

## Obama Campaign Failure

I would not call the Obama campaign a failure per se, the software eventually worked and Obama was elected in 2012. The software development did however face significant failures during its build. The 2016 Democratic campaign was the first attempt to bring make the enormous amount of data from each voter file, field offices, analytics people, the website and mobile and make everything available real time. To make this a reality, the team was split into four divisions: Tech (Harper Reed), Digital (Joe Rospar), Analytics (Dan Wagner), and Field (Jeremy Bird). All of these teams had individual goals and were most likely right in trying hard to get their own way. Campaigns are naturally very unique business in that there is no job security, poor pay, and deadlines are non-negotiable. This leads to a structure of freestanding departments that are organized like disaster response with a super hierarchical system for moving quickly. This inevitably led to conflict where each team pushed to achieve their individual goals. One serious failure during the development of Narwhal was the development of Dashboard, an outreach tool was supposed to be rolled out iteratively with additional features. It faced serious issues under Tech team development and Digital/Field teams were agitated to get rid of the Tech team and start over. The main issue was a difference in goals: they all shared the central goal of winning the presidency but established different subgoals. Avoiding such issues requires clear and succinct communication that is respected across teams. Hierarchical teams lead to prioritized goals and does not support a communicative environment. The failures of the Narwhal system could be prevented in the future by implementing an effective method of communication and requirement derivation across teams.

## Patriot Missile Defence System:

During the Gulf War, 28 US soldiers were killed by a Patriot missile defense that erroneously tracked an Iraqi Scud missile. The error was caused by an embedded software issue in the Patriot missile that was present in all the deployed Patriot missiles but only manifested when the weapons control computer was running for several days. The issue boiled down to different parts of the Scud-interception calculations utilizing time in its decimal calculation whereas other parts used the fixed-point binary representation. This wasn't an issue when uptime was low, but became a serious problem when the computer was running for a significant amount of time. It was an incredibly simple fix that was overlooked even after supposed "thousands of hours of testing" with a plethora of different scenarios. Leading up to the tragedy there were reports of inaccuracies after hours of uptime and the software was hurriedly updated to work with the faster Scud missiles. These issues of not listening to user issues and rapidly deploying new changes under stress should be addressed to prevent such a crucial failure. Making sure a system is properly tested in its new environment is crucial, even if it has operated in similar environments before. Bugs often arise from the simplest changes in code and environment. It is equally crucial to listen to user reports of the system and actively make changes to the system to prevent software failures of this magnitude. Developers design for something they expect to see but users apply this tool in situations that actually happen. Feedback is essential to ensure expectations match reality.