Яндекс

Менеджмент кода, или почему SCM

Сергей Сергеев
руководитель группы разработки интерфейсов

Я.Субботник, Киев, 27 апреля 2013 года

Сегодня
я побуду
«капитаном
очевидность»

Git

> *Git — the stupid content tracker*
>
> ***Linus Torvalds***

SCM — source code management

Регламент

А зачем нам регламент?

- **ЗНАЕМ!** что происходит
- понимаем когда релизим
- понимаем где брать свежий код
- понимаем куда изменения «вливать» и откуда «отпочковывать»

Git-Flow

Git-Flow — наш выбор

- как вести разработку и не бояться «закопаться» в процедуре формирования релиза
- как и когда формировать релиз
- как внешнему (для проекта) пользователю понять где брать стабильный релиз и релиз более старой версии
- как делать hotfix'ы тогда, когда основная ветвь разработки ушла уже далеко

GitHub

GitHub и pull-request'ы

- pull-request для ревью
- один pull-request на одну задачу
- причёсываем историю до подачи pull-request'а
- простое правило — `dev` всегда стабилен!

«Гигиена» коммитов

Базовые правила

- коммитим часто
- коммитим атомарно
- подробно описываем коммит
- помним, что это git

Анти-паттерны

Анти-паттерны регламента

- «меня попросили влить и я влил как есть»
- «нам нужно было быстро, поэтому мы никому не рассказали что…»
- «отдали всем одну ветку, а баги фиксили в другой»
- «я сосквошил всё в один коммит, т.к. это одна задача»
- «мне для этой задачи понадобилось поправить API…»

Журнал Капитана

```
abc2390 багфикс 2 [captain]
eec5401 багфикс [captain]
53aeex5 добавил файлы [captain]
04012ea минорные изменения [captain]
cade034 убрал ненужное [captain]
```

Анти-паттерны git'a

- `git push --force`
- `git rebase [ -i ] public-branch`

Вопросы?

Сергей Сергеев
руководитель группы разработки интерфейсов

gurugray@yandex-team.ru

@gurugray