

2 АНАЛИТИЧЕСКИЙ РАЗДЕЛ

2.1 Новая технология выполнения выбранной функции управления

Выполнение функции управления непосредственно связано с взаимодействием с системой управления проектами. Новое приложение призвано решить проблемы пользователей, которые возникали при работе в старой системе управления проектами. А именно:

- переключение внимания на неиспользуемые функции;
- трудоёмкость доработки системы под специфические требования организации;
- требование запуска системы и хранения данных на оборудовании организации.

Последнее требование обусловлено политикой информационной безопасности, которая может применяться в IT-компании. Человеческие ресурсы являются самым важным активом любой IT-компании. Это значит, что применяемые оптимизации бизнес-процесса должны быть в первую очередь направлены на снижение трудозатрат сотрудников. Для системного понимания недостатков юзабилити существующей системы управления проектами будет полезно рассмотреть подробнее процессы взаимодействия участников, а именно менеджера проекта, разработчика и тестировщика. Так как есть всего три роли, а количество сочетаний из трёх по два равно трём, следует рассмотреть каждую пару участников процесса отдельно.

Менеджер проекта - разработчик Цикл разработки начинается с получения технического задания разработчиком от менеджера проекта. В некоторых случаях до начала разработки менеджер уточняет у разработчиков сложность реализации требований клиента для более точного планирования этапов и сроков. В ходе разработки менеджер консультирует программистов по вопросам, которые не были оговорены в техническом задании. Само техническое задание обновляется. В конце цикла разработки менеджер пишет проектную документацию по стандарту компании, обращаясь за консультациями к разработчикам.

Менеджер проекта - тестировщик После завершения логически значимого спринта менеджер передаёт новое программное обеспечение в тестирование. Тестировщик получает от менеджера техническое задание.

Разработчик - тестировщик Тестировщик получает от разработчиков целевое программное обеспечение, в виде репозитория исходного кода и запущенного экземпляра приложения. В ходе тестирования тестировщик обращается за консультациями к разработчику. Выполнив тестирование каждой задачи, тестировщик отмечает её статус в системе управления проектами. Задачи, которые не прошли проверку назначаются разработчику. Разработчик обращается к тестировщику по вопросам настройки тестовой среды и условиям воспроизведения найденных ошибок, если в отчёте о тестировании недостаточно информации.

В рассматриваемом бизнес-процессе основным риском является увеличение сроков разработки. Соблюдение планов можно считать метрикой успешности коллектива компании. Когда задачи и проекты выполняются вовремя, в команде присутствует доброжелательная атмосфера, что немаловажно для продуктивного труда. Ниже перечислены основные причины увеличения сроков.

- Недооценка сложности требований;

- Часто изменяющиеся требования;
- Недостаточное количество или компетенция разработчиков;
- Отсутствие автоматизации повторяющихся технологических процессов.

Часть проблем, связанных с оценкой сложности и изменениями требований, должны быть решены путём создания новой системы управления проектами, в которой взаимодействие между менеджером и разработчиком становится удобней. Если требования меняются реже чем завершаются спринты, то изменения могут быть успешно добавлены в следующие этапы разработки.

Проблемы количества и компетенции разработчиков, решаются построением новой кадровой политики. Руководитель организации должен уделять большое внимание на формирование сильной команды разработки. Кадровая политика может быть направлена как на поиск новых опытных компетентных разработчиков, так и на выращивание их в коллективе. Зачастую IT-компании комбинируют эти направления.

Практика показывает, что применение автоматизации технологических процессов сборки, настройки и развёртывания программного обеспечения положительно влияет на продуктивность компании. DevOps инженер является специалистом по автоматизации технологических процессов. Такая автоматизация снимает обязанности по сборке, настройке и развёртыванию с разработчика, что позволяет ему сфокусироваться на реализации требований технического задания. В крупных IT-компаниях специалисты по развёртыванию (или DevOps) могут быть включены в подразделение отдела администрирования, а в небольших эти функции берёт на себя отдел разработки. Помимо развёртывания DevOps оказывает помощь в автоматизации процесса тестирования, участвует в создании интеграционных, нагрузочных и регрессионных тестов.

Пользовательский интерфейс новой системы управления проектами должен быть интуитивен и лаконичен т.е. состоять из минимального набора элементов необходимых для выполнения основного сценария использования. Это сохранит фокус внимания пользователя на важных аспектах взаимодействия и позволит избежать ошибок. Однако базовые принципы работы следует унаследовать от старой системы или любой другой популярной системы управления проектами т.к. в них прослеживается схожесть и большая часть пользователей уже имеет опыт работы в подобных системах. Если выполнение стандартных процедур в новой системе будет отличаться от привычного пользователям, то это обязательно вызовет отторжение, сотрудники будут просто игнорировать эти процедуры.

Новая система управления проектами должна быть качественно спроектирована с учётом инкрементальной модели разработки. Требуется выделить фундаментальные сущности логической модели данных и их программные интерфейсы, как неизменяемые в будущих версиях системы. Это поможет избежать трудозатрат при доработках системы. При выполнении этих доработок в будущем следует добавлять новые (менее значимые) сущности данных и их программные интерфейсы, а фундаментальные оставлять в неизменном виде.

Таким образом приведённые модели новой технологии демонстрируют положительные изменения в выполнении бизнес-процесса, а именно:

- Увеличение удобства использования системы управления проектами
- Ускорение процесса разработки
- Разделение труда по более узким специализациям

- Снижение трудозатрат на переобучение для доработок системы управления проектами

2.2 Информационное обеспечение ИС

Пункт состоит из 4-х подпунктов (2.2.1 2.2.2 2.2.3 2.2.4) Привести определение информационного обеспечения. Показать какую роль оно играет в работе программного продукта.

2.2.1 Логическая модель данных и ее описание

Модель данных — это абстрактное, самодостаточное, логическое определение объектов, операторов и прочих элементов, в совокупности составляющих абстрактную машину доступа к данным, с которой взаимодействует пользователь. Упомянутые объекты позволяют моделировать структуру данных, а операторы — поведение данных. Используя это определение, можно эффективно разделить понятия модели данных и её реализации. Реализация (implementation) заданной модели данных — это физическое воплощение на реальной машине компонентов абстрактной машины, которые в совокупности составляют эту модель. [2.1 С 57]

Логическая модель базы данных необходима для понимания, как циркулируют данные в информационной системе. Эта модель строится проектировщиком, а используется разработчиком. Существует несколько нотаций для построения логической модели данных. Наиболее распространены нотации Питера Чена, Crows Foot, IDEF1X.

Выбрать одну (IDEF1X или Crows Foot) и обосновать. Построить саму модель в выбранной нотации. Все отношения должны быть нормализованы до 3-й нормальной формы. Написать о том, что была проведена работа по нормализации и учтены фундаментальные свойства отношений. Детально описать логическую модель, можно в виде таблицы (название сущности, атрибуты, первичный ключ) Описать связи между сущностями. Связь "многие ко многим" может присутствовать в логической модели, но в физической модели недопустима т.е. её надо стараться избегать (через добавление промежуточных таблиц) Не должно быть атрибутов, которые не будут использоваться в ИС т.е. добавлять минимум, только необходимые.

#2.2.2 Используемые классификаторы и система кодирования Входная и выходная информация может состоять из различных классификаторов и кодов. Их и требуется описать в данном пункте. Есть презентация `Prezentatsia_klassifikatsia_i_kodirovanie.pptx` Сделать всё по аналогии с `Klassifikatory_i_kodirovanie.docx`

#2.2.3 Нормативно-справочная и входная информация Дать определение нормативно-справочной и входной информации, показать разницу. Описать, что является нормативно-справочной, а что входной информацией во новой ИС. Какие цели реализуются с помощью справочной информации (документы, справки и т.д.). Показать макет экранной формы справочника (не заполненные поля без данных) Показать макет экранной формы входной информации (не заполненные поля без данных) Подробно описать входную информацию и плавно перейти к тому, как она будет преобразована выходную информацию.

#2.2.4 Результатная информация Описать, что является результатной информацией во новой ИС. Отчёты, таблицы, документы, графики, экранные формы показать в виде макетов без данных. Должна быть чёткая логическая связь с пунктом 2.2.3

#2.3 Математическое обеспечение ИС Дать определение математического обеспечения ИС с учётом понятия обеспечивающих подсистем. Перечислить и описать все математические формулы, по которым проходят вычисления в ИС. Не должно быть формул, которые не будут использоваться в ИС. Использовать международный стандарт для оформления формул (файл 1200088826.pdf)

#2.4 Техническое обеспечение ИС Определения технического обеспечения с ссылкой на источник. Прототип ИС представляет собой клиент-серверное веб-приложение. В качестве языка программирования был выбран Javascript, как для реализации сервера, так и для клиента. Код клиента состоит из дерева React компонентов и Material UI. Сервер работает на Node JS. Используется Веб-фреймворк Express. В качестве СУБД выбран PostgreSQL. Деплой выполняется путём создания docker образа и запуска его в контейнере на сервере компании.

Выделить аппаратный комплекс, который необходим для работы прототипа ИС. Можно указать конкретные марки/бренды оборудования, пояснить выбор. Построить диаграмму IT-инфраструктуры.

=====

[2.1] К. Дж. Дейт. Введение в системы баз данных

В этом разделе повествование идёт о проектировании новой модели бизнес-процесса и приложения для него. Веб-приложение реализующее: signup, login, user-role, user-profile, issue-tracking (new-project, project-list, new-task, task-list, view-task, reports)

В прототипе будет продемонстрирован базовый сценарий использования для всех ролей пользователей. Помимо базового сценария могу сделать реализацию какого-нибудь специфического требования конкретной фирмы, чтобы показать гибкость прототипа.

Разрабатываемый прототип информационной системы, в случае успешной реализации, будет опубликован в учебном репозитории под открытой лицензией т.е. будущие выпускники смогут переиспользовать его пользовательские интерфейсы для реализации других информационных систем.