

Автоматизация работы стоматологической клиники с использованием спиралевидной модели внедрения информационных систем (часть 1)

Худяков Сергей Дмитриевич

Аннотация: в статье описывается реализация приложения для стоматологии, обеспечивающее электронное хранение информации пациентов и данных, связанных с их лечением. Проводится автоматизация процесса поиска информации о состоянии пациента и совершённых процедурах над конкретными зубами пациента, что позволяет упростить работу врача-стоматолога, сократив время и количество действий.

1. Введение

С изобретением компьютера человек старается заменить бытовые функции и действия автоматизированными системами, во многих ситуациях это является более выигрышным вариантом в виду сокращения времени, затрачиваемого на обработку данных и циклические процессы. Применение компьютерных технологий позволяет снизить фактор «человеческой ошибки» по причине усталости и невнимательности человека, недостаточной наглядности данных и прочее.

Для любого автоматизированного процесса необходимо программное обеспечение (далее – ПО), которое обеспечивает работу необходимых функций и обслуживание процесса, имеет возможность его дальнейшего усовершенствования, доработки и внесения изменений. В общем комплексе автоматизации жизненных сфер и процессов отдельным классом можно выделить информатизацию здравоохранения. Автоматизация заключается в переводе бумажных документов в цифровой формат и создании баз данных с возможностью ведения и отслеживания хода лечения пациента.

В работе описан процесс автоматизации деятельности стоматологической клиники. Актуальность данного вопроса объясняется ежегодно увеличивающимся количеством стоматологических клиник и не снижающимся количеством обращений в них, что приводит к росту очередей и высокой нагрузке менеджеров и медицинского персонала стоматологии. Современные рабочие места врачей-стоматологов оборудованы персональным компьютером, что даёт теоретическую возможность внедрить программный продукт, тем самым упростить работу врача и сделать

информацию наглядной и быстро доступной по сравнению с использованием бумажных носителей.

В настоящее время на рынке программного обеспечения имеется ряд продуктов [1], решающих вопрос автоматизации деятельности стоматологии. Однако их отличает сложность внедрения, требовательность к вычислительной мощности персональных компьютеров (ПК), зачастую высокая стоимость, а также отсутствие возможности доработки с учетом особенностей работы конкретной клиники.

2. Спиралевидная модель внедрения информационных систем

Процесс создания ПО является достаточно трудоемким и сложным. Продукт претерпевает ряд событий, которые в совокупности называются жизненным циклом (далее – ЖЦ) [2]. На данный момент существует множество моделей внедрения, позволяющих наладить и структурировать процесс создания ПО от начала задумки до этапа промышленной эксплуатации. Все эти модели схожи по наличию следующих действий [3]:

- анализ требований;
- рабочее проектирование или создание технического проекта;
- программирование;
- пробная эксплуатация;
- сопровождение и улучшение.

При выборе модели внедрения необходимо учитывать масштабность проекта и возможности заказчика. Так же важным фактором является принятие решения о необходимости учёта рисков в ходе реализации проекта. Данная статья подразумевает использование спиралевидной модели внедрения программного продукта, что аргументируется следующим:

- необходимость внесения изменений в продукт в ходе его реализации;
- ПО создаётся для частной клиники, поэтому необходимо учесть возникновение возможных рисков и оценить их значимость для проекта;
- требуется демонстрация промежуточных результатов подготовки продукта.

Идея спиралевидной модели заключается в разбиение задач реализации продукта на четыре последовательных этапа, которые будут выполняться итеративно. Количество итерации или витков получения программного продукта определяется в ходе жизненного цикла проекта внедрения в зависимости от требований заказчика. Этапы представимы следующими задачами (рисунок 1):

- формирование цели и требований;
- качественная оценка рисков;
- конструирование (разработка, кодирование и тестирование) продукта;
- оценка результата заказчиком.



Рис. 1. Схема спиралевидная модели внедрения программного продукта

Отличительной особенностью данной модели является возможность принятия решения о необходимости завершения разработки программы не только по результатам выполнения финального витка спирали, но также на основе оценивания рисков, что способствует существенному сохранению времени и ресурсов. Каждый виток спирали в частности может быть рассмотрен как V-образная модель разработки через тестирование, причём по завершении итерации разработки имеется продукт с готовым функционалом, который совершенствуется на следующем цикле. Рассмотрим подробнее каждый из этапов разработки.

На заданном витке разработки уточняются требования к продукту. Например, что функционально должна выполнять программа и что реализуем в первую очередь из этого; какие бизнес-цели преследуются заказчиком (определяем ключевой бизнес-процесс на данном витке); какие имеются ограничения по техническим и бизнес-ресурсам; что предлагают конкуренты и какие существуют на данный момент

альтернативы. Так же определяются все аспекты условий выполнения проекта. На последующих стадиях заказчик может внести корректировки и новые предложения.

Далее идентифицируются риски, например, на основе метода Боэма. Проводится качественный и количественный анализ рисков. Качественный анализ рисков позволяет распознать возможные негативные события, а также их причины и факторы влияния, кроме того предложить стратегию реагирования; количественный анализ рисков обеспечивает расчет численно-экономических показателей проекта с учетом действия неблагоприятных факторов. Главной задачей разработчиков является выявление всех возможных рисков, после оценки которых, руководитель проекта принимает одно из следующих решений:

- создание прототипа и переход к следующему этапу;
- остановка дальнейшего усовершенствования продукта и передача заказчику имеющихся результатов;
- возврат к предыдущей стадии разработки проекта;
- прекращение разработки в силу неоправданности проекта.

Создание прототипа минимизирует затраты средств и времени, а также позволяет нагляднее понять требования к продукту на этапе разработки. В последующем ведется реализация и написание программного кода продукта, в том числе пользовательского интерфейса. На первой итерации создается концепция продукта (Proof Of Concept) [4], необходимая для первоначальной оценки заказчиком, после чего на дальнейших витках спирали создаются готовые версии программы, которые демонстрируют ход работы над проектом, уточняют требования и описывают необходимые исправления на пути к готовому решению. По завершении создания производится тестирование продукта и исправление сопутствующих ошибок.

Последний этап витка спирали включает в себе разворачивание и внедрение полученного промежуточного продукта или конечного приложения с возможностью его сопровождения и усовершенствования. В случае необходимости дальнейшего усовершенствования программы или исправления недоработок проводится анализ проделанной работы и переход на новый виток разработки.

Спиралевидная модель содержит итерации разработки, которые шаг за шагом улучшают программное решение. Количество спиралей уточняется после формирования списка требований, анализа их сложности и необходимости выполнения. Предварительно к использованию планируются 4-е спирали, которые будут содержать примерно следующие этапы разработки (таблица 2.1).

Таблица 2.1. Предварительное содержание витков спирали

Этап спирали	Номер спирали	Предполагаемые действия
Поиск решения задач, сформулированных в требованиях	I	Общая структура
	II	Взаимосвязи данных
	III	Интерфейс
	IV	Доработка дополнительных функций и дизайна приложения
Оценка возможных рисков и их влияния	I, II, III, IV	Рассмотрение рисков и вариантов взаимодействия с ними
Разработка и кодирование части проекта	I, II, III, IV	Реализация требований в программной среде Microsoft Access 2016
Приемочное тестирование приложения	IV	Проведение тестирования приложения

3. Идентификация требований и формирование бэклога

Пользовательские требования преимущественно описывают желаемый результат, но никак не учитывают технический аспект осуществления проекта [5]. Требования заказчика к продукту фиксируется исходя из результатов интервью. Для корректности и точности собранной информации необходимо:

- выбрать стейкхолдеров, т.е. определить пользователей системы, которые будут использовать ПО, чтобы учесть мнение большего числа заинтересованных сторон;
- собрать требования, непосредственное общение со стейкхолдерами позволяет провести анализ предметной области. Определяется, что именно пользователи хотят видеть в программе и какие функции она должна выполнять. При этом учитываются специфика внедрения продукта и его дальнейшее обслуживание;
- проанализировать требования, что подразумевает проверку требований на понятность, однозначность, полноту, а также поиск взаимосвязей между ними. При этом требования классифицируются по значимости, что в дальнейшем влияет на очередность их выполнения;
- документирование требований, т.е. фиксация всех требований в выбранном формате для дальнейшего использования, что обеспечивает возможность

отслеживания хода реализации, а также позволяет согласовывать требования заказчиком.

В процессе выполнения проекта стейкхолдерами выступили директор стоматологии и старшая медсестра. Посредством интервьюирования были получены пользовательские требования, приведенные в табл. 3.1. Идентифицированные требования подвергались процедуре приоритизации с целью выявления первостепенных задач, с последующим определением номера витка спирали для их выполнения. Часто используемыми методами приоритизации на данный момент являются: Кано, MoSCoW, QFD, User Story Mapping и Lean Prioritization [6]. Наиболее наглядным видится метод MoSCoW, суть которого заключается в присваивание к каждому требованию одной из четырех категорий важности:

- наиболее важное и срочное (Must);
- важное (Should);
- может быть отложено на некоторое время (Could);
- не является приоритетным вообще (Would).

Совокупность требований и их приоритетов даны в табл. 3.1.

Таблица 3.1. Бэклог требований и их приоритетов

№	Пользовательское требование	Функциональное требование	Предполагаемый компонент программы, реализующий требование	Приоритет
1	Наличие личной карточки пациента с основными данными	Таблица «Пациент» с необходимыми полями для внесения данных из анкеты и документов	Экран по ведению данных о пациенте	Must
2	Наличие перечня сопутствующих заболеваний и противопоказаний	Таблица «Сопутствующие заболевания» и таблица «Противопоказания»	Экран по ведению данных сопутствующих заболеваниях и противопоказаниях	Must
3	Наличие анамнеза пациента	Таблица «Анамнез» с полями для данных из опроса пациента с	Экран по ведению данных об анамнезе пациента	Must

№	Пользовательское требование	Функциональное требование	Предполагаемый компонент программы, реализующий требование	Приоритет
		датой обращения		
4	Возможность фиксации результатов осмотра ротовой полости в «Зубной карте»	Таблицы «Карта зубов» и «Описание зуба»	Экран по ведению записи результатов осмотра	Must
5	Наличие информация о специалистах стоматологии	Таблица «Персонал», с необходимыми данными о специалистах	Экран по ведению данных о специалистах стоматологии	Must
6	Наличие информации об оказываемых в стоматологии услугах	Таблица «Услуги и цены» содержащая оказываемые процедуры и их стоимость	Экран по ведению данных об услугах	Must
7	Возможность записи и изменения информации в таблицах	–	Экран для ввода информации	Should
8	Показ информации по таблицам	Вывод на экран запрашиваемой информации из таблиц	Экран для вывода информации из запрашиваемых таблиц	Should
9	При запросе информация не должна быть показана вся сразу	Диалоговая форма и разворачивающиеся списки	Диалоговый экран программы и разворачивающиеся списки	Could
10	Возможность хранения рентген снимков	Таблица «Рентген снимки» с сохранёнными фотографиями	Средства работы с ссылками на изображения	Could

№	Пользовательское требование	Функциональное требование	Предполагаемый компонент программы, реализующий требование	Приоритет
11	Возможность отдельного доступа для врача и пациента с различным уровнем доступа к просмотру и изменению информации	Возможность авторизации пользователей	Разделение информации по уровню доступа	Could

Для реализации проекта с использованием спиралевидной модели необходимо распределить требования по виткам спирали в соответствии с их важностью. Последовательность шагов реализации приложения принимает вид:

- предварительные действия:
 - проработка предметной области;
 - опрос стейкхолдеров и фиксация требований (таблица 3.1);
 - моделирование бизнес-модели с использованием нотаций ARIS VACD и eEPC;
 - проектирование таблиц баз данных и схемы приложения;
- 1-й виток спирали:
 - проектирование решения для требований 1-6 из таблицы 3.1;
 - оценка возможных рисков и их влияния;
 - разработка требований и проведение модульного испытания;
 - предоставление заказчику разработанной части программы;
- содержание 2-го витка спирали аналогично 1-му, за исключением требований: анализируются результаты предыдущей итерации разработки и реализуется требование 7 из табл. 3.1;
- 3-й виток спирали, его содержание аналогично 2-му витку, однако разрабатывается требование 8 из табл. 3.1;
- содержание 3-го витка аналогично 2-му: реализуются требования 9-11 из табл. 3.1.

Литература

1. Программы для стоматологий – URL: <http://www.livemedical.ru/tools/dental/> (дата обращения 12.12.2018).
2. Карпович Е.Е. Жизненный цикл программного обеспечения – М.: Центр дистанционного обучения. НИТУ «МИСиС», 2016. – 131 с.
3. Кумагина Е.А., Неймарк Е.А. Модели жизненного цикла и технологии проектирования программного обеспечения: учебно-методическое пособие – Нижний Новгород: Изд-во ННГУ, 2016. – 41 с.
4. Гурендо Д. Жизненный цикл разработки ПО – URL: <https://xbsoftware.ru/blog/zhiznennyj-tsykl-razrabotki-spiral/> (дата обращения 12.12.2018).
5. Требования. Анализ требований, виды требований – URL: <https://intellect.ml/trebovaniya-analiz-trebovanij-vidy-trebovanij-5188> (дата обращения 12.12.2018).
6. Методы приоритизации – URL: <https://habr.com/company/hygger/blog/359208/> (дата обращения 12.12.2018).
7. Степанов Д.Ю. Анализ, проектирование и разработка корпоративных информационных систем: теория и практика // Российский технологический журнал. – 2015. – т.8, №3. – с.227-238.
8. Разработка управленческих решений / Ю.Г. Учитель, А.И. Терновой, К.И. Терновой. – 2-е изд., перераб. и доп. – М., 2007. – 383 с.
9. Виды тестирования программного обеспечения – URL: <http://www.protesting.ru/testing/testtypes.html> (дата обращения 12.12.2018).
10. Куликов С. С. Тестирование программного обеспечения. Базовый курс. – Минск: Четыре четверти, 2017. – 312 с.

Выходные данные статьи

Худяков С.Д. Автоматизация работы стоматологической клиники с использованием спиралевидной модели внедрения информационных систем (часть 1) // Корпоративные информационные системы. – 2021. – №1 (13) – С. 1-10. – URL: <https://corpinfosys.ru/archive/issue-13/93-2021-13-dentalautomation>.

Об авторе



Худяков Сергей Дмитриевич – студент 4-го курса кафедры оптических и биотехнических систем и технологий физико-технологического института РТУ МИРЭА. Тема выпускной квалификационной работы бакалавра «Автоматизация ключевых бизнес-процессов стоматологической клиники с использованием спиралевидной модели внедрения». Электронная почта: khudyakov.sd@yandex.ru.