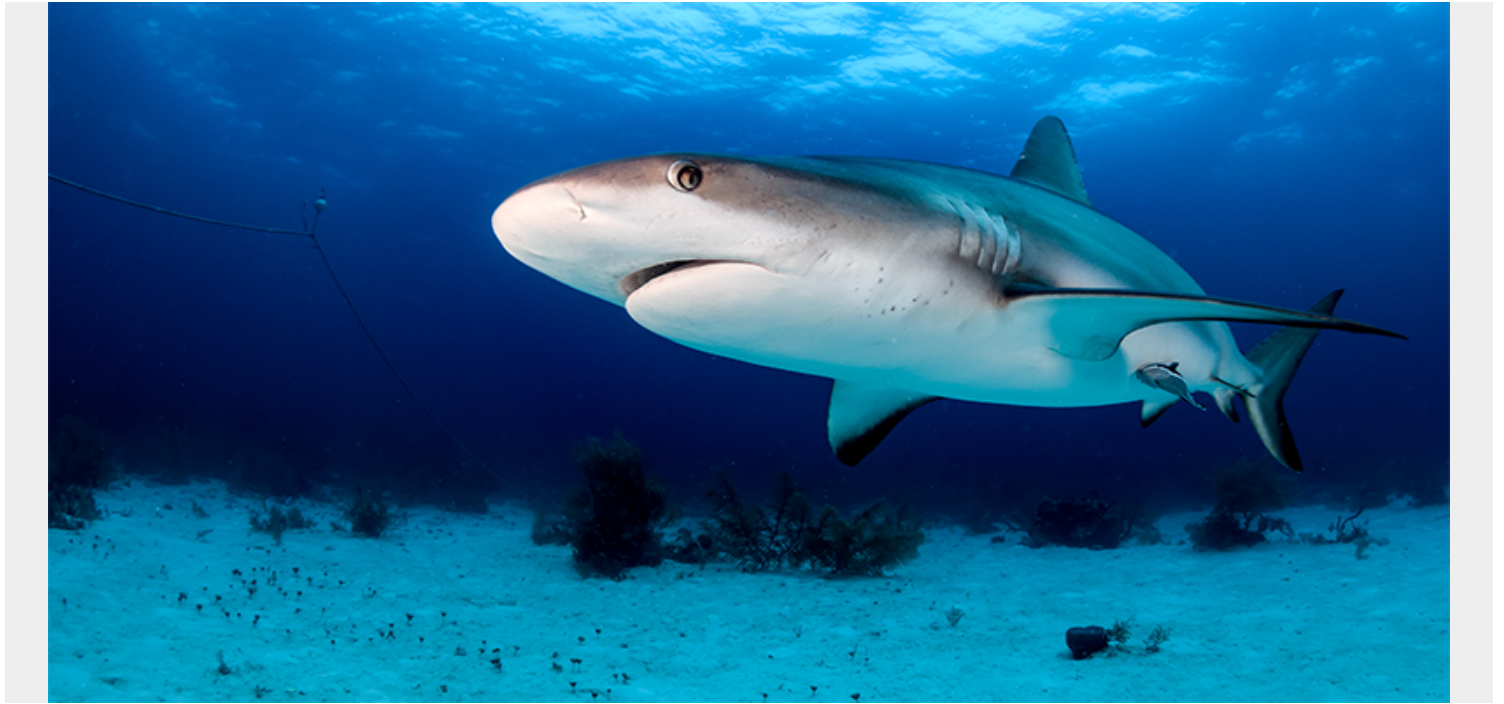


# ARE YOU “JUMPING THE SHARK” WITH THAT NEW SOFTWARE FEATURE?



**Overview: Applications should be modified to stay current and live up to their original mission of solving a particular problem, not solely because they exist and have users. Have you “Jumped the Shark” with your new features?**

You may be familiar with the term “Jumping the Shark.” If you aren’t, let me quickly explain. It is used for television shows that reach a point where they are in a rut, and to break out, they do something that is dramatic, but isn’t authentic. Something crazy the show never really recovers from. It came from a “Happy Days” episode where Fonzie actually jumped a shark... on water skis. Ever since that point long-running shows are watched for when they, too, “jump the shark” and never recover.

This concept is useful I feel in looking at application development. At what point do you have something that is perfectly good? Yes, you can add more, but *should* you? Where is the point you start adding features that don’t provide value?

Here are some questions to ask when considering adding a new feature:

- **Who is asking for this?** We need to know where the request came from, so we have someone to speak with and gain an understanding of the request.
- **What problem does this solve?** Getting an understanding of the need will help us in knowing how best to respond. This leads to three different questions with their own paths:
  - Is it still a problem? It may be that when discussing the problem, you find it is no longer an issue.

- Is there another way to solve the problem? Are there existing features that can be used, if so, then why weren't they used?
- Is this a problem we should solve with the application? This is a very big question, but one that is rarely asked. You can acknowledge that it is a problem, but not one within the scope of this application to solve.
- **How does this affect the user?** Is this a minor, occasional annoyance or a real blocker for them? If it's minor and occasional, we may not want to add a lot to the application for something that is really of low value.
- **What is the total satisfaction with the application?** What percentage of users are being well served by what we have? Will we be adding complexity for the sake of an additional percent or two? Is that worth it?
- **Does the proposed solution fit with our vision for the application,** or is it actually something else we are trying to add on because of convenience? Will it add tech debt?

Applications need to start with a clearly defined of how it will solve an existing problem. They should be modified to stay current and to better serve this mission, not because they exist and have users. Have you "Jumped the Shark" with your new features?