



## DEALING WITH DIVERGENCE: When the Business Doesn't Take IT's Advice



## Dealing with Divergence: When the Business Doesn't Take IT's Advice

You've just completed a lengthy requirements gathering phase with business users and are evaluating 3 software solutions. The solutions vary in price, but all 3 meet your "must-have" requirements.


The highest-priced solution offers extensive capabilities which were identified as "nice-to-haves" that would ease multiple pain points. You appreciate the importance of these features and the positive impact they could have on the business.

You present the potential solutions, heavily stressing the benefits of the nice-to-have features: ease of use, visual data reporting, compatibility with third party solutions, and more.

At the end of day, it's the team's decision, and the bottom line is all too influential. They select the cheapest option.

Now what?

You're faced with implementing a solution you might not believe in. While it checks the boxes on minimum requirements, it doesn't offer all the wonderful things some business users will expect with a large-scale undertaking like this.

 [Tweet: What do you do when the business doesn't take your IT solution recommendation? #Abraic https://ctt.ec/Y8Xb5+](https://twitter.com/Abraic/status/888888888888888888)

After a pivotal decision is made by the business, involving key stakeholders and/or a steering committee, there is little to do other than move forward with the hand that is dealt.


The next steps are critical to ensure the software's shortcomings do not become the failures of the implementation.

## Set Expectations and Define Limitations

**First, illustrate the implications of the chosen solution to the business.** Clearly demonstrate the advantages of implementing the chosen software, focusing on the business improvements and how life for the user will change for the better. One technique to accomplish this is to map the new workflows and define the details of how each use case will be handled. This will allow the business to envision how their day-to-day work will be impacted, with any benefits the new system may bring.

**Next, explain what this software package will NOT do,** therefore setting the expectations appropriately. For any missing features or requirements, brainstorm some workarounds to address these

shortcomings. Just because a new system does not replicate prior functionality in its entirety, does not mean that it's flawed. Understanding specific business goals of each user will allow you to tailor the new system or find new ways to achieve the desired results.

 [Tweet: They selected a suboptimal solution? It's time to set expectations. #ProjectManagement #Abraic https://ctt.ec/0rya9+](#)

Transparency upfront will pave the way for understanding down the road when the users are pulled into [User Acceptance Testing](#), and have the chosen system at their fingertips.

The successful rollout of any new system is directly impacted by **how you handle expectations** during the early phases of the project.