

Применение Agile Scrum для реализации автоматизированного рабочего места врача терапевта в городской больнице (часть 1)

Болохнов Александр Андреевич

Аннотация: в статье продемонстрированы возможности гибкой методологии разработки Agile, в частности метод Scrum. Сформулированы пользовательские требования врача-терапевта, составлен бэклог продукта и спринтов. Для моделирования ключевых бизнес-процессов верхнего уровня применена нотация ARIS VACD.

Введение

Всемирное использование компьютерных технологий в медицине началось с появлением компьютеров в начале 1950-х годов. В 1949 году в Германии Густав Вагнер учредил первую профессиональную организацию по информатике в области здравоохранения. Информатизация здравоохранения (Health Information Systems) – это дисциплина на стыке информации, информатики и здравоохранения. Дисциплина касается ресурсов, устройств и методов, необходимых для оптимизации сбора, хранения, извлечения и использования информации в области здравоохранения и биомедицины. В состав инструментов информатики в области здравоохранения включаются компьютеры, клинические руководства, официальная медицинская терминология, информационные и коммуникационные системы. Все это применяется в областях сестринского дела, клинической помощи, стоматологии, фармацевтики, здравоохранения, профессиональной терапии, биомедицинских исследований и многих других.

Важным фактором развития информационных технологий в области медицины стало создание медицинских баз данных и автоматизированных рабочих мест (далее – АРМ). Ежедневно врач принимает значительное количество пациентов за смену. Медицинские базы данных помогают решать проблему учета всех пациентов, систематизацию их данных и ускоряют процесс получения необходимой информации. Автоматизированное рабочее место врача представляет собой комплекс вычислительной техники и программного обеспечения, который располагается на рабочем месте врача-специалиста и служит для автоматизации его деятельности. Совокупность обозначенных выше решений позволяет не тратить рабочее время врача

на поиск информации в бумажных карточках и картотеках, а использовать его непосредственно для обследования пациента и оперативно назначать необходимое лечение.

Использование систем управления базами данных, как например, Microsoft Access, в которой и будет разработано приложение АРМ врача-терапевта, позволит существенно упростить и ускорить процесс приема врачом пациента и работы с его персональными данными. Поэтому цель данной работы состоит в автоматизации ключевых бизнес-процессов врача-терапевта на основе метода Agile Scrum [1]. Разрабатываемая программа позволит оптимизировать работу врача-терапевта и улучшить качество предоставляемых медицинских услуг населению.

1. Идентификация требований

Идентификация предварительных требований позволяет, заданных пожеланиями клиента, позволяет определить характеристики нового продукта. Зачастую клиент не может описать конкретные возможности желаемой системы, а лишь формулирует свои нечёткие представления о конечном виде продукта. В связи с этим грамотное определение требований становится первым важным этапом на пути создания программного продукта или модернизации уже имеющегося. Следует выделить четыре основных этапа работы с требованиями:

- сбор требований;
- анализ требований;
- документирование требований;
- проектирование системы.

Заказчиком был выбран врач-терапевт. В качестве ключевого процесса для оптимизации является учет анализов пациентов. Терапевт – врач общей практики, его задача – поставить диагноз, проанализировав жалобы пациента и при необходимости направить его к более узкому специалисту. Врач-терапевт должен обладать широкими знаниями, в том числе в области гастроэнтерологии, хирургии, кардиологии и др. Участковые врачи-терапевты ведут прием в поликлинике и по вызову больного на дому. В обязанности данного врача входит также заполнение большого объема бумажных документов: амбулаторных карточек, больничных листов и отчетов [2].

После обсуждения с заказчиком был составлен список пользовательских требований. Пользовательские требования – это короткие, простые описания функций, которые рассматриваются с точки зрения конечного пользователя. Пользовательские

требования описывают, что хочет пользователь и почему. Они помогают создать упрощенное описание конечного продукта. Обычно пользовательские требования имеют следующий вид: «Как < тип пользователя >, я хочу < какая-то цель >, исходя из того, что < причина >» [3]. Пример: как администратор, я хочу просмотреть фотографии до их публикации, чтобы иметь возможность убедиться, что они соответствуют требованиям. Для автоматизации выбраны следующие операции:

- учет анализов пациентов;
- выдача направлений на обследование;
- назначение лекарственных препаратов.

На основании пользовательских пожеланий и с учетом ожидаемых улучшений составлен следующий список требований:

- возможность хранения и просмотра данных о пациенте;
- возможность создания, редактирования и удаления данных о пациентах;
- возможность поиска по заданным параметрам;
- возможность назначить анализ для пациента;
- возможность назначить обследование для пациента;
- возможность создания рецепта на выдачу лекарственных препаратов;
- возможность отображения выданных пациенту рецептов на лекарственные препараты;
- возможность редактировать и удалять назначенный анализ; возможность редактирования и удаления рецептов на выдачу лекарственных препаратов; возможность редактировать и удалять ранее назначенные обследования;
- возможность поиска выданных рецептов пациенту по заданным параметрам; возможность поиска назначенных пациенту обследований по заданным параметрам;
- возможность отображения даты и времени назначенного анализа;
- возможность отображения даты и времени поступления результатов анализа;
- возможность отображения результатов анализов пациента; защита данных от несанкционированного доступа;
- возможность создания отчетов о доступных и выданных препаратах, выписанных направлениях на обследование;
- возможность отображения наличия на складе аптечного киоска больницы лекарственных препаратов.

Приоритизация требований является одной из основных концепций в Agile практики. Приоритизация – это принятие решений о том, в каком порядке команда разработчиков будет работать над требованиями в проекте. Понимание

приоритетности имеет особенно важное значение при проектировании, поскольку проект имеет временную шкалу с фиксированным набором ресурсов. Приоритеты задаются в зависимости от значимости на три основные категории: высокая, средняя, низкая. Высокая значимость определяет требования, необходимые для полноценной работы всей программы. Средняя значимость характеризует требования, значительно упрощающие процесс работы с программой. Низкая значимость определяет требования, которые не являются необходимыми для функционирования программы, но их реализация положительно отразится на релизе. На основании приведенного выше подхода к приоритизации, составляется список ранжированных пользовательских требований. Далее задаем функциональные требования (то, какие действия будут выполняться системой для реализации пользовательских требований):

- отображение в таблице данных о пациентах;
- использование компонентов программы, обеспечивающие создание, редактирование и удаление данных о пациентах;
- использование компонентов программы, обеспечивающие поиск данных по ключевым параметрам;
- использование компонентов программы, обеспечивающих возможность назначать анализы пациентам;
- использование компонентов программы, обеспечивающих возможность назначать обследование пациентам;
- использование компонентов программы, обеспечивающих возможность создания рецептов на выдачу лекарственных препаратов пациентам;
- отображение выданных пациенту рецептов на лекарственные препараты;
- использование компонентов программы, обеспечивающих возможность редактировать и удалять: назначенный анализ, рецепт на выдачу лекарственных препаратов, назначенное обследование;
- использование компонентов программы, обеспечивающих поиск выданных пациенту рецептов на лекарственные препараты и назначенных обследований по заданным параметрам;
- отображение даты и времени назначенного анализа в отдельной графе интерфейса программы;
- отображение на экране данных о дате и времени поступления результатов анализа;
- отображение информации, содержащей результаты анализов пациента;
- использование программного кода, обеспечивающего защиту от несанкционированного доступа, реализация формы авторизации пользователя в системе;

- использование программного кода, позволяющего вводить и проверять введенный пользователем пароль;
- использование компонентов программы, обеспечивающих возможность создания отчетов;
- использование компонентов программы, позволяющих отображать наличие лекарственных препаратов на складе аптечного киоска больницы.

После сбора и анализа пользовательских и функциональных требований строится бэклог продукта, что упрощает отслеживание требований и позволяет контролировать их выполнение. Бэклог дан в таблице 1.1.

Таблица 1.1. Бэклог продукта

№	Приоритет	Пользовательское требование	Функциональное требование
1	Высокая	Возможность хранения и просмотра данных о пациенте	Отображение в таблице данных о пациентах
2	Высокая	Возможность создания, редактирования и удаления данных о пациентах	Использование компонентов программы, обеспечивающие создание, редактирование и удаление данных о пациентах
3	Высокая	Возможность поиска по заданным параметрам	Использование компонентов программы, обеспечивающие поиск данных по заданным параметрам
4	Высокая	Возможность назначить анализ для пациента	Использование компонентов программы, обеспечивающих возможность назначать анализы пациентам
5	Высокая	Возможность назначить обследование для пациента	Использование компонентов программы, обеспечивающих возможность назначать обследование пациентам
6	Высокая	Возможность создания рецепта на выдачу лекарственных препаратов	Использование компонентов программы, обеспечивающих возможность создания рецептов на выдачу лекарственных препаратов пациентам
7	Высокая	Возможность отображения выданных пациенту	Отображение выданных пациенту рецептов на

№	Приоритет	Пользовательское требование	Функциональное требование
		рецептов на лекарственные препараты	лекарственные препараты
8	Средняя	Возможность редактировать и удалять назначенный анализ; возможность редактирования и удаления рецептов на выдачу лекарственных препаратов; возможность редактировать и удалять ранее назначенные обследования	Использование компонентов программы, обеспечивающих возможность редактировать и удалять: назначенный анализ, рецепт на выдачу лекарственных препаратов, назначенное обследование
9	Средняя	Возможность поиска выданных рецептов пациенту по заданным параметрам; возможность поиска назначенных пациенту обследований по заданным параметрам	Использование компонентов программы, обеспечивающих поиск выданных пациенту рецептов на лекарственные препараты и назначенных обследований по заданным параметрам
10	Низкая	Возможность отображения даты и времени назначения анализа	Отображение даты и времени назначенного анализа в отдельной графе интерфейса программы
11	Низкая	Возможность отображения даты и времени поступления результатов анализа	Отображение на экране данных о дате и времени поступления результатов анализа
12	Низкая	Возможность отображения результатов анализов пациента; защита данных от несанкционированного доступа	Отображение информации, содержащей результаты анализов пациента; использование программного кода, обеспечивающего защиту от несанкционированного доступа, реализация формы авторизации пользователя в системе; использование программного кода, позволяющего вводить и проверять введенный

№	Приоритет	Пользовательское требование	Функциональное требование
			пользователем пароль
13	Низкая	Возможность создания отчётов о: доступных и выданных препаратах, выписанных направлениях на обследование	Использование компонентов программы, обеспечивающих возможность создания отчетов
14	Низкая	Возможность отображения наличия на складе аптечного киоска больницы лекарственных препаратов	Использование компонентов программы, позволяющих отображать наличие лекарственных препаратов на складе аптечного киоска больницы

Один из основных элементов при разработке проекта согласно Agile Scrum – спринт (Sprint). По сути, это короткий цикл разработки, следующий один за другим без перерывов. Приоритизированные выше требования служат основой для формирования спринтов. Первый спринт позволит спроектировать будущее приложение с точки зрения моделирования процесса, данных и экранных форм. Ко второму спринту будут относиться все требования с высокой степенью приоритизации, третьему – средней, четвертому – низкой. Разработка приложения в среде MS Access 2010 будет осуществляться в последовательности, данной ниже. Первый спринт включает активности по:

- моделированию ключевого бизнес-процесса нотации VACD и DFD;
- проектированию баз данных и моделированию пользовательских интерфейсов;
- уточнению пользовательских и функциональных требований;
- планированию последующих спринтов.

Для каждого следующего спринта будут проведены следующие действия: детализация пользовательских и функциональных требований, планирование спринта, тестирование реализованных возможностей, обзор спринта и ретроспектива. Поэтому опишем только отличительные особенности каждого спринта. Второй спринт, сформированный на основе требований высокого приоритета, определяется задачами:

- разработка возможности отображения, изменения и удаления данных о пациентах; поиска данных пациентов по заданным параметрам; назначения

анализов и обследований; выдачи рецептов на лекарственные препараты пациентам (требования 1-7 в бэклоге продукта из таблица 1.1);

- создание первой тестовой версии интерфейса программы.

Третий спринт, заданный средней степенью важности требований, включает задачи:

- разработки возможностей редактирования и удаления назначенных пациенту анализов, рецептов на выдачу лекарственных препаратов, а также обработки ранее назначенных обследований (требования 8-9 в бэклоге продукта из таблицы 1.1);
- доработки интерфейса программы.

Четвертый спринт включает низкоприоритетные требования и активности по:

- разработке возможности отображения даты и времени назначения анализа, даты и времени поступления результатов анализа; результатов анализов пациента; использования компонентов программы, позволяющих вводить и проверять введенный пользователем пароль; создания отчетов, включая отображение наличия на складе аптечного киоска больницы лекарственных препаратов (требования 10-14 в бэклоге продукта из таблицы 1.1);
- созданию итоговой версии программы.

2. Проектирование процессов, данных и приложения на 1-м спринте

Бизнес-процесс – это деятельность или набор действий, которые выполняют определенную организационную цель. Бизнес-процессы должны иметь целенаправленность, быть максимально конкретными и иметь последовательные результаты. Одним из эффективных методов преобразования идеи в конкретные результаты является разработка и заполнение диаграмм As-Is (как есть) и To-Be (как будет) [4]. Диаграмма As-Is описывает текущее состояние процесса и возможностей организации. Диаграмма To-Be описывает будущее состояние, другими словами, какие процессы и возможности организации появятся в будущем [5].

Для начала необходимо определить процесс в As-Is, как только он задан, необходимо его проанализировать. В момент анализа требуется задокументировать текущее состояние процесса и сформировать точку отсчета. Требуется четко понимать функционирование, возможности процесса и системы. После чего можно приступить к графическому отображению процессов в модели As-Is. Далее выявляются

недостатки в существующих процессах, после чего ведется документирование и реализация процесса модели To-Be. На этом же этапе производится детальный анализ желаемых улучшений процесса и формируются конкретные предложения по его усовершенствованию. Далее, как и в модели As-Is, создается графическое отображение. Это наиболее сложный этап, при его реализации часто может оказаться, что желаемые улучшения не столь эффективны, как планировалось ранее. Последним шагом будет отслеживание реальной эффективности и актуальности введенных улучшений.

Одной из наиболее популярных методологий для моделирования бизнес-процессов является методология ARIS (Architecture of Integrated Information Systems), часто называемая архитектурой интегрированных информационных систем [6]. Методология состоит из различных нотаций (система условных обозначений, принятая в конкретной модели) и позволяет описать деятельность организации с нескольких точек зрения. ARIS рассматривает компанию с четырех направлений: организационная структура, информационная структура, функциональная иерархия, управление бизнес-процессами. Преимуществом моделирования ARIS является наглядность в представлении сложных фактов, что позволяет систематизировать подход к анализу. В зависимости от интересующей информации используются различные методы моделирования. К числу часто используемых нотаций ARIS относятся: VACD (Value-added Chain Diagram, диаграмма цепочки процесса, добавляющего стоимость), нотации eEPC (Extended Event-driven Process Chain, расширенная нотация цепочки процесса, управляемого событиями), PCD (Process Chain Diagram, диаграмма цепочки процесса), Organizational Chart (организационная диаграмма), FunctionTree (дерево функций) и Product Tree (дерево продуктов) [7]. В работе будет рассмотрено применение графической нотации VACD.

Нотация VACD - это тип модели, используемый для формулировки основных уровней процесса. VACD применяется для описания процессов верхнего уровня внутри организации. Основной объект нотации - процесс или группа функций организации, необходимые для получения добавленной стоимости [8].

Для описания нижних уровней организации будет использована диаграмма потоков данных (DFD, Data flow diagrams) [9]. Эта диаграмма отображает поток информации для любого процесса или системы, для чего используются определенные графические символы, такие как: прямоугольники, круги и стрелки, а также короткие текстовые метки для отображения входов данных, выходов, точек

хранения и маршрутов между каждым пунктом. Блок-схемы данных могут варьироваться от простых отрисованных вручную процессов, до глубоких многоуровневых, которые постепенно детализируют процесс обработки данных. Диаграммы могут использоваться для анализа существующей системы или моделирования новой [8].

2.1. Проектирование процессов в модели AS-IS с помощью VACD/DFD

На рисунке 2.1 представлена диаграмма функционирования больницы As-Is (как есть) в нотации VACD. Основу функционирования составляют три процесса, а именно: пациент обратился в больницу, лечить пациента и выписать пациента. Также на диаграмме даны ответственные за каждый процесс и входящие объекты.

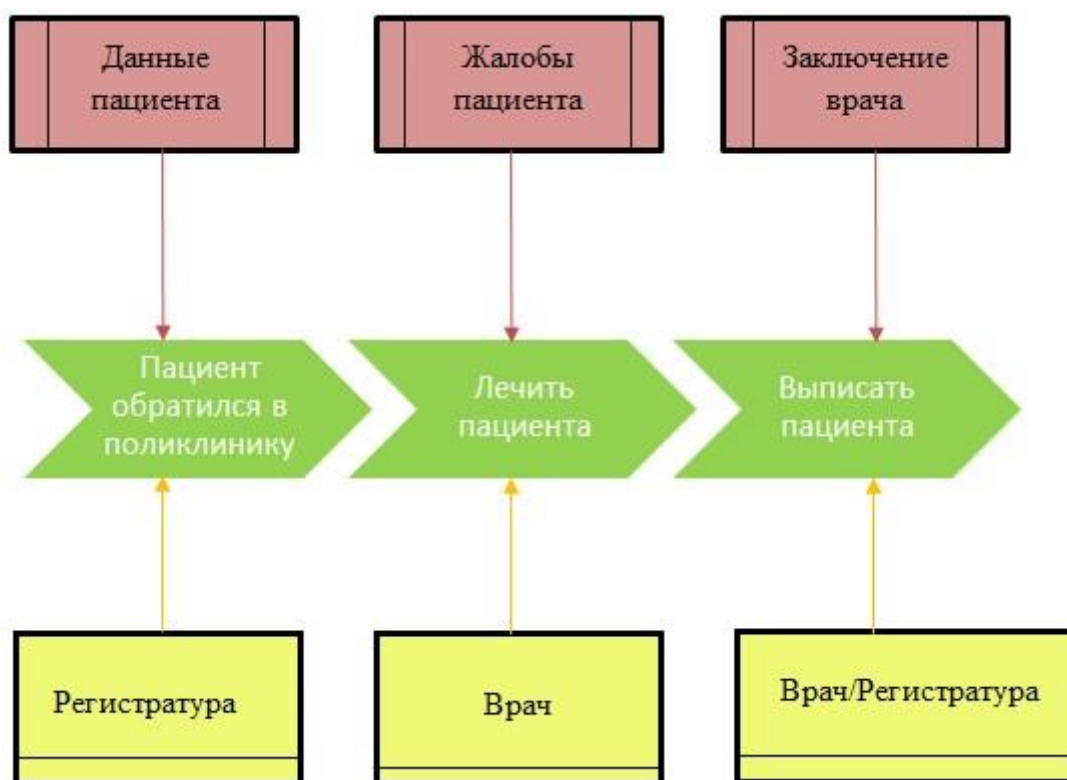


Рис. 2.1. Ключевые процессы 1-го уровня описания в нотации VACD

Процесс лечения пациента следует рассмотреть более подробно, так как именно он представляет наибольшую ценность в работе врача-терапевта. Для этого

подвергнем процесс декомпозиции. Для чего используем нотацию DFD. Результат декомпозиции процесса на втором уровне с применением нотации DFD представлен на рисунке 2.2. Как видно из рисунка, процесс лечения пациента является сложным и многоступенчатым, состоящим из множества подпроцессов.

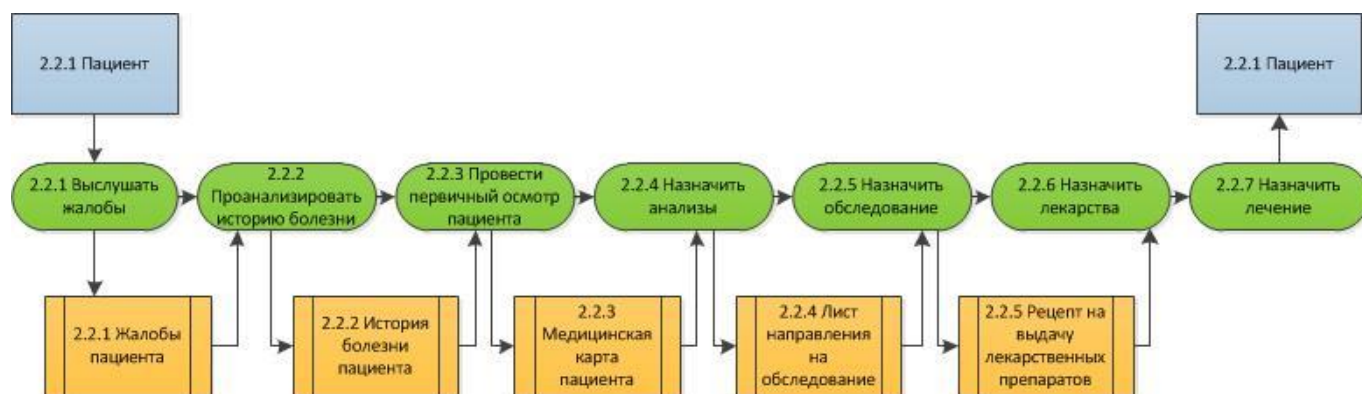


Рис. 2.2. Процесс «Лечить пациента» на 2-м уровне модели As-Is в нотации DFD

Подпроцессы назначения анализов, обследования и лекарств выбраны в качестве ключевых операций для оптимизации, поэтому подвергаем их дальнейшей декомпозиции (рис. 2.3-2.5).

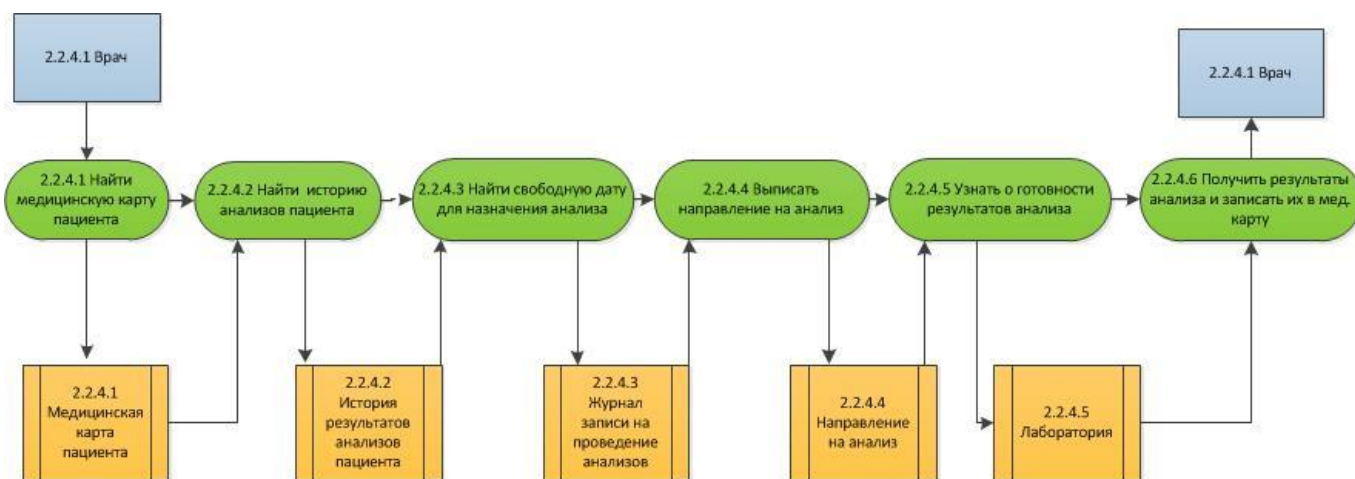


Рис. 2.3. Процесс «Назначить анализы» на 3-м уровне модели As-Is в нотации DFD



Рис. 2.4. Процесс «Назначить обследование» на 3-м уровне модели As-Is в нотации DFD



Рис. 2.5. Процесс «Назначить лекарства» на 3-м уровне модели As-Is в нотации DFD

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-protssesov-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 1) // Корпоративные информационные системы. – 2020. – №2 (10) – С. 24-37. – URL: <https://corpinfosys.ru/archive/issue-10/107-2020-10-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.