incident or about Samuels and his work habits.

"We were a democratic team, except for the managerial guidance provided by Sam [Reynolds]", another team member observed. In the world of software development, a democratic team is a team in which all team members have an equal say in the decision-making process. "Unfortunately, we were a team of very ambitious, very talented - if I must say so myself - and very opinionated individualists. Randy [Samuels] was just the worst of the lot."

One co-worker told of an incident in which Samuels stormed out of a quality assurance meeting. This meeting involved Samuels and three 'readers' of a software module which he had designed and implemented. Such a meeting is called a code review. One of the readers mentioned that Samuels had used a very inefficient algorithm (program) for achieving a certain result and Samuels "turned beet red". He yelled a stream of obscenities and then left the meeting. He never returned. "We sent him a memo about the faster algorithm and he eventually did use the more efficient algorithm in his module", the co-worker added.

The software module in the quality assurance incident was the very one which was found to be at fault in the robot operator 'murder'. However, this co-worker was quick to point out that the efficiency of the algorithm was not an issue in the malfunctioning of the robot.

"It's just that Randy made if very difficult for people to communicate their concerns to him. He took everything very personally. He graduated tops in his class at college and later graduated with honors in software engineering from Silicon Valley University. He's definitely very bright."

"Randy had this big computer-generated banner on his wall", this co-worker continued. "It said, 'YOU GIVE ME THE SPECIFICATION AND I'LL GIVE YOU THE COMPUTATION'. That's the kind of arrogance he had and it also shows that he had little patience for developing and checking the specifications. He loved the problem-solving aspect, the programming itself".

"It doesn't seem that Randy Samuels caught on to the spirit of 'egoless programming' ", Professor Skinner observed upon hearing this part of the interview with Samuels' co-workers. "The idea of egoless programming is that a software product belongs to the team and not to the individual programmers. The idea is to be open to criticism and to be less attached to one's work. Code reviews are certainly consistent with this overall philosophy." A female co-worker spoke of another aspect of Samuels' personality - his helpfulness. "Randy hated meetings, but he was pretty good one on one. He was always eager to help. I remember one time when I ran into a serious roadblock and instead of just pointing me in the right direction, he took over the problem and solved it himself. He spent nearly five entire days on my problem".

"Of course, in retrospect, it might have been better for poor Mr. Matthews and his family if Randy had stuck to his own business", she added after a long pause.

'KILLER ROBOT' PROJECT MIRED IN CONTROVERSY RIGHT FROM START

WARRING FACTIONS FOUGHT OVER HOW PROJECT SHOULD PROCEED

Special to the SILICON VALLEY SENTINEL-OBSERVER
Silicon Valley, USA
by Mabel Muckraker

Two groups, committed to different software development philosophies, nearly came to blows during the initial planning meetings for Robbie CX30, the Silicon Techtronics robot which killed an assembly line worker last May. At issue was whether the Robbie CX30 project should proceed according to the 'waterfall model' or the 'prototyping model'.

The waterfall model and the prototyping model are two common methods for organizing a software project. In the waterfall model, a software project goes through definite stages of development. The first stage is requirements analysis and specification, during which an attempt is made to arrive at an agreement concerning the detailed functionality of the system. As the project passes from one stage to the next, there are limited opportunities for going back and changing earlier decisions. One drawback of this approach is that potential users do not get a chance to interact with the system until very late in the system's life cycle. In the prototyping model, great emphasis is placed on producing a working model or prototype early during the life cycle of a system. The prototype is built for the purpose of arriving at a final specification of the functionality of the proposed system. Potential users interact with the prototype early and often until the requirements are agreed upon. This approach affords potential users the opportunity to interact with a prototype system early during the development cycle and long before the final system is designed and coded.

In a memo dated December 11th of the year before last, Jan Anderson, a member of the original Robbie CX30 project team, bitterly attacked the decision of the project manager, Sam Reynolds, to employ the waterfall model. The Sentinel-Observer has obtained a copy of Anderson's memo, which is addressed to Reynolds, and Anderson verified the authenticity of the memo for this reporter.

Reynolds fired Anderson on December 24th, just two weeks after she wrote the memo.

The Anderson memo refers to an earlier meeting at which an angry exchange occurred relating to software development philosophy. Anderson underlined the following passage in her memo:

"I did not intend to impugn your competence at our meeting yesterday, but I must protest most vehemently against the idea that we complete the Robbie CX30 software following the waterfall model which you have used in previous projects. I need not remind you that those were data processing projects involving the processing of business transactions. The Robbie CX30 project will involve a high degree of interaction, both between robot components and between the robot and the operator. Since operator interaction with the robot is so important, the interface cannot be designed as an afterthought."

Randy Samuels, who has been charged with manslaughter in the death of robot operator Bart Matthews, father of three, was in attendance at the December 11th meeting.