Software Project Survival Test

An assessment instrument, which provides a “clue” to the project’s chances of successful completion: Will the software be delivered on time and within budget? Will it be a glorious achievement or an embarrassing mistake? To help you find out, the team should spend time analyzing and reflecting on the results of the survey.

SCORING:

3 = Definitive YES.
2 = Probably.
1 = Kind of, but not really.
0 = Definite NO.

If the project is in the early stages, answer the questions based on the project plan. If the project is well underway, answer the questions based on what has actually happened on the project.


Requirements

___  1. Did the project have a clear, unambiguous vision or mission statement?

___  2. Did all team members believe that statement was realistic?

___  3. Did the project have a user interface prototype that realistically and vividly demonstrated the functionality that the actual system would have?

___  4. Did the project have a detailed, written specification of what the software was supposed to do?

___  5. Did the project team interview people who would actually use the software early in the project and continue to involve them throughout the project?

Planning

___  6. Does the project have a detailed, written Software Management Plan?
7. Does the project’s task list include creation of an installation program, conversion of data from previous versions of the system, integration with third-party software (if applicable), meetings with the sponsor, and other “minor” tasks?

8. Were the schedule estimates and project status officially updated during the most recent team meeting?

9. Were the resource needs officially updated at the end of the most recently completed phase?

10. Does the project have detailed written architecture and design documents?

11. Does the project have a detailed, written Quality Assurance Plan that requires design and code reviews in addition to system testing?

12. Does the project’s plan include time for personal interrupts (emergencies or otherwise) and the possibility that needed resources will not be available?

13. Did the development team/the quality assurance team/the technical writing team – in other words, the team members responsible for doing the work, approve of the management plan, including the schedule? Approval in this sense is not just the signing on the approval page… but rather the implications of the plan and schedule for each team member.

**Project Control**

14. Has a single team member who has decision-making authority been made responsible for the project, and does the project have that person’s active support?

15. Does the project manager’s workload allow him or her to devote an adequate amount of time to the project?

16. Does the project have well-defined, detailed milestones (“binary milestones”) that are considered to be either 100 percent done or 100 percent not done?

17. Can the sponsor, the faculty adviser and the senior project instructor easily find out which of these binary milestones have been completed?

18. Does the project have feedback channel by which project members can report problems to the team, the faculty adviser and/or the senior project instructor?
19. Does the project have a written plan for controlling changes to the software’s specification?

20. Does the project have the equivalent of a Change Control Board that has final authority to accept or reject proposed changes?

21. Are planning materials and status information for the project – including effort and schedule estimates, task assignments, and progress compared to the plan thus far – available to every team member?

22. Is the implementation work (e.g. coding, data base creation, etc.) placed under some form of structured revision control?

23. Does the project environment include the basic processes needed to complete the project, including defect tracking, source code revision control, and project management oversight?

Risk Management

24. Does the project plan articulate a list of current risks to the project? Has the list been updated recently?

25. Does the project team have a formal process for identifying emerging risks to the project?

Personnel

26. Does the project team have all the technical expertise needed to complete the project?

27. Does the project team understand the environment in which the software will operate?

28. Does the project have the technical leadership necessary to complete the project successfully?

29. Are there enough people to do all the work required?

30. Does everyone work well together?

31. Is each person committed to the project?