

УДК 338.012

В.В. Милоков, канд. техн. наук, доцент

Ю.В. Сосновский, канд. техн. наук, доцент

П.А. Кухнин, магистр

Таврический национальный университет им. В.И. Вернадского.

Просп. Вернадского 4, г. Симферополь, 95000.

E-mail: mailsender256@mail.ru, victor.milyukov@tnu.edu.ua

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ВЗАИМОДЕЙСТВИЯ КОМПЛЕКСА КЛИЕНТ-РАЗРАБОТЧИК С ПОМОЩЬЮ СИСТЕМЫ АВТОМАТИЗАЦИИ УПРАВЛЕНИЯ ПРОЕКТОМ

Рассмотрен вопрос создания нового программного инструмента для автоматизации взаимодействия клиент-разработчик программного обеспечения. Предлагаются дополнения к существующим концепциям, позволяющие достигнуть поставленных целей.

Ключевые слова: CRM-системы, взаимодействие клиент-заказчик, управление проектом, программирование

В современном практическом программировании управление проектом – область деятельности, в ходе которой определяются и достигаются определенные цели проекта при балансировании между объемом работ, ресурсами, временем исполнения, качеством и рисками. Ключевым фактором успеха проектного управления является наличие четкого плана, минимизации рисков и отклонений от него, а также эффективное управление изменениями.

Управление проектами, программами и портфелями проектов имеет свою специфику и требует наличия у менеджеров и участников проектных команд специальных навыков и знаний [1]. Общий подход к минимизации и уклонению от типовых проблем реализации проектов заключается в организации формализованной системы управления, учитывающей интересы различных участников процесса, и в первую очередь – заказчика.

В качестве основных принципов управления реализацией проектов и программ развития называют следующие [2]:

- разработка единого, структурированного представления проекта и четкое описание его основных элементов;
- построение команды и адекватной организационной структуры управления проектом, соответствующей структуре проекта и условиям его реализации;
- обеспечение функционирования процессов управления проектом, взаимодействия участников;
- разработка единого, структурированного представления проекта и четкое описание его основных элементов.

Разработка структурированного представления проекта необходима для обеспечения согласованного понимания всеми участниками целей, масштаба, объемов работ, промежуточных и конечных результатов, обоснованного выделения ресурсов и распределения ответственности, организации процессов планирования и контроля осуществления проекта.

Основной задачей руководителя проекта является создание организационной структуры, обеспечивающей баланс ответственности, полномочий и реальной власти основных участников проекта [3]. Причем, если полномочия команды управления проектом обеспечиваются разработкой и принятием нормативно-регламентных документов, то для фактической реализации властных полномочий необходимо создание функционирующей системы, опирающейся на управленческие процедуры, систему отчетности и контроля, квалифицированных исполнителей.

Вопросы управления проектами поднимаются многими исследователями, которые акцентируют внимание на разработках разносторонних моделей, стандартов, направленных на повышение эффективности работы по управлению проектами.

В настоящее время не существует единого общепринятого определения понятия «проект». Анализ определений этого понятия показывает, что в зарубежной практике методология управления проектами полностью отлична от чисто технической методологии. В реальной жизни существует множество аспектов проекта, которые лежат вне границ технических областей которые, в свою очередь, обеспечивают ощутимый прирост производительности команды программистов.

В основе современных методов управления проектами лежат методики структуризации работ и сетевого планирования. Однако данная концепция не характеризует управляемость как важное свойство системы, где под управляемостью понимается способность к подчинению целенаправленным воздействиям, обеспечивающим устойчивость функционирования, сохранения или приобретения системы тех или иных качественных особенностей, выполнения заданной программы действий [4].

Следовательно, система управления проектами – организационно-технологический комплекс методических, технических, программных и информационных средств, направленный на поддержку и повышение эффективности процессов планирования и управления проектами, в основе которого лежит программное обеспечение календарного планирования.

В анализ систем управления проектами включены основные и наиболее известные, такие как «JIRA», «Redmine», «OTRS». Остальные рассмотренные системы применяются значительно реже.

Система «JIRA», продукт компании «Atlassian», предназначенный для организации процесса контроля запросов и задач. Система обладает широкими возможностями обработки заданий, но в ней отсутствует возможность организации спринтов, автоматизации повторной обработки заданий, прикрепления к пользователям групп заданий. Несмотря на возможность связи с системами хранения исходных кодов, отсутствует его анализатор.

Данная система интегрируется со средой хранения документации «Confluence», разработанной той же компанией. Эта система предоставляет не только возможности по хранению документации, но и позволяет писать собственную. В целом, «JIRA» чаще используется как система обработки заданий или как система отслеживания ошибок, так как функции, относящиеся к управлению проектами и коммуникации с клиентами не представлены.

Открытое серверное веб-приложение для управления проектами и задачами «Redmine» создано с помощью известного веб-фреймворка «Ruby on Rails». «Redmine» обладает не достаточно продуманным интерфейсом, в котором не учтены современные требования к т.н. «юзабилити» программы. Проект обладает достаточными возможностями в части отслеживания и обработки заданий, однако, отсутствие спринтов и слабые возможности по планированию делают затруднительным использование «Redmine» в роли системы управления проектом. Документация хранится как отдельные файлы, нет возможности ее написания или редактирования непосредственно из проекта. Несмотря на хорошую интеграцию с системами хранения исходных кодов, отсутствует возможность планировать и отслеживать тестирование приложений непосредственно из системы.

«OTRS» – открытая производительная система обработки задач и заявок с динамической сменой очередей и удобны распределением ролей пользователей, которые могут изменяться и настраиваться при необходимости. В тоже время, это сложная система, требующая длительной настройки, крайне сложно интегрируемая в процесс разработки программных продуктов в связи с отсутствием возможностей по внедрению как систем хранения исходных кодов, так и систем тестирования. Предоставляемого ей функционала достаточно на обеспечение технической поддержки, однако для полноценного управления проектом она не применяется.

В анализ были включены и другие системы управления проектом. Однако, основной проблемой существующих систем управления проектом является отсутствие возможности осуществлять настройку и управление ролями пользователей и клиентов, невозможность определять вид и тип предоставляемой информации и, как следствие, невозможность введения клиента в процесс планирования и разработки проекта, так как в одном случае он одновременно находится в избытке нежелательной информации и, при этом, в недостатке необходимой.

Цель работы – повысить эффективность взаимодействия заказчика с разработчиком за счет внедрения новых элементов в систему управления проектом.

Для достижения поставленной цели необходимо модифицировать применяемый при проектировании систем управления проектом подход и обеспечить функционал системы, удовлетворяющий критериям удобства использования и интеграции с IT сервисами, а также обладающий возможностью широкой настройки ролей пользователей. Подобная система управления проектом поможет избежать сложностей при планировании, выработать четкую стратегию, своевременно известить клиента и предоставить ему информацию о прогрессе работы и достигнуть желаемого результата. Структурная схема предлагаемой системы показана на рисунке 1.

Следует отметить основное отличие предлагаемой концепции от существующих на сегодняшний день – гибкая система создания и контроля ролей пользователей, а также расширение возможностей по предоставлению информации клиенту о процессе и результатах разработки.

Таким образом, клиент будет введен в производственный процесс, а разработчик, в свою очередь, может определять объем и детализацию информации, предоставляемой клиенту. При этом важным элементом является выполнение требований т.н. «юзабилити» для команды разработчика, так как, несмотря на сближение клиента с процессом разработки, основную часть времени с данным продуктом будут работать сотрудники предприятия. Проект должен взаимодействовать с распространенной системой хранения исходных кодов «Git» и с системой непрерывной интеграции «Jenkins». Глава предприятия сможет распределять задания между сотрудниками, отслеживать их выполнение, прогнозировать и планировать последующие изменения при помощи графиков и данных о выполненной работе.

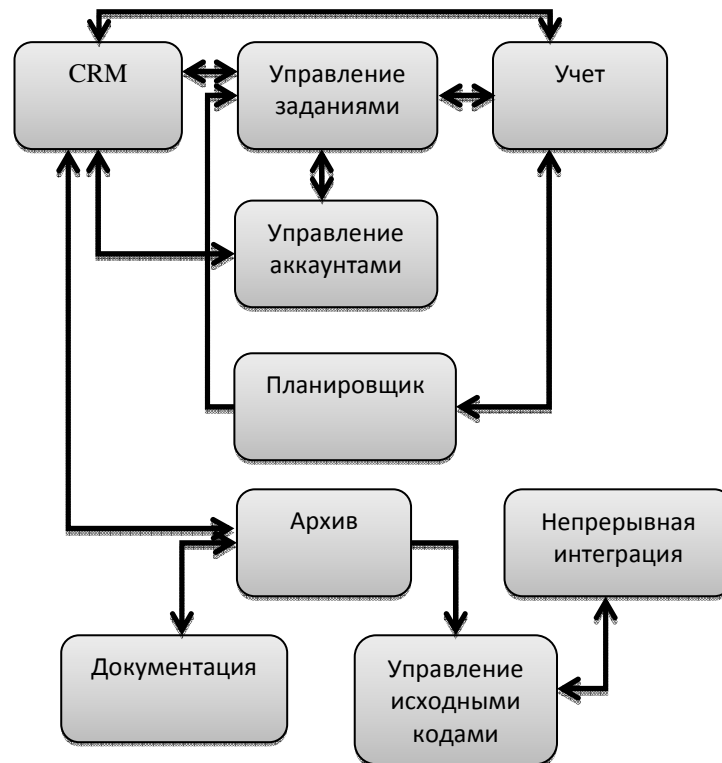


Рисунок 1 – Структурная схема системы

Система управления проектом, основанная на представленной концепции, должна содержать 3 основные подсистемы: подсистему управления взаимоотношениями с клиентами (CRM-система), подсистему отслеживания заданий, и подсистему архивов.

Подсистема CRM отвечает за взаимодействие с пользователем и управляет такими подсистемами как подсистема управления учетными записями и подсистема учета. В ее функционал входит предоставление пользователю, в том числе и администраторам графической информации о данных находящихся в системе, она отвечает за аутентификацию, авторизацию и обеспечивает подспорье для коммуникации различных ролей пользователей.

Главной задачей системы CRM является предоставление пользователю инструментария для общения с другими пользователями, управления их учетными записями или ролями, а также предоставление графической информации о процессах, протекающих внутри приложения. Доступ к истории активности пользователя, а также документации по проектам и исходным кодам приложений осуществляется через подсистему архивов. С ней взаимодействует подсистема CRM для получения необходимых данных и вывода их пользователю.

Подсистема отслеживания заданий является ядром системы. Все взаимодействие и передача данных проводится через ее модули. Посредством взаимодействия подсистемы отслеживания заданий, подсистемы учета и планировщика производится процесс инициализации, планирования и создания проекта. Структура классов подсистем учета, отслеживания заданий и планирования представлена на рисунке 2.

В проекте пользователи могут быть как создателями/владельцами проекта, так и его участниками. За счет внедренной системы настройки ролей их участие может быть как непосредственным, так и прозрачным.

В подсистеме обработки заданий, основным рабочим элементом является задание. Они создаются в ходе разработки, выявления ошибок и появления новых требований. Заданиям может быть выставлен тип и приоритет. В систему встроены базовые наборы типов и приоритетов заданий, которые могут быть изменены и расширены администратором. Задания могут быть добавлены в спринте, прикреплены к отдельному сотруднику или клиенту, как напоминание о необходимости каких-либо действий. Каждое задание после его завершения может быть повторно открыто и прикреплено. Жизненный цикл задания может быть представлен в виде диаграммы видов деятельности, представленной на рисунке 3.

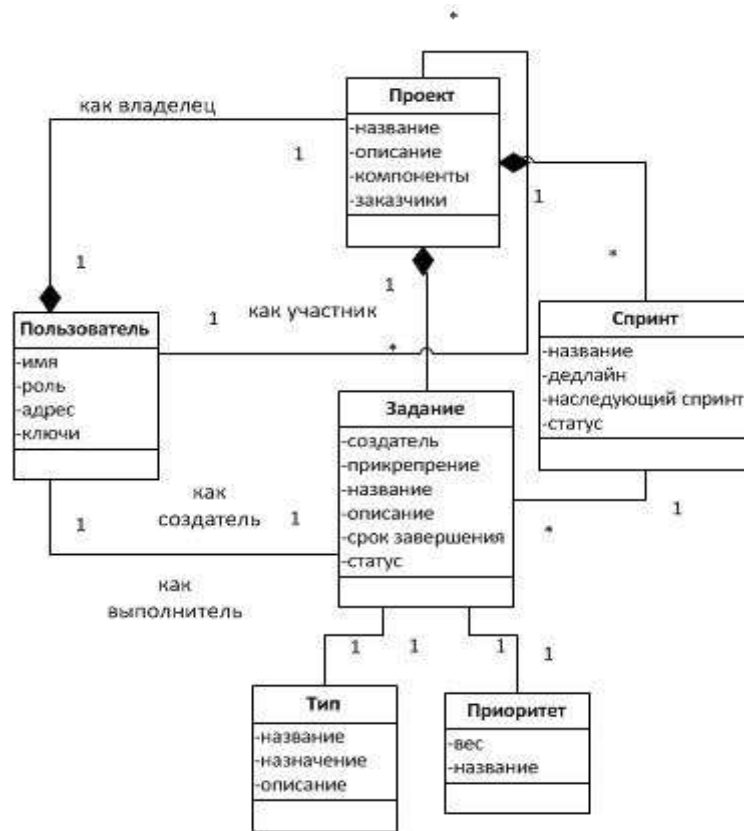


Рисунок 2 – Диаграмма классов подсистем ядра проекта

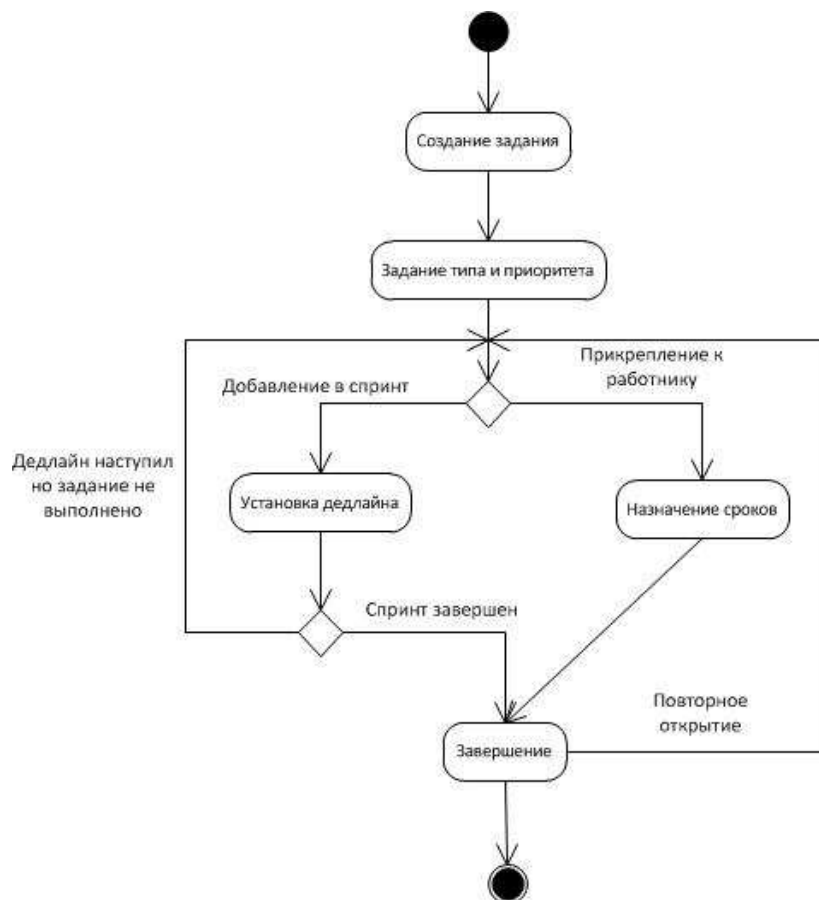


Рисунок 3 – Диаграмма видов деятельности задания

Сотрудники предприятия наибольшее количество времени будут проводить, работая со списками заданий или ошибок работы программы, основанными на данных, присылаемых и обрабатываемых данной подсистемой.

Подсистема учета предназначена для подсчета затраченного времени и оценки эффективности работы через систему CRM и подсистему архивов.

Подсистема архивов отвечает за хранение истории изменений проектов, взаимодействие с системами хранения исходных кодов и системой непрерывной интеграции, для проведения автоматических и мануальных тестов над разрабатываемым проектом.

В разработке приложения, основанного на предлагаемом подходе, были использованы следующие системы:

«Sublime Text» – для написания и верифицирования исходных кодов, а также управления структурой файлов и папок проекта.

Локальный сервер и консоль разработчика «Rails» для отладки и тестирования кода. Пример программного кода в интегрированной среде «Ruby on Rails», описывающий модель пользователя и его отношения с другими классами представлен ниже.

```
class User < ActiveRecord::Base
  devise :database_authenticatable, :registerable, :confirmable, :token_authenticatable,
  :recoverable, :rememberable, :trackable
  ## Validations
  with_options presence: true do |pt|
    pt.validates :first_name
    pt.validates :last_name
    pt.validates :email,format:{with:Devise.email_regexp, allow_blank: true },
    uniqueness: { allow_blank: true }
  end
  validates :password, presence: { on: :create }, confirmation: { if: ->{
  self.password.present? } }, length: { within: Devise.password_length, allow_blank: true }
  attr_accessible :name, :username, :uid, :email, :token, :gravatar_id, :notification_email
  ## Relationships
  has_many :activities, as: :creator
  has_many :roles
  has_many :permissions, :through => :roles
  has_many :projects
  has_many :team_members
  has_many :teams, :through => :team_members
  has_many :owned_projects, :through => :teams, :class_name => "Project", :source =>
  :project, :conditions => { :teams => { :admin_privileges => true } }
  has_many :accessible_projects, :through => :teams, :class_name => "Project", :source =>
  :project
  has_many :dashboards
  has_many :issues, :through => :projects
  has_many :reported_issues, :class_name => "Issue", :foreign_key => :reported_by_id
  after_initialize :set_defaults
```

Подсистема отслеживания заданий имеет следующую функциональность: создание и обработка проектов, заданий, спринтов и т.н. «бэклог». Создание задания производится через соответствующий диалог, каждое задание прикрепляется к определенному проекту. Однако, если несколько проектов содержат схожие задания – они могут быть клонированы из одного проекта в другой с внесением, если необходимо, изменений.

При создании задания необходимо указать его тип и приоритет. Несмотря на то что, проект обладает встроенным набором типов задания и пятью уровнями их приоритетов, администратор может добавлять собственные типы заданий и уровни приоритетов. Более того, он может изменять или удалять как встроенные типы заданий, так и приоритеты. Также есть возможность указать предположительные сроки или же предположительное время, которое потребуется на выполнение данного задания, которое будет учтено при его прикреплении к разработчику. На основании этого появляется возможность прогнозировать сроки завершения проектов и оценивать эффективность работы отдельных разработчиков.

Как правило, процесс разработки разделяется на некоторую последовательность итераций, каждая из которых закрепляется за определенным разработчиком (спринт). Создание, управление и удаление

спринтов реалізовано через т.н. «контроллер». В спринтах є можливість назначення дат початку і кінця, поля спринтів мають функцію вбудованого редагування.

Пользователю може бути надана інформація про прогрес роботи і співвідношенні типів завдань у графічному вигляді. Це діаграма співвідношенні типів завдань і діаграма згорання задач. Приклад генерації подібної діаграми в створеній системі показаний на рисунку 4.

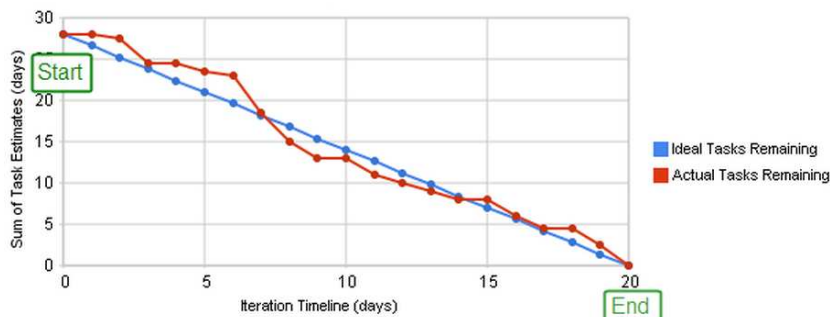


Рисунок 4 – Діаграма згорання задач

Реалізація діаграми згорання задач на мові «Ruby» вивнесена в окремий модуль, для зручності використання і відображення користувачеві. Обсяг друкованої роботи не дозволяє показати всі графічні і управлінські елементи програми, однак функціонал повністю відповідає поставленим завданням.

Об'єднання всього необхідного функціоналу в одному програмному пакеті служить досягненню мети створення багатофункціональної системи управління проектом з гнучкою налаштуванням ролей користувачів і систем обслуговування завдань, що має власну систему зберігання документації і що має інтеграцію з системою зберігання вихідних кодів. Використання подібної системи, в свою чергу, дозволяє підвищити ефективність взаємодії клієнт-розробник, що, в свою чергу, прискорює процес створення, тестування і впровадження програмного продукту. Використання фреймворку «Ruby on Rails» спрощує подальше супроводження застосунку, а також його налагодку і інтеграцію з зовнішніми API.

Реалізація у вигляді веб-застосунку вирішує проблему з мультиплатформенністю, т.к. будь-яка операційна система, що має браузер і доступ до інтернету може використовувати цю систему.

Інтеграція з системою зберігання версій «Git», як з найсучаснішою системою контролю і верифікації вихідного коду, задовольнить потреби різних за розміром підприємств. Крім того в зв'язі з гнучкістю платформи є можливість, при незначальних змінах, інтегрувати з необхідною підприємству системою зберігання вихідного коду.

Подібна система дозволить ефективно розподіляти матеріальні ресурси при розробці проекту, встановити паралелі або взаємозв'язки між проектами, а зберігання історії дозволить врахувати проблеми при плануванні майбутніх проектів.

Реалізація концепції динамічних ролей дозволить розробнику впровадити замовника в процеси планування, розробки, а, можливо, і переробки проекту. В той же час, проект дозволяє об'єднати персонал підприємства єдиною програмою, яку, в цілях розробки можна використовувати як систему відстеження помилок, що створює базу для комунікації замовника з розробником.

Описаний підхід до підвищення ефективності взаємодії «клієнт-розробник», а також створене програмне забезпечення впроваджене на підприємстві, що виконує ряд великих проектів з створення програмних систем. Виконана повна автоматична верифікація програмного коду відповідно до сучасними стандартами.

Кількісна оцінка ефективності впровадження даного підходу і програмного забезпечення ускладнена через відсутність впроваджених кількісних показників роботи розробників. Однак, згідно зібраної статистики за півроку, кількість електронних повідомлень з переписки «клієнт-розробник» зменшилася на 17%, а кількість повідомлень з переписки «керівник проекту-клієнт» зменшилася на 25%. Дані результати свідчать про досягненні поставленої мети і сприяють підвищенню ефективності роботи підприємства.

Перспективами подальших досліджень є розробка кількісних критеріїв оцінки ефективності систем управління проектами і роботи розробників в залежності від ступеня впровадження замовника в роботу над програмним проектом.

Библиографический список использованной литературы

1. Управление проектами: учеб. пособие для вузов/ И.И. Мазур, В.Д. Шапиро, Н.Г. Ольдерогге, А.В. Полковников; под общ. ред. И.И. Мазура. — М.: Экономика, 2009. — 960 с.
2. Кузнецов А.А. Процессное управление проектами на предприятии / А.А. Кузнецов // Менеджмент сегодня. — 2011. — № 4. — С. 206–212.
3. Руководство к Своду знаний по управлению проектами (Руководство PMBOK). — 4-е изд., 2008. — Project Management Institute, Four Campus Boulevard, Newtown Square, PA 19073-3299 USA / США ANSI/PMI 99-001-2008
4. Система управления проектами [Электронный ресурс]. — Электрон. текстовые и граф. дан. (118365 bytes). — Режим доступа: <http://www.pmphelp.net/index.php?id=127> Monday, 2 December 2013, 14:02:45.

Поступила в редакцию 13.01.2014 г.

Мілюков В.В., Сосновський Ю.В., Кухнін П.А. Підвищення ефективності взаємодії клієнт-розробник за допомогою системи автоматизації управління проектом

Розглянуто питання створення нового програмного інструменту для автоматизації взаємодії клієнт-розробник програмного забезпечення. Пропонуються доповнення до існуючих концепцій, що дозволяють досягти поставлених цілей.

Ключові слова: CRM-системи, взаємодія клієнт-замовник, управління проектом, програмування.

Milukov V.V., Sosnovskij Y.V., Kuhnin P.A. Automation project management system of a client-developer intercommunication

The question of creating a new software tool to automate customer intercommunication software developer is considered. For renewed concept a set of UML-diagram and a program release are elaborated.

Keywords: CRM-systems, client-developer intercommunication, programming.