A successful project is the result of a number of things being done correctly. Looking at one area in isolation without taking account of others will result in failure. In essence, this is what the TOGAF ADM is about; bringing together the activities that need to be carried and in a coherent manner.

The list below is a “brain dump” of things that contribute towards success. Some of them could be regarded as principles, some as checklist items etc. They are not there to be categorized, they are there to provide a sense check.

1. Common Sense
2. Clear, SMART, objectives
3. Don’t take on too many things at once
4. Have a plan and share it
5. Have a single manager responsible for each area
6. Establish clear, simple and effective governance
7. Good communications
8. Re-evaluate regularly
9. Start as you mean to go on. Have good foundations and avoid a build up of technical debt.
10. Don't play the blame game, work closely together
11. It's the product that counts, not the process
12. Write pragmatic tests as part of development
13. Start with a thin line through the application and build on this
14. Prototype risky Use Cases early on
15. Everything should be transactional unless this is not possible
16. Build in security from the beginning
17. Separation of concerns
18. Build and test an application in tiers, working from the ground up
19. Start with a clear, simple and commented Vision which is agreed
20. Always have a plan
21. Test that a clean recovery happens when a transaction fails
22. Carry out performance testing from the beginning
23. Favour unit testing over manual testing
24. Apply all principles from the beginning. Start as you mean to go on, otherwise things will start failing
25. Provide good templates, standards and documentation from the beginning
26. Use a Wiki to provide documentation in an accessible and agile manner. Documentation has to be up to date.
27. Avoid template documents, focus on the content and usability of a document instead.
28. Don't start development work until requirements are understood
29. Don't start development work until there is a development environment
30. Keep things simple. Don't introduce new technologies and complexity without good justification
31. Only employ good people. Interview hard and don't just take people because they are all that are available
32. Build less, build what you need