

Human Factors in Software Development

The first two articles in this series introduced the human factors profession and provided examples of human factors contributions in various industries. This article, Part 3, will focus on one specific industry, software development, and show how human factors engineers participate in the software development process. (Here are links to [Part 1](#) and [Part 2](#), if you want to refer to them.)

As you read this article, be aware that there are different software development methodologies. If you know them, you may be familiar with names such as waterfall or agile. A discussion of these methodologies is beyond the scope of this article. But keep in mind that the activities discussed in this article may vary in their relevancy depending on the methodology a particular team follows. Also, while this article is written from a human factors perspective, there are other disciplines, such as user experience design and information architecture, that bring similar and overlapping skills to software development.

Analysis

It's not wise to just jump in and start writing code, especially on large projects. First, you need to determine the functionality of the software (what is it going to do?), the budget, the schedule, the team, and so on. The team will make some high-level technical decisions as well, such as the platform/operating system, the conceptual design of the application, and the data model. All of this occurs in the analysis phase.

During this phase, human factors engineers provide input based on how the software will be used. In doing so, they represent the user. One key activity they may undertake is a **task analysis**, which is defined here in the context of a web site but applies to all software development:

Task analysis is the process of learning about ordinary users by observing them in action to understand in detail how they perform their tasks and achieve their intended goals. Tasks analysis helps identify the tasks that your website and applications must support and can also help you refine or re-define your site's navigation or search by determining the appropriate content scope.¹

There are well-developed methodologies for conducting task analyses, and they can be informal and simple or quite complex. Two common techniques are:

- Hierarchical task analysis, which identifies the user's goals, breaks the activities required to accomplish these goals into hierarchical steps and sub-steps, and then optimizes the process.

¹ <http://www.usability.gov/how-to-and-tools/methods/task-analysis.html>, 9 June 2015

The result is valuable documentation for the entire team.² Frederick Taylor's pioneering time-and-motion studies, conducted in the early 1900s, could be considered an early example of hierarchical task analysis.³

- Cognitive task analysis "is the study of what people know, how they think, how they organize and structure information, and how they learn when pursuing an outcome they are trying to achieve."⁴ Cognitive task analyses dive deeper than hierarchical analyses and help us understand what is going on inside the user's head. They are useful for understanding performance differences between novices and experts, effects of workload, mental models (see [Simplifying Complexity, Part 2](#)), and approaches to troubleshooting.⁵

A related activity is a **user analysis**, which determines user characteristics. Human factors engineers may interview users, conduct focus groups, analyze data (such as web metrics), or perform other research. The results are typically user personas, which are profiles of typical users, or user stories, which are scenarios of how users typically interact with the software or web site. This information is useful for ensuring that the software design and functionality meet user needs and can also be used during quality assurance testing to develop what are called test cases -- sequences of actions users are likely to perform.

Finally, human factors engineers can perform a **content analysis**, which is an examination of content that is presented in the web site or application. Typically, a sample of the content is evaluated. For example, content analysis provides helpful input for organizing a complex web site. According to *Information Architecture for the World Wide Web*, content analysis reveals "patterns and relationships within content and metadata that can be used to better structure, organize, and provide access to... content."⁶

Design

Once analysis is completed the team moves to the design phase.

Prototyping is a key aspect of design. Prototyping varies from throwaway prototyping (often just sketches on paper) to wireframes to functional prototypes written in a dedicated prototyping language. The idea is to mock up the user interface before devoting resources to building the real thing. Wireframes can include some functionality, such as links between screens. Here's an example of a wireframe.⁷ Wireframes are designed to look like sketches to indicate that they are drafts.

² <http://www.uxmatters.com/mt/archives/2010/02/hierarchical-task-analysis.php>, 9 June 2015

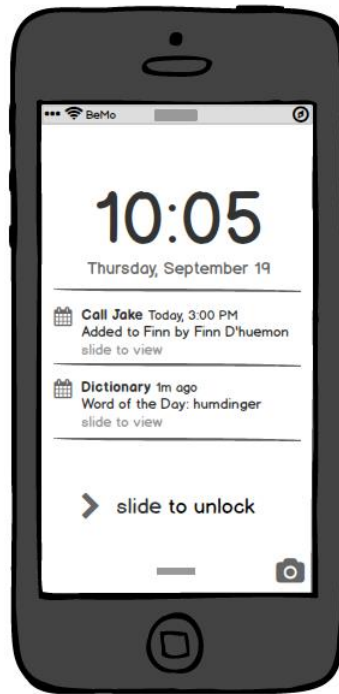
³ <http://www.mbsportal.bl.uk/taster/subjareas/busmanhist/mgmtthinkers/taylor.aspx>, 9 June 2015

⁴ <http://www.nwlink.com/~donclark/hrd/isd/cognitive-task-analysis.html>, 9 June 2015

⁵ <http://www.usabilitybok.org/cognitive-task-analysis>, 9 June 2015

⁶ Morville, P. & Rosenfeld, L. (2007). *Information Architecture for the World Wide Web*, 3rd Edition. Sebastopol, CA: O'Reilly Media, Inc. p. 242.

⁷ <https://balsamiq.com/products/mockups/>, 9 June 2015



Once a prototype is built, it can be tested. Testing can be as simple as getting feedback from users on paper prototypes. Higher fidelity testing might require users to perform tasks using functional prototypes.

There are other ways to solicit user input during the design phase. One common technique is **card sorting**, which "can provide insight into users' mental models, illuminating the ways they often tacitly group, sort, and label tasks and content in their own heads."⁸ Card sorting can be quite low tech. Users are given a stack of index cards. Each card has a term written on it, such as user interface label. Users organize the cards into groups and may be asked to label the groups."⁹ Card sorting can help determine the structure and organization for a software application or web site.

The culmination of the design phase is **specification writing**. The specification is the blueprint for the software application or web site. There is strong agreement among software development professionals that having a good specification is key to efficient, effective software development.

Quality Assurance

After a software product is coded it is tested. This testing is often called quality assurance, and is intended to ensure that the software or web site was built according to the specification and functions

⁸ Morville, P. & Rosenfeld, L. (2007). *Information Architecture for the World Wide Web*, 3rd Edition. Sebastopol, CA: O'Reilly Media, Inc. p. 255.

⁹ <http://www.usability.gov/how-to-and-tools/methods/card-sorting.html>, 9 June 1015

properly (i.e., doesn't have any bugs). While human factors engineers may review the product during the testing phase, they normally do not play a key role in quality assurance.

One type of testing where human factors is more likely to become involved in **accessibility testing**, which is intended to ensure that people with disabilities are able to use the software. Accessibility should be considered during the design phase, of course, and verified during testing. Accessibility is generally assessed against established standards, such as [W3C](#).

Usability testing can occur in the design phase, as discussed above, and can also occur during quality assurance.

Usability testing refers to evaluating a product or service by testing it with representative users.

Typically, during a test, participants will try to complete typical tasks while observers watch, listen and takes notes. The goal is to identify any usability problems, collect qualitative and quantitative data and determine the participant's satisfaction with the product.¹⁰

Findings from usability testing during quality assurance may lead to improvement in future versions of the software.

Striving for Simplicity

An [earlier series of articles](#) looked at ways to simplify the complexity that has become inherent in modern software and presented several techniques for dealing with complexity. This article has discussed the process that experts such as human factors engineers follow in implementing such techniques. By providing design input based on a sound initial analysis and performing usability testing, human factors professionals are the voice of the user in software development.

Dr. Craig Rosenberg is an entrepreneur, human factors engineer, computer scientist, and expert witness. You can learn more about Dr. Rosenberg and his expert witness consulting business at www.ui.expert

Craig Rosenberg, Ph.D.

www.ui.expert

craig@globaltechnica.com

206-552-9898

¹⁰ <http://www.usability.gov/how-to-and-tools/methods/usability-testing.html>, 9 June 2015