Analyst Complexity

Model. Expert Systems.[1]

Details   Marilyn Corbo-Crehan the head of a group of civil engineers (Civilizeng) authorized the use of stress analyses software (StressAn) to provide advice on a major construction project for *Really Tall Buildings P/L* owned by Anna Friedman.

Unknown to Civilizeng, StressAn had a bug that produced arithmetic errors.[2] Civilizeng engineers did not notice the odd values produced by the system which were approved for use by Denise Bobrowsky (the lead civil engineer) to construct the new building.[3]

Half-way through the construction process, a crane on the uppermost floor crashed through several floors killing three RTB workers[4].

An analysis of the disaster showed that the arithmetic bug and the misunderstanding of the structural properties of the steel were the cause of the failure. When combined with the weaker structural properties of the steel used, the calculations were totally inappropriate. The blame game began.[5]

The head software developer of StressAn, Judith Apel, argued that they had a responsibility to their shareholders' investment: too much testing costs too much. They relied on their expert to provide them with correct facts and rules for StressAn; it was the expert's failure that caused the incident.[6,7] It's not my fault that this new technology did not make the grade. It was too hard for the engineering team to check the calculations by other simple means: they were too complex, too numerous, and the project deadlines didn't allow for it. It seems that it just wasn't worth the trouble.

Apel argued that the arithmetic bug would have been of little consequence had more conventional construction materials been used. "Everyone knows" all software has bugs and that no system can be said to be entirely error-free. The users of the system should have been aware of this fact. Apel blamed the engineers for not checking at least some of the calculations: this would have revealed the errors. Apel argued that they are not in the business of building perfect systems, since this is both practically and theoretically impossible for the scale of systems they market. No amount of further knowledge, no amount of further checks and validations could have guaranteed that the software was correct. Users must reconcile themselves to the real world where systems do fail, despite the best efforts, methodologies, and procedures. Users should not place blind faith in their technology; users should not abbreviate design work and engineering processes.

The lawsuits started.

[1] This is modified from a case told to me a long time ago. I tried but failed to locate the original. Source: Unknown.

Cast	Role
Judith Apel	Developer of StressAn
Martin Berkley	RTB worker
Denise Bobrowsky	Senior Civil Engineer
Marilyn Corbo-Crehan	Civilizeng CEO
Anna Friedman	RTB CEO
George Heidegger	RTB worker
Tammy Oram	RTB worker
Rebecca Simon	Structural engineering expert

Table 1: A_3 Cast

[2] A rather famous bug in the IBM software ACRITH had pretty much this same property.

[3] StressAn contained incomplete information concerning an alloy that had very good corrosion properties but slightly diminished ability to bear loads compared to more conventional steel. Rebecca Simon, the expert consulted when building StressAn did not fully understand the difference in the load-bearing ability and this difference was not used. In selling the package to engineers and architects, the developers of the system promoted it as being the safest in the world.

[4] Tammy Oram, Martin Berkley, and George Heidegger.

[5] It started with the engineers arguing that checking the calculations was simply impractical given the deadline; they relied on the claims of the developers.

[6] The expert, Rebecca Simon, claimed that all human knowledge is inadequate and incomplete. Much of any expert's knowledge and rule-based behaviour is actually contradictory. Furthermore, Simon said that no one individual can be expected to know or to keep pace with such a fast-moving and complex area of study. Apel should have realized this and used knowledge from other sources. Appel should have built more checks and controls into the software.

[7] SoftStress was assured by architects, engineers, and designers that automation of the design/engineering process would save time and cut costs.