Brady Field

Managing the Development of Large Software Systems

**Important points**

“There are two essential steps common to all computer program developments, regardless of size or complexity. There is first an analysis step, followed second by a doing step as depicted in Figure 1.” [328]

“These phenomena are not precisely analyzable. They are not the solutions to the standard partial differential equations of mathematical physics for instance. Yet if these phenomena fail to satisfy the various external constraints, then invariably a major redesign is required.” [329]

“Each and every worker must have an elemental understanding of the system.” [331]

“The first rule of managing software development is ruthless enforcement of documentation requirements.” [332]

“If the documentation is in serious default my first recommendation is simple. Replace project management. Stop all activities not related to documentation. Bring the documentation up to acceptable standards. Management of software is simply impossible without a very high degree of documentation.” [332]

“Without this simulation the project manager is at the mercy of human judgment. With the simulation he can at least perform experimental tests of some key hypotheses and scope down what remains for human judgment, which in the area of computer program design (as in the estimation of takeoff gross weight, costs to complete, or the daily double) is invariably and seriously optimistic.” [334]

“The previous three recommendations to design the program before beginning analysis and coding, to document it completely, and to build a pilot model are all aimed at uncovering and solving problems before entering the test phase.” [335]

“If it is argued that only the designer can perform a thorough test because only he understands the area he built, this is a sure sign of a failure to document properly. With good documentation it is feasible to use specialists in software product assurance who will, in my judgment, do a better job of testing than the designer.” [335]

**Disagreements**

“However, I believe the illustrated approach to be fundamentally sound.” [329]

The author obviously does not find the mentioned approach to be fundamentally sound or there would be no need for the remainder of the paper. This is further supported by other statements referring to the same approach.

“I believe in this concept, but the implementation described above is risky and invites failure. The problem is illustrated in Figure 4.” [329]

“The remainder of this discussion presents five additional features that must be added to this basic approach to eliminate most of the development risks.” [329]

These two statements, taken from the same page, illustrate that the author believes the aforementioned approach to be risky. Risk is anything but sound.

**Questions**

“One might note that there has been a skipping-over of the analysis and code phases. One cannot, of course, produce software without these steps, but generally these phases are managed with relative ease and have little impact on requirements, design, and testing.” [329]

I was wondering if this is true. I instinctually would like to disagree with it, but I lack a sound basis of experience to do so. Are the analysis and code phases really relatively untethered from the other processes? How would this be so?