

Для разработки использовалась платформа Java EE. Приложение построено по клиент-серверной модели. Клиентская часть (отображение html-страниц) выполняется браузером. Выполнение серверной части происходит в среде сервера приложений Apache Tomcat, который обрабатывает http-запросы от клиента, формирует SQL-запросы в СУБД, получает от базы данных необходимую информацию и после обработки выдаёт её клиенту. Структурная схема приложения показана на рисунке 1. База данных построена по реляционной модели в СУБД MySQL, но посредством конфигурирования сервера приложение может быть переориентировано на работу с СУБД и других производителей. Эта возможность имеется благодаря использованию интерфейса JNDI — специализированного Java API для доступа к объектам

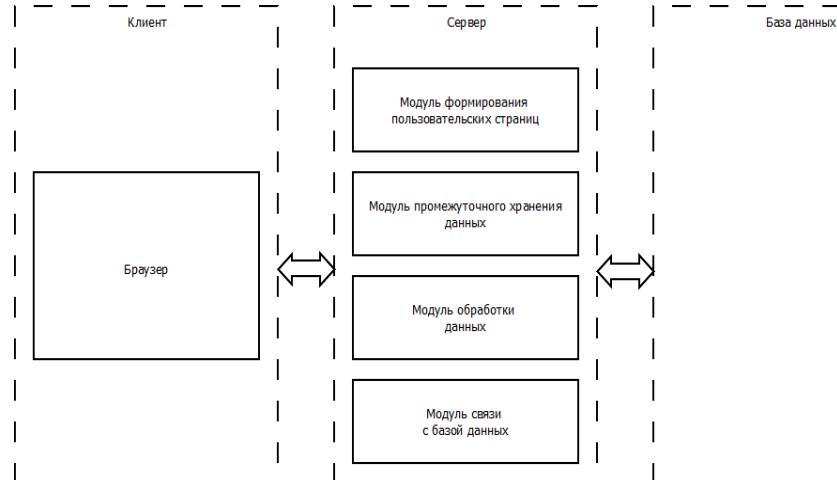


Рисунок 10 — Структурная схема приложения

различного типа. Сервер приложений Tomcat реализует этот интерфейс так, что настройки соединения с БД указываются в конфигурационном файле сервера web.xml, а соединениями с БД управляет сам сервер. Он контролирует их состояние и при необходимости передаёт их приложению, что позволяет обеспечить многопоточность приложения — возможность его использования сразу несколькими пользователями.

Предполагается дальнейшее развитие проекта: расширение функционала приложения (возможность обмена сообщениями между участниками производственного процесса, автоматизация подбора персонала на рабочие проекты исходя из навыков сотрудников), а так же его адаптация под мобильные устройства.

Список использованных источников:

- [1] Перри, Б. Java сервлеты и JSP : сборник рецептов. / Б. Перри — М. : КУДИЦ-ПРЕСС, 2006. — 768 с.
- [2] Thomas, T. M. Java Data Access : JDBC, JNDI, and JAXP / T. M. Thomas — NY : Hungry minds Inc, 2002 — 377 p.
- [3] The Java™ Tutorials : Java Naming and Directory Interface [Электронный ресурс]. — Режим доступа: <https://docs.oracle.com/javase/tutorial/jndi/index.html>
- [4] Apache Tomcat 8 : JNDI Resources HOW-TO [Электронный ресурс]. — Режим доступа: <http://tomcat.apache.org/tomcat-8.5-doc/jndi-resources-howto.html>
- [5] Аткинсон, Л. MySQL : Библиотека профессионала / Л. Аткинсон — М. : Издательский дом "Вильямс", 2002. — 624 с.

## АНАЛИЗ ВЫБОРА СТРУКТУРЫ КОМАНДЫ ДЛЯ РАЗЛИЧНЫХ ПРОЕКТОВ

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Жданов А. Ю.

Волочко А. Т. — д-р. техн. наук, профессор

Современные методы разработки программного обеспечения нацелены на организацию труда в небольших командах, путем повышения эффективности отдельных членов команды.

На сегодняшний день идеальным числом членов команды является  $7 \pm 2$ . В своей статье Миллер утверждает, что 7 является важнейшим числом для описания мощности обрабатывающих возможностей человеческого мозга. Выбранная цифра определяет максимальное количество «кусков» информации для одновременной обработки мыслительным центром.

Однако подавляющее большинство практик не масштабируются для больших проектов с большим количеством членов команды. С ростом проекта возрастает количество связей, которые должны поддерживать каждый отдельный ее член. На рисунке 1 представлена модель, отражающая связи: которые необходимо поддерживать на различных уровнях работы команды.

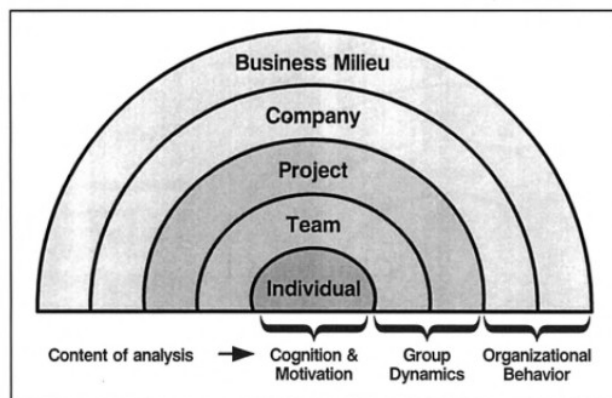


Рисунок 1 – Многослойная модель разработки программного обеспечения

Таким образом на первый план выходит вопрос об эффективной организации членов команды в некоторую структуру, которая обеспечила бы максимальную производительность труда.

В результате проведенного анализа, выделены две:

- команда с централизованным управлением, где в команде выделяется лидер, отвечающий за принятие всех ключевых решений и за осуществление контактов с другими командами
- распределенная команда, где решения принимаются коллективно. При этом важно понимать, что каждый член команды должен обладать всеми знаниями для принятия решения, тогда как в модели с централизованным управлением таковыми знаниями обладает лишь руководитель.

Каждая из моделей имеет свои плюсы и минусы и выбор одной из них должен осуществляться в зависимости от условий проекта. Ниже приведена таблица 1, отражающая критерии выбора.

Таблица 1 – Условия для выбора структуры команды

Условие	Централизованное управление	Распределенная команда
Сложность задачи	простая	сложная
Размер программы	большая	маленькая
Возможные риски	маленькие	большие
Продолжительность проекта	короткая	длительная
Сроки разработки	сжатые	длительные

В ходе работы проанализированы различные подходы к организации команды в зависимости от исходных данных проекта. При возникновении конфликта между требованиями, следует выбирать подход опираясь на требование с наивысшим приоритетом.

Список использованных источников:

1. Healy, S. D., & Rowe, C. (2007). A critique of comparative studies of brain size. *Proceedings of the Royal Society B: Biological Sciences*, 274(1609), 453-464.
2. Dunbar, R.I.M. (1993). *Coevolution of neocortical size, group size and language in humans*, *Behavioral and Brain Sciences* 16 (4): 681-735.

## ИНФОРМАЦИОННАЯ СИСТЕМА РАСЧЁТОВ С КОНТРАГЕНТАМИ

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Заяц В.В.

Борисик М.М. – магистр технических наук

Цель работы – разработать информационную систему расчётов с контрагентами.

Необходимость автоматизации такой части деятельности организации, как учёт расчётов с контрагентами, обусловлена не только причинами усложнения и трудоёмкости данного процесса, но и общей тенденцией к компьютеризации всего общества. В условиях современной конкуренции, организации занимающиеся торговлей, стремятся максимально быть лояльными для своих клиентов, организаций с которыми сотрудничают, своих сотрудников. Для клиентов и организаций это выражается в максимально быстром оформлении всех необходимых сопроводительных документов, предоставлении необходимой информации о товарах и услугах. Для сотрудников это предоставление необходимого, удобного программного решения, для выполнения своих обязанностей. Разработанная программа позволит упростить процесс расчётов с контрагентами, удобна для работы сотрудников организации.