анализы пациентов»

Направления на обследования

Код пациента Номер ОМС

Фамилия Постоянное место жительства

Имя Номер мобильного

Отчество

Дата рождения

Пол

Направления на обследования

Код направл	Название обследования	Номер кабинета	Дата оформления
13	УЗИ	217	07.05.2019
14	МРТ	218	06.05.2019
*	(№)		

Запись: 14 3 из 3 Нет фильтра Поиск

Рис. 3.4. Фрагмент рабочей формы «Направления на обследования»

3.2. Разработка программы на 3-м спринте

Третий спринт предполагает следующие функции, необходимые для реализации: разработка возможности редактирования и удаления назначенных пациенту анализов, рецептов на выдачу лекарственных препаратов, а также обработки ранее назначенных обследований. Для создания рабочей программы требуется создать и связать релевантные таблицы данных. На рисунке 3.5 дана схема данных, для ведения которой настраиваются пользовательские формы обработки данных.

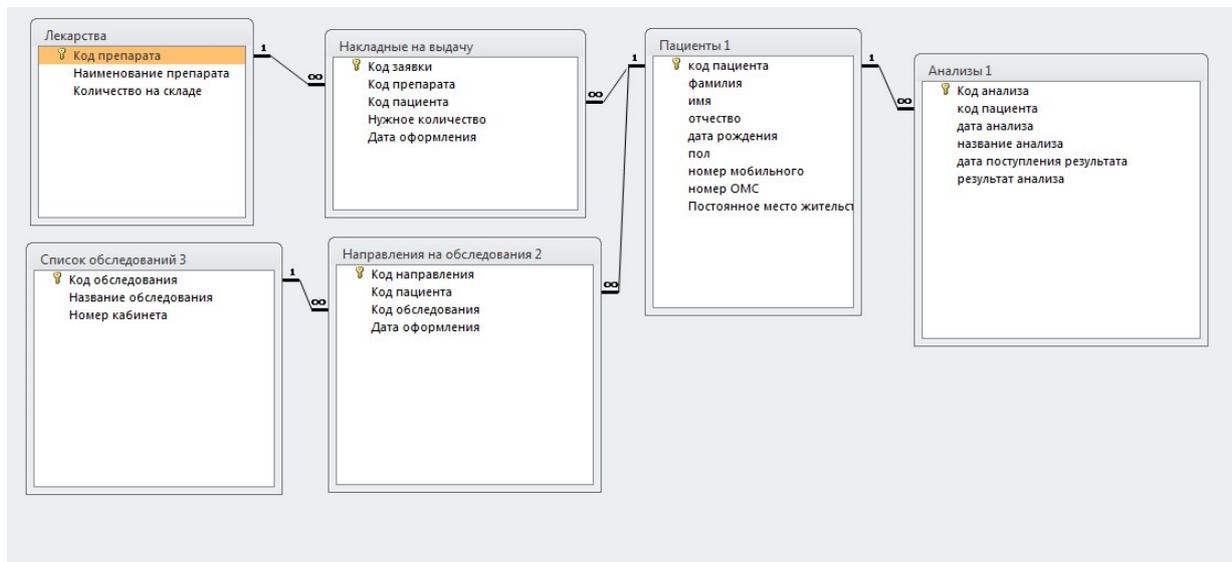


Рис. 3.5. Схема данных с применяемыми связями

3.3. Создание программы на 4-м спринте

Текущий спринт предполагаем разработку следующих возможностей: отображение даты и времени назначения анализа, даты и времени поступления результатов анализа, показ анализов пациента, проверку введенного пользователем пароля, создания отчетов в том числе отображение наличия на складе аптечного киоска больницы лекарственных препаратов. На рисунке 3.6 дан фрагмент формы «Анализы пациентов», с доработанными функциональными возможностями.

В завершении спринта разработана форма программы с возможностью защиты данных от несанкционированного доступа путем проверки пароля при входе в приложение. При вводе верного пароля пользователь попадает в главное меню, показанное на рисунке 3.7. При нажатии на кнопку «Данные пациентов» пользователь переходит в рабочую форму для добавления, редактирования и удаления данных пациентов. Используя кнопку «Анализы пациентов», врач-терапевт может обрабатывать направления на анализы. Кнопки «Направления на обследования» и «Аптечный киоск» открывают небольшие подменю для дальнейшего перехода в соответствующие подэкраны.

В пользовательских формах «Анализы пациентов», «Направления на обследования» и «Форма выдачи препаратов» были реализованы восемь основных активных кнопок. Кнопки «В начало», «Предыдущая запись», «Следующая запись» и «В конец» позволяют переключаться между пациентами, кнопка «Поиск» - искать

информацию. Для добавления новых и редактирования уже существующих записей реализована специальная табличная часть внизу формы ввода. Также существует возможность вывести информацию на печать, нажатием соответствующей кнопки. Для выхода из программы предназначена кнопка «Выход».

Анализы пациентов

Код пациента: Номер ОМС:

Фамилия: Постоянное место жительства:

Имя:

Отчество:

Дата рождения:

Пол:

Номер мобильного:

Список анализов пациента

Код анализа	код пациента	дата анализа	название анализа	дата поступления результата	результат анализа
1	1	04.04.2019	анализ крови общий		показатели в норме
3	1	01.04.2019	биохимия		
4	1	08.04.2019	Общеклинический анализ мочи		
5	1	02.04.2019	Биохимический анализ крови		
6	1	07.04.2019	Кровь на гормоны		
*	(№)				

Запись: 1 из 5 Нет фильтра Поиск

В начало Пред. запись Сл. запись В конец Поиск по пациентам Сохранить изменения Печать формы Закрыть

Рис. 3.6. Фрагмент формы ввода «Анализы пациентов»

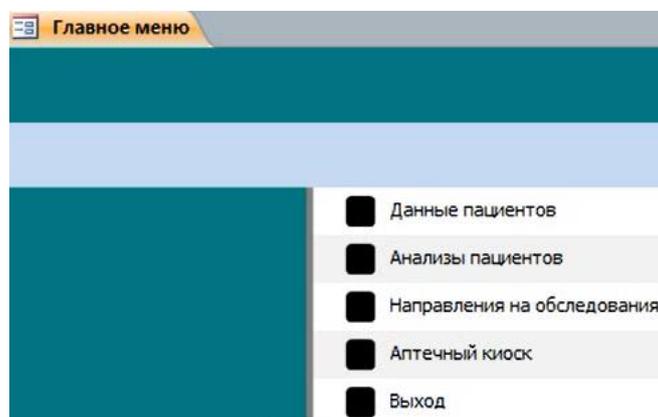


Рис. 3.7. Главное меню программы

3.4. Тестирование приложения

Для выявления неточностей и ошибок в работе программного обеспечения используется процесс анализа продукта - его тестирование. Благодаря тестированию команда разработчиков может устранить существующие недостатки и улучшить продукт в дальнейшем. Выделяется три основных вида тестирования: функциональное, интеграционное и нагрузочное.

Функциональное тестирование применяется для проверки того, что программа обеспечивает именно тот результат, который требуется конечному пользователю. Как правило, функциональное тестирование включает в себя оценку и сравнение каждой функции программного обеспечения с бизнес-требованиями. Кроме того, в ходе функционального тестирования проверяется программное обеспечение на удобство использования.

Интеграционное тестирование - это испытание программного обеспечения, в котором отдельные единицы объединяются, и тестируется как группа. Целью данного тестирования является выявление ошибок во взаимодействии между интегрированными модулями. Нагрузочное тестирование проводится для определения быстродействия системы. Для проведения функционального тестирования составляется список функциональных требований и в нем расписываются реализованные программные компоненты:

- отображение в таблице данных о пациентах - таблица с персональными данными пациентов;
- использование компонентов программы, обеспечивающие создание, редактирование и удаление данных о пациентах - форма с активными элементами и полями для добавления, редактирования и удаления данных о пациенте;
- использование компонентов программы, обеспечивающие поиск данных по ключевым параметрам - активные кнопки на форме, позволяющие выполнить поиск по заданным параметрам;
- использование компонентов программы, обеспечивающих возможность назначать анализы пациентам - форма с активными элементами и полями для добавления новых данных по анализам пациентов;
- использование компонентов программы, обеспечивающих возможность назначать обследование пациентам - форма с активными элементами и полями для добавления новых направлений на обследование;

- использование компонентов программы, обеспечивающих возможность создания рецептов на выдачу лекарственных препаратов пациентам – форма для добавления новых рецептов на выдачу лекарственных препаратов;
- отображение выданных пациенту рецептов на лекарственные препараты – форма с элементами, позволяющая отобразить выданные рецепты по каждому пациенту;
- использование компонентов программы, обеспечивающих возможность редактировать и удалять: назначенный анализ, рецепт на выдачу лекарственных препаратов, назначенное обследование – форма с полями для редактирования и удаления данных по каждому пациенту;
- использование компонентов программы, обеспечивающих поиск выданных пациенту рецептов на лекарственные препараты и назначенных обследований по заданным параметрам – активные кнопки на форме, позволяющие выполнять поиск по заданным параметрам;
- отображение даты и времени назначенного анализа в отдельной графе интерфейса программы – форма, отображающая дату и время назначенного анализа в отдельной графе интерфейса;
- отображение на экране данных о дате и времени поступления результатов анализа – отдельная графа в рабочей форме, отображающая дату и время поступления результатов анализа;
- отображение информации, содержащей результаты анализов пациента – отдельная графа в рабочей форме, отображающая результаты анализа пациента;
- использование программного кода, обеспечивающего защиту от несанкционированного доступа, реализация формы авторизации пользователя в системе – отдельная форма авторизации пользователя в системе при открытии базы данных;
- использование программного кода, позволяющего вводить и проверять введенный пользователем пароль – поле для ввода пароля на отдельной форме авторизации;
- использование компонентов программы, обеспечивающих возможность создания отчетов – активные кнопки на формах, позволяющие создать отчет;
- использование компонентов программы, позволяющих отображать наличие лекарственных препаратов на складе аптечного киоска больницы – форма с активными элементами, отображающая остаток на складе аптечного киоска больницы лекарственных препаратов.

На основании этого списка установлено, что все заявленные функциональные требования были выполнены, программа полноценно работает, таким образом

функциональное тестирование было успешно пройдено. Нагрузочное тестирование служит для анализа производительности разработанного решения. Рассмотрим время отклика программы при работе с разным количеством записей в базе данных (1, 10, 50, 100). На рисунке 4.1 представлены результаты измерений. С помощью электронного секундомера пять раз была проведена оценка времени отклика системы на такие действия, как: запись и поиск данных. Затем высчитывалось среднее время отклика и среднее квадратичное отклонение. Далее по формуле (4.1)

$$\Delta t = \sqrt{\left(\frac{\sigma}{\sqrt{n}} t_{a(n-1)}\right)^2 + A^2} \quad (4.1)$$

вычислялась погрешность измерений. Тогда время отклика задавалось формулой (4.2):

$$t_{\text{отк}} = t_{\text{ср.ар.}} \pm \Delta t \quad (4.2)$$

Количество записей	Действие	t1,с	t2,с	t3,с	t4,с	t5,с	Среднее время отклика, с	Средн. Квадр. Отклон.,с	Погрешность измерений, с	Время отклика, с
1	Запись	0,09	0,10	0,1	0,10	0,11	0,1040	0,0102	0,0066	0,104±0,006
	Поиск	0,11	0,1	0,1	0,09	0,10	0,1020	0,0117	0,0070	0,102±0,007
10	Запись	0,09	0,10	0,1	0,09	0,1	0,0980	0,0117	0,0070	0,098±0,007
	Поиск	0,10	0,10	0,1	0,12	0,1	0,1140	0,0120	0,0071	0,114±0,007
50	Запись	0,19	0,2	0,2	0,20	0,20	0,2000	0,0110	0,0068	0,200±0,006
	Поиск	0,20	0,20	0,2	0,19	0,20	0,1940	0,0080	0,0060	0,194±0,006
100	Запись	0,27	0,3	0,3	0,28	0,3	0,2860	0,0136	0,0076	0,286±0,007
	Поиск	0,29	0,30	0,3	0,31	0,30	0,3020	0,0075	0,0059	0,302±0,007

Рис. 4.1. Расчеты времени отклика для нагрузочного тестирования

Как видно из результатов оценивания выше, задержка незначительно возрастает при увеличении количества записей. Однако это не оказывает существенного влияния на стабильное функционирование программы и практически незаметно для пользователя.

Заключение

В данной работе продемонстрированы возможности гибкой методологии разработки Agile, в частности метод Scrum. Сформулированы пользовательские и

функциональные требования, составлен бэклог продукта и спринтов. Для моделирования ключевых бизнес-процессов верхнего уровня применена нотация ARIS VACD. Ключевые бизнес-процессы декомпозированы с помощью метода DFD до третьего уровня детализации. Описание процессов создано в моделях As-Is и To-Be. Для большей наглядности и удобства внедрения улучшений была подготовлена карта процессов. Для графического отображения процессов и создания карты использовано решение MS Visio.

На основе подготовленных и проанализированных данных была реализована программа в среде Microsoft Access. Согласно запланированным спринтам программа была доработана, улучшена и дополнена новыми возможностями. Затем разработанное приложение было испытано путем проведения функционального и нагрузочного тестирования. Результаты испытаний показали стабильность и безошибочность работы программы.

Определенно, созданное приложение может быть доработано с точки зрения графического интерфейса пользователя, что обеспечит удобство работы врача-терапевта, также функционал системы может быть расширен, исходя из меняющихся пользовательских требований.

Литература

1. Product development. What is Agile methodology [Электронный ресурс]. – Режим доступа: URL: <https://luis-goncalves.com/what-is-agile-methodology/> (дата обращения: 29.10.2019).
2. Учёба.ру. Определение врача-терапевта [Электронный ресурс]. – Режим доступа: URL: <https://www.ucheba.ru/prof/137> (дата обращения: 29.10.2019).
3. Habr. Техники определения приоритетов [Электронный ресурс]. – Режим доступа: URL: <https://habr.com/company/hygger/blog/351238/> (дата обращения: 29.10.2019).
4. Маргарет Роуз. Бизнес процессы [Электронный ресурс]. – Режим доступа: URL: <https://searchcio.techtarget.com/definition/business-process> (дата обращения: 11.11.2019).
5. Uchebnik.Online. Методология ARIS [Электронный ресурс]. – Режим доступа: URL: <http://uchebnik.online/biznes-protsessov-modelirovanie/metodologiya-aris-54083.html> (дата обращения: 11.11.2019).

6. Charles Sturt University. ARIS Standards and Conventions Manual [Электронный ресурс]. - Режим доступа: URL: https://cdn.csu.edu.au/__data/assets/pdf_file/0020/1314173/CSU_ARIS_Modelling_Standards_and_Conventions_Manual_V1_0.pdf (дата обращения: 25.11.2019).
7. Степанов Д. Ю. Анализ, проектирование и разработка корпоративных информационных систем: уровень бизнес-процессов / МИРЭА. - М., 2017 [Электронный ресурс]. - Режим доступа: URL: https://stepanovd.com/documents/training/1_erp/7-processlevel/processlevel.pdf (дата обращения: 11.11.2019).
8. What is Agile. What is Scrum? [Электронный ресурс]. - Режим доступа: URL: <https://www.cprime.com/resources/what-is-agile-what-is-scrum> (дата обращения: 29.11.2019).
9. Lucidchart. What is a Data Flow Diagram [Электронный ресурс]. - Режим доступа: URL: <https://www.lucidchart.com/pages/data-flow-diagram> (дата обращения: 17.11.2019).
10. Е.М. Карчевский, И.Е. Филиппов, И.А. Филиппова. Access 2010 в примерах. Учебное пособие. Казанский университет. Казань. 2012. [Электронный ресурс]. - Режим доступа: URL: https://kpfu.ru/docs/F1448756111/Access_2010.pdf (дата обращения: 21.04.2019).

Выходные данные статьи

Болохнов А.А. Применение Agile Scrum для реализации автоматизированного рабочего места врача терапевта в городской больнице (часть 2) // Корпоративные информационные системы. - 2020. - №3 (11) – С. 49-67. – URL: <https://corpinfosys.ru/archive/issue-11/108-2020-11-agilescrum>.

Об авторе



Болохнов Александр Андреевич – студент 4-го курса кафедры оптических и биотехнических систем и технологий физико-технологического института РТУ МИРЭА. Тема выпускной квалификационной работы бакалавра «Применение методологии Agile Scrum для реализации автоматизированного рабочего места врача терапевта в городской больнице». Электронная почта: bolokhnov@sumirea.ru.