**11.Навіщо потрібні розподілені системи**

Как следует из названия одна из основных идей распределенных систем — это отсутствие четко выделенного центрального хранилища версий - репозитория. В случае распределенных систем набор версий может быть полностью, или частично распределен между различными хранилищами, в том числе и удаленными. Такая модель отлично вписывается в работу распределенных команд, например, распределенной по всему миру команды разработчиков работающих над одним проектом с открытым исходным кодом. Разработчик такой команды может скачать себе всю информацию по версиям и после этого работать только на локальной машине. Как только будет достигнут результат одного из этапов работы, изменения могут быть залиты в один из центральных репозиториев или, опубликованы для просмотра на сайте разработчика, или в почтовой рассылке. Другие участники проекта, в свою очередь, смогут обновить свою копию хранилища версий новыми изменениями, или попробовать опубликованные изменения на отдельной, тестовой ветке разработки. К сожалению, без хорошей организации проекта отсутствие одного центрального хранилища может быть минусом распределенных систем. Если в случае централизованных систем всегда есть один общий репозиторий откуда можно получить последнюю версию проекта, то в случае распределенных систем нужно организационно решить какая из веток проекта будет основной. Почему распределенная система контроля версий может быть интересна кому-то, кто уже использует централизованную систему - такую как Subversion? Любая работа подразумевает принятие решений, и в большинстве случаев необходимо пробовать различные варианты: при работе с системами контроля версий для рассмотрения различных вариантов и работы над большими изменениями служат ветки разработки. И хотя это достаточно естественная концепция, пользоваться ей в Subversion достаточно непросто. Тем более, всё усложняется в случае множественных последовательных объединений с одной ветки на другую — в этом случае нужно безошибочно указывать начальные и конечные версии каждого изменения, чтобы избежать конфликтов и ошибок. Для распределенных систем контроля версий ветки разработки являются одной из основополагающих концепций — в большинстве случаев каждая копия хранилища версий является веткой разработки. Таким образом, механизм объединения изменений с одной ветки на другую в случае распределенных систем является одним из основных, что позволяет пользователям прикладывать меньше усилий при пользовании системой.

Краткое описание популярных распределенных СУВ

* [Git](https://git-scm.com/) - распределенная система контроля версий, разработанная Линусом Торвальдсом. Изначально Git предназначалась для использования в процессе разработки ядра Linux, но позже стала использоваться и во многих других проектах — таких, как, например, X.org и Ruby on Rails, Drupal. На данный момент Git является самой быстрой распределенной системой, использующей самое компактное хранилище ревизий. Но в тоже время для пользователей, переходящих, например, с Subversion интерфейс Git может показаться сложным;
* [Mercurial](https://www.selenic.com/mercurial/) - распределенная система, написанная на языке Python с несколькими расширениями на C. Из использующих Mercurial проектов можно назвать, такие, как, Mozilla и MoinMoin.
* [Bazaar](https://bazaar-vcs.org/) - система разработка которой поддерживается компанией Canonical — известной своими дистрибутивом Ubuntu и сайтом httpss://launchpad.net/. Система в основном написана на языке Python и используется такими проектами, как, например, MySQL.
* [Codeville](http://codeville.org/) - написанная на Python распределенная система использующая инновационный алгоритм объединения изменений (merge). Система используется, например, при разработке оригинального клиента BitTorrent.
* [Darcs](http://darcs.net/) - распределенная система контроля версий написанная на Haskell используемая, например, проектом Buildbot.

[Monotone](https://monotone.ca/) - система написанная на C++ и использующая SQLite как хранилище ревизий.