Dave Rico's 1,600-Word Scrum Story

In 2008, one of the largest universities in the world, set out to teach Agile Methods to its master's level graduate students. The setting was the final capstone course towards a master's of science degree in software engineering. The course had been taught 15 times over the prior eight years. The primary objective was for the students to apply the traditional methods, they'd been taught over the two years of their coursework, to the design of a real-world software-based system for a real client. The secondary objective was to use agile methods, namely Scrum, so that the students could gain some experience with agile methods and determine how many traditional methods and practices would be necessary as stop-gap measures. Three teams of five students signed up for the final, capstone course. The teams were assigned a customer, an agile coach, and instructed to build competing electronic commerce website designs. The objective of the final product would be for the customer to successfully purchase an electronic textbook using an actual credit card. The students were assigned customer requirements, given ample background material in agile methods, a modicum of training in agile methods, and little else. Keep in mind that the students had no formal training in agile methods during the coursework period of their degree. All of them were working professionals and many of them were newly-minted project managers, technical leads, analysts, and other functional specialists. They welcomed the principles of traditional methods taught to them in their courses and were actively introducing them into their workplaces as a means of attempting to control the chaos, where ad hoc development practices were the norm. The notion of agile methods was completely new to most of the students, and seemingly contrary to what they'd been taught during their courses. The bottom line is that these students were steeped in traditional methods and were faced with building an operational e-commerce website for a real-world customer using agile methods in little more than 13 weeks. The first three weeks were spent forming teams, self-selecting roles, familiarizing themselves with agile methods, and receiving just-in-time training and education in agile methods from an agile coach. Many of the students had selected roles such as project manager, tester, and quality assurance analyst, and few had selected roles as developers or programmers. Much of this time was spent mapping traditional methods and practices to those of agile methods (i.e., project plans to release and iteration plans, requirements to user stories, etc.). This was probably the most critically-important and tension-filled part of the course. There were also competing interests and objectives among the instructional staff. The senior instructors encouraged the students to question everything, in order to intentionally stoke the coals of a fierce debate between traditional and agile methods. The agile coach on the other hand was trying to help the students acclimate to the culture of agile methods, answer questions, prepare the students for their journey, and, of course, quench the fire started by the senior instructors. The instructional team also participated in a fierce debate about whether the students should be able to construct their websites using commercial web services, or build them one-line-of-code-at-a-time (i.e., buy-versus-build). The senior instructional team insisted that the websites must be constructed using third-generation computer programming languages one-line-of-code-at-a-time, as that was the primary mission of a graduate student in software engineering. The students probably felt like they just landed on an alien world, and so did the agile coach. Everyone certainly wanted to return to Earth at that point. At the end of the third week, the three teams were formally introduced to their customer and provided identical lists of high-level customer requirements. The teams were asked to develop release plans, iteration plans, user stories, development tasks, unit and integration tests, and of course, a working operational software. Then, the next phase of the program began, which was the first Iteration. We could also call this "Iteration 0", since this was the first of three iterations in which the student teams would begin applying agile methods to the development of their websites. One of the teams had actually started Iteration 0 activities prior to Iteration 1, set up their web servers, selected their technology stack, and put up a simple website with a splash screen before meeting with the customer. The first team used Iteration 1 as their Iteration 0, adopted an agile project management tool to help them along, and demonstrated their basic website at the end of the sixth week. The second team also used Iteration 1 as their Iteration 0, used a Wiki for

managing release/iteration plans and user stories, and also performed a basic demonstration. The third team had already demonstrated their basic website earlier, also used a Wiki for release management, began in earnest to implement their user stories, and demonstrated a slightly more advanced website at the end of Iteration 1. The first team had the most talent, including the most developers and applied Scrum to the letter-of-the-law. However, they also had a very traditional adversarial relationship with their customer, to which they attributed to their real world jobs. They decomposed all of their user stories into technical tasks, discarded their user stories, and scheduled out their development tasks for the remainder of the term. The agile coach gently asked them to reconstitute their user stories and use these as a basis for their demonstrations, which they promptly did. The second team insisted upon translating all user stories into wire frames, getting the customer buy-in on the designs along with screen shots before actual implementation. The agile coach gently reminded them to take more responsibility for design-level decisions. The third team was led by their primary developer, who started out quickly, but grew frustrated with the rest of the team who had roles such as tester, quality assurance analyst, etc. The third team began implementing the website the way they saw fit, ignored their user stories, and started grinding to a halt. The third team required a lot of care and feeding by the agile coach. The agile coach gently reminded the third team to adhere to the user stories as agreed-upon. The agile coach also had to mentor the third team's lead, primarily on the merits of teamwork and gently guide the lead away from their fiercely individualistic nature and qualities. The agile coach established virtual pair programming sessions using WebEx and Skype to bring all of the third team's members together, share the burden of Web development, and encourage the team lead to keep moving forward and not give up, because of the perception of unfair burden. The first team really hit stride by the third iteration, completed all of the user stories, and demonstrated the most mature website design enabling the customer to purchase and download an e-book. The second team did similarly well, and developed a website enabling the purchase of an e-book. The third team stayed together in-spite of their difficulties, began depending on one another instead of leaning on the agile coach all of the time, and completed a website as well. In the end, all three teams completed their websites without having to employ traditional practices, as desired by the senior instructors. Furthermore, all three teams wrote case studies of their experiences praising the merits of agile methods, including teamwork, flexibility, and intensive customer collaboration. Of the 15 prior capstone courses, this was the only semester in which an operational project was ever completed. Prior semesters had made it as far as producing project plans, requirements documents, designs, and tests, but none had ever actually implemented an operational product in a single semester. However, prior students were often rewarded for successfully implementing the tenets of traditional methods (i.e., processes, documents, etc.). More teamwork among the instructional staff would have helped, or at least the lack of competing objectives. Formal training in agile methods would have been great. Providing a technology stack to the students would have been beneficial, instead of expecting the students to self-identify one (including agile project management tools). Setting up virtual collaboration tools would have been even better. Having a full-time agile coach available to the teams on a 7x24 basis, with a vested interest in the success of agile methods was probably the single most important element of success that brought everything together. The agile coach was able to plan for as many contingencies as possible, observe the teams, gently guide them when they steered off course, run interference for the teams when their success was being undermined, set up safety nets to minimize the impacts of falling, improvise new tools and techniques when all else failed, and assume a vested interest in individual success as well. All of the students passed this course and earned their degrees. On a side note, the agile coach spent a lot of time mentoring the lead of the third team who had highly individualistic qualities, which threatened to undermine the success of the entire team. That lead probably learned more about agile methods than the other 14 students. The lead went on to become a web developer for the IT department of a small U.S. government agency's field office. In little more than 90 days, the lead transformed the IT infrastructure, bringing it into the 21st century. The lead could have spent years documenting every conceivable requirement or embody the principles of agile methods and focus on delivering business value. In this case, the lead focused on the latter.