

SoDIS Project Auditor™

Process Description

By David H. Gleason
Principal Consultant, ITforProgress.com
www.Linkedin.com/in/dgleason

Copyright © 2001
Rev. 11/13/09

I. Overview

Software projects and programs can be significantly improved by two activities: 1) project management, which is well understood and 2) analysis of the impact of activities on relevant stakeholders. The latter is not rigorously included in most standards and methodologies.

In order to succeed at a basic level, software must meet operational objectives and be timely and affordable. To excel, software must also have a net positive effect on stakeholders.

The cost of design flaws increase exponentially as IT projects progress. Early assessment of the impact of activities on each stakeholder can identify and mitigate unforeseen risks and problems. This method reduces project cost, and allows limited resources to be deployed where they are most needed.

II. SoDIS™ Project Auditing

A software tool for project auditing, called the *SoDIS Project Auditor*™,¹ has been created by Don Gotterbarn at East Tennessee State University and Simon Rogerson² at DeMontfort University in the U.K. This tool both addresses project management risks and provides a framework for stakeholder impact analysis based on the work breakdown structure (WBS) of the project or program under review.

Project stakeholders can be broadly defined to include those directly and indirectly affected by both a project or program and its products.

SoDIS is the result of careful analysis of software development activities on identified stakeholders. It applies a set of questions derived from the Software Engineering Code of Ethics and Professional Practice³ to work breakdown structures (project task outlines) and

¹ SoDIS stands for Software Development Impact Statement. SoDIS is trademarked and associated materials are copyright protected by the Software Development Research Foundation, Inc. (www.sdresearch.org).

² Gotterbarn and Rogerson are Professors in Computer Science at East Tennessee State University and DeMontfort University, respectively.

³ The Software Engineering Code of Ethics and Professional Practice has been adopted by the Association of Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers Computer Society (IEEE-

stakeholders. By exploring the impact of each task in software development to relevant stakeholders, SoDIS seeks at minimum to avoid reworking software after its completion. At its best, the SoDIS process, can help produce exemplary software.

The SoDIS software has three phases: Feasibility Analysis; Requirements Analysis and Detailed Analysis. Feasibility Analysis applies basic PM principles to software under consideration. Requirements Analysis identifies major project risks. Detailed analysis applies a sophisticated question set to each task in the WBS for relevant stakeholders.

The use of all three phases can help avoid the failures so prevalent in software development. Ultimately, this approach can produce better software.

III. What's new here?

There exists an extensive body of knowledge on project management. Stakeholder impact analysis using the SoDIS and its application of the Software Engineering Code of Ethics and Professional Practice is new.

In software development, step one is to get project management practices in place. Step two is to run feasibility analysis in relation to stakeholders. Step three is to do extensive impact analysis of individual project activities and outcomes on each relevant stakeholder.

Today, most software projects are still done at the ad-hoc level. Project management practices, the Capability Maturity Model (CMM) from the Software Engineering Institute at Carnegie Mellon, subsumption ethics and other techniques, bring project management to a level of reliability where expectations and specifications can be consistently met. The next step in improving project outcomes is to apply rigorous stakeholder impact analysis.

CS) as the standard for teaching and practicing software engineering. Gotterbarn and Rogerson were two of the lead authors of this code.