blue wren

How to make your software project a success: 20-point checklist	Yes	No
1. Document existing processes Capture current workflows so you know what the software is replacing or improving. *see our blog: How a simple process map can improve your business*		
2. Define what success looks like Set specific outcomes such as reduced errors, faster processing, or increased revenue.		
3. Ensure executive sponsorship Confirm that leadership is committed and will remove obstacles.		
4. Form a cross-functional project team Include people from all departments affected by the change.		
5. Set up project governance Define roles, responsibilities, and decision-making processes. *see our blog: Our 4-step process for software delivery*		
6. Write clear, validated requirements Ensure requirements are complete, agreed upon, and understood.		
7. Engage stakeholders throughout Maintain regular input and feedback from all key groups.		
8. Plan for change management Address resistance before it occurs with communication and training.		
9. Develop realistic test scenarios Ensure tests reflect real use cases and expected outcomes.		
10. Provide targeted training Tailor training to specific user roles and needs.		
11. Plan and test data migration Map, clean, and verify all data before moving to the new system.		
12. Verify system integrations Ensure reliable connections to existing tools and databases.		
13. Identify and manage risks Document risks and mitigation plans early.		
14. Set a realistic timeline and budget Include contingency for unforeseen issues.		
15. Establish a clear communication plan Define who needs to know what, when, and how.		
16. Plan for post-launch support Allocate resources for helpdesk, maintenance, and troubleshooting.		
17. Assign change champions Identify staff who will promote adoption within their teams.		
18. Support continuous improvement Plan for future updates based on user feedback and evolving needs.		
19. Avoid unnecessary complexity Keep solutions as simple as possible to aid adoption and maintenance. *see our blog: What is MVP?*		
20. Build in flexibility Allow for adjustments as understanding and requirements evolve.		