

Министерство образования и науки Российской Федерации  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
Кафедра «Прикладная математика и информатика»

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА<sup>1</sup>

на тему «Разработка автоматизированного нагрузочного контура для  
электронной торговой системы»

Студент	<u>Д.А. Чижов</u>	_____
	(И.О. Фамилия)	(личная подпись)
Руководитель	<u>А.В. Шляпкин</u>	_____
	(И.О. Фамилия)	(личная подпись)

**Допустить к защите**

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский \_\_\_\_\_  
(ученая степень, звание, И.О. Фамилия) (личная подпись)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ г.

Тольятти 2018

<sup>1</sup> Бакалаврская работа



## АННОТАЦИЯ

Нагрузочное тестирование является важным звеном в процессе тестирования и отладки программного обеспечения. К настоящему моменту на тему автоматизации процесса нагрузочного тестирования написано несколько диссертаций, других научных работ и статей. Разработана теория нагрузочного тестирования, методики его автоматизации, а также некоторые технологические подходы к его осуществлению. Однако из-за специфики отдельных информационных систем необходимо приложить серьёзные усилия для реализации общих подходов на практике применительно к конкретному программному продукту.

Тема данной работы «Разработка автоматизированного нагрузочного контура для электронной торговой системы». В результате выполнения работы был произведён анализ предметной области функционирования исследуемой торговой системы – валютного дилинга. Было установлено, что существующая в компании процедура нагрузочного тестирования не является оптимальной. Для устранения этих недостатков, а также уменьшения трудовых и временных затрат сотрудников компании, было предложено разработать и реализовать программные средства развёртывания и функционирования нагрузочного контура электронной торговой платформы. Результатом работы является спроектированный и реализованный набор инструментов для организации нагрузочного тестирования. Отдельное внимание уделяется обоснованию экономической эффективности внедрения реализованных решений в бизнес-процессы компании.

Работа включает 666 страниц печатного текста, количество таблиц – 8, количество рисунков – 17. Количество библиографических источников – 19, из них на иностранном языке – 7.

## ОГЛАВЛЕНИЕ

Введение.....	5
Глава 1. Анализ предметной области о деятельности электронной торговой системы.....	8
1.1. Техничко-экономическая характеристика деятельности электронной торговой системы.....	8
1.2. Концептуальное моделирование деятельности электронной торговой системы.....	17
1.3. Постановка задачи на разработку автоматизированного нагрузочного контура для электронной торговой системы.....	28
1.4. Анализ существующих разработок и обоснование выбора технологии проектирования.....	33
Выводы по первой главе.....	35
Глава 2. Разработка и реализация автоматизированного нагрузочного контура для электронной торговой системы.....	36
2.1. Логическое моделирование автоматизированного нагрузочного контура для электронной торговой системы.....	36
2.2. Физическое моделирование автоматизированного нагрузочного контура для электронной торговой системы.....	48
2.3. Технологическое обеспечение задачи автоматизированного нагрузочного контура для электронной торговой системы.....	52
2.4. Контрольный пример реализации автоматизированного нагрузочного контура для электронной торговой системы.....	53
Выводы по второй главе.....	56
Глава 3. Оценка и обоснование экономической эффективности внедрения автоматизированного нагрузочного контура для электронной торговой системы.....	57

3.1. Выбор и обоснование методики расчета экономической эффективности от автоматизированного нагрузочного контура для электронной торговой системы.....	57
3.2. Расчёт показателей экономической эффективности проекта реализации автоматизированного нагрузочного контура для электронной торговой системы.....	59
Выводы по третьей главе.....	62
Заключение.....	63
Список используемых источников.....	65

## ВВЕДЕНИЕ

В современном мире работа торговой биржи буквально пронизана информационными технологиями на каждом уровне организационной иерархии и на любом этапе совершения сделки. Традиционные формы торговли с использованием телефонов и факсимильных аппаратов ушли в прошлое. На смену им пришла работа с торговыми терминалами. Больше половины сделок на таких международных финансовых рынках как Forex совершают алгоритмические трейдеры – специальные программы, выставляющие заявки по определённому алгоритму.

К крупным участникам рынка, выполняющим роль поставщиков ликвидности на международных финансовых рынках, предъявляются жёсткие требования по скорости обработки клиентских торговых поручений. В идеале время отклика не должно превышать нескольких микросекунд. В последнее время активно развивается такое направление финансовых услуг как агрегация ликвидности, то есть объединение ценовых потоков от нескольких поставщиков. Для выполнения такой задачи используются самые современные информационные и технические средства, так как положительный эффект от агрегирования должен превышать накладные расходы на него [1].

Информационные технологии в финансовой сфере представляют собой область применения передовых решений. С точки зрения архитектуры электронные системы торгов представляют собой сложные распределённые системы, использующие параллельные вычисления. К таким программным продуктам предъявляются повышенные требования не только в плане производительности, но также надёжности, отказоустойчивости, наличия технологии восстановления после сбоев. В связи с этим особую важность приобретает вопрос тестирования и отладки сложных информационных систем.

Помимо функционального тестирования важным является нагрузочное тестирование, так как оно позволяет установить допустимые пределы нагрузки, которые может выдержать данная информационная система. Нагрузочное тестирование — это автоматизированное тестирование, имитирующее работу определенного количества бизнес пользователей на каком-либо общем (разделяемом ими) ресурсе.

Тема данной работы «Разработка автоматизированного нагрузочного контура для электронной торговой системы». Существующий вариант подготовки тестовой среды, развёртывания системы и анализа результатов является трудоёмким, а полученные в результате данные не отвечают требованиям полноты и достоверности. Автоматизация процесса нагрузочного тестирования позволит устранить данные недостатки, повысить надёжность и отказоустойчивость программного обеспечения, что в конечном итоге принесёт компании ощутимый экономический эффект.

К настоящему моменту на тему автоматизации процесса нагрузочного тестирования написано несколько диссертаций, других научных работ и статей. Также изданы книги, посвящённые тестированию, в том числе тестированию производительности [2]. Разработана теория нагрузочного тестирования, методики его автоматизации, а также некоторые технологические подходы к его осуществлению. Однако из-за специфики отдельных информационных систем необходимо приложить серьёзные усилия для реализации общих подходов на практике применительно к конкретному программному продукту.

Целью работы является разработка автоматизированного нагрузочного контура используемой в компании информационной торговой системы. Задачами работы являются выработка подходов и методов, применимых для нагрузочного тестирования информационной системы, моделирование предметной области, разработка архитектуры программного обеспечения, используемого в целях тестирования, а также реализация разработанных моделей на практике.

Объектом исследования данной работы является производительность информационной торговой системы, используемой в компании. Предметом рассмотрения являются способы автоматизации процесса исследования производительности. Для исследования предмета и достижения целей работы применяются методы структурного подхода к анализу и проектированию, методология объектно-ориентированного анализа и проектирования.

Исследуемая информационная система электронных торгов, хотя и имеет аналоги, является уникальной в своём роде. Автоматизированный нагрузочный контур для неё разрабатывается впервые. Решение и реализация поставленных задач принесёт ощутимые практические результаты, выраженные в снижении трудоёмкости процессов, финансовых затрат, а также улучшении конкурентного положения компании на рынке финансовых услуг.

# ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ О ДЕЯТЕЛЬНОСТИ ЭЛЕКТРОННОЙ ТОРГОВОЙ СИСТЕМЫ

## 1.1. Техничко-экономическая характеристика деятельности электронной торговой системы

### 1.1.1. Характеристика предприятия и его деятельности

Научно-технический прогресс и развитие информационно-коммуникационных технологий способствуют глобализации экономических и финансовых процессов. Появление электронных торговых систем сделало возможным создание глобальных торговых площадок. Участником торгов на таких площадках может быть экономический субъект, географически находящийся в любой точке земного шара. С выходом российских экономических субъектов на мировые рынки возрастает взаимосвязь их бизнес-процессов с зарубежными контрагентами. Благодаря этому упрощаются процессы совершения торговых сделок, увеличиваются доступные объёмы ликвидности, расширяются возможности инвестирования капитала. Это позволяет банкам, брокерским организациям и инвестиционным фондам предлагать клиентам широкий спектр финансовых продуктов и услуг, удовлетворяющих любому профилю риска.

Одновременно возрастает уровень конкуренции на всех нишах финансовой иерархии: от крупных банков до небольших обслуживающих фирм, предоставляющих услуги по автоматизации финансовой деятельности. Тот поставщик ликвидности, который сможет предоставить своим клиентам более качественный сервис, меньшее время отклика и требуемый объём ликвидности, будет иметь преимущество на этом динамично развивающемся рынке [3].

Современной мировой тенденцией является рост доли высокочастотной торговли в общем количестве и объёме совершаемых операций. Современные торговые площадки имеют возможность обработать

клиентскую заявку в течение десятков микросекунд. Для достижения таких показателей используются самые передовые информационные технологии. Для обеспечения максимальной производительности создаются распределённые электронно-коммуникационные сети, включающие торговые сервера, оснащённые процессорами с несколькими десятками ядер. Для уменьшения задержек, связанных с передачей данных по сети, прокладываются высокопропускные оптоволоконные каналы. Для обеспечения максимальной информационной безопасности торговый контур размещается в специализированных дата-центрах.

Крупные поставщики ликвидности, например Barclays, JPMorgan, Credit Suisse, Société Générale и другие являются одновременно пионерами электронного трейдинга, поэтому к настоящему времени имеют самую развитую инфраструктуру. Казалось бы, более мелким игрокам трудно составить конкуренцию этим гигантам. Однако, при более внимательном рассмотрении оказывается, что предоставляемые мировыми лидерами услуги не всегда подходят для российского потребителя. Основной причиной этого является ориентация западных финансовых структур на западные торговые площадки. Для российских трейдеров важно иметь доступ к российским рынкам ценных бумаг и совершать конверсионные операции с использованием национальной валюты. Кроме того, российские провайдеры ликвидности могут иметь преимущество за счёт изначального размещения в юрисдикции РФ и адаптированности бизнес-процессов к российскому законодательству.

Отдельным направлением в области услуг электронной торговли является интеграция финансовых потоков с нескольких мировых площадок. Клиенту может быть неудобно и накладно иметь несколько счетов в разных банках и брокерских фирмах. Поэтому брокеры пытаются организовать сбор котировок с нескольких торговых площадок, предоставляя клиентам агрегированный рыночный стакан с лучшими ценами.

Рассмотрим деятельность Компания NTFX Capital Ltd. (далее «NTFX»), которая является международным брокером рынка Форекс, предоставляющим свои услуги, как частным, так и институциональным клиентам. В целом полный пакет данных услуг можно разделить на инвестиционные и дополнительные. К инвестиционным услугам относятся: прием и передача ордеров в отношении одного или нескольких финансовых инструментов, выполнение ордеров по поручению клиентов, а также различные операции по клиентскому счету. Современный финансовый брокер, как правило, не ограничивается только инвестиционной деятельностью. Для того, чтобы замкнуть на себя удовлетворение всех потребностей клиентов, связанных с торговой деятельностью, такие компании могут выполнять функции, свойственные банкам, депозитариям, клиринговым организациям, а также осуществлять аналитическую и консультационную деятельность. Так компания NTFX обеспечивает хранение и управление финансовыми инструментами, в том числе хранение ценностей и сопутствующие услуги [13].

Маржинальная торговля предоставляет, как брокеру, так и его клиентам новые возможности по увеличению финансового оборота – одного из главных показателей финансовой деятельности. Под маржинальной, в данном случае, понимается ведение спекулятивной торговли с использованием финансовых инструментов, предоставляемых торговцу в кредит под залог оговоренной суммы. Компания NTFX предоставляет кредиты или займы на один или более финансовых инструментов. Также стоит отметить услуги по обмену валюты, без чего сложно представить инвестиционную деятельность.

В NTFX применяется стандартное для средней финансовой организации разбиение бизнес-архитектуры на фронт-, мидл- и бэк-офис [4].

К фронт-офису относятся отделы продаж, финансовых продуктов и служба клиентской поддержки.

К мидл-офису относятся отделы расчётов и оформления сделок, работы с банками, работы с поставщиками ликвидности, операций на финансовом рынке, а также дилерский и аналитический отделы. Кроме-того к данному объекту бизнес-архитектуры относится управление торговых клиентских операций.

К бэк-офису относятся бухгалтерия, службы финансов, кадров, юридическая служба и отдел информационных технологий.

На рисунке 1.1 изображена организационная структура компании NTFX.

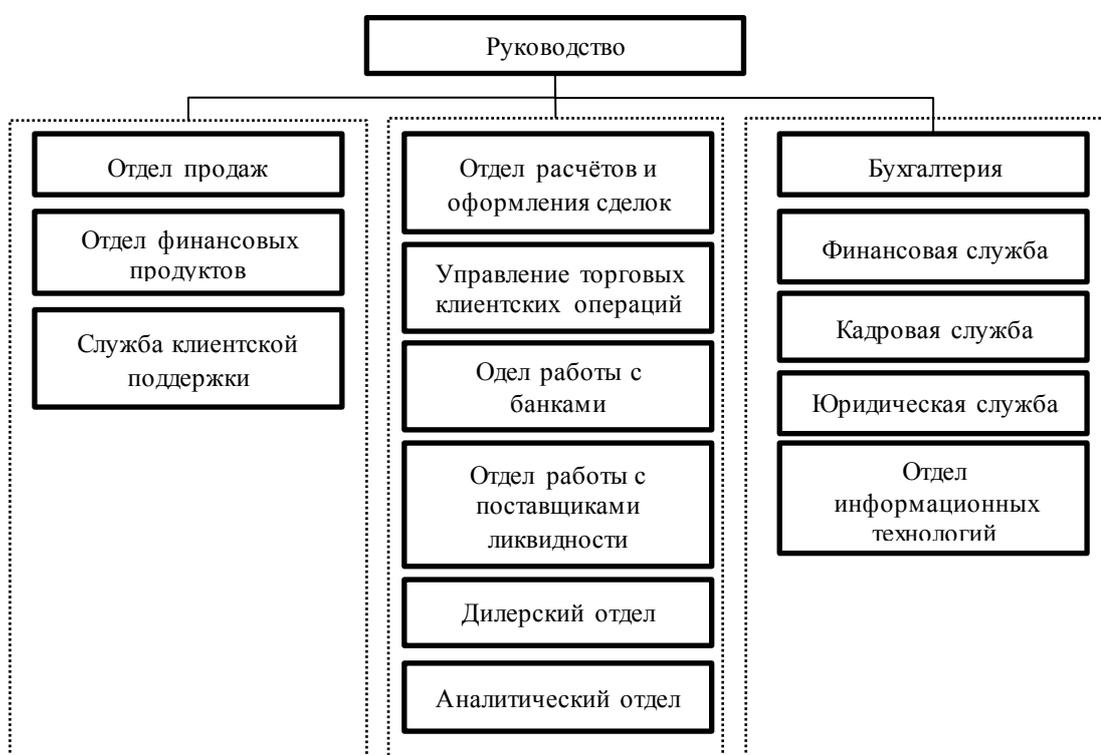


Рис. 1.1 – Организационная структура компании NTFX Capital.

Как видно из представленной схемы на уровне мидл-офиса функционируют отделы и подразделения, специфические для данного вида организаций. Бэк-офис и фронт-офис имеют черты, схожие с большинством организаций социальной сферы обслуживания.

### 1.1.2. Краткая характеристика подразделения и его видов деятельности

Ключевым подразделением для брокерской фирмы является дилерский отдел. Дилеры занимаются покупкой и продажей ценных бумаг, зарабатывая на перепродаже или, получая проценты с приобретенных бумаг или с суммы прибыли клиента. На рынке Форекс, где торговля ведётся круглосуточно, работа дилерского отдела построена по сменному принципу. В настоящее время дилеры используют для торговли информационные системы, предоставляемые провайдерами ликвидности, а также информационно-технологические решения, позволяющие компании выступать в роли агрегатора ликвидности. Именно от качества работы дилерского отдела зависит размер получаемой компанией прибыли. Доход брокерской фирмы образуется за счёт комиссии, налагаемой на клиентскую сделку. Комиссия может взиматься явно, с каждой торговой операции или учитываться в котировках, предоставляемых фирмой своим клиентам [11].

Работа любого финансового посредника в немалой степени зависит от взаимодействия с поставщиками ликвидности и банками. В число функций, выполняемых соответствующими отделами, входят заключение выгодных контрактов с финансовыми контрагентами и разрешение спорных ситуаций. Долгосрочная прибыльная работа на финансовом рынке требует от организации наличия торговой и рискованной стратегий, построенных по принципу научности, адаптивности и достоверности. За это отвечают аналитический отдел и управление торговых клиентских операций. Взаимодействие с собственным бэк-офисом обеспечивает отдел расчётов и оформления сделок.

Подразделение валютного дилинга занимается управлением активами путем проведения текущих конверсионных и депозитно-кредитных операций на межбанковском рынке. Конверсионные операции — это сделки агентов валютного рынка по обмену оговоренных сумм денежной единицы одной страны на валюту другой страны по согласованному курсу на определенную дату. Конверсионные операции делятся на две группы: операции типа спот

или текущие конверсионные операции, форвардные конверсионные операции. Основным бизнес-процессом организации является процесс проведения клиентской торговой операции.

Помимо получаемой прибыли деятельность дилерского отдела оценивается и другими показателями [12]. Например, количество клиентов, обслуживаемых дилерами. Количественно-стоимостные оценки деятельности дилерского отдела типичного финансового посредника представлены в таблице 1.

Таблица 1.1. Количественно-стоимостные оценки деятельности дилерского отдела

№ п/п	Показатель	Единицы измерения	Средние значения
1	Количество клиентских фирм	шт.	10 – 400
2	Количество клиентских счетов	шт.	10 – 500
3	Количество учётных записей в электронной торговой системе (с учётом технических)	шт.	20 – 1500
4	Количество финансовых инструментов, доступных для клиентов	шт.	30 – 800
5	Количество ордеров, выставяемых в электронную систему от одного клиента за сутки	шт.	100 – 100 000
6	Время обработки одной заявки	мкс	50
7	Количество источников для агрегации ликвидности	шт.	10 – 50
8	Частота поступления котировок от провайдера ликвидности	шт./мин	1 000 – 5 000
9	Период обновления котировок у клиента	мкс	5000

Для удобства все клиенты группируются по принадлежности к юридическим лицам, т.к. финансовый посредник может оказывать услуги как юридическим, так и физическим лицам. Для учёта финансового результата клиентов дилеры заводят торговые счета. С одним торговым счётом может быть связано несколько учётных записей в электронной торговой системе. Учётные записи могут быть нескольких типов, например торговые, технические, сервисные, и др. Клиенты осуществляют торговлю по нескольким финансовым инструментам. Чем больше инструментов компания предоставляет своим клиентам, тем больше инвестиционных возможностей у них имеется. Особые требования предъявляются к временным и количественным показателям, связанным с производительностью дилерского отдела. Такими показателями являются: количество ордеров, выставляемых в электронную систему, частота выставления ордеров, время обработки одной заявки, а также период обновления котировок у клиента. С увеличением значений данных показателей увеличивается нагрузка на систему. С другой стороны работа дилерского отдела зависит от провайдеров ликвидности. Чем больше провайдеров агрегирует электронная система, чем объёмнее поток котировок от каждого отдельного провайдера, тем точнее котировки финансовой организации отражают рыночную ситуацию. Однако неограниченный рост нагрузки со стороны маркетмейкеров также чреват сбоями в работе электронной торговой системы. Производительность дилерских торговых систем, как правило, в разы выше клиентских. Поэтому потоки данных на клиентские устройства жёстко ограничиваются с соблюдением оптимальных показателей полноты, достоверности и достаточности рыночной информации.

### 1.1.3. Сущность задачи автоматизации нагрузочного контура для электронной торговой системы

Для уверенной и стабильной деятельности финансовой организации, оказывающей посреднические услуги на финансовых рынках, критически

важно бесперебойное функционирование электронных торговых систем, обеспечивающих исполнение клиентских заявок и предоставление требуемой рыночной информации [5]. При этом неограниченный рост основных количественных показателей работы дилерского отдела может привести к частичной или полной неспособности компании выполнять свои обязательства перед клиентами на короткие (до несколько секунд), средние (несколько минут) и длительные отрезки времени. При этом бизнес-цели финансовых игроков диктуют им необходимость расширения клиентской базы, количества поставщиков ликвидности и темпов торговли. Таким образом, нередко случается ситуация когда запросы пользователей электронной торговой системы намного превосходят возможности её экстенсивного масштабирования. Для удовлетворения растущих запросов пользователей торговая система компании должна подвергнуться оптимизации, усовершенствованию архитектуры и другим изменениям. Однако такие работы требуют от компании инвестиций в разработку и усовершенствование цифрового инструментария [6]. Для оценки необходимости таких инвестиций необходимо наличие однозначных количественно-стоимостных показателей, отражающих возможности системы.

В общем такие показатели устанавливают верхнюю границу нагрузки, при которой торговая система имеет возможность выполнять свои функции штатно. Наиболее надёжным способом определения граничных значений, предоставляющим данные, характеризующие конкретную информационную систему, является запуск системы в тестовой нагрузочной среде. Тестовая среда должна обеспечивать быструю и гибкую настройку ключевых параметров, влияющих на производительность системы. В число таких параметров входят количество пользователей системы, частота выставления заявок пользователями, количество провайдеров ликвидности, частота обновления котировок от провайдеров. Также имеется ряд дополнительных параметров, которые не оказывают прямого воздействия на

производительность системы, однако могут являться причиной появления различных сторонних эффектов. Например, свойства пользователя, провайдера ликвидности, влияющие на процесс исполнения заявок и поток котировок.

Таким образом, задача автоматизации сводится к поиску наборов значений ключевых показателей, оказывающих влияние на производительность системы и стремящихся к максимуму, при которых система работает штатно. Так как построение математической модели динамично меняющейся сложной компьютерной системы дорогостоящее и ресурсоёмкое занятие, то подобные задачи оптимизации решаются итерационными методами. Сотрудники, отвечающие за функционирование информационной системы, производят серию тестовых запусков с различными параметрами. По результатам работы системы в тестовой среде производится анализ её состояния. Ключевыми характеристиками, определяющими состояние системы в конкретный момент или период времени являются время обработки одного сообщения в одном из компонентов системы, размер очереди сообщений на входе в одном из компонентов системы, время ожидания обработки сообщения.

Особенности реализации данной задачи для дилерского отдела компании NTFX заключены в наборе используемых информационно-технических средств организации торговой деятельности. Для агрегации ликвидности от провайдеров и предоставления её клиентам используется торговая платформа NTPro. В данной платформе изначально не предусмотрена возможность запуска в тестовой среде, поэтому каждый такой запуск требует длительной настройки входных данных, конфигурирования, выполняемых в ручном режиме [14].

#### Выводы по параграфу

В динамично развивающемся мире финансовых информационных систем предъявляются жёсткие требования к надёжности, стабильности и

производительности программного обеспечения. Поэтому разработчикам, пользователям, администраторам таких систем критически важно понимать их возможности. Такое понимание даёт гарантию бесперебойного функционирования программных продуктов, достигаемое за счёт административного ограничения нагрузки и заблаговременного планирования инвестиций в увеличение производительности используемых средств.

## **1.2. Концептуальное моделирование деятельности электронной торговой системы**

Технология проектирования информационных систем подразумевает создание сначала модели AS–IS, ее анализ и улучшение бизнес-процессов, то есть создание модели TO–BE. На основе последней строится модель данных, прототип и окончательный вариант информационной системы. Модель деятельности (AS–IS), представляющая собой «снимок» положения дел в организации на момент обследования и позволяющая понять, что делает и как функционирует организация с позиций системного анализа, а также на основании автоматической верификации выявить ряд ошибок и узких мест и сформулировать предложения по улучшению ситуации. Для создания упомянутых моделей необходимо выполнить консалтинг деятельности исследуемой организации [7].

При осуществлении клиентской операции через дилерский отдел проходят два информационных потока: поток котировок и поток заявок. К работе с данными информационными потоками предъявляются различные требования, вытекающие из их природы.

Котировка — это цена (курс, процентная ставка) товара, которую объявляет продавец или покупатель и по которой они готовы совершить покупку или продажу (предлагается оферта). Обычно подразумевается относительно быстро меняющаяся цена, например биржевая.

Иногда котировкой считается цена только заключенной сделки, а не цена оферты покупки или продажи. Как правило, это справедливо для потока котировок, получаемого от поставщика ликвидности. После получения котировок от провайдеров дилерский отдел обрабатывает их по определённому алгоритму, формируя клиентский ценовой стакан. Наиболее частыми операциями являются наложение комиссии на каждую котировку, сужение или расширение спреда (разницы между курсом покупки и продажи), фильтрация и агрегирование котировок.

Рассмотрим подробнее процесс фильтрации котировок. Каждый источник цен имеет свои характеристики: скорость исполнения заявки, максимальный объём заявки, минимальный интервал времени между заявками, различные характеристики канала коммуникации до торговой площадки. Кроме того, дилерский отдел может вести статистику исполнения ордеров, главным показателем которой является процент отклонённых площадкой заявок от общего числа отправленных на неё ордеров. Источники с наихудшими показателями исполнения могут исключаться из стакана или использоваться в последнюю очередь при отсутствии источников с прочими равными показателями. Кроме того, поток котировок будет исключаться из рассмотрения при временном изменении площадкой торговых условий, например, отличии дат валютирования инструментов в связи с государственными праздниками в стране, резидентом которой является биржа.

Агрегирование котировок заключается в формировании ценового стакана, предоставляемого клиентам. Как правило, котировки, отражающие реальную торговлю на бирже, представляются в виде ордерного стакана, а котировки, представляющие собой оферту контрагента – в виде бендового (от англ. band – ансамбль). Характеристиками ордерной котировки является цена и объём. При этом в стакане может быть несколько котировок с одинаковыми ценами и объёмами. Характеристикой бендовой котировки являются цена, минимальный и максимальный объёмы. Бенды,

составляющие стакан не могут пересекаться или иметь разрывы. Минимальный объём первого бэнда должен быть нулевым. Бэндовая котировка формируется по определённому алгоритму. Наиболее часто применяются следующие: лучшая цена, худшая цена или VWAP – средневзвешенные цены. Клиентский стакан является бэндовым.

Для формирования итогового клиентского стакана агрегированный стакан подвергают дальнейшей обработке, накладывая комиссии, сужая или расширяя спрэд.

На рисунке 1.2 представлена функциональная IDEF0 - диаграмма «Как есть» для бизнес-процесса формирования клиентского торгового стакана.

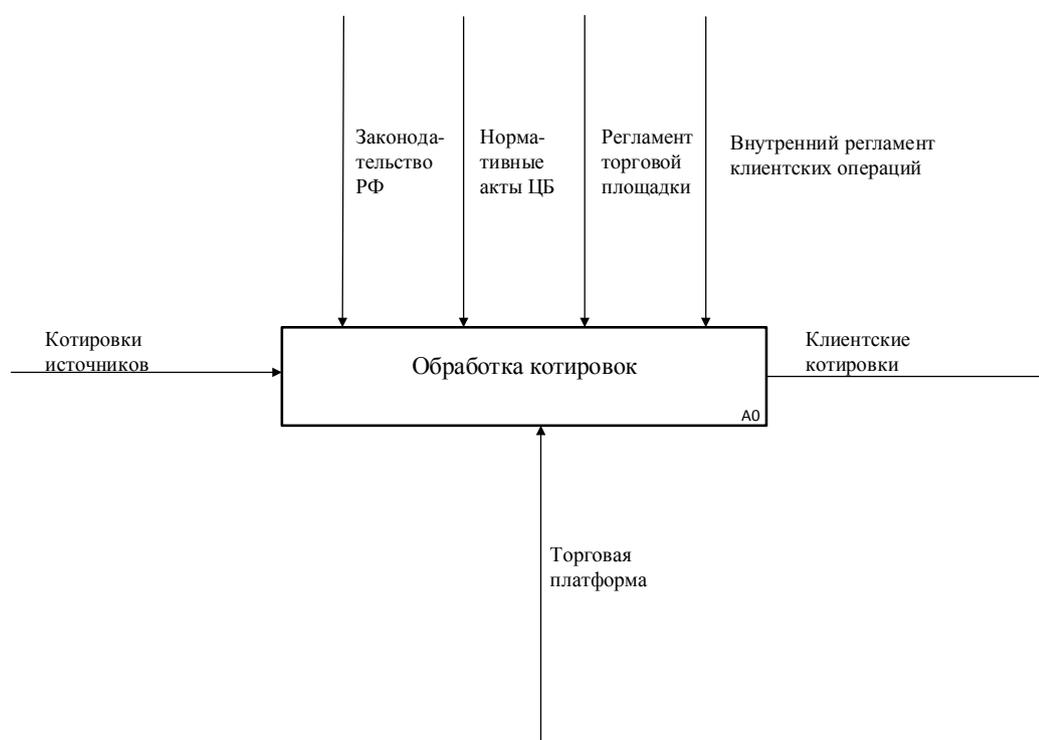


Рис. 1.2. – Обработка котировок

Обработка котировок представляет собой достаточно трудоёмкий процесс. Кроме того, частота обновления данных может измеряться микросекундами. Поэтому для выполнения данной операции применяются средства автоматизации. Распространённой схемой является подключение биржевой ликвидности к специализированной клиентской торговой системе, такой как Quick [15] или MetaTrader [16].

Обработка котировок должна соответствовать законодательству РФ, а также внутреннему и клиентскому регламентам. Так, например, клиент брокера должен быть осведомлён о правилах предоставления ликвидности, применяемых комиссиях, а также об основных характеристиках ценового потока, таких как размер спреда, тип спреда (фиксированный или плавающий).

На рисунке 1.3 представлена DFD-декомпозиция процесса обработки котировок.

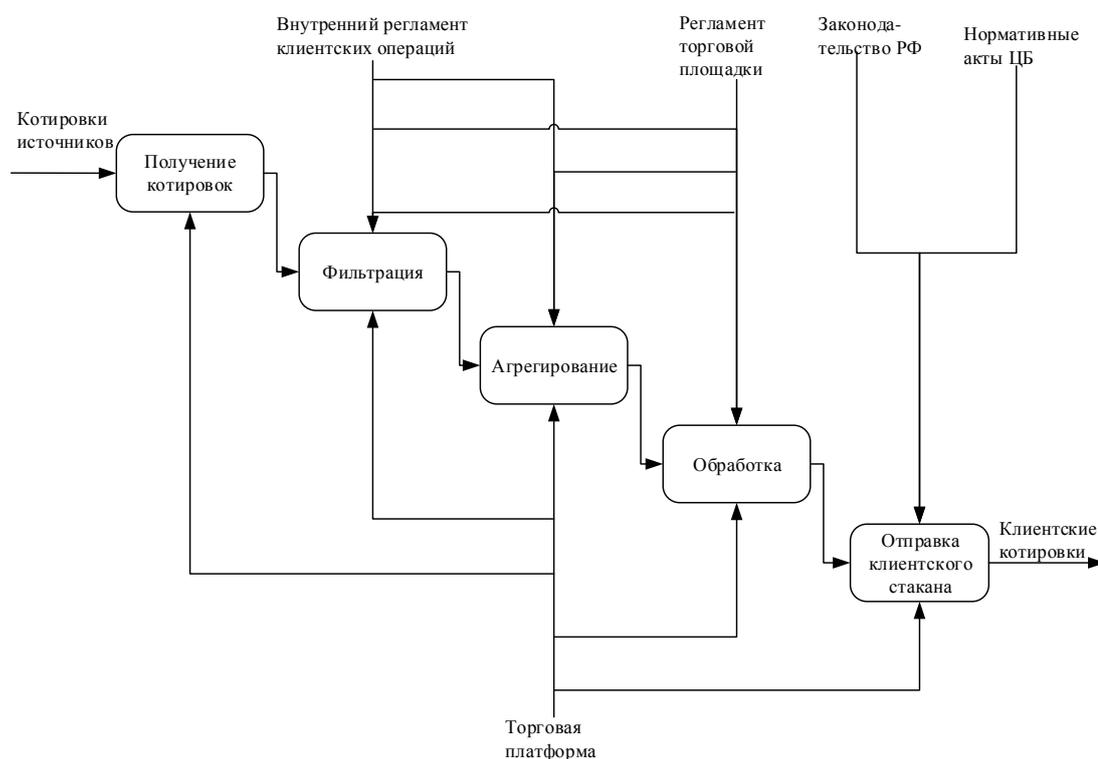


Рис. 1.3 – Формирование клиентского торгового стакана

Рассмотрим процесс обработки информационного потока клиентских заявок. При этом данное исследование не ставит целью описать все возможные случаи или обстоятельства, которые могут относиться к определенному ордеру, порученному компании. Клиентская заявка на выполнение торговой операции (иначе торговый ордер) представляет собой информационное сообщение, передаваемой клиентом компании NTFX по информационно-коммуникационному каналу. В настоящее время общепринятым является использование сети Ethernet в качестве канала

передачи информации. В отличие от потока котировок поток клиентских заявок должен передаваться с использованием защищённого протокола, в качестве которого часто используется SSL. В области электронной торговли стандартным протоколом прикладного уровня для передачи торговых сообщений является протокол FIX. Он может использоваться торговыми роботами напрямую, а также клиентскими торговыми системами.

Торговая заявка должна содержать следующую обязательную информацию: торговый инструмент, цена, объём, сторона сделки, тип заявки. В качестве торгового инструмента может использоваться валютная пара (например, USDRUB, EURUSD), товары (золото, серебро), а также крипто-инструменты (например, Bitcoin). Под стороной сделки понимается покупка или продажа. Тип ордера характеризует режим его исполнения. Рыночный ордер подразумевает исполнение по цене рынка. Лимитированная заявка обеспечивает исполнение не хуже указанной цены. Рыночная лимитированная заявка обеспечивает исполнение по рыночной цене, но не хуже указанной в заявке. Ордер Stop Loss обеспечивает выставление рыночной лимитированной заявки, если цена рынка стала хуже указанной. Указанные ордера могут иметь отметку о времени жизни: GTC или IOC. IOC-ордер немедленно отменяется, если не может быть исполнен в момент выставления. GTC ордер остаётся активным до тех пор, пока не будет исполнен или отменён. Если ордер содержит пометку FOK, то он может быть исполнен только на весь указанный объём. В противном случае допускается частичное исполнение [17].

Цена клиентской сделки складывается из двух составляющих: цены сделки перекрытия и издержек, понесённых брокером в процессе перекрытия. Цена меняется в зависимости от рыночной ликвидности и правил площадки. Издержки состоят из операционных издержек, комиссий, налогов и сборов, выплачиваемых клиентом третьим лицам, а также площадке. Помимо цены характеристиками исполнения заявки, по которым оценивается качество работы дилерского отдела, являются объём

совершенных сделок, скорость исполнения, вероятность исполнения, а также надёжность расчётов. Под скоростью исполнения заявки понимается время с момента получения брокерской компанией клиентского ордера до момента отправки ордера перекрытия или до принятия решения о совершении сделки без перекрытия.

Выделяют три основные стратегии перекрытия клиентских ордеров. Стратегия Back-to-back является наиболее безопасной с точки зрения рисков, принимаемых на себя дилером. При получении клиентской заявки формируется ордер перекрытия на целевую площадку. После получения ответа от целевой площадки отправляется соответствующий ответ клиенту. Если сделка перекрытия завершилась успешно, формируется клиентская сделка на перекрытый объём. При формировании заявки перекрытия учитываются разнообразные настройки исполнения, такие как клиентская комиссия, комиссия источника ликвидности, минимальная прибыль банка при перекрытии, проскальзывание исполнения и другие. Стратегия Position Keeping предполагает немедленное формирование клиентской сделки при получении заявки. При этом финансовый результат операции относится на позицию брокера для отложенного перекрытия в будущем. Стратегия Crazy предполагает немедленное формирование клиентской сделки при получении заявки. После этого происходит процесс перекрытия клиентской сделки на целевой площадке. Неперекрытый объём относится на позицию брокера.

Получение клиентских котировок может осуществляться как с одного источника, так и с нескольких, при этом формируется агрегированный ценовой стакан. Перекрытие клиентских заявок в этом случае также происходит на нескольких площадках.

На рисунке 1.4 представлена функциональная IDEF0 - диаграмма «Как есть» для бизнес-процесса совершения клиентской торговой операции.



Рис. 1.4. – Клиентская торговая операция

Помимо законодательства РФ и внутреннего регламента работа брокера регулируется несколькими документами и соглашениями, обозначенными на схеме наименованием «Регламент торговой площадки». Основным биржевым регулятором является Комиссия по ценным бумагам и биржам. Брокеру, имеющему дело с биржами нескольких стран, приходится удовлетворять требованиям соответствующего регулятора. Также значимыми документами, регулирующими деятельность международного финансового брокера являются Директива о рынках финансовых инструментов (MiFID), Foreign Account Tax Compliance Act («FATCA») и Единый стандарт по обмену налоговой информацией, разработанный в рамках Организации экономического сотрудничества и развития (ОЭСР).

На рисунке 1.5 представлена DFD-декомпозиция процесса совершения клиентской торговой операции.

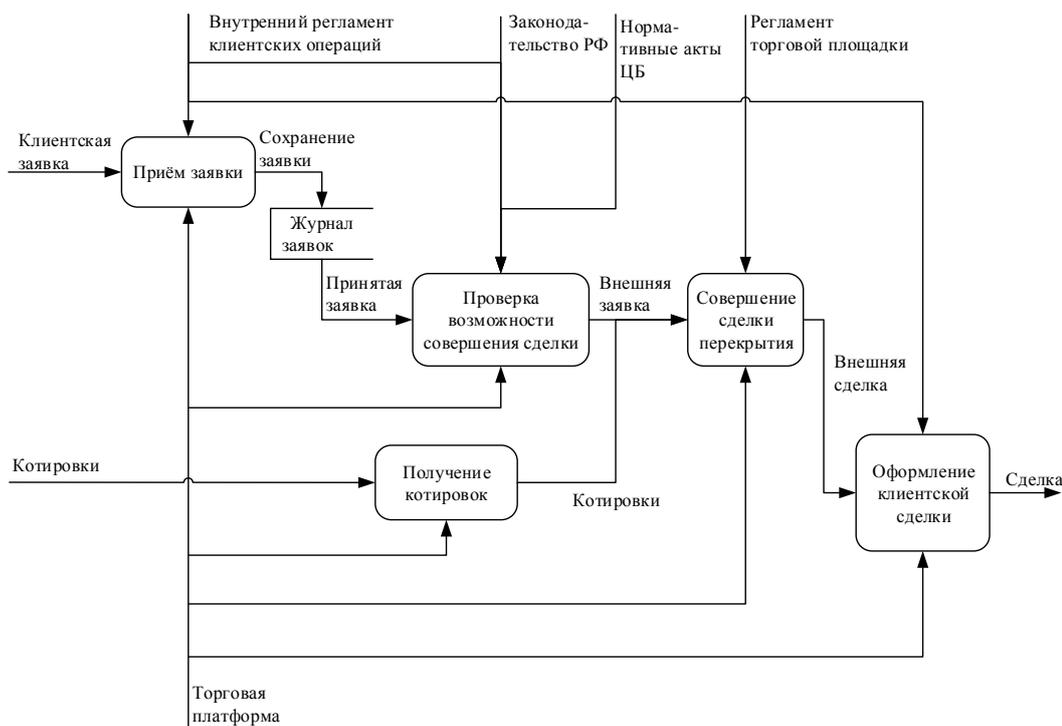


Рис. 1.5. – Перекрытие клиентской торговой операции

Исполненная клиентская заявка является сделкой или торговым контрактом. В соответствии с законодательством компания-брокер должна вести и предоставлять клиентам и контролирующим органам соответствующую отчётность. Данные функции выполняет подсистема учёта сделок, принимающая информацию по протоколу FIX.

Процесс нагрузочного тестирования торговой платформы, функционирующей в дилерском отделе, выполняется с помощью технических специалистов: программистов и инженеров сопровождения. Их обязанности в рамках данного процесса состоят в выборе методики тестирования, подготовке исходных данных, настройке системы, запуске тестового контура, анализа результатов работы сконфигурированной системы на подготовленных данных. Результатом нагрузочного тестирования являются наборы граничных значений основных параметров, влияющих на производительность системы. На рисунке 1.6 представлена функциональная IDEF0-диаграмма «Как есть» для бизнес-процесса нагрузочного тестирования.



Рис. 1.6. – Нагрузочное тестирование

Программно-техническими средствами осуществления нагрузочного тестирования системы являются торговая платформа NTPro и тестовая среда. Платформа NTPro является распределённой серверной системой построенной по принципу компонентной архитектуры. В настоящее время компания располагает несколькими контурами для запуска торговой системы. Основной production-контур или prod используется для обработки реальной торговой информации и подключения клиентов компании для осуществления торговой деятельности. Контур пользовательского тестирования (uat) необходим для предоставления потенциальным и действующим клиентам компании демонстрационного доступа к системе. Компания также располагает двумя тестовыми контурами (dev и test), используемыми с целью отладки программного обеспечения и тестирования. Prod- и uat-контур имеют различную конфигурацию. Prod-конфигурация включает три сервера. Для экономии средств uat-контур состоит из двух серверов, располагающихся в основной локации компании. Процедура нагрузочного

тестирования выполняется на контурах dev и test, соответствующих prod и uat средам соответственно.

Подготовка к нагрузочному тестированию в настоящий момент представляет собой трудоёмкую операцию. При этом из-за особенностей торговой платформы для получения исчерпывающей картины требуется многократное развёртывание тестового контура с нуля. Для получения полного набора граничных значений тестирование производится многократно. На рисунке 1.7 представлена DFD-декомпозиция одной итерации процесса нагрузочного тестирования.

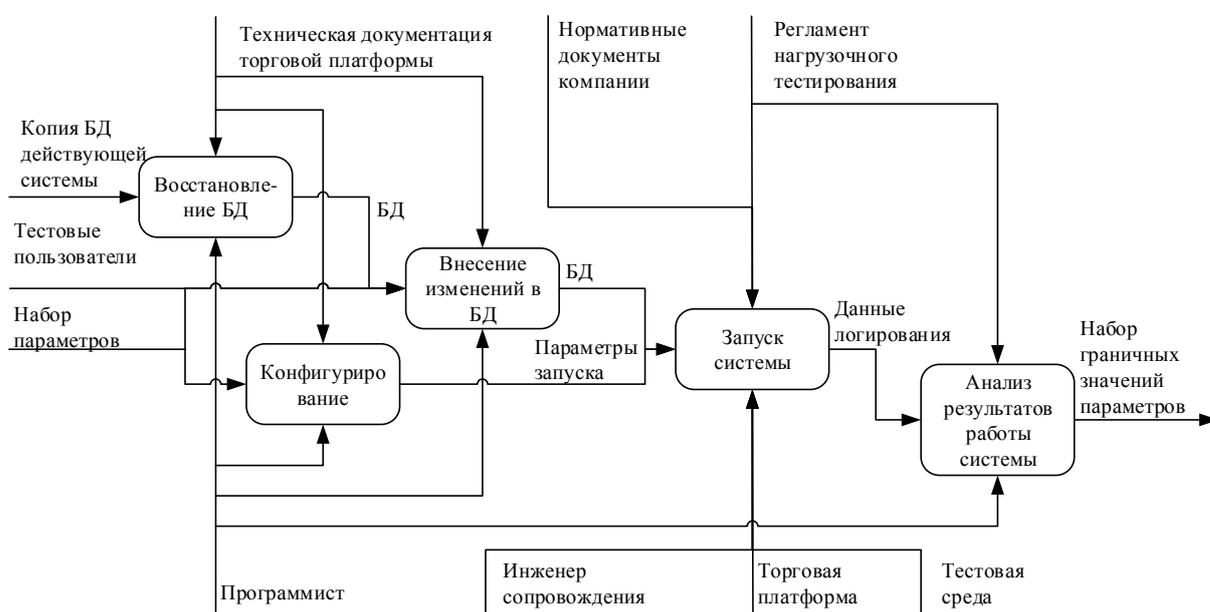


Рис. 1.7. – Типичная итерация процесса нагрузочного тестирования

Перед запуском очередной итерации нагрузочного тестирования программист убеждается в том, что в тестовую базу данных могут быть внесены изменения, требующиеся на данном шаге. Если трудоёмкость внесения таких изменений превышает трудоёмкость восстановления базы данных из копии, то производится восстановление базы с нуля. В соответствии с документацией на торговую платформу программист вносит изменения в БД, касающиеся количества и типа активных клиентов, количества и типа провайдеров ликвидности. После этого производится

конфигурирование системы для запуска на тестовом окружении. После автоматизированного стандартного конфигурирования программист осуществляет ручную настройку системы. В конфигурационных файлах торговой платформы задаётся информация, представляющая параметры провайдеров ликвидности, а также параметры сетевого обмена информацией между системой и клиентами компании. После того, как тестовая среда и торговая платформа будут сконфигурированы для текущей итерации нагрузочного тестирования, инженер сопровождения запускает систему в соответствии с регламентом нагрузочного тестирования. При этом доступ к тестовой среде определяется нормативными документами компании.

Торговая платформа во время своего функционирования генерирует информацию о происходящих в ней событиях. Данная информация сохраняется в файлах регистрации. Протоколируемая информация имеет уровень (Trace, Debug, Info, Warning, Error) и может быть нескольких типов. Для важных событий, происходящих в среднем не чаще нескольких десятков миллисекунд, устанавливается однозначное соответствие в файле регистрации в виде записи, содержащей достаточную информацию о событии. Существующая система логирования не позволяет задать такое соответствие для событий, происходящих чаще одной миллисекунды. Такая информация записывается в файл регистрации в статистическом агрегированном виде. В таком виде представлена информация о количестве сообщений в очередях компонентов, а также временных характеристиках процесса обработки сообщений.

Для сбора достаточного количества данных достаточно, чтобы система работала под нагрузкой в течение 10 минут. Программист анализирует полученную информацию на предмет наличия отклонений в показателях работы системы. Устойчивое ухудшение показателей в течение нескольких минут свидетельствует о критическом характере нагрузки. Если показатели находятся в норме на всём интервале тестирования, то система успешно справляется с нагрузкой. Чередование положительной и отрицательной

динамики показателей функционирования системы свидетельствует о пограничном характере нагрузки. Результатом тестирования являются наборы максимальных параметров, влияющих на производительность системы, при которых показатели производительности остаются в норме.

Стоит отметить, что из-за отсутствия достаточного уровня автоматизации процесс нагрузочного тестирования является трудоёмким, а результаты не являются полными. Как правило, большая часть показателей принимается константными, а динамически меняется только один или два, например частота котирования провайдерами ликвидности при постоянном количестве провайдеров и клиентов. В результате повышается риск отказа системы, вызванный ростом нагрузки.

### **1.3. Постановка задачи на разработку автоматизированного нагрузочного контура для электронной торговой системы**

1.3.1. Цель и назначение автоматизации нагрузочного контура для электронной торговой системы

Одной из целей разработки автоматизированного нагрузочного контура для электронной торговой системы является снижение трудоёмкости проведения нагрузочного тестирования. В рамках данной работы не ставится цель добиться автоматического выполнения нагрузочного тестирования в целом. Основной упор сделан на совершенствовании проведения одной итерации тестирования на конкретном наборе параметров. Реализация данной задачи является первым шагом на пути к автоматизации всего цикла нагрузочного тестирования. Уменьшение времени, необходимого на проведение одной итерации нагрузочного тестирования, должно происходить за счёт сокращения временных затрат на развёртывание нагрузочного контура, а также на анализ показателей производительности системы. Отдельной задачей в рамках реализации работы является снижение трудоёмкости данного процесса. Это будет достигаться за счёт исключения

участия программиста на подготовительном и заключительном этапах итерации.

Целью решения задачи также является увеличение степени полноты и достоверности информации, характеризующей предельно допустимую нагрузку на электронную торговую систему. Возможность достижения данной цели вытекает главным образом из двух предыдущих, поскольку при уменьшении временных и трудовых затрат становится возможным формирование более полного набора входных параметров [8].

Назначением реализации работы «Разработка автоматизированного нагрузочного контура для электронной торговой системы» может служить:

- 1) автоматизация подготовки базы данных системы к нагрузочному тестированию;
- 2) автоматизация конфигурирования системы для нагрузочного тестирования;
- 3) автоматизация процесса расчёта настроечных значений системы на основе параметров, влияющих на производительность системы.
- 4) автоматизация анализа информации о состоянии системы.

### 1.3.2. Общая характеристика организации решения задачи автоматизации нагрузочного контура для электронной торговой системы на ЭВМ

В ходе решения данной задачи основные функции подразделения не должны претерпеть существенных изменений. Вместе с тем изменится процедура поддержки информационной торговой системы. Наличие в коде модулей, отвечающих за нагрузочное тестирование, требует поддерживать их работоспособность при внесении изменений в программный продукт. Также по результатам реализации задачи необходимо внести изменения в регламент проведения нагрузочного тестирования.

Источниками поступления входной информации является анализ функционирования текущей версии торговой платформы и результаты

предыдущих циклов нагрузочного тестирования. На основе анализа функционирования текущей конфигурации системы можно сделать вывод о том, являются ли допустимыми текущие параметры, оказывающие влияние на её производительность. Если текущие параметры являются допустимыми, то их можно принять за минимальные значения в наборе входных параметров для нагрузочного тестирования. Результаты предыдущих циклов тестирования необходимы для сравнения. На основе динамического ряда показателей производительности можно сделать вывод о влиянии доработок, вносимых в систему.

Задача должна быть реализована в несколько этапов. На первом этапе необходимо добиться автоматического развёртывания конфигурации для нагрузочного тестирования на основе текущих параметров. При этом такая конфигурация должна иметь возможность функционирования на всех тестовых контурах компании. Нужно предусмотреть возможность подачи нагрузочных данных как на всю систему в целом, так и на отдельные её части. На втором этапе необходимо реализовать возможность задания любого набора параметров, влияющих на производительность системы. На данном этапе в набор должны входить количество и тип провайдеров ликвидности, количество и тип клиентов, частота обновления рыночной информации, частота выставления клиентами заявок. На третьем этапе необходимо реализовать средства автоматизированного анализа информации о состоянии системы. На выходе программного продукта должна быть информация, позволяющая без дополнительных расчётов сделать вывод о состоянии системы в период нагрузки. В перечень выходной информации должны войти минимальное, максимальное и среднее нахождение сообщения определённого типа в очереди каждого компонента, минимальный, максимальный и средний размер очереди компонентов. Данные должны выводиться в виде таблицы, при этом список компонентов должен быть отсортирован в порядке убывания по среднему значению показателя. Если максимальное значение показателя не превышает регламентное значение, то

система работает штатно. Если максимальное значение показателя превышает регламентное значение, а среднее близко к норме, то система при данных параметрах работает на грани своих возможностей. Если среднее значение показателя превышает нормативное, то при данных параметрах система работать не может.

В рамках реализации указанных трёх этапов будут достигнуты минимальные цели данной работы. После чего можно переходить к дальнейшему усовершенствованию процесса нагрузочного тестирования.

Перечень входной информации вводимой вручную должен быть сведён к минимуму. Количество провайдеров ликвидности и их тип должен задаваться в отдельном конфигурационном файле, представленном на рисунке 1.8.

```
1 # Add specific quotes type, location and principal connectors
2 AlfnPrincipal.Moscow.Order=5
3 AlfnPrincipal.Moscow.Band=4
4
5 # Add connectors of each quotes type for specific principal and location
6 AlfnPrincipal.London.*=3
7
8 # Add connectors for each location of each quotes type for specific principal
9 NtfxPrincipal.*.*=2
10
11 # Add connectors for each principal, location and quotes type
12 *.*.*=1
```

Рис. 1.8. – Пример конфигурационного файла добавления провайдеров ликвидности.

Частота обновления рыночной информации задаётся в основном конфигурационном файле процесса, эмулирующего нагрузку от провайдеров ликвидности. Количество клиентов и их тип также должно задаваться через конфигурационный файл. Частота обновления рыночной информации должна указываться в основном конфигурационном файле процесса, обеспечивающего нагрузку от клиентов.

Задача должна решаться при каждом внесении существенных изменений в код и параметры информационной торговой платформы, а также при необходимости получения результирующей информации, а основе нового набора входных данных.

### 1.3.3. Формализация расчётов

Характеристика входной информации представлена в таблице 1.2.

Таблица 1.2. – Входная информация

№	Наименование	Единицы измерения	Диапазон	Источник
1.	Количество провайдеров ликвидности	шт.	1 – 1 000	БД, конфигурационный файл
2.	Количество клиентов	шт.	1 – 10 000	БД, конфигурационный файл
3.	Частота обновления рыночной информации	мс	1 – 1 000	Конфигурационный файл
4.	Частота выставления заявок	мс	1 – 1 000	Конфигурационный файл

Под частотой обновления рыночной информации понимается периодичность отправки сообщения типа MarketData по Fix-протоколу. Сообщение относится к определённому финансовому инструменту. Под частотой выставления заявок понимается периодичность отправки в систему сообщения типа OrderRequest по протоколу Fix.

Характеристика входной информации представлена в таблице 1.3.

Таблица 1.3. – Выходная информация

№	Наименование	Единицы измерения	Диапазон	Источник
1.	Количество сообщений в очереди компонента	шт.	0 – $10^9$	Файл регистрации системных событий
2.	Время обработки сообщения	нс	1 - $10^{12}$	Файл регистрации системных событий
3.	Отсортированный список компонентов	список	-	Файл регистрации системных событий

Стоит отметить, что разрабатываемый нагрузочный контур будет специфичным для конкретной торговой платформы. Контур включает в себя не только эмулятор провайдеров ликвидности и клиентских торговых сессий, но и саму платформу. Эмуляторы имеют слабую зависимость от самой платформы, т.к. не подключаются к БД и обмениваются информацией по стандартному протоколу Fix. Вместе с этим они настроены на конкретную спецификацию данного протокола. Скрипт настройки и конфигурирования самой системы для нагрузочного тестирования является частным решением задачи для конкретной торговой платформы.

#### Выводы по параграфу

На основе анализа предметной области и существующей торговой платформы можно утверждать, что целью реализации задачи является сокращение трудовых и временных затрат сотрудников компании на получение граничных значений ключевых параметров торговой системы. Решение задачи позволяет более уверенно внедрять в существующую систему новые разработки, гарантируя компании стабильное и безотказное выполнение своих обязательств. Задача нагрузочного тестирования актуальна для большинства сложных высоконагруженных систем, однако реализацию нагрузочного контура для конкретного программного продукта трудно применять для тестирования других систем.

#### **1.4. Анализ существующих разработок и обоснование выбора технологии проектирования**

Существующий в компании вариант решения задачи предусматривает использование в качестве объекта нагрузочного тестирования торговую платформу NTPro. Она поставляется вместе с оптимизированной инфраструктурой и может быть подвергнута процедуре гибкой настройки по нужды конкретного брокера. Платформа построена по модульному

принципу. Помимо ядра системы, отвечающего за базовые функции и рассылку настроек, к системе могут быть добавлены модули агрегации и распределения, срочного и денежного рынка, дизайнер стратегий, аналитики и статистики.

Развёртывание системы NTPro в различных конфигурациях осуществляется с помощью программ, реализованных на скриптовых языках python и bash. При выполнении данной процедуры можно указать несколько параметров, например, контур и набор сервисов.

После реализации разрабатываемого решения появится возможность указать для программы установки дополнительный параметр «нагрузочное тестирование», после чего система должна быть сконфигурирована соответствующим образом.

В целом создание тестовых и нагрузочных стендов для отладки и исследования сложных изделий является обычной практикой разработки. Однако при решении подобного типа задач необходимо учитывать индивидуальные особенности конкретного программно-аппаратного продукта.

В ходе реализации задачи необходимо разработать программу-эмулятор. Она должна быть построена на основе компонентной архитектуры. Связь между компонентами должна осуществляться с помощью API. К программе-эмулятору предъявляются повышенные требования в плане производительности и устойчивости работы. Она должна обеспечивать нагрузку, превышающую возможности существующей системы. Для достижения указанных целей применяют современные объектно-ориентированные языки программирования с возможностью компиляции кода, например C++ или C#. В данных языках оптимальным образом сочетается гибкость разработки, производительность продукта и удобство реализации.

Для реализации процедуры настройки среды для нагрузочного тестирования необходимо реализовать скриптовую программу,

выполняющую анализ текущей конфигурации, и способную внести изменения, соответствующие входным параметрам.

Для выполнения автоматизированного анализа результатов тестирования также необходимо реализовать программу на одном из скриптовых языков, обеспечивающую чтение файлов регистрации системы, их анализ, а также предоставление выходной информации в удобном для восприятия виде.

### **Выводы по первой главе**

Существующая в компании процедура нагрузочного тестирования не является оптимальной, так как выходная информация, получаемая в результате выполнения данной процедуры, не удовлетворяет требованиям полноты, достоверности и оперативности. Для устранения этих недостатков, а также уменьшения трудовых и временных затрат сотрудников компании, необходимо разработать и реализовать программные средства развёртывания и функционирования нагрузочного контура электронной торговой платформы. Кроме того, требуется усовершенствовать процедуру сбора и обработки информации о состоянии системы, представив её в аналитическом, удобном для восприятия виде.

Анализ существующих разработок показывает, что данная задача достаточно часто решается в подобных целях для различных программно-аппаратных продуктов. Однако каждый раз при реализации тестовых контуров приходится учитывать индивидуальные особенности конкретной среды и конкретного программного изделия.

## **ГЛАВА 2. РАЗРАБОТКА И РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОГО НАГРУЗОЧНОГО КОНТУРА ДЛЯ ЭЛЕКТРОННОЙ ТОРГОВОЙ СИСТЕМЫ**

### **2.1. Логическое моделирование автоматизированного нагрузочного контура для электронной торговой системы**

#### 2.1.1. Логическая модель автоматизированного нагрузочного контура для электронной торговой системы и её описание

Как было установлено в данной работе, процесс нагрузочного тестирования торговой платформы содержит в себе ряд недостатков, в числе которых необходимость осуществления ручной настройки параметров системы, участие в процессе тестирования разработчиков информационной системы. Трудоёмкость развёртывания нагрузочного контура и сложность контроля процесса и результатов тестирования приводят к тому, что получаемые данные не отвечают требованиям полноты и достоверности. Следовательно, процесс нагрузочного тестирования нуждается в реинжиниринге. После анализа и концептуального моделирования предметной области, выявления недостатков в существующих процессах и постановки задачи перейдём к логическому и физическому моделированию информационной системы. Модель автоматизации (ТО–ВЕ) интегрирует перспективные предложения руководства и сотрудников организации, экспертов и системных аналитиков и позволяет сформировать видение новой (автоматизированной) системы, а именно: что вновь создаваемая система будет делать и как она будет функционировать [9].

Таким образом, решением перечисленных проблем могло бы стать внедрение автоматизированного нагрузочного контура. На рисунке 2.1 представлена функциональная IDEF0 - диаграмма «Как должно быть» для процесса нагрузочного тестирования.



Рис. 2.1 – Модель ТО-ВЕ процесса «Нагрузочное тестирование»

Автоматизированный тестовый контур замыкает на себя процессы генерации тестовых данных, подготовки системы и среды к тестированию. Внедрение такого решения позволяет исключить программистов из участия в данном процессе. Роль инженера сопровождения сводится к запуску тестирования согласно с установленным регламентом, контролю за ходом тестирования и последующему анализу выходных данных. Набор настроек нагрузочного контура должен быть минимизирован и соответствовать входным данным задачи. Реализация такого средства представляет собой сложную задачу, поэтому целью данной работы является проектирование и внедрение автоматизированного решения, устраняющего недостатки, свойственные одной итерации нагрузочного тестирования. Достижение этих целей является первым шагом на пути автоматизации всей системы тестирования программного обеспечения компании.

На рисунке 2.2 представлена модель ТО-ВЕ DFD-декомпозиции типичной итерации нагрузочного тестирования 1-го уровня.

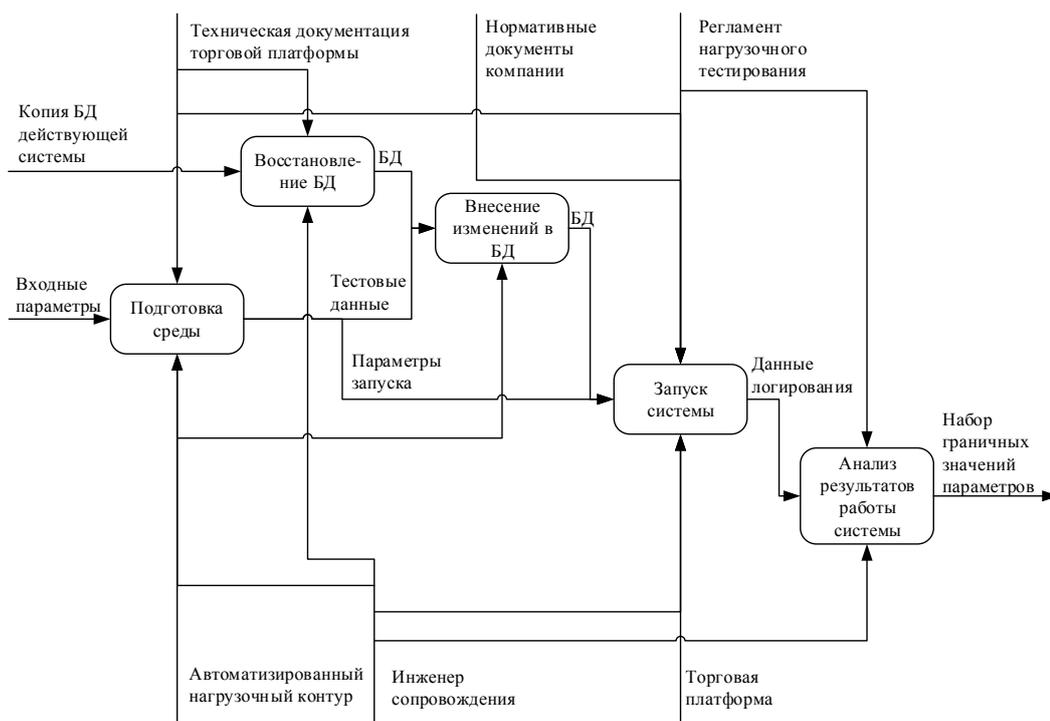


Рис. 2.2. – Декомпозиция модели ТО-ВЕ БП «Итерация нагрузочного тестирования»

В результате реализации первого этапа работ по автоматизации нагрузочного тестирования подготовка среды будет производиться с помощью процедур автоматизированного нагрузочного контура. Инициирование запуска процесса подготовки должно производиться инженером сопровождения. Инженер сопровождения также отвечает за развёртывание копии базы данных. В результате автоматизированной подготовки тестовой среды производится генерация конфигурационных файлов системы, а также внесение требуемых изменений в базу данных. После завершения процедуры настройки среды система полностью готова к запуску в режиме нагрузочного тестирования. Инженер сопровождения осуществляет стандартный запуск торговой платформы в соответствии с регламентом компании и технической документацией информационной системы. Работа системы завершается согласно регламенту нагрузочного тестирования. По завершении работы системы производится анализ данных

логирования, в результате чего делается вывод о допустимости текущей нагрузки для электронной системы торгов.

Рассмотрим функциональный аспект логической модели системы. Диаграмма вариантов использования автоматизированного нагрузочного контура представлена на рисунке 2.3.

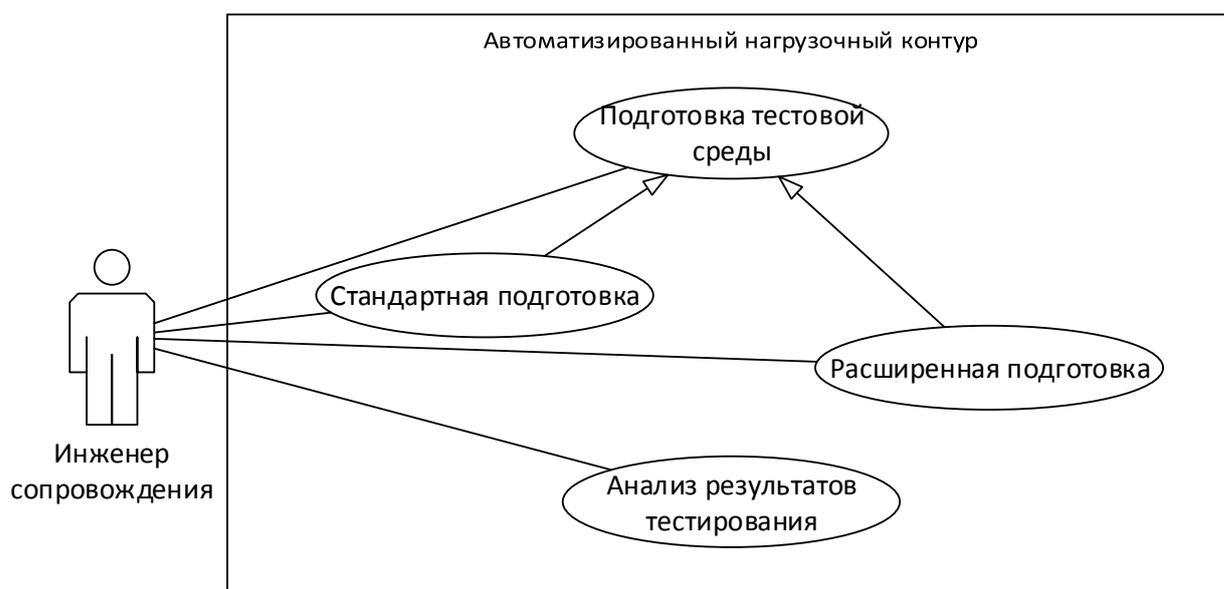


Рис. 2.3. – Диаграмма вариантов использования нагрузочного контура

Основным пользователем автоматизированного нагрузочного контура является инженер сопровождения электронной системы торгов. Взаимодействие проектируемой системы с другими внешними системами не планируется. В рамках проектируемого решения пользователи могут выполнять два вида действий: осуществлять подготовку системы к тестовому запуску и анализировать результаты такого запуска. При этом подготовка может быть стандартной и расширенной. Последний вариант предусматривает автоматическое применение к существующей среде необходимых изменений в соответствии с входными параметрами текущего запуска. Процедура стандартной подготовки предусматривает внесение минимально возможных изменений для обеспечения как можно большего соответствия текущей конфигурации системы. Вариант расширенной подготовки позволяет экономить трудовые и временные ресурсы при настройке системы в соответствии с требуемыми параметрами, однако имеет

ограниченную степень гибкости и расширяемости. Поэтому вариант стандартной настройки необходим для компенсации этих недостатков.

Рассмотрим элементный аспект логической модели системы. Стоит отметить, что проектируемое решение тесно связано с существующей информационной торговой системой, поэтому для эмуляции нагрузки от пользователей и провайдеров необходимо реализовать тестовую подсистему, содержащую в своём ядре обобщённые элементы объекта тестирования. Дилеры осуществляют управление клиентским котированием и исполнением с помощью настройки правил. Данные правила представлены классами `QuoterGroup` и `ExecutorGroup` соответственно.

Клиенты финансовой организации представлены в системе с помощью класса `Login`, в котором есть информация о торговом счёте клиента. Торговый счёт представлен классом `Account`. Для аккаунта задаются правила, по которым для него будут формироваться клиентские котировки и происходить исполнение клиентских ордеров. Ключевыми компонентами системы являются `ClientQuoter` и `ClientExecutor`, отвечающие за формирование котировок и исполнение ордеров соответственно.

Котировка представлена классом `Quote`. Для компонента `ClientQuoter` она может быть входящей и исходящей. Объекты класса `Order` также представляют собой входную и выходную информацию компонента `ClientExecutor`. Источниками котировок для ядра системы являются провайдеры ликвидности, которые также выступают площадками перекрытия клиентских заявок. Источниками заявок являются клиентские терминалы, клиентские приложения `Fix-сервера`, а также мобильные приложения. Каждый ордер выставляется в систему от имени какого-либо клиента, поэтому класс `Order` содержит ссылку на класс `Login`. В результате исполнения ордера формируется объект класса `Deal`, который также содержит ссылку на объект класса `Order`. Диаграмма классов тестовой подсистемы представлена на рисунке 2.4.

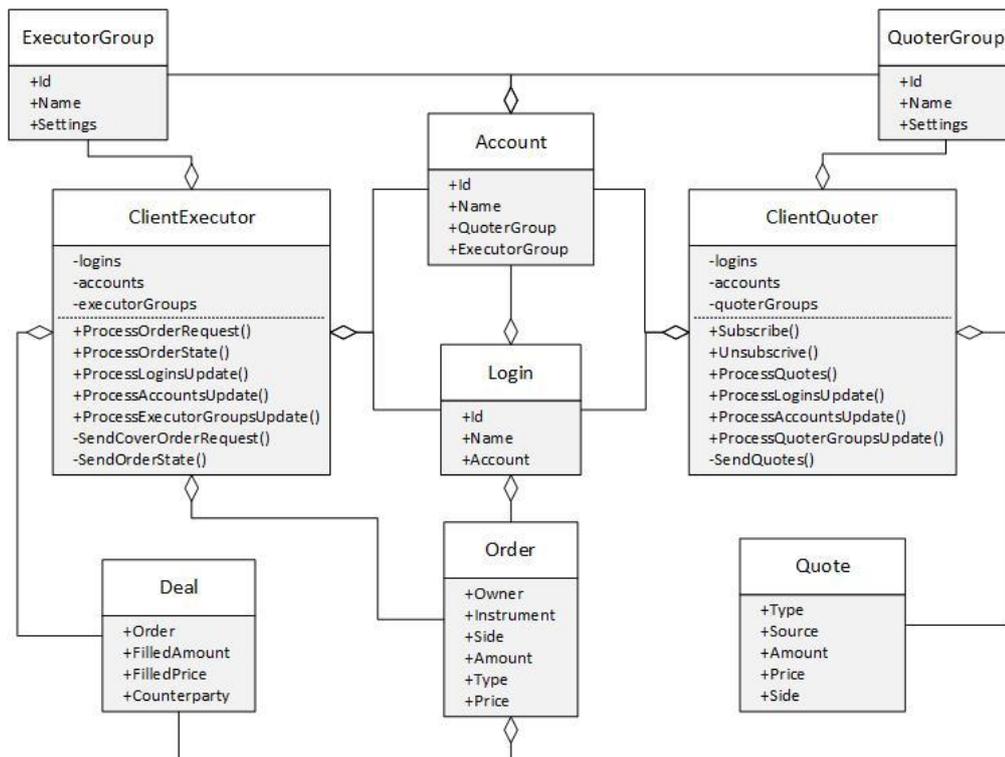


Рис. 2.4. – Диаграмма классов тестовой подсистемы

Рассмотрим взаимодействие подсистемы развёртывания нагрузочного контура с торговой системой и инженером сопровождения. В результате реализации проектируемого решения вся деятельность инженера сопровождения по настройке нагрузочной среды должна сводиться к запуску соответствующих программных средств. В первую очередь запускается подсистема развёртывания тестовой среды. По завершении данной процедуры торговая система полностью готова к запуску в режиме нагрузочного тестирования. После запуска, работы под нагрузкой и остановки торговой системы на жёстком диске появятся файлы логирования, которые представляют собой первичные результаты тестирования. После запуска средства анализа файлов логирования системы будет сформирован файл отчёта.

Диаграмма последовательности использования автоматизированного средства развёртывания нагрузочного контура представлена на рисунке 2.5.

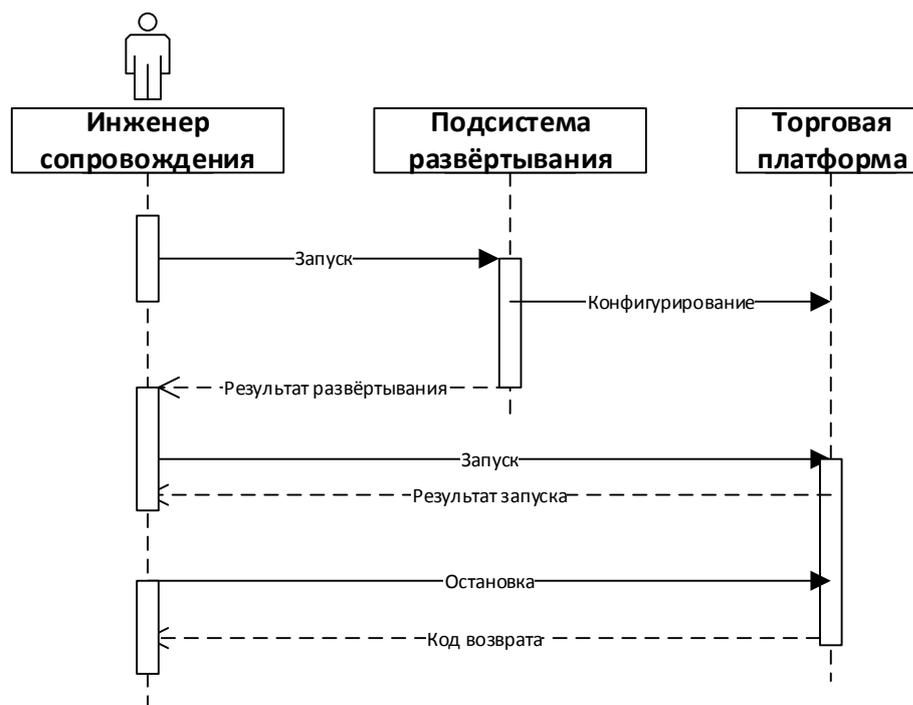


Рис. 2.5. – Диаграмма последовательности развёртывания нагрузочной среды

Выполним процедуру трансформации диаграммы классов в логическую модель данных (ER-диаграмму) по методологии IDEF1X [10]. Основными сущностями, отражающими описание работы тестовой подсистемы, являются: правила формирования котировок и исполнения, торговый счёт, клиент, торговая заявка; финансовая сделка, представляющая собой исполненную торговую заявку и котировка. Формирование котировок и исполнение ордеров для клиентов происходит по определённым правилам. Поэтому каждый торговый счёт принадлежит какой-либо группе котирования и исполнения. Одной группе могут соответствовать несколько счетов. Клиент проходит авторизацию в торговой системе с помощью логина, при этом каждому логин имеет торговый счёт. Стоит отметить, что клиенты под несколькими логинами могут осуществлять торговлю на одном торговом счёте. Клиент выставляет торговые заявки (ордера). В результате исполнения ордера формируется сделка. На рисунке 2.6. представлена логическая модель данных.

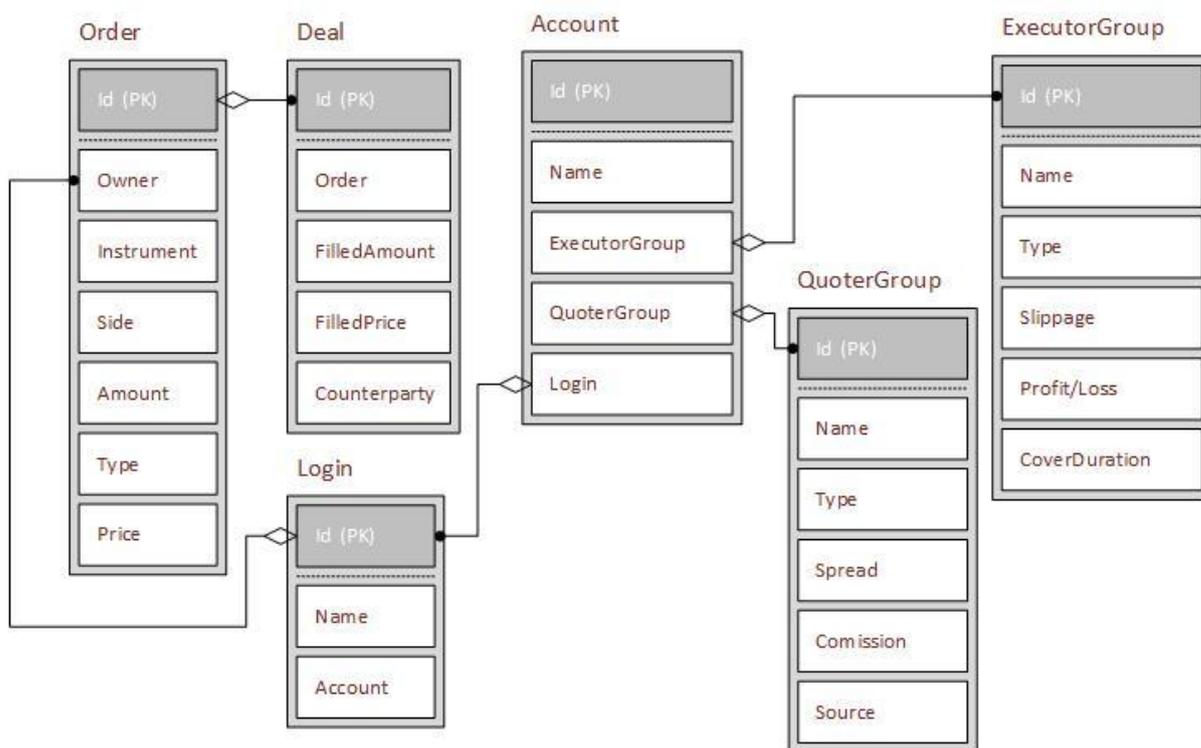


Рис. 2.6. – Логическая модель данных

Разработанная модель не содержит повторяющихся групп в отдельных таблицах, все наборы связанных данных выделены в модельные сущности, при этом каждая сущность идентифицирована с помощью первичного ключа. В модели отсутствуют наборы значений, относящихся к нескольким записям. Связи всех сущностей модели построены на основе первичных ключей. Таким образом, разработанная модель нормализована до второй нормальной формы.

### 2.1.2. Используемые классификаторы и системы кодирования

В компании приняты и используются следующие классификаторы: диалект, тип инструмента, тип котировок, тип логина. Данные классификаторы являются специфическими для данной информационной системы, однако их аналоги существуют в других похожих продуктах. Все перечисленные классификаторы имеют порядковую систему кодирования, представлены в базе данных с помощью целочисленного типа `integer`, а в языке программирования `C++` с помощью перечисления `enum` или `enum class`.

Ключевым для реализации тестовой подсистемы является классификатор диалекта. Он необходим для организации обмена сообщениями по протоколу Fix между адаптером и эмулятором провайдера ликвидности, а также между Fix-сервером торговой системы и эмулятором пользовательских подключений. В настоящее время торговая платформа NTPro поддерживает несколько десятков диалектов для адаптеров провайдеров ликвидности. В тоже время число диалектов пользовательских подключений не больше десяти. Из этого факта вытекает невозможность эмуляции провайдеров ликвидности с использованием их диалектов существующими средствами самой торговой платформы. Поэтому для этих целей предлагается использовать два диалекта: Fastmatch и NTPro. Первый обеспечивает систему ордерными котировками, второй – бэндновыми. Тип котировок представлен классификатором, имеющим два значения: ордерный и бэндновый.

Классификатор типа инструмента представляет вариант использования финансового инструмента в торговой платформе. Тип может принимать следующие значения: торговый, форвардный, своп-инструмент, поставочный. В настоящее время обмен информацией с внешними системами осуществляется только по торговым инструментам, поэтому остальные типы в рамках реализации задачи не фигурируют.

Классификатор типа логина представляет роль пользователя в торговой системе. Укрупнённо можно выделить три категории ролей: клиентские, менеджерские и технические. В рамках реализации задачи используются клиентские торговые логины и технические логины провайдеров ликвидности.

### 2.1.3. Характеристика нормативно-справочной и входной оперативной информации

Нормативно-справочная информация автоматизированного тестового контура представляет собой конфигурационные файлы электронной торговой системы. Каждому процессу распределённой торговой системы

соответствует свой конфигурационный файл, содержащий общую информацию, необходимую для запуска и специфические настройки каждой подсистемы. Структурно конфигурационные файлы представляют собой xml-документы с древовидной структурой, имеющей несколько корневых элементов. Процессу эмулятора, имитирующего поток данных от провайдеров ликвидности и от клиентов, также соответствует свой конфигурационный файл, структурно единообразный с другими конфигурационными файлами системы. Описание структуры конфигурационных файлов процессов представлено в таблице 2.1.

Таблица 2.1. – Структура конфигурационных файлов

Наименование	Описание	Дочерние теги
General	Общая информация, необходимая для запуска процесса	ServiceId, ServiceName, Cores, Mode, Environment, LocationName, DataBaseConnectionString
Subsystems	Описание и определение подсистем процесса	Subsystem
Subsystem	Описание конкретной подсистемы.	Name, Type
Subsystem/Type	Тип подсистемы. Для процесса эмулятора допустимыми значениями являются FixServer и Testing.	
MakerSettings/Maker	Настройки генерации тестовых данных	MakerCode, OrderQuoterConfig
MakerEmulator/Services/Service	Настройки сетевого соединения провайдера ликвидности и его адаптера	Code, MakerCode, (Sender Target)CompId

Общая информация, необходимая для запуска процесса эмулятора, задаётся в дочерних тегах General. Важным здесь является тег Cores, который позволяет задать ядра процессора, выделяемые процессу, а также их тип: busywait или sleep. В процессе эмулятора будут использоваться подсистемы двух типов: Testing (для генерации тестовых данных) и FixServer (для организации сетевого обмена сообщениями по протоколу Fix). В описании

подсистемы Testing задаётся количество и тип генераторов тестовых данных и их распределение на сетевые компоненты.

Оперативная информация представляет собой настройки генерации тестовых данных и входные параметры итерации тестирования. Настройки генерации потока котировок задаются в конфигурационных файлах `quote_generator_....conf`, получаемых на основе шаблонов `quote_generator_scenario.conf.template` и `quote_generator_scenario_item.conf.template`. Первый шаблон задаёт поля файла, второй правила формирования каждой записи. Описание структуры конфигурационного файла генератора котировок представлено в таблице 2.2.

Таблица 2.2 – Конфигурационный файл генератора котировок.

Наименование поля	Описание	Источник
InstrumentName	Финансовый инструмент	БД
MaxAmount	Максимальный котируемый объём	БД
QuoteCount	Количество котировок	Константа
QuoteType	Тип котировок (ордерные, бэндовые)	БД
QuoteMode	Алгоритм генерации	Константа
Period	Период колебаний	Константа
Price	Цена	БД
Amplitude	Амплитуда колебаний	$1000 * PriceIncrement$
PriceIncrement	Приращение цены	БД
Spread	Спред ценового стакана	$10 * PriceIncrement$

Для каждого эмулятора провайдера ликвидности генерируется свой конфигурационный файл генератора котировок, так как каждый провайдер ликвидности имеет свой набор финансовых инструментов и их параметров.

Файл добавления новых провайдеров ликвидности в тестовый контур `add_connectors.conf` позволяет указать количество добавляемых адаптеров определённого типа.

#### 2.1.4. Характеристика базы данных

Нормативно-справочная информация автоматизированного тестового контура, хранящаяся в БД, представляет собой настройки адаптеров провайдеров ликвидности, настройки финансовых инструментов, последние

цены по финансовым инструментам. Настройки провайдеров ликвидности представлены в таблицах `connectors` и `connector_dialects`. Последняя содержит перечисление всех поддерживаемых в системе диалектов сетевого протокола `Fix`. В таблице `connectors` имеется ссылка в виде внешнего ключа на таблицу `connectors_dialects`. Информация по инструментам хранится в таблицах `instruments_basis` и `instruments`. Первая содержит базовые настройки финансового инструмента, единые для всех платформ: тип инструмента и его описание. Вторая таблица содержит настройки инструмента, специфичные для каждой платформы (точность, диапазон объёмов и другие настройки). Таблица `last_prices` содержит последние цены покупки и продажи финансового инструмента, которые используются в тестовой подсистеме для инициализации процесса генерации котировок.

#### 2.1.5. Характеристика результатной информации

В результате работы торговой системы, обрабатывающей нагрузочную информацию, формируются файлы логирования, содержащие информацию о состоянии системы в каждый момент времени. Разработанный инструмент анализа нагрузки системы позволяет получить файл, содержащий агрегированную информацию за весь период тестирования. В выходном файле содержится три таблицы: обработка событий, состояние очередей и состояние потоков.

Каждая запись таблицы «Обработка событий» содержит характеристики, позволяющие однозначно идентифицировать тип сообщения в системе: компонент источник, компонент приёмник, API и идентификатор сообщения. Время обработки котировок представлено в виде минимального, среднего и максимального времени. На основании анализа данных показателей можно сделать вывод о состоянии системы. Таблица «Состояние очередей» содержит информацию о среднем количестве сообщений в очереди компонента. Если данный показатель превышает нормативные значения, можно сделать вывод о недопустимости указанной нагрузки.

Таблица «Состояние потоков» показывает степень загруженности всех потоков системы в процентах. На основании анализа данной таблицы можно сделать вывод о наиболее уязвимых частях системы с точки зрения производительности.

## **2.2. Физическое моделирование автоматизированного нагрузочного контура для электронной торговой системы**

### **2.2.1. Выбор архитектуры автоматизированного нагрузочного контура для электронной торговой системы**

Автоматизированный нагрузочный контур торговой системы представляет собой инструментарий (toolkit), необходимый для анализа системы. Следовательно, архитектура реализации данной задачи зависит от архитектуры существующей платформы. Торговая платформа NTPro реализована с использованием распределённой клиент-серверной архитектуры. Такое решение позволяет использовать возможности гибкого масштабирования системы. Компонентный подход в представлении внутренних сущностей системы обеспечивает надёжную инкапсуляцию и разграничение ответственности различных участков кода. Эмулятор нагрузки также должен быть выполнен в виде распределённого серверного приложения. Это обеспечит гарантированную возможность адаптации конфигурации нагрузочного контура в соответствии с конфигурацией тестируемого объекта. Приложение должно быть многопоточным, что обеспечит отсутствие ограничений при увеличении требуемой выходной нагрузки.

Средства развёртывания контура и анализа результатов нагрузочного тестирования должны представлять собой программы, написанные на скриптовом языке. Преимуществом такого подхода является высокая скорость реализации решения и минимальные затраты на его сопровождение. Такой выбор обусловлен минимальными требованиями к

производительности и отказоустойчивости данных элементов нагрузочного контура.

### 2.2.2. Функциональная схема автоматизированного нагрузочного контура для электронной торговой системы

Автоматизированный нагрузочный контур реализуется на серверной стороне системы, поэтому не предполагает наличия графического диалога. Пользовательское управление осуществляется с помощью консольных средств диалога и конфигурационных файлов. Управление процедурой развёртывания, запуска и анализа результатов работы нагрузочного контура должно быть предельно упрощено, т.к. предполагается достаточно частое использование данного функционала. Функции управления нагрузочным контуром делятся на служебные и основные. К первым относятся ввод сервисных параметров функционирования системы и просмотр файлов логирования, для получения дополнительной информации о состоянии системы. К основным функциям относятся ввод параметров нагрузочного контура, запуск торговой системы, анализ выходного файла нагрузочного контура.

Ввод основных и сервисных параметров в конфигурационные файлы осуществляется с помощью стандартных текстовых редакторов, поддерживаемых операционной системой. Развёртывание системы осуществляется с помощью вызова из консоли скрипта PrepareLauncher с параметрами. Параметр `--emulator` обеспечивает развёртывание среды в режиме нагрузочного тестирования. Параметр `--add-emulator-connectors` обеспечивает добавление дополнительных адаптеров провайдеров ликвидности и их эмуляторов для увеличения нагрузки. Процедура запуска системы в режиме нагрузочного тестирования ни чем не отличается от процедуры запуска системы в обычном режиме. Анализ выходного файла нагрузочного тестирования, как и анализ файлов логирования системы, производится с помощью стандартных средств просмотра текстовых файлов.

### 2.2.3. Структурная схема автоматизированного нагрузочного контура для электронной торговой системы

Программа развёртывания нагрузочной среды состоит из нескольких модулей, описание которых представлено в таблице 2.3.

Таблица 2.3 – Модули программы развёртывание тестовой среды.

Наименование модуля	Идентификатор	Описание
Модуль запуска программы	Prepare Launcher	Основной модуль программы. Содержит функции запуска, анализа параметров командной строки, а также корневые функции развёртывания всех основных элементов системы.
Модуль константной информации	Constants	Содержит условно-постоянную информацию процесса развёртывания среды. Например, пути к целевым и исходным директориям, строковые и временные константы.
Модуль преобразования данных	DataConversion Utils	Содержит функции конвертации различных форматов данных, используемые в программе.
Модуль работы с БД	DbUtils	Инкапсулирует работы с БД. Содержит функции выполнения скриптов, создания и удаления БД, проверки сервисной информации.
Модуль работы с файловой системой	FileSystemUtils	Вспомогательный модуль, облегчающий работу с файловой системой, позволяющий перемещаться по дереву каталогов, создавать директории и др.
Модуль формирования конфигурационных файлов	PrepareConfigs	Служит для генерации, копирования, перемещения, резервного копирования конфигурационных файлов.
Модуль развёртывания эмуляторов	PrepareEmulator	Содержит функции и классы получения информации из БД и конфигурационных файлах об основных параметрах системы. На основе полученной информации происходит изменение стандартного окружения системы для обеспечения требуемой нагрузки на систему.
Модуль сервисных функций	ScriptUtils	Инкапсулирует сервисные функции, такие как вывод информации об ошибке, аварийное завершение работы программы.
Модуль работы с форматом XML	XmlUtils	Содержит функции чтения и записи xml-файлов, перемещения по дереву элементов, получения информации из элемента и его редактирование.

Как видно из представленной таблицы дерево проекта содержит сервисные, управляющие и основные функции, объединённые и сгруппированные в несколько модулей. Такая организация проекта позволяет обеспечить требуемый уровень инкапсуляции, что облегчает разработку и поддержку программного обеспечения.

#### 2.2.4. Описание программных модулей

Рассмотрим функционирование основного модуля программы развёртывания нагрузочного контура – модуля развёртывания эмулятора, блок-схема алгоритма которого приведена на рисунке 2.7.

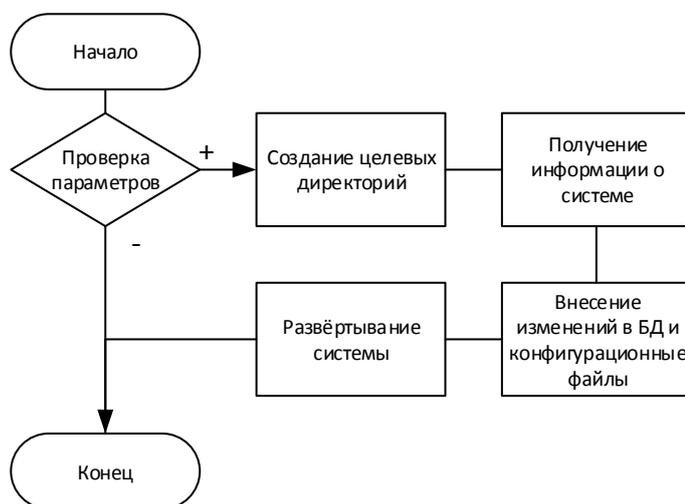


Рис. 2.7. – Блок-схема алгоритма развёртывания эмулятора

На первом шаге алгоритма развёртывания нагрузочного контура производится анализ входных параметров. Входными параметрами в данном контексте являются параметры командной строки и список конфигурационных файлов, соответствующих текущему варианту развёртывания. Если проверка пройдена, начинается процедура подготовка файловой системы, создаются требуемые директории. Далее происходит анализ БД и конфигурационных файлов. В результате анализа определяется список адаптеров провайдеров ликвидности и сервисов, запущенных на

данном сервере. На основании полученных данных происходит внесение соответствующих изменений в БД и конфигурационные файлы системы. Адаптеры провайдеров ликвидности настраиваются на получение данных от эмулятора. Эмулятор конфигурируется в соответствии с имеющимися адаптерами провайдеров ликвидности и сервисами клиентов. На последнем шаге происходит перемещение созданных файлов конфигурации в целевые директории. В число таких файлов входят конфигурационный файл эмулятора, конфигурационные файлы сценариев генерации котировок для каждого провайдера ликвидности.

### **2.3. Технологическое обеспечение задачи автоматизированного нагрузочного контура для электронной торговой системы**

Эмулятор представляет собой приложение, состоящее из нескольких компонентов. Компоненты должны быть изолированы друг от друга. Такое решение позволяет построить гибкую информационную систему с динамичным потолком производительности. При достижении предельных показателей производительности систему можно быстро усовершенствовать путём механического добавления вычислительных мощностей. Кроме того, это позволит облегчить процесс добавления новой функциональности, даст возможность привлекать для этого несколько специализированных команд разработчиков. Компоненты должны обмениваться информацией друг с другом с помощью API, поддерживающих как локальный, так и удалённый режим работы.

Для реализации серверного приложения подойдёт язык программирования C++, обеспечивающий наилучшую производительность, за счёт предоставления доступа к низкоуровневым средствам операционной системы. Недостатком данного языка является сложность и трудоёмкость разработки. Для реализации сетевого обмена подойдут библиотеки `libuv` [18] или `asio`. Для преобразования информации в бинарный формат можно

использовать библиотеку `cereal`. Также рекомендуется использовать библиотеку `boost`, которая предоставляет надёжные средства, существенно расширяя возможности библиотеки `stl`.

В качестве операционной системы серверной части рекомендуется выбрать один из дистрибутивов Linux, например Arch, сочетающий в себе удобство использования, гибкость настройки и большой репозиторий пакетов.

Программа развёртывания нагрузочного контура представляет собой скриптовую программу, реализованную на языке программирования Python [19]. В результате достигается наибольшая скорость разработки и простота реализации и сопровождения разрабатываемого решения.

#### **2.4. Контрольный пример реализации автоматизированного нагрузочного контура для электронной торговой системы**

Для тестирования процесса развёртывания нагрузочной среды необходимо вызвать команду `PrepareLauncher` с различными параметрами. При попытке развёртывания тестового окружения на тех контурах, на которых это запрещено регламентными документами компании, в консоль выведется сообщение об ошибке, программа завершит своё выполнение. Допустимыми вариантами вызова программы являются: `./PrepareLauncher [DbName] [local|dev|test] [alfa|others|moex|london] [--emulator [--add-emulator-connectors]]`. Допустимыми окружениями являются `local`, `dev` и `test`. При попытках вызвать процедуру на окружениях `prod` или `uat` в консоль выведется сообщение об ошибке «Wrong environment for emulator». Допустимыми типами серверов являются `alfa`, `others`, `moex` и `london`. При попытках развернуть эмулятор с другими типами серверов в консоль выведется сообщение об ошибке «Wrong server type for emulator». При указании опции `--emulator` адаптеры провайдеров ликвидности будут настроены на получение информации от эмулятора. В конфигурационных

файлах будут проставлены ip-адреса локального хоста 127.0.0.1 и порты эмулятора. При запуске системы в файлах логирования сообщений протокола Fix провайдеров ликвидности появятся сообщения, представленные на рисунке 2.8.

```
/home/admin/NTPPro/log/C~ Market_2018-04-30.txt 1620/50G 0%
[03:00:00.000678108] [I] IN 8=FIX.4.4|9=0601|35=W|49=ABOT-DEMO-MD|56=NTPRO-DEMO
-MD|34=9679718|52=20180430-00:00:00.000|262=AUD/CAD84|55=AUD/CAD|64=SPOT|268=10|
269=0|270=60.49224|271=4000000|299=0|110=0|269=0|270=60.49223|271=8000000|299=1|
110=4000000|269=0|270=60.49222|271=12000000|299=2|110=8000000|269=0|270=60.49221
|271=16000000|299=3|110=12000000|269=0|270=60.4922|271=20000000|299=4|110=160000
00|269=1|270=60.49235|271=4000000|299=5|110=0|269=1|270=60.49236|271=8000000|299
=6|110=4000000|269=1|270=60.49237|271=12000000|299=7|110=8000000|269=1|270=60.49
238|271=16000000|299=8|110=12000000|269=1|270=60.49239|271=20000000|299=9|110=16
000000|10=160|
[03:00:00.000822122] [I] IN 8=FIX.4.4|9=0591|35=W|49=ABOT-DEMO-MD|56=NTPRO-DEMO
-MD|34=9679719|52=20180430-00:00:00.000|262=AUD/CHF85|55=AUD/CHF|64=SPOT|268=10|
269=0|270=0.73892|271=4000000|299=0|110=0|269=0|270=0.73891|271=8000000|299=1|11
0=4000000|269=0|270=0.7389|271=12000000|299=2|110=8000000|269=0|270=0.73889|271=
16000000|299=3|110=12000000|269=0|270=0.73888|271=20000000|299=4|110=16000000|26
9=1|270=0.73903|271=4000000|299=5|110=0|269=1|270=0.73904|271=8000000|299=6|110=
4000000|269=1|270=0.73905|271=12000000|299=7|110=8000000|269=1|270=0.73906|271=1
6000000|299=8|110=12000000|269=1|270=0.73907|271=20000000|299=9|110=16000000|10=
213|
```

Рис. 2.8. – Пример файла логирования провайдера ликвидности

Тип сообщения указывается в теге 35, тип W – MarketDataFullSnapshot содержит стакан котировок. В файлах логирования сообщений клиентских сервисов появятся сообщения, представленные на рисунке 2.9.

```
/home/admin/NTPPro/log/P~M Order_2018-04-30.txt 6021/895K 0%
[00:03:38.587590145] [I] OUT 8=FIX.4.4|9=0227|35=8|49=NTPRO|56=BBRU_OM|34=8|52=2
0180429-21:03:38.585|6=0.0|11=ce616bbc-e043-4f74-aa45-8238005676a8|14=0|17=11247
201-n|37=b8812c99-be78-4be8-b3a1-394cd8ae6b9f|38=22853|39=0|40=2|44=1.59558|54=1
|55=EUR/AUD|59=3|150=0|151=22853|10=153|
[00:03:38.587598025] [I] OUT 8=FIX.4.4|9=0293|35=8|49=NTPRO|56=BBRU_OM|34=9|52=2
0180429-21:03:38.585|6=1.5944698726644|11=ce616bbc-e043-4f74-aa45-8238005676a8|1
4=22853|17=1829659|32=22853|31=1.59447|37=b8812c99-be78-4be8-b3a1-394cd8ae6b9f|3
8=22853|39=2|40=2|44=1.59558|54=1|55=EUR/AUD|59=3|60=20180429-21:03:38.585|64=20
180503|150=F|151=0|10=091|
```

Рис. 2.9. – Пример файла логирования клиентского сервиса

Тип сообщения указывается в теге 35, тип 8 – ExecutionReport содержит информацию об исполнении клиентской заявки. Наличие указанных сообщений говорит о том, что система работает под нагрузкой. После завершения работы системы будет сформирован табличный файл,

содержащий информацию о состоянии системы в период работы под нагрузкой. Пример содержимого файла агрегированной статистики представлен в таблице 2.4.

Таблица 2.4 – Пример файла статистики.

Раздел	Пример содержимого
Очереди событий	<b>Process;SrcComponent;SrcProject;SrcApi;SrcEvent;DstComponent;DstProject;DstApi;DstEvent;MinTime(ns);MaxTime(ns);AvgTime(ns);Count;SumTime(ns)</b> AlfaPrincip;QUOTERMAKERMD;10;22;4;ALFNNTPROFSSM;10;22;4;26464686002;180306050251;95858553199;4202317;402828027706833548 AlfaPrincip;ALFNNTPROFSSM;0;0;188;ALFNNTPROFSSMBS;0;0;188;14022290273;165193960150;75906519347;6654659;505132002131335472 NewUiServer;NewUiServerUVRD;0;0;0;NewUiServerBGTD;0;0;0;998614271;998661794;998638032;2;1997276065
Обработка событий	<b>Process;Api;Min;Max;MaxTime;Avg;MeasCount;Velocity(Q/sec)</b> AlfaPrincip;AlfaPrincip.AlfaFix.MarketManager.MarketEngineApiEngine;820082;6402392;2018-04-0214:57:42.003788472;3458035;3002;18753.09 AlfaPrincip;Client.MarketApi.ALFNNTPROFSSM;828637;5207576;2018-04-0214:57:42.038808776;2939148;2993;14711.68 AlfaPrincip;Tracing2;0;39282;2018-04-0214:57:20.720393248;345;2975;0.53
Загрузка потоков	<b>Process;Thread;CycleCount;AvgQueueHits(%)</b> <b>MaxQueueHits;TotalLoadPercent;MaxTime(ns);Count</b> AlfaPrincip;NewFixServerManagersThread;826000;165.48;100.0;99.447;62022989;5 NewUiServer;UvServerTransportThread0;124873800;100.00;100.0;98.793;50977;5 UiServerLnd;UvServerTransportThread0;168211800;100.00;100.0;98.575;4084;5 AlfaPrincip;NewFixServerNetworkThread;1327200;200.05;100.0;98.428;118154515;5 AlfaPrincip;AggregatorThread;31031400;3.82;5.218;95.701;32619462;5 PrincipisLnd;AggregatorThread;47319600;1.93;1.725;89.284;2239003;5

На основании данной информации делается вывод о допустимости текущих параметров для существующей конфигурации системы. Поля MinTime, MaxTime и AvgTime раздела «Очереди событий» содержат минимальное, максимальное и среднее время обработки события за период измерения. Поля Min и Max раздела «Обработка событий» содержат минимальное и максимальное количество сообщений в очереди за период измерения. Поле TotalLoadPercent раздела «Загрузка потоков» показывает какое время (в процентах от периода измерения) поток выполнял полезную работу.

## **Выводы по второй главе**

На основании концептуального моделирования предметной области была разработана логическая модель предметной области. В разделе 2.1.1. Логическая модель и её описание был осуществлён переход от логических моделей ТО-ВЕ к описанию вариантов использования системы, а также последовательности работы с системой. В данном разделе также представлены разработанные диаграммы классов, а также схема базы данных. Анализ условно-постоянной, и переменной информации показал, что в системе используется большое количество специфических данных, которые должны быть учтены при разработке тестовой подсистемы. Основная задача при принятии проектных решений – поиск «золотой середины» между универсальностью, простотой, лёгкостью поддержки разрабатываемой системы и её соответствием всему многообразию данных представленных в торговой системе. В разделе 2.2. Физическое моделирование АИС был осуществлён выбор архитектуры решения. Так как проектируемые программные средства представляют собой набор инструментов для организации нагрузочного тестирования, то архитектурные подходы к разработке каждого инструмента различны и учитывают его особенности. Разработанное дерево модулей и функций программного продукта позволяет оптимальным образом инкапсулировать функциональность и данные каждой сущности предметной области. Технологические решения, принятые в разделе 2.3, также имеют разнообразный характер и адаптированы под специфику решаемой подзадачи. Контрольный пример отражает наиболее общие варианты использования разработанного решения и примеры взаимодействия пользователя с программными средствами.

# **ГЛАВА 3. ОЦЕНКА И ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ВНЕДРЕНИЯ АВТОМАТИЗИРОВАННОГО НАГРУЗОЧНОГО КОНТУРА ДЛЯ ЭЛЕКТРОННОЙ ТОРГОВОЙ СИСТЕМЫ**

## **3.1. Выбор и обоснование методики расчета экономической эффективности от автоматизированного нагрузочного контура для электронной торговой системы**

Рассмотрим варианты методик обоснования экономической эффективности автоматизации нагрузочного контура. Расчет экономической эффективности, исходя из жизненного цикла проекта разработки и внедрения подсистем корпоративной информационной системы, предполагает наличие разрабатываемой в рамках решения комплексной информационной системы, с помощью которой меняются бизнес-процессы организации. В рамках данной работы подобные подходы применяться не будут, так как разрабатываемое решение будет функционировать в рамках существующей автоматизированной электронной системы торгов. Применимым к данной работе является расчет прямой эффективности от внедрения информационной системы по сравнению с базовым вариантом существующей организации обработки информации.

Для анализа экономической эффективности данной работы необходимо выполнить сравнение старого и нового вариантов организации процесса нагрузочного тестирования. Экономическая эффективность разработки (Э) складывается из двух составляющих:

— косвенного эффекта, который, характеризуется увеличением надёжности и отказоустойчивости всей торговой системы. Рост данных характеристик достигается с помощью чёткого понимания предельно допустимой нагрузки на систему. Внедрение решения позволит тем самым добиться уменьшения количества внештатных ситуаций, связанных с

превышением нагрузки, и как следствие – снижение уровня сбоев при исполнении клиентских заявок. Повышение надёжности системы исполнения заявок в свою очередь приведёт к росту привлекательности компании на конкурентном рынке и увеличению числа клиентов.

— прямого эффекта, который характеризуется снижением трудовых, стоимостных показателей за счет сокращения времени развёртывания нагрузочного контура, сокращения трудоёмкости работы и стоимостных затрат анализа результатов нагрузочного тестирования, повышением достоверности и точности получаемых результатных показателей.

Для расчета прямого эффекта от внедрения разработанного программного продукта необходимо рассмотреть показатели трудовых и стоимостных затрат.

К трудовым показателям относятся следующие:

1. Абсолютное снижение трудовых затрат, рассчитываемое по формуле:

$$\Delta T = T_0 - T_1, \quad (1)$$

где  $T_0$  – время, затрачиваемое на выполнение подготовки и развёртывания нагрузочного контура, а также анализа полученных результатов в базовом варианте;

$T_1$  – время, затрачиваемое на выполнение автоматизированного развёртывания нагрузочного контура в проектном варианте.

2. Коэффициент относительного снижения трудовых затрат  $K_T$  (в процентах), для расчета которого используется следующая формула:

$$K_T = (\Delta T / T_0) * 100 \%. \quad (2)$$

3. Индекс снижения трудовых затрат, рассчитываемый следующим образом:

$$Y_T = T_0 / T_1. \quad (3)$$

К стоимостным показателям относятся следующие:

1. Абсолютное снижение стоимостных затрат.

$$\Delta C = C_0 - C_1, \quad (4)$$

где  $C_0$  – стоимостные затраты на нагрузочное тестирование по базовому варианту,

$C_1$  – стоимостные затраты на нагрузочное тестирование по предлагаемому варианту.

2. Коэффициент относительного снижения стоимостных затрат  $K_C$  (в процентах), определяемый по следующей формуле:

$$K_C = (\Delta C / C_0) * 100 \% . \quad (5)$$

3. Индекс снижения стоимостных затрат, рассчитываемый по формуле:

$$Y_C = C_0 / C_1 . \quad (6)$$

Коэффициенты  $K_C$  и  $Y_C$  характеризуют рост производительности труда за счет внедрения более экономичного варианта проектного решения.

Помимо рассмотренных показателей целесообразно также рассчитать срок окупаемости затрат на внедрение новой технологии нагрузочного тестирования ( $T_{OK}$ ):

$$T_{OK} = K_{\Pi} / \Delta C , \quad (7)$$

где  $K_{\Pi}$  – капитальные затраты на создание проекта.

Учитывая, что в среднем нагрузочное тестирование выполняется раз в месяц, рассчитываемый таким образом срок окупаемости измеряется в месяцах.

### **3.2. Расчёт показателей экономической эффективности проекта реализации автоматизированного нагрузочного контура для электронной торговой системы**

Временные затраты на нагрузочное тестирование складываются из временных затрат на подготовку тестовой среды, само тестирование и анализ его результатов. В базовом варианте на подготовку и развёртывание нагрузочного контура, внесение требуемых изменений в конфигурационные файлы, настройку базы данных требовался один рабочий день. Данная деятельность осуществлялась программистом и инженером службы

поддержки попеременно. Трудозатраты программиста – 7 часов, инженера сопровождения – 1 час. Собственно тестирование занимало в среднем четыре часа рабочего времени инженера сопровождения, а анализ результатов также в среднем требовал в сумме восьми часов рабочего времени нескольких программистов. Таким образом  $T_0 = 7 + 1 + 4 + 8 = 20$  часов.

В проектном варианте трудовые затраты на развёртывание нагрузочной среды сократились до 1 часа. При этом все работа по настройке выполняется инженером сопровождения согласно регламенту нагрузочного тестирования. Трудовые затраты на само тестирование остались неизменными, однако увеличилось количество прогонов, выполняемых за тоже время. Временные затраты на анализ результатов тестирования сократились до 1 часа. Таким образом  $T_1 = 1 + 4 + 1 = 6$  часов.

Стоимостные затраты на нагрузочное тестирование складываются в основном из затрат на заработную плату сотрудников. Расходами на электроэнергию, аренду здания, оплату интернет-трафика в данном случае можно пренебречь из-за их незначительного влияния на итоговый результат. Средние расходы компании на оплату труда программиста с учётом всех отчислений в государственные фонды составляет 200 000 рублей в месяц. Средние расходы компании на оплату труда инженера сопровождения с учётом всех отчислений в государственные фонды составляет 120 000 рублей в месяц. Стоимость одного часа работы программиста составляет  $200\,000 / 168 = 1190,48$  рублей. Стоимость одного часа работы инженера сопровождения составляет  $120\,000 / 168 = 714,29$  рублей.

Трудовые затраты программиста в базовом варианте составляют  $T_{\text{опр}} = 7 + 8 = 15$  часов. Трудовые затраты инженера сопровождения в базовом варианте составляют  $T_{\text{ис}} = 1 + 4 = 5$  часов.

Трудовые затраты программиста в проектном варианте составляют  $T_{\text{1пр}} = 1$  час. Трудовые затраты инженера сопровождения в проектном варианте составляют  $T_{\text{1ис}} = 1 + 4 = 5$  часов.

Стоимостные затраты компании на проведение нагрузочного тестирования в базовом варианте  $C_0 = 15 * 1190,48 + 5 * 714,29 = 21428,65$  рублей. Стоимостные затраты компании на проведение нагрузочного тестирования в проектном варианте  $C_1 = 1 * 1190,48 + 5 * 714,29 = 4761,93$  рублей.

Расчёт показателей экономической эффективности, выполненный на основании описанной методики, представлен в таблице 3.1.

Таблица 3.1. – Показатели экономической эффективности

	Затраты		Абсолютное изменение затрат	Коэффициент изменения затрат	Индекс изменения затрат
	Базовый вариант	Проектный вариант			
Трудоёмкость	T <sub>0</sub> (час)	T <sub>1</sub> (час)	$\Delta T = T_0 - T_1$ (час)	$K_T = (\Delta T / T_0) * 100 \%$	$Y_T = T_0 / T_1$
	20	6	14	70	3,33
Стоимость	C <sub>0</sub> (руб.)	C <sub>1</sub> (руб.)	$\Delta C = C_0 - C_1$ (руб.)	$K_C = (\Delta C / C_0) * 100 \%$	$Y_C = C_0 / C_1$
	21428,65	4761,93	16666,72	77,78	4,5

На основании рассчитанных показателей можно сделать вывод об эффективности реализации проекта. Так трудовые затраты сотрудников компании на проведение нагрузочного тестирования снизились в 3,33 раза, а стоимостные затраты компании в 4,5 раза.

Капитальные затраты на создание проекта состоят в расходах компании на оплату труда разработчика. Расходами на электроэнергию, аренду здания, оплату интернет-трафика в данном случае можно пренебречь из-за их незначительного влияния на итоговый результат. Трудовые затраты разработчиков на проектирование, разработку, и внедрение решения составили 80 часов. Таким образом, капитальные затраты на создание проекта  $K_{\Pi} = 1190,48 * 80 = 95238,4$  рубля. Срок окупаемости  $T_{OK} = 95238,4 / 16666,72 = 6$  месяцев.

## **Выводы по третьей главе**

На основании анализа внедряемого проектного решения был сделан выбор в пользу методики расчёта экономической эффективности от внедрения программного продукта по сравнению с базовым вариантом существующей организации нагрузочного тестирования. Было установлено, что выгоды от реализации проекта носят прямой и косвенный характер. Прямые выгоды заключаются в снижении трудовых затрат сотрудников компании на проведение нагрузочного тестирования в 3,33 раза, а стоимостных затрат компании в 4,5 раза. Затраты на реализацию проекта окупятся через 6 месяцев после его внедрения. Помимо этого результатами внедрения проекта станут увеличение надёжности и отказоустойчивости всей системы электронной торговли. Всё это благоприятно повлияет на конкурентоспособность компании и увеличение её клиентской базы.

## ЗАКЛЮЧЕНИЕ

Нагрузочное тестирование является важным элементом тестирования и сопровождения программного обеспечения. Оно позволяет исследовать производительность информационной системы, определить предельно допустимые нагрузки на неё. В результате выполнения работы был произведён анализ предметной области функционирования исследуемой торговой системы – валютного дилинга; была составлена концептуальная и логическая модель «Как есть» процесса нагрузочного тестирования информационной системы.

В результате моделирования было установлено, что существующая в компании процедура нагрузочного тестирования не является оптимальной, так как выходная информация, получаемая в результате выполнения данной процедуры, не удовлетворяет требованиям полноты, достоверности и оперативности. Для устранения этих недостатков, а также уменьшения трудовых и временных затрат сотрудников компании, было предложено разработать и реализовать программные средства развёртывания и функционирования нагрузочного контура электронной торговой платформы.

В работе представлены разработанные диаграммы классов, а также схема базы данных. Анализ условно-постоянной, и переменной информации показал, что в системе используется большое количество специфических данных, которые должны быть учтены при разработке тестовой подсистемы. При проектировании программного обеспечения было учтено противоречие между универсальностью, простотой, лёгкостью поддержки разрабатываемой системы и её соответствием всему многообразию данных представленных в торговой системе. Так как проектируемые программные средства представляют собой набор инструментов для организации нагрузочного тестирования, то архитектурные подходы к разработке каждого инструмента различны и учитывают его особенности. Разработанное дерево модулей и функций программного продукта позволяет оптимальным образом

инкапсулировать функциональность и данные каждой сущности предметной области.

В работе установлено, что выгоды от реализации проекта носят прямой и косвенный характер. Прямые выгоды заключаются в снижении трудовых затрат сотрудников компании на проведение нагрузочного тестирования в 3,33 раза, а стоимостных затрат компании в 4,5 раза. Затраты на реализацию проекта окупятся через 6 месяцев после его внедрения. Помимо этого результатами внедрения проекта станут увеличение надёжности и отказоустойчивости всей системы электронной торговли. Всё это благоприятно повлияет на конкурентоспособность компании и увеличение её клиентской базы.

Логическим продолжением данной работы является переход от реализации автоматизированного нагрузочного контура к разработке полностью автоматических средств тестирования и отладки программного обеспечения. Становится понятно, что основной проблемой при реализации таких решений является их адаптация и масштабирование в соответствии с постоянно изменяющимся объектом тестирования.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. James Chen. Essentials of Foreign Exchange Trading. [Электронный ресурс] : прикладное пособие / - John Wiley & Sons, 2013.- 221 с. - Режим доступа: [https://books.google.ru/books?id=8zTsnBYiDGkC&dq=currency+trading+books&hl=ru&source=gbs\\_navlinks\\_s](https://books.google.ru/books?id=8zTsnBYiDGkC&dq=currency+trading+books&hl=ru&source=gbs_navlinks_s).
2. Куликов С. Тестирование программного обеспечения. Базовый курс [Электронный ресурс] - EPAM Systems, 2018. – 297с. – Режим доступа: [http://svyatoslav.biz/software\\_testing\\_book\\_download/](http://svyatoslav.biz/software_testing_book_download/)
3. Ключко И. А. Информационные технологии в профессиональной деятельности [Электронный ресурс] : учеб. пособие / И. А. Ключко. - Саратов : Вузовское образование, 2014. - 236 с. - (Высшее образование).
4. Бурняшов Б. А. Информационные технологии в менеджменте. Облачные вычисления [Электронный ресурс] : учебное пособие / Б. А. Бурняшов. - Саратов : Вузовское образование, 2013. - 87 с.
5. Вдовин В. М. Информационные технологии в финансово-банковской сфере [Электронный ресурс] : учебное пособие / В. М. Вдовин, Л. Е. Суркова. - Москва : Дашков и К°, 2014. - 301 с. - ISBN 978-5-394-01213-6.
6. Стешин А. И. Информационные системы в организации [Электронный ресурс] : учебное пособие / А. И. Стешин. - Саратов : Вузовское образование, 2013. - 194 с.
7. Силич В. А. Реинжиниринг бизнес-процессов [Электронный ресурс] : учеб. пособие / В. А. Силич, М. П. Силич. - Томск : ТУСУР, 2014. - 199 с. - ISBN 5-86889-330-1.
8. Золотов С. Ю. Проектирование информационных систем [Электронный ресурс] : учеб. пособие / С. Ю. Золотов ; Томский гос. ун-т систем управления и радиоэлектроники. - Томск : Эль Контент, 2013. - 86 с. - ISBN 978-5-4332-0083-8.
9. Artur Lugmayr, Emilija Stojmenova, Katarina Stanoevska, Robert Wellington. Information Systems and Management in Media and Entertainment

Industries [Электронный ресурс]. - Springer, 2016. - 339 с. - ISBN 978-3-319-49407-4.

10. Фадеева О.Ю. Информационные системы в экономике [Электронный ресурс] : учебное пособие / О.Ю. Фадеева, Е.А. Балашова. — Электрон. текстовые данные. — Омск: Омский государственный институт сервиса, 2015. — 100 с. — 978-5-93252-360-5. — Режим доступа: <http://www.iprbookshop.ru/32786.html>

11. Michael Lewis. Flash Boys A Wall Street Revolt. — W. W. Norton & Company, 2014. — 274 с. — ISBN 978-0-393-24466-3

12. Форекс [Электронный ресурс] // Википедия. Режим доступа: <https://ru.wikipedia.org/wiki/Форекс> (дата обращения: 05.05.2018)

13. NTFX Forex on click [Электронный ресурс] // NTFX. Режим доступа: <http://ntfxpro.com/> (дата обращения: 17.03.2018)

14. О компании [Электронный ресурс] // НТ Прогресс. Режим доступа: <http://ntprogress.ru/ntpro/> (дата обращения: 05.05.2018)

15. Quick [Электронный ресурс] // Википедия. Режим доступа: <https://ru.wikipedia.org/wiki/QUIK> (дата обращения: 05.05.2018)

16. MetaTrader 4 [Электронный ресурс] // MetaQuotes Software Corp. Режим доступа: <https://www.metatrader4.com/ru/releasenotes> (дата обращения: 05.05.2018)

17. FinTech [Электронный ресурс] // STARTUPLIFE. Режим доступа: <http://startuplife.by/fintech-startups.html> (дата обращения: 05.05.2018)

18. Libuv [Электронный ресурс] // Википедия. Режим доступа: <https://en.wikipedia.org/wiki/Libuv> (дата обращения: 05.05.2018)

19. Python [Электронный ресурс] // Python. Режим доступа: <https://www.python.org/> (дата обращения: 05.05.2018)