



Stalwart Analysis: The Effluvia of Determined Thought

Don Sengroiux
Founder of **Iceberg.com**.

Fellow of the:
L. E. Ott Foundation of:
Holistic and Overanalytic Studies.

Citizens! Good news! Our forces have struck a crushing blow against the Right Reverend C. Marin III¹, and his reactionary schemes of piecemeal development. In this talk, Don will lead us through the dialectical subtleties to the final truth!

We have long known that ANALYSIS LEADS THE DEVELOPMENT PROCESS! And now we have proof! STALWART ANALYSIS is the product of vast amounts of analytistic thought. For years Don Sengroiux, and his associates at Iceberg.com have been thinking, and thinking, and thinking. And now, finally, they are ready to present to us, for the first time, a process of software development that is perfect in every detail.

STALWART ANALYSIS is ready to be used on its first, ever, project. It has not been sullied with experience, nor battered by tests. No team has ever used it, or even kicked its tires. This process is NEW.

STALWART ANALYSIS has been analyzed from every possible vantage point. Thought experiment after thought experiment has been applied. There is no possibility of error, no chance of faltering. STALWART ANALYSIS is *done*.

STALWART ANALYSIS is the only process ever to have been fully formalized with a meta modeled meta model. This meta-meta-meta model allows complete and unambiguous representation of the process in meta tools; and supports full and accurate simulation of development teams of any ethnic composition or geographical bias.

This means that you can now predict the outcome of your projects by SIMULATING the development teams. The simulated developers make simulated plans and then execute them to write simulated code. The simulated code is then applied to simulated users who make simulated use of the simulated systems, and provide simulated feedback as part of the meta-meta-meta report language; eventually leading to simulated enterprises, simulated customers, and, of course, simulated profits.

¹ Pattern Languages of Program Design 3, page 579.