Don't Go Dark

A few times during the project, a team took on a task and did not complete it within the iteration. They wanted to complete the task and kept it for the next iteration. This generally turned out to be a mistake.

If the estimates for a task are so wrong that it doesn't complete within an iteration, it is likely that something isn't understood about the task. The manager should detect this before the iteration ends in most cases.

The problem is as likely outside the task than inside. Quite possibly the team is trying to solve a problem that is too hard because of something wrong elsewhere in the system. Consider getting a larger group together to do CRC cards on the problem.

~~In the next iteration, change out at least one of the team members.~~

On the contrary, the existing team has more experience with the problem, and they are probably close to cracking it. You should let them go ahead.

- Experience on this project, and on many others, clearly indicates otherwise: if a task isn't going well, it's probably on the wrong track and needs to be reset. Even if the code finally works, it will continue to be a problem.
- When a solution isn't fitting into the system, either the system is wrong or the solution is wrong. In Smalltalk we have the ability to do substantial refactoring, even late in the project. Find out what is wrong and fix it.

On the contrary, it's demoralizing to be declared to be a failure by having someone else take over your task. You should let them go ahead.

- No one is declaring anyone to be a failure. Just as two sets of eyes make development go faster, bringing in a fresh set of eyes will help a bogged-down task come back on track.

In addition, frequently changing tasks is a way to keep the team fresh. Even though we do not practice code ownership, sometimes developers become particularly enamored of certain objects that they have created. This usually indicates a problem. Switch Teams Around.