

Стратегия ветвлений

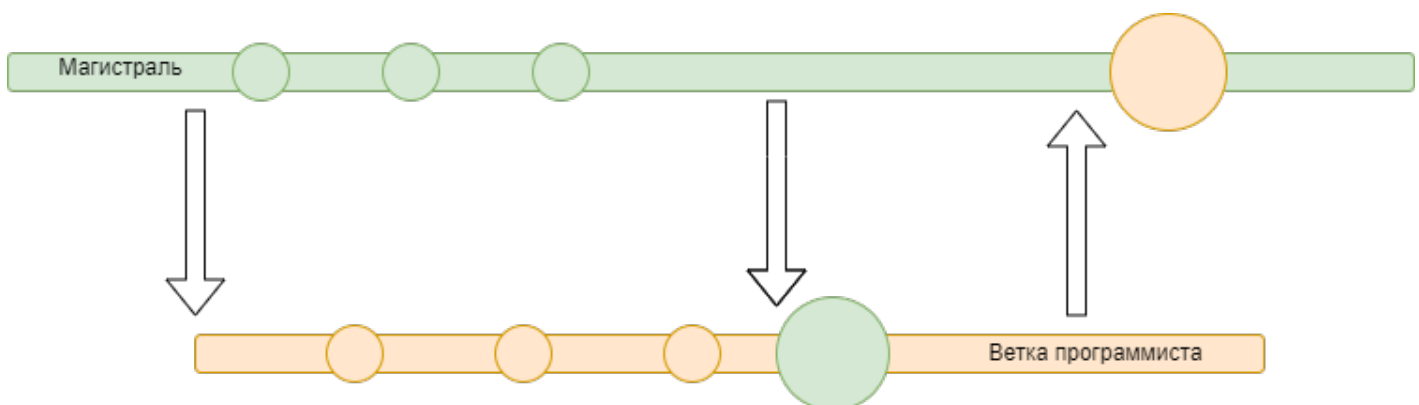
Ветвление - это поиск компромисса между изоляцией и интеграцией. Заставить всех постоянно работать над одной общей базой кода не получится. Разная скорость и качество работы программистов неизбежно вызывают конфликты. Нужно понятие частного рабочего пространства, в котором программист работает над своей задачей. В этом пространстве программист изолирован от других разработчиков. И другие программисты тоже работают в своих частных пространствах, их изменения не подвергают риску весь проект в целом. В какой-то момент потребуется интеграция, т.е. объединение результатов работы программистов. И если СУВ позволяет легко создавать ветви и отслеживать изменения в этих ветвях, то объединение происходит довольно сложно. Не существует общих критериев для автоматического объединения. Конфликтные ситуации решаются в ручном режиме.

Таким образом выбор стратегии ветвления на самом деле сводится к принятию решения о том, как и когда происходит интеграция - как часто и в каком объёме придётся решать конфликты.

Общая схема работы команды выглядит следующим образом:

1. Программист создаёт новую ветку для задачи. Предположим, что источником будет магистраль.
2. Для решения задачи вносит изменения в свою ветку
3. Выполняет тестирование на ветке. Убеждается в работоспособности программы.
4. До переноса изменений из своей ветки в магистраль следует учесть все изменения, произошедшие со времени создания ветки. Т.е. перенести новые коммиты из магистрали в свою ветку.
5. Проверяет влияние изменений (из магистрали) на работу ветки.
6. Переносит изменения из своей ветки в магистраль.

По схеме видно, что сначала выполняется изоляция (разделение на части), затем - интеграция (объединение частей).



Факторы, осложняющие интеграцию

Существуют два взгляда на суть коммита.

1. Коммит - это запись того, что уже произошло. Изменить прошлое нельзя, коммиты не изменяются.
2. Коммит - это история того, как была решена задача. Если задачу будет опубликована только после её решения, то можно изменять коммиты.

Из-за двух взглядов в СУВ имеются две команды для объединения веток.

1. слияние (merge) - слияние берет две конечные точки и сливает их вместе.
2. перебазирование (rebase) - Перебазирование повторяет изменения из одной ветки поверх другой в том порядке, в котором эти изменения были сделаны

Слияние более подвержено конфликтам, но перебазирование можно делать только с ранее неопубликованными коммитами. Иначе при повторном объединении перебазированных коммитов получится путаница.

Частота интеграции влияет на сложность объединения ветвей. Чем чаще выполняется интеграция, тем раньше обнаруживаются и быстрее разрешаются конфликты, почти исключается риск невозможного слияния

Различают следующие конфликтные ситуации в процессе объединения:

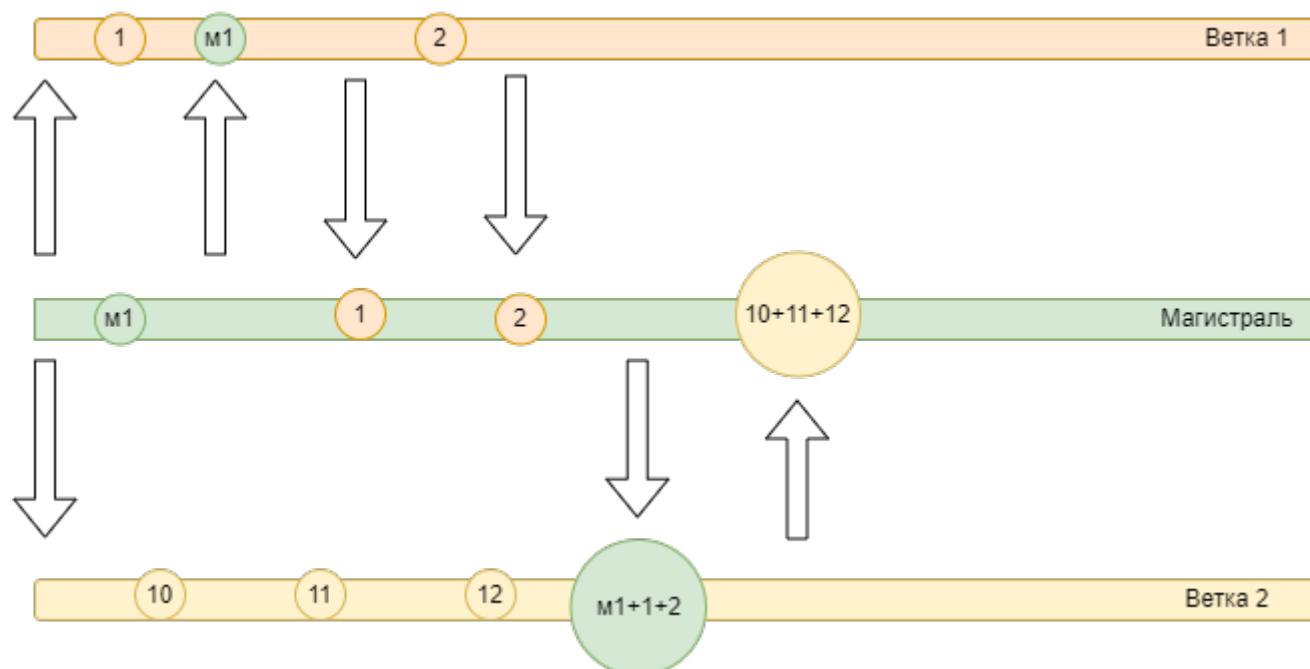
- текстовый конфликт. Например, два программиста исправили одно место. СУВ найдёт и подсветит такие места.
- семантический конфликт. Например, первый программист исправил заголовок функции, а второй вызывает эту функцию в старом виде. СУВ такие места не заметит.

Шаблоны ветвления

В литературе описаны несколько шаблонов (методик), которые позволяют командам эффективно использовать ветвление. В их основе лежат две схемы и их комбинация:

- непрерывная интеграция с магистралью
- ветвление задач и функций

Условное сравнение объёма переносимых изменений при непрерывной интеграции (ветка 1) и ветвлении по задачам (ветка 2).



Версия #9

[Демонов Сергей](#) создал Wed, Jan 31, 2024 9:58 AM

[Демонов Сергей](#) обновил Tue, Mar 19, 2024 9:30 AM