

**В.В.Кокорышкин, С.В.Косачев**

## **Использование эволюционного подхода в проектировании баз данных для упрощения проведения рефакторинга на примере аналитического инструмента поддержки принятия решений в сфере здравоохранения «WorkspaceManager»**

В статье рассмотрена методология эволюционной разработки базы данных на основе СУБД MySQL. Также описан процесс проведения рефакторинга базы данных на примере аналитического инструмента «WorkspaceManager».

**Ключевые слова:** База данных (БД), рефакторинг, эволюционный подход разработки БД, поэтапный подход разработки БД, механизм миграций.

### **Цель проекта**

Целью проекта является создание аналитического инструмента поддержки принятия решений, направленного на значительное сокращение временных издержек, повышение эффективности работы и оптимизации работы медицинского статистика и предоставления наглядной информации для главного врача медицинского учреждения.

Как и при разработке любой информационной системы с большой и сложной базой данных (БД), команда разработчиков столкнулась с необходимостью частого проведения рефакторинга. Чтобы избавиться от сомнений, что процесс разработки идет верно, было принято решение проанализировать один из подходов поэтапной разработки БД и её рефакторинга.

В данной статье предложено описание одного из подходов рефакторинга БД, с помощью которого проводился рефакторинг базы данных для системы «WorkspaceManager».

### **Эволюционный подход к проектированию баз данных и упрощение процесса проведения рефакторинга**

Новым вектором развития в разработке крупных и сложных баз данных становится использование эволюционного подхода проектирования. Использование данного подхода в разработке баз данных крайне эффективно и в скором времени оно должно вытеснить остальные методы.

Эволюционный подход предполагает, что схема базы данных не должна полностью проектироваться заранее еще до начала проекта; схема должна поэтапно наращиваться на протяжении всего процесса разработки проекта, показывая все изменения в требованиях заказчика.

Данный подход базируется на нескольких важных методиках:

1. Весь процесс проектирования и дальнейшего рефакторинга должен проходить в при постоянном взаимодействии разработчика и заказчика.

2. Процесс проведения рефакторинга должен быть итеративным и представлять собой жизненный цикл, который может проходить несколько раз.

3. Все изменения базы данных должны проходить с помощью механизма миграций. Данный механизм позволит внести программистам изменения в структуру БД, которые позднее можно протестировать.

4. Использовать систему контроля версий для всех своих артефактов: кода приложения, функциональных и модульных тестов. Преимуществом данного подхода является то, что всё находится в одном месте и доступно любому участнику группы разработки. При любом возникновении проблемы, мы с лёгкостью можем сделать проверку, так как все изменения БД сохраняются.

5. Каждый разработчик должен иметь свой экземпляр БД, который после изменения, сливается на главную ветку проекта.
6. Как можно чаще интегрировать изменения БД.
7. Весь дальнейший процесс изменения БД, без изменения её семантики считать рефакторингом.

Преимущества эволюционного подхода к разработке баз данных включает в себя следующее:

1. Минимизация бесполезных затрат.
2. Предотвращение необходимости в существенных переделках.
3. Постоянная уверенность в наличии работоспособной системы.
4. Постоянная уверенность в том, что существующий на данный момент проект базы данных имеет наивысшее возможное качество.
5. Применение подхода к разработке, совместимого с подходом других разработчиков.
6. Сокращение общей трудоемкости.

## Процесс проведения рефакторинга

Рефакторинг баз данных — развитие существующей схемы базы данных путем единовременного внесения небольших изменений в целях повышения качества проекта базы данных без изменения ее семантики [1].

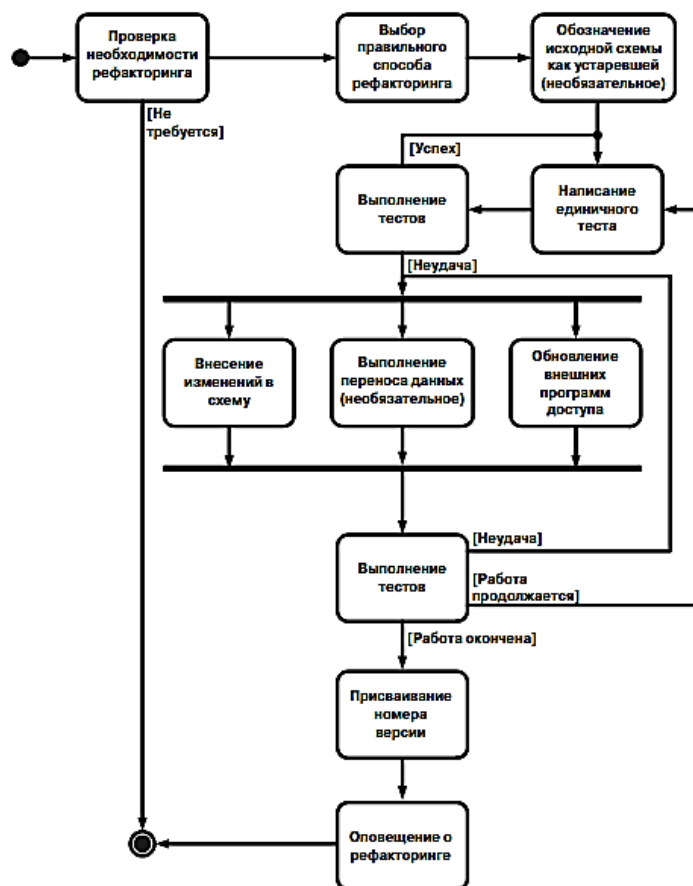


Рисунок 1 - жизненный цикл поэтапного процесса проведения рефакторинга базы данных

Процесс рефакторинга, представляет собой циклический жизненный цикл разработки базы данных. На рисунке 1 представлен весь цикл прохождения рефакторинга: от проверки необходимости проведения до оповещения о проведении.

Данный подход нашел применение в создании аналитического инструмента принятия решений. Во время разработки «WorkspaceManager» в базу данных было внесено множество изменений.

## **Применение эволюционного подхода в системе «WorkspaceManager»**

В ходе работы группа разработчиков, проанализировав представленный в статье подход, применила его при проектировании БД системы «WorkspaceManager». В рамках внедрения эволюционного подхода в проектную деятельность была налажена постоянная связь с заказчиком системы, что позволило разработчикам гораздо быстрее реагировать на внесение изменений в проект. Дальнейшее взаимодействие с прямыми носителями информации позволит предотвратить необходимость заново делать огромную часть работы при изменении БД. В рамках внедрения в проект эволюционного подхода на этапе начального проектирования, внедрили практику использования механизма миграций, который в дальнейшем значительно упростил процесс изменения в БД. Контроль версий системы в рамках процесса рефакторинга осуществлялся в системы контроля версий GitHub.

В ходе работы были добавлены новые сущности и атрибуты, отсутствующие в предыдущих версиях. В связи с требованием заказчика была добавлена сущность «Клиника» с подключенными библиотеками адресов. Из-за того, что многие значения рассчитывались в зависимости от временного промежутка, пришлось добавить атрибут «Дата» в сущности «Стационар», «Дневной стационар» и «Поликлиника». При разработке новых версий базы данных, была проведена оптимизация, в ходе которой команда разработки старалась избавиться от избыточности, которая присутствовала в предыдущих версиях.

При проведении последнего зафиксированного процесса рефакторинга из сущности «Сотрудник» была выделена отдельная сущность «Статус сотрудника», что позволило отразить статус сотрудника во времени.

В конце каждого цикла рефакторинга БД проходила набор функциональных и модульных тестов, в ходе которого выявлялись новые недочеты и дефекты. Новой базе данных присваивалась новая версия и повторно проходил процесс фиксации в системе контроля версий GitHub.

## **Заключение**

Таким образом, применение эволюционного подхода к разработке базы данных увеличило эффективность всего процесса разработки, значительно снизило трудоемкость и временные затраты. На текущий момент проект «WorkspaceManager» насчитывает третью версию в системе контроля версий GitHub. Данный подход позволил избавиться от избыточности и провести полную оптимизацию БД. Постоянное обновление информации, поддержание её в актуальном состоянии и постоянные процессы тестирования позволили повысить качество создаваемой БД.

## **Литература**

1. Эмблер Скотт, Садаладж Прамодкумар Дж. Рефакторинг баз данных: эволюционное проектирование. Пер. с англ. — М.: Вильямс, 2007. — 672 с.
2. Сенченко П.В. Организация баз данных: учебн. пособие. – Томск: Томск. гос. ун-т систем управления и радиоэлектроники, 2004. – 170 с.