

Mars versus Venus

You've undoubtedly heard about John Gray's best-selling pop-psychology book, *Men Are from Mars, Women Are from Venus* (HarperCollins Publishers, 1992). It describes the postures, assumptions, and behaviors of the gender cultures; how they seem so opposite from each other you'd think they came from different planets; and how they can build bridges for more harmonious relationships. I've observed that men and women are not the only two groups suffering from a great planetary divide. It sometimes strikes me that product engineers (software developers and development managers) are from Mars and process engineers (software quality engineers, quality assurance engineers, process improvement specialists, and change agents) are from Venus. Product engineers appear to have one belief system and values, and process engineers seem to have another.



Differences

There is antagonism and distrust between product and process engineers, and they frequently work at cross-purposes to each other. Product engineers can drive process engineers crazy with comments such as the following (from *Six Warning Signs That Your Software Project May Be In Trouble*, www.fullspectrumsw.com):

- “We thought up some cool new features and added them over the weekend.”
- “We'll fix the bugs in beta.”

- “We don't have time to test.”
- “We're behind schedule; we don't have time for team meetings.”
- “We're going to start coding the stuff we know and design the rest on the fly.”
- “We could make the current schedule if we could add five new developers.”

Likewise, process engineers can drive product engineers crazy with questions such as

- “I know this is short notice, but can we schedule an ISO audit with you next week?”
- “Can you fold in time for training the developers for an IPI-CBA?”
- “Can you show me all your quality records?”
- “How many KPAs have you implemented?”
- “Can you show me all your meeting minutes over the last year?”
- “What is your quality policy?”

What appears glaringly obvious to one group leaves the other's members shaking their heads, wondering what planet they came from.

Moving Forward

However, a company's survival depends on both groups' contributions. Product engineers focus on building the product and the “present” goal of getting that product out the door. Process engineers tend to focus on engineering the product development effort, assessing the “as is” activities with the goal of finding opportunities to improve the process

in the near future. You can see how these two groups might annoy each other. Product engineers find the process work irrelevant and distracting, and process engineers find product engineers arrogant and evasive.

To help these two groups work more cooperatively and constructively, each group must understand the other's perspective. No one would argue the product developer's ultimate value. Most companies wouldn't exist without products. Product engineers are frequently under incredible time pressure to hit a market window or a specific customer deadline. Having time to step back and engineer the development process seems like a wasteful luxury. If a developer's management doesn't value process engineering and improvement, it's a nearly impossible task for a process engineer to engage with the product engineers simply because it's not rewarded behavior.

On the other hand, the process engineer's value is not as self-evident. The origin of software process engineering is rooted in Edward Deming's work on statistical process control. (See D.W. Hutton, *Change Agent's Handbook*, ASQ Quality Press, 1994, pp. 292–295 for more details.) The basic premise is if you build a product and it has flaws, don't just fix the flaws. Fix the process that allowed the flaws into the product so you won't have to keep fixing the problems in subsequent productions.

Although the US showed interest in Deming's work during WWII, there wasn't much after the war. Deming took his approach overseas where the Japanese auto manufacturers listened very closely and followed his teachings. Auto manufacturers in Detroit have yet to fully recover from the impact Japan's high-quality cars made when they came to the US in the '70s. Deming's quality approach has moved from manufacturing into engineering, including software engineering, and standards such as ISO 9000 and SEI's CMM have captured it (for better or worse).

Although process improvement work has proven effective, process engineers in today's software companies

are frequently under pressure to get the development organization registered to ISO 9001 or assessed at CMM Level 2 and higher for marketing purposes. If the directive comes without the mandate to reap the potential benefits accruing from these process frameworks, then the process engineers are stuck with the task of getting the organization to "pass a test" with potentially no long-term impact on company practice.

Putting Collaboration into Practice

So just what are tactics for a constructive collaboration between these two groups? For one, process engineers need to focus on improving areas that have real significance to the developers and the business as opposed to instituting bureaucracy. They need to look at the development effort at hand, look for the opportunity to help, and roll up their sleeves and pitch in. For example, if a development group is plagued by bugs and hasn't started using design and code reviews, the process engineer can do the training, help gather the statistics on bugs, and demonstrate the improvement reasonably quickly. Having small but tangible successes is a great sales technique that can eventually win over product engineers.

Teams and Management

In balancing the pull and tug between product and process work, there is no substitute for enlightened leadership—leaders and managers who see the value in both product and process engineering and can do the tricky job of balancing the timing for both efforts. Management also needs to set a tone of trust with product engineers so that they will open up to an assessment of their engineering processes. They must believe that management will use the assessment data to improve the system instead of punishing people for poor performance. Without trust, it will be hard to collect the data, and its accuracy will be highly suspect.

In addition to good management, teams with a highly disciplined engi-

neering process will not only improve product quality but will actually improve their time to market. An outstanding example of what well-defined process and discipline can achieve is captured in a construction-industry video, *The 4 Hours House*, developed by the Building Industry Association of San Diego, California. Amazingly, the construction team built a house from pouring the foundation to move-in readiness in less than four hours (see P. Hantos and M. Gisbert's related case study in the Jan./Feb. 2000 issue).

Improvement Strategies

An effective tactic one company uses is to tie employee bonuses to an on-going customer satisfaction survey. In this company, almost all the employees are eligible for bonuses, not just management or senior management. As you might imagine, the responses provide the basic feedback loop to leverage process improvement across the company without management having to utter the "P" word.

If used effectively, process engineering can create and sharpen a highly effective engineering discipline. By analogy, in the sports arena, game play needs to be practiced and instilled at almost an instinctive level to improve player performance in the game. For musicians in an orchestra, both the individual musician and the orchestra as a whole need to practice to achieve a disciplined and flowing performance.

The Mars and Venus factions in a software project have much to gain from each other. Their interactions might always have an element of creative tension, but collaboration between product and process engineers has the potential to improve both product quality and time to market, which in turn can lead to improved customer satisfaction. And increased customer satisfaction can reap the company excellent business rewards. ☞

Karen Mackey is a quality systems engineer at Cisco Systems. She has previously worked as a software developer and manager at Lockheed Martin, Lotus, TRW, and Bell Labs. She received a PhD in computer science from Pennsylvania State University. Contact her at kmackey@best.com.