The continuing evolution of the web requires that browsers adapt. New browsers, devices, and systems used to access content online regularly continue to replace old ones. Not surprisingly, this has resulted in a great amount of uneasiness for web developers. Content on websites designed today may potentially not be accessible as new technologies and standards emerge. Thus, one solution proposed to resolve this problem is better known as progressive enhancement. In progressive enhancement, attention is put towards making content accessible first, and enhancements coming later (Dwyer, 2009). Websites are developed in 3 layers; the developer does not usually proceed to one without the current one being functional.

In the first layer, also referred to as the structural layer (Desruelle, Blomme, and Gielen, 2011), attention is put on the actual HTML being displayed properly (Dwyer, 2009). Those with the most primitive browsers should be able to view content. For example, one should be able to view text, though without the special fonts with text-based browsers such as Lynx. Although this appearance may not be particularly engaging or user-friendly, the content is still accessible to everybody.

In the second layer, also known as the presentation layer (Boudreaux, 2012), more focus is put on the appearance of the web page. Styling, usually through CSS is added. Examples are background colors, stylized text, and gradients. Images can also be added. Those with newer browsers should be able to view more stylized text if the web designer has enabled it. Ideally, if components added in this second layer are removed, the user should still be able to access the web page’s content.

The third and last layer, sometimes referred to as the “client-side scripting layer”, enhancements such as JavaScript and Flash are added (Boudreaux, 2012). The web designer must take great care as to add these components to the system in an unobtrusive manner, usually through external files. As is the case with the second layer, the removal of these components should not hamper the accessibility of the web page’s content.

In conclusion, practicing progressive enhancement when designing web sites ensures that content can be viewed by a wide spectrum of browsers, devices, and systems. From a software engineering perspective, I also believe that websites designed using this method are also easier to debug because it somewhat mirrors an iterative and incremental development style, in which systems are refined in phases before proceeding to the next stage.

References:

Boudreaux, R. (2012) ‘Understanding Progressive Enhancement Techniques In Web Design’, *TechRepublic* [Online]. Available from: <http://www.techrepublic.com/blog/web-designer/understanding-progressive-enhancement-techniques-in-web-design/> (Accessed: 7 June 2014)

Desruelle, H., Blomme, D. & Gielen, F. (2011) ‘Adaptive mobile web applications through fine-grained progressive enhancement’, *Adaptive 2011*, The Third International Conference on Adaptive and Self-Adaptive Systems and Applications, pp. 51.

Dwyer, S. (2009) ‘Progressive Enhancement: What It Is, And How To Use It?’, Smashing Magazine [Online]. Available from: <http://www.smashingmagazine.com/2009/04/22/progressive-enhancement-what-it-is-and-how-to-use-it/> (Accessed: 7 June 2014)