



# All We Need to Know About Software Project Management, We Can Learn From Watching Star Trek

David R. Webb

309 Software Maintenance Group, Hill Air Force Base

*Are you Kirk, Spock, Sulu, Chekov, Uhura, or Scotty? If you are part of a modern-day software engineering team, chances are you fit one of these roles. As we have read about Star Trek's influence on technology, sit back for a warp drive look at how a software engineering team's structure mimics that of the Star Trek bridge crew.*

A few months ago, I watched a cable TV show which asserted that a *Star Trek* actor had *changed the world*. While the idea that William Shatner had single-handedly brought about the 21st century as we know it was funny, the information presented was pretty convincing that the science fiction show had a major impact on modern technology. Just call your bank's voice-recognition computer using your flip-phone and ... well, you get the picture. And it does not end with the original series; those reconfigurable, flat-panel touch screens that Geordi and Data sit in front of in *The Next Generation* can now be found in just about every fast food restaurant!

As I marveled at the effects a television series had had on our modern lives, I suddenly realized that most of the current software generation had grown up with *Star Trek*. I began wondering what effect this had on our approaches to software development and management. What I discovered I have termed *The Gene Roddenberry Effect*: Everything we do in software project management originated with *Star Trek*. I have developed a list of lessons learned from *Star Trek* that I regularly employ.

## The Bridge Crew

One surprising example of how *Star Trek* has affected software project management is how the composition of the bridge crew on the starship Enterprise reflects that of current software teams. In the original

series, most of the stories centered around the bridge crew: Kirk, Spock, Sulu, Chekov, Uhura, Scotty (you know the names). Kirk was the ultimate leader and decision-maker, Spock was second-in-command and science officer, and all the others had specific job titles as well.

Modern industry has pretty much borrowed this structure for software engineering teams. Every project has a project manager (PM), a technical expert who acts as backup PM, a chief engineer, a communications officer (configuration manager), and so forth. The PM calls the shots, much like the captain (Kirk), and everyone else performs their particular jobs and regularly reports back to him. When the team has issues, it gathers in a conference room and projects everything on a flat panel monitor for all to review. The team makes a group decision, guided and ratified by Kirk. Watch those conference meetings in the original series and see if you get goose pimples at how similar they are to the meetings you go to each week.

Perhaps most surprising is the similarity between the bridge crew concept in conjunction with the Software Engineering Institute's (SEI) Team Software Process<sup>SM</sup> (TSP<sup>SM</sup>). The TSP is SEI's *how-to* guide for implementing high-maturity software engineering teams. In the TSP, each member of the team is given one of eight management roles. As I began looking at them, I was amazed at how closely they matched the

bridge crew concept (see Table 1).

What is even more interesting is that many TSP teams actually do double up on roles as shown in Table 1. If they do not have eight people to perform the different roles, people take on more than one.

But wait; it gets even better.

Once we move to *The Next Generation* series, we find a whole new bridge crew with expanded roles. In this case, Picard, the captain, is no longer the PM, but has risen to the rank of *senior management* (as described in SEI's Capability Maturity Model Integration 1.1). We can see from Table 2 that the team concept has matured aboard the newer Enterprise and that our more modern approaches to the software team concept have mimicked this structure.

We could go even further and examine the newer *agile* development methodologies only to discover that as the *Star Trek* series matured, so did their concepts of teams and agility. Captain Janeway's Federation/Maquis team aboard Voyager, for example, is the epitome of an agile team. In any case, whether or not it was done intentionally, it seems that modern software teams are indeed modeled after the *Star Trek* bridge crews. Some of the agile methods even measure a project *velocity*. A coincidence? Well, maybe.

## Lessons Learned

So, with the obvious links between the *Star Trek* universe and the way we manage software projects, what else we can learn from *Star Trek* that can actually help us in our day-to-day management of software projects? The following are the 10 project management tips I use most often:

10. When push comes to shove, it's always a computer geek who comes to the rescue! (Watch just about every *Star Trek: The Next Generation* episode with Wesley Crusher.)
9. Your greatest challenges can be the ones you thought you got rid of 200 years ago. (Remember Kahn [Ricardo Montalban] from the original episode

Table 1: TSP Roles vs Enterprise Bridge Crew

TSP <sup>SM</sup> Roles	Bridge Crew Role	Bridge Crew Member
Team Leader	Captain	Kirk
Customer Interface Manager	Communications Officer	Uhura
Design Manager	Science Officer	Spock
Implementation Manager	Chief Engineer	Scotty
Planning Manager	Navigation Officer	Sulu
Process Manager	Operations Officer	Chekov
Quality Manager	Medical Officer	Bones
Support Manager	Science Officer	Spock
Test Manager	Chief Engineer	Scotty

*Space Seed* and the movie *Star Trek II: The Wrath of Khan*? Remember Y2K?)

8. If everyone on your project is too happy, you probably are not accomplishing anything. (Yes, it's the hippie episode about the spores, called *This Side of Paradise*. For those who are not ardent fans, trust us on this one.)
7. You cannot change the laws of physics, but you *can* bend them. (This is a line made famous by Scotty in the original series episode entitled *The Naked Time*.)
6. There is no such thing as a no-win scenario, especially if you change the conditions of the test. (This is Kirk's philosophy and fits pretty well into his profile above! Watch *Star Trek II – The Wrath of Khan* for how this worked for him ... or did not.)
5. Every successful project manager has both a good side and bad side and knows how to balance them. (Let's just hope you do not beat yourself up the way Kirk did in *The Enemy Within* – literally. In that episode, Kirk had an *evil twin* created by a transporter malfunction.)
4. Don't feed the Tribbles or they will overrun you! (Substitute your own set of problems for Tribbles as seen in *The Trouble with Tribbles*. If you do not know what a Tribble is, why are you reading this article?)
3. If you do not apply a lot of power to break away from your routine, you are doomed to repeat the same mistakes over and over again for eternity. (*The Causality Loop* from *The Next Generation* – great episode! Go rent it!)
2. Some people can be very afraid of change. (Let us just hope it doesn't lead to the shedding of blood, red or pink as it did in the movie *Star Trek VI – The Undiscovered Country*.)

TSP <sup>SM</sup> Roles	Bridge Crew Role	Bridge Crew Member
Senior Management	Captain	Picard
Team Leader	First Officer	Riker
Customer Interface Manager	Security Officer	Worf
Design Manager	Science Officer	Data
Implementation Manager	Chief Engineer	Geordi
Planning Manager	Navigation Officer	Wesley
Process Manager	Operations Officer	Data
Quality Manager	Medical Officer	Crusher
Support Manager	Ship's Councilor	Troi
Test Manager	Chief Engineer	Geordi

Table 2: TSP Roles vs The Next Generation Enterprise Bridge Crew

1. Always multiply your estimates by a factor of four so that you will be known as a *miracle worker*. (This comes from the movie *Star Trek III – The Search for Spock*, and we have no comment as to why this is number one or how we implement it; especially if any of our customers are reading this!)

### Conclusion

While this article is a bit tongue-in-cheek, it's amazing how closely our software engineering practices mirror *Star Trek*. So live long, prosper, and remember to thank (or maybe curse) the *great bird of the galaxy* next time you put together those PowerPoint slides for your management review!◆

### About the Author

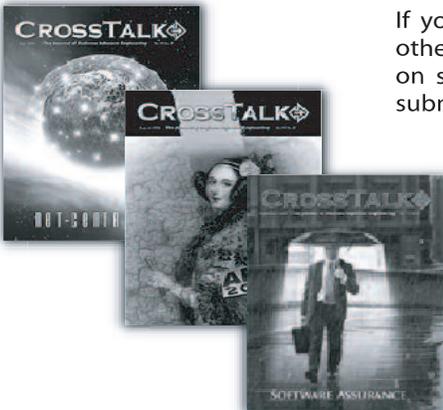


**David R. Webb** is a senior technical program manager for the 309th Software Maintenance Group at Hill Air Force Base, Utah, a CMMI Level 5 software organization. He is a project management and process improvement specialist with 19 years of technical, program management, and process improvement experience with Air Force software. Webb is a Software Engineering Institute-authorized instructor of the Personal Software Process<sup>SM</sup>, a Team Software Process launch coach, and has worked as

an Air Force flight director, Software Engineering Process Group member, systems software engineer, lead software engineer, and test engineer. He is a frequent contributor to CROSSTALK and has a bachelor's degree in electrical and computer engineering from Brigham Young University.

**309 SMXG/520 SMXS**  
**7278 4th ST**  
**Hill AFB, UT 84056**  
**Phone: (801) 940-7005**  
**Fax: (801) 775-3023**  
**DSN: 775-3023**  
**E-mail: david.webb@hill.af.mil**

## CALL FOR ARTICLES



If your experience or research has produced information that could be useful to others, CROSSTALK can get the word out. We are specifically looking for articles on software-related topics to supplement upcoming theme issues. Below is the submittal schedule for three areas of emphasis we are looking for:

#### Agile Development

April 2007

Submission Deadline: November 15

#### Software Acquisition

May 2007

Submission Deadline: December 4

#### COTS Integration

June 2007

Submission Deadline: January 22

Please follow the Author Guidelines for CrossTalk, available on the Internet at <[www.stsc.hill.af.mil/crosstalk](http://www.stsc.hill.af.mil/crosstalk)>. We accept article submissions on all software-related topics at any time, along with Letters to the Editor and BackTalk.