

## Re: Rule against Circular Dependencies?

**Source:** <http://coding.derkeiler.com/Archive/General/comp.object/2004-04/1184.html>

---

**From:** Robert C. Martin ([unclebob\\_at\\_objectmentor.com](mailto:unclebob_at_objectmentor.com))

**Date:** 04/24/04

Date: Sat, 24 Apr 2004 08:14:16 -0500

On 22 Apr 2004 06:54:33 -0700, [kk\\_oop@yahoo.com](mailto:kk_oop@yahoo.com) (Ken) wrote:

>Hi. I'm currently working on an object oriented design standard for  
>our project. The subject of circular dependencies between classes has  
>come up. My inclination is to have a rule that says they are not  
>allowed. However, I've been hard pressed to see such a rule  
>recommended out in the OO literature.

Why are you writing a OO design standard, if you are unsure of what should be in the standard. Standards are written once you are *\*sure\**. Standards are written by a group of people who are so sure about things that they can all agree. Premature standards cause angst, wailing, gnashing of teeth, wasted effort, and are eventually ignored. Take care.

>  
>Would you recommend such a rule? Could you cite any literature that  
>supports such a rule?

There are some very common heuristics and principles that give advice on this. See "The Acyclic Dependencies Principle" in the paper "Granularity" on the [www.objectmentor.com](http://www.objectmentor.com) website.

<http://www.objectmentor.com/resources/listArticles?key=topic&topic=Design%20Principles>

>Note that any of our rules can be bypassed via official waiver.

Who's going to give a waiver if nobody knows if the rule should be there? Here's a better idea. Work closely together on a project for a period of a few months. Learn from your mistakes. Gather a body of shared knowledge. Then, if you think you still have to, write your standard. Most teams, however, realize that once they have a body of shared knowledge they don't need the standard anymore.

>So I  
>would consider a rule to be necessary if it should hold true 90% of  
>the time--and if violating it will create high risk of runtime failure  
>or cause maintenance nightmares.

comp.object: Re: Rule against Circular Dependencies?

Cyclic dependencies carry a high risk. If you have  $N$  modules in a cycle, then the number of dependencies in that cycle is  $N*(N-1)$  or  $O(N^2)$ . This can cause compile times to grow geometrically, and cause huge maintenance burdens. Cycles are generally bad.

On the other hand, when cycles are very small (i.e. two classes) they have virtually no impact at all ( $2*1 == 2$ ). So the rule is really about large cycles ( $N \gg 2$ ). This brings a lot of judgement into play. Are three enough? Are six too many? A heuristic to provide guidance can be a good thing. A rule that requires a waiver is likely to be more trouble than it's worth.

Rules and waivers imply distrust. Better to work closely together and learn to trust each other. Then you can do away with rules and waivers, and work as a team instead.

-----  
Robert C. Martin (Uncle Bob)  
Object Mentor Inc.  
unclebob @ objectmentor . com  
800-338-6716

"Distinguishing between the author  
and the writing is the essence of civilized debate."  
-- Daniel Parker