

Companies with Web 2.0 characteristics are Google (web search), YouTube (video sharing), Facebook (social networking), Twitter (microblogging), Groupon (social commerce), Foursquare (mobile check-in), Salesforce (business software offered as online services “in the cloud”), Craigslist (mostly free classified listings), Flickr (photo sharing), Skype (Internet telephony and video calling and conferencing) and Wikipedia (a free online encyclopedia).

Web 2.0 *involves* the users—not only do they create content, but they help organize it, share it, remix it, critique it, update it, etc. Web 2.0 is a *conversation*, with everyone having the opportunity to speak and share views. Companies that understand Web 2.0 realize that their products and services are conversations as well.

Architecture of Participation

Web 2.0 embraces an **architecture of participation**—a design that encourages user interaction and community contributions. You, the user, are the most important aspect of Web 2.0—so important, in fact, that in 2006, *TIME* magazine’s “Person of the Year” was “You.”¹⁰ The article recognized the social phenomenon of Web 2.0—the shift away from a *powerful few* to an *empowered many*. Popular blogs now compete with traditional media powerhouses, and many Web 2.0 companies are built almost entirely on user-generated content. For websites like Facebook, Twitter, YouTube, eBay and Wikipedia users create the content, while the companies provide the *platforms* on which to enter, manipulate and share the information.

1.13 Some Key Software Development Terminology

Figure 1.27 lists a number of buzzwords that you’ll hear in the software development community. We’ve created Resource Centers on most of these topics, with more on the way.

Technology	Description
Ajax	Ajax is one of the premier Web 2.0 software technologies. Ajax helps Internet-based applications perform like desktop applications—a difficult task, given that such applications suffer transmission delays as data is shuttled back and forth between your computer and servers on the Internet.
Agile software development	Agile software development is a set of methodologies that try to get software implemented faster and using fewer resources than previous methodologies. Check out the Agile Alliance (www.agilealliance.org) and the Agile Manifesto (www.agilemanifesto.org).
Refactoring	Refactoring involves reworking programs to make them clearer and easier to maintain while preserving their correctness and functionality. It’s widely employed with agile development methodologies. Many IDEs include <i>refactoring tools</i> to do major portions of the reworking automatically.

Fig. 1.27 | Software technologies. (Part 1 of 2.)

10. www.time.com/time/magazine/article/0,9171,1570810,00.html.

Technology	Description
Design patterns	Design patterns are proven architectures for constructing flexible and maintainable object-oriented software. The field of design patterns tries to enumerate those recurring patterns, encouraging software designers to <i>reuse</i> them to develop better-quality software using less time, money and effort.
LAMP	LAMP is an acronym for the set of open-source technologies that many developers use to build web applications—it stands for Linux, Apache, MySQL and PHP (or Perl or Python—two other languages used for similar purposes). MySQL is an open-source database management system. PHP is the most popular open-source server-side Internet “scripting” language for developing Internet-based applications.
Software as a Service (SaaS)	Software has generally been viewed as a product; most software still is offered this way. If you want to run an application, you buy a software package from a software vendor—often a CD, DVD or web download. You then install that software on your computer and run it as needed. As new versions of the software appear, you upgrade your software, often requiring significant time and at considerable expense. This process can become cumbersome for organizations with tens of thousands of systems that must be maintained on a diverse array of computer equipment. With Software as a Service (SaaS) , the software runs on servers elsewhere on the Internet. When that server is updated, all clients worldwide see the new capabilities—no local installation is needed. You access the service through a browser. Browsers are quite portable, so you can run the same applications on a wide variety of computers from anywhere in the world. Salesforce.com, Google, and Microsoft’s Office Live and Windows Live all offer SaaS. SaaS is a capability of cloud computing.
Platform as a Service (PaaS)	Platform as a Service (PaaS) , another capability of cloud computing, provides a computing platform for developing and running applications as a service over the web, rather than installing the tools on your computer. PaaS providers include Google App Engine, Amazon EC2, Bungee Labs and more.
Software Development Kit (SDK)	Software Development Kits (SDKs) include the tools and documentation developers use to program applications.

Fig. 1.27 | Software technologies. (Part 2 of 2.)

Figure 1.28 describes software product-release categories.

Version	Description
Alpha	An <i>alpha</i> version is the earliest release of a software product that’s still under active development. Alpha versions are often buggy, incomplete and unstable and are released to a relatively small number of developers for testing new features, getting early feedback, etc.

Fig. 1.28 | Software product-release terminology. (Part 1 of 2.)

Version	Description
Beta	<i>Beta</i> versions are released to a larger number of developers later in the development process after most major bugs have been fixed and new features are nearly complete. Beta software is more stable, but still subject to change.
Release candidates	<i>Release candidates</i> are generally <i>feature complete</i> and (supposedly) bug free and ready for use by the community, which provides a diverse testing environment—the software is used on different systems, with varying constraints and for a variety of purposes. Any bugs that appear are corrected, and eventually the final product is released to the general public. Software companies often distribute incremental updates over the Internet.
Continuous beta	Software that's developed using this approach generally does not have version numbers (for example, Google search or Gmail). The software, which is hosted in the cloud (not installed on your computer), is constantly evolving so that users always have the latest version.

Fig. 1.28 | Software product-release terminology. (Part 2 of 2.)

1.14 C++11 and the Open Source Boost Libraries



C++11 (formerly called C++0x)—the latest C++ programming language standard—was published by ISO/IEC in 2011. Bjarne Stroustrup, the creator of C++, expressed his vision for the future of the language—the main goals were to make C++ easier to learn, improve library building capabilities and increase compatibility with the C programming language. The new standard extends the C++ Standard Library and includes several features and enhancements to improve performance and security. The major C++ compiler vendors have already implemented many of the new C++11 features (Fig. 1.29). Throughout the book, we discuss various key features of C++11. For more information, visit the C++ Standards Committee website at www.open-std.org/jtc1/sc22/wg21/ and isocpp.org. Copies of the C++11 language specification (ISO/IEC 14882:2011) can be purchased at:

<http://bit.ly/CPlusPlus11Standard>

C++ Compiler	URL of C++11 feature descriptions
C++11 features implemented in each of the major C++ compilers.	wiki.apache.org/stdcxx/C%2B%2B0xCompilerSupport
Microsoft® Visual C++	msdn.microsoft.com/en-us/library/hh567368.aspx
GNU Compiler Collection (g++)	gcc.gnu.org/projects/cxx0x.html
Intel® C++ Compiler	software.intel.com/en-us/articles/c0x-features-supported-by-intel-c-compiler/

Fig. 1.29 | C++ compilers that have implemented major portions of C++11.