# ImplementationCriticisms

This page captures some of the criticisms that are made of ZooKeeper's implementation and the current thinking about them:

## PMD doesn't like ZooKeeper

Currently we are using FindBugs. Is PMD better? Should we switch? Here are the big issues identified by http://berlinbuzzwords.de/sites/berlinbuzzwords.de/files/thomas_koch_zookeeper.pdf

### Cyclomatic complexity and NPath complexity

SendThread.readResponse in ClientCxnx.java has an extremely high NPath complexity, but the code is rather simple. That might be a nice method to start with to see what we can do to improve that metric.

### Assigning an Object to null is a code smell.

### Avoid really long methods.

### A high ratio of statements to labels in a switch statement.

### This class has too many methods.

Note that fixing the previous two items can cause this one to kick in.

### (Class has) Too many fields

Fixing this one and the previous one usually implies adding more classes which also increases complexity.

### Avoid empty catch blocks.

### Avoid really long parameter lists.

## "I'm looking at some code, and I think I can write it nicer. Please accept this patch."

The current policy is that if there is a bug in the code, then great! Let's fix it;
if there isn't a bug but the complex code is making another fix or enhancement more difficult, then great! Let's fix it;
if it makes the code more maintainable/testable, then we need to make sure it objectively does increase maintainability/testability before accepting the change.
if aesthetics are the only things motivating the change, then we would rather live with the ugly code that works.

## Most of the JUnit and CPPUnit tests are not unit tests!

Correct. We use them as generic test harnesses. We would like to get more unit tests, but at the end of the day the most important tests for us are the ones that show the system works. It would be awesome (!!!) if someone makes it their mission to add more unit tests.

## Feature bloat

Should we really be adding more features, such as multi-txn? This is a question we ask everytime someone submits a new feature suggestion that is just a simple addition. Features tend to make the system exponentially more complex, so we need to be careful of them. However, there are some things that are really needed. For example, ZOOKEEPER-22 is desperately needed. It would drastically simplify the use of ZooKeeper. SSL is also on the roadmap and is needed as are other authentication protocols such as kerberos.

## Why don't you accept my cool simple extension to ZooKeeper?

See the above and the Tao of ZooKeeper. We really want to keep ZooKeeper focused on its core mission and are averse to code stability and maintainability risks. If you do have an idea, don't lead with "i really want this and it is simple to add"; try to show that it helps zookeeper in general.

## There is too object synchronization happening between threads

yes, particularly between PrepRP and FinalRP. See https://issues.apache.org/jira/browse/ZOOKEEPER-1092