ARE YOU DEVELOPING LIKE SANTA?

When we really think about the project that Santa has, to deliver a year's worth of gifts in one night, it doesn't add up. The project plan doesn't make sense to us. Santa and his team of elves must produce all the toys, store them, load the sleigh, and then deliver them all in one night. It's the ultimate waterfall project, working 364 days to prepare for a giant "release" on one day. This means trying to predict requests, maintaining massive warehouse space, and risking a high degree of delivery errors with no way to correct them. We look at this and find all the flaws, but when we look at our own projects, do we overlook these same flaws? Shouldn't we have some of the same concerns with our own projects?

Use these questions to see if you are release-planning like Santa.

- Requests—Do you have set times for requests and planning? Do you find it difficult to handle last-minute requests? If so, take requests continuously instead. The goal is to never have an occasion where it is the wrong time to make—or take—a request.
- Missing—Do you put so much into a release that you are concerned about missing something on the list? Are there elaborate checks to make sure everything is covered (making the list and checking it twice)? More frequent, smaller releases will be easier to track.
- Long wait—Do you fear that if you miss this release, you will have to wait a long time for the next drop? This leads to trying to jam in more now instead of having a smooth incremental flow. More frequent releases will help you avoid jamming to get improvements, without the long wait.
- Warehouse—Do you produce features that have to be "warehoused" until their scheduled release because they are complete, but you aren't ready to deliver them? Instead of letting them gather dust, plan to get them out sooner and delight your users now, rather than later.

- Making lists—Do you categorize users as naughty or nice? There are no naughty users. They are just users having problems with the application as they understand it. When users have difficulty, we should be looking for the cause of the behavior and what can be done to create a better experience. We can [learn more about usability](#) from the "naughty" users than from the "nice" ones and those lessons improve the product for all.

While we may all admire Santa for his altruism and generosity, we shouldn't emulate his release schedule and all that goes with it. Instead, we should spread out our giving to users throughout the year, and as often as possible.