

Sponsored by:

NETWORKWORLD

This story appeared on Network World at <http://www.networkworld.com/news/2008/121008-how-to-know-a-software.html>

How To Know A Software Development Project Is Doomed

By Esther Schindler , CIO , 12/10/2008

Despite all our efforts to make every software development project a success, some are cursed from the very start. Here are 26 early warning signs—all, alas, real-world experiences—that an enterprise software development project is headed for a [death march](#).

The project name changes for the third time in as many months.

The development manager decides that it is better to write a completely separate version of the software for the U.K. rather than to internationalize a single version.

The requirements definition is begun four months after development started.

Related Content

The newly hired director of R&D proudly informs the board of directors that the project will be 99 percent completed six months ahead of schedule, and assures the board that the software can ship directly to clients without going through beta testing.

You are a Web developer. You open the ZIP file with the HTML documents the client produced for the site scripts you need to integrate with the Web application. And you discover the client's HTML documents are all Microsoft Word files, saved in HTML format.

You realize the reason the company hired you as a consultant is to referee a dispute among two competing departments over which technical platform to use.

The memo says you will develop a 64-bit application using a 16-bit platform.

The developer doesn't understand the spec document and continues to develop anyway. And the QA team

Sponsored by:



Own a Xerox color printer
for under \$350.
Tell your boss.

Join us to experience the
colorful side of business →

frugalcolor.com

xerox

The advertisement features a white Xerox color printer with a red flower on the output tray. The text is arranged in a clean, modern layout with a green call-to-action button.

doesn't know how to test, but they "test" anyway.

When you see the project budget, you realize that over half of it was spent on a Web designer to create a Photoshop mock-up of the home page-with no regard to whether that design is feasible. Or with any attention to the thousands of pages of content that will exist underneath that home page.

The user or client requests new features instead of focusing on bug fixing and performance enhancements.

You find a list of 16 software development best practices and realize that not a single one of them is being followed.

You are asked to port your project from Windows to MS-DOS.

The technical project manager asks you to compose the list of user requirements-without consulting any actual potential users.

People started sending notes "to file" rather than to each other. The notes are alibis about why the sender has nothing to do with the upcoming (but unacknowledged) failure.

Status reports are seen as insubordinate.

The new CIO replaces all the people who have deep organizational knowledge with outsiders from his old firm.

It is a big project and is named Project Iceberg. Or it's the third time the company is trying to pull this off, and the project is code-named "Phoenix." Somehow, you don't believe this one can spring from the ashes.

Even the customers who got the free version are pissed off.

Related Content

The manager of your mission-critical project (handling 80 percent of the company's revenue) has three months exposure to the technology of choice, and is training four brand-new developers at once. The manager is given a three-month project deadline.

You learn that management had to insist that the interface definitions be checked into version control after the first code freeze.

They change the project manager and relocate the whole project from one city to another. (You consider yourself lucky that the cities are on the same continent.)

The QA team is told, "We've only allocated three weeks for testing" (on a project that has lasted six months already). Or QA is told, "The date is fixed. We have to have all this functionality by that date."

The program manager decided to try Agile methodology " to save time."

In a previous era, pre-cell-phones and ubiquitous Internet access: You get screeching abuse from a new project manager hired three days ago in New York, after you return from three days locked in regional CIO meetings in Frankfurt. Why? Because you hadn't responded to the e-mail messages she had sent (and which you didn't get), and you hadn't updated her "project dashboard" that you knew nothing about.

Management decides to spend a million dollars on a US\$20,000 project. Then the managers start agreeing with computer company salespeople that the \$1 million in software requires \$2 million of hardware. Meanwhile, a

secretary purchases an off-the-shelf PC and a shrink wrapped CD containing some new office automation packages. She implements the project during her lunch break. (Arguably, we should count this one as a success.)

The lead developer tells you that maintaining a complete history of all database updates is a requirement for the application, but he hasn't had time to (read: doesn't know how to) design a data model for it yet. So he decides to go ahead and start with the Web front end and worry about it later. And this is the lead developer.

The business line leader/project funder says, " Get creative." This happens after management reduces the project headcount by 20 percent. And after the IT team pulls out the hardware that had been slated for recycling, saying it's your project's new hosting environment.

That's a list based on input from dozens of software developers and IT professionals. But, alas, it is incomplete. It requires you to [add your moment of realization to the article comments](#).

All contents copyright 1995-2008 Network World, Inc. <http://www.networkworld.com>