It's often said that if you're not measuring, you're not managing, and this includes the world of IT. Software metrics are generally quantitative measures of some property of software or its specifications, and it's essential that these measurements be objective and reproducible. The metrics we use in IT are as diverse as the activities encompassed within IT, although many practitioners tend to recognize and use metrics only for the software development phases. Here, I explore a variety of metrics and how they relate to both quality and user satisfaction, including how they affect our interactions with social media.

Quality Metrics
Generally, in management, we use metrics to develop and manage scheduling, budget planning and execution, and project cost estimation. Throughout the life of the project, we continually use metrics to evaluate quality assurance, testing, and reliability, although some projects attempt to measure these aspects only at the end of the project, which isn't successful.

In software development, IT professionals often use metrics, such as the number of lines of code, bugs per line of code, down time, up time, throughput, cohesion, coupling, and comment density. We often use more complex metrics, such as McCabe's cyclomatic complexity, Halstead complexity, function point analysis, and models such as Cocomo. All of these, when used correctly, will help software developers release code and applications that are more reliable, thus saving the company money in recalls, new releases, and patches, in addition to maintaining a company image of high-quality software products. Another aspect of IT metrics is in cybersecurity—evaluating the networks for penetration attempts, intrusions, and insider threats—all vital for successful data protection.

Usability Metrics
There's another set of metrics related not to the quality of the software products or networks but to the usability of a software application, process, or machine. The objective of these metrics is to determine if the website or application was designed considering the user's psychology and physiology, efficiency, ease of learning, and satisfaction. Methods of measuring usability often include a needs analysis and studying perceived efficiencies and elegance.

Human-computer interaction (HCI) usability studies the elegance and clarity of the design of the interaction with the computer program or website. Although it looks at the usefulness and the value to the user, it's not the study of user satisfaction. Usability metrics evaluate software efficiency—considering questions such as how long does it take to accomplish a task, is the program easy to learn, and is it satisfying to use? One method for evaluating usability testing is to observe real system users as they perform specific tasks, noting their reactions and difficulties.

User Satisfaction
User satisfaction, not to be confused with usability, is the user's attitude toward the system. This is generally not measured quantitatively but rather qualitatively, although the user can rate the system numerically through a user satisfaction survey. Users are often asked to use a multipoint scale to rate factors ranging from feelings of control, volume of output, and degree of training, to more complex factors such as accuracy, timeliness, and confidence in the system.

Although these aren't the metrics that come first to mind when we talk about software metrics, if a user is dissatisfied with the software, it will be broadcast on websites, blogs, and Twitter, and the sales and profit often plummet proportionately with user dissatisfaction.

Ergonomics
Related to HCI is ergonomics—the understanding of the interactions...
between humans and the computer components to optimize comfort and system use and improve health, safety, and productivity. The objective is to design and use IT equipment and prevent repetitive strain injuries and other musculoskeletal disorders that can, over time, lead to long-term disability.

The metrics of ergonomics are as diverse as the way we use IT equipment, but the ones that have always interested me relate to how a person sits and uses a desktop computer. Specific measurements reveal the ideal distance from the screen (18 to 24 inches), position for arms and legs (90-degree angles), and keyboard height (23 to 28 inches from the floor) to optimize performance and minimize strain.

Social Engineering Metrics
This special issue is on social computing, the area of computer science where social behavior and computational systems interact (see the sidebar). These systems recreate social conversations, such as blogs, email, and instant messaging. The metrics I’ve discussed all converge in the area of social computing, and many are used in the development of computational systems to ensure the network quality, security, and dependability that users demand.

However, the metric that often gets lost in the social computing arena is that of ergonomics. How do you sit at home, if you even use a desktop computer? Where does the wireless keyboard sit? Is it in your lap? Are you sitting straight with your feet on the floor, or are you leaning back, with your feet propped up? If the keyboard is on the desk, is it flat or angled? Although developers have focused on metrics for quality and reliability, in the years to come, what will be the impact of ignoring the metrics of ergonomics?

There have been many articles written about problems in the wrists due to repetitive keyboard use, issues with shoulders and backs when slouching, and even issues with circulation. Now picture yourself in your favorite chair, with your tablet in your lap. Do you have your chiropractor on speed dial? Look at your kids, as they spend hours on the computer doing schoolwork and socializing. How is their posture affecting their future physical health?

This is a field of metrics that, although it exists, receives little attention, because we’re too comfortable in our current positions. But we need to recognize that we’re inviting future health risks for ourselves and our children. Ignoring this issue will, ultimately, result in high user dissatisfaction.

Linda Wilbanks is the Chief Information Security Officer for Federal Student Aid, US Department of Education. Contact her at linda.wilbanks@ed.gov.

Selected CS articles and columns are available for free at http://ComputingNow.computer.org.

LEAD DEVELOPER
Yellowbook, Inc., Cedar Rapids, IA. Perform highly complex programming tasks to support business systems of Yellowbook. Manage development efforts of other developers. Analyze systems, design, and implementation tactics across aspects of a project and produce system analysis, design, test case and implementation docs. Bachelor’s Degree in Computer Science or Computer Engineering and 5 years software development experience req. Must have proficiency in complex SQL DML/DQL usage, complex component design and development, new multi-platform programming languages and environments. Must demonstrate proficiency in a minimum of 3 or more of the following: JAVA, Enterprise JAVA Beans, JSP, JAVA Script, HTML, XML, SOAP, Cold Fusion, RPG III, RPG IV, RPG ILE, MS-Visual Basic, MS-Visual C++, COM, Dcom, Oracle Forms and Reports, Oracle PL-SQL, or DB/2 400. Fax cover letter and resume to Justin Doerner at 319-790-8213.