

Методы организации
работы в команде
разработчиков. Системы
контроля версий

Все множество разработок в зависимости от количества участников и типов взаимоотношений между ними может быть сведено к триаде разработок.

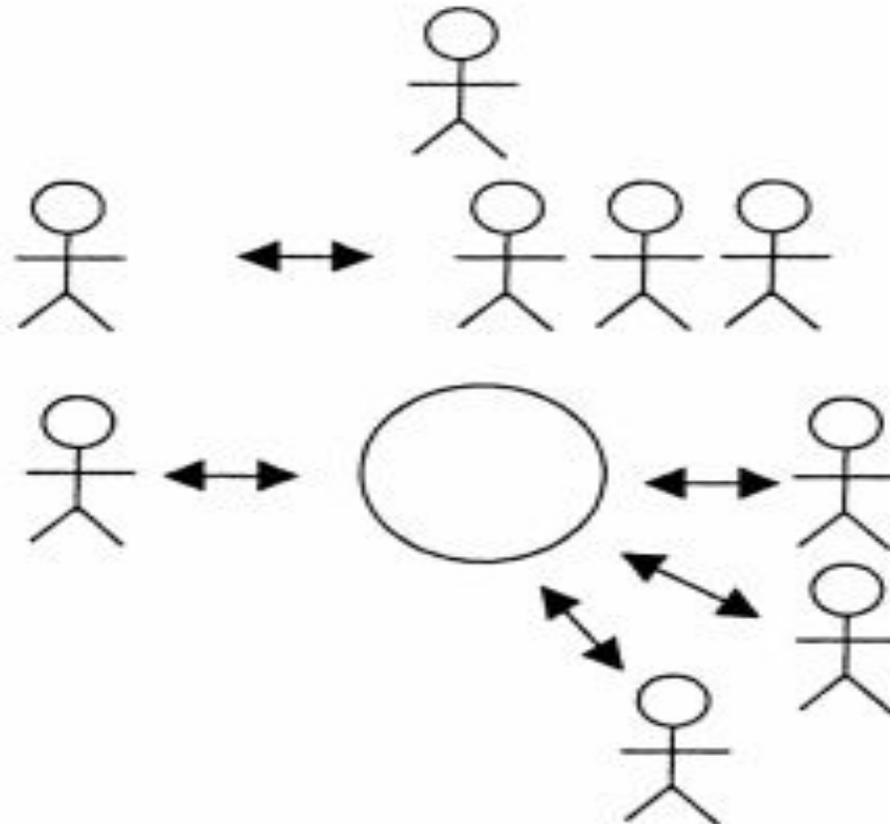
Авторская разработка



Коллективная разработка



Общинная разработка



• 1. Авторская разработка

Авторская разработка - принцип создания программных продуктов, при котором весь жизненный цикл разработки поддерживается одним единственным человеком.

• 2. Коллективная разработка

Одним из основных вопросов коллективной разработки является разделение труда - от равноправных соисполнителей до организации в виде жесткой иерархии (например, бригады главного программиста).

• 3. Общинная модель разработки

Идеология общинной ("базарной") модели разработки сформулирована в программной статье Эрика Раймонда (Eric Raymond) "Собор и Базар". Общинная модель характеризуется тремя основными факторами: децентрализованность разработки, разработка ведется на базе открытых исходных текстов, большое количество внешних тестеров (бета-тестеров), позволяющих быстро обнаруживать ошибки и проблемы в программе.

Коллективная разработка

Бригада равноправных соисполнителей обычно состоит из специалистов, занимающихся примерно подобными задачами в рамках одного проекта. Естественно, специализаций в рамках одной бригады может быть несколько:

- инженеры-разработчики (специалисты по инженерии программирования и программисты);
- технические писатели;
- инженеры тестирования;
- инженеры качества;
- специалисты по сопровождению продукта;
- специалисты по продажам продукта.

Основные этапы разработки программного обеспечения

- Анализ - определение процесса разработки ПО
- Проектирование - управление проектом разработки
- Конструирование - описание целевого программного продукта
- Программирование - проектирование продукта
- Разработка продукта
- Тестирование - тестирование частей программного продукта
- Отладка - интеграция частей и тестирование продукта в целом
- Развертывание - сопровождение продукта, обучение пользователей
- Выпуск продукта

Минимальные функции системы коллективной разработки:

- это регистрация изменений, вносимых в проект,
- хранение файлов проекта.

Система контроля версий

Системы управления версиями (Version Control Systems, VCS) или Системы управления исходным кодом (Source Management Systems, SMS) — важный аспект разработки современного ПО. Это программное обеспечение, предназначенное для работы с постоянно изменяющейся информацией.

VCS предоставляет следующие возможности:

- 1) Поддержка хранения файлов в репозитории.
- 2) Поддержка истории версий файлов в репозитории.
- 3) Нахождение конфликтов при изменении исходного кода и обеспечение синхронизации при работе в многопользовательской среде разработки.
- 4) Отслеживание авторов изменений.

Классификация систем контроля версий

1) *Централизованные/распределённые* — в централизованных системах контроля версий вся работа производится с центральным репозиторием, в распределённых — у каждого разработчика есть локальная копия репозитория.

2) *Блокирующие/не блокирующие* — блокирующие системы контроля версий позволяют наложить запрет на изменение файла, пока один из разработчиков работает над ним, в неблокирующих один файл может одновременно изменяться несколькими разработчиками.

3) *Для текстовых данных/для бинарных данных* — для VCS для текстовых данных очень важна поддержка слияния изменений, для VCS с бинарными данными важна возможность блокировки.

Мониторинг работоспособности некоторых из систем контроля версий.

- **Bazaar**, ранее известная как Bazaar-NG, утилита командной строки bzd, — это распределённая система управления версиями, разработка которой спонсируется фирмой Canonical Ltd, в последнюю версию по сравнению с предыдущей было внесено более 50 изменений. Данная система разработана в целях облегчения создания и развития проектов для пользователей.
- **Mercurial**, в переводе с англ. «подвижный», — распределённая система управления версиями, способная функционировать на многих операционных системах и различных аппаратных платформах, разработанная для эффективной работы с очень большими кодами.
- **Git** — распределённая система управления версиями файлов. Код программы был написан на языке «С», проект создан Линусом Торвальдсом в 2005 году для управления разработкой ядра Linux, является общедоступным программным обеспечением. Данная система была введена многими ведущими разработчиками, используется в известных Linux-сообществу проектах.
- **Concurrent Versions System** (или CVS, в переводе «Система Одновременных Версий») — представляет собой программный продукт, который относится к разряду систем управления версиями. Программа хранит историю изменений исходного кода программного обеспечения, тем самым облегчая совместную работу программистов над одним проектом. CVS популярна в мире открытого программного обеспечения.

- **Bazaar** — удобная система контроля версий с приятным интерфейсом, она хорошо подходит для пользователей, которых не привлекает перспектива работы с командной строкой. Имеется множество дополнительных опций и расширений, что позволяет настроить программу под свои нужды.
- Говоря о **Mercurial** следует отметить, что простой и отточенный интерфейс, и набор команд, возможность импортировать репозитории с других систем контроля версий, — сделают переход на данную программу безболезненным и быстрым, а её надёжность и скорость работы позволяют пользоваться им для контроля версий огромных проектов. Все это позволяет Mercurial стать достойным конкурентом git'а.
- В свою очередь **Git** — это гибкая, удобная и мощная система контроля версий, способная удовлетворить абсолютное большинство пользователей. Git — один из лидеров систем контроля версий.
- Несмотря на то, что программа **CVS** достаточно устарела и обладает весомыми недостатками, она все ещё является одной из самых популярных систем контроля версий и отлично подходит для управления малыми проектами, не требующих создания нескольких параллельных версий, которые надо периодически соединять.