Данный файл не является требованиями к учебной практике. Строго следовать не надо, ознакомиться полезно.

* + 1. О технологии работы в проектах

***Далее в тексте встречаются некоторые слова из (полу) профессионального жаргона. Они поясняются в примечаниях в конце текста.***

Общие положения

Студенты выполняют мини-проекты в командах из 3 человек.

Используемые инструменты

Для трекинга (*отслеживания выполнения*) задач используется redmine со стандартным workflow\* (ссылка про redmine: <http://ru.wikipedia.org/wiki/Redmine>, см. также в конце текста).

В качестве системы контроля версий используется svn (Subversion)\*\*, размещаемая каждой командой исполнителей на code.google.com (?).

Итеративная разработка

При разработке проектов, применяются следующие роли:

* *разработчик* – студент, участник команды. Ответственность – выполнять текущие задачи итерации, проектировать и реализовывать классы и алгоритмы, загружать код в репозиторий\*\*\*.
* *ответственный за итерацию* – разработчик, дополнительной обязанностью которого является проверка качества и полноты исполнения задач в трекере и изменение их статуса.
* *куратор* – преподаватель, в задачи которого входит определение состояния исполнения задач проекта на каждой итерации, помощь студентам при решении задач программирования, проверка реальности планов для следующей итерации.

Разработка проектов выполняется итеративно с длиной итерации 1 рабочий день. Каждая команда ежедневно на очной встрече с куратором представляет результат разработки на предыдущей итерации; предлагает и обсуждает план на следующую итерацию; озвучивает текущие проблемы и предпринятые шаги для их решения.

Список проверки готовности итерации к встрече с куратором

Перед встречей с куратором разработчик, ответственный за итерацию проверяет готовность по следующему списку:

* выполненные задачи текущей итерации переведены в состояние “Resolved”
* невыполненные задачи текущей итерации помещены в список проблем для обсуждения
* исходный код «закоммичен»\*\*\*\* в репозиторий;
* состояние головы репозитория может быть выкачано и откомпилировано без ошибок и предупреждений.
* (необязательно, но желательно) Unit-тесты к разработанным классам закоммичены в репозиторий.

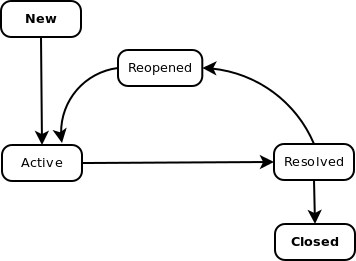
Список проверки для очной встречи

На очной встрече выполняются следующие действия:

* Проверяется готовность кода в репозитории.
* Проверяется завершенность и корректность задач, находящихся в состоянии Resolved. При положительном результате задачи переводятся в состояние Closed, при отрицательном результате - в состояние Reopened.
* Оказывается помощь при решении задач, находящихся в списке проблем
* Задачи на следующую итерацию переводятся в состояние Active. Проверяется реальность их исполнения за итерацию.
* Назначается ответственный за итерацию.

Дополнительные рекомендации

* Любой участник проекта может создавать любые задачи в трекере в состоянии New, но планироваться на исполнение (переводиться в состояние Active) эти задачи могут только на очной встрече
* Задачи в трекере в состоянии Active должны обладать четкой оценкой времени планируемого исполнения не превосходящей 2-3 часа.

Используемый **workflow:**

**Примечания.**

**\*Workflow** - это полная или частичная автоматизация бизнес-процесса, при которой документы, информация или задания передаются от одного участника (бизнес-процесса) к другому для выполнения действий согласно набору руководящих правил.

Workflow - в дословном переводе с английского означает поток работ /операций.

Вот еще одно определение, наилучшим образом отражающее процессную сущность Workflow

**Workflow** - это процесс, произвольное задание, выполняемое последовательно или параллельно двумя или более участниками рабочей группы с целью достижения общей цели.

# \*\*svn (Subversion) - [свободная](http://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) централизованная [*система управления версиями*](http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8), официально выпущенная в [2004 году](http://ru.wikipedia.org/wiki/2004_%D0%B3%D0%BE%D0%B4) компанией [CollabNet](http://ru.wikipedia.org/w/index.php?title=CollabNet&action=edit&redlink=1) Inc (<http://ru.wikipedia.org/wiki/Subversion> )

# Система управления версиями (<http://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8> )

# \*\*\* - Репозиторий (repository - хранилище) (<http://ru.wikipedia.org/wiki/%D0%E5%EF%EE%E7%E8%F2%EE%F0%E8%E9> )

# Иногда встречается форма *репозитарий* (по поводу *репозитарий* или *репозиторий* есть некоторая вялая полемика в интернете)

\*\*\*\* коммитить (Словарь бизнес-сленга)

(от англ. to commit) – **1.** означает подписаться на выполнение сделки на определенных условиях, в определенный срок. **2.** Подтвердить свое участие в мероприятии (относится к деловым переговорам, вечеринкам и т.п.): «Завтра в девять не могу, я уже закомитился на это время».

**Коммит** (от англ. commit - зд. фиксировать) - сохранение, фиксация (в архиве, репозитории и др.) изменений в программном коде.

**Коммитить** - выполнить коммит.

**Коммиттер** - специалист, выполняющий коммиты.

* + 1. Система управления проектами и задачами

Важным этапом обучения работы в команде является предоставление навыков работы с ПО совместной работы и ПО управления проектами. Программное обеспечение такого рода может быть предоставлено студентам на время работы над проектом. На данный момент существует множество систем групповой работы и управления проектами.

Анализ бесплатных систем управления проектами с учетом необходимости их интеграции с системой управления кодом показывает, что для использования в образовательном процессе наиболее функциональны две системы: Redmine [10] и Trac [11].

Система Redmine отличается от Trac более высокой производительностью, способностью работать "из-коробки" на множестве «\*nix» и Windows систем, и что самое главное – ориентирована на мультипроектность. В свою очередь, система Trac хороша для поддержки единичных проектов.

Redmine – это веб-система управления проектами и задачами (в том числе для отслеживания ошибок при разработке ПО). Является свободным программным обеспечением, распространяемым по лицензии GNU GPL.

Основные возможности системы:

* ведение нескольких проектов;
* гибкая система доступа, основанная на ролях;
* система отслеживания ошибок;
* диаграммы Ганта и календарь;
* ведение новостей проекта и документов, управление файлами;
* оповещение об изменениях с помощью RSS-потоков и электронной почты;
* вики для каждого проекта;
* форумы для каждого проекта;
* учёт временных затрат;
* настраиваемые произвольные поля для инцидентов, временных затрат, проектов и пользователей;
* лёгкая интеграция с системами управления версиями кода (SVN, CVS, Git, Mercurial, Bazaar и Darcs);
* создание записей об ошибках на основе полученных писем;
* многоязыковой интерфейс (в том числе русский).

Таким образом, Redmine предоставляет большинство современных средств для групповой работы и управления проектами. Отличительным свойством Redmine является кросс-платформенность и возможность работать на широком спектре Windows и Unix-систем с различными СУБД и почтовыми серверами.

Недостатки Redmine отечаются в [10]:

* Управление файлами и документами в Redmine сводится к их добавлению, удалению и редактированию. Правами доступа ни к файлам, ни к отдельным документам управлять нельзя.
* Отсутствуют оповещения об изменении документов.
* В Redmine нельзя управлять правами доступа на уровне отдельных полей задачи. Например, на данный момент от клиентов нельзя скрыть оценки времени работы над проектом или информацию о потраченном времени.
* В Redmine все дополнительные поля доступны всем пользователям, все участники проекта смогут их видеть и изменять. Это ограничение может привести к сложностям при наличии неоднородной команды, когда доступ к проекту имеют и менеджеры, и разработчики, и клиенты.
* В Redmine нет прав на отдельные типы переходов в workflow. Например, сейчас нельзя указать, что когда кто-то заканчивает исправлять ошибку, он должен выбрать ответственным тестировщика и должен указать номер билда. Также нельзя скрыть внутреннюю переписку между программистами от клиента.
* В Redmine в список задач не выводится общая трудоемкость задач, а в отчетах по трудоемкости нельзя делать отборы, в том числе и по исполнителю.

Отметим, что в контексте учебной деятельности не все из этих недостатков существенны, а некоторые с точки зрения функций преподавателя полезны.

* + 1. Система управления версиями

Важным дополнением к системе групповой работы и управления проектами является система управления кодом или системы управления версиями.

Система управления версиями (Version Control System, VCS или Revision Control System) [12] — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

Такие системы наиболее широко используются при разработке программного обеспечения для хранения исходных кодов разрабатываемой программы. Однако они могут с успехом применяться и в других областях, в которых ведётся работа с большим количеством непрерывно изменяющихся электронных документов. В частности, системы управления версиями применяются в САПР, обычно в составе систем управления данными об изделии. Управление версиями используется в инструментах конфигурационного управления (Software Configuration Management Tools).

Традиционные системы управления версиями используют централизованную модель, когда имеется единое хранилище документов, управляемое специальным сервером, который и выполняет большую часть функций по управлению версиями. Пользователь, работающий с документами, должен сначала получить нужную ему версию документа из хранилища; обычно создаётся локальная копия документа, т. н. «рабочая копия». Может быть получена последняя версия или любая из предыдущих, которая может быть выбрана по номеру версии или дате создания, иногда и по другим признакам. После того, как в документ внесены нужные изменения, новая версия помещается в хранилище. В отличие от простого сохранения файла, предыдущая версия не стирается, а тоже остаётся в хранилище и может быть оттуда получена в любое время.

В качестве системы управления версиями можно использовать SVN (subversion), которая широко применяется в большинстве современных компаний по разработке программных систем. Кроме того SVN достаточно легко интегрируется с Redmine.