

Парное программирование

В рамках лабораторной работы был опробован стиль Пинг-понг, а стиль Ведущий-ведомый применялся во время работы над проектом интуитивно.

Стиль Пинг-понг применялся для реализации нового модуля нашего проекта. Для того чтобы внедрить его вместо Ведущий-ведомый, пришлось целиком менять порядок разработки. Таким образом Пинг-понг сильно замедлил процесс разработки, а позволить себе это в ситуации с жесткими дедлайнами мы не могли.

А вот стиль Ведущий-ведомый применялся куда успешнее. Правда он применялся с некоторыми поправками, так же в короткие сроки нужно было выполнить задачу, поэтому код писался не только ведомым, но и ведущим, в основном в тех случаях, когда нужно было поправить что-то уже написанное.

Стиль Ведущий-ведомый кажется более удобным и простым потому что:

- Ситуация когда в команде есть ведущий работник, у которого больше опыта и знаний намного популярнее, чем ситуация, когда такого человека нет.
- Скорость разработки однозначно выше, по крайней мере на первых парах. Конечно, если четко спланировать процесс разработки и постоянно соблюдать и оптимизировать график, то остальные приемы не будут проигрывать по скорости. Однако по началу преимущество в скорости есть, т. к. Ведущий-ведомый это естественное положение в команде.
- Всегда найдется материал для работы. Если в Пинг-понг приходится ждать юнит-тестов, то в Ведущий может занять себя анализом уже написанного кода. Уже написанного кода, которому нужен рефакторинг обычно больше, чем юнит-тестов без реализованного функционала. Таким образом зависимость членов команды друг от друга значительно ниже, что ускорит процесс разработки, по крайней мере в малых проектах вроде нашего.
- Постоянный обмен опытом. Ведущий может наблюдать за действиями ведомого и постоянно давать советы не только по алгоритмизации и программированию, но и так же по более эффективному использованию инструментов разработки, или посоветовать читать одни источники информации, и не читать другие.

Т.к. Ведущий-ведомый использовался довольно долго в процессе разработки, то пункт о том, что не получилось можно обобщить. Ведущий главный, а ведомый нет. Поэтому ведущий может быть несправедливо уличен в том, что он бездельник. Или ведущий может быть справедливо уличен в том, что он бездельник. Эти проблемы можно решить верно мотивируя членов команды, но это за рамками синтетического проекта по ТРиТПО.

Проблемы Пинг-понг, описанные выше можно решить жестким и целенаправленным внедрением этого стиля разработки. Мы же понимали, что это всего лишь для пробы, поэтому стиль не получил нужной поддержки.

Стиль Ведущий-ведомый будем и дальше использовать во время разработки. В зависимости от сложности и специфики задачи этот стиль будет использоваться с разными оговорками и поправками.