

## Code Ownership

We do not practice code ownership. When the classes for some feature are first developed, only one team will typically work on them, during the one iteration it takes to develop them initially.

Thereafter, whenever any team has occasion to work with a class, whether to extend it or repair it, they do so freely. Because we release frequently, there are few conflicts. We have extended ENVY to warn us when we version a class that is not based on the released, and when it does, we integrate our changes into the released version if that isn't the one we started with. (The team who released ahead of us will help us if we need it.)

**On the contrary**, lack of ownership might cause people to feel less responsible for their work.

- Since the entire group will soon see our work, we become more aware of quality concerns, not less.

**On the contrary**, the non-owner might not understand the class and break it.

- Since the class must run its unit tests at 100%, changes made by anyone are automatically subject to the quality requirements.
- Pair programming ensures that even if the team changing the class has no experience with it, two pairs of eyes look at the changes.
- Commonly, if the class is complex, one of the original authors is invited to help with changes.

---

[\[ Home \]](#) [\[ XP \]](#) [\[ XP Practices Frame \]](#)

© 1997, 1998, Ronald E Jeffries  
ronjeffries@acm.org  
<http://www.armaties.com>